ABSTRACT

Title of Dissertation: PARAMETER SENSITIVITY MEASURES FOR SINGLE OBJECTIVE, MULTI-OBJECTIVE, AND FEASIBILITY ROBUST DESIGN OPTIMIZATION

Subroto Gunawan, Doctor of Philosophy, 2004

Dissertation directed by: Shapour Azarm, Professor

Department of Mechanical Engineering

Uncontrollable variations are unavoidable in engineering design. If ignored, such variations can seriously deteriorate performance of an optimum design. Robust optimization is an approach that optimizes performance of a design and at the same time reduces its sensitivity to variations. The literature reports on numerous robust optimization techniques. In general, these techniques have three main shortcomings: (i) they presume probability distributions for parameter variations, which might be invalid, (ii) they limit parameter variations to a small (linear) range, and (iii) they use gradient information of objective/constraint functions. These shortcomings severely restrict applications of the techniques reported in the literature.

The objective of this dissertation is to present a robust optimization method that addresses all of the above-mentioned shortcomings. In addition to being efficient, the robust optimization method of this dissertation is applicable to both single and multi-objective optimization problems.

There are two steps in our robust optimization method. In the first step, the method measures robustness for a design alternative. The robustness measure is developed based on a concept that associated with each design alternative there is a sensitivity region in parameter variation space that determines how much variation a design alternative can absorb. The larger the size of this region, the more robust the design. The size of the sensitivity region is estimated by a hyper-sphere, using a worst-case approach. The radius of this hyper-sphere is obtained by solving an inner optimization problem. By comparing this radius to an actual range of parameter variations, it is determined whether or not a design alternative is robust. This comparison is added, in the second step, as an additional constraint to the original optimization problem. An optimization technique is then used to solve this problem and find a robust optimum design solution.

As a demonstration, the robust optimization method is applied to numerous numerical and engineering examples. The results obtained are numerically analyzed and compared to nominal optimum designs, and to optimum designs obtained by a few well-known methods from the literature. The comparison study verifies that the solutions obtained by our method are indeed robust, and that the method is efficient.

PARAMETER SENSITIVITY MEASURES FOR SINGLE OBJECTIVE, MULTI-OBJECTIVE, AND FEASIBILITY ROBUST DESIGN OPTIMIZATION

by

Subroto Gunawan

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Professor Shapour Azarm, Advisor/Chair
Professor Amr Baz
Professor Roberto Celi
Assistant Professor Steven A. Gabriel
Associate Professor Jeffrey Herrmann
Associate Professor Peter Sandborn

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Dr. Shapour Azarm, for his guidance and support throughout my graduate study. I also would like to thank the committee members, Dr. Baz, Dr. Celi, Dr. Gabriel, Dr. Herrmann, and Dr. Sandborn, for their inputs and comments. Secondly, I would like to thank our sponsor, Dr. Ng from ONR, whose generous support made this research possible. Special thanks to Mr. Art Boyars from NSWC Indian Head who in numerous occasions helped and provided us with the information necessary to formulate the payload optimization problem. His comments and suggestions throughout the research have been invaluable as well. Thanks also to Dr. Magrab who provided support during the first year of my graduate study. The research work presented in this thesis was also supported in part by the National Science Foundation through Grant DMI 0200029. Such support does not constitute an endorsement by the funding agency of the opinions expressed in the paper.

Thirdly, I want to thank my colleagues, past and present, from the Design Decision Support Laboratory: Mr. Kumar Maddulapalli, Mr. Babak Besharati, Mr. Genzi Li, Dr. Ali Farhang-Mehr, Dr. Jin Wu, and others who have provided much valuable inputs and suggestions. I also want to thank my parents and my family for their continuous support during my study. Very special thanks to my girlfriend, Agnes Susianti, whose love and understanding helped me carry on during difficult times. Lastly, I wish to thank all my friends, Koswara, Franky, Andre, Steven, Ronald, and many others, for their support and encouragements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

FSR	Feasibility Sensitivity Region
FWCSR	Feasibility Worst Case Sensitivity Region
GA	Genetic Algorithm
MC	Monte Carlo simulation
MOGA	Multi Objective Genetic Algorithm
MORO	Multi Objective Robust Optimization
SR	Sensitivity Region
WCSR	Worst Case Sensitivity Region
$\Delta \boldsymbol{f}$	variations in objectives
$\Delta \boldsymbol{f}_0$	maximum acceptable variations in the objectives
$\Delta \mathbf{g}$	variations in constraints
$\Delta \mathbf{g}_0$	maximum acceptable increment in the constraints
$\Delta \mathbf{p}$	variations in design parameters
$\Delta \mathbf{p}_0$	known ranges of parameter variations
$\Delta \mathbf{p}_0^-$	lower bound of known parameter variation ranges
$\Delta \mathbf{p}_0^+$	upper bound of known parameter variation ranges
$\Delta \overline{\mathbf{p}}$	normalized design parameters
$\boldsymbol{f}$	design objectives
$\mathbf{g}$	inequality design constraints
G	number of design parameters
$\mathbf{h}$	equality design constraints

