
Reliability-Based Optimization for Multidisciplinary System Design**Dhanesh Padmanabhan****Publication Date**

02-07-2003

License

This work is made available under a All Rights Reserved license and should only be used in accordance with that license.

Citation for this work (American Psychological Association 7th edition)

Padmanabhan, D. (2003). *Reliability-Based Optimization for Multidisciplinary System Design* (Version 1). University of Notre Dame. <https://doi.org/10.7274/2r36tx33k39>

This work was downloaded from CurateND, the University of Notre Dame's institutional repository.

For more information about this work, to report or an issue, or to preserve and share your original work, please contact the CurateND team for assistance at curate@nd.edu.

RELIABILITY-BASED OPTIMIZATION FOR MULTIDISCIPLINARY
SYSTEM DESIGN

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

by

Dhanesh Padmanabhan, B.Tech, M.S.M.E.

Dr. Stephen M. Batill, Director

Graduate Program in Aerospace and Mechanical Engineering
Notre Dame, Indiana
July 2003

RELIABILITY-BASED OPTIMIZATION FOR MULTIDISCIPLINARY SYSTEM DESIGN

Abstract

by

Dhanesh Padmanabhan

Reliability-Based Optimization (RBO) for engineering design deals mainly with two design attributes, namely the merit, for example cost, and the reliability of the design. In this work the class of design problems which are considered, are designs characterized by a minimum merit function and that satisfy certain reliability constraints. The reliability constraints are typically constraints on the probabilities of failure due to component failure events or a system failure event. These are obtained using standard reliability analysis techniques such as First Order Reliability Method (FORM), Second Order Reliability Methods (SORM) and Monte Carlo Simulation (MCS) techniques.

The reliability analysis and RBO are very expensive for multidisciplinary systems consisting of various disciplines that are dependent on each other or coupled, for example, an aeroelastic structure. Hence, the primary goal of the research is to develop efficient methodologies that perform RBO for multidisciplinary systems. The methodologies considered incorporates a Concurrent Subspace Optimization technique that allows concurrent design optimization in each discipline. The methodologies also incorporate approximation concepts to reduce the computational costs. There are essentially two methodologies, one that uses a traditional reliability analysis method and the other that uses a new reliability analysis method geared towards reduction of computational expenses for coupled multidisciplinary problems. A new reliability analysis tool based on

Trust Region methods was developed for the latter case. Both methodologies were applied to multidisciplinary test problems and about 20%-30% computational savings were observed.

A second goal of the research was to investigate the use of Monte Carlo Simulation (MCS) techniques for reliability analysis in RBO, that are more accurate but more expensive than FORM or SORM. In this work, conditional expectation MCS was selected over indicator-based MCS techniques based on smoothness criteria and the availability of analytic sensitivities. A MCS-based RBO methodology was developed and successfully implemented to problems with both component and series failure events. It was observed that designs with significantly lower merit functions were obtained for the application problems considered, compared to a FORM-based RBO approach. It was also observed that the computational costs were extremely high for one of the application problems. Some suggestions for future research are made regarding development of efficient methodologies for the MCS-based RBO.

To my late mother Sudha, father Padmanabhan,
sister Varuna and brother-in-law Kumaran.

CONTENTS

FIGURES	vii
TABLES	x
ACKNOWLEDGEMENTS	xii
SYMBOLS	xiv
CHAPTER 1: INTRODUCTION	1
1.1 Traditional Design Optimization	1
1.2 Optimization Under Uncertainty	3
1.3 Research Objectives	6
1.3.1 Reliability-Based Multidisciplinary Optimization Methodologies	6
1.3.2 Robust Computation of Most Probable Point	7
1.3.3 Monte Carlo Simulation in Reliability-Based Optimization	7
1.4 Overview of Dissertation	8
CHAPTER 2: TRADITIONAL DESIGN OPTIMIZATION	10
2.1 Problem Formulation and Terminology	10
2.2 Multidisciplinary Analysis	11
2.3 Multidisciplinary Design Optimization (MDO)	14
2.3.1 Concurrent Subspace Optimization	14
2.3.2 Collaborative Optimization	17
2.4 Summary	18
CHAPTER 3: UNCERTAINTY IN SIMULATION-BASED DESIGN	20
3.1 Classification of Uncertainties	20
3.1.1 Variation Type Uncertainties	21
3.1.2 Uncertainties in Decision Making	21
3.1.3 Modeling and Simulation Uncertainties	22
3.2 Modeling Uncertainty	22
3.2.1 Probability Theory	23

3.2.2	Convex Models of Uncertainty	24
3.2.3	Non-Probabilistic Techniques	25
3.2.4	Fuzzy Set Theory	26
3.2.5	Propagation of Uncertainty in a Multidisciplinary Systems	26
3.3	Design Optimization Under Uncertainty	27
3.3.1	Robust Design	27
3.3.2	Reliability-Based Design	29
3.4	Probabilistic Reliability Analysis	31
3.5	Summary	35
 CHAPTER 4: RELIABILITY-BASED OPTIMIZATION		37
4.1	RBO Formulation	37
4.1.1	Merit and Constraints	38
4.1.2	Random Variables and Design Variables	40
4.1.3	Computational Issues	41
4.1.4	Screening of Hard Constraints	42
4.2	Alternative RBO Formulations	43
4.2.1	Uni-level RBO	43
4.2.2	Performance Measure Approach (PMA)	44
4.2.3	Semi-infinite Optimization	45
4.2.4	Miscellaneous Approaches	46
4.3	Approximations Concepts in RBO	46
4.3.1	Limit State Approximations	47
4.4	Summary	49
 CHAPTER 5: RBO METHODOLOGIES FOR MULTIDISCIPLINARY PROBLEMS		51
5.1	Traditional RBO	51
5.2	A Concurrent Subspace Optimization Framework for RBO	53
5.2.1	Exact Deterministic and Reliability Analysis	54
5.2.2	Approximation Building Phase	55
5.2.3	Approximate RBO	56
5.3	Improved CSSO Approach for RBO	57
5.3.1	A CSSO Approach for rSA	58
5.3.2	A Local CA Based MPP Search	60
5.4	Bounded MPP Searches	62
5.4.1	Trust Region-based Equality Constrained Optimization	64
5.4.2	Elastic Sequential Quadratic Programming	74
5.5	Summary	75
 CHAPTER 6: MONTE CARLO SIMULATION IN RBO		77
6.1	Monte Carlo Simulation Techniques	77
6.1.1	Indicator-based MCS	78
6.1.2	Conditional Expectation-based MCS	80
6.1.3	Series System Probability of Failure	86
6.1.4	Sensitivities of Monte Carlo Simulations	87

6.2	MCS-Based RBO Methodology	90
6.3	Summary	91
 CHAPTER 7: IMPLEMENTATION STUDIES		93
7.1	RBO Application Problems	94
7.1.1	Analytic Test Problem	94
7.1.2	Control Augmented Structures Problem	95
7.1.3	High Performance, Low-Cost Structure Problem	98
7.2	RBO-CSSO Implementation Studies	101
7.2.1	Analytic Test Problem	101
7.2.2	Controls Augmented Structure Problem	103
7.2.3	High Performance, Low-Cost Structure	108
7.2.4	Summary	115
7.3	MPP Search Implementation Studies	116
7.3.1	Test Problems and Results	118
7.3.2	Summary	125
7.4	RBO-CSSO2 Implementation Studies	127
7.4.1	Analytic Test Problem	127
7.4.2	Control Augmented Structure	129
7.4.3	High Performance, Low-Cost Structure	133
7.4.4	Summary	140
7.5	MCS-based RBO Implementation Studies	141
7.5.1	Analytic Test Problem	142
7.5.2	Control Augmented Structures Problem	146
7.5.3	Summary	148
 CHAPTER 8: CONCLUSIONS		150
8.1	RBO-CSSO Approach	152
8.2	Bounded MPP Search Algorithms	153
8.3	Multidisciplinary MPP Search Methodologies	153
8.4	MCS-based RBO	154
8.5	Recommendations for Future Work	155
 APPENDIX A: ROSENBLATT TRANSFORMATION AND ITS INVERSE . . .		158
 APPENDIX B: HL-RF FAMILY OF ALGORITHMS		160
B.1	Basic Algorithm	160
B.2	Implementation of iHL-RF	161
 APPENDIX C: CONTROL AUGMENTED STRUCTURES		163
C.1	System Analysis	163
C.1.1	Structures Subsystem	164
C.1.2	Controls Subsystem	164
C.2	Sensitivity Analysis	166

C.2.1	Structures Sensitivities	166
C.2.2	Controls Sensitivities	167

FIGURES

2.1	Coupled System Analysis - N^2 Diagram	12
2.2	Concurrent Subspace Optimization Approach for Traditional Design Optimization	15
3.1	Design for Robustness	28
3.2	Design for Reliability	29
3.3	First Order and Second Order Reliability Methods	34
4.1	Cost Vs Risk	39
5.1	Reliability Analysis in a Multidisciplinary Problem	52
5.2	Flowchart of RBO-CSSO Framework	54
5.3	A CSSO Approach for MPP Searches	58
5.4	Local CA Based MPP Searches	60
5.5	Bounded MPP Search	62
5.6	Flowchart of the Trust Region-based Equality Constrained Optimization .	65
6.1	Standard Monte Carlo Simulation Technique	78
6.2	An Importance Sampling-based MCS Technique	79
6.3	Directional Simulation	81
6.4	Multiple Roots in Directional Sampling	82
6.5	Component Reliability Using Axis Orthogonal Simulation	83
6.6	Parallel System Reliability Using Axis Orthogonal Sampling	84
7.1	Control Augmented Structure	95
7.2	Coupling in Control Augmented Structures	96

7.3	High Performance, Low-Cost Structure Problem	98
7.4	System Analysis for the High Performance, Low-Cost Structure Problem	99
7.5	Iteration History in \mathbf{d} Space for the RBO-CSSO Implementation of the Analytic Test Problem	102
7.6	RBO-CSSO Convergence Plots for the Control Augmented Structure Problem	105
7.7	RBO-CSSO Convergence Plots for the Control Augmented Structure Problem (continued)	106
7.8	Traditional RBO Convergence Plots for High Performance, Low-Cost Structure Problem	111
7.9	Traditional RBO Convergence Plots for High Performance, Low-Cost Structure Problem (Continued)	112
7.10	RBO-CSSO Convergence Plots for High Performance, Low-Cost Structure Problem	113
7.11	RBO-CSSO Convergence Plots for High Performance, Low-Cost Structure Problem (Continued)	114
7.12	MPP Search Iteration History in X Space for the Analytic Test Problem at Initial Design	128
7.13	RBO-CSSO2 Convergence Plots for the Analytic Test Problem	130
7.14	RBO-CSSO2 Convergence Plots for the Analytic Test Problem (Continued)	131
7.15	Comparison of CA Calls for the Control Augmented Structures Problem .	133
7.16	Traditional RBO Convergence Plots for Coupled High Performance, Low-Cost Structure Problem	136
7.17	Traditional RBO Convergence Plots for Coupled High Performance, Low-Cost Structure Problem (Continued)	137
7.18	RBO-CSSO2 Convergence Plots for Coupled High Performance, Low-Cost Structure Problem	138
7.19	RBO-CSSO2 Convergence Plots for Coupled High Performance, Low-Cost Structure Problem (Continued)	139
7.20	Comparison of CA calls for Coupled High Performance, Low-Cost Structure Problem	140
7.21	Component MCS for Analytic Test Problem at Initial Design	143
7.22	Convergence Plots for Analytic Test Problem - Case A	144
7.23	Series System MCS for the Analytic Test Problem at Initial Design	145

7.24	Convergence Plots for the Analytic Test Problem - Case B	146
7.25	Designs for Analytic Test Problem .	147
7.26	Convergence Plots for Control Augmented Structure Problem	149

TABLES

7.1	LOAD COEFFICIENTS	100
7.2	LOAD COEFFICIENTS (CONTINUED)	101
7.3	DESIGNS AND COMPUTATIONAL COMPARISON FOR THE CONTROL AUGMENTED STRUCTURE	104
7.4	DESIGN VARIABLE MOVE-LIMIT STUDY FOR CONTROL AUGMENTED STRUCTURE PROBLEM	107
7.5	DESIGNS AND COMPUTATIONAL COMPARISON FOR UNCOUPLED HIGH PERFORMANCE LOW-COST STRUCTURE PROBLEM .	110
7.6	DESIGN VARIABLE MOVE-LIMIT STUDY FOR HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM	115
7.7	FUNCTION AND GRADIENT CALLS FOR ALL ALGORITHMS FOR THE ANALYTIC PROBLEMS	119
7.8	MOST PROBABLE POINTS FOR THE CONTROL AUGMENTED STRUCTURE LIMIT STATES	123
7.9	BOUNDED SOLUTIONS FOR THE CONTROL AUGMENTED STRUCTURE LIMIT STATES	124
7.10	FUNCTION AND GRADIENTS CALLS FOR ALL ALGORITHMS FOR CONTROL AUGMENTED STRUCTURE PROBLEM	126
7.11	COMPUTATIONAL COMPARISONS FOR CONTROL AUGMENTED STRUCTURE PROBLEM	132
7.12	UNCERTAIN VARIABLE MOVE-LIMIT STUDY FOR CONTROL AUGMENTED STRUCTURE PROBLEM	132
7.13	DESIGNS AND COMPUTATIONAL COMPARISON FOR COUPLED HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM	135
7.14	DESIGN VARIABLE MOVE-LIMIT STUDY FOR COUPLED HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM	141

7.15 UNCERTAIN VARIABLE MOVE-LIMIT STUDY FOR COUPLED HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM	142
7.16 DESIGNS FOR CONTROL AUGMENTED STRUCTURE PROBLEM .	148

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor, Dr. Stephen M. Batill, for his unwavering support and guidance during my stay at Notre Dame. He has given me a lot of freedom in my research work and been extremely supportive and understanding during the tough times.

I thank Dr. John E. Renaud for his guidance and for all the help he has provided me over these years. I gratefully acknowledge the faculty members at Notre Dame who taught me and helped me develop the skills and increase my knowledge in various branches of applied science and engineering. I specially thank Dr. Billie F. Spencer Jr. whose course in Structural Reliability marked a turning point in my research. I would like to thank Dr. Ravindra V. Tappeta of General Electric, Corporate Research and Development Center and Dr. Scott A. Burton of General Electric, Aircraft Engines who played a crucial role in my professional development.

I extend my thanks to my readers comprising of Dr. Stephen M. Batill, Dr. John E. Renaud, Dr. Michael M. Stanisic and Dr. Timothy C. Ovaert, for their help in the successful completion of this dissertation. I appreciate the administrative help provided by the department administrative assistants Ms. Nancy Davis, Ms. Evelyn Addington, Ms. Judith Kenna, Ms. Lisa Tranberg and Ms. Nancy O'Connor. I would like to thank the National Science Foundation, the Center for Applied Mathematics and Aerospace and Mechanical Engineering Department for their financial support.

Last but not the least, I thank all my lab members - Stacey, Victor, Will, Andres, Harish, Shawn and Alejandro, my buddies and partners-in-crime over these five years

at Notre Dame - Gaurav, Shyni, Smarajit, Zoltan, Saibal, Ajay, Ravikiran, Sadia, Ajit, Jindal, Debashish, Raja, Arvind, Deepak, Snehashish, Emilia, Eddie, Elke, Sameer, Krishnan, Vikas, Erzsebet, Peter, Dylan, Elisa, Guillermo, Brendan, Rohit, Kini to name a few, and other folks in this wonderful cozy Michiana region - Mickey, John, Mario, Amber, and many others.

SYMBOLS

d	Design Variables
p	Fixed Parameters
y	Intermediate State Variables (SVs)
<i>f</i>	Merit Function
g ^R	Hard or Critical Constraints
<i>N_{hard}</i>	Number of Hard Constraints
g ^D	Soft or Deterministic Constraints
<i>N_{soft}</i>	Number of Soft Constraints
d _l	Lower Bounds for d
d _u	Upper Bounds for d
y _{cvgd}	Converged SVs for a coupled system
y ^{nl,i}	SVs non-local to CA _i
X	Random Variables
<i>f_X</i>	Probability density function (p.d.f.) of X
<i>F_X</i>	Cumulative distribution function (c.d.f.) of X
$\mu_{\mathbf{X}}$	Mean of X
$\sigma_{\mathbf{X}}$	Standard Deviation of X
θ	Distribution Parameters
η	Deterministic Design Variables
U	Standard Normal random variables
<i>T</i>	Transformation from X to U

\mathbf{u}^*	Most Probable Point (MPP) in \mathbf{U} -space
\mathbf{x}^*	\mathbf{x} corresponding to \mathbf{u}^*
\mathbf{q}	Reliability Parameters for RBO
P_i	Probability of Failure for i^{th} failure mode
β_i	Reliability Index for i^{th} failure mode
Φ	Uni-variate Normal or Gaussian c.d.f.
ϕ	Uni-variate Normal or Gaussian p.d.f.
\mathbf{g}^{rbo}	Reliability Constraints
$\beta_{reqd,i}$	Required Reliability Index for i^{th} failure mode
$P_{all,i}$	Allowable Probability of Failure for i^{th} failure mode
$\tilde{\mathbf{g}}^R$	Limit State Approximations
$\tilde{\mathbf{g}}^{rbo}$	Approximations to \mathbf{g}^{rbo}
$\mathbf{y}_{cnvgd}^{nl,i}$	Approximations to converged $\mathbf{y}^{nl,i}$
ψ	Distance function in MPP search i.e. $\frac{1}{2}\mathbf{u}^T\mathbf{u}$
\mathbf{z}	Represents either \mathbf{x} or \mathbf{u}
\mathbf{z}_l	Lower Bounds for \mathbf{z}
\mathbf{z}_u	Upper Bounds for \mathbf{z}
Δ	Trust Region Radius
λ	Lagrangian Multiplier
ρ	Penalty Parameter
\mathbf{s}	Trial Step
\mathbf{s}_n	Normal Step
\mathbf{s}_t	Tangential Step
L	Lagrangian
\tilde{L}	Approximate Lagrangian
Θ	Augmented Lagrangian

$\tilde{\Theta}$	Approximate Augmented Lagrangian
$pred$	Predicted Reduction
$ared$	Actual Reduction
κ	Reduction Ratio
\mathbf{H}	Hessian of Lagrangian
ϵ	Convergence Tolerance
ξ	Penalty Parameter in Elastic Sequential Quadratic Programming (ESQP)
\mathbf{V}	Simulation Random Variable
$h_{\mathbf{V}}$	Sampling Density Function
P	Probability of Failure
Ω_f	Failure Region
\hat{P}	Estimate of Probability of Failure
I	Indicator Function
R	Radial Random Variable
\mathbf{A}	Angular Random Variable
K_n	χ^2 c.d.f. with n degrees of freedom
\mathbf{G}	Gram-Schmidt Orthogonalization Linear transformation matrix
w_j	Weighting functions in Stratified Sampling
τ	Limit State Parameter in \mathbf{X} or \mathbf{U} space
$N(\mu, \sigma)$	Normal random variable with mean μ and standard deviation σ
$LN(\mu, \sigma)$	Lognormal random variable with mean μ and standard deviation σ
$U(\mu, \sigma)$	Uniform random variable with mean μ and standard deviation σ

CHAPTER 1

INTRODUCTION

In this chapter, an introduction to traditional optimization and optimization under uncertainty is given. The key areas of interest are identified while discussing optimization under uncertainty and subsequently research objectives are defined to address these issues. Finally, an overview of the following chapters in this dissertation is presented.

1.1 Traditional Design Optimization

Since the advent of high speed computing in the 1950s numerical simulations for complex engineering problems have become very popular. This has seen the emergence of simulation tools like Finite Element Methods for analysis of complex engineering problems, that are typically based on ordinary and partial differential equations. The development of these simulation tools provided an opportunity to improve existing designs and identify better designs early in the design process of machines and mechanisms. The process of improvement of these designs basically forms the scope of *design optimization*. Since the development of Linear Programming in 1948, various numerical optimization techniques have been developed. The application of these numerical optimization techniques for simulation-based design optimization began during 1950s (see Venkayya[1]) and since have been extended to solve different engineering design optimization problems.

The main criteria typically used in a traditional design optimization for measuring

the effectiveness of a design are cost and performance. An example of a traditional design optimization problem is a structural optimization problem with cost proportional to weight of the structure and with the stress or deflection due to some given load as the performance characteristics. Traditional design optimization can be cast as a multi-objective problem, minimize cost and maximize performance. A special case of the multi-objective problem is the single objective problem, minimize cost subject to constraints on performance characteristics. Hence the example optimization problem can be cast as minimize weight subject to constraints on stresses. The scope of this dissertation is restricted to such single-objective design optimization problems. The design variables in a design optimization problem are quantifiable parameters of the artifact that have to be determined. These design variables can be either continuous, for example dimensions of a beam, or discrete, for example number of strands in a cable, or block, for example type of material, or a combination of the three. The constraints in a general optimization problem include inequality constraints and equality constraints. In this dissertation, problems with continuous design variables and inequality constraints are considered.

In multidisciplinary problems, for example an aeroelastic design of an aircraft wing, many disciplines or subsystems exist, for example controls, structures, fluids, aerodynamics etc. In these problems, the analysis tools of a discipline typically require the outputs from other disciplines. Hence these problems can be potentially coupled. For example, in an aircraft wing design problem the computation of the deflection (structures discipline) will require the aerodynamic loads or pressure distribution (aerodynamics discipline) that changes with the deflection of the wing. Hence the evaluation of a single design for such coupled problems will require several iterations of the coupled analysis tools. Hence a traditional design optimization, that treats the multidisciplinary analysis as a black-box, will be very expensive especially for coupled problems. Hence researchers have proposed various specialized optimization methodologies[2, 3] that employ decom-

position strategies and approximation concepts to reduce the computational costs for such traditional design optimization problems involving multidisciplinary systems. There have been many advances in multidisciplinary design optimization for aerospace engineering and certain automobile engineering applications. There is promise of extending the scope of this field to other engineering applications where there will be a need to account for multidisciplinary nature of the problem for obtaining improved designs in future. Hence this dissertation will mainly focus on multidisciplinary problems.

1.2 Optimization Under Uncertainty

Traditional design optimization does not account for the uncertainties present in the actual modeling and simulation, manufacturing processes and in the external conditions for the time the product is in service. These uncertainties can be classified as the ones that arise due to manufacturing variability like uncertainties in material properties and variability in external conditions like temperature, loading, modeling errors (simulations vs. real outcomes) and numerical errors in simulation-based analysis.

These uncertainties might cause large variations in certain performance characteristics of the artifact. Robust designs are designs at which the variations in performance characteristics are minimal[4]. These uncertainties also cause an increased chance of structural failure in artifacts that consist of some structure or machine. The chance of failure is especially high for designs obtained from traditional design optimization, since at these designs failure driven constraints such as stress constraints are usually active or close to being active. Reliable designs are designs at which the chance of failure of the product or artifact is low[5]. Robust and reliable designs are highly desirable because they have huge implications in commercial applications such as gaining greater market share for higher profits in the long run. Hence the issue of robustness and reliability are extremely important, but this dissertation will mainly focus on reliability-based design optimization

problems.

In Reliability-Based Optimization (RBO) problems[6], there is a trade-off between obtaining greater reliability and minimum cost, since greater reliability implies greater cost, but smaller reliability also implies greater cost due to failure costs. Hence there is an optimum reliability that can be achieved. In some applications, reliability requirements can be chosen by the manufacturer or the designer and such an optimum reliability can be set. There are also many applications where the reliability requirements are set by stringent regulatory or statutory laws, for example in designing buildings, bridges etc. Both these Reliability-Based Optimization problems can be stated as: minimize cost subjected to reliability constraints and other deterministic constraints.

To be able to perform a Reliability-Based Optimization, one has to characterize the important uncertain variables. It is desirable to reduce the number of the uncertain variables since the computational costs of computing the reliability of the design increase with the number of random variables. The probability distributions of the random uncertain variables also have to be established. This is mainly obtained using statistical models fit to existing data or by assuming probability distributions based on limited statistical information. After this, all the important failure modes have to be established. Probabilistic reliability analysis is used to obtain the reliability constraints with respect to these important failure modes. Some of the techniques used in the reliability analysis methods are the First Order Reliability Methods, Second Order Reliability Methods and Monte Carlo Simulation techniques[7, 8]. The most advanced and popular version of the First Order Reliability Method, the earliest of the three, was developed during the 1970's for probabilistically-based building codes in Europe[9, 10]. These reliability analysis methods typically require the computation of a Most Probable Point (MPP) of failure with respect to each failure mode. The Most Probable Point of failure is computed by solving a sub-optimization problem, which is basically an equality constrained optimization

problem. The cost function and other deterministic constraints are computed using a deterministic analysis, which is a single evaluation at the mean values of uncertain variables.

When an RBO is performed for multidisciplinary systems, the computational costs are extremely high especially for coupled systems. This is because in multidisciplinary systems the uncertainties propagate between analysis tools and hence it is important to perform complete multidisciplinary analysis for the reliability computations. This dissertation deals with extending Reliability-Based Optimization to multidisciplinary system design. The work mainly focuses on incorporating efficient decomposition strategies for reducing the computation costs associated with the evaluation of coupled multidisciplinary analysis.

The dissertation also addresses the issue of robustness of existing optimization algorithms for the computation of the MPP for cases when the failure mode is hardly active and when the MPP searches are infeasible because of additional strict bound constraints. The former case happens in large optimization problems where many critical and non-critical failure modes exist. The latter case is required for the implementation of the one of reliability-based multidisciplinary optimization methodology, which will be explained later.

The dissertation also focuses on the issue of inclusion of Monte Carlo simulation techniques in RBO. Monte Carlo techniques are the most accurate among all the reliability analysis methods but very limited work has been done to include them in a reliability-based optimization framework. The main concerns that arise in an RBO using Monte Carlo simulation techniques are the high computational expenses and the usability with gradient-based optimizers. The work presented in the dissertation mainly addresses the latter concern and some pointers are given to address the former concern.

1.3 Research Objectives

Having identified the main research issues for this work, the following research objectives for addressing these issues were set:

1. Develop efficient reliability-based multidisciplinary optimization methodologies.
2. Develop a robust optimization algorithm for the computation of the Most Probable Point.
3. Investigate the existing Monte Carlo simulation techniques and implement a Reliability-Based Optimization framework employing Monte Carlo simulation techniques.

The above research objectives are explained in detail in following part of this section.

1.3.1 Reliability-Based Multidisciplinary Optimization Methodologies

The main objective is to use decomposition strategies and approximation concepts to reduce the computational costs associated with using a coupled multidisciplinary analysis for Reliability-Based Optimization. Two levels of decomposition are possible; one at the main optimization level for the deterministic analysis and other at the reliability analysis level during the MPP searches. The level one approach, decomposes the problem in the design variable space. The level two approach, decomposes the problem in the random variable space. The work presented in this dissertation presents two methodologies that both employ a Concurrent Subspace Optimization framework for level one decomposition. The second methodology also implements a level two decomposition in the Most Probable Point searches. Existing multidisciplinary optimization approaches can be used for the decomposed MPP searches. A special approach was developed that exploits the special problem structure of the multidisciplinary MPP searches. Approximations were used along with the decomposed level one and two optimizations in a sequential fashion,

to account for the multidisciplinary interdependence and for approximate evaluation of reliability constraints.

1.3.2 Robust Computation of Most Probable Point

In the decomposed MPP searches, there are additional bound constraints or move-limits on the optimization variables of the MPP searches to account for the goodness or fidelity of the approximations that are used. When these move-limits are small, the equality constraint of the MPP search problem need not be satisfied. Existing popular algorithms for the MPP searches cannot handle such a case. Hence the objective here was to develop optimization methods that can handle such a case. Such an optimization method is highly useful since it can also be used as a tool for determining important failure modes in a problem. This can be achieved by setting the bound constraints based on a region of importance.

1.3.3 Monte Carlo Simulation in Reliability-Based Optimization

First order reliability methods are used very widely for Reliability-Based Optimization, but first order reliability methods can give inaccurate estimates of reliabilities for highly nonlinear problems and are known to suffer from some weaknesses[11]. Another shortcoming of first and second order reliability methods are that estimates of error bounds or confidence intervals cannot be obtained. Monte Carlo simulation techniques on the other hand do not suffer from these disadvantages but are very expensive. But the main concern that is addressed in this work is the usability of Monte Carlo simulation techniques with a gradient-based optimizer. The objective here is to investigate Monte Carlo simulation techniques for smoothness properties and availability of analytic derivatives, and to develop a Reliability-Based Optimization framework using certain selected Monte Carlo simulation techniques.

1.4 Overview of Dissertation

In Chapter 2, an optimization problem formulation for traditional design optimization will be presented, multidisciplinary systems will be discussed and some multidisciplinary design optimization methods will be discussed.

Chapter 3 begins with an explanation of different types of uncertainties and different uncertainty modeling techniques. Then, methodologies for design optimization under uncertainty are presented. Reliability analysis is explained in detail in this chapter.

In Chapter 4, the general Reliability-Based Optimization formulation and some pertinent issues are discussed. Then work done by other researchers are presented in two major sections that consist of explanations of various alternative Reliability-Based Optimization formulations and the use of approximation concepts.

Chapter 5 discusses the Reliability-Based Optimization methodologies for multidisciplinary problems, developed in this research work. In the first section, a traditional Reliability-Based Optimization methodology for multidisciplinary problems is presented and some of the issues are discussed. In the second section, a Concurrent Subspace Optimization framework for performing Reliability-Based Optimization for multidisciplinary problems is presented and discussed in detail. In the third section, an improved version of the framework presented in the second section is presented. This improved version incorporates decompositions strategies in the exact reliability analysis. The need for a bounded Most Probable Point search that can handle infeasibilities in Most Probable Point search formulation is established. In the fourth section, a couple of candidate algorithms for the bounded Most Probable Point search are discussed. First, a modified Trust Region-based Equality Constrained Optimization is presented and discussed in detail. Second, an Elastic Sequential Quadratic Programming approach is presented.

In Chapter 6, different kinds of Monte Carlo Simulation techniques are presented that can be classified as indicator-based and conditional expectation Monte Carlo Simulation

techniques. Next, a method to obtain probability of failure for a series failure event is discussed. Then, derivation of analytic sensitivities and pertinent issues related to their computations are discussed. Finally, a Reliability-Based Optimization framework using a specific Monte Carlo simulation technique is presented.

Chapter 7 consist of implementation studies and presents results of all the methodologies and algorithms developed in chapters 5 and 6.

In chapter 8, the advantages and disadvantages, mainly based on results from the implementation studies of the various methodologies developed in this research work, are discussed. Various outstanding research issues are discussed and recommendations for future work are made.

CHAPTER 2

TRADITIONAL DESIGN OPTIMIZATION

In this chapter, first a problem formulation for traditional design optimization problem will be presented and key terminologies will be established. Next, multidisciplinary systems will be discussed followed by a discussion of selected multidisciplinary optimization techniques used for traditional design optimization for multidisciplinary systems.

2.1 Problem Formulation and Terminology

The class of problems considered for traditional design optimization in this work are given by the Equations 2.1-2.4. In this problem, it is required to find the values of design variables, \mathbf{d} so that a minimum value of a merit function f is obtained while satisfying prescribed inequality constraints \mathbf{g}^R and \mathbf{g}^D and design variable bounds \mathbf{d}_l and \mathbf{d}_u .

$$\text{Minimize} \quad f(\mathbf{d}, \mathbf{p}, \mathbf{y}(\mathbf{d}, \mathbf{p})) \quad (2.1)$$

$$\text{s.t.} \quad g_i^R(\mathbf{d}, \mathbf{p}, \mathbf{y}(\mathbf{d}, \mathbf{p})) \geq 0 \quad \text{for } i=1,..,N_{hard} \quad (2.2)$$

$$g_j^D(\mathbf{d}, \mathbf{p}, \mathbf{y}(\mathbf{d}, \mathbf{p})) \geq 0 \quad \text{for } j=1,..,N_{soft} \quad (2.3)$$

$$\text{Bounds:} \quad \mathbf{d}_l \leq \mathbf{d} \leq \mathbf{d}_u \quad (2.4)$$

\mathbf{p} are the fixed parameters of the optimization problem. g_i^R is the i^{th} hard or failure driven constraint that models the critical component failure mode of the artifact. g_j^D is the j^{th} soft constraint that models a deterministic constraint due to other design considerations like cost, marketing etc. If $g_i^R < 0$ at a given design \mathbf{d} then the artifact is said to have

failed with respect to the i^{th} failure mode. The complete failure of the artifact depends on how all the failure modes contribute to what is known as the system failure of the artifact. On the other hand violations in soft constraints g_j^D due to inherent uncertainties is assumed to be non-critical to the design and hence acceptable. Though, a clear distinction is made between hard and soft constraints, traditional design optimization treats both these constraints in a similar fashion, and the failure of the artifact is not taken into consideration. This distinction is made to facilitate the introduction of concepts like reliability analysis and reliability-based optimization in later chapters. It has to be noted that equality constraints could also be present in the optimization formulation, but are not very common in traditional design optimization and hence have not been included.

The merit function and constraints in the above formulation are explicit expressions in terms of \mathbf{d} , \mathbf{p} and $\mathbf{y}(\mathbf{d}, \mathbf{p})$. $\mathbf{y}(\mathbf{d}, \mathbf{p})$ are intermediate quantities that are outputs of analysis tools used in predicting the performance characteristics of an artifact and are typically not explicit in \mathbf{d} and \mathbf{p} . These intermediate quantities will be referred to as State Variables (SVs).

2.2 Multidisciplinary Analysis

In Multidisciplinary problems, various disciplines exist, like controls, structures and aerodynamics discipline in an aeroservoelastic structure. The analysis tools corresponding to these disciplines are typically referred to as Contributing Analyses (CAs). In multidisciplinary problems, each CA often requires other SVs, i.e. outputs of other CAs, as inputs in addition to design variables and parameters. Figure 2.1 shows a general coupled system involving 3 CAs with inter-disciplinary coupling among them. To obtain the merit function and constraints, for a given set of design variables and parameters, one needs to obtain the SVs through a process called System Analysis (SA). The SA for a coupled system with SV feedback would require an iterative process to determine consistent SVs,

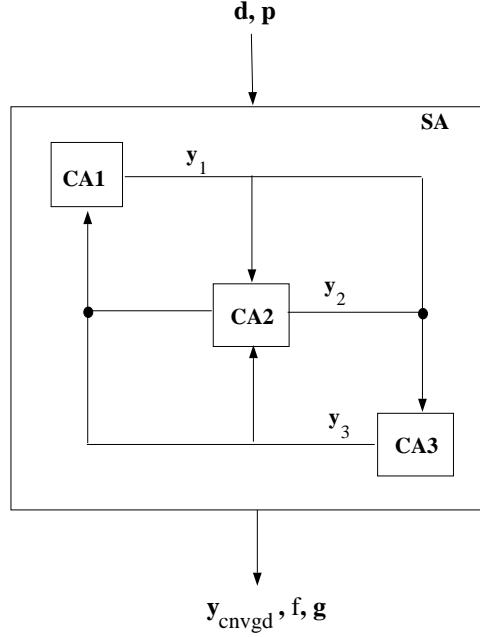


Figure 2.1. Coupled System Analysis - N^2 Diagram

$\mathbf{y}_{\text{cnvgd}}$. This can be mathematically formulated as a procedure to solve a system of nonlinear equations as given by the following equations for a multidisciplinary system with k CAs:

$$\mathbf{y}_1 = CA_1(\mathbf{d}, \mathbf{p}, \mathbf{y}^{nl,1})$$

$$\mathbf{y}_2 = CA_2(\mathbf{d}, \mathbf{p}, \mathbf{y}^{nl,2})$$

.

.

$$\mathbf{y}_k = CA_k(\mathbf{d}, \mathbf{p}, \mathbf{y}^{nl,k})$$

where $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k]^T$ and $\mathbf{y}^{nl,i}$ is the set of state variables required by CA_i from other CAs. $\mathbf{y}^{nl,i}$ will be referred to as the SVs non-local to CA_i . This system of equations can have more than one solution for a given \mathbf{d} and \mathbf{p} , but this is not very common. If the system equations are highly coupled, several iterations might be required. When the CAs are

expensive to evaluate, such a highly coupled SA becomes very expensive to evaluate.

The sensitivities of the converged SVs with respect to \mathbf{d} and \mathbf{p} can be obtained by solving the Global Sensitivity Equations (GSE)[12] obtained by the implicit function differentiation rule. GSEs provide a more accurate and efficient way of computing the sensitivities compared to a finite difference scheme, which typically requires expensive re-computations of the SA. The total sensitivity can be calculated using the following formula after a SA has been performed.

$$\Delta_d \mathbf{y} = \mathbf{C}^{-1} \nabla_d \mathbf{y} \quad (2.5)$$

$$\Delta_p \mathbf{y} = \mathbf{C}^{-1} \nabla_p \mathbf{y} \quad (2.6)$$

where Δ_d and Δ_p are the total differential operators with respect to \mathbf{d} and \mathbf{p} , whereas ∇_d and ∇_p are partial differential operators with respect to \mathbf{d} and \mathbf{p} . \mathbf{C} is the coupling matrix obtained from the Global Sensitivity Equations. The \mathbf{C} (which is a function of \mathbf{d} and \mathbf{p}) will be a matrix of dimension $M \times M$, where M is the length of \mathbf{y} , and will consist of various rectangular matrices and identity matrices as shown below

$$\mathbf{C} = \begin{bmatrix} I & -\nabla_{y_2}y_1 & \cdots & -\nabla_{y_k}y_1 \\ -\nabla_{y_1}y_2 & I & \cdots & -\nabla_{y_k}y_2 \\ \cdots & \cdots & \cdots & \cdots \\ -\nabla_{y_1}y_k & -\nabla_{y_2}y_k & \cdots & I \end{bmatrix} \quad (2.7)$$

$\nabla_d \mathbf{y}$, $\nabla_p \mathbf{y}$ and $\nabla_{y_j} \mathbf{y}_i$ ($i \neq j$) are first obtained and then Equations 2.5 and 2.6 are solved to get the total differentials or overall consistent sensitivities. The SA and sensitivities from solution of GSE can be used to perform the traditional design optimization 2.1-2.4, using many standard nonlinear programming techniques available[13].

2.3 Multidisciplinary Design Optimization (MDO)

When solving the traditional design optimization given by Equations 2.1-2.4 using multidisciplinary SA to obtain \mathbf{y} and GSE to obtain sensitivities $\Delta_{d\mathbf{y}}$, the approach can become fairly expensive for large systems with strongly coupled analysis tools that are computationally expensive to evaluate. Various MDO approaches have been developed for performing traditional optimization in an efficient manner[2].

A traditional approach is basically a “black-box” approach and typically referred to Nested ANalysis and Design (NAND) Optimization. Another popular approach is the Simultaneous ANalysis and Design (SAND) optimization where the design optimization is formulated such a way that both the optimization and SA, i.e. solving the system of equations, are performed simultaneously. This typically requires augmenting the design variables \mathbf{d} with \mathbf{y} and adding other equality constraints that make sure that the system of equations represented by the SA are satisfied. SAND is typically more efficient than NAND when expensive numerical root solving techniques are required for the SA in NAND. The common criticism about NAND and SAND approaches is that they do not allow design activities in individual disciplines. This is important from a concurrent engineering perspective. Hence to address this issue a number of other MDO approaches were developed. The important ones are the Concurrent SubSpace Optimization (CSSO)[14] and Collaborative Optimization (CO)[15]. These approaches make use of decomposition and concurrent optimization strategies. A detailed explanation of the CSSO approach, which is important for this research work, will be first presented. A brief explanation of the basic CO approach will then be given.

2.3.1 Concurrent Subspace Optimization

In a CSSO approach illustrated in Figure 2.2, the following basic steps are performed, starting from $i = 0$ and an initial design \mathbf{d}^0 :

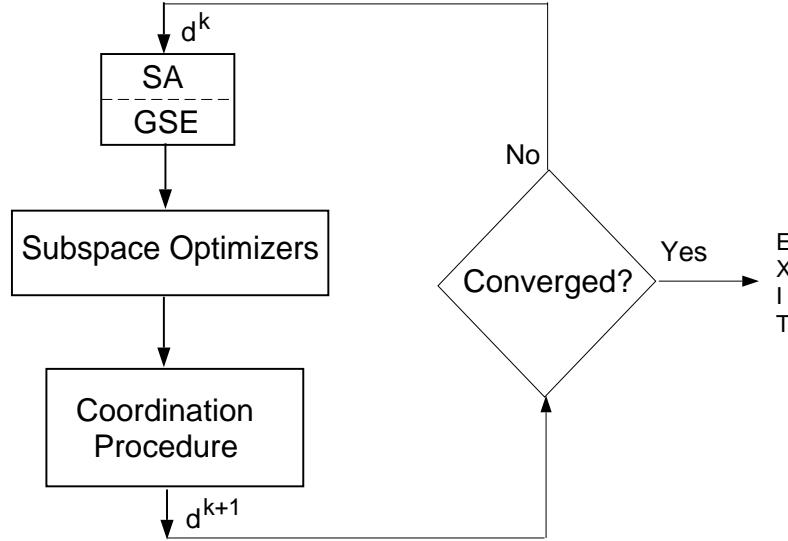


Figure 2.2. Concurrent Subspace Optimization Approach for Traditional Design Optimization

1. Perform SA and GSE at \mathbf{d}^i to obtain \mathbf{y} and $\Delta_d \mathbf{y}$.
2. Perform approximate optimization, that consists of various Subspace Optimizers (SSOs) and a Coordination Procedure(CP), to obtain a new design \mathbf{d}_{new} .
3. Check for convergence. If converged stop, else $i = i + 1$ and $\mathbf{d}^i = \mathbf{d}_{new}$ and go to step 1.

In the approximate optimization phase, there are SSOs for each discipline. The SSOs can take place in a parallel or concurrent fashion. The SSO can control only certain allocated design variables and use its CA alone during optimization. The multidisciplinary effect is approximately maintained during each SSO by using an approximation to the non-local state variables based on the most recent exact SA and sensitivity analysis. This is equivalent to having an approximation to the SA and combining with exact CAs. A new design is obtained from a Coordination Procedure that coordinates the results of the SSOs. The SSOs and CP are both subjected to move-limit constraints on design variables, to account for the fidelity of the approximations used in SSO.

The attractive features of CSSO are that it allows concurrent design in each discipline and it is usually computationally efficient for complex multidisciplinary problems with expensive SAs. One of the issues in CSSO is that the allocation of design variables among the SSOs is not very straightforward. It is usually desirable to allocate design variables that are important to a SSO. If unimportant design variables are allocated, a SSO can be infeasible at an infeasible design, from an optimization problem perspective, and hence will not generate enough information that will influence the overall design during the CP. This can also happen in multidisciplinary problems that are strongly influenced by only few disciplines. In general, there can be shared design variables between SSOs. In the original approach proposed by Sobieszczanski-Sobieski[14], the design variables of the SSOs had to be disjoint. The original approach due to Sobieszczanski-Sobieski was also shown to converge to pseudo-optimal solutions (See Wujek and Renaud[16]). This is mainly due to the CP used in the original approach.

There are various other CPs that have been proposed for CSSO. The CP proposed by Renaud and Gabriele[17] can account for shared design variables. This CP is implemented as an optimization problem solving the actual design optimization formulation using approximations for f and g that are constructed by fitting response surfaces to iteration histories obtained from all the SSOs. The main problem associated with this CP is that the approximations for f and g used in this CP are of low fidelity. This is mainly because the iteration histories are usually not well distributed to accurately capture the trends. The low fidelity of the approximations are known to cause failure of CP when large move-limits are used. Wujek and Renaud[16] used a move-limit management strategy to increase the robustness of the CSSO approach using this CP. Another disadvantage of this CP is that the optimization results from the SSOs have no influence on the CP and the SSOs are basically used only as a sampling schemes for creation of the response surfaces.

A Sequential Approximate Optimization (SAO) was proposed by Rodríguez et. al.[18] that addressed these issues but deviated from a CSSO approach substantially. The same basic steps are performed in SAO as in the CSSO, but the approximate optimization is implemented differently. In the approximate optimization, only one optimization is present and is same as the optimization done in CP proposed by Renaud and Gabriele. The approximations used for f and g in this approximate optimization are constructed by sampling exact CAs using Design of Experiment concepts[19] mixed with approximations to the non-local states, that was performed in a similar fashion as in SSOs in CSSO. The approximations of f and g constructed in this fashion are usually better than the approximations constructed from iteration histories. Though there is no concurrent design taking place in each discipline in a SAO, issues regarding allocation of design variables among SSOs are not present. The computational costs associated with the sampling of the CAs can be high, but this can be parallelized[20]. Both CSSO and SAO can be viewed as efficient implementations of the “black-box” i.e. the NAND approach. The main advantage of CSSO and SAO is that one has a consistent design even if the design optimization is stopped before convergence.

2.3.2 Collaborative Optimization

In CO, a system level optimization is performed in addition to Subspace Optimizations. There is no CP in a CO and no exact SA or GSE is ever required. Like in the CSSO, there is a SSO corresponding to each CA and each SSO is allowed to use only its corresponding CA. Each SSO is required to minimize a discrepancy function subject to those constraints that explicitly depend on the CA used in the SSO. The SSO is allowed to vary the design variables allocated to it and the non-local state variables of the CA. The discrepancy function is a cumulative function of the difference between shared design variables and their targets, difference between non-local state variables and their

targets and difference between the CA outputs and their targets. The SSOs are subject to move-limit constraints on all the optimization variables of the SSOs. The targets for the shared design variables and all the state variables are set at the system optimization level. In the system level optimization, the targets are set such that the merit f is minimized subjected to equality constraints that require the discrepancy functions corresponding to each SSO to be equal to zero. The system level optimization is also subject to move-limit constraints for all these targets.

It can be seen that the CO allows for design by each discipline in a concurrent fashion. Unlike CSSO, exact SA and GSE are never required. In a CO, the system of equations in the SA are simultaneously solved with the actual traditional design optimization. Hence this method can be seen as an efficient implementation of the SAND. One of the major disadvantages of CO is that one does not have a consistent design if the design optimization has to be stopped before convergence, and additional analysis have to be performed to attain consistency. A detailed discussion of merit and drawbacks of the CO is beyond the scope of this dissertation and interested readers can refer Alexandrov and Lewis[21] for further details.

2.4 Summary

In a traditional design optimization, it is usually required to minimize a merit function subject to inequality constraints and design variable bounds. The inequality constraints can be hard (i.e. failure critical) and soft or deterministic. The merit function and constraints explicitly depend on intermediate variables also called state variables.

In simulation-based design of multidisciplinary systems, these intermediate variables are obtained by the solution of system of nonlinear equations representing each analysis tools, typically known as Contributing Analysis (CA), which is usually referred to as System Analysis (SA). For coupled systems, the system analysis requires iterative root

solving methods. The sensitivities of the state variables can be efficiently obtained by solving Global Sensitivity Equations (GSE) that are based on the implicit differentiation rule.

A “black-box” approach to traditional design optimization using SA and GSE, also known as Nested ANalysis and Design (NAND) is expensive. A Simultaneous ANalysis and Design (SAND) can be performed that simultaneously solves the system of equations of the SA and the traditional design optimization problem. Concurrent Subspace Optimization (CSSO) and Collaborative Optimization (CO) are other MDO approaches that allow concurrent design in all disciplines. The CSSO approach is comprised of Subspace Optimizers (SSOs) corresponding to each discipline and a Coordination Procedure (CP) that coordinates the results of the SSOs. There are issues regarding allocating design variables among the SSOs and regarding the robustness of the CP. A Sequential Approximate Optimization (SAO) that replaces the SSOs and the CP with a single approximate optimization alleviates the shortcomings of the CSSO. The CSSO and SAO approach can be seen as efficient implementations of NAND. The CO approach is basically an efficient implementation of SAND. It consists of a system level optimizer that sets targets to subspace optimizers. It is formulated such a way that an exact SA is never required.