$\eta^{-/+}$	asymmetric robustness index
$\eta_0^{-/+}$	desired level of asymmetric objective robustness
η_0^{+}	desired level of one-sided objective robustness
η^{+}	one-sided robustness index
η	robustness index
η_0	desired level of objective robustness specified by designer
η_g	feasibility robustness index
$\eta_{g,0}$	desired level of feasibility robustness specified by designer
η_{min}	overall robustness index
J	number of inequality constraints
K	number of equality constraints
M	number of design objectives
N	number of design variables
$\mathbf{p}$	design parameters
$\mathbf{p}_0$	nominal value of design parameters (values used in the optimization)
P_s	probability of constraint satisfaction
$\overline{R}_f$	normalized WCSR radius
$R_f^{-/+}$	radius of asymmetrical WCSR
R_f^{+}	radius of one-sided WCSR
R	radius of the worst-case estimate of the SR and FSR intersection
R_f	radius of WCSR
R_g	radius of FWCSR

$\mathbf{S}_f$	sensitivity set
$\mathbf{S}_g$	feasibility sensitivity set
$\boldsymbol{x}$	design variables
$\boldsymbol{x}_0$	a particular instance of design variables

CHAPTER 1

INTRODUCTION

1.1. MOTIVATION AND OBJECTIVE

Back in the late 1970's, there was a tile manufacturer in Japan called Ina Tile Company. One day the company discovered that an uneven temperature profile of its kilns was causing unacceptable variations in the size of its manufactured tiles. An obvious way to solve the problem was to modify the kilns by adding thermocouples and temperature controllers to monitor and correct the malfunction. However, this modification would have been very expensive. Instead, the company chose to make an inexpensive modification to their tile design to reduce the sensitivity of the manufactured tiles to temperature variations. Using statistically designed experiments, they found that increasing the lime content of their clay-mix from 1% to 5% reduced the variations in their tile size by a factor of 10 (Leon et al., 1987).

Uncontrollable variations and noises are unavoidable in engineering design. Temperature variations, deviation of material properties from specifications, and dimensional tolerances of a design are just a few examples of uncontrollable parameter variations. When designing a system, these variations cannot and should not be ignored because they can seriously affect the performance of a design. As in the Ina Tile Company example above, one way to counter the effects of these variations is to try to reduce or eliminate the parameter variations themselves. However, this approach is usually very difficult to undertake and/or expensive to implement. Furthermore, it is quite possible that such variations will re-appear some other time in the future. A better

approach is to try to reduce the sensitivity of the design to the variations so that deteriorations caused by these variations are kept within an acceptable level.

Dr. Genichi Taguchi from Japan is commonly credited for introducing the idea of reducing the sensitivity of a design, a process he called parameter design. Since then, this "least-sensitive design" idea has been developed much further, and later the term "robust design" was coined to refer to a design alternative that is insensitive to parameter variations. With the introduction of design optimization into system design, it was not long before the idea of a robust and optimum design surfaced, and the concept of robust optimization became popular among researchers in the field. Following conventional terminologies, Parkinson et al. (1993) later introduced the term "objective robustness" and "feasibility robustness" to refer to robustness with respect to objective and constraint functions in an optimization problem, respectively.

Many robust optimization methods have been developed in the literature, as will be reviewed in detail in Chapter 2. However, the applicability of these methods is limited to optimization problems with small variations, and continuous and/or differentiable objective and constraint functions. In addition, these methods typically presume a certain form of probability distribution function of the uncertain parameters, and are applicable for single objective optimization problems only. The computational cost of these methods also often limits their application to relatively simple optimization problems.

Practically, real world optimization problems rarely exhibit the properties mentioned above. The functions involved in real world optimization problems are typically non-differentiable. The parameter variations of interest are very often large, beyond the validity of gradient estimation. Probability distribution of the uncertain parameters is also

generally not known, or is difficult and expensive to estimate accurately. Many problems have multiple objectives that need to be considered simultaneously. In our case study for example, the design of a payload for an Undersea Autonomous Vehicle (Chapter 6), the objective and constraint functions are discontinuous, the parameter variations are large, and the probability distribution of parameter variations is unknown. This problem also has multiple objectives, instead of just one objective, for which we want to find the robust optimum solutions. To make matter worse, in reality it is computationally expensive to compute the functions involved in the problem, so computational efficiency of the method used is important.