CHAPTER 3

UNCERTAINTY IN SIMULATION-BASED DESIGN

In this chapter, different types of uncertainties that arise in a simulation-based design problem will be presented. Then, different uncertainty modeling techniques will be discussed and uncertainty propagation in a multidisciplinary system will be discussed. Next, the major issues that arise due to these uncertainties in a traditional design optimization will be presented. Finally, probabilistic reliability analysis techniques used to quantify the reliability of a design will be discussed.

3.1 Classification of Uncertainties

Traditional design optimization does not take into consideration the inherent uncertainties present in certain quantities and also the uncertainties related to modeling and simulation. The main types of uncertainties, one encounters in design, are variations in parameters and design variable settings, uncertainties related to decision making or in other words uncertainty in design problem formulations, modeling and numerical errors associated with the analysis tools used. The term “uncertainty” is used in a very general context in this document for convenience purposes but different types of uncertainties exist and consistent terminologies are required. Different taxonomies of uncertainties have been presented by Parsons[22] and a taxonomy for uncertainty was recently proposed by Oberkampf et. al.[23] for the uncertainties that arise in engineering problems.

The main types of uncertainties that arise during the design of a product and during the

operation of the product, can be categorized as variation type uncertainties, uncertainties in decision making, uncertainties in modeling and simulation. A discussion of these uncertainties follow:

3.1.1 Variation Type Uncertainties

Variation type uncertainties can occur in the form of manufacturing tolerances, uncontrollable variations in external operating conditions. They are usually modeled as random phenomena, which are characterized by probability distributions that are constructed using relative frequency of occurrence of events, which require large amounts of information. Most often such information does not exist and designers usually make assumptions for the characteristics (mean, variances, correlation coefficients) of the random phenomena causing the variation. Such variation type uncertainties characterized by incomplete information can be represented using techniques such as Convex Models of Uncertainty, Interval Methods, Possibility Theory etc. Oberkampf et. al.[23] suggest that the variation uncertainties that can be probabilistically characterized be classified as *aleatory uncertainty* and the variations with incomplete information be classified under *epistemic uncertainty*, i.e. uncertainties arising due to lack of knowledge or information.

3.1.2 Uncertainties in Decision Making

Every design problem can be formulated as a multi-objective optimization problem where various goals or objectives have to be met. Usually, the objectives are conflicting and hence a trade-off strategy is often required. Setting the trade-offs is a decision making problem and sometimes the trade-offs cannot be clearly established. This gives rises to a special class of uncertainty that is referred to as vagueness. The taxonomy presented in Oberkampf et. al. classifies these uncertainties under *epistemic uncertainty*. Fuzzy set theory has been used extensively to model such uncertainties.

3.1.3 Modeling and Simulation Uncertainties

There are uncertainties that arise due to the discrepancies between the predictions of analysis tools and the “real-world” outcomes. Some of these uncertainties arise due to abstraction and idealizations of true physical phenomena in the models used in the analysis tools. In some applications, analysis tools are conservative or consistently over-predict or under-predict certain characteristics but others are not. For example in structural dynamics, the use of Consistent Mass Matrix is known to consistently overestimate the natural frequencies of a given structure, whereas using a Lumped Mass Matrix does not follow such a trend. Tool uncertainties can be modeled as random variations, with assumed probability distributions or can be modeled as variations as fixed bounds or percentage bounds. The taxonomy presented in Oberkampf et. al. classifies such uncertainties under *epistemic uncertainty*. These uncertainties can be modeled using probabilistic methods or other non-probabilistic methods like interval methods, evidence theory etc.

Another class of uncertainties arising in modeling and simulation are errors associated with numerical methods. These errors can be quite crucial in some cases. Some of the common examples of such errors are, error tolerance in a convergence of a coupled system analysis, round-off errors, truncation errors, discretization errors associated with the integration of differential equations and finite element analysis. The errors are typically small when high-fidelity analysis tools are used in simulation-based design.

3.2 Modeling Uncertainty

The most popular way to model uncertainties is using probability theory. Typically, substantial amount of statistical information is needed for building good statistical models required by a probabilistic approach. Convex sets have been used to model uncertainties with less statistical information. Various other non-probabilistic techniques have been used such as possibility theory and evidence theory to model uncertainties with less

statistical information and fuzzy set theory is used to model uncertainties related to a decision making problem. A brief discussion of modeling of uncertainties and estimation of propagated uncertainties using these theories especially in a multidisciplinary system will be presented.

3.2.1 Probability Theory

In a probabilistic approach, uncertainties are modeled as random variables. Let the uncertain variables, X_1, X_2, \dots, X_n be continuous random variables and be denoted by the vector \mathbf{X} . The joint probability that $X_1 \leq x_1$ and $X_2 \leq x_2 \dots$ and $X_n \leq x_n$ is referred to as the Joint Cumulative Distribution Function (j.c.d.f.), $F_{\mathbf{X}}(x_1, x_2, \dots, x_n)$ or simply $F_{\mathbf{X}}(\mathbf{x})$. The Joint Probability Density Function (j.p.d.f.), $f_{\mathbf{X}}(\mathbf{x})$ is given by the expression $\frac{\partial^n}{\partial x_1 \partial x_2 \dots \partial x_n} F_{\mathbf{X}}(\mathbf{x})$. The j.p.d.f. is the probability per unit n -dimensional volume. The probability that $X_i \leq x_i$ is given by the marginal cumulative distribution function, $F_{X_i}(x_i)$, for any i . The corresponding marginal probability density function is given by $f_{X_i}(x_i)$. The mean of X_i , given by μ_i , the variance of X_i , given by σ_i^2 , and correlation between random variables X_i and X_j given by $\rho_{ij}\sigma_i\sigma_j$, where ρ_{ij} is correlation coefficient, can be obtained by the following expressions:

$$\mu_i = \int_{-\infty}^{\infty} x_i f_{X_i}(x_i) dx_i \quad (3.1)$$

$$\sigma_i^2 = \int_{-\infty}^{\infty} (x_i - \mu_i)^2 f_{X_i}(x_i) dx_i \quad (3.2)$$

$$\rho_{ij}\sigma_i\sigma_j = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_i - \mu_i)(x_j - \mu_j) f_{X_i, X_j}(x_i, x_j) dx_i dx_j \quad (3.3)$$

If the random variables are statistically independent, they are uncorrelated, i.e. $\rho_{ij} = 0$, but the converse need not be true. It is easy to obtain estimates of μ , σ and ρ by collecting statistical data of all the random variables. The marginal and joint density functions can be obtained by techniques such as maximum likelihood estimation.

When the random variables are propagated through an analysis tool, represented by a

scalar equation, $Y = h(X_1, X_2, \dots, X_n)$, the distribution, $F_Y(y)$, mean, μ_y and variance, σ_y of the output, Y , are given as follows:

$$F_Y(y) = \int \dots \int_{h(x_1, x_2, \dots, x_n) \leq y} f_{\mathbf{X}}(\mathbf{x}) dx_1 dx_2 \dots dx_n \quad (3.4)$$

$$\mu_y = \int \dots \int h(x_1, x_2, \dots, x_n) f_{\mathbf{X}}(\mathbf{x}) dx_1 dx_2 \dots dx_n \quad (3.5)$$

$$\sigma_y^2 = \int \dots \int (h(x_1, x_2, \dots, x_n) - \mu_y)^2 f_{\mathbf{X}}(\mathbf{x}) dx_1 dx_2 \dots dx_n \quad (3.6)$$

These are in general difficult to evaluate numerically beyond a few dimensions for non-linear h . Hence typically h is linearized about the mean of \mathbf{X} and the mean and standard deviation are obtained using the following expressions:

$$\mu_y \approx h(\mu_1, \mu_2, \dots, \mu_n) \quad (3.7)$$

$$\sigma_y^2 \approx \sum_{i=1}^n \left(\frac{\partial h}{\partial x_i} \right)^2 \sigma_i^2 + \sum_{i=1}^n \sum_{j=i+1}^n 2 \frac{\partial h}{\partial x_i} \frac{\partial h}{\partial x_j} \rho_{ij} \sigma_i \sigma_j \quad (3.8)$$

The above expressions are good when the uncertainties are small and when h is not very non-linear.

3.2.2 Convex Models of Uncertainty

In some cases, uncertain events form patterns that can be modeled using Convex Models of Uncertainty[24]. Examples of convex models include intervals, ellipses or any convex sets. Convex Models of uncertainties require less detailed information to characterize uncertainties than a probability model for uncertainties. Hence these can be used to model variation uncertainties and other model uncertainties. When the uncertainties, modeled using convex models, are propagated through an analysis tool, the uncertainties in the outputs of the analysis tools are intervals. The extremes of these intervals can be solved from two optimization problems one for each extreme. Depending on the nature of the performance function, local or global optimization techniques will be required. The com-

putational costs increases with the number of uncertain variables. This can be avoided by linearization of the analysis tool or by assuming monotonicity of the analysis tool. When the convex models are intervals, techniques developed in interval analysis can be used. Interval analysis is becoming popular in the structures community since it can be readily applied to finite element methods [25, 26].

3.2.3 Non-Probabilistic Techniques

Some of the non-probabilistic techniques used to model uncertainties are possibility theory and evidence theory. Probability theory, possibility theory and evidence theory are all special cases of fuzzy measure theory. In possibility theory[27], two dual measures are used namely, *possibility* and *necessity*. An uncertain variable is given a possibility distribution, that is analogous to a CDF in probability theory. The definitions for possibility and necessity of complement, union and intersections of sets are different from that in probability theory. In evidence theory, two dual measures are also used namely, *Belief* and *Plausibility*. Evidences, in form of probabilities of occurrence of an uncertain variable in various sets (also called basic probability assignments) can be assembled to obtain belief and plausibility measures. The basic definitions for belief and plausibility for complement of an event, union or intersection of events are also different from that used in probability theory.

Possibility and evidence theory[27, 28] can be used when there is insufficient information about random variations. When the uncertainties, \mathbf{X} , characterized using these theories are propagated through an analysis tool, i.e. $Y = h(\mathbf{X})$, the possibility or necessity, and belief or plausibility distributions for Y can be obtained by a combinatorial interval analysis, and computational costs grow exponentially as the number of uncertain variables increase.

3.2.4 Fuzzy Set Theory

In design optimization, it is typically required to minimize certain performance parameters, while maintaining certain performance parameters within a given range, or maintaining some strictly below specific values (hard constraints) and maintain some performance parameters approximately below specified values (soft constraints) or some approximately/strictly equal to certain values (equality constraints). The target values or ranges for these performance parameters are not fixed but fuzzy. One can model such targets as fuzzy sets[29, 30]. Fuzzy sets are normal sets with fuzzy boundaries. One can usually associate membership functions for fuzzy sets that characterize the degree of membership of a point in a fuzzy set. In crisp sets the degree of membership of a point is 0 if it is outside the set and 1 if inside the set. For fuzzy sets, the degree of membership can vary from 0 to 1.

In a multi-objective design problem, degrees of membership are assigned for the fuzzy targets and these can be interpreted as relative preferences of target values[31, 32, 33]. All the preferences of the various objectives, can be combined to get an aggregate preference, that represents the degree of satisfaction of all the objectives. In a multi-objective optimization problem, designs are found that maximize this aggregated preference. Probability, possibility and evidence theory can also be extended to fuzzy sets. Hence one can potentially treat fuzzy and other types of uncertainties simultaneously.

3.2.5 Propagation of Uncertainty in a Multidisciplinary Systems

In a multidisciplinary system, variation type uncertainties and model uncertainties propagate between analysis tools because of their interdependence. It can be seen from the development of GSE in Equations 2.5 and 2.6, that the change in the SVs due to a change in a variable (in this case an uncertain variable) depends on the coupling matrix **C**. This shows that the resulting uncertainties in SVs of a single analysis tool are not due

to local uncertainties alone. The effect of other non-local uncertainties depends on the coupling and can be estimated from the non-diagonal terms of \mathbf{C} .

So it is essential that for highly coupled multidisciplinary problems the inter-disciplinary interactions be modeled appropriately, and an uncertainty analysis be performed on the entire system. A rigorous estimation of uncertainty of outputs of a SA will be very expensive for complex SA with many sources of uncertainties, irrespective of how the uncertainties are modeled. One of the earliest attempts to characterize the propagation of uncertainty for multidisciplinary systems was done by Gu. et al.[34], who showed how to obtain worst case linearized uncertainty estimates for a multidisciplinary system with variation type uncertainties and model uncertainties.

3.3 Design Optimization Under Uncertainty

The main concerns that arise in practical design optimization problems due to uncertainties are robustness of certain performance parameters, and reliability of the design. Design optimization methodologies that take care of uncertainties should address these concerns. Some of the methodologies that do so are robust design and reliability-based design, or a hybrid of both. The discussions of these methodologies follow:

3.3.1 Robust Design

Due to variation type uncertainties, there can be large variations in certain performance parameters. In some applications this is highly undesirable, for example it is usually preferred to reduce the variation in fuel economy of a product like car. A way to reduce these large variations is to reduce the source uncertainties like manufacturing uncertainties by using better but more expensive manufacturing processes. Generally, this could lead to very large manufacturing costs. It is preferable to obtain designs characterized by low variations early during design of a product. An example is shown in Figure

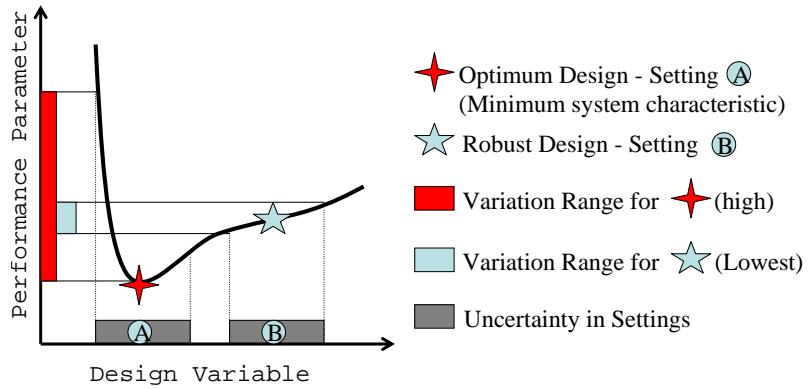


Figure 3.1. Design for Robustness

3.1, where the optimal design A is characterized by large variation of the performance parameter but the design B is characterized by a smaller variation in the performance parameter. Hence design B is more robust. The variation of a performance parameter can be characterized by its standard deviation, when probability theory is used, or by its variation interval when convex models of uncertainty are used, or by an interval of some significance level when possibility or evidence theory is used.

In a robust design optimization, either the variation in performance parameter can be minimized, or it can be constrained to be lower than some value. In a traditional robust optimization[4] due to Taguchi the main aim is to find designs with minimum variation of certain performance characteristics. The optimization is performed by selecting the best design from an orthogonal array experiments conducted for various levels of design variables. At each experimental setting of the design variables, the performance parameter variation is calculated using another set of orthogonal array experiments for different settings of the uncertain variables. Using the property of orthogonal arrays, the robust design with minimum performance parameter variation can be identified. Taguchi's method is useful for cases when uncertain conditions can be simulated by experiments and it does not require much statistical information of the uncertainties. It can be modified to account for cases when there is probabilistic information, and as the number of experiments (in

uncertain variables) tends to infinity, Taguchi's method becomes equivalent to minimizing the variance of the performance parameter as shown by Otto and Antonsson[35]. In another paper, Otto and Antonsson[36] extended Taguchi's method to constrained optimization problems.

3.3.2 Reliability-Based Design

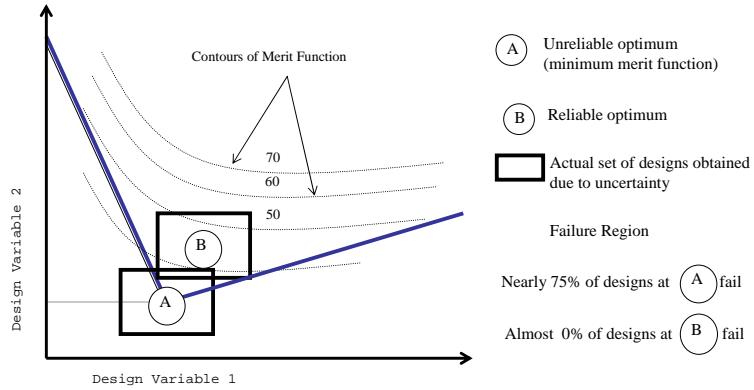


Figure 3.2. Design for Reliability

Due to variation type uncertainties and model uncertainties, certain hard constraints that model failure mechanisms or modes of the artifact can be violated. This is especially very likely at the optimum corresponding to the traditional design optimization, at where typically some hard constraints are active or close to being active. Figure 3.2 illustrates such a case, where design A is the optimum from traditional design optimization but the chance of failure is about 75% due to uncertainty in design variable settings. Design B is a more reliable optimum, that is characterized by a slightly higher merit function but with low chance of failure. When probability theory is used to model uncertainties, probabilistic reliability analysis methods can be used to obtain probability of failure with respect to the failure modes. Probabilistic reliability analysis is discussed in detail in the next section. When possibility theory or evidence theory is used to model the uncertainties then possibility or belief measures can be used to characterize the chance of failure.

Hence, design optimization under uncertainty should also include the chance of failure corresponding to the various failure modes or mechanisms. In a reliability-based design, the chance of failure is either minimized or constrained to be below an acceptable value that can be a design variable itself. When intervals or convex models are used for uncertainties, the worst case extreme value of the hard constraints can be required to be greater than zero, which would guarantee a low chance of failure.

Reliability-Based Optimization (RBO) using probabilistic reliability analysis is the main focus of this dissertation and hence is explained in detail in the next chapter. Both possibility theory and evidence theory have also been used to perform Reliability-Based Optimization. Researchers[37, 38, 39] have shown that using possibility theory can yield more conservative designs than probability theory especially when there is very less information available but can also yield less conservative designs in certain cases, when the main design criteria was to minimize the probability and possibilities of failure. Agarwal et. al.[40] implemented an RBO with constraints based on belief measure of the failure events, using approximation concepts in a trust region framework.

One can have a general optimization under uncertainty formulation with both robustness and reliability constraints. One such formulation that minimizes both mean of merit function and its variance subject to constraints on worst value of hard constraints within intervals of required confidence level is shown below[41]:

$$\text{Min.} \quad \mu_f + w\sigma_f \quad (3.9)$$

$$\text{s.t.} \quad \mu_{gi} - \beta_{reqd}\sigma_{gi} \geq 0 \quad \text{for } i=1..N_{hard} \quad (3.10)$$

$$\mathbf{d}_l \leq \mathbf{d} \leq \mathbf{d}_u \quad (3.11)$$

where β_{reqd} is the required reliability or confidence level that the hard constraint will not be violated and w is a weighting function. Sundaresan et al.[42] and Yu and Ishii[43] use Design of Experiments concepts to compute the mean and standard deviations. Typ-

ically a mean value based linear estimate is used to estimate means and standard deviations. These estimates are not good for large uncertainties and nonlinear functional f and g . Padmanabhan and Batill[44] employed global approximation concepts to solve such a formulation for multidisciplinary problems within a Concurrent Subspace Design[45] framework. Gu and Renaud[46] solved this formulation in a Collaborative Optimization framework for multidisciplinary problems.

3.4 Probabilistic Reliability Analysis

This dissertation mainly focuses on the issue of reliability using probability theory. Hence probabilistic reliability analysis is discussed in detail in this section. Reliability analysis is used to estimate the probabilities of failure due to probabilistic violation of certain failure driven constraints in the presence of random uncertainties, \mathbf{X} . Reliability analysis is used to find component probabilities of failure, i.e. probability of failure with respect to each hard or failure driven constraint, and also the system probability of failure. Let the hard constraint representing a failure mode be denoted by $g(\mathbf{X}, \boldsymbol{\eta})$ where $\boldsymbol{\eta}$ are deterministic parameters. It has to be noted that hard constraints were represented with a superscript ‘R’, which has been dropped in this section for convenience. If $g < 0$ then the artifact is said to have failed, $g > 0$ means that the artifact is safe and $g = 0$ is typically referred to as a Limit State Surface. The probability of failure due to g can be written as the following multi-dimensional integral

$$P(g(\mathbf{X}, \boldsymbol{\eta}) < 0) = \int_{g(\mathbf{X}, \boldsymbol{\eta}) < 0} \cdots \int f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (3.12)$$

This multi-dimensional integral becomes very expensive to compute when the dimension of \mathbf{X} is large. Reliability analysis provides a way to compute good estimates of the probability of failure with significantly lower computational effort.

A convenient way to represent reliability is by a reliability index, β , that can be defined

as $-\Phi^{-1}(P)$, where Φ is the univariate Gaussian (or normal) c.d.f. One of the earliest techniques to characterize reliability was the Mean Value First Order Second Moment (MVFOSM) method. In this method, μ_g and σ_g are computed using a linearized model of g as given by Equations 3.7 and 3.8. Assuming that g is normally distributed, a first order reliability index can be obtained as given below:

$$\beta = \frac{\mu_g}{\sigma_g} \quad (3.13)$$

The above equation is true for all First Order Reliability Methods (FORM). The MVFOSM was introduced previously in Equation 3.10. The MVFOSM, in general is not very accurate when considering general non-normal distribution functions for \mathbf{X} and nonlinear g . This method also lacks the property of invariance in formulation of g . For example, the probability of failure of $g = X_1 - X_2$ and $g = \log(\frac{X_1}{X_2})$ are the same, since the limit state surface is same for both, i.e. $X_1 - X_2 = 0$. MVFOSM will give different reliability indices for both formulations. This lack of invariance motivated the development of the advanced FORM method that is based on calculation of a Most Probable Point(MPP).

The Most Probable Point (MPP) of failure is found in a standard normal space, \mathbf{U} , for a single hard or failure driven constraint $g(\mathbf{x}, \boldsymbol{\eta})$. The components of \mathbf{U} are normally distributed with zero means and unit variance and are statistically independent. Any set of continuous random variables, \mathbf{X} , can be transformed to \mathbf{U} using an one-to-one transformation i.e. $\mathbf{U} = T(\mathbf{X}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are distribution parameters (means, modes, standard deviations, etc..) of \mathbf{X} . Many such transformations are possible, but the one typically used is called the *Rosenblatt* transformation (See Appendix A). The MPP, \mathbf{u}^* , lies on the failure surface or the limit state surface $g(\mathbf{x}, \boldsymbol{\eta}) = g(\mathbf{u}, \boldsymbol{\eta}) = 0$, and its location is the closest point on the limit state surface to the origin in \mathbf{U} -space. The MPP can be found by

solving the following constrained optimization.

$$\min_{\mathbf{u}} \quad \mathbf{u}^T \mathbf{u} \quad (3.14)$$

$$\text{s.t.} \quad g(\mathbf{u}, \boldsymbol{\eta}) = 0 \quad (3.15)$$

Various algorithms exist to perform the MPP searches [47, 48]. One of the approaches is the Hasofer-Lind and Rackwitz-Fiessler (HL-RF) algorithm that is based on a Newton-Raphson root solving approach. Variants of the HL-RF methods exist that use additional line searches to HL-RF scheme. The family of HL-RF algorithms can exhibit poor convergence for highly nonlinear and badly scaled problems, since they are based on first-order approximations of the hard or failure driven constraint. For further details on HL-RF methods see Appendix B. A Sequential Quadratic Programming (SQP) approach is often a more robust approach.

After the MPP search, the component probability of failure can be estimated using the advanced First Order Reliability Method (FORM), which is based on linear approximation of the limit state built at the MPP in \mathbf{U} space. A signed reliability index, $\beta = \frac{-\nabla_{\mathbf{u}}g}{\|\nabla_{\mathbf{u}}g\|} \mathbf{u}^*$, that is positive if vector joining the origin and the MPP point in infeasible region ($g < 0$) and vice-versa. The magnitude of β from FORM is equal to the distance from the origin to the MPP, \mathbf{u}^* . It has to be noted that β is invariant to the formulation of g and in general more accurate than MVFOSM.

In a Second Order Reliability Method (SORM) second order approximations are fit to the limit state surface. The probability of failure can be found using an asymptotic formula as given by Breitung[49] based on parabolic approximations of the limit state surface and is exact as $\beta \rightarrow +\infty$. Tvedt[50] presented a numerical integration scheme to obtain exact probability of failure for a general quadratic limit state. Kiureghian et. al.[51] presented a method to estimate probability of failure based on half-parabolic approximations to the limit state surface. The probability of failure can also be found using Monte

Carlo Simulation techniques[52]. Monte Carlo Simulation techniques will be discussed in detailed later in Chapter 6.

The concept of FORM and SORM is illustrated in the Figure 3.3 for an example with two random variables, X_1 and X_2 .

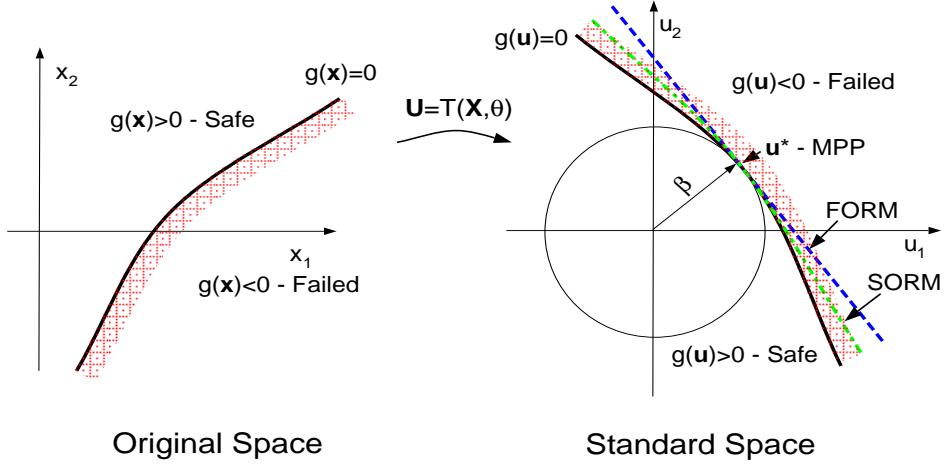


Figure 3.3. First Order and Second Order Reliability Methods

A sensitivity analysis of the probabilities of failure and reliability index with respect to distribution parameters and limit state parameters can be obtained[53]. The sensitivities in FORM are given as follows

$$\frac{\partial \beta}{\partial \theta} = -\frac{\nabla_u g(\mathbf{u}^*)}{\|\nabla_u g(\mathbf{u}^*)\|} \frac{\partial T(\mathbf{x}^*, \boldsymbol{\theta})}{\partial \theta} \quad (3.16)$$

$$\frac{\partial \beta}{\partial \eta} = \frac{1}{\|\nabla_u g(\mathbf{u}^*)\|} \frac{\partial g(\mathbf{u}^*, \boldsymbol{\eta})}{\partial \eta} \quad (3.17)$$

A big advantage in using reliability analysis is that one can find the uncertainties with the most influence on the probability of failure from the components of $\frac{\nabla_u g(\mathbf{u}^*)}{\|\nabla_u g(\mathbf{u}^*)\|}$. So a designer can recommend that some of such uncertainties be reduced by appropriate quality control purposes.

To compute the system probability of failure, the system failure has to be defined. The system failure can be a series event of all the component failure modes, which means that

if a failure occurs with respect to any of the component failure modes then the system will fail. System failure can be a parallel event, where failure with respect to all the failure modes should occur, or a K-out-of-N system where K failures have to occur. For complex systems, fault tree diagrams are used to analyze the system failure. Bounds for probability of failure estimates of series and parallel systems can be found using unimodal and bimodal or Ditlevsen bounds[8]. The formulas for unimodal bounds for series and parallel systems are shown below

$$\text{Unimodal: } \max_i P(g_i \leq 0) \leq P(\cup g_i \leq 0) \leq \sum_{i=1}^n P(g_i \leq 0) \quad (3.18)$$

$$0 \leq P(\cap g_i \leq 0) \leq \min_i P(g_i \leq 0) \quad (3.19)$$

In chapter 6, techniques to obtain probabilities of failure of series and parallel systems using Monte Carlo methods are presented.

One of the major advantages of using probabilistic reliability analysis is that the statistical models and reliability models can be updated when new statistical information are available using Bayesian statistics. One of the major disadvantages of using probabilistic techniques is that substantial amount of statistical information is required to model uncertainties. Kiureghian and Liu[54] present techniques to incorporate uncertainties with incomplete probability information in probabilistic reliability analysis. The other major disadvantage is that the reliability estimates for applications with low probabilities of failure typically depend on probabilistic information for extreme values of uncertain variables, that are difficult to model due to lack of statistical information on such extreme variations.

3.5 Summary

The main uncertainties that arise during simulation-based design of a product and during its life-time can be classified as variation type uncertainties, uncertainties in de-

cision making and modeling and simulation uncertainties. Probability theory is used to model variation type uncertainties. Convex models of uncertainty, possibility theory and evidence theory can be used to model variation type uncertainties with less statistical information and modeling and simulation uncertainties. Fuzzy set theory is used to model uncertainties involved in decision making. In multidisciplinary systems, the uncertainties propagate between the different analysis tools. Hence uncertainty estimation for multidisciplinary systems can be very expensive. The major issues that arise in engineering design due to variation type uncertainties and modeling and simulation uncertainties are robustness of certain performance parameters and reliability with respect to critical failure modes or mechanisms.

When probability theory is used to model uncertainties, probabilistic reliability analysis methods can be used to obtain reliability estimates with respect to various failure modes. The probability of failure due to a failure mode is a multi-dimensional integral that is very expensive to calculate. Probabilistic reliability analysis methods efficiently compute this integral using techniques such as First Order Reliability Method (FORM), Second Order Reliability Method (SORM) and Monte Carlo Simulation (MCS) techniques. Typically in most of these techniques, the original uncertain variables are transformed to a standard normal space. A Most Probable Point (MPP) of failure is found in this standard normal space. FORM is based on a first order approximation of the failure surface. SORM is based on second order approximations of the failure surface. Sensitivities of reliability or probabilities of failure can be obtained and relative importance of the uncertain variables can also be obtained.

CHAPTER 4

RELIABILITY-BASED OPTIMIZATION

This chapter deals with important aspects of a Reliability-Based Optimization (RBO) problem and presents some of the issues that arises in an RBO problem and some of the work that has been done by other researches in addressing some of these issues. First the different aspects of formulating a Reliability-Based Optimization (RBO) formulation will be discussed. Next, some of the alternative RBO formulations will be presented and discussed. Finally, the use of approximation concepts in RBO application will be introduced.

4.1 RBO Formulation

Traditional design optimization does not take into consideration the inherent variation type uncertainties present in design variables and parameters and modeling uncertainties that arise in development of the analysis tools. Due to these uncertainties the design could have a high probability of failure because of the violations of hard or failure driven constraints (g^R in traditional design optimization formulation 2.1-2.4). This is particularly true if the hard constraints are active at the optimum design. So one would have to replace a traditional design optimization with an RBO formulation[55], where the hard

constraints are replaced with reliability constraints, as shown below:

$$\text{Minimize} \quad f(\mathbf{d}, \mathbf{p}, \mathbf{y}(\mathbf{d}, \mathbf{p}), \mathbf{q})) \quad (4.1)$$

$$\text{s.t.} \quad \mathbf{g}^{rbo}(\mathbf{d}, \mathbf{p}, \mathbf{q}) \geq 0 \quad (4.2)$$

$$g_j^D(\mathbf{d}, \mathbf{p}, \mathbf{y}(\mathbf{d}, \mathbf{p})) \geq 0 \quad \text{for } j=1, \dots, N_{soft} \quad (4.3)$$

$$\text{Bounds:} \quad \mathbf{d}_l \leq \mathbf{d} \leq \mathbf{d}_u \quad (4.4)$$

where \mathbf{g}^{rbo} represents constraints on probabilities of failure or reliability indices with respect to the different failure modes in the problem and \mathbf{q} are other parameters such as required reliabilities, allowable probabilities of failure. \mathbf{g}^{rbo} can be written as given below.

$$g_i^{rbo} = P_{all,i} - P_i \quad \text{or} \quad \beta_i - \beta_{reqd,i} \quad (4.5)$$

where P_i and β_i are the failure probability and reliability index respectively due to i^{th} failure mode at the given design and $P_{all,i}$ and $\beta_{reqd,i}$ are the allowable probability of failure and required reliability level for this failure mode.

4.1.1 Merit and Constraints

The merit function, f , for a structural optimization problem can include the initial cost, i.e. the cost of the structure, and it can also include costs due to failure of the artifact that include maintenance and repair costs. The initial cost typically is high for structures with low chance of failure or risk, and vice-versa. The failure cost increases as the risk or chance of failure increases. Hence there is an optimum risk level or reliability level that can be obtained that corresponds to a minimum attainable total cost. This concept is illustrated in Figure 4.1 adapted from Frangopol and Moses[56]. Hence a structural RBO problem is essentially a multi-objective problem that deals with the minimization of both these costs. Hence the required reliabilities can be included in the merit function, f and can be made as one of the design variables for the RBO problem. In some problems, the

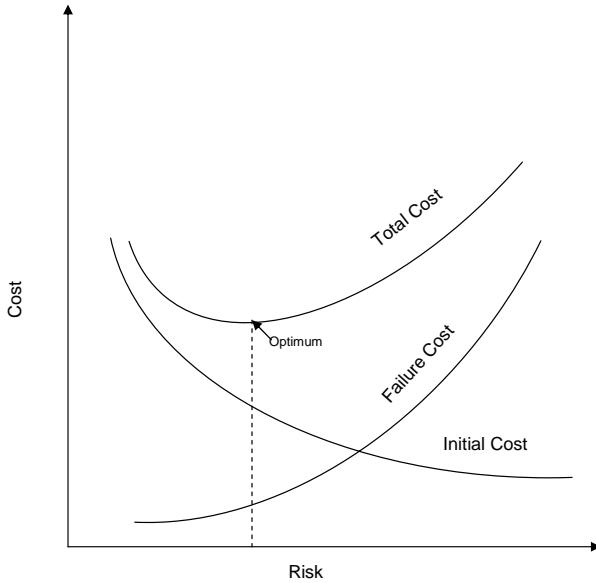


Figure 4.1. Cost Vs Risk

required reliabilities are already set a priori due to strict regulatory or statutory laws. For such cases, the required reliabilities are already set. One of the other possible formulation is to minimize the failure costs subject to constraints on initial costs.

It has to be noted that the RBO formulation given by Equations 4.1-4.4 assumes that the violation of soft constraints due to variation uncertainties are permissible and can be traded off for more reliable designs. For practical problems, robustness of the design requirements represented by the merit function and soft constraints could be a significant issue, one that would require a hybrid robustness and reliability based design optimization formulation. Another characteristic of the RBO formulation is that it is time-invariant. In practical applications, the reliabilities are time dependent. Reliabilities typically decay with time. In designing products with high operational lifetime like offshore structures, inspection and maintenance plans have to be constructed to account for the time dependence of reliability. A general time-variant RBO formulation should also consider the problem of optimal scheduling of inspections and maintenances along with minimization of initial costs and failure costs[57]. An inspection is scheduled when the overall reliabil-

ity of a structure is believed to have gone below a certain level. It is usually desirable to reduce the number of inspections and maintenances. Another alternative is to express the reliability constraints in terms of Mean-Time-To-Failure (MTTF). This is very commonly used in reliability-based design of electrical and electronic systems where time dependent statistical models are used to model failure or reliability of such systems[58]. Such topics, though very interesting, are beyond the scope of this dissertation and hence time-variant reliability will not be considered.

The component failure modes are $\mathbf{g}_i^R < 0$. The system failure mode is a combination of all the component failure modes such as series, parallel or both. Enevoldsen and Sorensen[59] perform FORM-based RBO for problems with system failure mode defined as series systems of parallel systems of component failure modes. In problems such as redundant structures, the failure modes can be very complicated to determine and the failure modes are defined by order of failure of structural elements of the redundant structure[60]. One can also have redundant reliability constraints that impose constraints on reliabilities of selected failure modes after certain key structural elements have failed in a redundant, statically indeterminate structure. Standard reliability techniques, as discussed in Chapter 3, are required to estimate these probabilities of failure and reliability indices.

4.1.2 Random Variables and Design Variables

Probabilistic Reliability Analysis techniques require complete statistical information of uncertainties, that are modeled as continuous random variables \mathbf{X} . Components of \mathbf{d} are either distribution parameters, $\boldsymbol{\theta}$, of \mathbf{X} like means, modes, standard deviations and coefficients of variation, or deterministic parameters, also called limit state parameters, denoted by $\boldsymbol{\eta}$. The deterministic parameters $\boldsymbol{\eta}$ represent design variables with no uncertainties in their settings, whereas design variables that are means or modes of certain uncertainties are actually desired settings of the quantities represented by the uncertainties. There can

be design variables that characterize standard deviations of certain uncertainties, especially in quality control applications where the magnitudes of certain uncertainties can be controlled. \mathbf{p} characterizes the means or modes or any first order distribution parameters of some random variables.

The random variables can be continuous, discrete or a combination of the two. In this dissertation only continuous random variables are considered. Some of the commonly used distributions to model random variables are normal, lognormal, exponential, uniform, Weibull and extreme value distributions. It is desirable to keep the number of random variables to a minimum from a computational efficiency point of view. The screening of random variables is necessary for practical problems. This can be done using engineering intuition or from a post-optimality sensitivity analysis of the results obtained from the MPP searches.

4.1.3 Computational Issues

Traditional RBO given by Equations 4.1-4.4 is a very expensive approach for problems with complex and expensive analysis tools and a large number of random and design variables, due to inherent high computational costs associated with the reliability analysis. The high computational costs arise in a reliability analysis due to the following reasons:

- i. In reliability analysis, MPP searches have to be performed for each hard constraint and this would require that an exact system analysis including sensitivity analysis be performed during each iteration. The computational costs increases with the number of hard constraints.
- ii. A MPP search is usually expensive for hard constraints that are very nonlinear in the \mathbf{U} space. The computational costs also increase for nonlinear constraints if the MPP is far away, or in other words the failure with respect to that failure mode is very low or the reliability index is very high.

- iii. If Monte Carlo Simulation techniques are used for estimating probabilities of failure then the computational costs increase further.
- iv. Reliability analysis for all the hard constraints have to be performed during each iteration in the RBO.

Extension to SORM, Monte Carlo Simulation-based component and system reliability problems are difficult and very expensive. Pu et al.[61] present an approach to perform system reliability-based optimization in a less expensive way. They perform a component reliability-based RBO but vary the target reliability levels by performing intermediate system reliability evaluations. **The main aim of the work presented in this dissertation is to reduce the computational costs associated with the traditional RBO when applied to multidisciplinary systems.**

4.1.4 Screening of Hard Constraints

As mentioned earlier, the computational expenses increase with the number of failure or hard constraints. So it is desirable to filter hard constraints that have insignificant contribution to the failure of the system, and ones that never fail throughout the optimization. Preliminary investigation along with engineering judgment can be used to exclude such hard constraints. But sometimes certain failure modes can be highly inactive in a certain region in the design space but could become active in other regions of the design space. So intermediate checks might be needed during the RBO. RBO techniques that treat the number of important hard constraints as variable can be used. In Murotsu et al[62] a branch and bound technique is used to find the set of important failure modes from a total of 1064 failure modes in one of their test problems involving a redundant structural optimization problem, and then the optimizer is invoked that does the RBO with only those constraints.

Another computational issue that arises due to the treatment of non-critical hard constraints is that the MPP searches in the probabilistic reliability analysis can fail because the MPP search algorithm can go to certain domains in the random variable space, where the transformations between \mathbf{X} and \mathbf{U} cannot be computed due to limitations of the available floating-point precision and where the mathematical models used by analysis tools can fail. Robust MPP search algorithms that can handle such cases will be presented in Chapter 5.

4.2 Alternative RBO Formulations

Various alternative RBO formulations have been proposed by various researches to address the issue of high computational costs associated with traditional RBO. A brief discussion of few of such formulations are presented in this section.

4.2.1 Uni-level RBO

Traditional RBO can be called a bi-level approach, since during each iteration of the main optimization various sub-optimization phases (MPP searches) are required to evaluate the reliability constraints. Uni-level RBO is based on the idea that all these optimization phases can be merged together and reformulated as a single optimization. Such a uni-level RBO using FORM component reliability constraints can be implemented

using the following optimization formulation

$$\min_{\mathbf{d}, \mathbf{u}_1, \dots, \mathbf{u}_{N_{hard}}} f(\mathbf{d}, \mathbf{p}) \quad (4.6)$$

$$\text{s.t.} \quad \|\mathbf{u}_i\| \|\nabla_u g_i^R\| + \nabla_u g_i^R \mathbf{u}_i = 0 \quad (4.7)$$

$$g_i^R(\mathbf{u}_i, \boldsymbol{\eta}) = 0 \quad (4.8)$$

$$\|\mathbf{u}_i\| \geq \beta_{reqd,i} \quad \text{for } i = 1, \dots, N_{hard} \quad (4.9)$$

$$g_j^D(\mathbf{d}, \mathbf{p}) \geq 0 \quad \text{for } j = 1, \dots, N_{soft} \quad (4.10)$$

$$\text{Bounds:} \quad \mathbf{d}_l \leq \mathbf{d} \leq \mathbf{d}_u \quad (4.11)$$

The equalities 4.7 and 4.8 are obtained from first order Karush-Kuhn-Tucker (KKT) optimality conditions for the MPP search formulation given by Equations 3.14-3.15, for positive values of β .

One of the implementation issue that arises is that Equation 4.7 can be nonlinear for g 's that are highly non-linear in U space. Gradients of Equation 4.7 requires the Hessian of g , that is typically very expensive to calculate and can be very memory-intensive from an algorithmic perspective when there are large number of random variables. Alternatively, the Hessian can be obtained efficiently by Hessian update schemes like Broyden-Fletcher-Goldfarb-Shanno (BFGS), Davidon-Fletcher-Powell (DFP) or Symmetric Rank 1 (SR1) update schemes[63]. It also has to be noted that Equation 4.7 is not differentiable at $\mathbf{u}_i = 0$ and hence this can present some difficulties when used with a gradient-based optimizer.

4.2.2 Performance Measure Approach (PMA)

Tu and Choi[64, 65] reformulate the FORM reliability constraints with constraints on the worst case values of the hard constraints obtained using an inverse reliability analysis. The worst case value of g_i^R , $g_i^{R,w}$ can be formulated by the following inverse reliability

analysis formulation:

$$g_i^{R,w} = \min_{\mathbf{u}_i} \quad g_i^R(\mathbf{u}_i, \boldsymbol{\eta}) \quad (4.12)$$

$$\text{s.t.} \quad \mathbf{u}_i^T \mathbf{u}_i = \beta_{reqd,i}^2 \quad (4.13)$$

The FORM reliability constraint is replaced with the constraint $g_i^{R,w} \geq 0$. Such a formulation will give the same result as a component FORM-based RBO. Tu and Choi show that PMA can work for cases where conventional MPP searches can fail in an RBO. The advantage of this approach is that when performing reliability analysis at a design where the hard or failure driven constraint in consideration is highly feasible, or in other words, the reliability index is too high, PMA is less expensive. A disadvantage of the approach is that it might require more computations at designs where the reliability index is lower than the required reliability level.

4.2.3 Semi-infinite Optimization

Royset et al.[66, 67] and Kirjner-Neto et al.[68] use a semi-infinite optimization approach to perform a decoupled RBO. A semi-infinite optimization is a constrained optimization with infinite constraints. A typical example is an optimization problem with a time dependent constraint that is required to be satisfied for a certain time range.

The concept is very similar to the PMA-based RBO. The reliability constraint is replaced with $g_i^R(\mathbf{u}_i, \boldsymbol{\eta}) \geq 0$ and it is required that this constraint be satisfied for all \mathbf{u}_i contained in a sphere of radius $\beta_{reqd,i}$. Such a requirement is equivalent to imposing the constraint $\min_{\{\|\mathbf{u}_i\| \leq \beta_{reqd,i}\}} g_i^R(\mathbf{u}_i, \boldsymbol{\eta}) \geq 0$. Such a formulation, can be readily solved using existing semi-infinite optimization techniques. Kirjner-Neto et al. simplify the formulation further for RBO for systems with failure modes in series. They replace the reliability constraints with a single deterministic constraint in form of a min-max formulation.

4.2.4 Miscellaneous Approaches

A single level approach has been used by Wang and Kodiyalam[69] where the traditional bi-level RBO is replaced by a single level RBO. In this approach, the FORM component reliability constraints are replaced with inequality constraints $g_i^R(\mathbf{x}^*, \boldsymbol{\eta}) \geq 0$. \mathbf{x}^* is computed based on a linear approximation of g in \mathbf{U} -space. Hence no MPP search algorithm is required to compute \mathbf{x}^* during the RBO. The RBO is repeated till the design obtained has converged. This approach will work well for problems where g is not very nonlinear in \mathbf{U} space. Unfortunately this depends on the type of distributions used for the original random variables, \mathbf{X} , and hence the convergence for this approach for non-linear g 's can be poor.

A variable complexity algorithm has been used by Koch and Kodiyalam[70] to perform RBO. The RBO is performed using MVFOSM based reliability constraints. At the result of the MVFOSM-based RBO, an exact reliability analysis is performed using FORM or SORM and the required reliability levels of the MVFOSM reliability constraints are scaled by the ratio of MVFOSM estimates and the actual reliability values. The RBO is repeated with the new scale parameters. This is repeated till convergence. The performance of this methodology will be good if the nonlinearity of the constraints in standard space is less and convergence of this methodology could be poor for cases where the hard constraints are very nonlinear.

4.3 Approximations Concepts in RBO

Approximation concepts can be used for improving the efficiency of RBO. Grandhi and Wang[71] perform reliability analysis efficiently by using a Two-point Adaptive Non-linear Approximation (TANA) of the hard constraint[72] in a sequential fashion. Wang et. al.[73] use the sequential reliability analysis using TANA and multivariate spline approximations for reliability constraints to achieve significant computational savings. Chandu

and Grandhi[74] also use approximations to reliability constraints to achieve significant computational savings compared to a traditional RBO without approximation concepts. Rajasekhar and Ellingwood[75] use sequential Design of Experiments based Response Surface Methodology to perform reliability analyses and they use the final approximation at the MPP to find estimates of probability of failure using Monte Carlo simulation. By constructing the approximation at the MPP, the actual $g = 0$ surface is captured by the approximation very accurately. Such approximations, that are constructed at the MPPs, will be referred to as limit state approximations in this dissertation. The reliability estimates obtained from limit state approximations will be very good because they are constructed at the MPPs. This is a very attractive way of creating approximations, since highly accurate limit state approximations for MPP searches in an approximate RBO can be used rather than computing actual hard constraint information. Hence limit state approximations will be discussed in detail in the rest of the section.

4.3.1 Limit State Approximations

The limit state approximations, \tilde{g}^R can be constructed at the MPP in terms of \mathbf{x}^* or \mathbf{u}^* and either in $\{\mathbf{x}^*, \boldsymbol{\eta}\}$ or $\{\mathbf{u}^*, \boldsymbol{\eta}\}$ space. \tilde{g}^R has to be built in $\boldsymbol{\eta}$ space since one has to capture the trends of the limit state surface with respect to deterministic design variables, $\boldsymbol{\eta}$.

The limit state approximations can be used in an approximate RBO phase that can be used in an RBO framework consisting of the following major steps as presented by Oakley et. al.[76].

- i. Perform exact MPP searches at given design.
- ii. Construct limit state approximations.
- iii. Perform approximate RBO where the reliability analysis is performed using the

limit state approximations. Obtain a new design. Oakley et. al.[76] use Monte Carlo Simulation on the limit state approximations for reliability analysis in RBO.

iv. Repeat i-iii till convergence is achieved.