The overall objective of this dissertation is to develop an efficient robust optimization method for both single- and multi-objective design optimization problems to obtain optimum designs that are robust with respect to both objectives and constraints, without having to: (1) *presume a probability distribution of the parameter variations,* (2) *limit parameter variations to a small* (*linear*) *range, and* (3) *use the gradient information of objective/constraint functions.*

Before we continue, it is important to provide a distinction between the concept of robust optimization and Post Optimality Sensitivity Analysis (POSA) (e.g., Fiacco, 1983). Both concepts deal with the uncertainties and variabilities that exist in an optimization problem. However, POSA is a posteriori approach where it determines the sensitivity and stability of the solution due to variability *after* the optimization process is completed. The goal of POSA is to provide information to the designer regarding the behavior of the optimum solution and the active constraints if some of the parameters vary. It is hoped that based on this information, the designer can then take appropriate

measures to maintain the performance of the optimum design. It is essentially a passive approach to optimization under uncertainty. In contrast, robust optimization is an active approach in dealing with uncertainty whereas the sensitivity of the design is considered *during* the optimization process. The goal of robust optimization is not to inform the designers of how to guard the optimum design against variations, but rather to reduce the sensitivity of the optimum design obtained so that there is little need for the designer to devise corrective measures when the variabilities exist.

1.2. RESEARCH COMPONENTS

To achieve the overall objective, we developed a step-by-step approach for the research in this dissertation. We first developed three research components for different types of robustness. These research components are: (1) single objective robustness, (2) multi-objective robustness, and (3) feasibility robustness. Next, we combined these different types of robustness to obtain a robust optimization method that accounts for both objective and feasibility robustness. In the next three sub-sections, an overview and objective of each research component is given.

1.2.1. Research Component 1: Single Objective Robust Optimization

The first research component is concerned with variations in the objective value of an optimum design due to uncontrollable variations in the parameters. This so-called "objective robustness" of an optimum design is important because if its objective value changes significantly, then performance of the design can degrade so much that it may be deemed unsatisfactory. Objective robustness of an optimum design is especially

important if the design is part of a larger system because deviations in the design's performance could affect the rest of the system.

The objective of the first research component is to develop a novel robust optimization method for single objective optimization problems that can obtain an optimum design solution that is robust in terms of the objective function.

1.2.2. Research Component 2: Multi-Objective Robust Optimization

The second research component is concerned with performance variations of an optimum design when there are multiple objectives involved. Similar to the first research component, here we also look into the changes in the objective values of a design due to variations in the parameters. However, since there are now multiple objectives, we have to examine the performance variation of the design with respect to each objective, and then based on it, determine the overall robustness of the design.

The objective of the second research component is twofold. First, since the notion of a design that is optimum and robust for multiple objectives has not yet been defined in the literature, this research component aims to introduce and develop the concept of "multi-objective robustness" and "multi-objective robust optimality" of a design alternative. Second, this research component seeks to develop a novel method for robust optimization of a design in multi-objective optimization problems.

1.2.3. Research Component 3: Feasibility Robust Optimization

The third research component is concerned with the feasibility of an optimum design due to uncontrollable variations in parameters. Typically, an optimum solution to an

engineering optimization problem is a boundary optimum, i.e., at the optimum at least one of the constraints is active (Papalambros and Wilde, 2000). Because of this, if some of the problem's parameters vary, the optimum design may no longer be feasible. A design that is always feasible even if there are parameter variations is called "feasibly robust," and the method to obtain a feasibly robust solution is called "feasibility robust optimization" method.

The objective of the third research component is to develop a novel and efficient feasibility robust optimization method to obtain an optimum design that is always feasible regardless of parameter variations.

1.3. ASSUMPTIONS

In developing our robust optimization method, we make the following assumptions:

- For objective robust optimization, we assume that there exists a trade-off between objective values of a design, and its robustness. If such a trade-off does not exist, then an optimum design is also a robust design, and there is no need to conduct robust optimization.
- For feasibility robust optimization, we assume that the optimum is on (or near) the boundary of the feasible domain. If the optimum is well inside the feasible domain, then most likely that optimum is already feasibly robust, and we do not need to conduct robust optimization.
- We assume that the range of parameter variations is known a priori, and that they are symmetric. (Robust optimization with asymmetric parameter variations will be discussed briefly in Chapter 6.)

1.4. ORGANIZATION OF DISSERTATION

The rest of the dissertation is organized as follows. Chapter 2 gives the definitions of concepts and terminologies used throughout the dissertation, as well as a comprehensive review of related previous work in the literature. We develop the method for objective robust optimization of a single objective optimization problem in Chapter 3 (Research Component 1), and extend it to multi-objective problems in Chapter 4 (Research Component 2). In Chapter 5, we develop a method for feasibility robust optimization (Research Component 3). Chapter 6 presents our combined objective and feasibility robust optimization method. To demonstrate the applications of our method, several numerical and engineering examples are given in Chapters 3 through 6. Chapter 7 concludes the dissertation with some remarks as well as a discussion on the contributions of the dissertation and potential future research directions.