An example of a limit state approximation constructed in $\{\mathbf{x}^*, \boldsymbol{\eta}\}$ space, would be a quadratic approximation as given below:

$$\begin{aligned}\tilde{g}^R(\mathbf{x}, \boldsymbol{\eta}) &= g^R(\mathbf{x}^{*,k}, \boldsymbol{\eta}^k) + \nabla_x g^R(\mathbf{x} - \mathbf{x}^{*,k}) + \nabla_{\boldsymbol{\eta}} g^R(\boldsymbol{\eta} - \boldsymbol{\eta}^k) \\ &\quad + \frac{1}{2} \begin{bmatrix} \mathbf{x} - \mathbf{x}^{*,k} \\ \boldsymbol{\eta} - \boldsymbol{\eta}^k \end{bmatrix}^T \mathbf{Q} \begin{bmatrix} \mathbf{x} - \mathbf{x}^{*,k} \\ \boldsymbol{\eta} - \boldsymbol{\eta}^k \end{bmatrix}\end{aligned}$$

In the above equation $\mathbf{x}^{*,k}$ and $\boldsymbol{\eta}^k$ are the MPP and limit state parameter settings during the k^{th} overall iteration of the RBO, and \mathbf{Q} is the Hessian of the quadratic approximation. The Hessian can be obtained by standard Hessian update schemes like BFGS and SR1 that matches the gradient at another point other than the current point, or by a fitting process based on a least squares formulation that would include the function and gradient information from the search histories[77], or modified Hessian update schemes to include function as well as gradient information from search histories[78].

Another alternative is to develop approximations at the MPPs for the intermediate variables \mathbf{y} , especially when the hard constraints are explicit expressions in \mathbf{y} . When one is able to create good approximations for \mathbf{y} , the resulting approximations of the limit states will be good too. Approximations for \mathbf{y} can be developed using a fitting process or a Hessian update scheme as explained before.

The main advantage of using limit state approximations is that it can be used in a computationally efficient manner for RBO with general reliability constraints even those obtained from importance sampling or other Monte-Carlo techniques.

4.4 Summary

In a Reliability-Based Optimization (RBO) problem, a merit function is minimized subject to reliability constraints and deterministic constraints. The reliability constraints are constraints on the reliabilities with respect to the various failure modes. These are obtained using standard probabilistic reliability methods presented in Chapter 3. The merit can include initial costs, failure costs and scheduling and maintenance costs. The uncertain variables are modeled as random variables. The design variables can be distribution parameters like means or standard deviations of the random variables or can be deterministic. The computational expenses of a traditional RBO increases with the number of failure modes and the nonlinearity of hard constraints in standard normal space. The number of random variables and hard constraints have to be kept to a minimum for reducing the computational effort.

Alternative RBO formulations have been suggested to reduce the computational costs associated with a traditional RBO. In a uni-level RBO, the FORM based reliability constraints are replaced with the first order optimality conditions of the MPP searches. In a Performance Measure Approach, the FORM-based reliability constraints are replaced with the worst case values of hard constraints that occur within a given required reliability level. In a Semi-infinite Optimization approach that is based on the performance measure approach, semi-infinite optimization techniques are used to satisfy the hard constraints within a sphere on radius equal to required reliability. In a single level approach, an iterative RBO scheme with sequential linearization of hard constraints is used. In a variable complexity algorithm, scaled Mean Value First Order Second Moment (MVFOSM) reliability constraints are used during RBO, where the scale parameters are ratio of MVFOSM estimates and actual reliability estimates.

Approximation concepts have been used to reduce the computational costs in an RBO. Approximations for hard constraints and reliability constraints have been used in a se-

quential fashion during MPP searches and RBO to reduce the computational costs. Approximations to limit state surfaces have been used and are constructed at the Most Probable Points. In an RBO framework that uses limit state approximations, first exact MPP searches are performed, then the limit state approximations are created and then reliability analysis is performed on these limit state approximations during the RBO. This is done in a sequential fashion till convergence.

CHAPTER 5

RBO METHODOLOGIES FOR MULTIDISCIPLINARY PROBLEMS

In this chapter, different RBO methodologies for multidisciplinary design problems will be presented. First, a traditional RBO methodology, that essentially treats the multidisciplinary analysis as a “black-box”, will be described. Second, an RBO methodology will be presented that employs Concurrent Subspace Optimization (CSSO) technique, in the main optimization level. Next, an improvement of the CSSO based RBO methodology will be presented that employs MDO approaches for MPP searches. The MDO approaches for MPP searches require an algorithm that can give the best alternative solutions when the bounded MPP searches are infeasible. Finally, two algorithms are presented that implement the bounded MPP searches.

5.1 Traditional RBO

In a multidisciplinary system, the net uncertainties in outputs of the SA result from the propagation of all the source uncertainties due to the information exchanges between the disciplines. Hence, while performing a reliability analysis, the whole SA has to be evaluated. During a traditional RBO, a deterministic analysis is needed in addition to the reliability analysis to determine the merit function and soft constraints and their sensitivities. The deterministic analysis and reliability analysis can be viewed as additional contributing analyses of an equivalent system analysis, referred to as rSA in this work. The rSA performs a reliability analysis and a deterministic SA, hence the name rSA. This

is illustrated in Figure 5.1. The rSA also outputs sensitivities of merit function, deterministic constraints and reliability constraints with respect to the design variables when needed. It has to be noted that the deterministic analysis is redundant when the starting point of the random variables for the MPP searches given by Equation 3.14-3.15 correspond exactly to the current \mathbf{d} and \mathbf{p} values. In such a case the deterministic analysis is just a by-product of the reliability analysis. The figure 5.1 shows a general case where the starting points for the MPP searches need not correspond to the current \mathbf{d} and \mathbf{p} values. The starting point for the MPP searches are typically set as the mean values of \mathbf{X} or the origin in \mathbf{U} space. Such a strategy can be computationally expensive when this is done for every iteration, especially during the final stages of the RBO. In this work, the starting point for the MPP search during the $k + 1^{th}$ iteration in the RBO is set as the MPP obtained in the k^{th} iteration for each hard constraint.

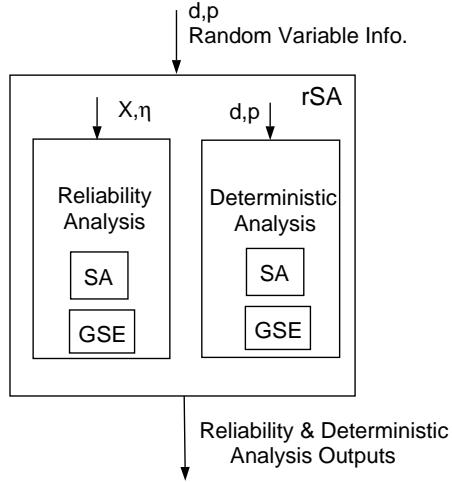


Figure 5.1. Reliability Analysis in a Multidisciplinary Problem

In a traditional RBO approach, that treats rSA as a “black-box”, repeated evaluations of the rSA is required. A traditional RBO approach is expensive due to the inherent computational intensity of the reliability analysis. The MPP search has to be performed for

each hard constraint and this would require that exact SA and GSE be performed during each iteration. The computational cost increases with the number of hard constraints and usually is expensive for hard constraints that are very nonlinear in the \mathbf{U} space. The computational costs also increase for nonlinear constraints if the MPP is far away from origin in \mathbf{U} space, or in other words the failure with respect to that failure mode is very low or the reliability index is very high. A traditional RBO framework for coupled multidisciplinary problems can be viewed as a tri-level optimization framework, where the first level of optimization is in \mathbf{d} space, the second level of optimizations are the MPP searches and the third level of optimizations are the iterative processes involved during SAs, that is required during each iteration of the MPP searches. The number of the second level optimizations, during one iteration of the first level optimization, is equal to the number of hard constraints in the problem. The number of the third level optimization or number of SA calls is equal to the number of iterations required during MPP searches for all the hard constraints, that require the coupled SA.

5.2 A Concurrent Subspace Optimization Framework for RBO

In this section, an approach is presented that aims at reducing computational costs associated with performing a standard RBO for a multidisciplinary problem by employing MDO approaches and approximation concepts[79]. Various MDO approaches have been proposed for performing traditional optimization of multidisciplinary problems. A number of the approaches; the Concurrent Subspace Optimization (CSSO)[14], Collaborative Optimization (CO)[15] and Sequential Approximate Optimization (SAO)[18] make use of decomposition and concurrent optimization strategies, and have been discussed in Chapter 3. The methodology presented here is based on the CSSO approach.

The flowchart of the methodology used for the current study is shown in Figure 5.2. The methodology is essentially a CSSO framework with modifications to enable a multi-

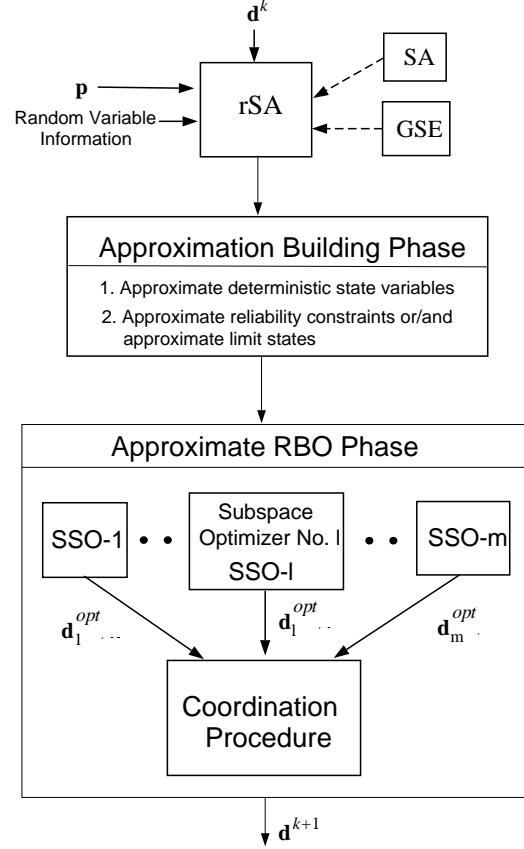


Figure 5.2. Flowchart of RBO-CSSO Framework

disciplinary RBO, and will be referred to as the RBO-CSSO approach. The basic framework of the methodology is similar to that proposed by Sues and Cesare[80]. It consists of three main phases, (1) the rSA, (2) approximations building phase and (3) the approximate RBO phase. The new design obtained from the approximate RBO is carried over to the rSA module again, and this is repeated till convergence is obtained. Detailed description of the three main phases follow:

5.2.1 Exact Deterministic and Reliability Analysis

The first phase in the framework is the rSA. In this framework, during the k^{th} iteration, the rSA is required to output merit function, f , deterministic constraints, \mathbf{g}^D , determin-

istic converged SVs, \mathbf{y}_{cnvgd} , reliability analysis outputs, \mathbf{g}^{rbo} , and sensitivities for both deterministic quantities and reliability constraints with respect to the design variables, \mathbf{d}^k . The starting points for the MPP searches, obtained by solving Equations 3.14-3.15, can be either set as the MPPs obtained during the exact rSA or as the MPPs obtained during approximate reliability analysis from any of the Subspace Optimizers in $k - 1^{\text{th}}$ iteration. One of the application problems uses the former strategy and another application problem uses the latter. The details will be provided in chapter 7 where the implementation studies of the RBO-CSSO approach are presented.

5.2.2 Approximation Building Phase

In the approximations building phase, two kinds of approximations are developed, one used for the approximate deterministic analysis during the Subspace Optimizations (SSOs), and the other for the approximate evaluation of reliability constraints in the SSOs. The approximations for the deterministic converged SVs, $\tilde{\mathbf{y}}_{cnvgd}$, at the current design \mathbf{d}^k are developed in this phase based on available information from the exact deterministic analysis in the rSA. These approximations are typically linear in CSSO. The other approximations, required for the approximate evaluation of reliability constraints, can be either approximations of reliability constraints, $\tilde{\mathbf{g}}^{rbo}$, or approximations of hard constraints, $\tilde{\mathbf{g}}^R(\mathbf{x}, \boldsymbol{\eta})$.

As presented in Chapter 4, hard constraint approximations have been used by other researchers for reducing the computational expenses during the MPP searches[71], in a sequential manner. But in this work, exact MPP searches are performed in the rSA, and limit state approximations are constructed about the MPPs. Limit state approximations can be viewed as approximations of the limit state surfaces i.e. $g^R = 0$ surfaces. The limit state approximations, $\tilde{g}^R(\mathbf{X}, \boldsymbol{\eta})$ about the MPP, \mathbf{x}^* , are formulated in $\{\mathbf{x}, \boldsymbol{\eta}\}$ space.

All the approximations, $\tilde{\mathbf{y}}_{cnvgd}$, $\tilde{\mathbf{g}}^{rbo}$ and $\tilde{\mathbf{g}}^R(\mathbf{x}, \boldsymbol{\eta})$ can be formulated from the previous

and current rSA evaluations, and various kinds of approximations can be used. In this work, linear approximations for all the approximations are used and the functional forms of these approximations are given below,

$$\tilde{\mathbf{y}}_{cnvgd}(\mathbf{d}) = \mathbf{y}_{cnvgd}(\mathbf{d}^k) + \nabla_d \mathbf{y}_{cnvgd} [\mathbf{d} - \mathbf{d}^k] \quad (5.1)$$

$$\tilde{\mathbf{g}}^{rbo}(\mathbf{d}) = \mathbf{g}^{rbo}(\mathbf{d}^k) + \nabla_d \mathbf{g}^{rbo} [\mathbf{d} - \mathbf{d}^k] \quad (5.2)$$

$$\tilde{g}_i^R(\mathbf{x}, \boldsymbol{\eta}) = g_i^R(\mathbf{x}_i^{*,k}, \boldsymbol{\eta}^k) + \nabla_x g_i^R [\mathbf{x} - \mathbf{x}_i^{*,k}] + \nabla_{\boldsymbol{\eta}} g_i^R [\boldsymbol{\eta} - \boldsymbol{\eta}^k] \quad (5.3)$$

In Equation 5.3, $\mathbf{x}_i^{*,k}$ and $\boldsymbol{\eta}^k$ are the MPP and limit state parameter settings, respectively, during the k^{th} overall iteration for the hard constraint g_i^R . Another alternative for constructing $\tilde{\mathbf{g}}^R(\mathbf{x}, \boldsymbol{\eta})$ is to develop approximations at the MPP for the intermediate variables \mathbf{y} , especially when the hard constraints are explicit expressions in \mathbf{y} . In the current work, only linear approximations are used.

5.2.3 Approximate RBO

The approximate RBO consists of various Subspace Optimizers (SSOs) and a Coordination Procedure (CP). All the subspace optimizations can be implemented in parallel fashion. All the SSOs perform RBO but each has control of only certain design variables. There could be shared design variables among SSOs. There are additional move-limit constraints in each SSO and the SSOs perform approximate deterministic and deterministic analysis.

In the i^{th} SSO, the design variables \mathbf{d}_i , a subset of \mathbf{d} , are allowed to vary. In this SSO, \mathbf{y}_i , i.e the output of i^{th} CA is computed by using the approximation for the non-local state variables, $\tilde{\mathbf{y}}_{cnvgd}^{nl,i}$, a subset of $\tilde{\mathbf{y}}_{cnvgd}$, and it can be expressed as $\mathbf{y}_i = CA_i(\mathbf{d}, \mathbf{p}, \tilde{\mathbf{y}}_{cnvgd}^{nl,i})$. In each SSO, the reliability constraints are obtained from either $\tilde{\mathbf{g}}^{rbo}$ or obtained by performing a reliability analysis on $\tilde{\mathbf{g}}^R(\mathbf{x}, \boldsymbol{\eta})$.

Each SSO develops an optimal design and other information required for the next

phase which is the Coordination Procedure (CP). A new overall design, \mathbf{d}^{k+1} is found after the CP. In this work, the CP proposed by Renaud and Gabriele[17] is used, along with user-defined move-limits. In this CP, the histories of the approximate f , \mathbf{g}^{rbo} and \mathbf{g}^D are stored for all the designs evaluated during the SSOs, within the user-defined move-limits. Quadratic approximations for f , \mathbf{g}^{rbo} and \mathbf{g}^D are determined, using the approximate information obtained from the SSOs, about the current design setting. The approximations are created such that they have exact sensitivities at the current design, which is available from the output of rSA. A system of linear equations of the form $\mathbf{A}\mathbf{h} = \mathbf{b}$ can be written with respect to the unique terms in the Hessian Matrix, \mathbf{h} , as an unknown vector. The solution, \mathbf{h}^* is computed using a pseudo inverse technique[81], i.e. $\mathbf{h}^* = \text{pinv}(\mathbf{A})\mathbf{b}$, where $\text{pinv}(\mathbf{A})$ is the Moore-Penrose pseudo inverse of matrix \mathbf{A} . The system level approximate optimization problem, i.e. the RBO is solved using these quadratic approximations.

Trust region management strategies could be used to set the move-limits for the SSOs and CP, such as the one presented in Wujek and Renaud[16]. But for the test problems in this work, constant move-limits were used and the same move-limits were used for SSOs and CP. The new design from the CP is carried over to the rSA and this entire process is repeated in an iterative fashion till convergence is achieved. The convergence is achieved when $\|\mathbf{d}^{k+1} - \mathbf{d}^k\|$ is less than a user-defined tolerance.

5.3 Improved CSSO Approach for RBO

In the RBO-CSSO approach, the MPP searches in the rSA are performed using SA, i.e. exact multidisciplinary analysis. MPP searches are equality constrained optimization problems and hence decomposition strategies can be used to efficiently perform MPP searches for multidisciplinary problems. Implementation of such strategies in the rSA will increase the computational efficiency of the CSSO framework that is used for RBO for highly coupled problems. Such an improved CSSO approach for RBO will be referred

to as RBO-CSSO2 in this work[82]. Two decomposition strategies have been used in this work. Both these decomposition strategies require an algorithm that can solve bounded MPP searches. The details of the decomposition strategies and the bounded MPP searches follow.

5.3.1 A CSSO Approach for rSA

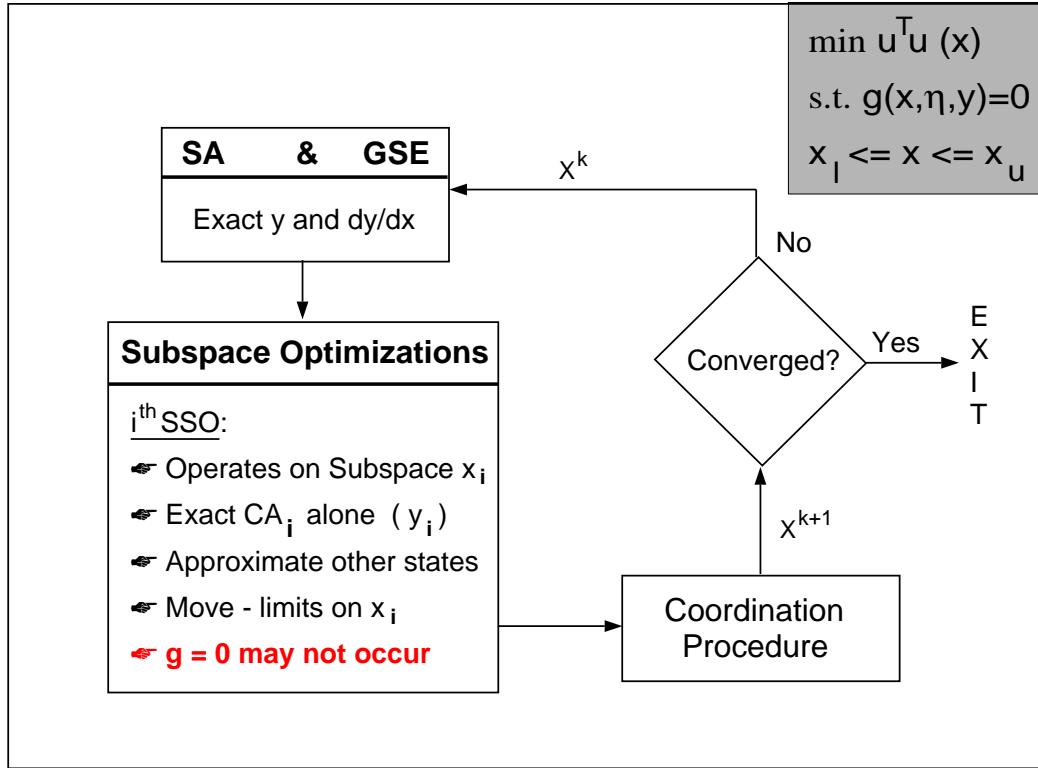


Figure 5.3. A CSSO Approach for MPP Searches

Figure 5.3 illustrates a CSSO approach for performing the MPP search for a multidisciplinary system. The CSSO version of the MPP search will be herein referred to as MPP-CSSO. The number of MPP-CSSOs in a rSA will equal the number of hard constraints. The final optimum from each MPP-CSSO is the MPP in X-space of the corresponding hard constraint. It has to be noted that MPP-CSSO will solve the **U**-space

MPP formulation in \mathbf{X} space. So the solution obtained from MPP-CSSO, \mathbf{x}^* , corresponds to the \mathbf{U} -space MPP, \mathbf{u}^* , obtained by solving Equations 3.14 and 3.15, i.e. $\mathbf{x}^* = \mathbf{T}^{-1}(\mathbf{u}^*)$. While performing the MPP search in \mathbf{X} space, the bound constraints, $\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$ are also required. These bounds can be based on intervals on which the individual distribution function of each component of \mathbf{x} is defined.

The MPP-CSSO consists of the following steps:

1. Initially a SA and GSE are performed at a given value of \mathbf{x} to obtain the corresponding hard constraint function and sensitivity information.
2. Then Subspace Optimizations (SSOs) are performed. The number of SSOs equals the number of disciplines or CAs. In SSO_i , only certain \mathbf{x} 's are allowed to vary. It has to be noted that, it is generally easier to assign the subspace optimization variables in \mathbf{x} space than in \mathbf{u} space. This is because, for a general \mathbf{x} , whose components are all statistically dependent, the k^{th} component of \mathbf{u} depends on the previous $k - 1$ components of \mathbf{x} , while using a Rosenblatt Transformation (see Appendix A). In SSO_i , only evaluations of CA_i (local) are allowed. Linear approximations are typically used to approximately evaluate the output of other CAs, based on the results from the SA.
3. The results of the SSOs are coordinated using a Coordination Procedure (CP). Various CPs exist in various versions of CSSO. The CP that is used in this work is similar to the one proposed by Renaud and Gabriele[17]. In this CP, a quadratic approximation is built for the hard constraint from the SSO iteration history. The \mathbf{U} space MPP formulation is solved using the quadratic approximation to the hard constraint.
4. Repeat steps 1 to 3 with the new \mathbf{x} from step 3 until convergence is achieved.

The SSOs and CP solve the \mathbf{U} space MPP formulation, with additional move-limit or trust region constraints on the corresponding variable x 's. So, if the trust region is small, then the hard constraint may not be active, or in other words, the limit state surface does not intersect or touch the trust region. Standard algorithms for MPP searches like HL-RF or SQP cannot be used in such a case. Special algorithms are required for the MPP searches and these will be presented in the next section.

5.3.2 A Local CA Based MPP Search

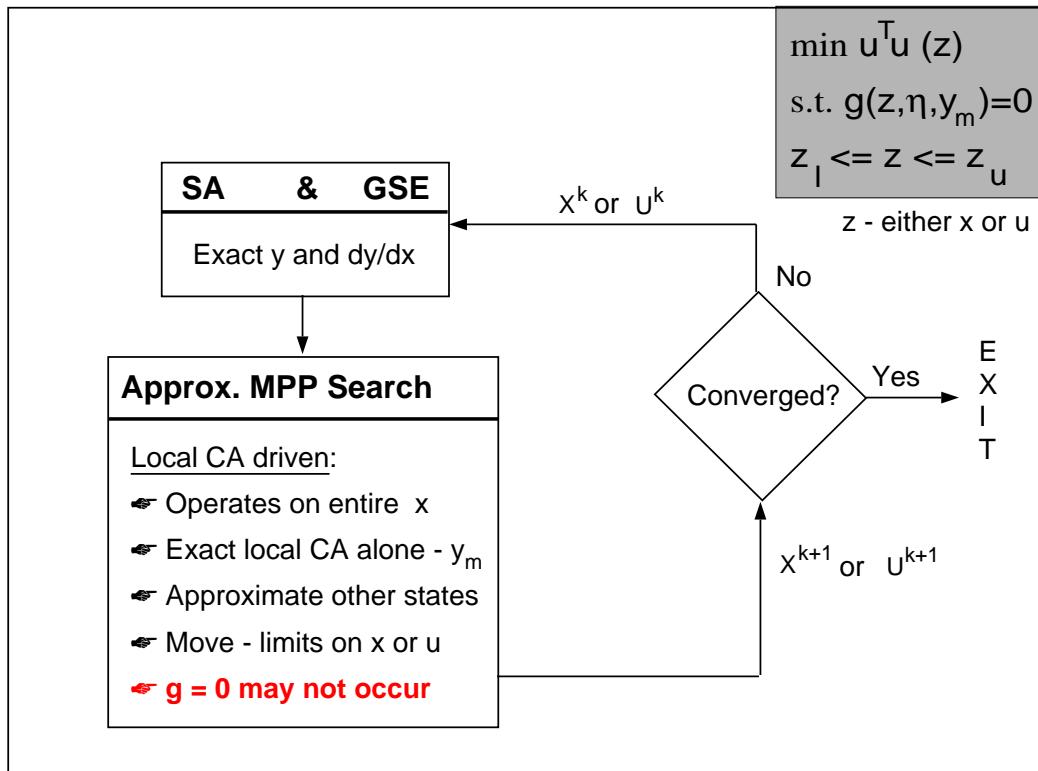


Figure 5.4. Local CA Based MPP Searches

In many of the multidisciplinary design problems, the hard constraints are just explicit functions of certain discipline outputs alone, for example stress constraints explicitly depend on the stresses obtained from the statics discipline, frequency constraints explicitly

depend on the frequencies obtained from the dynamics discipline, etc. It has to be noted that a discipline's outputs depend on the other disciplines' outputs in a multidisciplinary problem. Hence the hard constraints under consideration are potentially implicit functions of all the disciplines' outputs, even-though they are explicit functions of certain discipline's outputs alone. The MPP search for such a hard constraint, as explained earlier, is still a MDO problem. But such an MPP search is a special MDO problem where the merit does not depend on any discipline or SA, and the constraint explicitly depends on only one discipline. For a standard MDO problem, say a traditional design optimization problem, the merit and all the constraints explicitly depend on the outputs of all the disciplines. This suggests that the standard MDO approaches can be modified for solving such MPP searches.

When a CSSO approach is used, all the SSOs are unnecessary since the hard constraint in the MPP search explicitly depends on only one of the CA. For example, if the hard constraint explicitly depends on \mathbf{y}_m , the output of CA_m , then all the SSOs can be replaced with only one optimization which is driven by the local CA, i.e. CA_m alone. This local CA based optimizer is allowed to vary all the optimization variables of the MPP search, which can be either \mathbf{x} or \mathbf{u} . It has to be noted, that in this approach there is no problem associated with partitioning or allocation of the optimization variables, i.e. \mathbf{x} or \mathbf{u} , for the SSOs, hence either \mathbf{x} or \mathbf{u} can be used with this approach. In this approach, a CP is not required since only one optimizer exists unlike many SSOs in a standard CSSO approach. The framework for such a local CA based MPP search is shown in Figure 5.4. \mathbf{z} represents either \mathbf{x} or \mathbf{u} . This framework is a special case of the MPP-CSSO approach with just a single SSO that can control all the optimization variables. The details of those steps in the local CA based MPP search that are common to the MPP-CSSO approach, are exactly the same as explained earlier and will not be repeated here. There exist trust region constraints in the local CA based optimizer and as explained earlier, the hard constraints

(equality constraints) will not be active when the trust regions are small. Hence, special algorithms for the bounded MPP searches are required and is explained in the following section.

5.4 Bounded MPP Searches

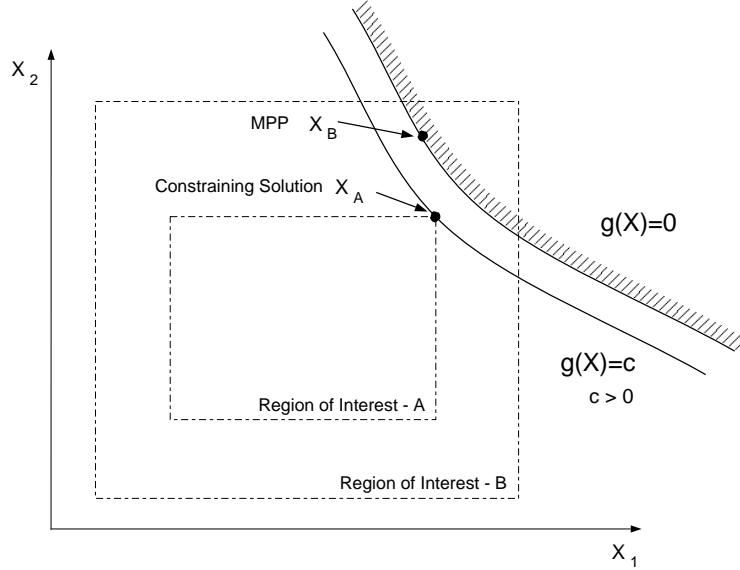


Figure 5.5. Bounded MPP Search

An algorithm for an MPP search is needed that is robust, one that can automatically detect if a hard constraint is active or not within a given region of interest, defined by bounds on the random variables, while searching for the MPP. Figure 5.5 shows two cases A and B, where the solution for the A-region of interest is x_A which occurs on minimum g contour in the domain A, whereas the solution for the B-region of interest is x_B and is the MPP point, since the region of interest contains the actual MPP. In this section, algorithms will be presented that can give solutions x_A and x_B when implemented for scenarios A and B respectively. This can be done by essentially performing a bounded MPP search that relaxes the equality constraint $g = 0$ in the MPP formulation. The regions of interest

are the move-limit or trust region constraints that arise in the MPP-CSSO approach and local CA based MPP search. Hence the bounded MPP search algorithms can be used for the SSOs and the local CA based optimizer since they give a solution corresponding to minimum infeasibility even if $g = 0$ does not occur within the move-limits, unlike standard algorithms like Sequential Quadratic Programming (SQP) that cannot handle such cases.

Another advantage of a bounded search algorithm is that it can be used for screening hard constraints or failure modes in a problem. The regions of interest can be based on a sufficiently high probability content region, for example $\pm 6\sigma$ limits. Only if $g = 0$ occurs within this region of interest, then the hard constraint can be considered important for the design optimization. This is important because for large problems, there are constraints that are not active in certain regions of the design space. Usually intuition and engineering judgment is used to determine the critical constraints. This will work for small problems, but need not work for large problems and especially in RBO where the design may undergo significant changes. Standard reliability analysis could fail or lead to excessive functional evaluations if a hard constraint is far from being active i.e. β is very large.

As has been noted before, to implement the decomposition strategies in the MPP search and to perform the constraint screening, optimization algorithms are needed that can accept an infeasibility in the equality constraint. Two algorithms are presented here. The first algorithm is based on standard techniques used in Trust Region-based Equality Constrained Optimization[83, 84]. The second algorithm is based on a variation of a standard Sequential Quadratic Programming approach, which is referred to as Elastic Sequential Quadratic Programming[85] in this dissertation.

5.4.1 Trust Region-based Equality Constrained Optimization

One of the algorithms considered for solving the bounded MPP searches is the Trust Region-based Equality Constrained Optimization (TRECO). The TRECO algorithm that is used here is based on SQP techniques. The TRECO can be used to perform a bounded MPP search in both \mathbf{x} or \mathbf{u} space. The TRECO algorithm, solves the MPP search given by the following optimization problem:

$$\min_{\mathbf{z}} \quad \psi(\mathbf{z}) \quad (5.4)$$

$$\text{s.t.} \quad g(\mathbf{z}) = 0 \quad (5.5)$$

$$\mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u \quad (5.6)$$

where $\psi(\mathbf{z}) = \frac{1}{2}\mathbf{u}^T\mathbf{u}$ and \mathbf{z} denotes either \mathbf{x} or \mathbf{u} . The variable bounds given by Equation 5.6 are strictly satisfied throughout the TRECO algorithm. For cases when $g(\mathbf{z}) = 0$ does not occur within the variable bounds, the TRECO algorithm gives the solution corresponding to minimum infeasibility given by the following optimization problem

$$\min_{\mathbf{z}} \quad \|g(\mathbf{z})\|^2 \quad (5.7)$$

$$\text{s.t.} \quad \mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u \quad (5.8)$$

The problem in Equations 5.7-5.8 can be solved first to obtain a feasible point. If no feasible point exists then one need not solve Equations 5.4-5.6. This becomes similar to an inverse reliability analysis. But if a feasible point exists then Equations 5.4-5.6 have to be separately solved, and this requires additional computational effort. It is desired that the TRECO algorithm simultaneously solve Equations 5.7-5.8 and Equations 5.4-5.6.

The TRECO algorithm is a trust region method, where the iterates are constrained within trust-regions during every iteration. The iterates are obtained by solving two sub-problems. One subproblem solves the problem given by Equations 5.7-5.8, and the other

subproblem solves the problem given by Equations 5.4-5.6. Both the subproblems are cast as Quadratic Programming (QP) problems. The standard TRECO algorithms typically assume that a feasible solution exists. The algorithm presented here is a modified version of standard techniques. It does not require the existence of a feasible solution and allows the algorithm to converge to the solutions of the problem given by Equations 5.7-5.8.

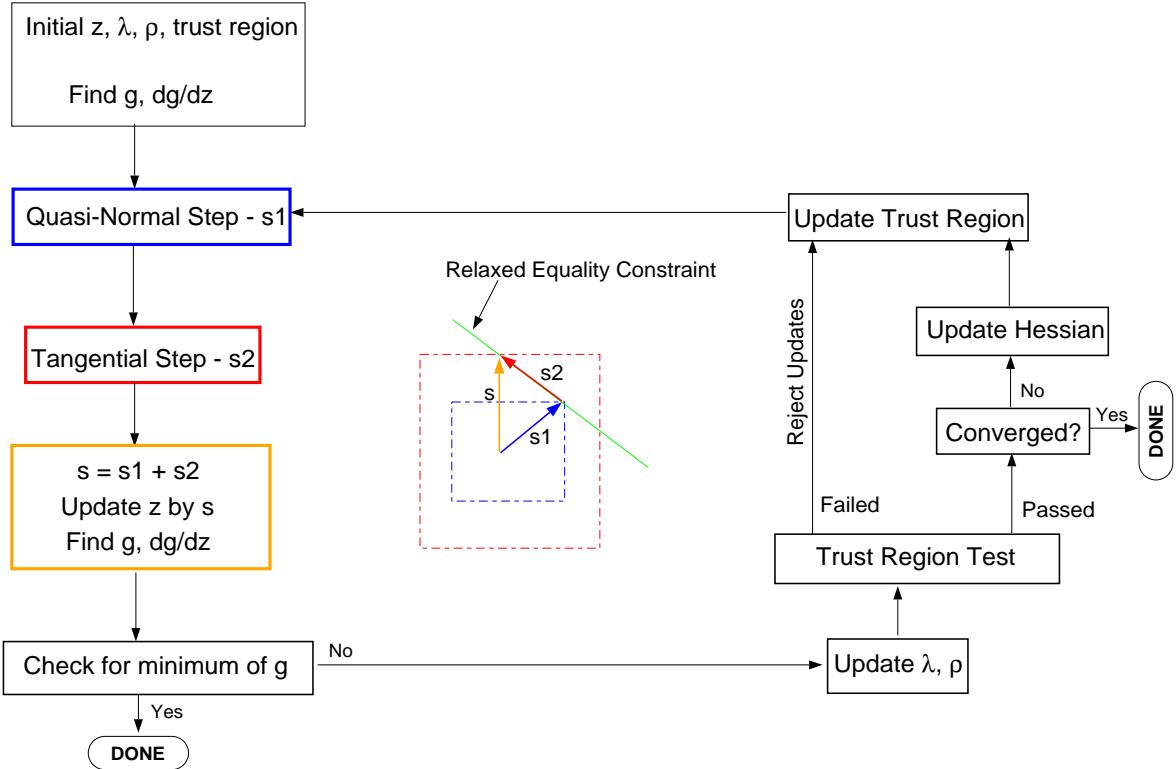


Figure 5.6. Flowchart of the Trust Region-based Equality Constrained Optimization

The flowchart of the TRECO algorithm is shown in Figure 5.6. Given a starting point \mathbf{z}^0 , an initial choice of Lagrangian multiplier λ_0 , penalty parameter $\rho^{-1} > 0$ (λ and ρ are required for the Lagrangian and Augmented Lagrangian) and an initial trust region radius or move-limit Δ^0 , the following steps are performed, starting with $k = 0$, where k is the iteration counter.

1. Compute Quasi-Normal Step: The first step is to compute a quasi-normal step, \mathbf{s}_n , that finds the minimum absolute value of the linear approximation of g at \mathbf{z}^k , $\tilde{g}^k(\mathbf{s})$, occurring within certain bounds on \mathbf{s}_n . The linear approximation, $\tilde{g}^k(\mathbf{s})$, is given by the following equation.

$$\tilde{g}^k(\mathbf{s}) = g(\mathbf{z}^k) + \nabla_z g(\mathbf{z}^k)\mathbf{s}$$

This quasi-normal step can be computed using the following Quadratic Programming (QP) problem,

$$\min_{\mathbf{s}_n} \quad \left| \tilde{g}^k(\mathbf{s}_n) \right|^2 \quad (5.9)$$

$$\mathbf{s}_{n,l} \leq \mathbf{s}_n \leq \mathbf{s}_{n,u} \quad (5.10)$$

The bounds on \mathbf{s}_n are such that the result lies in a reduced trust region obtained by scaling Δ^k by factor v , $0 < v < 1$. Typically v is selected in the range 0.5 to 0.8. It has to be made sure that the updates are within the bounds in \mathbf{z} , also. So the bounds are given by the following equations

$$\begin{aligned} \mathbf{s}_{n,l} &= \max(\mathbf{z}_l - \mathbf{z}^k, -v\Delta^k) \\ \mathbf{s}_{n,u} &= \min(\mathbf{z}_u - \mathbf{z}^k, v\Delta^k) \end{aligned}$$

The quasi-normal step finds the minimum of the infeasibility function, $|\tilde{g}^k(\mathbf{s}_n)|^2$ that occurs within the trust region in \mathbf{s}_n . The step is called quasi-normal because it is close to the normal of the infeasibility function at the solution when the bound constraints are active. If the trust region constraint on \mathbf{s}_n were based on l-2 norm i.e. trust regions were spherical or elliptical, then \mathbf{s}_n would truly be normal to the infeasibility function, when the trust region constraint is active.

2. Compute Tangential Step: After the quasi-normal step is computed, a tangential

step, \mathbf{s}_t , is computed by solving the following QP problem.

$$\min_{\mathbf{s}_t} \quad \tilde{L}^k(\mathbf{s}_n + \mathbf{s}_t, \lambda^k) \quad (5.11)$$

$$\text{s.t.} \quad \tilde{g}^k(\mathbf{s}_n + \mathbf{s}_t) = \tilde{g}^k(\mathbf{s}_n) \quad (5.12)$$

$$\mathbf{s}_{t,l} \leq \mathbf{s}_t \leq \mathbf{s}_{t,u} \quad (5.13)$$

where

$$\begin{aligned} \tilde{L}^k(\mathbf{s}, \lambda) &= L(\mathbf{z}^k, \lambda) + \nabla_z L(\mathbf{z}^k, \lambda) \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{H}^k \mathbf{s} \\ L(\mathbf{z}, \lambda) &= \psi(\mathbf{z}) + \lambda g(\mathbf{z}) \\ \mathbf{s}_{t,l} &= \max(\mathbf{z}_l - \mathbf{z}^k - \mathbf{s}_n, -\Delta^k - \mathbf{s}_n) \\ \mathbf{s}_{t,u} &= \min(\mathbf{z}_u - \mathbf{z}^k - \mathbf{s}_n, \Delta^k - \mathbf{s}_n) \end{aligned}$$

\tilde{L}^k is a quadratic model for $L(\mathbf{z}, \lambda)$, the Lagrangian of the MPP formulation given by Equations 5.4 and 5.5. Note that this is not the Lagrangian when the bounds (Equation 5.6) are included. \mathbf{H}^k , is the approximate Hessian of $L(\mathbf{z}, \lambda)$ at \mathbf{z}^k . An initial guess, \mathbf{H}_0 , is made for the Hessian, and for subsequent iterations a Hessian update scheme is used to calculate the approximate Hessian, which will be explained later in the algorithm. The tangential step basically minimizes the quadratic approximation of the Lagrangian, with a linear model for the equality constraint. It can be seen that the equality constraint given by Equation 5.12 requires the left hand side equal to the value of g obtained from the quasi-normal step. The use of this relaxed equality constraint makes this formulation always feasible. This is called the tangential step because \mathbf{s}_t is tangential to the null space of $\nabla_z g$, or in other words, \mathbf{s}_t lies in the tangential plane of the infeasibility function at solution obtained after the quasi-normal step. It can be seen that the TRECO solves Equations 5.7-5.8 and Equations 5.4-5.6 simultaneously, using the quasi-normal and tangential steps respectively.

Typically Equation 5.12 is eliminated by performing an orthogonal transformation of variables so that the optimization variables are in the null space of $\nabla_z g$. This transformation can be performed using QR factorization of $\nabla_z g$. The transformed problem is a simple minimization problem with a quadratic merit function and trust region constraints and hence can be easily solved. In the present work, the tangential step is solved using a QP instead and no transformation is performed. This is done so that the Lagrangian multipliers from the tangential step solution can be used for updating the Lagrangian multiplier λ , which will be explained later.

3. Update Lagrange multiplier and penalty term: Standard TRECO techniques typically update the Lagrangian multiplier based on a least squares solution of gradient of Lagrangian, i.e. $\nabla_z L(\mathbf{z}^k, \lambda^{k+1}) = 0$, with λ^{k+1} as the unknown. This update rule does not take the variable bounds into consideration. The variable bounds can be taken into account by a Coleman-Li scaling technique[86], where $\nabla_z L$ is scaled by a matrix \mathbf{S} , that is based on Karush-Kuhn Tucker conditions (first order optimality conditions) for Equations 5.4-5.6. The scaling matrix, \mathbf{S} , is a diagonal matrix with the i^{th} diagonal term given by the following equations:

$$\mathbf{S}_{ii} = \begin{cases} (\mathbf{z}_u)_i - \mathbf{z}_i & \text{if } (\nabla_z L)_i < 0 \text{ and } (\mathbf{z}_u)_i < \infty \\ \mathbf{z}_i - (\mathbf{z}_l)_i & \text{if } (\nabla_z L)_i \geq 0 \text{ and } (\mathbf{z}_l)_i > -\infty \\ 1 & \text{otherwise} \end{cases} \quad (5.14)$$

The Lagrangian multiplier update is now obtained from the least squares solution of $\mathbf{S}\nabla_z L(\mathbf{z}^k, \lambda^{k+1}) = 0$. Das[87] indicates that there are issues regarding such an update because of the dependence of \mathbf{S} on λ^{k+1} that necessitates an iterative scheme. It has to be noted that in standard techniques, the Coleman-Li scaling of $\nabla_z L$ is also done during the tangential step.

In this work, an alternative and a simpler rule for the Lagrangian multiplier update is

used that is given by the following equation:

$$\lambda^{k+1} = \lambda^k + \lambda_t \quad (5.15)$$

where λ_t is the Lagrangian multiplier corresponding to the equality constraint in the tangential step. This update rule can be obtained by matching the first order optimality conditions of the tangential step and original problem at the new point obtained after the tangential step, and by assuming that the quadratic model of the Lagrangian and the linear model of the hard constraint are exact. This update rule seems to be a good choice and the algorithm performs quite well for the test problems implemented to date.

El-Alem's scheme, as presented in Dennis et. al.[84], is used to update the penalty parameter, ρ . The penalty parameter update rule is based on the predicted reduction, $\text{pred}^k(\mathbf{s}^k; \rho^{k-1})$, in the augmented Lagrangian, Θ , based on the quadratic model of the Lagrangian and a linear model of constraint, g . The update rule is given as follows:

$$\begin{aligned} \text{If } \quad \text{pred}^k(\mathbf{s}^k; \rho^{k-1}) \geq \frac{\rho^{k-1}}{2} \left[|\tilde{g}^k(0)|^2 - |\tilde{g}^k(\mathbf{s}^k)|^2 \right] \text{ then} \\ \rho^k = \rho^{k-1} \end{aligned} \quad (5.16)$$

else

$$\rho^k = \frac{2 [\tilde{L}^k(\mathbf{s}^k, \lambda^{k+1}) - \tilde{L}^k(0, \lambda^k)]}{|\tilde{g}^k(0)|^2 - |\tilde{g}^k(\mathbf{s}^k)|^2} + \rho_{constant} \quad (5.17)$$

where

$$\text{pred}^k(\mathbf{s}^k; \rho^{k-1}) = \tilde{\Theta}^k(0, \lambda^k, \rho^{k-1}) - \tilde{\Theta}(\mathbf{s}^k, \lambda^{k+1}, \rho^{k-1}) \quad (5.18)$$

$$\Theta(\mathbf{z}, \lambda, \rho) = L(\mathbf{z}, \lambda) + \rho |g(\mathbf{z})|^2 \quad (5.19)$$

$$\tilde{\Theta}^k(\mathbf{s}, \lambda, \rho) = \tilde{L}^k(\mathbf{s}, \lambda) + \rho |\tilde{g}^k(\mathbf{s})|^2 \quad (5.20)$$

$\tilde{\Theta}$ is the approximate augmented Lagrangian, based on the approximate models for the Lagrangian and the constraint, g . $\rho_{constant}$ is some preselected positive constant. The

penalty parameter update rule guarantees a non-decreasing sequence of penalty parameters.

4. **Evaluate Trial Step:** The trial step can be calculated, $\mathbf{s}^k = \mathbf{s}_n + \mathbf{s}_t$ and the trial update can be calculated, $\mathbf{z}^{k+1} = \mathbf{z}^k + \mathbf{s}^k$. Evaluate $g(\mathbf{z}^{k+1})$ and $\nabla_z g(\mathbf{z}^{k+1})$.

5. **Check for extrema of g :** After the evaluation of the trial step, one can check for the optimality conditions for the constrained case, given by Equations 5.7-5.8. This is not done in standard TRECO algorithms. The satisfaction of the optimality conditions for the constrained case can be done by calculating the projected gradients of $g(\mathbf{z}^{k+1})$ onto the variable bounds, given by $P_1(\nabla_z g(\mathbf{z}^{k+1}))$ and $P_2(\nabla_z g(\mathbf{z}^{k+1}))$. The i^{th} components of the projected gradients are given by the following equations:

$$P_1(\nabla_z g(\mathbf{z}))_i = \begin{cases} \min((\nabla_z g(\mathbf{z}))_i, 0) & \text{if } (\mathbf{z}_i - (\mathbf{z}_l)_i) < \epsilon_4 \min((\mathbf{z}_u)_i - (\mathbf{z}_l)_i, 1), \\ \max((\nabla_z g(\mathbf{z}))_i, 0) & \text{if } (\mathbf{z}_i - (\mathbf{z}_u)_i) < \epsilon_4 \min((\mathbf{z}_u)_i - (\mathbf{z}_l)_i, 1), \\ (\nabla_z g(\mathbf{z}))_i & \text{otherwise.} \end{cases}$$

$$P_2(\nabla_z g(\mathbf{z}))_i = \begin{cases} \max((\nabla_z g(\mathbf{z}))_i, 0) & \text{if } (\mathbf{z}_i - (\mathbf{z}_l)_i) < \epsilon_4 \min((\mathbf{z}_u)_i - (\mathbf{z}_l)_i, 1), \\ \min((\nabla_z g(\mathbf{z}))_i, 0) & \text{if } (\mathbf{z}_i - (\mathbf{z}_u)_i) < \epsilon_4 \min((\mathbf{z}_u)_i - (\mathbf{z}_l)_i, 1), \\ (\nabla_z g(\mathbf{z}))_i & \text{otherwise.} \end{cases}$$

where ϵ_4 is some preselected tolerance. If $\|P_1(\nabla_z g(\mathbf{z}^{k+1}))\| \leq \epsilon_5 \max(1, \|\nabla_z g(\mathbf{z}^{k+1})\|)$ then the minimum of g is obtained at \mathbf{z}^{k+1} within or on the bounds in \mathbf{z} , where ϵ_5 is some preselected tolerance. The projected gradient is based on the first order optimality conditions of the optimization problem: $\min_{\mathbf{z}} g(\mathbf{z}) | \mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u$. This condition happens when the g remains greater than zero throughout the given bounds on \mathbf{z} .