After reading this chapter and the next, we recommend that the readers continue with Chapter 3 first because it contains the bulk of our research results that will become the foundation of Chapters 4, 5 and 6. Chapters 4 and 5 may be read separately; however, Chapter 6 should be read after Chapters 3, 4, and 5.

Figure 1.1 shows the organization and flow of information in this dissertation.

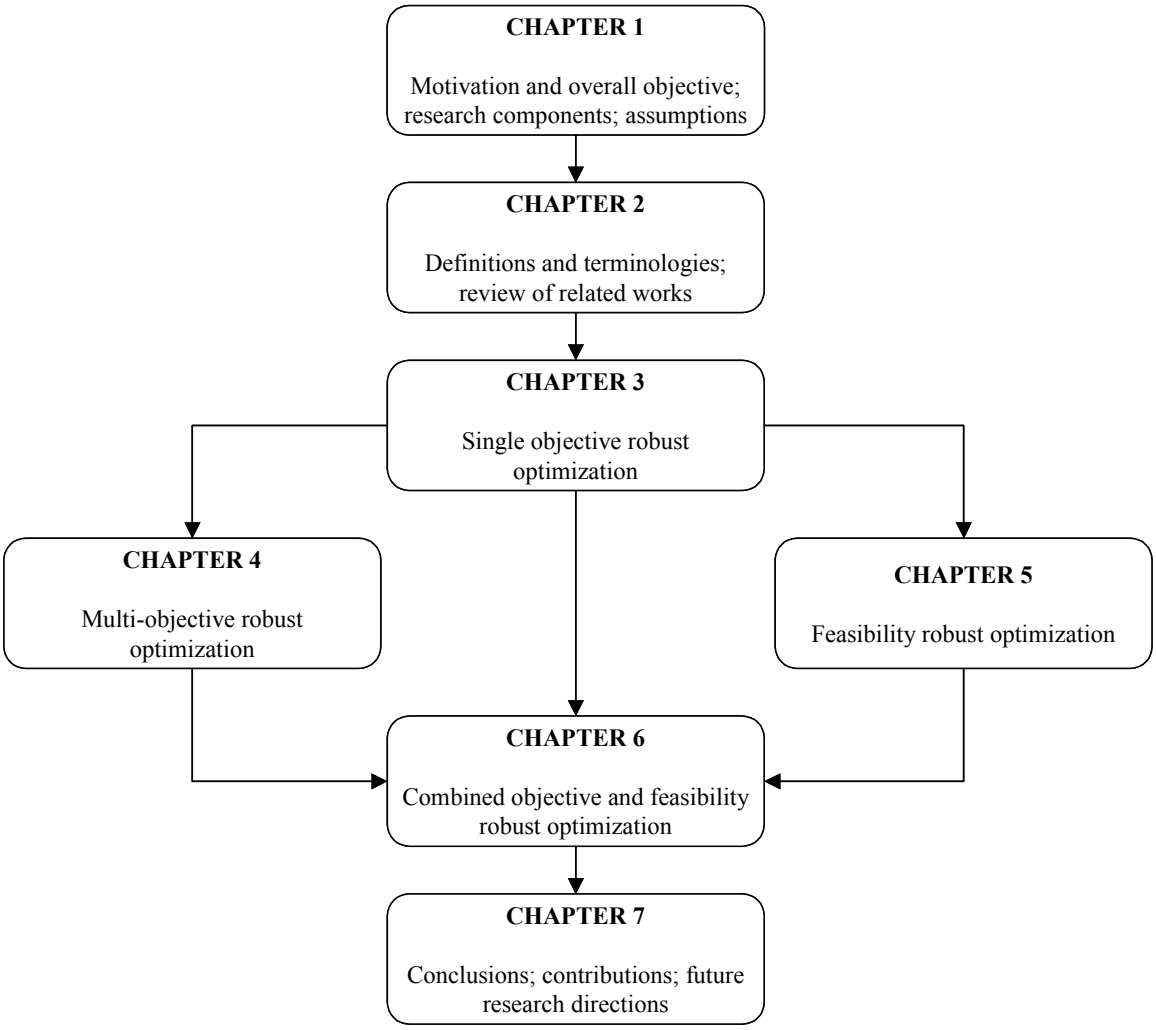


Figure 1.1: Organization of dissertation.

CHAPTER 2

DEFINITIONS AND PREVIOUS WORK

2.1. INTRODUCTION

In this chapter, we provide several definitions and terminologies that will be used throughout the dissertation. In addition, we also give a comprehensive review of previous work in the literature related to single objective, multi-objective, and feasibility robust optimization concepts and methods.

2.2. DEFINITIONS AND TERMINOLOGIES

A general single objective optimization problem can be formulated as shown in Eq. (2.1) below.

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x},\mathbf{p}) \\ \text{subject to}: \quad & \mathrm{g}_j(\boldsymbol{x},\mathbf{p}) \le 0 \quad ; \quad j = 1,...,\mathrm{J} \\ & \mathrm{h}_k(\boldsymbol{x},\mathbf{p}) = 0 \quad ; \quad k = 1,...,\mathrm{K} \end{aligned} \tag{2.1}$$

Here, f is the objective function to be optimized, $\boldsymbol{x} = [x_1,\ldots,x_N]^t$ is the design variable vector (with superscript 't' referring to the transpose of the row vector), and $\mathbf{p} = [p_1,\ldots,p_G]^t$ is the set of parameters. For practical reasons, $\boldsymbol{x}$ and $\mathbf{p}$ are restricted to real values. The problem has J inequality constraints, g_j, $j = 1,...,J$, and K equality constraints, h_k, $k = 1,\ldots,K$.

In this dissertation, the notation $\mathbf{p}$ represents problem factors that have variability, including design variables. Following this notation, if there are variations in some of the design variables, then a subset of $\boldsymbol{x}$ belongs to $\mathbf{p}$, i.e., these are the design factors that we control during the optimization of f but these factors have variability. Some researchers

prefer to differentiate between variations in $\boldsymbol{x}$ and variations in $\mathbf{p}$, the so-called type-1 and type-2 variations (Chen et al., 1996; Kalsi et al., 2001). For simplicity, we do not make that distinction. Also, the parameters value $\mathbf{p} = \mathbf{p}_0$ that we use to optimize a design is called the nominal parameters value.

Because of noise and uncertainty, parameter values vary by some amount: $\Delta\mathbf{p} = (\Delta p_1,\ldots,\Delta p_G)^t$, and in turn these variations affect the objective and constraint values of a design.

The goal of objective robust optimization is to obtain a design variable vector $\boldsymbol{x}_R$ whose objective value $f(\boldsymbol{x}_R,\mathbf{p})$ is not only minimum but also remains within an acceptable bound when $\mathbf{p}$ varies. In other words, the objective value of the design is insensitive to variations in $\mathbf{p}$.

The goal of feasibility robust optimization is to obtain a design variable vector $\boldsymbol{x}_R$ whose inequality constraint values $g_j(\boldsymbol{x}_R,\mathbf{p})$, j=1,..,J, are always feasible regardless of $\Delta\mathbf{p}$ variations. Feasibility robust optimization is concerned only with inequality constraints (i.e., equality constraints are not considered in feasibility robust optimization). This is because equality constraints are "hard" to consider, i.e., unless the $\Delta\mathbf{p}$ variations are such that $h_k(\boldsymbol{x},\mathbf{p}+\Delta\mathbf{p}) = 0$ for all k=1,…,K, there is no way to guarantee that these equality constraints will always be satisfied.

When there is more than one objective function to minimize, the problem becomes a multi-objective optimization problem as shown in Eq. (2.2).

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \boldsymbol{f}(\boldsymbol{x},\mathbf{p}) = \left[f_1, f_2,..., f_M\right] \\ \text{subject to}: \quad & g_j(\boldsymbol{x},\mathbf{p}) \le 0 \quad ; \quad j = 1,..., J \\ & h_k(\boldsymbol{x},\mathbf{p}) = 0 \quad ; \quad k = 1,..., K \end{aligned} \tag{2.2}$$

The formulation in Eq. (2.2) is the same as that of a single objective optimization problem, except that now we have $M > 1$ objectives to minimize simultaneously. Here, we assume that the number of objectives is finite (i.e., $M < \infty$). We also assume that at least two of the objectives are conflicting, i.e., for a given design, as we decrease (improve) the value of one objective, the value of at least one other objective increases (worsens). Because of this trade-off among the objectives, there is generally a set of optimum solutions to the problem in Eq. (2.2). This set is called a Pareto set (see definitions below), and the design solutions in the set are called Pareto designs. The Pareto set is a trade-off set meaning that there is no design in the set that dominates or is better than the other designs in the set. Extensive reviews of multi-objective optimization concepts and methods are given by Miettinen (1999), and in evolutionary multi-objective optimization by Deb (2001) and Coello Coello et al. (2002).

When there are variations in design parameters (i.e., $\Delta\mathbf{p}$), some or all of the M objectives will be affected. The goal of multi-objective robust optimization is to obtain a design variable vector $\boldsymbol{x}_R$ whose objective values $\boldsymbol{f}(\boldsymbol{x}_R,\mathbf{p})$ are Pareto optimum, and at the same time are insensitive to these $\Delta\mathbf{p}$ variations for all objectives.

Next, we provide several definitions and terminologies used in this dissertation.