Similarly if $\|P_2(\nabla_z g(\mathbf{z}^{k+1}))\| \leq \epsilon_5 \max(1, \|\nabla_z g(\mathbf{z}^{k+1})\|)$ then the maximum of g is obtained at \mathbf{z}^{k+1} within or on the bounds in \mathbf{z} . This projected gradient is based on the first order optimality conditions of the optimization problem: $\max_{\mathbf{z}} g(\mathbf{z}) | \mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u$. This condition happens when g remains lower than zero throughout the given bounds on \mathbf{z} .

When any of these two conditions occur it means that this is the best local solution for formulation 5.4-5.5, hence exit. If any of the bounds are active, then it means this corresponds to a solution of the type x_A in the Figure 5.5. If none of the bounds are active, then it means that a local minima or maxima for g exists. This step is essentially a convergence check, and hence can be viewed as a modification to the convergence checks typically done in standard algorithms.

6. Perform Trust Region Test and Update: The actual reduction, $\text{ared}^k(\mathbf{s}^k; \rho^k)$, and the predicted reduction, $\text{pred}^k(\mathbf{s}^k; \rho^k)$, in the augmented Lagrangian can be computed by using the following equations

$$\text{ared}^k(\mathbf{s}^k; \rho^k) = \Theta(\mathbf{z}^k, \lambda^k, \rho^k) - \Theta(\mathbf{z}^{k+1}, \lambda^{k+1}, \rho^k) \quad (5.21)$$

$$\text{pred}^k(\mathbf{s}^k; \rho^k) = \tilde{\Theta}^k(0, \lambda^k, \rho^k) - \tilde{\Theta}^k(\mathbf{s}^k, \lambda^{k+1}, \rho^k) \quad (5.22)$$

The reduction ratio, κ is the ratio of actual reduction and predicted reduction and given by the following equation:

$$\kappa = \frac{\text{ared}^k(\mathbf{s}^k; \rho^k)}{\text{pred}^k(\mathbf{s}^k; \rho^k)} \quad (5.23)$$

κ indicates how good the approximations used for the Lagrangian and the hard constraint really are. κ close to 1 indicates that the approximations are good. The step \mathbf{s}^k is rejected if $\kappa < 0$, i.e. set $\mathbf{z}^{k+1} = \mathbf{z}^k$, $\lambda^{k+1} = \lambda^k$ and $\rho^k = \rho^{k-1}$. The step and the updates are accepted otherwise. The trust region test is for the MPP formulation i.e. Equations 5.4 and 5.5. The trust region radius (move-limits) are updated according to the following rule:

$$\Delta^{k+1} = \gamma_0 \Delta^k \quad \text{if } \kappa < 0$$

$$\Delta^{k+1} = \gamma_1 \Delta^k \quad \text{if } 0 \leq \kappa \leq 0.25$$

$$\Delta^{k+1} = \Delta^k \quad \text{if } 0.25 < \kappa < 0.75$$

$$\Delta^{k+1} = \gamma_2 \Delta^k \quad \text{if } 0.75 \leq \kappa \leq 1.25$$

where the following values for γ s are chosen

$$\gamma_0 = 0.15,$$

$$\gamma_1 = 0.25,$$

$$\gamma_2 = 1.5 \quad \text{if } \max(\|\boldsymbol{\lambda}_{n,l}\|_\infty, \|\boldsymbol{\lambda}_{n,u}\|_\infty, \|\boldsymbol{\lambda}_{t,l}\|_\infty, \|\boldsymbol{\lambda}_{t,u}\|_\infty) > 0,$$

$$\gamma_2 = 1 \quad \text{otherwise}$$

where $\boldsymbol{\lambda}_{n,l}$ and $\boldsymbol{\lambda}_{n,u}$ are the multipliers corresponding to quasi-normal step lower bounds and upper bounds respectively. Similarly, $\boldsymbol{\lambda}_{t,l}$ and $\boldsymbol{\lambda}_{t,u}$ are the multipliers corresponding to tangential step lower bounds and upper bounds respectively. These multipliers are obtained from the solutions of QP subproblems solved during the computation of the quasi-normal and tangential steps. If any of these multipliers are greater than zero, then it indicates that trust region bound constraints are active.

7. Update Hessian: If the step is accepted, a Hessian update is required to compute the Hessian of L , \mathbf{H}^{k+1} , required for the next iteration. A standard damped Broyden-Fletcher-Goldfarb-Shanno (BFGS) update scheme is used to update Hessian in the work presented here. A standard BFGS update for an unconstrained optimization guarantees a positive semidefinite Hessian of the approximate merit function, when the initial guess is positive semidefinite. This is not true for a constrained optimization where the Hessian of the Lagrangian is being updated. Hence a BFGS update is modified to maintain positive semi-definiteness. The advantage of maintaining positive semi-definiteness is that the QP problems are bounded and a solution is guaranteed. The damped BFGS update, as

presented in Nocedal and Wright[63], is obtained by the following equations:

$$q' = L(\mathbf{z}^{k+1}, \lambda^{k+1}) - L(\mathbf{z}^k, \lambda^k) \quad (5.24)$$

$$\varphi = \begin{cases} 1 & \text{if } \mathbf{s}^k q' \geq (0.2 \mathbf{s}^k \mathbf{H}^k \mathbf{s}^k) \\ \frac{0.8 \mathbf{s}^k \mathbf{H}^k \mathbf{s}^k}{\mathbf{s}^k \mathbf{H}^k \mathbf{s}^k - \mathbf{s}^k q'} & \text{otherwise} \end{cases} \quad (5.25)$$

$$q = \varphi q' + (1 - \varphi) \mathbf{H}^k \mathbf{s}^k \quad (5.26)$$

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{q q^T}{\mathbf{s}^k q} - \frac{(\mathbf{H}^k \mathbf{s}^k)(\mathbf{H}^k \mathbf{s}^k)^T}{\mathbf{s}^k \mathbf{H}^k \mathbf{s}^k} \quad (5.27)$$

The use of Equation 5.26, that is obtained by modifying Equation 5.24 with φ , ensures positive semi-definiteness. In standard BFGS update, $\varphi = 1$. The BFGS update is given by Equation 5.27. This update is performed only if $\mathbf{s}^k q \neq 0$ and $\mathbf{s}^k \mathbf{H}^k \mathbf{s}^k \neq 0$.

It has to be noted that the damped BFGS update scheme has been selected only for convenience purposes since the QP problems are easily solvable. The damped BFGS update scheme is important for an algorithm like SQP that is based on a QP subproblem which can be unbounded for non-positive-semi-definite quadratic merit functions. One of the positive aspects of trust region methods are that nonlinear problems characterized by non-positive-semi-definiteness are treatable, since the solution of the underlying QP subproblem (assuming it is feasible) is bounded due to trust region constraints. Hence, it is not necessary to have Hessian update schemes that generate positive semi-definite Hessians. One can have a general Hessian Update scheme, like the Symmetric Rank One (SR1) update scheme. The resulting QP problem now becomes a global optimization problem, and one can use branch and bound techniques like the one implemented in MINQ routine[88].

8. Perform Convergence Checks: If the step is accepted then convergence checks are performed. Convergence is achieved if the following condition is satisfied:

$$\left(|g^{k+1}| < \varepsilon_1 \right) \text{ AND } \left[\left(\max |\mathbf{s}^k| < 2\varepsilon_2 \right) \text{ OR } \left(|\nabla_z \psi^{k+1} \mathbf{s}^k| < 2\varepsilon_3 \right) \right] \quad (5.28)$$

where ε_1 , ε_2 and ε_3 are preselected tolerances. This convergence check is for the problem given by Equations 5.4 - 5.5. The convergence criteria requires that the equality constraint be satisfied within a given tolerance and that either the step size is less than a given tolerance or the obtained decrease in the merit, ψ , is less than a given tolerance. If the convergence criterion is satisfied exit, or else go to step 1.

5.4.2 Elastic Sequential Quadratic Programming

In Elastic Sequential Quadratic Programming (ESQP), as presented by Marazzi and Nocedal[85], the following optimization problem for the MPP search is solved.

$$\min_{\mathbf{z}} \quad \psi(\mathbf{z}) + \xi(v + w) \quad (5.29)$$

$$\text{s.t.} \quad g(\mathbf{z}) - v + w = 0 \quad (5.30)$$

$$\mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u \quad (5.31)$$

$$v \geq 0, w \geq 0 \quad (5.32)$$

The equality constraint $g = 0$ is replaced with $g - v + w = 0$ in Equation 5.30 where v and w are elastic or slack variables that are greater than 0. The merit function is penalized by the term $\xi(v + w)$ in Equation 5.29, where ξ is a penalty term set at a sufficiently large value. Such a penalized SQP is used in the SNOPT optimization package[89] when the QP subproblem of the standard SQP problem, without the penalty terms and slack variables, is infeasible. Such a penalized approach is also referred to as the Tone's approach by Spellucci[90].

For the case, when $g = 0$ is obtainable, both the slack variables should go to zero. The penalty parameter, ξ has to be sufficiently large for this to happen. It can be shown that this is the case when $\xi > |\lambda^*|$, where λ^* is the Lagrangian multiplier corresponding to the equality constraint at the optimum solution. For the case, when $g = 0$ is not obtainable, it can be shown that the best constrained solution is obtained as given by Equations 5.7-5.8,

when ξ is sufficiently large. If ξ is not large enough, ESQP will give wrong results for both the feasible and infeasible cases. This was illustrated through various implementation studies which will be discussed in detail in Chapter 7.

In this work, the problem given by Equation 5.29-5.32 is solved using an SQP method based on line searches, that is implemented by the FMINCON subroutine available in the Matlab Optimization Toolbox. The convergence criteria for the ESQP routine is similar to that given in Equation 5.28 but with \mathbf{z} replaced by $\{\mathbf{z}, v, w\}$, ψ replaced by $\psi + \xi(v + w)$ and g replaced by $g - v + w$.

5.5 Summary

A traditional RBO for a multidisciplinary system design is very expensive since it requires the entire multidisciplinary analysis during the reliability analysis. Hence a Concurrent Subspace Optimization based methodology for RBO, called the RBO-CSSO approach, is proposed. This methodology is comprised of three main stages, namely, exact deterministic and reliability analysis, approximation building phase and approximate RBO phase. In the approximation building phase, approximations are developed for evaluation of deterministic quantities and reliability constraints during the approximate RBO phase. The approximate RBO phase consists of Subspace Optimizations taking place concurrently, and a Coordination Procedure that coordinates the results of the Subspace Optimizations. An extension of the RBO-CSSO approach, called the RBO-CSSO2, is also proposed, that employs decomposition strategies for the MPP searches in the reliability analysis. The proposed decomposition strategies include a Concurrent Subspace Optimization approach and a special case of the Concurrent Subspace Optimization approach that requires only one Subspace Optimization controlled by the discipline that outputs the hard constraint being considered in the MPP search. Both these decomposition strategies require an algorithm that can give a solution for an infeasible bounded

MPP search where the bounds are too small so that the equality constraint is infeasible.

Two algorithms are proposed for performing the bounded MPP searches, that can give the exact MPPs when the bounded MPP search is feasible and the best constrained solution for the infeasible case. The first algorithm, is a Trust Region-Based Equality Constrained Optimization (TRECO) algorithm. This approach mainly consists of two steps, a quasi-normal step that computes the best obtainable value of the equality constraint closest to zero, and a tangential step that minimizes the distance measure but constrains the optimizer to move along the contour corresponding to the smallest absolute value of the linearized equality constraint, which is obtained from the quasi-normal step. The quasi-normal step is based on a linear approximation of the equality constraint, and the tangential step is based on a quadratic approximation of Lagrangian and a linear approximation of the equality constraint. Convergence checks for the extrema of the equality constraint are additionally added to existing standard algorithms to account for the infeasible case. The algorithm features a trust region update scheme based on the comparison of predicted and actual function values at the end of a iteration. Both the quasi-normal and tangential steps are implemented as Quadratic Programming (QP) problems. A Lagrangian multiplier update based on the multipliers from the tangential step solution is used to account for the infeasible case. The second algorithm, is an Elastic Sequential Quadratic Programming (ESQP) algorithm, that additionally introduces two positive elastic variables so that the equality constraint is always feasible. The distance measure is penalized by a penalty term involving the elastic variables. The penalty term needs to be sufficiently large to obtain the correct solutions for the feasible and infeasible cases. This algorithm is implemented using a standard available SQP algorithm.

CHAPTER 6

MONTE CARLO SIMULATION IN RBO

In this chapter, various Monte Carlo Simulation (MCS) techniques used to estimate the probability of failure will be presented. Two main classes of the MCS techniques, namely the Indicator-based MCS and Conditional Expectation-based MCS techniques will be presented. A stratified sampling approach for series system probability of failure estimation is later discussed. Analytic sensitivities for the MCS techniques will then be derived and discussed. Certain MCS techniques will be chosen based on the smoothness properties and ease of computability of analytic sensitivities. Finally, a methodology for performing RBO using the chosen MCS techniques will be presented[91].

6.1 Monte Carlo Simulation Techniques

The probability of failure, P , is given by the following multidimensional integral:

$$P = \int_{\Omega_f} f_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z} \quad (6.1)$$

where Ω_f is the failure region, \mathbf{Z} is the random or uncertain variable vector (either original \mathbf{X} or transformed standard \mathbf{U} space), \mathbf{z} is an instantiation of \mathbf{Z} , and $f_{\mathbf{Z}}$ is the joint probability density function. Ω_f is given by $g(\mathbf{z}) < 0$ for a component failure mode. The integral is very expensive to compute numerically for large dimensions. An approximation to this probability of failure is provided by the First and Second Order Reliability Methods (FORM & SORM), that are based on linear and quadratic approximations re-

spectively to the limit state surfaces in the standard normal space. FORM and SORM can either under-predict or over-predict the probability of failure for limit state surfaces that are very nonlinear in the standard normal space. Moreover, sensitivity analysis of the reliability index obtained from FORM can wrongly result in increasing reliability index for increasing standard deviation of random variables for certain asymmetrical distribution types, as was shown by Sørensen and Enevoldsen[11]. MCS techniques are typically used to evaluate such multidimensional integrals accurately but more efficiently than numerical integration. They mainly fall in two categories; Indicator-based MCS and Conditional Expectation-based MCS. Details of these techniques and computation of its analytic sensitivities follow.

6.1.1 Indicator-based MCS

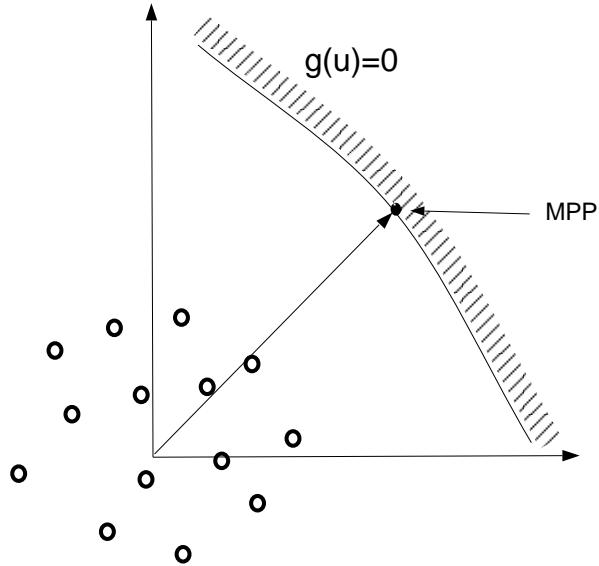


Figure 6.1. Standard Monte Carlo Simulation Technique

The probability of failure using a standard MCS with samples simulated according to

$f_{\mathbf{Z}}$ for a single failure mode can be estimated using the following equation.

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N I(g(\mathbf{z}_i) < 0) \quad (6.2)$$

In the above equation N is the number of sample points and I is the indicator function (1 if $g < 0$, 0 otherwise). The number of sample points required to estimate small magnitudes of probability of failure especially for high dimensional problems is extremely high. The standard MCS is illustrated in Figure 6.1, where the sample points are represented by the circles. Hence Quasi Monte Carlo methods are typically used.

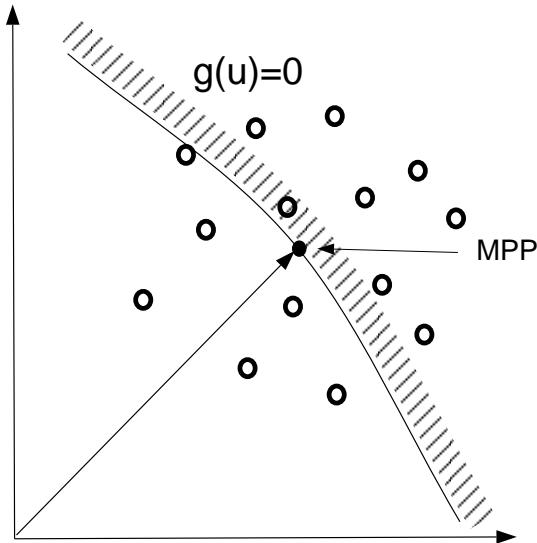


Figure 6.2. An Importance Sampling-based MCS Technique

A popular class of Quasi Monte Carlo methods are the Importance Sampling methods [52, 92]. In importance sampling, a simulation variable, \mathbf{V} , is chosen with a sampling density $h_{\mathbf{V}}$ such that most of samples are generated in important regions in \mathbf{Z} space that have major contributions to P . Hence P and its estimate for a single failure mode or

component can be written as follows:

$$P = \int_{\Omega_f} \frac{f_{\mathbf{Z}}(\mathbf{z}(\mathbf{v}))}{h_{\mathbf{V}}(\mathbf{v})} h_{\mathbf{V}}(\mathbf{v}) d\mathbf{v} \quad (6.3)$$

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N I(g(\mathbf{z}(\mathbf{v}_i)) < 0) \frac{f_{\mathbf{Z}}(\mathbf{z}(\mathbf{v}_i))}{h_{\mathbf{V}}(\mathbf{v}_i)} \quad (6.4)$$

An example of an importance sampling method is sampling around the MPP, which corresponds to a choice of standard normal random variable \mathbf{V} centered around the Most Probable Point (MPP) of $g(\mathbf{u})$ as illustrated in Figure 6.2. There are various techniques such as Adaptive Sampling, Asymptotic Importance Sampling[93], Stratified Sampling, Latin Hypercube Sampling etc. One of the main issues that arises in using these techniques is that analytic sensitivities cannot be obtained using most of these techniques, which will be discussed later. Another issue that arises in applying these techniques is the discontinuities of probability of failure estimates with changes in design. The discontinuities can be kept to a minimum by fixing the seeds of the random number generators and the number of sample points but the discontinuities can still arise. For the example mentioned earlier, where the sampling is performed around the MPP, changes in curvature of the limit state surface due to changes in design variables can result in shifting of sample points across the limit state surface that causes discrete jumps in probability of failure estimates.

6.1.2 Conditional Expectation-based MCS

There are other importance sampling techniques that are based on conditional expectations [92, 8]. These are the Directional Simulation and Axis-Orthogonal Simulation. Both these methods are typically done in \mathbf{U} -space and hence the discussion will be restricted to MCS in \mathbf{U} -space and for component failure estimation. As explained earlier in importance sampling, a simulation variable \mathbf{V} is chosen. In directional simulation, \mathbf{V} is transformed to random variables $\{R, \mathbf{A}\}$, where $\mathbf{V} = R\mathbf{A}$. R , the radial variable is the

length of \mathbf{V} i.e. $R = \|\mathbf{V}\|$ and the angular random variable \mathbf{A} is of unit length i.e. $\|\mathbf{A}\| = 1$. An instantiation of R is r and of \mathbf{A} is $\boldsymbol{\alpha}$. $h_{\mathbf{V}}(\mathbf{v})d\mathbf{v}$ can be written as $h_{\{R,\mathbf{A}\}}(\{r, \boldsymbol{\alpha}\})drd\boldsymbol{\alpha}$ and one can substitute the relation $h_{\{R,\mathbf{A}\}}(\{r, \boldsymbol{\alpha}\}) = h_R(r|\boldsymbol{\alpha})h_{\mathbf{A}}(\boldsymbol{\alpha})$. In directional simulation analytic or numerical integration is carried out in r conditioned on $\mathbf{A} = \boldsymbol{\alpha}$ and sampling is done in $\boldsymbol{\alpha}$ space according to sampling density h_A . The expression for P and its estimate are given by the following equations.

$$P = \int_{\|\boldsymbol{\alpha}\|=1} \left[\int_{g<0} \frac{f_Z(r\boldsymbol{\alpha})}{h_{\mathbf{V}}(r\boldsymbol{\alpha})} h_R(r|\boldsymbol{\alpha}) dr \right] h_{\mathbf{A}}(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \quad (6.5)$$

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N \int_{g<0} \frac{f_Z(r\boldsymbol{\alpha}_i)}{h_{\mathbf{V}}(r\boldsymbol{\alpha}_i)} h_R(r|\boldsymbol{\alpha}_i) dr \quad (6.6)$$

It is convenient to perform the calculations by sampling in \mathbf{v} according to $h_{\mathbf{V}}$ and by

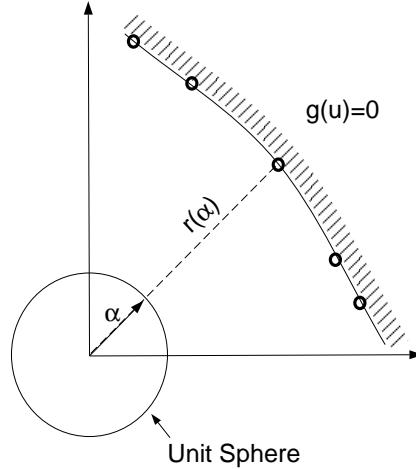


Figure 6.3. Directional Simulation

computing the integral in r (in square brackets in above equations) by fixing $\boldsymbol{\alpha} = \frac{\mathbf{V}}{\|\mathbf{V}\|}$. It has to be noted that the integral in r is not expensive to compute since it is an integration in one dimension. But to be able to perform the integration, the integration limits are required that can be obtained by solving $g(r\boldsymbol{\alpha}) = 0$ for r with $\boldsymbol{\alpha}$ fixed. Figure 6.3 shows a case with a single solution, $r(\boldsymbol{\alpha})$ and the integration limits are from $r(\boldsymbol{\alpha})$ to ∞ . Figure

6.4 shows a case with two solutions, $r_1(\alpha)$ and $r_2(\alpha)$ and hence the integration limits are from $r_1(\alpha)$ to $r_2(\alpha)$. It is possible to have cases with more than single solution. The root solving can be done using a Newton Raphson scheme. The interval in which the root is to be located can be chosen such that it corresponds to a range of high probability level, say $r = 0$ to 6 . Multiple solutions can be explored using branch and bound schemes. The root solving essentially projects the sample points radially onto the limit state surface. The circles in the Figure 6.3 are the projected sample points.

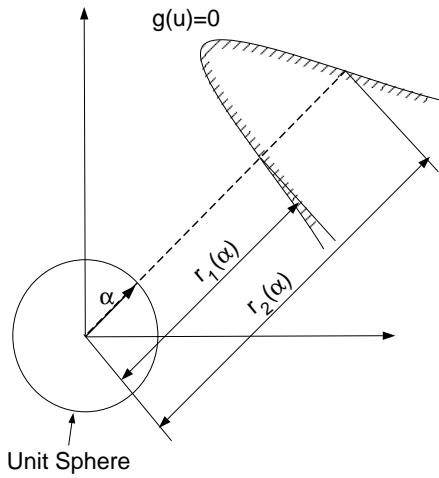


Figure 6.4. Multiple Roots in Directional Sampling

The integral in r is readily available in analytical form in certain cases. Equations 6.7 and 6.8 give the analytic expressions for estimates of P , assuming only one root for r exists, in \mathbf{U} -space i.e. $\mathbf{Z} = \mathbf{U}$ for the cases when \mathbf{V} is selected as \mathbf{U} (Uniform directional simulation), and when \mathbf{V} is selected as independent normally distributed random variables with mean \mathbf{u}_o and unit variance, respectively.

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N (1 - K_n(r(\boldsymbol{\alpha}_i)^2)) \quad (6.7)$$

$$\hat{P} = \frac{1}{N} \frac{K(0)}{\phi_n(\mathbf{u}_o)(2\pi)^{\frac{n-1}{2}}} \sum_{i=1}^N \left[\frac{\phi_1(\boldsymbol{\alpha}_i^T \mathbf{u}_o)}{K(\boldsymbol{\alpha}_i^T \mathbf{u}_o)} (1 - K_n(r(\boldsymbol{\alpha}_i)^2)) \right] \quad (6.8)$$

In the above equation, K_n is the χ^2 distribution function with n degrees of freedom and $K(\boldsymbol{\alpha}_i^T \mathbf{u}_o)$ is given by the following integral

$$K(x) = I_{n-1}(x) = \int_0^\infty s^{n-1} e^{-\frac{1}{2}(s-x)^2} ds \quad (6.9)$$

$K(0) = \Gamma\left(\frac{n}{2}\right) 2^{\frac{n}{2}-1}$ and for nonzero x the following recursion formula is required to find $K(x)$.

$$I_{n+1}(x) = nI_{n-1}(x) + xI_n(x) \quad (6.10)$$

$$I_0(x) = \sqrt{2\pi}\Phi(x) \quad (6.11)$$

$$I_1(x) = e^{-\frac{1}{2}x^2} \quad (6.12)$$

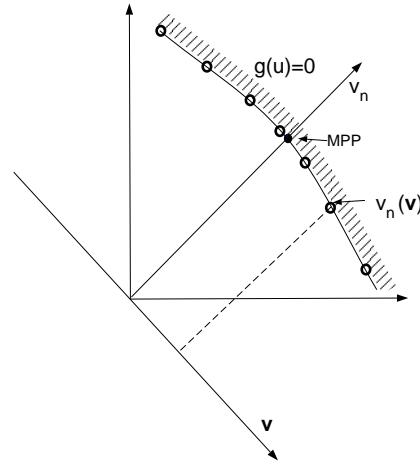


Figure 6.5. Component Reliability Using Axis Orthogonal Simulation

In an Axis-Orthogonal Simulation a new coordinate system $\{\mathbf{v}, \mathbf{v}_n\}$ is obtained by a linear or affine transformation of \mathbf{u} -space. This transformation is also used to transform the random variables \mathbf{Z} (i.e. \mathbf{U}) to $\{\mathbf{V}^\dagger, V_n\}$. The simulation variable \mathbf{V} is defined in \mathbf{v} space (which is $n - 1$ dimensional) and the sampling density is $h_{\mathbf{V}}$. It has to be noted that

\mathbf{V}^\dagger and \mathbf{V} are different random variables. For a single failure mode, v_n axis is picked such that it coincides with the vector joining the origin and the MPP, see Figure 6.5. A simple example for \mathbf{V} is a standard normal random vector, for this case. For parallel system as shown in Figure 6.6, the v_n axis is parallel to N_{mean} which is the mean vector of the normal vectors of g_1 and g_2 i.e. N_1 and N_2 , at their joint MPP. A simple example for \mathbf{V} is independent normally distributed random vector with mean μ and unit variance.

For a case with a single failure mode, there are many possible transformations that exist that transform \mathbf{u} to $\{\mathbf{v}, v_n\}$ and such a transformation can be found by a Gram-Schmidt orthogonalization of independent basis formed by v_n and $n - 1$ unit vectors of the \mathbf{u} space. Such a transformation can be written as $\{\mathbf{v}, v_n\} = \mathbf{G}\mathbf{u}$.

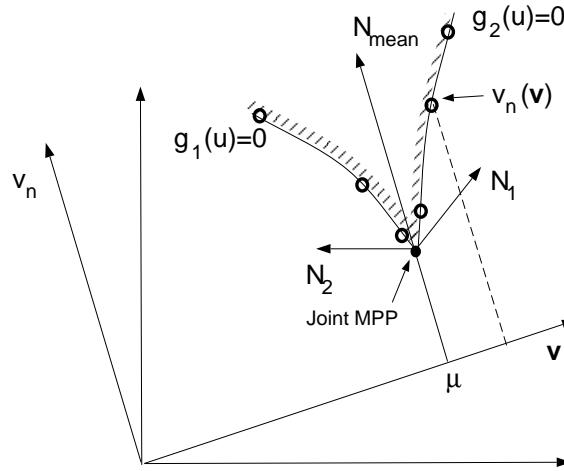


Figure 6.6. Parallel System Reliability Using Axis Orthogonal Sampling

P is computed by sampling in \mathbf{V} and an exact integral in v_n conditioned on $\mathbf{V} = \mathbf{v}$. v_n is analogous to r and \mathbf{v} is analogous to $\boldsymbol{\alpha}$ in the directional simulation. P and estimate of

P are given by the following equations.

$$P = \int_{\mathbb{R}^{n-1}} \left[\int_{g < 0} f_{V_n}(v_n | \mathbf{v}) dv_n \right] \frac{f_{\mathbf{V}^\dagger}(\mathbf{v})}{h_{\mathbf{V}}(\mathbf{v})} h_{\mathbf{V}}(\mathbf{v}) d\mathbf{v} \quad (6.13)$$

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N \left[\int_{g < 0} f_{V_n}(v_n) dv_n \right] \frac{f_{\mathbf{V}^\dagger}(\mathbf{v}_i)}{h_{\mathbf{V}}(\mathbf{v}_i)} \quad (6.14)$$

Like in Directional Simulation, the integral in v_n is a 1 dimensional integral. This is done by first solving for the roots of $g(\{\mathbf{v}_i, v_n\}) = 0$ for each sample \mathbf{v}_i . As in Directional Simulation, more than one root can exist. The root solving is equivalent to projecting the sample points along v_n axis. The 1-D integral is analytic and if a single root exists for the root solving problem, the estimate of probability of failure can be given by the following equation.

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N (1 - \Phi(v_n(\mathbf{v}_i))) \quad (6.15)$$

In both directional simulation and axis-orthogonal simulation techniques, it is easy to obtain analytic sensitivities since the sample points are projected on the limit state surface. This will be discussed later in detail. Discontinuities due to “shifting” of sample points across the limit state surface are absent for these MCS techniques. But discontinuities can still arise especially when the integration limit v_n or r for a given sample changes from a finite value to infinity or vice-versa with changes in design variables. This means that due to a change in the limit state surface the sample is no longer projectable or the sample that was earlier not projectable can now be projected onto the limit state surface. This happens when a local zero minimum of the 1-D root solving problem, when rewritten as an optimization problem $\min|g|$, occurs in either v_n or r direction for that corresponding sample. This essentially creates a variation in number of projectable sample points and hence causes a discontinuity in the probability of failure estimate. The magnitude of variation in the probability of failure estimate mainly depends on the magnitude of v_n .

or r at which this discontinuity arises. The larger this value, the smaller the variation. This phenomenon was observed in one of the test problems, which will be discussed in Chapter 7.

Now comparing the indicator-based MCS and conditional expectation-based MCS in terms of magnitude of the discontinuities, it is expected that in most application problems the former will exhibit larger discontinuities than the latter. This is because in indicator-based MCS potentially many samples can move across a limit state surface with curvature changes even for benign limit state surfaces, whereas the variation in number of projectable sample points in conditional expectation-based MCS will be fewer for such cases when there exist local zero minima in 1-D root solving problem.

6.1.3 Series System Probability of Failure

The probabilities of failure for pure series systems can be estimated using a stratified sampling approach. In a stratified sampling, sampling is done in more than one important region and then the probability estimates are determined as a weighted sum of the contributing probability estimate from each important region. In a stratified sampling approach for series system failure probability estimation, MPPs with respect to each failure mode are found, i.e. important region for each component failure mode is identified and sampling density for each failure mode is found. A weighted sampling density function is now defined by $h_{\mathbf{V}} = \sum_{j=1}^M w_j h_{\mathbf{V}^{(j)}}$, where M is the number of failure modes. In this approach, there are M simulation variables, $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \dots, \mathbf{V}^{(M)}$. The weights w_j can be chosen such that $w_j = \frac{p_j}{\sum_{i=1}^M p_i}$, where p_i is the estimate of probability of failure for the i^{th} failure mode, typically obtained from FORM. The probability of failure due to sample points of simulation variable $\mathbf{V}^{(j)}, \hat{P}^{(j)}$ using both directional and axis-orthogonal simu-

lation are given as follows

$$\hat{P}^{(j)} = \frac{1}{N_j} \sum_{i=1}^{N_j} \left[\int_{\cup[g_k < 0]} \frac{f_{\mathbf{Z}}(r\boldsymbol{\alpha}_i^{(j)})}{h_{\mathbf{V}}(r\boldsymbol{\alpha}_i^{(j)})} h_R(r|\boldsymbol{\alpha}_i^{(j)}) dr \right] \quad (6.16)$$

$$\hat{P}^{(j)} = \frac{1}{N_j} \sum_{i=1}^{N_j} \left[\int_{\cup[g_k < 0]} f_{V_n}(\mathbf{v}_n^{(j)}) d\mathbf{v}_n^{(j)} \right] \frac{f_{\mathbf{V}^*}(\mathbf{v}_i^{(j)})}{h_{\mathbf{V}}(\mathbf{v}_i^{(j)})} \quad (6.17)$$

where N_j is the number of simulations of $\mathbf{V}^{(j)}$. The series system probability of failure is the weighted average of all the contributing probabilities of failure i.e., $P = \sum_{j=1}^M w_j \hat{P}^{(j)}$. It has to be noted that the integration limits are now to be obtained by solving for the roots of $\min_k [g_k(r\boldsymbol{\alpha}_i)] = 0$ or $\min_k [g_k(\{\mathbf{v}_i, \mathbf{v}_n\})] = 0$, i.e., on the system failure domain defined by the series of the failure events.

It has to be noted also that transformation of the sample points in different coordinate systems are required in Axis-Orthogonal simulation during the evaluation of $h_{\mathbf{V}}$ in Equation 6.17. All the simulation variables are defined in coordinate systems obtained by linear transformations of the \mathbf{u} space. Hence if the coordinate system for the i^{th} simulation variable is obtained by the transformation $\{\mathbf{v}^{(i)}, \mathbf{v}_n^{(i)}\} = \mathbf{G}^{(i)}\mathbf{u}$. Then the transformation between coordinate systems of i^{th} and j^{th} simulation variables are given by $\{\mathbf{v}^{(i)}, \mathbf{v}_n^{(i)}\} = \mathbf{G}^{(i)} (\mathbf{G}^{(j)})^{-1} \{\mathbf{v}^{(j)}, \mathbf{v}_n^{(j)}\}$. Hence each sample point corresponding to a simulation variable has to be transformed using the above transformations to the coordinate systems of the other simulation variables during the computation of $h_{\mathbf{V}}$.

6.1.4 Sensitivities of Monte Carlo Simulations

In an RBO that is solved using a gradient-based optimizer, sensitivities of P are required with respect to distribution parameters, $\boldsymbol{\Theta}$, or limit state parameters, $\boldsymbol{\eta}$, that form the design variable vector, \mathbf{d} . The density function, $f_{\mathbf{X}}$, depends on $\boldsymbol{\Theta}$ and the limit state function, g , in \mathbf{X} space depends on $\boldsymbol{\eta}$. The sensitivity of P with respect to $\boldsymbol{\Theta}$ is given by

the following equation.

$$\frac{\partial P}{\partial \boldsymbol{\theta}} = \int_{\Omega_f} \frac{\partial f_{\mathbf{Z}}}{\partial \boldsymbol{\theta}} d\mathbf{z} \quad (6.18)$$

The above equation is valid when MCS is done in the original space i.e. \mathbf{X} . It has to be noted that the above integral can be computed using any MCS technique. The same simulations that was used to estimate P can be used to estimate $\frac{\partial P}{\partial \boldsymbol{\theta}}$.

If the MCS is done in \mathbf{u} space, both $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ will behave as limit state parameters in \mathbf{U} space, i.e. changes in $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ changes the limit state surface. Now let us represent the limit state parameter by $\boldsymbol{\tau}$ in a general sense, i.e., $\boldsymbol{\tau} = \{\boldsymbol{\theta}, \boldsymbol{\eta}\}$. The sensitivity of P with respect to $\boldsymbol{\tau}$ is given by the following equation.

$$\frac{\partial P}{\partial \boldsymbol{\tau}} = - \int_{g(\mathbf{z}, \boldsymbol{\tau})=0} \frac{\partial g}{\partial \boldsymbol{\tau}} \frac{f_{\mathbf{Z}}}{\|\nabla_z g\|} d\mathbf{z} \quad (6.19)$$

The above integral is performed on the $n - 1$ dimensional limit state surface $g(\mathbf{z}, \boldsymbol{\tau}) = 0$. This cannot be performed using the already existing samples in most of the indicator-based MCS, since the samples do not lie on this surface. The adaptive importance sampling technique[93] is an exception where the limit state surface is approximated as a parabolic surface and the surface integral can be done in the curvilinear coordinates of the parabolic surfaces. It has to be noted that the sensitivity is based on an approximation of limit state surface though.

On the other hand, in directional and axis orthogonal simulation all the sample points are strictly projected on to the limit state surface and hence the sensitivities of P can be very easily obtained. In these methods, the sensitivities of \hat{P} with respect to $\boldsymbol{\tau}$ would depend on the sensitivities of the roots used for exact 1 dimensional integration i.e. $\frac{\partial r(\boldsymbol{\alpha}_i)}{\partial \boldsymbol{\tau}}$

and $\frac{\partial v_n(\mathbf{v}_i)}{\partial \boldsymbol{\tau}}$. These values can be easily computed using the following relations

$$\frac{\partial r(\boldsymbol{\alpha}_i)}{\partial \boldsymbol{\tau}} = \frac{-\frac{\partial g}{\partial \boldsymbol{\tau}}}{(\nabla_u g \boldsymbol{\alpha}_i)} \quad (6.20)$$

$$\frac{\partial v_n(\mathbf{v}_i)}{\partial \boldsymbol{\tau}} = \frac{-\frac{\partial g}{\partial \boldsymbol{\tau}}}{\left(\nabla_u g \frac{\partial \mathbf{u}(\mathbf{v}_i, v_n)}{\partial v_n} \right)} \quad (6.21)$$

The same sample points are used for computing these sensitivities. It has to be noted that $\frac{\partial g}{\partial \boldsymbol{\tau}} = \frac{\partial g}{\partial \boldsymbol{\eta}}$ when $\boldsymbol{\tau} = \boldsymbol{\eta}$ and $\frac{\partial g}{\partial \boldsymbol{\tau}} = -\frac{\partial g}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \boldsymbol{\theta}}$ when $\boldsymbol{\tau} = \boldsymbol{\theta}$.

Similarly, sensitivities of series system probability of failure estimates can also be obtained but it has to be noted that while computing the sensitivity associated with a given sample point, the sensitivity measures of the limit state surface on which the projected sample point lies, have to be used. As a direct consequence of this, discontinuities in sensitivities can arise when the projected sample points switch from one limit state surface to another with design changes. This mainly happens at the intersection of two limit state surfaces at which the failure domain itself has derivative discontinuities. Another thing to remember while computing sensitivities for series system is that the overall weighted simulation density function $h_{\mathbf{V}}$, in general, is dependent on v_n corresponding to each projected sample point for each simulation variable in the Axis-Orthogonal Simulation. This is because a projected sample point corresponding to a simulation variable has to be transformed to the coordinate systems of other simulation variables for computation of $h_{\mathbf{V}}$. Hence while deriving the formula for sensitivities of series system, chain rule for differentiation has to be used and the dependence of $h_{\mathbf{V}}$ on v_n has to be accounted for. The expression for $\frac{\partial \hat{P}^{(j)}}{\partial \boldsymbol{\tau}}$ for axis orthogonal simulation, where all $\mathbf{V}^{(j)}$ are chosen as standard

normal random variables, is given by the following equation:

$$\frac{\partial \hat{P}^{(j)}}{\partial \tau} = \frac{1}{N_j} \sum_{i=1}^{N_j} \left[\frac{f_{\mathbf{V}^*}(\mathbf{v}_i^{(j)})}{h_{\mathbf{V}}(\mathbf{v}_i^{(j)})} \left\{ -\Phi(v_n^{(j)}(\mathbf{v}_i^{(j)})) \right. \right. \\ \left. \left. - \frac{\left(1 - \Phi(v_n^{(j)}(\mathbf{v}_i^{(j)}))\right)}{h_{\mathbf{V}}(\mathbf{v}_i^{(j)})} \sum_{k=1}^M \left(w_k \frac{\partial h_{\mathbf{V}^{(k)}}}{\partial \mathbf{v}_i^{(k)}} \frac{\partial \mathbf{v}_i^{(k)}}{\partial v_n^{(j)}} \right) \right\} \frac{\partial v_n^{(j)}}{\partial \tau} \right] \quad (6.22)$$

where $\frac{\partial \mathbf{v}_i^{(k)}}{\partial v_n^{(j)}}$ is the last column of the transformation matrix $\mathbf{G}^{(k)} \left(\mathbf{G}^{(j)} \right)^{-1}$, and $\frac{\partial v_n^{(j)}}{\partial \tau}$ is given by the Equation 6.21 with the properties of the limit state on which the simulation lies on. If the simulation point is on the intersection of two or more limit state surfaces, the sensitivities are discontinuous.

6.2 MCS-Based RBO Methodology

As mentioned in the earlier section, directional and axis orthogonal simulation are the MCS techniques that are considered best suited for RBO. In this section, the basic steps for performing reliability analysis using axis orthogonal simulation for both component and series system reliability estimation will be presented. First, the number of simulations, N_1, \dots, N_M , the sampling densities, $h_{\mathbf{V}^{(1)}}, \dots, h_{\mathbf{V}^{(M)}}$ (taken standard normal in this work) and seeds are fixed. The seeds are fixed to avoid discontinuities in probability of failure estimates. For series system, the weights, w_1, \dots, w_M are picked based on the FORM-based reliability estimates. After this the reliability analysis is performed that basically consists of the following steps 1-5:

1. Find the exact MPPs for all M limit states.
2. Find coordinate systems for the simulation variables for all limit states. Compute transformations $\mathbf{G}^{(i)}$ transforming \mathbf{u} to $\mathbf{v}^{(i)}$ using Gram Schmidt Orthogonalization.
3. For $i=1$ to M perform steps (i), (ii) and (iii)
 - (i) For i^{th} limit state or hard constraint generate N_i samples according to $\mathbf{V}^{(i)}$,
 - (ii) For each sample, compute the roots for finding the integration limits in v_n . Sin-

gle root is assumed throughout in this work here. For component reliability, evaluation of i^{th} limit state alone is sufficient. For series system reliability, evaluation of all hard constraints are required,

(iii) For component reliability, compute the probability of failure P_i . For series system reliability, compute the contributing probability of failure, $\hat{P}^{(i)}$.

4. For series system, find the weighted sum of the contributing probabilities of failure to get the overall probability of failure, P .

5. Compute the sensitivities of probability or probabilities of failure, if needed. For series systems, it is important to know on what limit state surface each projected sample point is on. At each projected sample point the sensitivities of the corresponding limit state with respect to \mathbf{x} and $\boldsymbol{\eta}$ are required.

The initial design for MCS-based RBO can be an approximate solution to the RBO problem, for example based on FORM. For series systems, series system FORM unimodal estimates can be used to get an approximate solution to the RBO problem. The series system unimodal upper bound for the probability of failure estimate is just the sum of the component probabilities of failure. In the test problems, the starting design for the MCS-based RBO were all solutions obtained from FORM-based RBO.

6.3 Summary

The indicator-based MCS techniques using importance sampling techniques do not give smooth estimates of the probabilities of failure. The analytic sensitivity of the probability of failure estimate with respect to a limit state parameter, for this class of techniques, requires the computation of a surface integral which is not easy to compute. In the conditional expectation-based MCS techniques the simulation points are all constrained to be on the boundary of the failure domain and hence there is no shifting of sample points across the limit state surface with change in curvature of the limit state surface, unlike

the indicator-based MCS techniques. Due to this reason, the estimates of probability of failure are expected to be smoother for this class of MCS techniques than the indicator-based MCS techniques. Analytic sensitivities of the probability of failure obtained from the conditional expectation-based MCS depend on the properties of the limit state surface on which the simulation point lies on and can be easily obtained.

There are two types of conditional expectation-based MCS; directional simulation and axis orthogonal simulation. In directional simulation, the sampling is done in polar coordinates. In axis orthogonal simulation all the sampling is done in the coordinate axis obtained by linear or affine transformation of the standard normal space. Series system probability of failure estimation is performed using a stratified sampling approach. In this approach a weighted sum of component simulation densities is chosen as the overall sampling density.

For the RBO, the seeds for the random number generators and the number of sampling points for each hard constraint is fixed at the same values throughout for continuity purposes. While performing RBO using MCS techniques, exact MPP searches will be performed for all hard constraints during each iteration. After this the probability of failure and its sensitivities will be estimated for each component or the series system using axis orthogonal simulation.

CHAPTER 7

IMPLEMENTATION STUDIES

The application problems used for implementation studies can be divided into two categories, ones used for the RBO applications, and others used for test studies of the bounded MPP search algorithms. The main goals of the implementation studies for the RBO applications are to demonstrate the working of the RBO-CSSO and RBO-CSSO2 approaches and their computational efficiency when applied to multidisciplinary design problems. The main goals of the implementation studies for the bounded MPP search algorithms are to investigate the robustness of the algorithm for cases with feasible and infeasible MPP searches, and also to compare the computational costs with those required by other algorithms.

The implementation studies are presented in four major parts. In the first part, results of the RBO-CSSO approach for the test problems are presented and computational comparisons are made for selected problems. In the second part, results of the bounded MPP search algorithms, that were developed in Chapter 5, are presented for various test problems and are compared with the results obtained from other MPP search algorithms. In the third part, results for the RBO-CSSO2 approach are presented and computational comparisons are made for selected problems. In the fourth part, results for the MCS-based RBO are presented. All the programming was done using Matlab¹ versions 6.1 and 6.5.

¹Copyright 1984-2002, The Mathworks Inc.

7.1 RBO Application Problems

The applications problems in this section consist of an analytic test problem, that was used for illustration purposes, and variations of a control augmented structure design problem[94] and a high performance, low-cost structure design problem[16]. For the latter two problems, consistent units were used for all the values of variables and parameters (dimensions, material properties and performance quantities) shown in the the following sections and they are representation of physical quantities. All the problems are multidisciplinary in nature. The Sequential Quadratic Programming (SQP) from the Matlab Optimization toolbox was used as the constrained optimization routine for all the test cases of the RBO application problems.