Objective space ($\boldsymbol{f}$-space): An M-dimensional space in which the coordinate axes are the objective values.

Parameter variation space ($\Delta\mathbf{p}$-space): A G-dimensional space in which the coordinate axes are the parameter variation ($\Delta\mathbf{p}$) values.

Normalized parameter variation space ($\Delta\bar{\mathbf{p}}$-space): A G-dimensional $\Delta\mathbf{p}$-space where all the entries are normalized by the known ranges of parameter variations ($\Delta\mathbf{p}_0$), i.e.,

$$\Delta\bar{p}_i = \frac{\Delta p_i}{\Delta p_{i,0}}, \text{ for all } i=1,\ldots,G.$$

Maximum acceptable performance variation ($\Delta \boldsymbol{f}_0 = [\Delta f_{1,0},\ldots,\Delta f_{M,0}]^t$): The maximum acceptable change in the objective function values $\boldsymbol{f}$. These values are determined by the designer. The maximum acceptable change may also be given as a percentage value.

Inferiority, non-inferiority, and dominance: In multi-objective minimization, a feasible design point $\boldsymbol{x}_a$ is said to be inferior with respect to (w.r.t.) another feasible design point $\boldsymbol{x}_b$ if $f_i(\boldsymbol{x}_b) \leq f_i(\boldsymbol{x}_a)$ for all $i=1,\ldots,M$, with strict inequality for at least one i. Correspondingly, the design point $\boldsymbol{x}_b$ is said to dominate $\boldsymbol{x}_a$. If $\boldsymbol{x}_a$ neither dominates nor is inferior to $\boldsymbol{x}_b$, then $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are said to be non-inferior w.r.t. each other.

Inferior, non-inferior, and dominant regions: In multi-objective minimization, the inferior region of a design $\boldsymbol{x}_a$ is defined to be the region in the objective space where the design points are dominated by $\boldsymbol{x}_a$. Similarly, regions in the objective space where the design points in the regions are non-inferior and dominate $\boldsymbol{x}_a$ are called the non-inferior and dominant regions of $\boldsymbol{x}_a$, respectively.

Trade-off set: A set of design points is a trade-off set if all points in the set are non-inferior with respect to each other (although there might be design points outside the set that dominate the points in the set).

Pareto optimality, Pareto set, and Pareto frontier: In multi-objective optimization, a feasible design point $\boldsymbol{x}_p$ is Pareto optimum if it is not inferior w.r.t. any other feasible design point. The set of all Pareto optimum points is called the Pareto set. The plot of the Pareto set in the objective space is called the Pareto frontier.

Nominal Pareto set: Pareto set of a multi-objective optimization problem without robustness consideration.

Robust Pareto set: A trade-off set whose elements are both multi-objectively robust and Pareto optimum (see Chapter 4 for further explanations).

Vector operator $\otimes$: Let **a** and **b** be nx1 vectors. We define the operation: $\mathbf{c} = \mathbf{a} \otimes \mathbf{b}$, where **c** is a nx1 vector whose elements are: $\mathbf{c} = (a_1 b_1, a_2 b_2, \ldots, a_n b_n)^t$.

Ternary vector operator $\langle \bullet \rangle$: Let **a**, **b**, and **c** be nx1 vectors. We define the operation: $\mathbf{d} = \langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle$, where **d** is a nx1 vector whose *i*-th element is $d_i = \begin{cases} a_i b_i, & \text{if } a_i \le 0 \\ a_i c_i, & \text{if } a_i > 0 \end{cases}$ for all $i=1,\ldots,n$.

2.3. OVERVIEW OF PREVIOUS WORK

Most of the robust optimization methods in the literature are developed to account for both objective and feasibility robustness of an optimum design. However, some methods are developed to account for objective robustness only, while others are for feasibility robustness only. There are also some related works that developed methods for robustness measurement of a design only, without optimization.

In general, these methods can be categorized into three main groups: (i) experiment-based methods, (ii) deterministic methods, and (iii) probabilistic methods. Experiment-based methods are those methods that perform local sampling around the nominal value of a design to probe its behavior under parameter variations. Taguchi's orthogonal array (Taguchi and Phadke, 1984; Kackar, 1985; Phadke, 1989) and simple random sampling (Branke, 1998, 2001) are examples of experiment-based methods. Deterministic methods

are those methods that do not use statistical measures in calculating a design's robustness; rather, they use some deterministic measures. Gradient minimization (Belegundu and Zhang, 1992), worst-case analysis (Parkinson et al., 1993), and the mini-max method (Hirokawa and Fujita, 2002) are examples of deterministic methods. Probabilistic methods are those methods that use statistical measures, such as mean and variance, to calculate a design's robustness. These methods often use a Taylor series expansion to estimate these statistics. Examples of probabilistic methods include Yu and Ishii's variation pattern method (1994, 1998), reliability index method (Tu et al., 1999; Youn et al., 2003), and most probable point method (Du and Chen, 2000).