7.1.1 Analytic Test Problem

The first problem is a simple analytic problem, used to illustrate the basic characteristics of the RBO-CSSO approach and allow for graphical presentation of the key results. There are 2 design variables, $\{d_1, d_2\}$, and one parameter, p , in the problem. There are two random variables, $\{X_1, X_2\}$. This problem is a coupled problem and has 2 CAs. Different versions of this problem are used for RBO implementation studies. There are two or three constraints depending on the version of the problem. One constraint is a soft constraint, g^D and the other constraints are hard, g_1^R and g_2^R . g_1^R and g^D are kept same for all the different versions. In one version, g_2^R is absent and in other versions, this constraint is present. The details will be provided later during the chapter, when the different versions are presented. The part of the formulation common among all the versions are given as

follows:

$$\text{Merit Function: } f = d_1^2 + y_1(\mathbf{d}, p) + e^{-y_2(\mathbf{d}, p)}$$

$$\text{Constraints: } g_1^R(\mathbf{X}, \eta) = Y_1(\mathbf{X}, \eta)/2 - 1$$

$$g^D(\mathbf{d}, p) = y_2(\mathbf{d}, p)/2 - 1$$

$$\text{where } d_1 = \mu_{X_1}, p = \mu_{X_2} = 0 \text{ and } d_2 = \eta$$

$$CA_1: \quad Y_1(\mathbf{X}, \eta) = X_1^2 + \eta - 0.2Y_2(\mathbf{X}, \eta)$$

$$y_1(\mathbf{d}, p) = d_1^2 + d_2 - 0.2y_2(\mathbf{d}, p)$$

$$CA_2: \quad Y_2(\mathbf{X}, \eta) = \sqrt{Y_1(\mathbf{X}, \eta)} + X_1 + X_2$$

$$y_2(\mathbf{d}, p) = \sqrt{y_1(\mathbf{d}, p)} + d_1 + p$$

$$\mathbf{d} \text{ bounds: } 0 \leq d_1 \leq 10, 0 \leq d_2 \leq 10$$

7.1.2 Control Augmented Structures Problem

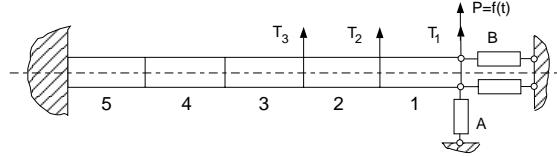


Figure 7.1. Control Augmented Structure

Figure 7.1 shows the control augmented structure as proposed by Sobieszczanski-Sobieski et. al.[94]. There are two disciplines or Contributing Analyses (CAs) in this problem which are the structures subsystem and the controls subsystem. The structure is a 5 element cantilever beam, numbered 1-5 from the free-end to the fixed-end, as shown in the Figure 7.1. Each element is of equal length, but the breadth and height are variable. Three static loads T_1 , T_2 and T_3 are applied to the first three elements. The beam is also acted on by a time varying force P , which is a ramp function. Controllers A and B are

designed as an optimal Linear Quadratic Regulator to control the lateral and rotational displacements of the free end of the beam, respectively.

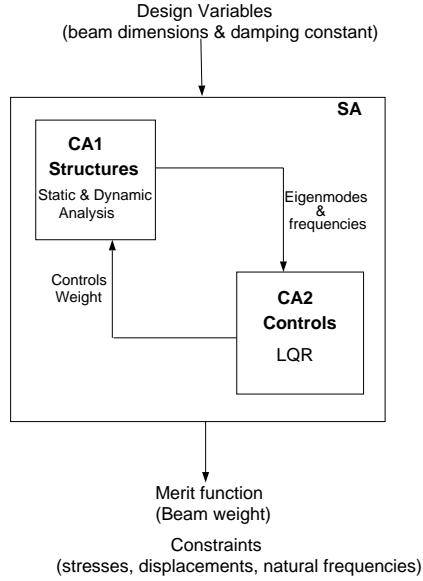


Figure 7.2. Coupling in Control Augmented Structures

The analysis is coupled since the mass of the controllers, m_{con} , is assumed to be proportional to the control effort, and it is required for the mass matrix of the structures. One requires the eigenfrequencies and eigenvectors of the structure in the modal analysis for designing the controller, as illustrated in Figure 7.2. The damping matrix is taken to be proportional to the stiffness matrix by a factor c for the dynamic analysis of the structure. This damping parameter is also a design variable. The constraints arise due to constraints on static stresses, static and dynamic displacements and the natural frequencies. The main objective is to minimize the total weight of the beam and the controllers. For the sensitivity analysis, analytic sensitivities for the static displacements, eigenfrequencies, eigenvectors[95], dynamic displacements and controller's weight[96] are used. Finite differencing is used for sensitivities of static stresses. Please refer to Appendix C for further details on this application problem. This problem was selected because this problem illustrates the multidisciplinary and coupled nature of controlled structure

problems, where the controllers and structures are designed simultaneously.

The design variables and design variable bounds used for different versions of this test problem are given below as,

$$\begin{aligned}\mathbf{d} &= [b_1, b_2, b_3, b_4, b_5, h_1, h_2, h_3, h_4, h_5, c]^T \\ 3 &\leq b_{1-5} \leq 36 \\ 3 &\leq h_{1-5} \leq 36 \\ 0.01 &\leq c \leq 0.06\end{aligned}$$

where b_i and h_i are the breadth and height of the i^{th} element respectively, c is the damping matrix to stiffness matrix ratio (scalar).

The problem as formulated by Sobiesczanski-Sobieski et. al.[94] had only 8 constraints. For a reliability analysis all the important failure modes had to be identified. For this problem, 32 limit states or failure modes were identified:

$$\begin{aligned}g_i &= 1 - (dl_i/50)^2, \quad i = 1..5 \\ g_{i+5} &= 1 - (dr_i/0.2)^2, \quad i = 1..5 \\ g_{11} &= (\omega_1/1.0) - 1 \\ g_{12} &= (\omega_2/1.25) - 1 \\ g_{2i+11} &= 1 - (\sigma_i^r/\sigma_a), \quad i = 1..5 \\ g_{2i+12} &= 1 - (\sigma_i^l/\sigma_a), \quad i = 1..5 \\ g_{i+22} &= 1 - (ddl_i/50)^2, \quad i = 1..5 \\ g_{i+27} &= 1 - (ddr_i/0.15)^2, \quad i = 1..5\end{aligned}$$

where dl_i and dr_i are the static lateral and rotational displacements of i^{th} element respectively, ω_1 and ω_2 are the first and second natural frequencies respectively, σ_i^r and σ_i^l are the maximum static stresses at the right and left ends of i^{th} element, σ_a is the ultimate

static stress, and ddl_i and ddr_i are the dynamic lateral and rotational displacements of i^{th} element respectively.

FORM-based traditional RBO, RBO-CSSO, RBO-CSSO2 and MCS-based traditional RBO were performed for this application problem. This application problem was also used to study the performance of MPP search algorithms.

7.1.3 High Performance, Low-Cost Structure Problem

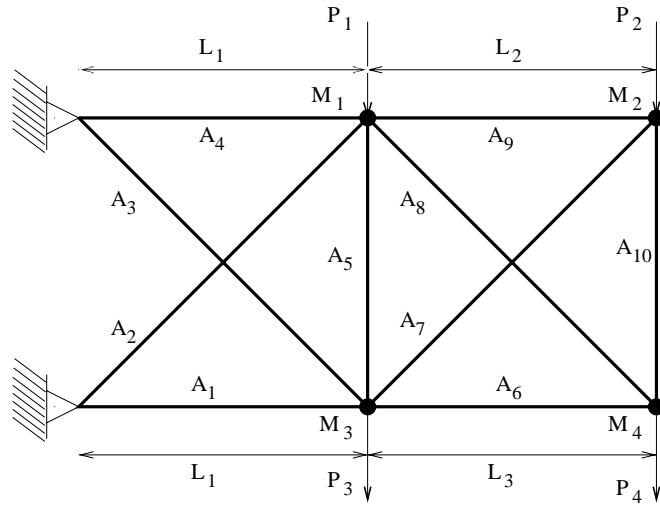


Figure 7.3. High Performance, Low-Cost Structure Problem

Figure 7.3 shows the structure to be designed for high performance (maximum load carrying) and low-cost (minimum weight). The structure consists of 10 elements and is a statically indeterminate 2-D plane truss structure. It is subjected to 4 loads, P_{1-4} and the structure carries 4 payload masses, M_{1-4} . In a traditional design optimization given by Equations 7.1-7.9, it is required to find values of $\{L_{1-3}, M_{1-4}, A_{1-10}\}$ such that a objective function given in Equation 7.1 is minimized subject to constraints on minimum total payload (Equation 7.2), minimum total load (Equation 7.3), minimum first natural frequency (Equation 7.4) and maximum axial stresses in each element (Equation 7.5). The objective function is formulated for multiple objectives, minimum weight, W , (low

cost), maximum $\sum M_i$ and maximum $\sum P_i$ (high performance i.e maximum load carrying).

$$\min \quad 0.003 W + (10^6 / \sum P_i) + (3.75 \times 10^6 / \sum M_i) \quad (7.1)$$

$$\text{s.t.} \quad g_1 = 1 - (5000 / \sum M_i) \geq 0 \quad (7.2)$$

$$g_2 = 1 - (100000 / \sum P_i) \geq 0 \quad (7.3)$$

$$g_3 = 1 - (2.0 / \omega_1) \geq 0 \quad (7.4)$$

$$g_{3+i} = 1 - (\sigma_i / \sigma_{yield,i})^2 \geq 0 \quad i = 1, \dots, 10 \quad (7.5)$$

$$\text{Bounds:} \quad 100 \leq L_1 \leq 1000 \quad (7.6)$$

$$30 \leq L_{2-3} \leq 2000 \quad (7.7)$$

$$250 \leq M_{1-4} \leq 500000 \quad (7.8)$$

$$1 \leq A_{1-10} \leq 20 \quad (7.9)$$

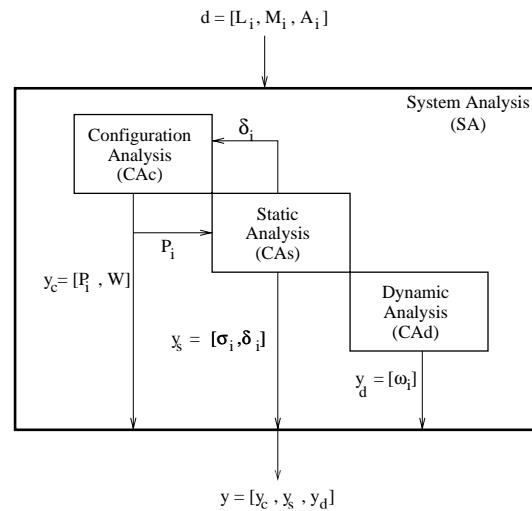


Figure 7.4. System Analysis for the High Performance, Low-Cost Structure Problem

There are three disciplines or CAs in this problem as shown in Figure 7.4. The first CA, CA_c , performs configurational analysis where it determines the loads P_{1-4} , and the

structures weight, W . The loads are given by the following equation:

$$P_i = \sum_{k=1}^3 a_k^i (L_k/60)^{b_k^i} + \sum_{j=1}^3 c_j^i (M_j/1250)^{d_j^i} - e_i \delta y_{J_i} \quad i = 1, \dots, 4 \quad (7.10)$$

where δy_{J_i} is the vertical nodal displacement at node J_i , which corresponds to the node at which P_i is acting. CA_s , performs static structural analysis for determining the stresses, σ_j , where $j=1..10$, horizontal and vertical nodal displacements, i.e. δx_k and δy_k respectively, where $k=1..6$. CA_s requires the loads P_{1-4} from CA_c . The coefficients a , b , c and d are as given in Tables 7.1 and 7.2. Two versions of the test problem were used for implementation studies. In one version, e was set as zero, hence making the problem uncoupled. In the other version, e was set as a non-zero vector, hence making the problem coupled. The third CA, CA_d , performs dynamic analysis for determining the natural frequencies, ω_i . This contributing analysis is completely independent of the other contributing analysis. Wujek[97] provides additional technical details on this problem. This problem was selected because it is similar to a class of aeroelastic design problems that exhibit coupling between the aerodynamics (loads) and structures (displacements) disciplines.

TABLE 7.1. LOAD COEFFICIENTS

$a_1^1=2500$	$a_2^1=2000$	$a_3^1=2000$	$b_1^1=-4$	$b_2^1=-3.7$	$b_3^1=-3.7$
$a_1^2=1700$	$a_2^2=1900$	$a_3^2=1500$	$b_1^2=-3.5$	$b_2^2=-3.8$	$b_3^2=-3$
$a_1^3=2500$	$a_2^3=2000$	$a_3^3=2000$	$b_1^3=-4$	$b_2^3=-3.7$	$b_3^3=-3.7$
$a_1^4=1700$	$a_2^4=1500$	$a_3^4=1900$	$b_1^4=-3.5$	$b_2^4=-3$	$b_3^4=-3.8$

FORM-based traditional RBO, RBO-CSSO and RBO-CSSO2 approaches were implemented for this problem.

TABLE 7.2. LOAD COEFFICIENTS (CONTINUED)

$c_1^1=50$	$c_2^1=37$	$c_3^1=35$	$c_4^1=37$	$d_1^1=4$	$d_2^1=2.9$	$d_3^1=2.9$	$d_4^1=2.9$
$c_1^2=25$	$c_2^2=27$	$c_3^2=25$	$c_4^2=27$	$d_1^2=2.7$	$d_2^2=2.7$	$d_3^2=2.7$	$d_4^2=2.7$
$c_1^3=35$	$c_2^3=37$	$c_3^3=50$	$c_4^3=37$	$d_1^3=2.9$	$d_2^3=2.9$	$d_3^3=4$	$d_4^3=2.9$
$c_1^4=25$	$c_2^4=27$	$c_3^4=25$	$c_4^4=27$	$d_1^4=2.7$	$d_2^4=2.7$	$d_3^4=2.7$	$d_4^4=2.7$

7.2 RBO-CSSO Implementation Studies

In this section, three application problems are presented for which the RBO-CSSO methodology was applied. The main aim of these implementation studies was to illustrate the working of the RBO-CSSO methodology using an analytic problem, and also show the computational benefits obtained using this approach over a traditional RBO methodology for a control augmented structures design problem and a high performance, low-cost structure design problem. In the present implementations of the RBO-CSSO approach, fixed move-limits were used for SSOs and CP. For all the test problems used for the RBO applications, the $\|\mathbf{d}^{k+1} - \mathbf{d}^k\|$ was required to be less than 0.001. For all the test problems the CP was performed using quadratic approximations of merit function, reliability constraints and soft constraints obtained from iteration histories of SSOs as presented in section 5.2.3. The quadratic approximations were constructed using a pseudo-inverse technique using only the functional values. The approximations were built about the current design, \mathbf{d}^k , such that the approximations matched the exact functional and gradient values of merit function and constraints of the RBO formulation at \mathbf{d}^k .

7.2.1 Analytic Test Problem

In this version of the analytic test problem, only one hard constraint was considered as given in previous section. $X_1 = N(d_1, 0.05)$ and $X_2 = N(0, 0.5)$ were set for this problem. The FORM based reliability index was required to be greater than 3. All the MPP searches

were performed using the iHL-RF algorithm as presented in Appendix B. In the RBO-CSSO approach, both the SSOs were allowed to change only their local design variables. So, SSO_1 was allowed to change both the design variables, but SSO_2 was allowed to change only d_1 . Thus d_1 was the shared design variable in this problem. In SSO_1 , g^{rbo} was estimated from reliability analysis using linear approximations of the limit states, and in SSO_2 , it was estimated from a linear approximation of reliability constraints. The move-limits were chosen to be 10 percent of the total design space, centered on the design variables' current settings.

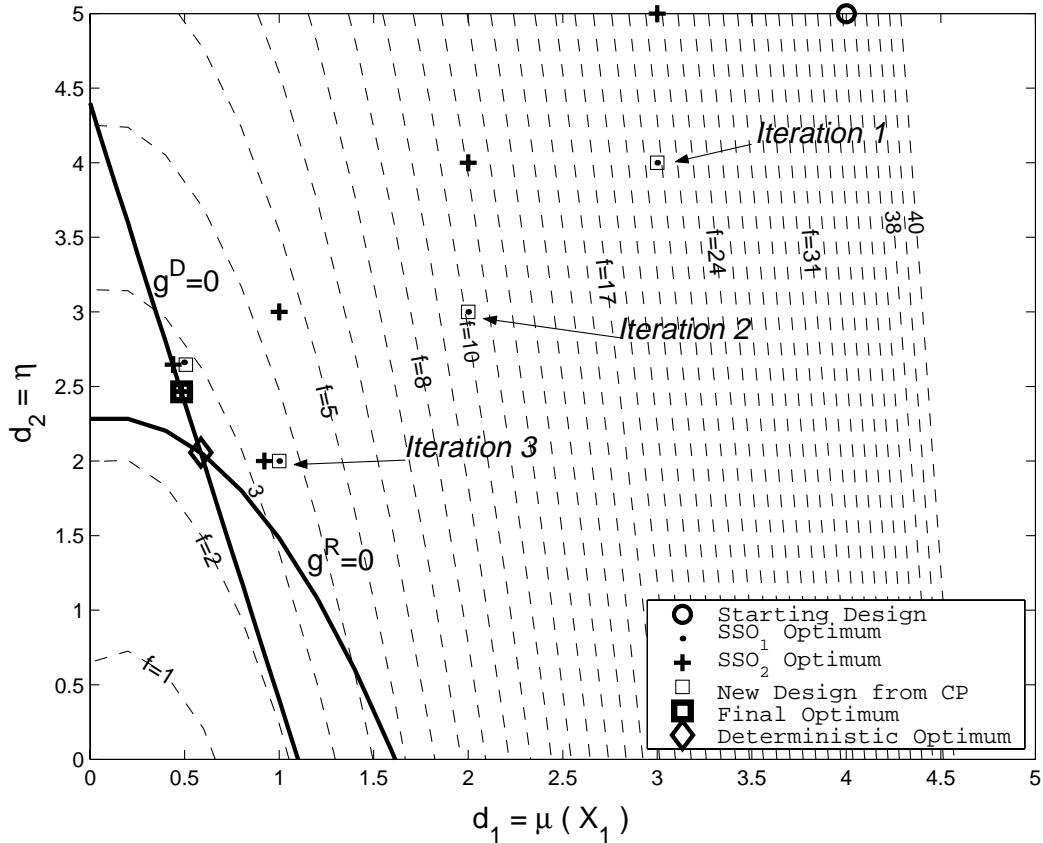


Figure 7.5. Iteration History in \mathbf{d} Space for the RBO-CSSO Implementation of the Analytic Test Problem

The initial design was selected as $\{4, 5\}$. The same final design, $\{0.484, 2.464\}$, was obtained using traditional RBO and the RBO-CSSO approaches for this test problem.

Figure 7.5 shows the iteration history in the design variable space which includes final design histories of SSOs and the CP. The traditional optimum design, $\{0.586, 2.057\}$, lies on the constraint surfaces. It can be seen that the final design from RBO is on the deterministic constraint boundary but slightly displaced from the hard constraint boundary in order to satisfy the reliability constraints. The Figure also shows the optima from the SSOs and CP at each overall iteration of the RBO-CSSO. The Figure shows that for the first three iterations the subspace optima and the design from the CP are all constrained by the move-limits alone. It can be seen that SSO_1 determines the new design in iterations 1-3 but for subsequent iterations the new designs are determined by the coordination between both SSOs. This problem is used only for demonstration purposes and hence no computational comparisons between the traditional RBO and the RBO-CSSO approaches are presented.

7.2.2 Controls Augmented Structure Problem

Six uncertain variables, ρ , E , σ_a , T_1 , T_2 and T_3 , were considered for this problem. $\rho = LN(0.1, 0.01)$, $E = LN(10500, 1050)$, $\sigma_a = LN(3000, 300)$, $T_{1,3} = N(1000, 100)$ and $T_2 = N(5000, 500)$ are set for this test case. The initial designs for both the traditional RBO and RBO-CSSO approaches were chosen to be the optimum from the traditional design optimization, that does not take uncertainties into consideration. The constraint g_1 was the only active constraint at this design. Hence g_1 was considered as a hard constraint for the RBO problem. The constraints g_i , $i = 6, 16, 18, 20, 22, 28$, were picked because their values at the traditional design optimum were close to zero. An initial reliability analysis revealed these constraints had low reliabilities and hence were important constraints. The constraint g_{11} , a dynamic constraint requiring the evaluation of SA, was included as a hard constraint for the sole purpose of increasing the computational costs for this problem. The static stress constraint, g_{14} , was included as a hard constraint because it

was expected from the importance of other static stress constraints that this constraint could be important as the design changed during the RBO. Hence, the constraints g_i , $i = 1, 6, 11, 14, 16, 18, 20, 22, 28$, were considered as hard constraints for the RBO. The remaining constraints were included as soft constraints.

TABLE 7.3. DESIGNS AND COMPUTATIONAL COMPARISON FOR THE CONTROL AUGMENTED STRUCTURE

	Initial Design	Trad. RBO	RBO-CSSO
b_{1-5}	3.00	3.00	3.00
h_1	3.703	3.762	3.762
h_2	7.040	8.589	8.589
h_3	9.807	12.125	12.125
h_4	11.998	14.844	14.844
h_5	13.840	17.139	17.139
c	0.06	0.06	0.06
f	1493.87	1754.38	1754.38
M_{con}		60.62	60.62
CA ₁ calls	-	1424	1061
CA ₂ calls	-	1293	977
SA calls	-	48	35

The RBO had 9 reliability constraints based on reliability indices from FORM and had 23 deterministic constraints. The allowable total system failure, P_{all} was set at 0.001, which was equally distributed among these 9 failure modes, based on unimodal FORM bounds. So $\beta_{reqd} = -\Phi^{-1}(P_{all}/9)$ that equals 3.692, for this problem. The starting point for the MPP search, during the traditional RBO, was set as the MPP obtained in the previous MPP search. In the RBO-CSSO approach the starting point for the MPP searches in the rSA was chosen to be the MPP obtained in the final MPP search in SSO_1 , for the previous iteration. The MPP searches were performed using the iHL-RF algorithm. For the RBO-CSSO approach, SSO_1 controlled the first 10 design variables while the

SSO_2 controlled the last design variable, c . It should be noted that c has no effect on the static constraints in this problem. Hence in this problem, SSO_2 always gave infeasible results when started with a design which was infeasible due to the violation of the static constraints. The reliability constraints in the SSOs were obtained by performing MPP searches on the linear approximations to the limit state. The move-limits were chosen to be 15 percent of the total design space, about the design variables' settings. Only the iteration histories corresponding to designs within the prescribed move-limits were taken for the construction of quadratic approximations for the CP. The quadratic approximations were constructed using the function values alone and constructed about the current design.

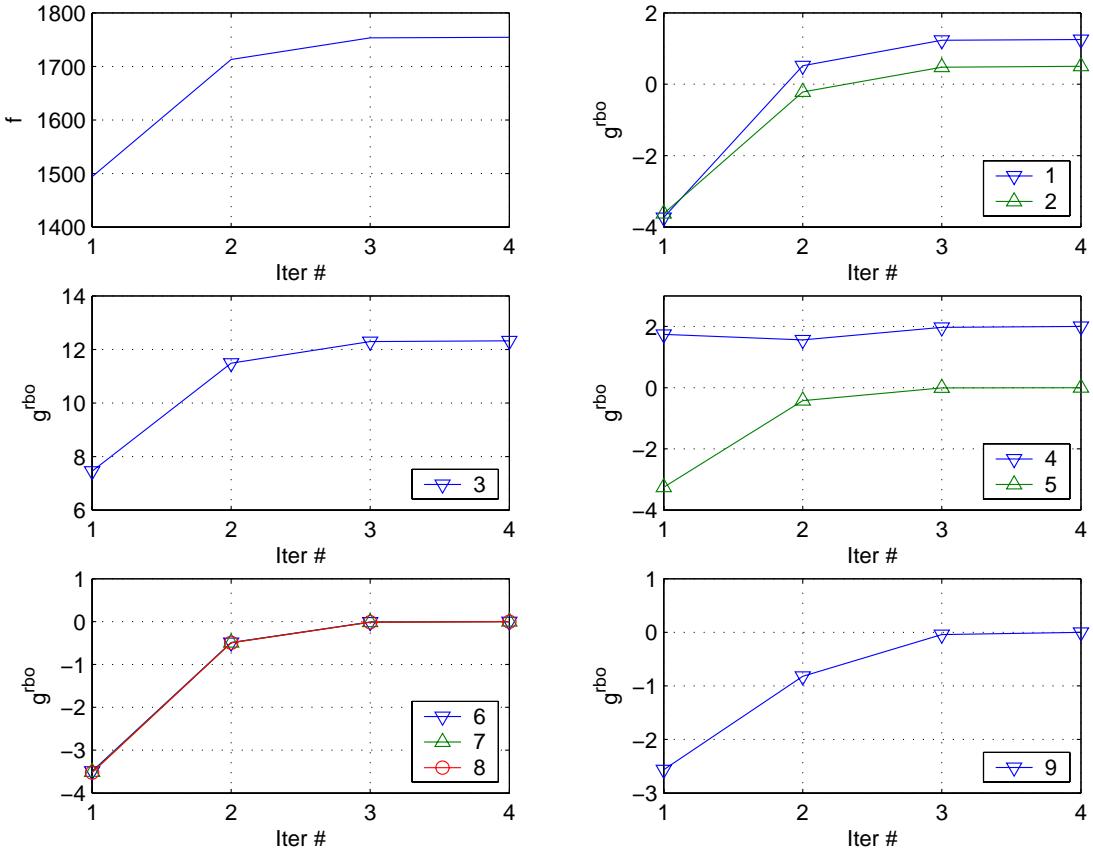


Figure 7.6. RBO-CSSO Convergence Plots for the Control Augmented Structure Problem

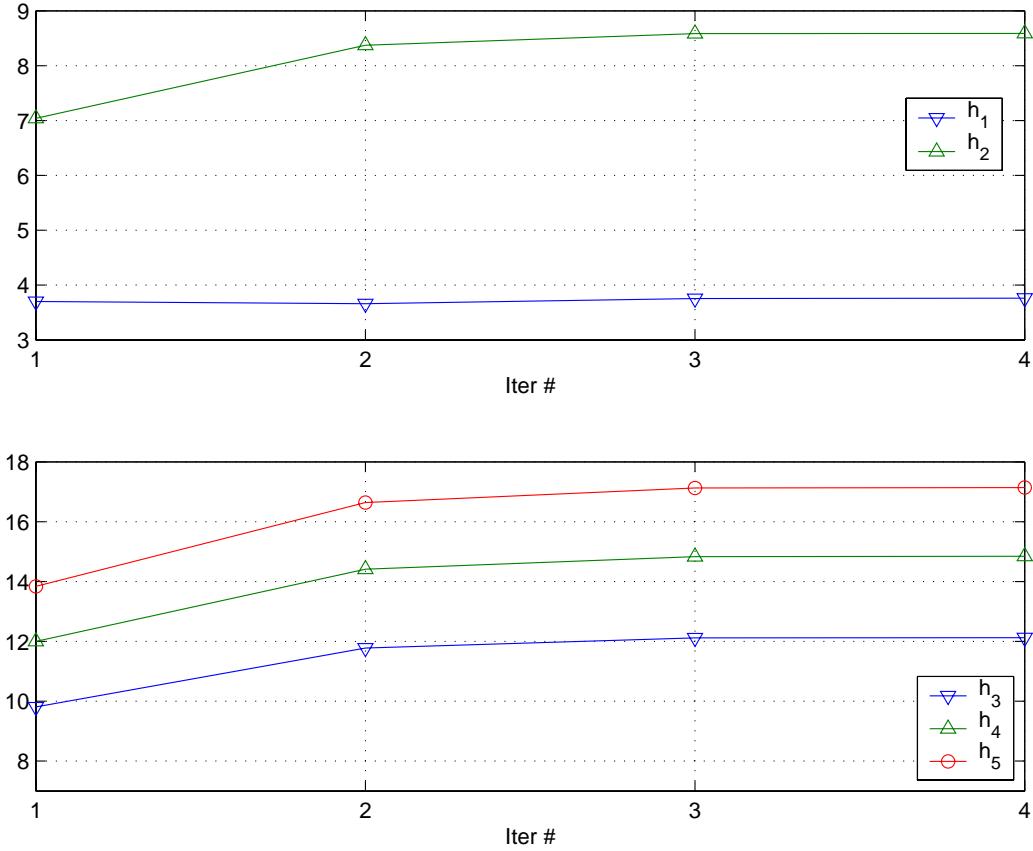


Figure 7.7. RBO-CSSO Convergence Plots for the Control Augmented Structure Problem (continued)

The starting and final designs for both the methods are given in Table 7.3. The traditional RBO and RBO-CSSO yielded the same final design. The design variables, b_{1-5} and c remained at their lower bounds throughout. Table 7.3 also gives the computational comparisons associated with the two methods. In terms of the number of CA calls, it can be seen that the RBO-CSSO approach, for this implementation, was approximately 25% more efficient than the traditional RBO approach. The convergence histories of the RBO-CSSO approach are shown in Figures 7.6 and 7.7. Figure 7.6 show the convergence histories of the exact merit function, reliability constraints and deterministic constraints obtained from rSA. It can be seen that at the starting design, the reliability constraints 1, 2, 5, 6, 7, 8 and 9 that correspond to $g_1, g_6, g_{16}, g_{18}, g_{20}, g_{22}$ and g_{28} respectively,

are violated, but at the final design only the latter 5 of these reliability constraints are active. It can also be seen that the first 2 iterations are infeasible with respect to all these constraints. But it has to be noted that the designs obtained from the CP were feasible with respect to the approximate models during all the iterations. The new design in each iteration of the RBO-CSSO was mainly driven by SSO_1 alone due to the infeasible results from SSO_2 , as was mentioned earlier. For this implementation, the move-limits for h_{1-5} were not active during any of the iterations and hence the results would remain the same if the move-limits were increased.

TABLE 7.4. DESIGN VARIABLE MOVE-LIMIT STUDY FOR CONTROL AUGMENTED STRUCTURE PROBLEM

	1%	5%	10%	15%
CA ₁ calls	3234	1063	1061	1061
CA ₂ calls	2973	978	977	977
SA calls	105	35	35	35

A move-limit study was performed for this problem. The RBO-CSSO approach was performed using 4 different move-limit choices, including the one for which the results were presented earlier. Same final design was obtained for all the 4 cases. Table 7.4 shows the computational costs associated with the 4 cases. It can be seen that the reduction of the move-limits from 15% to 10% does not affect the CA calls at all. For 5% move-limits, the computational costs is similar to the case with 15% move-limits, even-though the few components of b_{1-5} after the iteration 1 were not at their lower bounds. For 1% move-limits, the computational costs increase significantly. The first 2 iterations for this problem was actually infeasible. In the next few iterations feasibility was obtained by increases in b_{1-5} and h_{1-5} . But finally, b_{1-5} were pushed to their lower bounds due to steady increases in h_{1-5} .

7.2.3 High Performance, Low-Cost Structure

An uncoupled version of the high performance, low-cost structure problem was chosen, i.e. e was set to zero for this particular study. 30 uncertain or random variables were considered for this problem. The random variables were Young's Modulus, E_i , density, ρ_i , and yield stress, σ_{yield_i} , for each element i.e $i = 1 - 10$. $E_i = \text{LN}(10000, 1000)$, $\rho_i = \text{LN}(0.1, 0.01)$ and $\sigma_{yield_i} = \text{LN}(14000, 1400)$ were set through out for all the RBO applications. The constraints $g_3, g_4, g_5, g_6, g_7, g_{10}$ and g_{11} were selected as hard constraints and the rest were selected as soft constraints. For the reliability constraints, it was required that the FORM based reliability indices be greater than 3.

The starting points for the design variables for the traditional RBO and RBO-CSSO approaches were selected as the optimum from the traditional design optimization problem. The starting point for the MPP search for both the approaches was set as the MPP from the previous exact MPP search. The reliability analysis was performed using Sequential Quadratic Programming (SQP) method in the standard normal i.e. \mathbf{U} space, using the Matlab 6.5 Optimization Toolbox.

There were 3 Subspace Optimizations (SSOs) in the RBO-CSSO approach corresponding to each CA. Unlike the previous examples, each CA required all the design variables, without considering the effect due to interdependence between CAs. So any choice of design variables for each SSO could be made. In this implementation study, design variables for each SSO were set in accordance with those set by Wujek and Renaud[16]. So, SSO_1 was allowed to change $\{L_{1-3}, M_{1-4}\}$, SSO_2 was allowed to change A_{1-10} and SSO_3 was allowed to change M_{1-4} . So there were shared design variables between SSO_1 and SSO_3 . The design variable move-limits for the RBO-CSSO approach were chosen as $\Delta L_{1-3} = \pm 1$, $\Delta M_{1,3} = \pm 200$, $\Delta M_{2,4} = \pm 100$ and $\Delta A_{1-10} = \pm 0.5$.

Unlike the previous examples, move-limits based on a percentage of the entire design space given by the design variable bounds were not selected for this problem. A move-

limit based on a choice of, say a $\pm 1\%$ of the entire design space would allow excessive variations in lengths and masses compared to the variations in the areas. In this problem, the merit function is very non-linear with respect to lengths and masses. A few trial runs were performed with move-limits based on percentages of the total design space and in most of these trial runs, a cycling phenomenon was observed where the designs obtained after the CPs never converged but oscillated between two designs every alternate iteration. This was due to excessively large move-limits in the lengths and the masses. Hence the move-limits in lengths and masses had to be reduced while a relatively larger move-limits for the areas had to be set.

The reliability constraints in the SSOs were obtained using linear approximations of reliability constraints obtained from the initial reliability analysis. This was a particularly effective choice for this application problem over the use of linear approximation of limit state surfaces, because the SSOs were infeasible during the first three iterations and since Matlab SQP was used, which does not strictly enforce the inequalities represented by the variable bounds during all iterations, the optimizer goes to extreme values of design variables where reliability analysis for the limit state approximations failed. Dummy large objective function values were used in case negative values of design variables were obtained during the optimization in the SSOs, to avoid the failure of the automated analysis. The CP was performed in the same fashion as was done in the control augmented structure test problem.

The initial design and final designs obtained from the traditional RBO and RBO-CSSO are given in Table 7.5. The initial design was selected as the design obtained from the traditional design optimization. At this design $g_4, g_5, g_6, g_7, g_{10}$ and g_{11} were active. L_1 and $A_{5,6,9,10}$ remained at their lower bounds in both the final designs. The final designs from the traditional RBO and RBO-CSSO are slightly different. It can be seen that the values of $L_{2,3}, M_{1-4}$ and $A_{1-4,7,8}$ were larger in the final design obtained from the

traditional RBO than that was obtained in the final design from the RBO-CSSO approach. Consequently W obtained from the traditional RBO, $\sum_i P_i$ and $\sum_i M_i$ were larger than the result obtained from the RBO-CSSO. But the overall merit obtained from the traditional RBO was higher than that obtained from RBO-CSSO. Hence the design obtained from RBO-CSSO is a preferred design.

TABLE 7.5. DESIGNS AND COMPUTATIONAL COMPARISON FOR UNCOUPLED HIGH PERFORMANCE LOW-COST STRUCTURE PROBLEM

	Initial Design	Trad. RBO	RBO-CSSO
L_1	100.00	100.00	100.000
L_2	68.61	75.50	70.62
L_3	68.61	75.50	70.62
M_1	8136.1	8134.6	7536.2
M_2	11388.1	11387.4	11088.1
M_3	8135.9	8134.4	7536.0
M_4	11389.4	11388.7	11089.4
A_1	5.06	6.96	5.89
A_2	14.55	19.62	16.46
A_3	14.55	19.62	16.46
A_4	5.06	6.96	5.89
$A_{5,6}$	1.00	1.00	1.00
A_7	3.11	4.22	3.93
A_8	3.11	4.22	3.93
$A_{9,10}$	1.00	1.00	1.00
f	10.32	11.846	11.75
W	1501.64	2002.91	1722.36
$\sum_i P_i$	353415	350594	289424
$\sum_i M_i$	39049.5	39045.1	37249.7
CA _c calls	-	9629	7096
CA _s calls	-	11009	8427
CA _d calls	-	1936	1450

The RBO-CSSO approach, for this implementation study, required 23-26% fewer CA

calls than the traditional RBO.

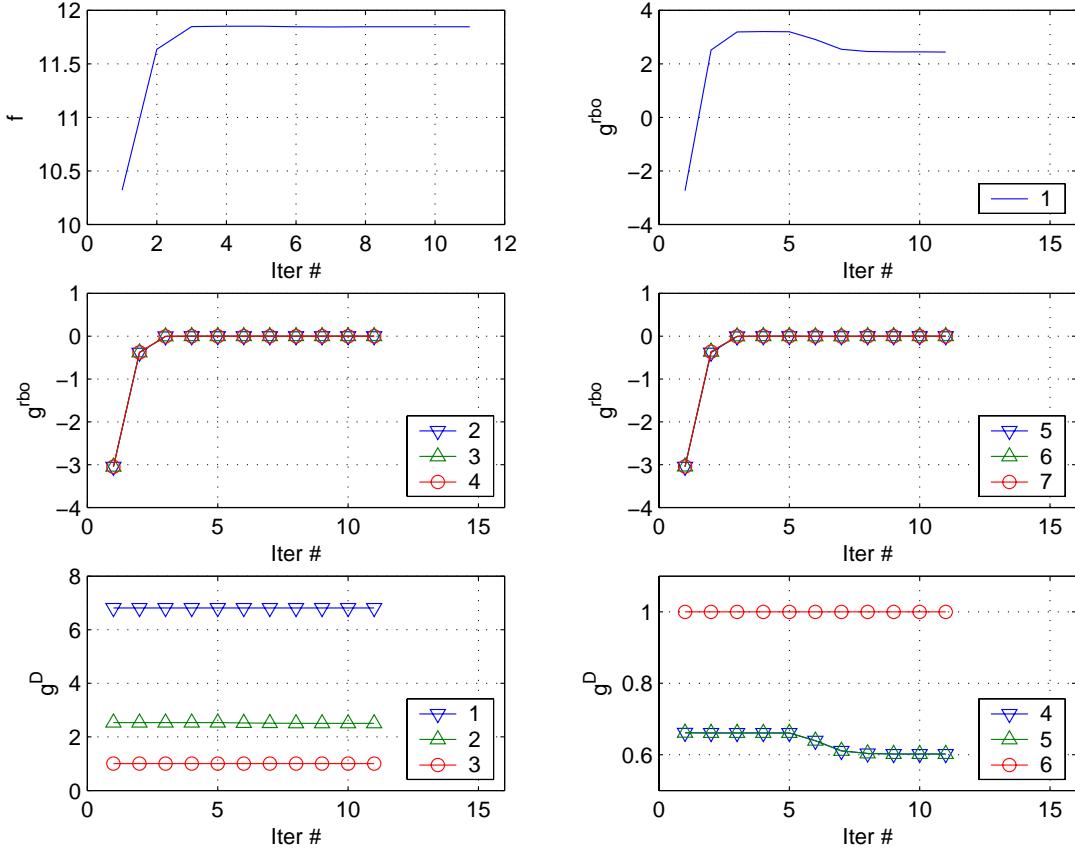


Figure 7.8. Traditional RBO Convergence Plots for High Performance, Low-Cost Structure Problem

Figures 7.8 and 7.9 show the convergence plots of merit function, constraints and the designs for the traditional RBO. At the final design, g_{2-7}^{rbo} were active. The masses and L_1 almost remained constant during all the iterations. There was an increase in L_2 and L_3 after the iteration 5, after minimal change in the first 5 iterations. This also resulted in a drop of g_1^{rbo} corresponding to g_3 after iteration 5. This behavior was indicative of the non-linearity present in the problem. By performing the traditional RBO using various randomly chosen starting points in the vicinity of the optimum from traditional design optimization, it was observed that different combinations of lengths, payloads and areas could yield similar final objective function in the range 11.75-11.85, and it was found

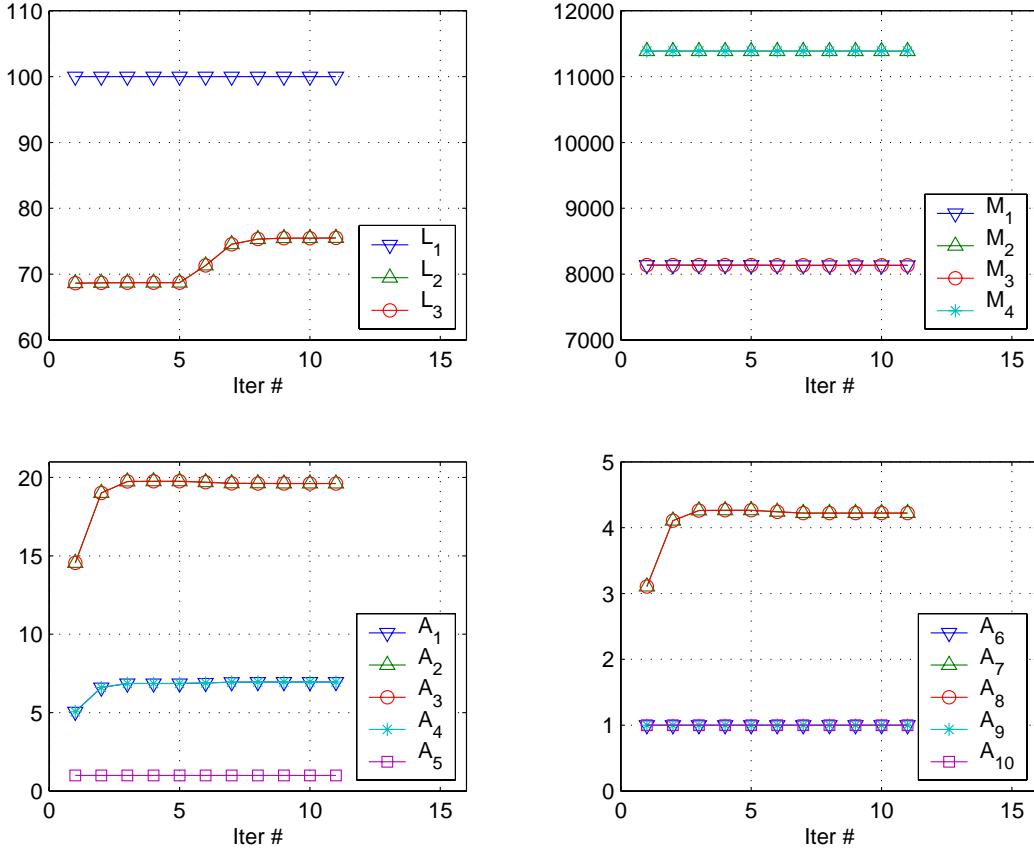


Figure 7.9. Traditional RBO Convergence Plots for High Performance, Low-Cost Structure Problem (Continued)

that other local minima also existed for this problem with higher final objective function. Similar observations were reported for the traditional design optimization by Wujek and Renaud[16] for this problem. This was mainly due to the nonlinearity of the merit function and this was an indication of the flatness of merit function in some design regions. The choice of lenient convergence tolerances could actually give premature convergence for this problem. This was particularly observed when the traditional design optimization was performed.

Figures 7.10 and 7.11 show the convergence plots for the RBO-CSSO. The first 3 iterations are actually infeasible in this problem. As in the traditional RBO, g_{2-7}^{rbo} were also active at the final design. Masses M_{1-4} decreased to their lower move-limits in the

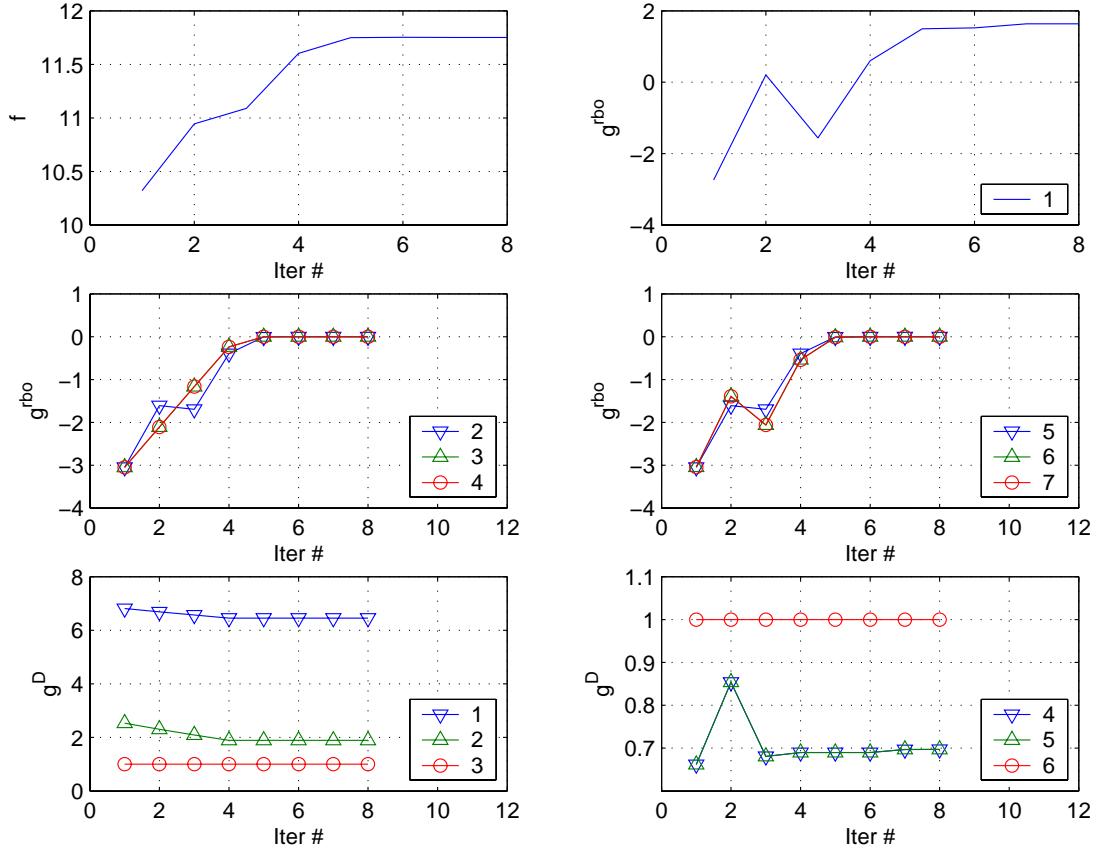


Figure 7.10. RBO-CSSO Convergence Plots for High Performance, Low-Cost Structure Problem

first 4 iterations and remained constant after that. The lengths L_2 and L_3 increased to their upper move-limits in the first 4 iterations, then reduced to their lower move-limits after iteration 6 and remained constant after that. L_1 remained constant throughout. Areas A_2 and A_3 constantly increased to their upper move-limits till iteration 5. Areas A_5 , A_6 , A_9 and A_{10} all increased to their upper move-limits after iteration 1 but they reduced to their lower bounds and remained at their lower bounds after iteration 2. A_1 , A_4 , A_7 and A_8 increased to their upper move-limits after iteration 1 and reduced to their lower move-limits after iteration 2 and then increased again to their upper move-limits after iteration 3 and remained constant after iteration 5.

A study on the effect of selecting different move-limits for \mathbf{d} was performed for this

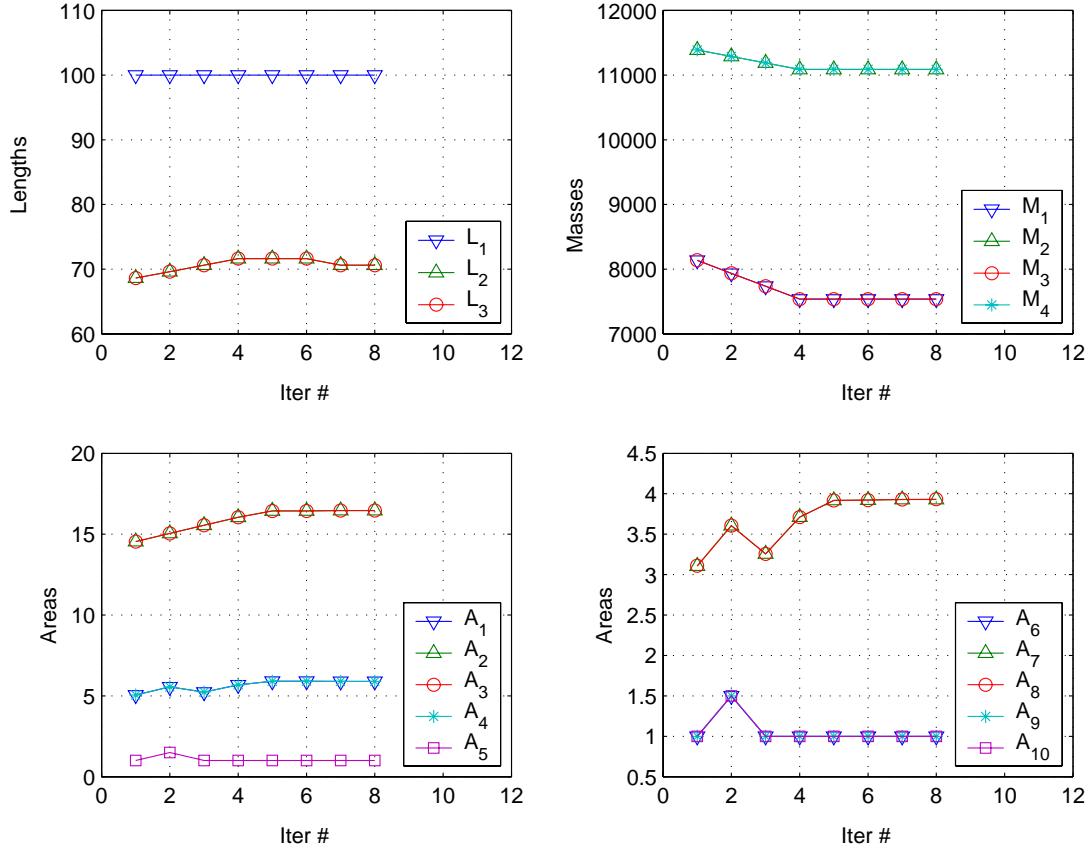


Figure 7.11. RBO-CSSO Convergence Plots for High Performance, Low-Cost Structure Problem (Continued)

problem. The move-limits for certain design variables were varied keeping the rest fixed at the original choice that was given earlier. 5 cases were considered where $\Delta M_{1,3} = 100$, $\Delta M_{2,4} = 200$, $\Delta L = 0.5$, $\Delta A_{1-5} = 1.0$ and $\Delta A_{6-10} = 1.0$. The results are presented in Table 7.6. It can be seen that different designs but with similar objective function are obtained again confirming the non-linearity of this problem and existence of multiple local minima. The number of CA calls required for these 5 cases, suggest that move-limits for $M_{2,4}$ and A_{1-5} could have been increased, and the move-limits for L_{1-3} could have been kept lower. One has to note that the designs obtained for the cases $\Delta L = 0.5$ and $\Delta A_{6-10} = 1.0$ are very close to the final design obtained for the original move-limits (see Table 7.5).