In addition to these three main groups, there are other more specialized robust optimization methods such as fuzzy robustness method (Otto and Antonsson, 1993; Arakawa and Yamakawa, 1998), tolerance maximization method (Balling et al., 1998), and Zhu and Ting's performance sensitivity distribution method (2001). There are also methods that combine several of the methods from the above three main groups.

Experiment-based methods are generally simple and straightforward, but their computational efforts grow rapidly, and eventually become impractical for problems with many parameters because they are essentially based on exhaustive permutations of all possible parameter variations. The computational cost of these methods is even more prohibitive as the number of discrete variation levels to be analyzed becomes large. Even for methods that use only partial permutations (Branke, 2001) the computational cost is still very high. Because of this computational issue, often these methods require preliminary experiments to eliminate those parameters that are statistically insignificant,

and to determine the levels of variations to be analyzed. These preliminary experiments are often costly and difficult.

Many deterministic methods (Belegundu and Zhang, 1992; Parkinson et al., 1993) try to reduce the computational effort by using gradient information to estimate a design's robustness. Estimating the gradient of a function is indeed computationally more efficient than exhaustive permutations. However, since these methods need gradient information, obviously they are only applicable to optimization problems whose functions are differentiable. These methods cannot solve robust optimization problems having non-smooth objective and/or constraint functions (e.g., a step function). Besides, as parameter variations grow large (beyond the range in which linear approximation is valid), gradient estimation will cease to be valid.

Some deterministic methods use worst-case analysis to calculate a design's robustness (Badhrinath and Rao, 1994; Hirokawa and Fujita, 2002). These methods are also computationally more efficient than experiment-based methods. However, the results obtained are typically conservative because they use the worst possible instance of a design's performance as its robustness measure.

Probabilistic methods (Yu and Ishii, 1994, 1998; Tu et al., 1999) extend the experiment-based methods by calculating probability information of a design based on a probability distribution of the parameters. For objective robustness, they calculate the mean and variance of a design's performance. For feasibility robustness, they calculate the probability of constraint satisfaction for a design. Clearly, these methods require that the probability distribution of parameters is known a priori (which often is not the case), and the results obtained from these methods are dependent on the validity of the assumed

distribution. The computational cost of these methods is also more prohibitive than experiment-based methods because they need to calculate probability information. Even for those probabilistic methods that claim to be efficient (Du and Chen, 2000), the number of function evaluations needed is still very high.

Almost all of the robust optimization methods in the literature are only applicable to single objective optimization problems. It is widely acknowledged, however, that an engineering design problem generally has multiple conflicting objectives. Very few papers address the issue of multiple objectives (sometimes also called multi-criteria): Rao (1984), Pignatiello (1993), and Ramakrishnan and Rao (1996). However, these methods essentially convert a multi-objective robust optimization problem into a single-objective one by aggregating the performance variations, and do not take into account trade-offs among the solutions. To our knowledge, there is no reported work in the literature that has formulated and defined the concept of a robust and multi-objectively optimum design as will be developed in this dissertation.

A more detail discussion of related works in the literature is given in the next three sections. For clarity, we divide our discussions according to our research components: (i) single objective robust optimization, (ii) multi-objective robust optimization, and (iii) feasibility robust optimization.

2.3.1. Single Objective Robust Optimization

One of the earliest works in objective robustness is the parameter design method of Taguchi (Taguchi, 1978; Taguchi and Phadke, 1984; Kackar, 1985; Phadke, 1989). This method is an experiment-based method that uses full-factorial experiments to obtain the

responses of a design, calculates the mean and variance of the responses, and then based on these values, minimizes a quantity called signal-to-noise (S/N) ratio. Taguchi's method has been widely used to obtain a robust design, e.g., Pignatiello and Ramberg (1985), Wang et al. (1999), Hwang et al. (2001). However, it also has received much criticism for its use of the S/N ratio because it could result in a design with very low performance or very high variance. Leon et al. (1987) proposed an alternative to S/N ratio, and developed a robustness measure called PerMIA (Performance Measure Independent of Adjustment). They showed that PerMIA is a quality loss measure that Taguchi originally proposed in his parameter design concept, but did not use in his formulation. They also showed that PerMIA is a more reliable measure than S/N ratio in terms of solution quality, and that for certain special cases, PerMIA simplifies to the S/N ratio. The use of PerMIA in factorial experiments is also proposed and discussed in Pignatiello and Ramberg (1987) and Box (1988).