TABLE 7.6. DESIGN VARIABLE MOVE-LIMIT STUDY FOR HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM

	$\Delta M_{1,3} = 100$	$\Delta M_{2,4} = 200$	$\Delta L = 0.5$	$\Delta A_{1-5} = 1.0$	$\Delta A_{6-10} = 1.0$
L_1	100.00	100.00	100.000	100.000	100.000
L_2	72.61	71.18	69.62	70.63	70.62
L_3	72.61	71.17	69.62	70.63	70.62
M_1	7729.9	7536.1	7536.2	7736.1	7536.2
M_2	10986	10853	11088	11188	11088
M_3	7729.7	7536	7536.0	7735.9	7536.0
M_4	10987	10854	11089	11189	11089
A_1	6.09	5.75	5.88	6.21	5.89
A_2	17.07	16.06	16.48	17.48	16.46
A_3	17.07	16.06	16.48	17.48	16.46
A_4	6.09	5.75	5.88	6.21	5.90
$A_{5,6}$	1.00	1.00	1.00	1.00	1.00
A_7	3.90	3.79	3.94	4.03	3.93
A_8	3.90	3.79	3.94	4.03	3.93
$A_{9,10}$	1.00	1.00	1.00	1.00	1.00
f	11.743	11.755	11.751	11.751	11.751
CA _c calls	8294	6218	6403	4874	7282
CA _s calls	9998	7350	7608	5933	8653
CA _d calls	1655	1275	1312	932	1449

7.2.4 Summary

The general characteristics of the RBO-CSSO approach were illustrated using the analytic test problem. Approximately 25% reduction in CA calls was observed for the control structures problem and high performance, low-cost structure problem. SSO₂ was ineffective in the CSSO implementation of the RBO for the control augmented structure problem. The control augmented structure problem was fairly benign and lenient move-limits could be set and were effective. The high performance, low-cost structure problem was fairly nonlinear and choice of different move-limits in the RBO-CSSO approach

resulted in different designs but with approximately same values of merit function.

7.3 MPP Search Implementation Studies

Thirty eight test problems were selected to test the Trust Region-based Equality Constrained Optimization (TRECO) and Elastic Sequential Quadratic Programming (ESQP) algorithms for the bounded MPP search. The test problems include problems with limit state functions that are explicit functions of random variables and limit state functions of the control augmented structure problem. The limit state functions that are explicit functions of random variables consist of examples that appeared in literature and 2-D analytic examples that were developed to simulate some extreme scenarios for testing the MPP algorithms. The main aim of these implementation studies was to test the TRECO and ESQP algorithms for cases where the bounded MPP searches were feasible and infeasible with respect to the equality constraints, compare the computational costs with those of other MPP search algorithms and also to illustrate the mode of failure of some of the traditional MPP search algorithms like iHL-RF and SQP. Two class of algorithms were considered. One class consisted of bounded MPP search algorithms, that consisted of the TRECO and ESQP. The other class of algorithms consisted of the traditional MPP search algorithms, that included SQP, HL-RF and iHL-RF algorithms. The algorithms were implemented in **X**-space and **U**-space. The algorithms that were implemented in **X**-space consisted of TRECO, ESQP and SQP. The algorithms that were implemented in **U**-space consisted of TRECO, ESQP, SQP and iHL-RF.

The convergence criteria common for all the algorithms required that at a given iteration, the absolute value of the limit state function of a hard constraint was less than some prescribed tolerance and either the maximum absolute value of the step size or the absolute value of the total reduction in merit obtained were lower than prescribed tolerances. This was essentially the same convergence criteria for the feasible case in TRECO

algorithm, that was presented in the chapter 5. This can be written as follows:

$$(|g| < \varepsilon_1) \quad \text{AND} \quad [(\max |\mathbf{s}| < 2\varepsilon_2) \quad \text{OR} \quad (|\nabla_z \psi \mathbf{s}| < 2\varepsilon_3)] \quad (7.11)$$

where \mathbf{s} is the trial step generated by the particular algorithm and $\psi = \mathbf{u}^T \mathbf{u}$. \mathbf{z} is a general notation for \mathbf{x} or \mathbf{u} . It has to be noted that \mathbf{s} and \mathbf{z} include 2 additional elastic variables for the ESQP algorithms. Tolerances $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-6}$ and $\varepsilon_3 = 10^{-6}$ were commonly set for all the algorithms. $\varepsilon_4 = 10^{-3}$ and $\varepsilon_5 = 10^{-6}$ were additionally set for the TRECO algorithms, that are required for convergence check for the infeasible case. The maximum allowable number of functional evaluations was set at 200 for all the algorithms. The bounds in \mathbf{u} are required for the TRECO algorithm alone among all the \mathbf{U} -space algorithms. This was set as ± 6 for each component of \mathbf{u} . The bounds for all the \mathbf{X} space algorithms are required and they were computed by the inverse transformation of ± 6 bounds that were set for the \mathbf{u} . The inverse transformation was computed numerically by a root solving approach implemented using the FZERO subroutine available in Matlab. The inverse transformation is designed for the Rosenblatt transformation as presented in Appendix A. First, \mathbf{x}_1 is determined, since it depends only on \mathbf{u}_1 . Then, \mathbf{x}_2 is found by fixing \mathbf{x}_1 . So, \mathbf{x}_k is determined by the previous $k - 1$ components of \mathbf{x} . Additionally, $v = 0.7$ was set for both the TRECO algorithms and the initial trust region was chosen 0.1 times the entire random variable space defined by the bounds. The value of $\rho_{constant}$, used for the penalty parameter update, was set at 2 for the TRECO algorithms. For the ESQP algorithm, the penalty parameter, ξ was set at a value of 10000.

7.3.1 Test Problems and Results

Analytic Problem 1: This problem was taken from Liu and Kiureghian[48]. It consists of a quadratic limit state function given by the following equation:

$$\begin{aligned} g(\mathbf{X}) = & 1.1 - 0.001155X_1X_2 + 0.00157X_2^2 + 0.00117X_1^2 \\ & + 0.0135X_2X_3 - 0.0705X_2 - 0.00534X_1 - 0.0149X_1X_3 \\ & - 0.0611X_2X_4 + 0.0717X_1X_4 - 0.226X_3 + 0.0333X_3^2 \\ & - 0.558X_3X_4 + 0.998X_4 - 1.339X_4^2 \end{aligned}$$

There are four statistically independent random variables in this problem, X_1, X_2, X_3 and X_4 . X_1 has a type-II largest value distribution with mean 10 and standard deviation 5. $X_2=N(25,0.8)$, $X_3=N(5,0.2)$ and $X_4=LN(0.0625,0.0625)$ are set. This problem was mainly chosen to illustrate the failure of HL-RF algorithm by Liu and Kiureghian. The MPP for this problem is $\mathbf{x}^*=\{14.91, 25.07, 0.8595, 0.04606\}$. The FORM reliability index for this problem is 1.33. All the algorithm converged for this example and the computational costs are listed in Table 7.7. The TRECO algorithms performed comparably to U-space SQP algorithm in terms of number of g and $\nabla_x g$ computation required. X-space ESQP was the most expensive of all the algorithms.

Analytic Problem 2: This problem was also taken from Liu and Kiureghian[48]. It consists of a noisy linear limit state function where the noise was artificially simulated using a high frequency sinusoidal term. The limit state function is given by the following equation:

$$g(\mathbf{X}) = X_1 + 2X_2 + 2X_3 + X_4 - 5X_5 - 5X_6 + 0.001 \sum_{i=1}^6 \sin(100X_i)$$

There are six statistically independent random variables in this problem. X_1, X_2, X_3 and X_4 are set as $LN(120,12)$. $X_5=LN(50,15)$ and $X_6=LN(40,12)$ are set. This problem

was chosen by Liu and Kiureghian to illustrate the performance of various MPP search algorithms when the limit state function is noisy. It can be seen from Table 7.7 that X-space ESQP and iHL-RF algorithms did not converge within 200 iterations. The TRECO algorithms were the most efficient in terms of number of function and gradient computations needed. It was expected that there were potentially more than one MPP for this problem because of the low-amplitude sinusoidal term. In the present implementations the MPPs were obtained in the range $\{117.17, 115.10, 115.10, 117.17, 83.12, 54.97\}$ - $\{117.41, 115.33, 115.33, 117.41, 84.17, 55.84\}$ by the algorithms that converged for this problem. The FORM reliability index for this problem is about 2.35.

TABLE 7.7. FUNCTION AND GRADIENT CALLS FOR ALL ALGORITHMS FOR
THE ANALYTIC PROBLEMS

No.	X-space			U-space			
	TRECO	ESQP	SQP	TRECO	ESQP	SQP	iHL-RF
1	8, 8	138,35	29, 22	8, 8	16, 8	8, 8	42, 21
2	24, 24	Failed ^e	62, 34	22, 22	48, 20	70, 38	Failed ^e
3	28, 28	Failed ^e	Failed ^t	138, 138	Failed ^e	Failed ^t	Failed ^t
4	3, 3	3, 3	5, 5	6, 6	6, 6	6, 6	6, 5
5	10, 10	14, 8 ^w	Failed ^t	7, 7	148, 34	Failed ^t	Failed ^t
6	10, 10	21, 15	22, 18	20, 20	7, 7	7, 7	9, 9

^t Failure of T or T^{-1} .

^m Failure of evaluation of g .

^e Exceeded maximum allowable functional evaluations.

^w Wrong Result.

Analytic Problem 3: This problem consists of a quadratic limit state function given by the following equation.

$$g(\mathbf{X}) = -X_1 + (X_2 - 3)^2 + 20$$

There are two independent random variables. Both X_1 and X_2 are set as $U(0,10)$. For

this example, g is always greater than zero in the region where the probability density function is greater than zero. Hence no MPP exists for this problem. g attains the least value of 10 at the point $\mathbf{U} = \{6, 0.524\}$ and $\mathbf{X} = \{10, 3\}$. It can be seen from Table 7.7 that only the TRECO algorithms succeeded in giving the solution for this example. It has to be noted that X_1 is close to 10 in the range $U_1 = 3 - 6$. The inverse transformation from U_1 to X_1 is extremely nonlinear in this range and consequently g is very nonlinear and badly scaled in this range near the solution. The U-space TRECO algorithm succeeded in converging to this problem but was very expensive and converged to $\mathbf{U} = \{5.559, 0.524\}$. This is mainly because the normal step in the TRECO is based on only a linear approximation of the limit state function and hence can perform very badly for badly scaled and non-linear limit states. The X-space TRECO, on the other hand, converged to $\mathbf{U} = \{5.999, 0.524\}$. The ESQP algorithms exceeded the allowable 200 function evaluations. The X-space SQP failed because the transformation from \mathbf{X} to \mathbf{U} failed. The U-space SQP and iHL-RF algorithms failed because the numerical inverse transformation failed. This happens when large steps are generated by the algorithms that fall out of the bounds set on the components of \mathbf{x} for the numerical inverse transformation.

Analytic Problem 4: This problem was taken from Engelund and Rackwitz[52]. It consists of a linear limit state function given by the following equation.

$$g(\mathbf{X}) = - \sum_{i=1}^{20} X_i + 36.72$$

There are twenty independent random variables. All the random variables are set as exponential random variables with means of 1. A more general version of this problem was selected by Engelund and Rackwitz for study of various Monte Carlo simulation techniques. It can be seen from Table 7.7 that all the algorithms give the solution for this problem. The MPP for this problem is $X_i = 1.836$, for $i = 1..20$. The computational costs were quite similar for all the X-space and U-space methods.

Analytic Problem 5: This problem consists of a quadratic limit state function given by the following equation.

$$g(\mathbf{X}) = -(X_1 - 0.5)^2 - (X_2 - 20.25)$$

There are two independent random variables. Both X_1 and X_2 are set as $U(0,10)$. The FORM reliability index for this problem is -0.148. The MPP for this problem is $\{4.41, 4.92\}$. It can be seen from Table 7.7 that the TRECO algorithms and the U-space ESQP algorithm converged to the MPP for this problem. The X-space SQP failed because of the failure of the transformation from X to U. The U-space SQP and iHL-RF algorithms failed because of the failure of inverse transformation from U to X. This failure occurred despite the fact that the actual problem is solvable, i.e. an MPP exists. There might be an inclination to think that if the standard MPP search algorithms fail due to failure of random variable transformation then the MPP formulation is infeasible, which is shown to be not true in this test problem. The X-space ESQP algorithm converged to a solution that did not correspond to the MPP. The algorithm prematurely converged because the change in x variables were less than the required tolerance. But the first order optimality was not completely satisfied according to the output of the algorithm. This happened because the algorithm was stuck in a region where ψ was very flat in x space. The U-space ESQP algorithm took many iterations to converge to the actual solution. The TRECO algorithms were extremely efficient for this problem.

Analytic Problem 6: This problem was taken from Wang and Kodiyalam[69]. The limit state function is a constraint on the first fundamental frequency of a tapered cantilever beam with rectangular cross-section and of length L. The dimensions at the fixed end are h_0 and w_0 and dimensions at the free end are h_L and w_L . It is required that the fundamental

frequency be greater than 220Hz. The limit state function can be written as follows

$$g = \frac{1}{480L^2} \sqrt{\frac{30386.4E\pi PE_{fac}}{\rho KE_{fac}}} - 220$$

where E is the Elastic Modulus and equals 17×10^6 and ρ is the density of the material and equals 0.161. PE_{fac} and KE_{fac} represent the potential energy and kinetic energy factor, respectively. They are given by the following relations

$$\begin{aligned} PE_{fac} &= -240h_L^3w_0 + 240h_0^3w_L - 480h_0^3w_0 + 2h_L^2h_0w_0\pi^4 \\ &+ h_L^3w_0\pi^4 - 80h_L^3w_L\pi^2 + 3h_Lh_0^2w_0\pi^4 - 20h_0^3w_L\pi^2 \\ &+ 60h_L^2h_0w_L\pi^2 - 60h_Lh_0^2w_0\pi^2 + 3h_L^2h_0w_L\pi^4 + 4h_L^3w_L\pi^4 \\ &+ 20h_L^3w_0\pi^2 + 80h_0^3w_0\pi^2 + h_0^3w_L\pi^4 + 2h_Lh_0^2w_L\pi^4 \\ &+ 4h_0^3w_0\pi^4 + 480h_L^3w_L - 720h_L^2h_0w_L + 720h_Lh_0^2w_0 \\ KE_{fac} &= 2\pi^3h_Lw_L + 32h_Lw_0\pi + 2h_0\pi^3w_0 \\ &- 60h_0w_0\pi - 16h_Lw_L\pi^2 - 128h_Lw_0 \\ &+ \pi^3h_Lw_0 + 32h_0w_L\pi - 4h_Lw_L\pi + h_0\pi^3w_L \\ &- 128h_0w_L + 128h_0w_0 + 128h_Lw_L \end{aligned}$$

The random variables for this problem were h_0 , w_0 , w_L and L . h_0 has a Type-I largest value distribution with mean 0.468 and standard deviation 0.01. $w_0=\text{LN}(2.0,0.2)$ and $w_L=\text{LN}(2.0,0.2)$ are set. L has a Type-I largest value distribution with mean 0.468 and standard deviation 0.01. $h_L = 0.2$ is chosen. The FORM based reliability index for this problem is 3.325. The MPP is $\{0.460, 1.739, 2.277, 8.373\}$. All the algorithms converged to this solution for this problem. The computational effort needed by U-space TRECO was higher than the other U-space algorithms, whereas the computational effort needed by X-space TRECO was the least among the X-space algorithms.

Control Augmented Structures Problem: For these test cases, all the 32 limit state func-

tions for the control augmented structures problem were selected. There are six independent random variables for these problems, the ultimate stress, σ_a , density, ρ , Young's Modulus, E and the static loads, T_1 , T_2 and T_3 . The first three random variables are set as lognormally distributed variables with coefficient of variation 0.1. The static loads are set as normally distributed variables with coefficient of variation 0.1. All the random variables were scaled by their means. Hence the means of the random variables were 1 and the standard deviation is 0.1. The reliability analysis is performed at the final design obtained from the RBO application problem presented in the previous section, which is listed in Table 7.3. The MPPs occur within the prescribed bounds for 9 of the limit states and are given in Table 7.8. For the remaining 23 limit states, the MPPs do not occur within the prescribed bounds and the best constraining solutions are given in Table 7.9.

TABLE 7.8. MOST PROBABLE POINTS FOR THE CONTROL AUGMENTED STRUCTURE LIMIT STATES

LS No.	β	\mathbf{x}^*
1	4.943	{0.995, 0.995, 0.652, 1.106, 1.231, 1.023}
6	4.196	{0.995, 0.995, 0.694, 1.150, 1.153, 1.012}
14	5.691	{0.631, 0.995, 0.995, 1.341, 1.000, 1.000}
16	3.692	{0.732, 0.995, 0.995, 1.076, 1.190, 1.000}
18	3.692	{0.731, 0.995, 0.995, 1.058, 1.192, 1.019}
20	3.692	{0.731, 0.995, 0.995, 1.051, 1.193, 1.026}
21	6.134	{0.589, 0.995, 0.995, 1.081, 1.304, 1.041}
22	3.692	{0.731, 0.995, 0.995, 1.048, 1.193, 1.029}
28	3.692	{0.995, 0.991, 0.688, 1.000, 1.000, 1.000}

For some of the limit states listed in Table 7.9, the U-space SQP and iHL-RF algorithms gave the exact MPPs, which are indicated in Table 7.10 with superscript *as*. This happened due to the absence of variable bounds in these algorithms unlike TRECO, X-

TABLE 7.9. BOUNDED SOLUTIONS FOR THE CONTROL AUGMENTED STRUCTURE LIMIT STATES

LS No.	\mathbf{x}_c
2-5	$\{0.995, 0.995, 0.547^l, 1.600^u, 1.600^u, 1.600^u\}$
7	$\{0.995, 0.995, 0.547^l, 1.132, 1.356, 1.028\}$
8-10	$\{0.995, 0.995, 0.547^l, 1.600^u, 1.600^u, 1.600^u\}$
11,12	$\{0.995, 1.810^u, 0.547^l, 1.000, 1.000, 1.000\}$
13	$\{0.547^l, 0.995, 0.995, 1.600^u, 1.000, 1.000\}$
15	$\{0.547^l, 0.995, 0.995, 1.600^u, 1.600^u, 1.000\}$
17	$\{0.547^l, 0.995, 0.995, 1.600^u, 1.600^u, 1.600^u\}$
19	$\{0.547^l, 0.995, 0.995, 1.111, 1.370, 1.037\}$
23-28, 29-32	$\{0.995, 0.547^l, 0.547^l, 1.000, 1.000, 1.000\}$

^lLower Bound

^uUpper Bound

space SQP and ESQP algorithms. For limit state 19, the solutions obtained from TRECO, ESQP and X-space SQP correspond to a feasible but bounded solution, which is listed in Table 7.10 with superscript *fb*. This solution is not the MPP and is listed in Table 7.9. The X-space SQP algorithm failed for certain limit states listed in Table 7.9, either due to failure of transformation from U to X or because the problem was shown to be infeasible or because the SQP algorithm exceeded the maximum allowable function evaluations.

The U-space SQP and iHL-RF algorithms failed due to failure of inverse transformation from U to X and failure of evaluation of g . This mainly happened because trial steps were generated that correspond to extreme values of random variables at which either the inverse transformation or the analysis tool that computes g failed. The failure of evaluation of g was mainly due to the failure of the controls analysis tool for this problem. For limit state 11, U-space SQP algorithm failed because the failure of the controls analysis during the evaluation of g . But the iHL-RF algorithm worked successfully for this problem, which indicates that SQP, even-though computationally more efficient, could

encounter problems.

It can be seen that from Table 7.10 that the ESQP algorithms in general take more function and gradient evaluations than the TRECO algorithms. For limit state 30, the X-space ESQP algorithm exceeded allowable function evaluations. There were also cases where the ESQP algorithms for some limit states listed in Table 7.9 gave solutions that were different from the bounded solutions listed. For these cases, the solutions were either infeasible and unbounded or infeasible and partially bounded, as listed in Table 7.10 with superscripts *ub* and *pb* respectively. This happened because the value of ξ was not large enough to yield sufficient reduction in the infeasibility of the equality constraint. It can also been seen that the computational costs associated with the TRECO algorithms were comparable to those of the standard MPP search methods such as SQP methods and iHL-RF, for feasible cases and in general inexpensive for the infeasible cases for this application problem.

7.3.2 Summary

7 different MPP search algorithms were applied to 38 test problems, consisting of various limit state functions and different types of random variables. The test problems consisted of both feasible and infeasible MPP search problems. The standard MPP searches, like U-space SQP and iHL-RF typically fail for infeasible problems mainly due to failure of numerical inverse transformation or due to model failure i.e. failure of the evaluation of g or they exceed the allowable functional evaluations, with the exception of analytic problem 5 and limit state 11 of the control augmented structures. The ESQP algorithms in general were quite expensive and exceeded the allowable number of function evaluations for some problems. They also yielded infeasible unbounded solutions and partially bounded solutions for certain limit states because the penalty parameter was not sufficiently large. For analytic test problem 5, the X-space ESQP converges to a wrong result.

TABLE 7.10. FUNCTION AND GRADIENTS CALLS FOR ALL ALGORITHMS FOR
CONTROL AUGMENTED STRUCTURE PROBLEM

No.	X-space			U-space			
	TRECO	ESQP	SQP	TRECO	ESQP	SQP	iHL-RF
1	9, 9	30, 27	26, 23	8, 8	50, 19	14, 13	7, 5
2	10, 10	16, 13	Failed ⁱ	8, 8	4, 4	17, 15 ^{as}	9, 6 ^{as}
3	10, 10	15, 12	Failed ⁱ	8, 8	18, 6	11, 8 ^{as}	13, 9 ^{as}
4	10, 10	16, 10	Failed ⁱ	9, 9	9, 7	37, 18 ^{as}	16, 10 ^{as}
5	10, 10	57, 27 ^{ub}	Failed ⁱ	9, 9	6, 6 ^{ub}	Failed ^t	Failed ^t
6	8, 8	27, 22	26, 24	8, 8	10, 9	10, 9	7, 5
7	9, 9 ^{fb}	21, 19 ^{fb}	24, 16 ^{fb}	12, 12 ^{fb}	26, 15 ^{fb}	16, 10 ^{as}	8, 6 ^{as}
8	10, 10	17, 11	Failed ⁱ	8, 8	5, 5	14, 12 ^{as}	10, 7 ^{as}
9	10, 10	14, 13	Failed ⁱ	8, 8	4, 4	23, 16 ^{as}	12, 8 ^{as}
10	10, 10	16, 11	Failed ^t	9, 9	11, 7	Failed ^t	17, 10 ^{as}
11	7, 7	10, 10	Failed ^e	6, 6	7, 7	Failed ^m	8, 7 ^{as}
12	6, 6	12, 11	Failed ^t	5, 5	4, 4	13, 13 ^{as}	9, 8 ^{as}
13	6, 6	53, 23	Failed ⁱ	5, 5	12, 11	Failed ^t	28, 19 ^{as}
14	8, 8	50, 25	13, 12	8, 8	14, 9	9, 9	8, 7
15	7, 7	15, 15 ^{pb}	Failed ⁱ	7, 7	6, 6 ^{pb}	14, 12 ^{as}	14, 11 ^{as}
16	7, 7	21, 20	22, 21	7, 7	12, 8	8, 8	6, 5
17	8, 8	18, 11 ^{pb}	Failed ⁱ	7, 7	4, 4 ^{pb}	9, 8 ^{as}	8, 6 ^{as}
18	7, 7	26, 22	26, 23	7, 7	8, 8	8, 8	6, 5
19	7, 7 ^{fb}	21, 15 ^{fb}	22, 19 ^{fb}	7, 7 ^{fb}	12, 12 ^{fb}	8, 8 ^{as}	9, 7 ^{as}
20	9, 9	26, 26	25, 24	7, 7	12, 10	9, 9	6, 5
21	8, 8	22, 16	26, 19	18, 18	123, 24	10, 10	8, 6
22	22, 22	27, 25	27, 25	7, 7	19, 12	9, 9	6, 5
23	8, 8	12, 11 ^{pb}	Failed ^t	7, 7	50, 9 ^{pb}	Failed ^t	Failed ^m
24	8, 8	42, 13 ^{pb}	Failed ^t	7, 7	58, 11 ^{pb}	Failed ^m	Failed ^m
25	8, 8	22, 12 ^{pb}	Failed ^t	7, 7	63, 9 ^{pb}	Failed ^m	Failed ^m
26	8, 8	22, 15 ^{ub}	Failed ^t	7, 7	24, 10 ^{ub}	Failed ^t	Failed ^t
27	8, 8	22, 14 ^{ub}	Failed ^t	5, 5 ^{pb}	4, 4 ^{ub}	Failed ^t	Failed ^t
28	7, 7	20, 18	17, 13	7, 7	12, 9	9, 9	6, 5
29	8, 8	15, 13 ^{pb}	Failed ^e	7, 7	39, 11 ^{pb}	Failed ^t	Failed ^m
30	8, 8	Failed ^e	Failed ^t	7, 7	13, 8 ^{pb}	Failed ^t	Failed ^m
31	8, 8	72, 19 ^{pb}	Failed ^t	7, 7	102, 9 ^{pb}	Failed ^m	Failed ^m
32	8, 8	20, 13 ^{pb}	Failed ^t	7, 7	49, 10 ^{pb}	Failed ^t	Failed ^t

^t Failure of T or T^{-1} .

^m Failure of evaluation of g .

ⁱ Terminated because of no feasible solution.

^e Exceeded maximum allowable functional evaluations.

^{as} Actual solution.

^{fb} Feasible bounded solution.

^{pb} Infeasible and partially bounded solution.

^{ub} Infeasible unbounded solution.

The U-space TRECO required many function and gradient evaluations, where the limit state was very nonlinear in U-space. Other than these cases, the TRECO algorithms worked for all the test problems, both feasible and infeasible, and the required computational effort was comparable to the standard MPP search algorithms for the feasible cases.

7.4 RBO-CSSO2 Implementation Studies

In this section, three application problems are presented for which the RBO-CSSO2 methodology was applied. The main aim of this implementation study is to illustrate the basic characteristics of the RBO-CSSO2 methodology using an analytic problem, and also show the computational benefits obtained using this approach over a traditional RBO and RBO-CSSO methodologies for the control augmented structures design problem and a high performance, low cost structure design problem. The implementation details for these studies are very similar to those in RBO-CSSO implementation studies, unless otherwise mentioned.

7.4.1 Analytic Test Problem

The version of the analytic test problem used for this implementation is exactly the same as the one use for the RBO-CSSO implementation except that a new hard constraint, g_2^R was added. g_2^R was a dummy hard constraint that was never active and was included solely for the purpose of illustrating the robustness of the TRECO based reliability analysis for cases where the MPP searches were infeasible. This hard constraint and the bounds for \mathbf{x} , that was required for the bounded MPP searches were, for this implementation are given below:

$$g_2^R = X_1 + X_2^2 + 50$$

$$\mathbf{x} \text{ bounds: } -5 \leq x_1 \leq 10, -5 \leq x_2 \leq 10$$

This problem was implemented using the RBO-CSSO2 approach, where the MPP search for g_1^R was implemented using CSSO, whereas the MPP search for g_2^R was implemented without any decomposition strategies, since g_2^R was independent of the SVs. X-space TRECO was used for SSOs in the CSSO based MPP search and for the exact MPP search for the second hard constraint. In the CSSO based MPP search for g_1^R , SSO₁ was allowed to change x_1 alone, while SSO₂ was allowed to change both random variables. The move-limits in \mathbf{x} were chosen to be ± 20 percent of bounds in \mathbf{x} . No constraint screening was implemented for this problem, and this did not pose any problems in the RBO.

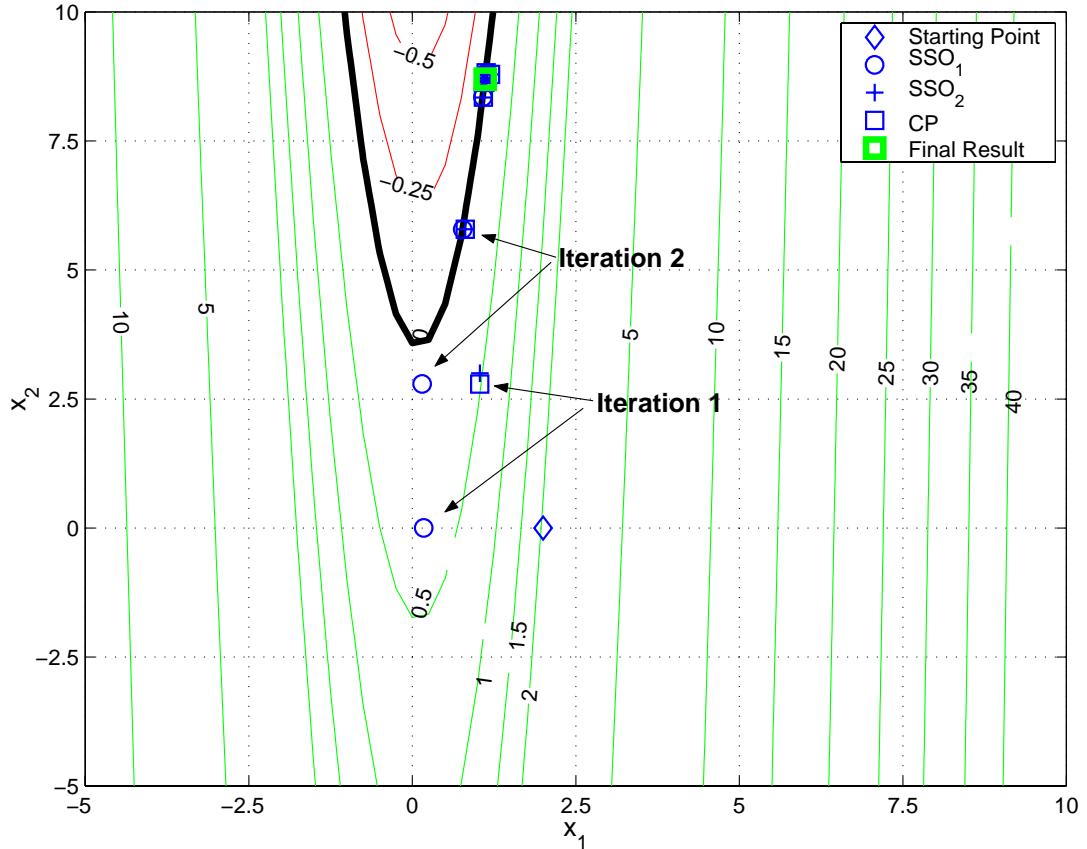


Figure 7.12. MPP Search Iteration History in X Space for the Analytic Test Problem at Initial Design

The initial starting point, \mathbf{d}^0 , for the RBO was chosen to be $\{2, 3\}$. The final design was the same as in traditional RBO. Figure 7.12 shows the SSO and CP results from

the MPP-CSSO for the constraint g_1^R in \mathbf{X} -space at the initial design. It can be seen that for the first two iterations, the optimum from SSO₁ (circles in the figure) occurs at the unbounded minima of g_1^R with respect to x_1 alone. The CP and SSO₂ results for the first iteration occur at the move-limits. Other than these two cases, the optima from SSOs and CP from other iterations all lie on the limit state surface $g_1^R = 0$. This illustrates that the proposed TRECO algorithm works as intended.

The iteration history in \mathbf{d} space is same as Figure 7.5 with the starting design corresponding to iteration 2. Figures 7.13 and 7.14 show the convergence histories of f , g^D , g_1^{rbo} and g_2^{min} . g_2^{min} is the minimum positive value of g_2^R obtained from the bounded MPP searches implemented using TRECO in X space. g_2^{min} remained constant at a value of 45 and the constrained solution occurred at $\{-5, 0\}$. This constrained solution was treated as the MPP for calculation of the g_2^{rbo} during the rSA, SSOs and CP in the RBO-CSSO framework and it remained inactive throughout. The treatment of this constraint in such a fashion did not present any issues with the convergence of the RBO framework and hence no special procedures to deal with this constraint was implemented.

7.4.2 Control Augmented Structure

The version of control augmented structure problem used here is exactly same as that used for the RBO-CSSO implementation study. The implementation used here and the one used in the RBO-CSSO implementation study are the same except for few differences. One difference is that the move-limits for the design variables was set as $\pm 10\%$, but the choice of this move-limit gives exactly the same convergence behavior as with a choice $\pm 15\%$. The other difference is that all the RBO methods i.e. traditional RBO, RBO-CSSO and RBO-CSSO2 approach here use \mathbf{X} -space TRECO as the reliability analysis routine. The bounds for the \mathbf{x} for the TRECO was set as $\mu \pm 9\sigma$, where μ is the mean and σ is the standard deviation. In the RBO-CSSO2 approach, exact MPP searches were

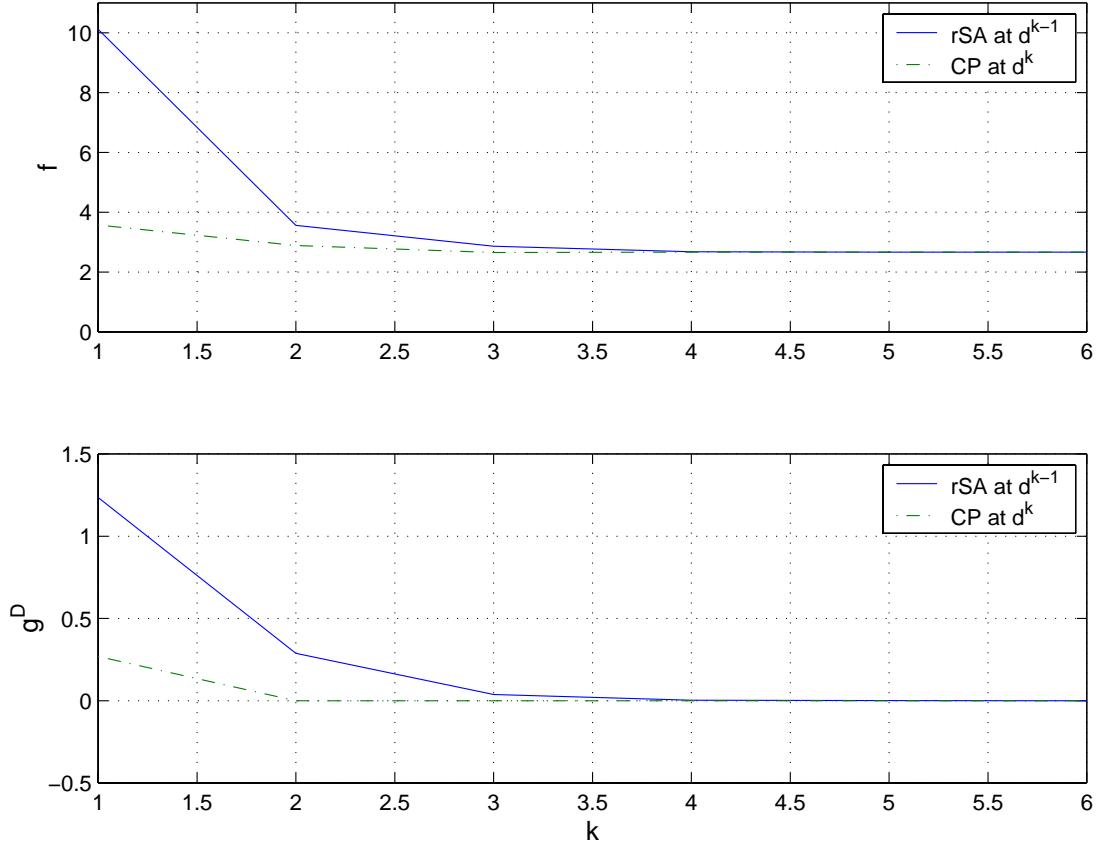


Figure 7.13. RBO-CSSO2 Convergence Plots for the Analytic Test Problem

implemented for the static constraints since they do not require the coupled system analysis, but just require the structures CA. MDO based MPP searches were performed for the dynamic constraints, g_{11} and g_{28} . The MDO approach used was a local CA based MDO approach. The MDO based MPP search for g_{11} consisted of an approximate MPP search using an exact structures analysis alone and a linear approximation to the controls analysis tools outputs. The MDO based MPP search for g_{28} consisted of an approximate MPP search using an exact controls analysis alone and a linear approximation to the structures analysis tool. The move-limits for \mathbf{x} in the approximate MPP search was set as $\pm 5\%$ of the overall bounds set for \mathbf{x} .

The final designs obtained for this test problem, using traditional RBO, RBO-CSSO

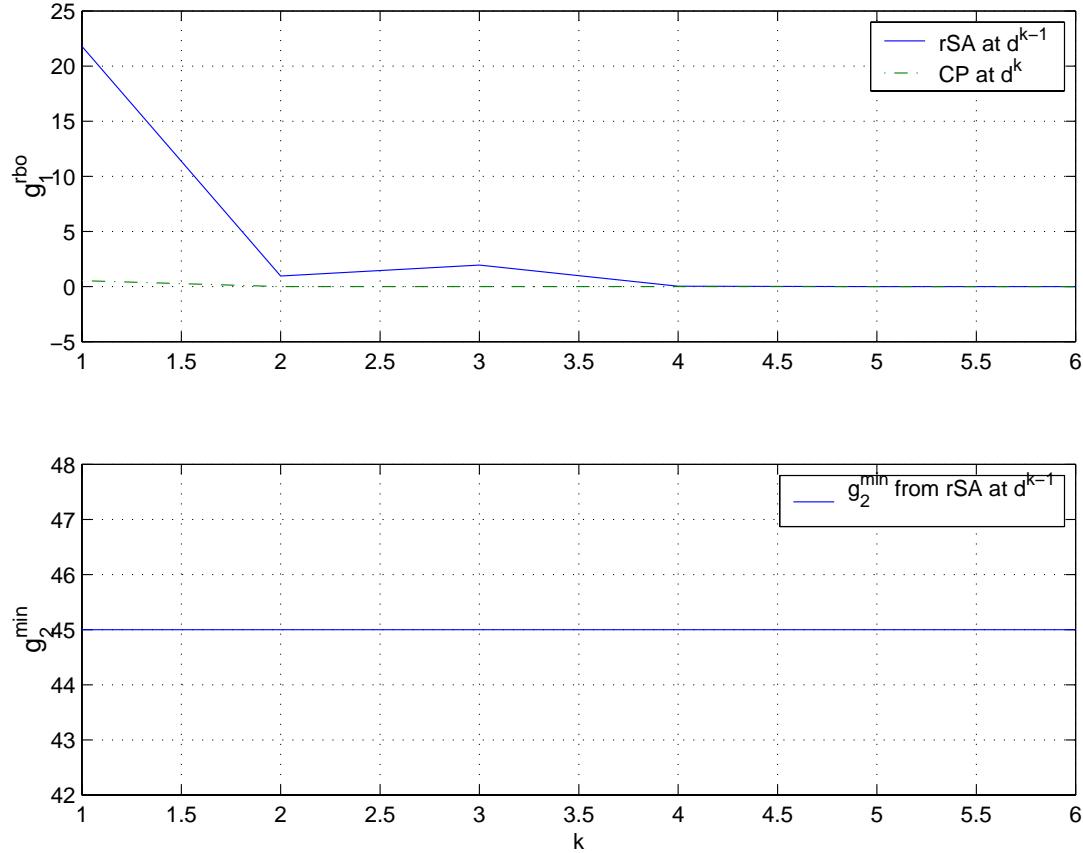


Figure 7.14. RBO-CSSO2 Convergence Plots for the Analytic Test Problem (Continued)

and RBO-CSSO2 approaches were the same, as was presented in Table 7.3. Table 7.11 show the computational comparison between the three methods. The computational savings of the RBO-CSSO over the traditional approach is minimal for this case, when compared to the results obtained in the RBO-CSSO implementation study. This is mainly because the TRECO is more efficient than the iHL-RF algorithm used in the RBO-CSSO implementation study. However, the RBO-CSSO2 approach requires approximately 30% fewer CA calls than both the traditional RBO and RBO-CSSO. The RBO-CSSO2 behaves very similar to RBO-CSSO approach with respect to iterations in \mathbf{d} , hence the convergence plots are similar to the ones presented during the RBO-CSSO implementation study section, and hence are not repeated here. The number of structures and controls

TABLE 7.11. COMPUTATIONAL COMPARISONS FOR CONTROL AUGMENTED STRUCTURE PROBLEM

	Trad. RBO	RBO-CSSO	RBO-CSSO2
CA ₁ calls	1071	1031	733
CA ₂ calls	973	949	652
SA calls	36	34	23

analysis calls used by the RBO-CSSO and RBO-CSSO2 approaches during each iteration after a rSA is performed is shown in the Figure 7.15. It can be seen the number of CA calls required by the local CA based MDO approach for the MPP search requires fewer CA calls during every iteration. The percentage reduction in CA calls is highest in the first iteration and reduces for subsequent iterations. This happens mainly because the starting points for the MPP searches in the rSA is set at the MPPs obtained during the previous iteration. Hence the number of analysis calls are very few during the later iterations where the changes in design variables gets smaller.

TABLE 7.12. UNCERTAIN VARIABLE MOVE-LIMIT STUDY FOR CONTROL AUGMENTED STRUCTURE PROBLEM

	1%	5%	15%	25%
CA ₁ calls	1119	733	744	750
CA ₂ calls	968	652	665	670
SA calls	32	23	23	23

A study on the choice of different move-limits in \mathbf{x} , for the MDO based MPP search, was conducted. Four different cases, including the main result presented earlier, were considered. It can be seen that by reducing the move-limits to 1% increased the number of required CA calls, and increasing the move-limits to 15% and 25% approximately

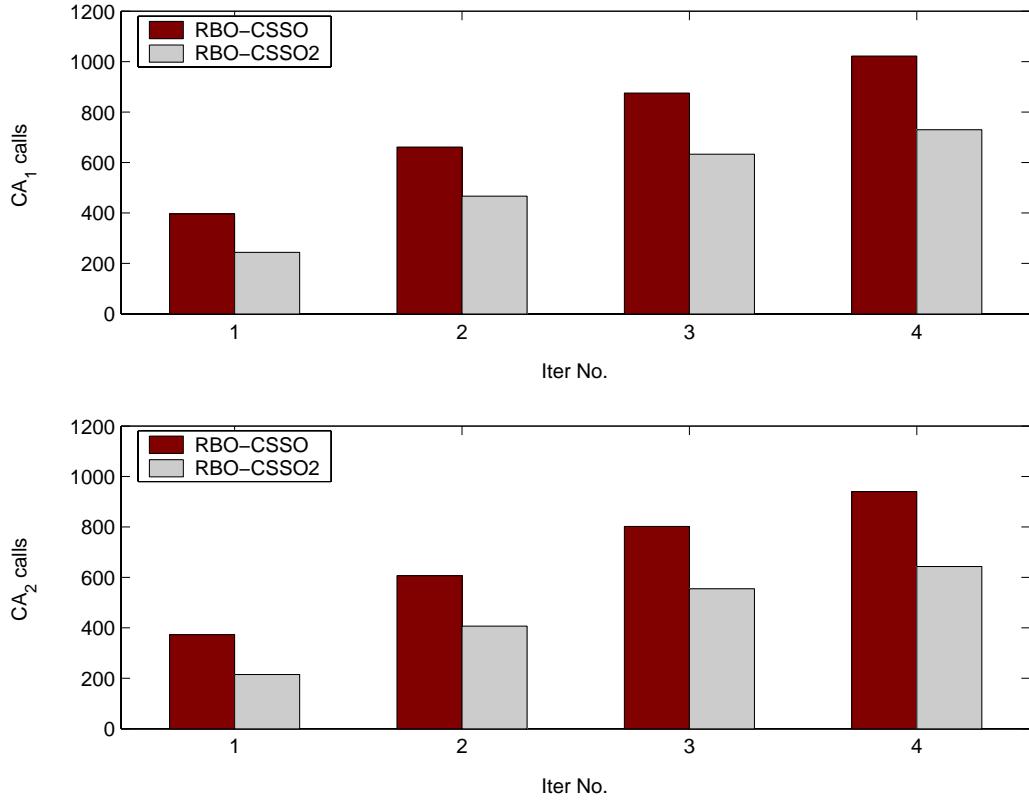


Figure 7.15. Comparison of CA Calls for the Control Augmented Structures Problem

required similar computational effort.

7.4.3 High Performance, Low-Cost Structure

A coupled version of the High Performance, Low-Cost Structure algorithm is used for the RBO-CSSO2 implementation study. $e_1=50000$, $e_2=5000$, $e_3=40000$ and $e_4=10000$ were set. The uncoupled version used for RBO-CSSO implementation study is unsuitable for RBO-CSSO2, due to the absence of coupling between CA_c and CA_s . Infact for the choice of random variables, used in the RBO-CSSO implementation study, the coupling is absent because the loads P obtained from CA_c are completely independent of any of the random variables. Hence one initial evaluation of P would be sufficient while performing the MPP searches for hard constraints local to CA_s i.e. stress constraints.

Hence the coupled version was chosen. Even in this case, the coupling is linear and can be easily and exactly solved using an approach like Newton-Raphson. This version was also implemented and it was observed that the local CA based MPP search was more expensive than a standard undecomposed MPP search. The important conclusion that was made from these failed attempts was that the SA had to be expensive in order for the RBO-CSSO2 approach to be more computationally efficient than RBO-CSSO. Hence for the implementation purposes here, the SA was solved using a rigorous expensive unconstrained pattern search algorithm assuming that the linear nature of the coupling was unknown.

The same random variables with same properties were selected as in the uncoupled version. The same hard and soft constraints were also selected for this problem. The design variable move-limits for RBO-CSSO and RBO-CSSO2 approach were set as $\Delta L_{1-3} = \pm 1$, $\Delta M_{1,3} = \pm 200$, $\Delta M_{2,4} = \pm 400$, $\Delta A_{1-5} = \pm 1.0$ and $\Delta A_{6-10} = \pm 0.5$. The algorithm used for MPP searches in traditional RBO, RBO-CSSO and RBO-CSSO2 approaches was **X**-space TRECO. The bounds for the **x** for the TRECO was set as $\mu \pm 9\sigma$. For the RBO-CSSO2 approach, the MPP searches for the stress constraints, that were chosen as hard constraints, were implemented using the local CA based MDO approach. So, during the approximate MPP search an exact CA_s analysis was used and linear approximation to CA_c analysis output was used. The move-limits for **x** for the approximate MPP search was selected as $\pm 25\%$ of the overall bounds.

The initial design for all the three RBO methods were set at the optimum obtained from the traditional design optimization. Table 7.13 shows the initial design and final designs obtained from all the three methods. At the initial design $g_4, g_5, g_6, g_7, g_{10}$ and g_{11} were active. The final design obtained from traditional RBO corresponds to a slightly higher merit function value than the designs obtained from RBO-CSSO and RBO-CSSO2 approach. The loads, lengths and areas obtained from the traditional RBO are larger

than the designs from the RBO-CSSO and RBO-CSSO2 approaches. Comparing the computational expenses, it can be seen that the RBO-CSSO is nearly 30% cheaper than traditional RBO in terms of CA calls. The RBO-CSSO2 is cheaper than RBO-CSSO in terms of CA_c and CA_s calls by roughly 30%, and its cheaper than traditional RBO by 50%.

TABLE 7.13. DESIGNS AND COMPUTATIONAL COMPARISON FOR COUPLED HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM

	Initial Design	Trad. RBO	RBO-CSSO	RBO-CSSO2
L_1	100.00	100.00	100.00	100.00
L_2	67.98	72.26	69.99	69.99
L_3	65.94	71.04	67.95	67.95
M_1	7510	7509.3	7117.6	7117.4
M_2	11262	11262	10767	10767
M_3	7504.8	7504.1	7106.5	7106.5
M_4	11259	11259	10765	10765
A_1	5.18	6.82	6.02	6.02
A_2	15.33	19.65	17.41	17.41
A_3	14.45	18.77	16.51	16.51
A_4	5.41	7.06	6.26	6.26
$A_{5,6}$	1.00	1.00	1.00	1.00
A_7	3.38	4.45	4.11	4.11
A_8	3.80	4.85	4.51	4.51
$A_{9,10}$	1.00	1.00	1.00	1.00
f	10.55	12.022	11.963	11.963
W	1566.77	2002.19	1792.02	1791.99
$\sum_i P_i$	364196	343588	300564	300558
$\sum_i M_i$	37535.8	37533.5	35756.7	35756.5
CA_c calls	-	69652	46563	30865
CA_s calls	-	69818	46789	34876
CA_d calls	-	1408	982	977

The convergence plots for the traditional RBO are shown in Figures 7.16 and 7.17.

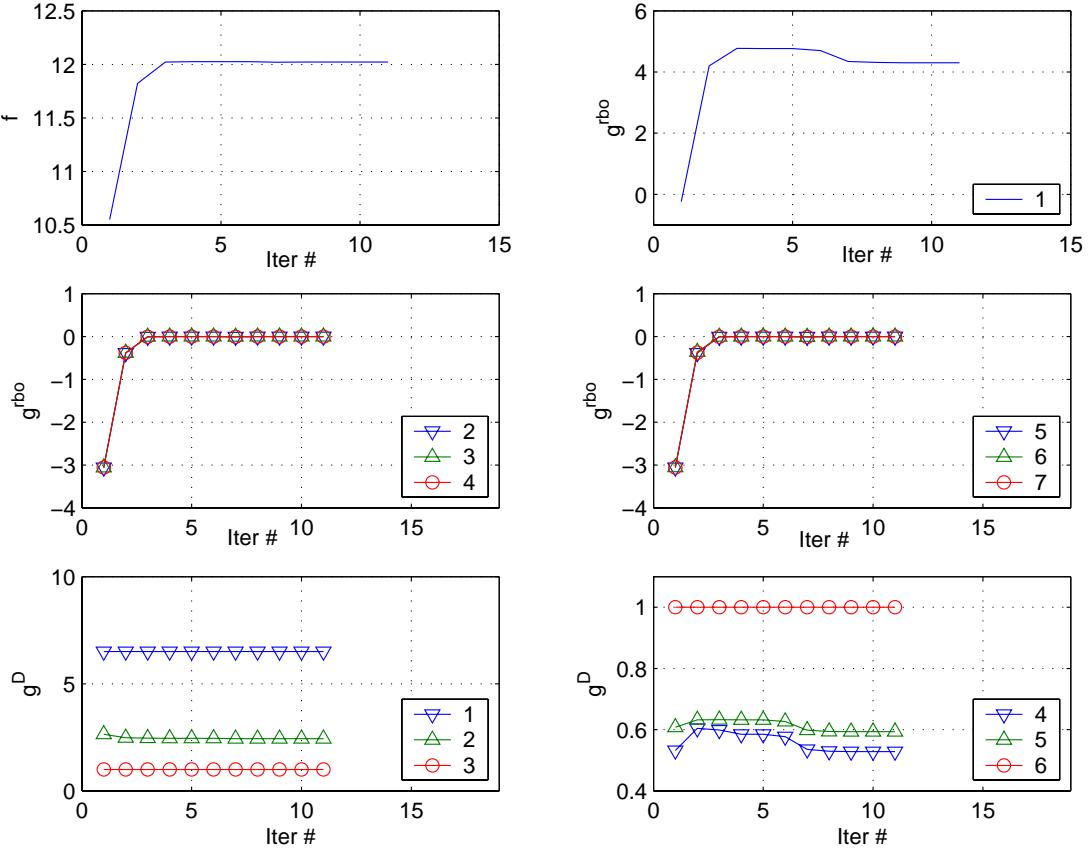


Figure 7.16. Traditional RBO Convergence Plots for Coupled High Performance, Low-Cost Structure Problem

The reliability constraints 2, 3, 4, 5, 6 and 7 were active at the final design. The masses remained almost constant. There was an increase in the lengths, L_2 and L_3 after iteration 5, that caused a decrease in reliability constraint 1. There was an increase in the areas, A_{1-4} , A_7 and A_8 until iteration 3 after which they remained constant. The other areas remained constant throughout.

Similar designs were obtained from RBO-CSSO and RBO-CSSO2. The convergence plots are shown in Figures 7.18 and 7.19. The same reliability constraints as in traditional RBO were also active. Masses M_{1-4} decreased to their lower move-limits in the first 3 iterations and remained constant after that. The lengths L_2 and L_3 increased to their upper move-limits in the first 3 iterations and remained constant after that. L_1 remained

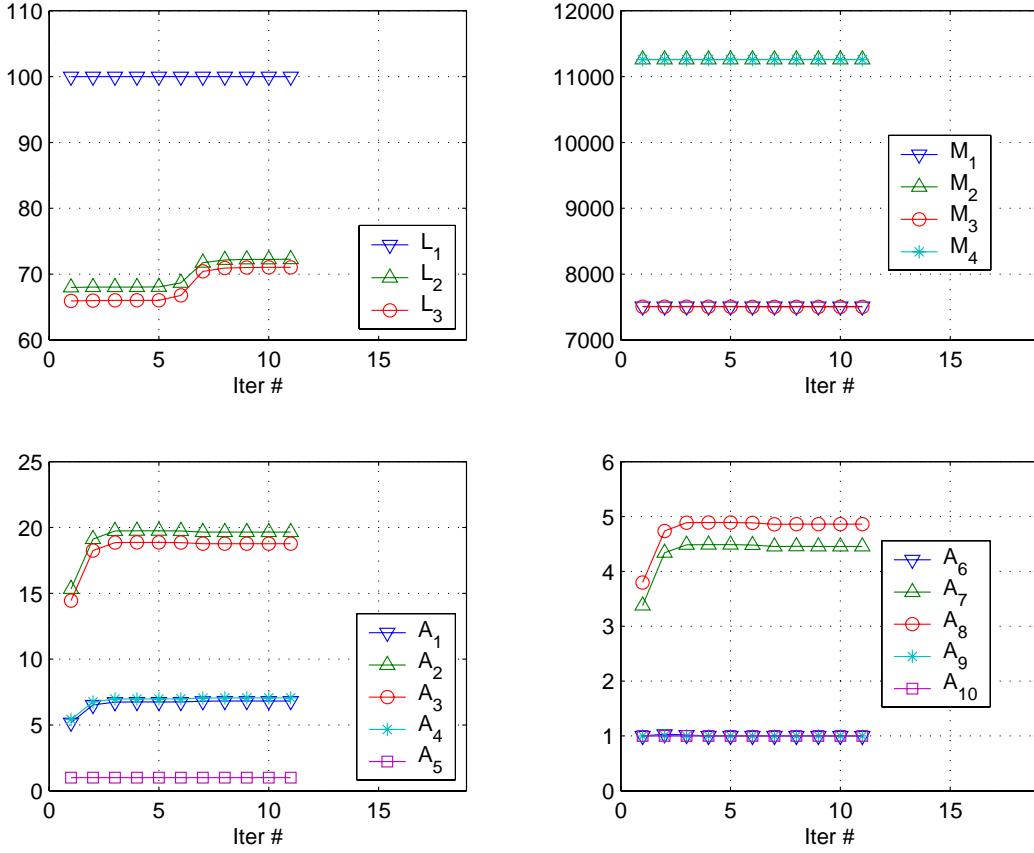


Figure 7.17. Traditional RBO Convergence Plots for Coupled High Performance, Low-Cost Structure Problem (Continued)

constant throughout. Areas A_2 and A_3 constantly increased to their upper move-limits till iteration 3 and remained almost constant after that. Areas A_5 , A_6 , A_9 and A_{10} all increased to their upper move-limits after iteration 1 but they reduced to their lower bounds and remain at their lower bounds after iteration 2. A_1 , A_4 , A_7 and A_8 increased to their upper move-limits till iteration 3 and reduced to their lower move-limits after iteration 3 and practically remained constant after iteration 4.

Figure 7.20 shows the number of CA calls used after a rSA call for each iteration for both RBO-CSSO and RBO-CSSO2. There is a good reduction in number of CA_c calls throughout in the RBO-CSSO2 approach. The number of CA_s calls for the first iteration is comparable in both methods, but after the first iteration there is continued reduction in

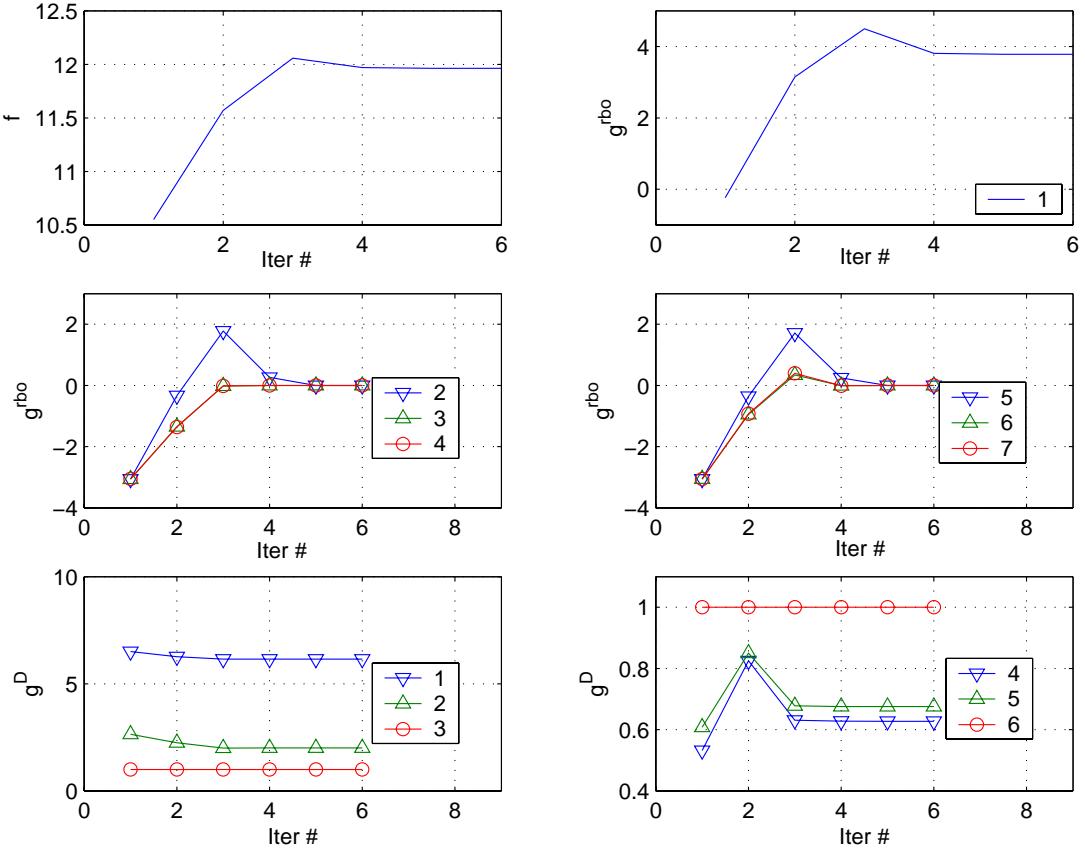


Figure 7.18. RBO-CSSO2 Convergence Plots for Coupled High Performance, Low-Cost Structure Problem

the number of CA_s calls. The percentage reduction in CA_s is smaller than CA_c primarily because the MDO based MPP searches used exact CA_s during the approximate MPP search.

A study on the effect of different choices of move-limits in **d** for the RBO-CSSO approach was performed for this problem. The move-limits for certain design variables were varied keeping the rest fixed at the original choice that was given earlier. 5 cases were considered as shown, where $\Delta M_{1,3} = 400$, $\Delta M_{2,4} = 200$, $\Delta L = 0.5$, $\Delta A_{1-5} = 0.5$ and $\Delta A_{6-10} = 1.0$. The results are presented in Table 7.14. It can be seen that different designs but with similar objective function are obtained again confirming the non-linearity of this problem and existence of several local minima. The number of CA calls required

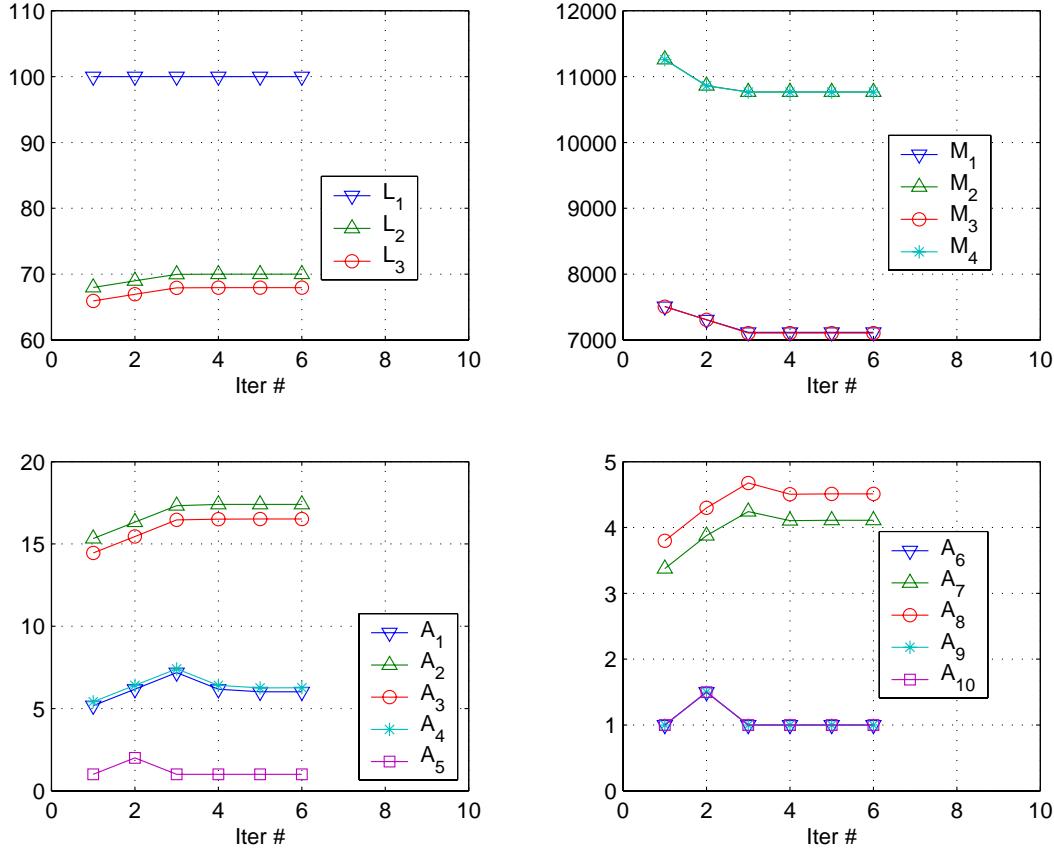


Figure 7.19. RBO-CSSO2 Convergence Plots for Coupled High Performance, Low-Cost Structure Problem (Continued)

for these 5 cases, suggest that move-limits for $M_{2,4}$ could have been increased. The designs obtained for the cases $\Delta L = 0.5$ and $\Delta A_{6-10} = 1.0$ are very close to the final design obtained for the original move-limits (see Table 7.13). A move-limit study in \mathbf{x} for the RBO-CSSO2 approach was performed. As shown in Table 7.15, 4 different choices of move-limits for \mathbf{x} were considered that included the original implementation presented earlier. Similar observations as in the control augmented structures problem can be made, that is decrease in move-limits from the original move-limits of 25% increased the computational effort and the increase in move-limits required slightly higher computational effort.

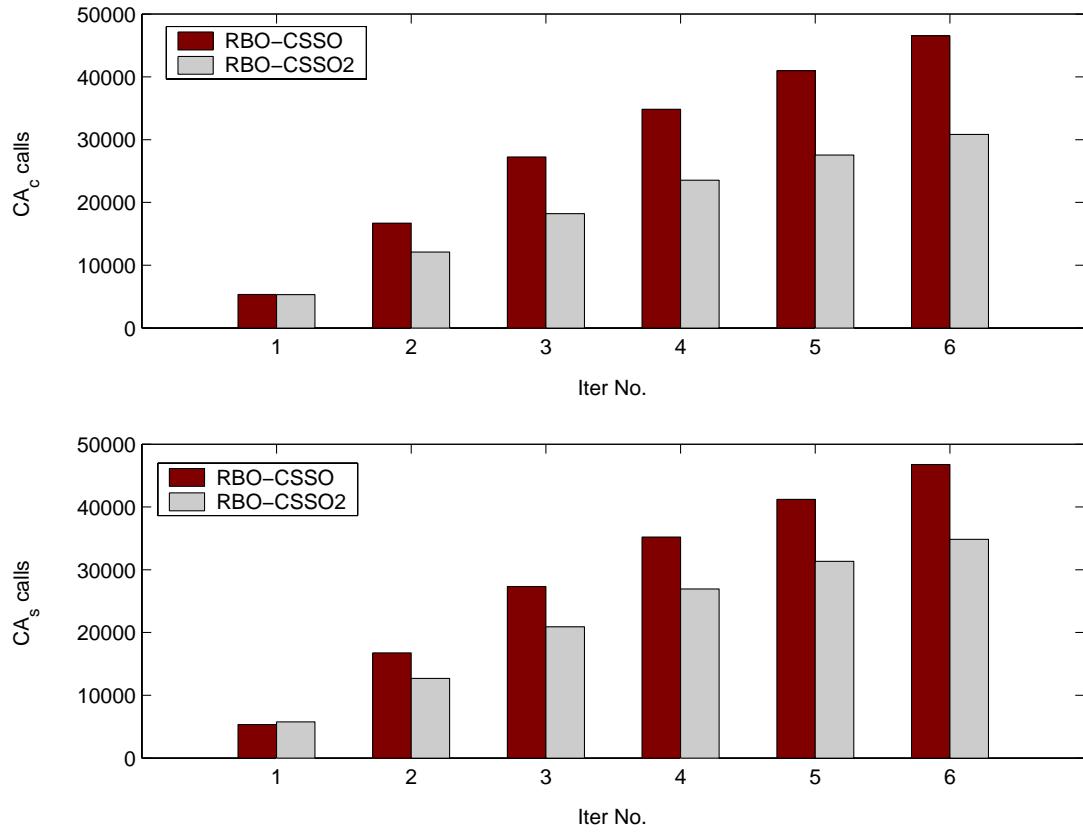


Figure 7.20. Comparison of CA calls for Coupled High Performance, Low-Cost Structure Problem

7.4.4 Summary

The basic working of the RBO-CSSO2 approach was shown using the analytic test problem. Significant computational savings (around 30%) in CA calls was observed for the control augmented structure problem and high performance, low-cost structure problem, over traditional RBO and RBO-CSSO approaches. It was observed from various test studies of the high performance, low-cost structure that RBO-CSSO2 approach was only more computationally efficient over the RBO-CSSO approach when the MPP searches required repeated evaluation of expensive iterative SAs. Move-limit studies for \mathbf{x} were made and it was observed that both the control augmented structures problem and high performance, low-cost structure problem behaved in a fairly benign fashion.

TABLE 7.14. DESIGN VARIABLE MOVE-LIMIT STUDY FOR COUPLED HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM

	$\Delta M_{1,3} = 400$	$\Delta M_{2,4} = 200$	$\Delta L = 0.5$	$\Delta A_{1-5} = 0.5$	$\Delta A_{6-10} = 1.0$
L_1	100.00	100.00	100.000	100.000	100.000
L_2	69.90	69.96	68.99	69.96	69.96
L_3	67.86	67.93	66.95	66.61	67.92
M_1	7070.8	7108.4	7108.4	7091.8	7135.9
M_2	10842	10862	10763	10455	10779
M_3	7059.5	7106.5	7106.5	7091.2	7106.5
M_4	10839	10859	10760	10452	10777
A_1	6.02	6.08	6.00	5.80	6.03
A_2	17.37	17.53	17.39	16.86	17.49
A_3	16.47	16.66	16.53	15.99	16.54
A_4	6.26	6.31	6.24	6.03	6.28
$A_{5,6}$	1.00	1.00	1.00	1.00	1.00
A_7	4.14	4.16	4.11	3.94	4.12
A_8	4.54	4.56	4.52	4.35	4.52
$A_{9,10}$	1.00	1.00	1.00	1.00	1.00
f	11.965	11.964	11.963	11.969	11.962
CA_c calls	37928	55071	46948	60829	57422
CA_s calls	38063	55398	47189	61161	57725
CA_d calls	777	1151	982	1325	1244

7.5 MCS-based RBO Implementation Studies

In this section, two application problems for the MCS-based RBO are presented. In these application problems, the reliability constraints are constraints on component and series system probabilities of failure obtained using Axis Orthogonal Simulation. The main aim of this somewhat brief implementation study, is to demonstrate the working of a MCS-based RBO methodology and point out some of the issues that arise while using this methodology. The SQP algorithm in Matlab Optimization Toolbox was used as the optimizer for both the traditional and MCS-based RBO methods. In the present

TABLE 7.15. UNCERTAIN VARIABLE MOVE-LIMIT STUDY FOR COUPLED HIGH PERFORMANCE, LOW-COST STRUCTURE PROBLEM

	5%	10%	25%	50%
CA _c calls	43933	32802	30865	31466
CA _s calls	54236	38218	34876	35233
CA _d calls	977	977	977	977

implementation, exact analysis are used for the MCS and no approximation concepts like limit state surface approximations have been used in this work.

7.5.1 Analytic Test Problem

In this version of the analytic test problem, the g_2^R was chosen as given by the following equation:

$$g_2^R = X_1 + 1.5\eta - 0.5X_2 - 2.5$$

In this version, the random variables were chosen to be independent and normally distributed with standard deviations of 0.5 and 0.5 for X_1 and X_2 respectively. Two cases were implemented. The first case was a component MCS-based RBO using 100 simulations for both hard constraints. The required reliability for both the hard constraints was set at 3. The initial design for this case was selected as the result from component FORM-based RBO. The second case was a series system MCS-based RBO using 100 simulations for both hard constraints. The required system reliability was set as 3 for this case. The weights for the sampling density function were chosen based on FORM results from component FORM-based RBO result. The initial design was chosen as the result obtained from a series system FORM-based RBO that used a unimodal upper bound.

The simulations for the first case, called Case A in this section, are shown in Figure

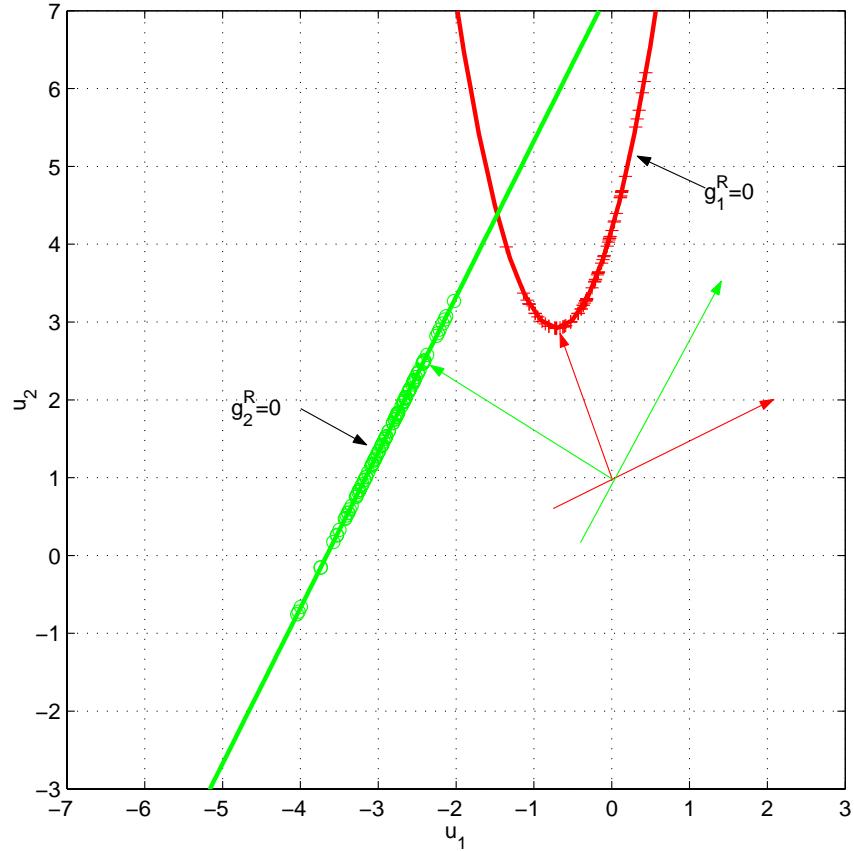


Figure 7.21. Component MCS for Analytic Test Problem at Initial Design

7.21. In the figure, ‘+’ signs are projected samples for g_1^R and the ‘o’ signs are projected samples for g_2^R . The coordinate axis for the simulation variables are also shown. During RBO for this problem, there were 29 samples that could not be projected onto $g_1^R = 0$ surface during the 1st iteration. But after the 2nd iteration the number of unprojectable sample points remained at 33. Consequently, a discontinuity in the probability of failure estimate occurs between the 1st and 2nd iteration but this did not hamper the convergence for this problem, which can be seen from the convergence plots shown in Figure 7.22. At the initial design, that is the result from the component FORM-based RBO, the reliability estimates for both the hard constraints were conservative according to the axis-orthogonal simulation results. Hence the optimizer was able to lower the reliability indices to the

required level and was also able to lower the merit function. During all the iterations, the deterministic constraint was active.

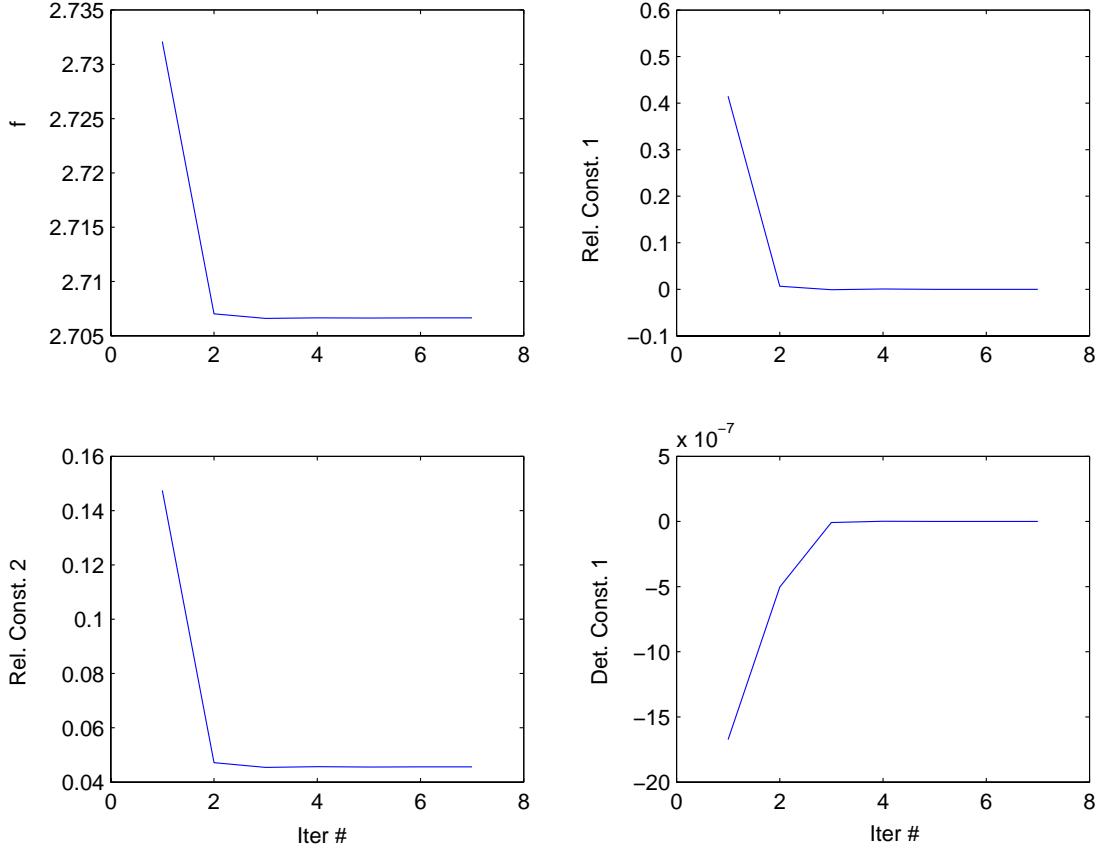


Figure 7.22. Convergence Plots for Analytic Test Problem - Case A

The simulations for the second case, called Case B in this section, are shown in Figure 7.23. The number of simulations and seeds were fixed at the same values for both cases. In the figure it can be seen that there are projected samples, shown as '+', on $g_2^R = 0$ surface that were previously unprojectable onto $g_1^R = 0$ in Case A. There were 28 such projected samples ('+') on $g_2^R = 0$ surface during the first iteration but then it changed and remained at 29 for the rest of the iterations. So a derivative discontinuity is encountered in this case between the 1st and 2nd iterations. The initial design was chosen as the result obtained from a series system FORM-based RBO that used a unimodal upper bound.

The convergence plots are shown in Figure 7.24 and it can be seen that the series system FORM-based RBO was conservative and hence a decrease in system reliability to the required reliability level and a decrease in merit function was achievable. During all the iterations, the deterministic constraint was active.

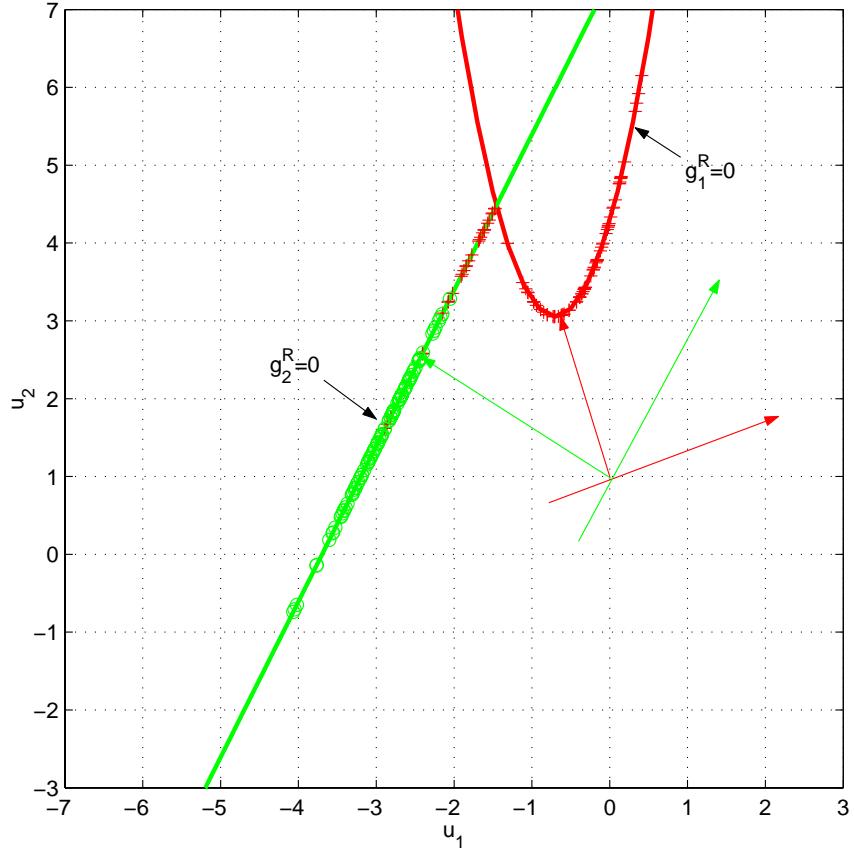


Figure 7.23. Series System MCS for the Analytic Test Problem at Initial Design

All the starting and final designs including the traditional optimum are shown in Figure 7.25. The traditional optimum is achieved at the intersection of $g_1^R = 0$ and $g^D = 0$ contours. All the RBO designs achieved, from FORM and MCS for both cases all lie on $g^D = 0$ contour.

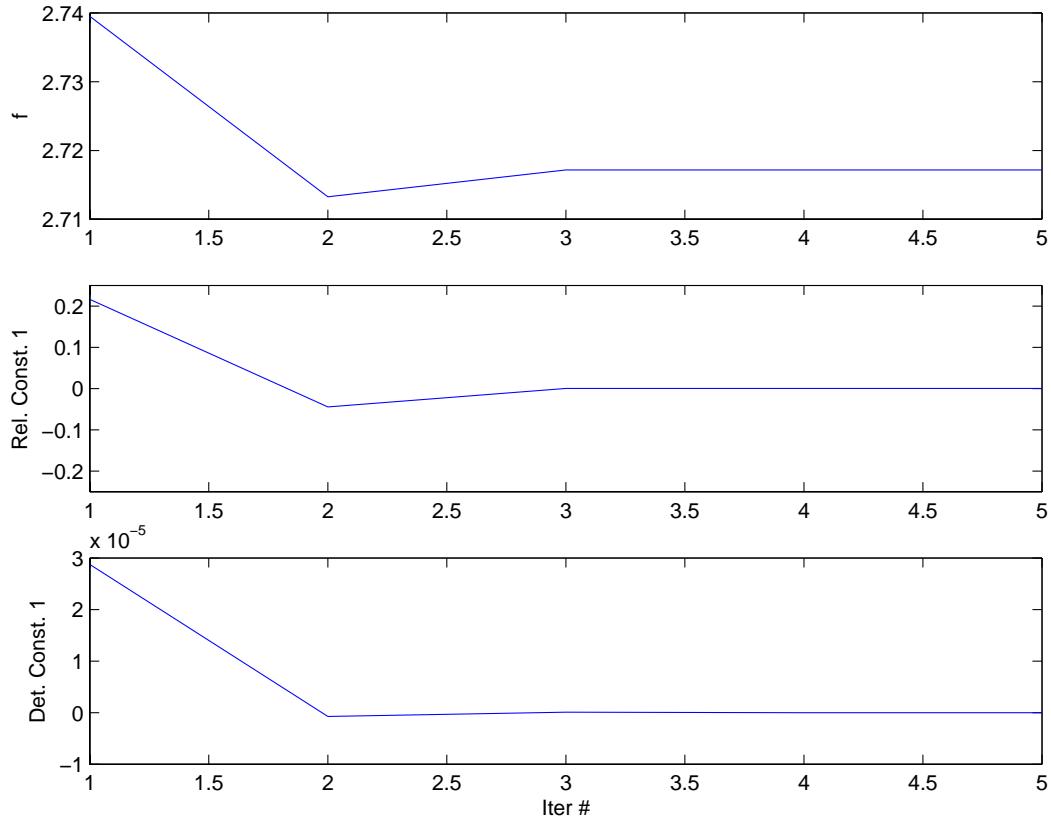


Figure 7.24. Convergence Plots for the Analytic Test Problem - Case B

7.5.2 Control Augmented Structures Problem

In this version of the control augmented structures problem, σ_a , ρ , E , T_1 , T_2 and T_3 were chosen as random variables. $\rho = \text{LN}(0.1, 0.01)$, $E = \text{LN}(10500, 1050)$, $\sigma_a = \text{LN}(3000, 300)$, $T_{1,3} = \text{U}(1000, 400)$ and $T_2 = \text{U}(5000, 2000)$ were set for this test case. Constraints 1, 6, 14, 16, 18, 20 and 22 were chosen as hard constraints for this problem and the rest were treated as soft constraints. All the hard constraints are static constraints and hence do not require the coupled analysis. A component MCS-based RBO was implemented for this problem. The required reliability was set at 3. The result from the component FORM-based RBO was selected as the initial design. The number of simulations were set as 500 for each hard constraint.

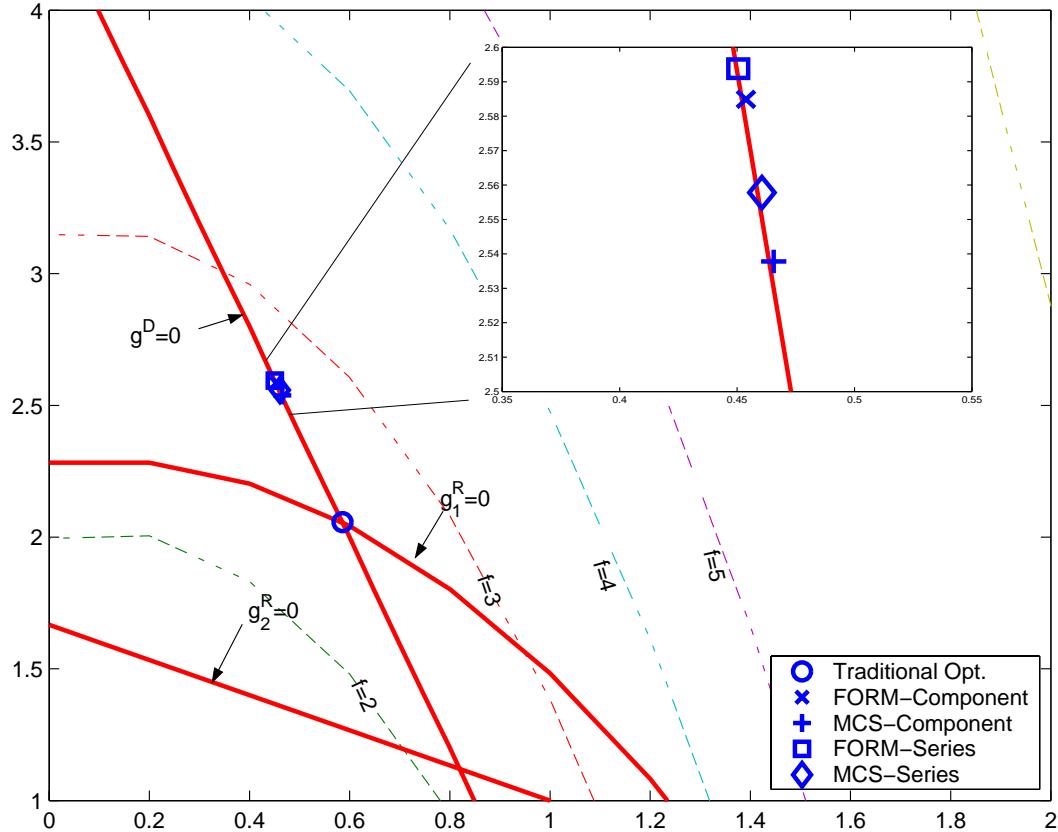


Figure 7.25. Designs for Analytic Test Problem

The traditional optimum, the design obtained from component FORM-based RBO and component MCS-based RBO are shown in Table 7.16. At the traditional optimum the hard constraint g_1 is active. At the final designs obtained from FORM and MCS-based RBO, constraints 14, 16, 18, 20 and 22 were active. It can be seen that the design obtained from FORM-based RBO corresponds to a larger and heavier structure than the design obtained from the MCS-based RBO. The convergence plots are shown in Figure 7.26. It can be seen that the FORM-based RBO is more conservative and hence a decrease in the reliabilities to the required levels as well as a decrease in merit function by 30 were possible. The computational expenses involved for this example was extremely high. A CPU time of 16243s was required for this example that was implemented in Matlab 6.1

on a Personal Computer with 1.8 GHz clock-speed, Pentium 4 Processor and Microsoft Windows XP operating system. The number of structures analysis calls required for this implementation study was 272034, which is nearly 250 times greater than the number of analysis call required by a FORM-based RBO starting from a traditional optimum design. Hence this implementation study showed that MCS-based RBO can be highly impractical even for medium sized problems and hence computationally efficient approximation concepts based RBO methodologies are needed for medium and large sized problems.

TABLE 7.16. DESIGNS FOR CONTROL AUGMENTED STRUCTURE PROBLEM

	Trad. Opt.	FORM-RBO	MCS-RBO
b_{1-5}	3.00	3.00	3.00
h_1	3.703	3.740	3.698
h_2	7.040	9.665	9.478
h_3	9.807	13.557	13.298
h_4	11.998	16.582	16.259
h_5	13.840	19.138	18.763
c	0.06	0.06	0.06
f	1493.87	1926.59	1894.01

7.5.3 Summary

The character of the axis orthogonal simulation and the MCS-based RBO methodology was illustrated using the analytic test problem for component and series system reliability constraints. In the analytic test problem, discontinuities in probabilities of failure and its sensitivities were observed during the RBO, but this did not cause any convergence problems. The computational expenses associated with the control augmented structures problem were extremely high and use of RBO methodologies based on approximations concepts[75, 71] and parallel processing are strongly recommended and are promising

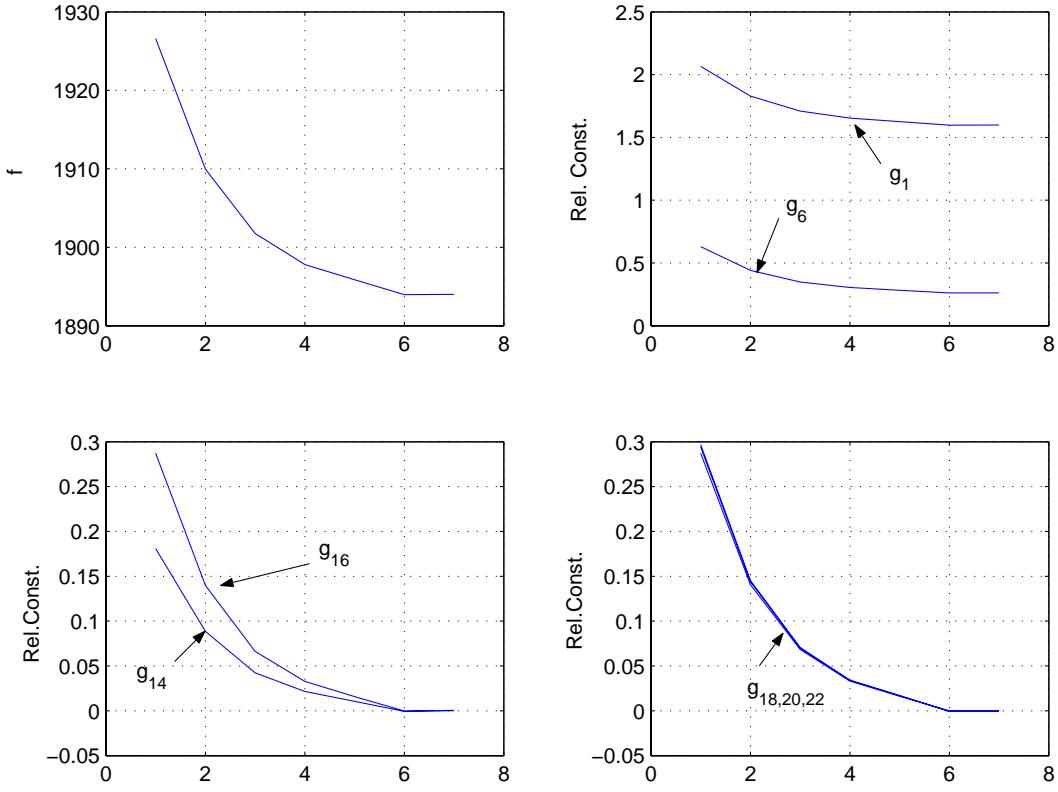


Figure 7.26. Convergence Plots for Control Augmented Structure Problem

candidates for future research work. On a positive note, despite the excessive computational costs, improved designs were obtained from the MCS-based RBO methodology compared to those obtained from FORM-based RBO methodologies.

CHAPTER 8

CONCLUSIONS

The RBO methodologies developed in this research use state-of-the-art probabilistic reliability analysis techniques that provide accurate probability of failure estimates in an efficient manner, and also have several advantages such as computability of relative importance of random variables useful for screening of uncertain variables or quality control applications where manufacturing variation is to be reduced and updatability of reliability estimates using newly available statistical information based on Bayesian updating. The methodologies assume prior statistical information about the random variables is available or reasonable statistical distributions can be assigned to the random variables, which often may not be the case. These methodologies are not intended to treat uncertainties that arise due to modeling and simulation. Another disadvantage of these techniques are that they are computationally intensive. RBO application problems with upto 17 design variables and 30 random variables were considered for the implementation studies and based on this experience, it is expected that the RBO methodologies can be extended to medium size problems consisting of atmost 100 continuous design variables and 100 continuous random variables.

The main research objectives that were set for this work were the development of computationally efficient Reliability-Based Optimization (RBO) methodologies for multidisciplinary system design, the development of robust Most Probable Point (MPP) search algorithms for probabilistic reliability analysis and the incorporation of Monte Carlo Sim-

ulation (MCS) techniques in RBO.

The first objective was realized by the development of two Concurrent Subspace Optimization (CSSO) frameworks for RBO that specifically addressed the issue of high computational costs associated with performing traditional RBO for multidisciplinary systems. These RBO approaches consist of three phases, an exact reliability and deterministic analysis phase, an approximation building phase and an approximate RBO phase based on CSSO principles. One of the approaches, called RBO-CSSO, employs exact multidisciplinary analysis during the exact reliability analysis phase. The other approach, called RBO-CSSO2, employs efficient multidisciplinary optimization for the MPP searches in the exact reliability analysis phase.

The second objective was realized by the development of two algorithms for MPP searches that could give constrained solutions to infeasible MPP searches when subject to strict bound constraints. These algorithms are a Trust Region-based Equality Constrained Optimization (TRECO) algorithm and an Elastic Sequential Quadratic Programming (ESQP) approach. These algorithms are required for the successful implementation of the multidisciplinary optimization approaches in the RBO-CSSO2 approach and can also be potentially used to screen inactive hard constraints in a RBO problem with many active and inactive hard constraints.

The third objective was realized by choosing those MCS techniques from existing techniques that provide smooth estimates of probabilities of failure and that provide analytic sensitivities of these probabilities of failure. Conditional expectation MCS, that consists of directional simulation and axis orthogonal simulation, was chosen based on these criteria.

In the remainder of this chapter, the merits and drawbacks of the RBO-CSSO approach, the bounded MPP search algorithms and the multidisciplinary MPP search methodologies used in the RBO-CSSO2 approach will be discussed. Finally, some possibilities

and directions for extension of this research work will be discussed.

8.1 RBO-CSSO Approach

The implementation studies of the RBO-CSSO approach presented in this dissertation show that the RBO-CSSO approach is more computationally efficient (nearly 25% cheaper) than a traditional RBO approach. The computational efficiency is obtained as a result of the use of CSSO for the approximate RBO phase and use of limit state approximations and approximations of exact reliability constraints in the approximate RBO phase. The main attractive feature of the RBO-CSSO and the RBO-CSSO2 approaches is that they allow for the concurrent design by different disciplines, in form of the Subspace Optimization (SSOs) and the SSOs can all be implemented in a parallel fashion.

The potential problems that can arise in the current implementation of the RBO-CSSO and RBO-CSSO2 approaches are mainly associated with features of the CSSO approach. The allocation of the design variables among the various SSOs is not straightforward and allocation of design variables, that have low influence on a design, to a particular SSO at an infeasible design can potentially render the SSO ineffective during the Coordination Procedure. The Coordination Procedure used in the current implementation uses the iteration histories from the SSOs for creating approximations of merit functions and constraints. The approximations can be of poor quality if there are not enough iteration histories and if the iteration histories are unevenly distributed in the design space. Additionally, when small move-limits are used at an infeasible design, the SSOs and CP can be infeasible. The current implementation uses an SQP optimizer that may not work well for such cases and hence optimization methodologies that deal with infeasibilities are desirable. Finally, to increase the robustness of the CSSO approach it is desirable to have dynamic move-limits with heuristics to change the move-limits.

8.2 Bounded MPP Search Algorithms

The implementation studies indicate that the TRECO is more efficient than the ESQP algorithm. The ESQP was also shown to fail under certain conditions. Hence TRECO was used for the RBO-CSSO2 approach. The TRECO can give the best constrained solution to an infeasible bounded MPP search, whereas existing conventional MPP search algorithms would fail. The TRECO algorithm gives the MPP when the bounded MPP search was feasible, whereas some conventional MPP search algorithms, like SQP could fail in some cases as was shown in the implementation studies. The computational costs of the TRECO and the algorithms are comparable to other algorithms for test cases with feasible bounded MPP searches.

Despite the advantages, the TRECO algorithm can have some potential problems. Since the constrained solution for the infeasible bounded MPP search, is based only on a linear approximation of the hard constraint, the method could fail or be very inefficient when the hard constraint is very nonlinear. This was shown in the results for the third analytic problem. The other potential problem is that the TRECO algorithm could be inefficient because of the use of positive-definite Hessians for highly non-linear problems that may not have the positive-definiteness property. It is preferred to have Hessian update schemes like SR1, that generate general Hessians.

8.3 Multidisciplinary MPP Search Methodologies

Two multidisciplinary MPP search methodologies were presented in this research. One methodology is based on the CSSO approach. The other methodology is similar to a CSSO but the approximate optimization phase is performed only by the local discipline that determines the state variables on which the hard constraints explicitly depend. The latter methodology, when used in the RBO-CSSO2 approach, was shown to be computationally more efficient (nearly 30%) for coupled multidisciplinary RBO problems, than

an RBO-CSSO approach.

The MPP search methodologies can be inefficient for weakly coupled or uncoupled multidisciplinary systems, as was the case with the uncoupled version of the high performance, low-cost structure application problem. Other deficiencies of the present implementation of the MPP search methodologies are that they use constant move-limits and it is preferred to have dynamic move-limits with heuristics for robustness of the methodologies when applied to nonlinear problems. The present implementation uses TRECO to solve the potentially infeasible approximate optimization problem to obtain a solution. If the starting point is infeasible, then this approach is equivalent to attaining feasibility first and then obtaining a solution to the MPP search problem. This can be inefficient if the number of iterations for which the approximate optimization is infeasible are large. Hence it is preferable to have an MPP search methodology that is based on a two-step approach, one step (quasi-normal) that reduces infeasibility and other step (tangential) that progresses towards the solution, like the one implemented in the TRECO algorithm.

8.4 MCS-based RBO

The conditional expectation-based MCS techniques are better suited for RBO than indicator-based MCS because of less discontinuities in the estimates of the probabilities of failure with design changes, but mainly because the analytic sensitivities are available. The implementation studies for the MCS-based RBO indicate that significant improvements in designs compared to a First Order Reliability Method based RBO can be achieved for certain problems.

The pressing issue in the current implementation of the MCS-based RBO is that it is computationally expensive. Hence it is necessary to use approximation concepts for constructing the hard constraints to be used in the MCS-based RBO. Another possibility is the parallelization of the MCS among various processors, that will significantly reduce

the computational time. One additional issue is that noise can arise due to variation of number of samples projectable on the limit state with change in designs, as was illustrated in the simple analytic problem. Another issue arises, when there are multiple roots for the root solving problem associated with the projection of the sample points on the limit state surface. For such a case, a branch and bound algorithm would be required to compute the multiple roots.

8.5 Recommendations for Future Work

Improvements to the current implementations of the methodologies presented in this dissertation have been suggested in the earlier sections. There are many possible extensions of this research. Some of the important future extensions of this work have been listed below:

1. MDO Methodologies for PMA and Uni-Level formulations: Both the PMA-based RBO and Uni-Level RBO, that were presented in Chapter 4, have certain advantages over a traditional RBO.

The PMA-based RBO is more robust to the presence of inactive hard constraints in the RBO problem than a traditional RBO. The extension of RBO-CSSO and RBO-CSSO2 methodologies for PMA formulation are fairly straightforward. The reliability analysis has to be replaced with the inverse reliability analysis in the rSA and the SSOs. In the approximation building phase, the approximation will have to be built for the worst case value of the hard constraint that is computed in a inverse reliability analysis in the PMA.

The Uni-Level RBO breaks the traditional bi-level FORM-based RBO to a single level optimization with both the design variables and Most Probable Points as optimization variables. It has been shown to be computationally more efficient than a traditional RBO. The main advantage of the Uni-Level RBO is that standard

MDO algorithms, that is used with traditional design optimization can be used. But the computational burden of requiring Hessian information of the hard constraints while implementing a Uni-Level RBO needs to be addressed. One can handle this by using Hessian update schemes that require only gradient information of the hard constraints at previous and current iteration.

2. Extension to discrete and block variables: The extension of RBO to discrete and block design variables can be done by incorporating discrete optimization techniques such as Simulated Annealing or Genetic Algorithms. This class of problems represent realistic engineering problems that can arise earlier in the design phase, where choices of the discrete or block variables have not been made. Such problems have not received much attention and hence provide a very promising area of research.

Such a problem can be solved in a discrete optimization phase followed by a continuous optimization phase. In the discrete optimization phase, the continuous design variables can be discretized, and block variables can be characterized by discrete values. Then simulated annealing or genetic algorithms can be used to obtain optimal values of discrete variables and block variables, and an approximate setting for the continuous design variables. A continuous optimization phase can be done after this to get optimal values of the continuous design variables with the discrete variables and block variables kept fixed at the optimal values obtained from previous phase. It has to be noted that evaluation of each design in both phases would require reliability analysis, that will make the problem extremely computationally intensive. The computational costs can be reduced by incorporating parallel or distributed computing in the discrete optimization phase.

3. Combination with other uncertainty types: The work presented here does not deal

with uncertainties with less detailed or incomplete statistical information and other uncertainties such as modeling uncertainties. Hence a hybrid RBO methodology can be developed that incorporates non-probabilistic theories such as possibility or evidence theory, fuzzy sets and convex models of uncertainty. It is very ambitious to include all uncertainty types in an RBO methodology but it is possible to include certain different types of uncertainties like interval uncertainties and random uncertainties.

A hybrid RBO that incorporates interval uncertainties and random uncertainties can use a worst case reliability analysis, where worst case reliability constraints are used in the RBO. Interval analysis will be required to obtain such worst case estimates. The use of rigorous interval analysis will increase the computational costs significantly. The use of linearization approximations to obtain worst case values would be cheaper and may be a sufficient alternative.

APPENDIX A

ROSENBLATT TRANSFORMATION AND ITS INVERSE

Given a continuous random vector, $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$, with a joint cumulative distribution function (c.d.f.), $F_{\mathbf{X}}(\mathbf{x})$, it is always possible to construct a transformation that will give independent, standard normal (Gaussian with 0 mean and unit variance) random variables, $\mathbf{U} = \{U_1, U_2, \dots, U_n\}$. For a general case, it is possible to have more than one transformation. One such transformation is given by the following equations:

$$U_1 = \Phi^{-1}(F_{X_1}(X_1)) \quad (\text{A.1})$$

$$U_j = \Phi^{-1}\left(F_{X_j|X_1, X_2, \dots, X_{j-1}}(X_n|X_1, X_2, \dots, X_{j-1})\right) \text{ for } j = 2..n \quad (\text{A.2})$$

where F_{X_1} is the marginal c.d.f. of X_1 , $F_{X_2|X_1}$ is the conditional c.d.f. of X_2 for a fixed value of X_1 , $F_{X_3|X_1, X_2}$ is the conditional c.d.f. of X_3 for fixed values of X_1 and X_2 , so on. It has to be noted that if the components of \mathbf{X} are all statistically independent then the conditional c.d.f.s equal marginal c.d.f.s. This transformation is a well known result in probability theory, but one of the papers by Rosenblatt, that contained this result, was first cited in one of the early works in the field of structural reliability and ever since has been known as the Rosenblatt transformation.

The transformation from \mathbf{x} to \mathbf{u} involves the computation of Φ^{-1} , which can be performed using numerical procedures. The Statistics toolbox in Matlab was used for this purpose in this work. It has to be noted that the transformations are analytic for cases when \mathbf{X} is normally or lognormally distributed. The inverse transformation from \mathbf{U} to \mathbf{X}

is given by the following equation:

$$X_1 = F_{X_1}^{-1}(\Phi(U_1)) \quad (\text{A.3})$$

$$X_j = F_{X_j|X_1, \dots, X_{j-1}}^{-1}(\Phi(U_j)) \text{ for } j = 2..n \quad (\text{A.4})$$

The inverse transformation for a general \mathbf{X} can be numerically computed at a given instantiation of \mathbf{U} , \mathbf{u} , as illustrated by the following algorithm, starting with $j = 0$.

1. Set $j = j + 1$.
2. If $j = 1$ then solve $\Phi(u_1) - F_{X_1}(x_1) = 0$ for x_1 using a root solving approach and go to step 4., else go to step 3.
3. Solve $\Phi(u_j) - F_{X_j|X_1, X_2, \dots, X_{j-1}}(x_j|x_1, x_2, \dots, x_{j-1}) = 0$ for x_j using a root solving approach, with x_1, x_2, \dots, x_{j-1} fixed at values that are obtained from previous values of j .
4. If $j < n$ then go to 1. else exit.

APPENDIX B

HL-RF FAMILY OF ALGORITHMS

B.1 Basic Algorithm

The MPP formulation is given by the following equation:

$$\min_{\mathbf{u}} \quad \frac{1}{2} \mathbf{u}^T \mathbf{u} \quad (\text{B.1})$$

$$\text{s.t.} \quad g(\mathbf{u}, \boldsymbol{\eta}) = 0 \quad (\text{B.2})$$

The family of Hasofer-Lind, Rackwitz-Fiessler (HL-RF) algorithms are based on successive linearization of $g(\mathbf{u}, \boldsymbol{\eta})$. The basic steps of the HL-RF algorithms, starting at $k = 0$ from a chosen \mathbf{u}^k , are as follows.

1. Find the step \mathbf{s}^k corresponding to a solution for \mathbf{u}' of minimum norm, that solves $g(\mathbf{u}^k) + \nabla_u g[\mathbf{u}' - \mathbf{u}^k] = 0$. This basically corresponds to the solution of Equations B.1-B.2 with linearized version of constraint replacing Equation B.2. The step is given by $\mathbf{s}^k = \mathbf{u}' - \mathbf{u}^k$, which can be computed using the minimum norm solution of \mathbf{u}' given by the following equation:

$$\mathbf{u}' = \left[\frac{-\nabla_u g}{\|\nabla_u g\|} \mathbf{u}^k + \frac{g(\mathbf{u}^k)}{\|\nabla_u g\|} \right] \frac{-\nabla_u g^T}{\|\nabla_u g\|} \quad (\text{B.3})$$

2. Find the update $\mathbf{u}^{k+1} = \mathbf{u}^k + \zeta \mathbf{s}^k$, where ζ is the step-size obtained by a line search method that guarantees a sufficient decrease in a merit, $m(\mathbf{u}, \rho)$. In the HL-RF algorithm, no line search is needed, hence $\zeta = 1$. In modified HL-RF (mHL-RF) and improved HL-RF (iHL-RF) algorithms, line searches are implemented. The

merit function for mHL-RF is constructed from the first order optimality conditions of Equations B.1-B.2, and is given by Equation B.4. The merit function of iHL-RF is based on an exact penalty function and is given by Equation B.5.

$$m(\mathbf{u}, \rho) = \frac{1}{2} \left\| \mathbf{u} - \frac{\nabla_u g \mathbf{u}}{\|\nabla_u g\|^2} \nabla_u g \right\|^2 + \frac{1}{2} \rho g(\mathbf{u})^2 \quad (\text{B.4})$$

$$m(\mathbf{u}, \rho) = \frac{1}{2} \mathbf{u}^T \mathbf{u} + \rho |g(\mathbf{u})| \quad (\text{B.5})$$

3. Convergence check is performed at \mathbf{u}^{k+1} . The convergence is achieved if the following convergence is satisfied:

$$\left(|g(\mathbf{u}^{k+1})| < \varepsilon_1 \right) \text{ AND } \left[\left(\max |\mathbf{s}^k| < 2\varepsilon_2 \right) \text{ OR } \left(|(\mathbf{u}^{k+1})^T \mathbf{s}^k| < 2\varepsilon_3 \right) \right] \quad (\text{B.6})$$

where ε_1 , ε_2 and ε_3 are preselected tolerances. If the convergence is not achieved, $k = k + 1$ go to Step 1, else exit.

The HL-RF algorithm fails to converge for many even well-scaled problems. Since it shares similarities with Newton-Raphson approach, like some of the problems associated with Newton-Raphson like cycling of iterates that occurs in this method[98]. The mHL-RF and iHL-RF algorithms control the step size to alleviate some of these problems. The mHL-RF algorithm is known to work well but the convergence cannot be proved, whereas the iHL-RF algorithm is globally convergent. It has to be noted that all the three methods can fail for badly scaled problems because all these methods are based on linearization of g . In this work, iHL-RF algorithm was used in many applications. Hence some of implementation details of this algorithm are discussed in the following section.

B.2 Implementation of iHL-RF

The iHL-RF is chosen over mHL-RF because the iHL-RF is known to be a globally convergent algorithm, and the merit in iHL-RF does not involve the gradients of g like

mHL-RF and hence is cheaper than mHL-RF. The basic steps of the iHL-RF algorithm is same except for the following additional steps:

1. At $k = 0$ set parameters $a, b \in (0, 1)$, $\delta > 0$, $\gamma > 0$ and $\epsilon_{tol} > 0$. $a = 0.1$, $b = 0.5$, $\delta = 10^{-6}$, $\gamma = 2$ and $\epsilon_{tol} = 10^{-9}$ was chosen for all the application and test problems.
2. After the computation of \mathbf{s}^k , the penalty parameter is computed as follows:

$$\rho^k = \begin{cases} \gamma \max \left[\frac{\|\mathbf{u}^k\|}{\|\nabla_u g\|}, \frac{1}{2} \frac{\|\mathbf{u}^k + \mathbf{s}^k\|^2}{|g(\mathbf{u}^k)|} \right] & \text{If } |g(\mathbf{u}^k)| \geq \delta \\ \gamma \frac{\|\mathbf{u}^k\|}{\|\nabla_u g\|} & \text{Otherwise} \end{cases} \quad (\text{B.7})$$

3. If $(\nabla_u m(\mathbf{u}^k, \rho^k) \mathbf{s}^k) \leq \epsilon_{tol}$ then $\zeta = 1$ and exit, else compute the step-size ζ using a line search based on Armijo's rule, which can be stated as follows:

$$\zeta = \max_j \{ [b]^j |m(\mathbf{u}^k + [b]^j \mathbf{s}^k, \rho^k) - m(\mathbf{u}^k, \rho^k) \leq -a [b]^j (\nabla_u m(\mathbf{u}^k, \rho^k) \mathbf{s}^k)\} \quad (\text{B.8})$$

where $[b]^j = b \times b \times .. (j \text{ times})$. j is initially set to 0 and then j is increased by 1 till the condition in Equation B.8 is satisfied. Initial negative values of j can also be set but in the application problems the initial value was set as zero.

4. Go to step 3. of basic HL-RF algorithm.

APPENDIX C

CONTROL AUGMENTED STRUCTURES

In this appendix, the details of the system analysis and the sensitivity analysis for the control augmented structures application problem are presented.

C.1 System Analysis

The system analysis for the control augmented structures problem consists of two contributing analysis, CA_1 and CA_2 from a structures subsystem and a controls subsystem respectively. The functional forms of CA_1 and CA_2 are given below:

$$[W_{st}, \mathbf{g}_{st}, \mathbf{y}_{st}] = CA_1(\mathbf{b}, \mathbf{h}, \mathbf{p}_{st}, m_{con}) \quad (C.1)$$

$$[\mathbf{g}_{con}, m_{con}] = CA_2(c, \mathbf{p}_{con}, \mathbf{y}_{st}) \quad (C.2)$$

where W_{st} is the structures weight, \mathbf{g}_{st} and \mathbf{g}_{con} comprise of the design constraints from the structures and controls analysis respectively, \mathbf{p}_{st} and \mathbf{p}_{con} comprise of fixed parameters from the structures and controls analysis respectively, \mathbf{y}_{st} and \mathbf{y}_{con} comprise of state variables from the structures and controls analysis respectively, \mathbf{b} and \mathbf{h} are the breadths and heights of the elements of the beam, c is a damping parameter, m_{con} is the mass of the controllers. The merit function is given by $f = W_{st} + 32.2m_{con}$.

The SA was performed as a univariate root search, with respect to m_{con} , using Matlab's FZERO subroutine. The aim of the root search is to find roots of $m'_{con} - m_{con} = 0$, where m_{con} is the optimization variable varied by the root search algorithm and the m'_{con} is

obtained from one feed-forward from CA_1 to CA_2 that is shown the following equations:

$$[W'_{st}, \mathbf{g}'_{st}, \mathbf{y}'_{st}] = CA_1(\mathbf{b}, \mathbf{h}, \mathbf{p}_{st}, m_{con}) \quad (C.3)$$

$$[\mathbf{g}'_{con}, m'_{con}] = CA_2(c, \mathbf{p}_{con}, \mathbf{y}'_{st}) \quad (C.4)$$

It was required that the absolute change in m_{con} be less than 10^{-10} for the root search algorithm to converge. The initial guess for m_{con} was taken as 3. Now, the details of the structures and the control subsystem will be presented.

C.1.1 Structures Subsystem

The CA_1 is based on Finite Element Method. Elemental Stiffness Matrix for a standard beam element and a lumped Mass Matrix was used for each element. The controls mass was added as a lumped mass on the first element. These elemental stiffness and mass matrices were assembled to obtain global stiffness matrix and mass matrix. The sizes of the global matrices were reduced by 2 d.o.f. by imposing zero displacements at the fixed end. The displacements, $\boldsymbol{\delta}$ were found by solving $\mathbf{K}\boldsymbol{\delta} = \mathbf{T}$.

The maximum axial stresses and shear stresses were computed at both ends of each element based on shear forces and bending moments acting at both ends of each element. The maximum principal stresses for both ends of each element are obtained. The stress constraints were formulated for these maximum principal stresses.

The first three eigenvalues, ω_1^2 , ω_2^2 and ω_3^2 , and the corresponding eigenvectors, $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$ and $\boldsymbol{\phi}_3$, were found for this problem, which were inputs to the controls module.

C.1.2 Controls Subsystem

A modal analysis is done in the controls subsystem model, where a modal transformation is done to decouple the dynamic system of equations given by Equation C.5 to

obtain the system of equations given by Equation C.6.

$$\mathbf{M}\ddot{\boldsymbol{\vartheta}} + \mathbf{C}\dot{\boldsymbol{\vartheta}} + \mathbf{K}\boldsymbol{\vartheta} = \mathbf{b}_u\mathbf{u} + \mathbf{b}_f f(t) \quad (\text{C.5})$$

$$\dot{\mathbf{r}} = \mathbf{A}\mathbf{r} + \mathbf{B}_u\mathbf{u} + \mathbf{B}_f f(t) \quad (\text{C.6})$$

where $\boldsymbol{\vartheta} = \Phi\mathbf{Y}$, $\mathbf{r} = [\mathbf{Y}, \dot{\mathbf{Y}}]^T$, $\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} \\ -\omega^2 & -c\omega^2 \end{bmatrix}$, $\mathbf{B}_u = \begin{bmatrix} 0 \\ \Phi^T \mathbf{b}_u \end{bmatrix}$, $\bar{\mathbf{B}} = \begin{bmatrix} 0 \\ \Phi^T \mathbf{b}_f \end{bmatrix}$, with $\boldsymbol{\vartheta}$ being the dynamic displacement vector, \mathbf{u} being the control input, Φ being the eigenvector matrix for the number of modes considered, $f(t)$ being the dynamic force, a ramp function of time in this problem and \mathbf{b}_u and \mathbf{b}_f being the coefficient matrices of the control input and the dynamic force respectively.

A gain matrix \mathbf{G} is found which gives an Optimal Linear Quadratic Regulator, $\mathbf{u} = -\mathbf{Gr}$, which gives the minimum performance measure $J = \int_0^\infty (\mathbf{r}^T \mathbf{Q} \mathbf{r} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt$, where \mathbf{Q} and \mathbf{R} are control weighting matrices. \mathbf{Q} and \mathbf{R} were set as diagonal matrices, $q\mathbf{I}$ and $r\mathbf{I}$ respectively $q = 1$ and $r = 10^{-4}$. It can be shown that $\mathbf{G} = \mathbf{R}^{-1} \mathbf{B} \mathbf{P}$ where \mathbf{P} is the solution of the Matrix Riccati Equation given by the following equation:

$$\mathbf{Q} + \mathbf{PA} + \mathbf{A}^T \mathbf{P} - \mathbf{PBR}^{-1} \mathbf{B}^T \mathbf{P} = 0 \quad (\text{C.7})$$

The equations of the motion in the transformed state vector are $\dot{\mathbf{r}} = \bar{\mathbf{A}}\mathbf{r} + \bar{\mathbf{B}}f(t)$ where $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{BG}$. The solution $\mathbf{r}(t)$ is given by the following equation:

$$\mathbf{r}(t) = e^{\bar{\mathbf{A}}t} \mathbf{r}(0) + \int_0^t e^{\bar{\mathbf{A}}(t-\tau)} \mathbf{B} f(\tau) d\tau \quad (\text{C.8})$$

The solution is obtained from 0 to 2 seconds, with zero initial conditions i.e. $\mathbf{r}(0) = 0$. In this example, $f(t) = kt$ where $k = 1000$. The solution can be obtain eigenvalue decomposition of the matrix, $\bar{\mathbf{A}}$. So let $\mathbf{VDV}^{-1} = \bar{\mathbf{A}}$ where \mathbf{V} is an orthogonal matrix and \mathbf{D} is a diagonal matrix. Then, $\mathbf{r}(t) = \mathbf{VE}(t)\mathbf{V}^{-1}\bar{\mathbf{B}}k$ where $\mathbf{E}(t) = -t\mathbf{D}^{-1} + \mathbf{D}^{-2}(e^{\mathbf{D}t} - \mathbf{I})$. At the end of 2 seconds, the dynamic displacement vector $\boldsymbol{\vartheta}(2)$ and controls input $\mathbf{u}(2)$

are found. The controls weight is then calculated as the $\sum_{i=1}^2 K_c |u_i(2)|^{\frac{1}{2}}$, where K_c is some constant, taken as 2.0.

C.2 Sensitivity Analysis

A mixture of analytic and finite difference sensitivities (mainly analytic) were used for computing the derivatives of the merit, constraints and state variables with respect to the design variables and parameters. Analytic sensitivities for the static displacements, eigenmodes and frequencies were used[95]. Finite difference sensitivities for the static stresses were used. Analytic sensitivities of the controls outputs were used[96]. The details are presented in following part of the section.

C.2.1 Structures Sensitivities

The analytic sensitivities for static displacements can be obtained by solving the following equation:

$$\mathbf{K} \frac{d\boldsymbol{\delta}}{d()} = \frac{\partial \mathbf{T}}{\partial()} - \frac{d\mathbf{K}}{d()} \boldsymbol{\delta} \quad (\text{C.9})$$

Finite difference sensitivities were used for the stress constraints.

The analytic sensitivities for natural frequencies and modes of the dynamic analysis can be obtained by solving the following system of equations:

$$\begin{bmatrix} \mathbf{K} - \omega^2 \mathbf{M} & -\mathbf{M}\boldsymbol{\phi} \\ -\boldsymbol{\phi}^T \mathbf{W} & 0 \end{bmatrix} \begin{Bmatrix} \frac{d\boldsymbol{\phi}}{d()} \\ \frac{d\omega^2}{d()} \end{Bmatrix} = \begin{Bmatrix} -\left(\frac{d\mathbf{K}}{d()} - \omega^2 \frac{d\mathbf{M}}{d()}\right) \boldsymbol{\phi} \\ \frac{1}{2} \mathbf{u}^T \frac{d\mathbf{W}}{d()} \mathbf{u} \end{Bmatrix} \quad (\text{C.10})$$

where \mathbf{W} is given by the normalization conditions, $\boldsymbol{\phi}^T \mathbf{W} \boldsymbol{\phi} = 1$ which is taken as identity matrix. The above system of equations cannot be directly solved because $\mathbf{K} - \omega^2 \mathbf{M}$ is singular. The sensitivities of frequencies can be obtained easily by premultiplying the

first equation in Equations C.10 by ϕ^T , and is given by the following equation:

$$\frac{d\omega^2}{d()} = \frac{\phi^T \left(\frac{d\mathbf{K}}{d()} - \omega^2 \frac{d\mathbf{M}}{d()} \right) \phi}{\phi^T \mathbf{M} \phi} \quad (\text{C.11})$$

The sensitivities of the eigenmodes are obtained using Nelson's Method. In Nelson's method, ϕ is re-normalized to $\bar{\phi}$, by scaling with the maximum component of ϕ , ϕ_m , where ϕ_m is the m^{th} component of ϕ . Hence $\phi = \phi_m \bar{\phi}$ and $\bar{\phi}_m = 1$. The sensitivity of the eigenmode is given by the following equation:

$$\frac{d\phi}{d()} = \frac{d\phi_m}{d()} \bar{\phi} + \phi_m \frac{d\bar{\phi}}{d()} \quad (\text{C.12})$$

By enforcing $\frac{d\bar{\phi}_m}{d()} = 0$ in the first equation of Equation C.10 with $\frac{d\phi}{d()}$ replaced by $\frac{d\bar{\phi}}{d()}$, one can solve for $\frac{d\bar{\phi}}{d()}$. By plugging Equation C.12 in the second equation of Equation C.10 one can obtain $\frac{d\phi_m}{d()}$ that is given by the following equation:

$$\frac{d\phi_m}{d()} = -\phi_m^2 \phi^T \mathbf{W} \frac{d\bar{\phi}}{d()} - \frac{\phi_m}{2} \mathbf{u}^T \frac{d\mathbf{W}}{d()} \mathbf{u} \quad (\text{C.13})$$

C.2.2 Controls Sensitivities

For obtaining the analytic sensitivities of the controls outputs, the sensitivities of $\bar{\mathbf{A}}$ are required, which depend on the sensitivities of the Riccati matrix. $\bar{\mathbf{A}}$ can be rewritten as, $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{X}\mathbf{P}$ where $\mathbf{X} = \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T$. The sensitivities of $\bar{\mathbf{A}}$ can be written as:

$$\frac{d\bar{\mathbf{A}}}{d()} = \frac{d\mathbf{A}}{d()} - \frac{d\mathbf{X}}{d()} \mathbf{P} - \mathbf{X} \frac{d\mathbf{P}}{d()} \quad (\text{C.14})$$

where the sensitivities of \mathbf{X} can be obtained from the sensitivities of \mathbf{B} and \mathbf{R} that depend on the eigenmodes and weighting factor r respectively. The sensitivities of \mathbf{P} is obtained from differentiating Equation C.7 which gives the following Lyapunov equation:

$$\bar{\mathbf{A}}^T \frac{d\mathbf{P}}{d()} + \frac{d\mathbf{P}}{d()} \bar{\mathbf{A}} = -\frac{d\bar{\mathbf{A}}^T}{d()} \mathbf{P} - \mathbf{P} \frac{d\bar{\mathbf{A}}}{d()} + \mathbf{P} \frac{d\mathbf{X}}{d()} \mathbf{P} \quad (\text{C.15})$$

The above equation is solved for $\frac{d\mathbf{P}}{d(\cdot)}$ using Matlab's LYAP subroutine. One can now obtain $\frac{d\bar{\mathbf{A}}}{d(\cdot)}$. Then one can differentiate Equation C.8 which gives the following equation:

$$\frac{d\mathbf{r}(t)}{d(\cdot)} = \int_0^t \frac{de^{\bar{\mathbf{A}}(t-\tau)}}{d(\cdot)} \mathbf{B} k \tau d\tau + \int_0^t e^{\bar{\mathbf{A}}(t-\tau)} \frac{d\mathbf{B}}{d(\cdot)} k \tau d\tau \quad (\text{C.16})$$

which can be rewritten as the following equation:

$$\frac{d\mathbf{r}(t)}{d(\cdot)} = \mathbf{V}\mathbf{F}(t)\mathbf{V}^{-1}\bar{\mathbf{B}}k + \mathbf{V}\mathbf{E}(t)\mathbf{V}^{-1}\frac{d\bar{\mathbf{B}}}{d(\cdot)}k \quad (\text{C.17})$$

where

$$[\mathbf{F}(t)]_{ij} = \begin{cases} \frac{[\mathbf{V}^{-1}\frac{d\bar{\mathbf{A}}}{d(\cdot)}\mathbf{V}]_{ij}}{[\mathbf{D}]_{jj}-[\mathbf{D}]_{ii}} ([\mathbf{E}(t)]_{jj}-[\mathbf{E}(t)]_{ii}) & \text{if } i \neq j \\ \frac{[\mathbf{V}^{-1}\frac{d\bar{\mathbf{A}}}{d(\cdot)}\mathbf{V}]_{ii}}{[\mathbf{D}]_{ii}} \left[\frac{t}{[\mathbf{D}]_{ii}} \left(e^{[\mathbf{D}]_{ii}t} - 1 \right) - 2[\mathbf{E}(t)]_{ii} \right] & \text{if } i = j \end{cases} \quad (\text{C.18})$$

From $\frac{d\mathbf{r}(t)}{d(\cdot)}$ one can obtain analytic sensitivities of the dynamic displacement constraints and the controls mass.

BIBLIOGRAPHY

- [1] V.B. Venkayya. Introduction: Historical perspective and future directions. In M.P. Kamat, editor, *Structural Optimization: Status and Promise*, volume 150 of *Progress in Astronautics and Aeronautics*, pages 1–10. American Institute of Aeronautics and Astronautics, Inc., 1993.
- [2] R.J. Balling and J. Sobiesczanski-Sobieski. Optimization of coupled systems: A critical overview of approaches. In *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 753–773, Panama City, FL., 1994. Paper No. AIAA-94-4330-CP.
- [3] N.M. Alexandrov and M.Y. Hussaini, editors. *Multidisciplinary Design Optimization - State of the Art*. SIAM, Philadelphia, PA, 1997.
- [4] M.S. Phadke. *Quality Engineering Using Robust Design*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [5] S.S. Rao. *Reliability-Based Design*. McGraw-Hill, Inc., 1992.
- [6] F. Moses. Problems and prospects of reliability based optimization. *Engineering Structures*, 19(4):293–301, 1997.
- [7] A. Haldar and S. Mahadevan. *Probability, Reliability and Statistical Methods in Engineering Design*. John Wiley & Sons, Inc., 2000.
- [8] O. Ditlevsen and H.O. Madsen. *Structural Reliability Methods*. John Wiley & Sons, Inc., 1996.
- [9] C.A. Cornell. Structural safety: Some historical evidence that it is a healthy adolescent. In T. Moan and M. Shinozuka, editors, *Structural Safety and Reliability, Development in Civil Engineering*, 4, pages 19–29. Elsevier, 1981.
- [10] R. Rackwitz. Reliability analysis - past, present and future. In *8th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, University of Notre Dame, Notre Dame, IN, 2000.
- [11] J.D. Sørensen and I. Enevoldsen. Sensitivity weaknesses in application of some statistical distributions in first order reliability methods. *Structural Safety*, 12:315–325, 1993.
- [12] J. Sobiesczanski-Sobieski. Sensitivity of complex, internally coupled systems. *AIAA Journal*, 28(1):153–160, Jan 1990.

- [13] G.V. Reklaitis, A. Ravindran, and K.M. Ragsdell. *Engineering Optimization : Methods and Applications*. John Wiley & Sons Inc., 1983.
- [14] J. Sobiesczanski-Sobieski. Optimization by decomposition: A step from hierachic to non-hierachic systems. In *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, Hampton, VA, Sep 28-30 1998. NASA Conference Publication 3031, Part 1.
- [15] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski. Multidisciplinary optimization methods for aircraft preliminary design. In *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 697–707, Panama City, FL, Sep 1994. Paper No. AIAA-94-4325-CP.
- [16] B. A. Wujek and J. E. Renaud. New adaptive move-limit management strategy for approximate optimization. *AIAA Journal*, 36(10):1911–1934, Oct. 1998.
- [17] J.E. Renaud and G.A. Gabriele. Sequential global approximation in non-hierachic system decomposition and optimization. In G.A. Gabriele, editor, *Advances in Design Automation*, pages 191–200, 1991. ASME Publication DE-Vol. 32-1.
- [18] J.F. Rodríguez, V. M. Pérez, D. Padmanabhan, and J.E. Renaud. Sequential approximate optimization using variable fidelity response surface approximations. *Structural and Multidisciplinary Optimization*, 22:24–34, 2001. Published by Springer-Verlag, Germany.
- [19] G.E.P. Box and N.R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., 1987.
- [20] V.M. Pérez, T.B. Apker, and Renaud J.E. Parallel processing in sequential approximate optimization. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHC/ASC Structural Dynamics, and Materials Conference*, Denver, CO, 2002. Paper No. AIAA-2002-1589.
- [21] N.M. Alexandrov and R.M. Lewis. Analytical and computational properties of distributed approaches to mdo. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Design & Optimization*, Long Beach, CA, 2000. Paper No. AIAA-2000-4718.
- [22] S. Parsons. *Qualitative Methods for Reasoning Under Uncertainty*. The MIT Press, 2001.
- [23] W.L. Oberkampf, S.M. DeLand, B.M. Rutherford, K.V. Diegert, and K.F. Alvin. Estimation of total uncertainty in modeling and simulation. Technical Report SAND2000-0824, Sandia National Laboratories, April 2000.
- [24] Y. Ben-Haim and Elishakoff. *Convex Models of Uncertainty in Applied Mechanics*. Studies in Applied Mechanics, 25. Elsevier Publications, 1990.
- [25] S.S. Rao and L. Berke. Analysis of uncertain structural systems using interval analysis. *AIAA Journal*, 35(4):727–735, Apr 1997.

- [26] V. Braibant, A. Oudshoorn, and C. Boyer. Analysis of uncertain structural systems using interval analysis. *AIAA Journal*, 35(4):1298–1303, Apr 1997.
- [27] D. Dubois and H Prade. *Possibility Theory: An Approach to Computer Processing of Uncertainty*. Plenum Press, NY, 1988.
- [28] M. Fedrizzi, J. Kacprzyk, and R.R. Yager. *Advances in Dempster-Shafer Theory of Evidence*. John Wiley & Sons, Inc., 1994.
- [29] C.T. Lin and C.S.G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall, 1996.
- [30] H.J. Zimmerman. *Fuzzy Set Theory and Its Applications*. Kluwer Academic Publishers, 2nd edition, 1991.
- [31] M. Sakawa. *Fuzzy Sets and Interactive Multi-objective Optimization*. Plenum Press, 1993.
- [32] E.K. Antonsson and K.N. Otto. Imprecision in engineering design. *Transactions of the ASME - Journal of Mechanical Design*, 117(B):25–32, 1995.
- [33] S.W. Law and E.K. Antonsson. Implementing the method of imprecision: An engineering design example. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, volume 1, pages 358–363, 1994.
- [34] X. Gu, J.E. Renaud, and S.M. Batill. An investigation of multidisciplinary design subject to uncertainty. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, Missouri, Sep 1998. Paper No. AIAA-98-4747.
- [35] K.N. Otto and E.K. Antonsson. Extensions to the taguchi method of product design. *Transactions of the ASME - Journal of Mechanical Design*, 115:5–13, March 1993.
- [36] K.N. Otto and E.K. Antonsson. Tuning parameters in engineering design. *Transactions of the ASME - Journal of Mechanical Design*, 115:14–19, March 1993.
- [37] E. Nikolaidis, R. Haftka, and R. Rosca. Comparison of probabilistic and possibility-based methods for design against catastrophic failure under uncertainty. Nsf report, University of Florida, Gainesville, FL, 1998.
- [38] G. Maglaras, E. Nikolaidis, R.T. Haftka, and H.H. Cudney. Analytical-experimental comparison of probabilistic methods and fuzzy set based methods for designing under uncertainty. *Structural Optimization*, 13:69–80, 1997.
- [39] S. Chen, E. Nikolaidis, H.H. Cudney, R. Rosca, and R.T. Haftka. Comparison of probabilistic and fuzzy set methods for designing under uncertainty. In *Proceedings of the 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, 1999.
- [40] H. Agarwal, J.E. Renaud, E. Preston, and D. Padmanabhan. Uncertainty quantification using evidence theory in multidisciplinary design optimization (accepted for publication). *Journal of Reliability Engineering*, 2003.

- [41] J. Su and J.E. Renaud. Automatic differentiation in robust optimization. *AIAA Journal*, 35(6):1072–1079, June 1997.
- [42] S. Sundaresan, K. Ishii, and D.R. Houser. A robust optimization procedure with variations on design variables and constraints. In *Proceedings of 1993 ASME Advances in Design Automation Conference*, volume 65-1, pages 379–386, Albuquerque, NM., Sep 1993.
- [43] J. Yu and K. Ishii. A robust optimization procedure with variations on design variables and constraints. In *Proceedings of 1993 ASME Advances in Design Automation Conference*, volume 65-1, pages 371–378, Albuquerque, NM., Sep 1993.
- [44] D. Padmanabhan and S.M. Batill. An iterative concurrent subspace robust design framework. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, Long Beach, CA, 2000. Paper No. AIAA 2000-4841.
- [45] S.M. Batill, M.A. Stelmack, and R.S. Sellar. Framework for multidisciplinary design based upon response-surface approximations. *AIAA Journal of Aircraft*, 36(1):287–297, 1999.
- [46] X. Gu and J.E. Renaud. Implicit uncertainty propagation for robust collaborative optimization. In *Proceedings of DETC'01, ASME 2001 Design Engineering Technical Conference and Computers and Informations in Engineering Conference*, Pittsburgh, PA, Sep 9-12 2001.
- [47] Y. Zhang and A. Der Kiureghian. Finite element reliability methods for inelastic structures. NSF Report UCB/SEMM-97/05, Dept. of Civil and Environmental Engg., Univ.of California, Berkeley, March 1997.
- [48] P-L. Liu and A. Der Kiureghian. Optimization algorithms for structural reliability. *Structural Safety*, 9:161–177, 1991.
- [49] K. Breitung. Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics*, 110(3):357–366, March 1984.
- [50] L. Tvedt. Distribution of quadratic forms in normal space-application to structural reliability. *Journal of Engineering Mechanics*, 116(6):1183–1197, 1990.
- [51] A. Der Kiureghian, H-Z. Lin, and S-J Hwang. Second-order reliability approximation. *Journal of Engineering Mechanics*, 113(8):1208–1225, August 1987.
- [52] S. Engelund and R. Rackwitz. A benchmark study on importance sampling techniques in structural reliability. *Structural Safety*, 12:255–276, 1993.
- [53] A. Karamchandani and C.A. Cornell. Sensitivity estimation within first and second order reliability methods. *Structural Safety*, 11:95–107, 1992.
- [54] A. Der Kiureghian and P.L. Liu. Structural reliability under incomplete probability information. *Journal of Engineering Mechanics*, 112(1):85–104, Jan 1986.
- [55] I. Enevoldsen and J.D. Sørensen. Reliability-based optimization in structural engineering. *Structural Safety*, 15(3):169–196, 1994.

- [56] D.M. Frangopol and F. Moses. Reliability-based structural optimization. In H. Adeli, editor, *Advances in Design Automation*, chapter 13, pages 492–570. Chapman and Hall, 1994.
- [57] D.M. Frangopol and K-Y. Lin. Reliability-based optimum design for minimum life-cycle cost. In D.M. Frangopol and F.Y. Cheng, editors, *Advances in Structural Optimization*, pages 67–78. American Society of Civil Engineers, 1996.
- [58] M.L. Shooman. *Probabilistic Reliability: An Engineering Approach*. Robert E. Krieger Publishing Company, Malabar, Florida, 2nd edition, 1990.
- [59] I. Enevoldsen and J.D. Sørensen. Reliability-based optimization of series systems of parallel systems. *Journal of Structural Engineering*, 119(4):1069–1084, April 1993.
- [60] P. Thoft-Christensen and Y. Murotsu. *Application of Structural Systems Reliability Theory*. Springer-Verlag, 1986.
- [61] Y. Pu, P.K. Das, and D. Faulkner. A strategy for reliability-based optimization. *Engineering Structures*, 19(3):276–282, 1997.
- [62] Y. Murotsu, S. Shao, and A Watanabe. An approach to reliability-based optimization of redundant structures. *Structural Safety*, 16:133–143, 1994.
- [63] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.
- [64] J. Tu and K.K. Choi. A performance measure approach in reliability-based structural optimization. Technical Report R97-02, Center for Computer Aided Design, Univ. of Iowa, 1997.
- [65] J. Tu and K.K. Choi. A new study on reliability-based design optimization. *Journal of Mechanical Design, Transactions of the ASME*, 121:557–564, Dec 1999.
- [66] J.O. Royset, A. Der Kiureghian, and E. Polak. Reliability-based optimal design of series structural systems. *Journal of Engineering Mechanics*, 127(6):607–614, June 2001.
- [67] J.O. Royset, A. Der Kiureghian, and E. Polak. Reliability-based optimal structural design by the decoupling approach. *Reliability Engineering and System Safety*, 73:213–221, 2001.
- [68] C. Kirjner-Neto, E. Polak, and A. Der Kiureghian. An outer approximations approach to reliability-based optimal design of structures. *Journal of Optimization Theory and Applications*, 98(1):1–16, July 1998.
- [69] L. Wang and S. Kodiyalam. An efficient method for probabilistic and robust design with non-normal distributions. In *Proceedings of the 43rd AIAA/ASCE/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, April 2002.

- [70] P. Koch and S. Kadiyalam. Variable complexity structural reliability analysis for efficient reliability-based design optimization. In *40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, pages 68–77, 1999. Paper No. AIAA-99-1210.
- [71] R.V. Grandhi and L. Wang. Reliability-based structural optimization using improved two-point adaptive nonlinear approximations. *Finite Elements in Analysis and Design*, 29, 1998.
- [72] L. Wang and R.V. Grandhi. Improved two-point function approximations for design optimization. *AIAA Journal*, 33(9):1720–1727, Sep 1995.
- [73] L. Wang, R.V. Grandhi, and D.A. Hopkins. Structural reliability optimization using an efficient safety index calculation procedure. *International Journal for Numerical Methods in Engineering*, 38:1721–1738, 1995.
- [74] S.V.L. Chandu and R.V. Grandhi. General purpose procedure for reliability based structural optimization under parametric uncertainties. *Advances in Engineering Software*, 23:7–14, 1995.
- [75] M. Rajashekhar and B. Ellingwood. A new look at the response surface approach for structural reliability analysis. *Structural Safety*, 12:205–220, 1993.
- [76] D.R. Oakley, R.H. Sues, and G.S. Rhodes. Performance optimization of multidisciplinary mechanical systems subject to uncertainties. *Probabilistic Engineering Mechanics*, 13(1):15–26, 1998.
- [77] R.A. Canfield. Multipoint quadratic approximation for numerical optimization. In *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 971–977, Panama City, Florida, Sep 1994.
- [78] R.A. Canfield. A rank two hessian matrix update for sequential quadratic approximation. In *Proceedings of the 36th AIAA/ASME/ASCE/AHC/ASC Structural Dynamics, and Materials Conference and AIAA/ASME Adaptive Structures Forum*, pages 1486–1493, New Orleans, LA, Apr 1995.
- [79] D. Padmanabhan and S.M. Batill. Reliability based optimization using approximations with applications to multi-disciplinary system design. In *40th AIAA Sciences Meeting and Exhibit*, Reno, NV, Jan 2002. Paper No. AIAA 2002-0449.
- [80] R.H. Sues and M.A. Cesare. An innovative framework for reliability-based mdo. In *Proceedings of the 41st AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*, Atlanta, GA, April 2000. Paper No. AIAA 2000-1509.
- [81] G. Strang. *Linear Algebra and its Applications*. Harcourt Brace Jovanovich, Inc., 3rd edition, 1988.
- [82] D. Padmanabhan and S.M. Batill. Decomposition strategies for reliability based optimization in multidisciplinary system design. In *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, Atlanta, GA, Sep 2002. Paper No. AIAA 2002-5471.

- [83] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. Mathematical Programming Society and Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [84] J.E. Dennis, Jr., M. El-Alem, and M.C. Maciel. A global convergence theory for general trust-region-based algorithms for equality constrained optimization. *SIAM Journal of Optimization*, 7(1):177–207, Feb 1997.
- [85] M. Marazzi and Jorge Nocedal. Feasibility control in nonlinear optimization. In A. De Vore, A. Iserles, and E. Suli, editors, *Foundations of Computational Mathematics*, London Mathematical Society Lecture Series 284, pages 125–154. Cambridge University Press, 2001.
- [86] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal of Optimization*, 6(2):418–445, May 1996.
- [87] Indraneel Das. An interior point algorithm for the general nonlinear programming problem with trust region globalizaion. Technical Report TR 96-17, Dept. of Computational and Applied Mathematics, Rice University, July 1996.
- [88] A. Neumaier. Minq - general definite and bound constrained indefinite quadratic programming. WWW-Document, 1998. <http://solon.cma.univie.ac.at/neum/software/minq/>.
- [89] P.E. Gill, W. Murray, and M.A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. Technical Report 97-2, Dept. of Mathematics, University of California, San Diego, 1997.
- [90] P. Spellucci. A new technique for inconsistent qp problems in the sqp method. *Mathematical Methods of Operations Research*, 47, 1998.
- [91] D. Padmanabhan, R.V. Tappeta, and S.M. Batill. Monte carlo simulation in reliability-based optimization applied to multidisciplinary system design. In *Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*, Norfolk, VA, Apr 2003. Paper No. AIAA 2003-1503.
- [92] P. Bjerager. Methods for structural reliability computations. Technical report, University of California, Berkeley, April 27-29 1989. Lecture Notes for the course "Structural Reliability: Methods and Applications".
- [93] M.A. Maes, K. Breitung, and D.J. Dupuis. Asymptotic importance sampling. *Structural Safety*, 12:167–186, 1993.
- [94] J. Sobiesczanski-Sobieski, C.L. Bloebaum, and P. Hajela. Sensitivity of control-augmented structure obtained by a system decomposition method. *AIAA Journal*, 29(2):264–270, Feb 1990.
- [95] R.T. Haftka, Z. Gürdal, and M.P. Kamat. *Elements of Structural Optimization, Solid Mechanics and Its Applications*, volume 1. Kluwer Academic Publications, 2nd revised edition, 1990.

- [96] N.S. Khot. Optimization of controlled structures. In H. Adeli, editor, *Advances in Design Automation*, chapter 8, pages 266–296. Chapman and Hall, 1994.
- [97] B.A. Wujek. *Automation Enhancements in Multidisciplinary Design Optimization*. PhD thesis, University of Notre Dame, Notre Dame, IN, 1997.
- [98] W.H. Press, S.A. Teukolsky, W.T. Vellerling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, chapter 9. Cambridge University Press, 2nd edition, 1992.