In a different approach to Taguchi's experiment-based method, some methods calculate a design's robustness deterministically. Many methods (e.g., Belegundu and Zhang, 1992) use the gradient of the objective function as a robustness measure, and minimize a weighted sum (or some other combinations) of the objective and gradient values. However, Badhrinath and Rao (1994) showed that in general this weighted-sum approach is not reliable because it could lead to a local maximum solution. Instead they proposed a worst-case approach where they minimize the maximum objective value within the given parameter range. Parkinson (1998, 2000) and Hirokawa and Fujita (2002) also developed methods based on this worst-case min-max strategy. However, this min-max approach is often computationally extensive because it calculates the maximum

objective value every time a new feasible solution is obtained. To reduce computation cost, Sundaresan et al. (1992, 1993) proposed to simply calculate objective values at the "corners" of parameter range, and used their average as a robustness measure.

Balling et al. (1986) proposed an interesting deterministic method for robust optimization. Unlike other methods, their method works "backward" in that they first specify an acceptable variation in objective value, and then maximize the parameter range. Zhu and Ting (2001) also proposed a similar "backward" approach where they define relative robustness of a design based on the relationship between variation in the objective value and the parameter range corresponding to this variation.

A large portion of the literature uses probabilistic measures to determine robustness, most commonly the mean and variance of objective value. The simplest of these methods is the sampling approach where the mean and variance values are estimated by local sampling around the nominal objective value: Tsutsui and Gosh (1997), Branke (1998, 2001), Tsutsui (1999). Many probabilistic methods estimate the mean and variance using a Taylor series expansion, and then minimize a weighted sum of the two values: Yu and Ishii (1994, 1998), Du and Chen (2000), Jung and Lee (2002). Many other methods propose to treat mean and variance as two conflicting objectives, and use multi-objective optimization to simultaneously optimize them: Chen et al. (1996), Simpsons et al. (1997), Chen and Yuan (1999), Chen et al. (2000), Kalsi et al. (2001).

2.3.2. Multi-Objective Robust Optimization

Only a few methods in the literature are applicable to multi-objective robust optimization problems, and most of them are generalizations of Taguchi's loss function

(i.e., the S/N ratio). The first work to extend the loss function to multiple responses is by Pignatiello (1993) for the case of nominal-the-best. Tsui (1999) later expanded the work to include smaller-the-better and larger-the-better cases. Not wanting to use the S/N ratio in their robust design, Elsayed and Chen (1993) developed a quantity called PerMQ (Performance Measure on Quality), a generalization of the PerMIA measure of Leon et al. (1987) for single-response problems.

Other non-Taguchi based methods have been developed as well: Rao (1984), Ramakrishnan and Rao (1996), Lee and Lee (2001), Messac and Yahaya (2002), Shelokar et al. (2002), but they all use different approaches and are not focused on a certain approach.

None of the methods mentioned above take into account the trade-off that exists among the multiple objectives. A few researchers attempt to include Pareto dominance when searching for a robust solution (Kunjur and Krishnamurty, 1997; Fernandez et al., 2001). However, a true multi-objective robust optimization method that accounts for both multi-response robustness and Pareto optimality, as presented in this dissertation, has not yet been developed.

2.3.3. Feasibility Robust Optimization

Many deterministic methods have been developed for feasibility robust optimization, the simplest being the use of a safety factor. Several methods propose a worst-case approach where they constrain the largest constraint variation instead of the original constraint value: Parkinson et al. (1993), Yu and Ishii (1994, 1998), Hirokawa and Fujita (2002). Other methods propose non-mathematical “procedures” to guarantee a design’s

robustness: “closest feasible point” method (Balling et al., 1986), “corner space” method (Sundaresan et al., 1992, 1993). Yet some other methods use fuzzy logic rules to determine if a design is feasibly robust: Arakawa and Yamakawa (1998), Rao and Cao (2002).

The most popular of all feasibility robustness approaches is the probabilistic approach. Some methods calculate the statistical variance of the constraint value using a Taylor series expansion, and then add this variance to the original constraint, a so-called “moment matching” approach: Wu et al. (1990), Parkinson et al. (1993), Chen et al. (1996), and Ramakrishnan and Rao (1996). Some other methods replace the original constraints with a constraint on the probability of constraint satisfaction of the optimum design. This “chance-constrained” approach has been well developed for linear problems (Charnes and Cooper, 1959; Charnes and Cooper, 1963), as well as non-linear ones (Du and Chen, 2000; Jung and Lee, 2002).

The chance-constrained approach has been much expanded and developed further using reliability analysis techniques, where a design’s probability of constraint satisfaction is calculated via a so-called “reliability index”: Chandu and Grandhi (1995), Choi et al. (2001), Choi and Youn (2002), Youn et al. (2003). Tu et al. (1999) generalize this approach to two approaches: the conventional Reliability Index Approach (RIA) and the Performance Measure Approach (PMA), and show that in most cases PMA is better than RIA. Du and Chen (2000) later propose to perform a local sampling around the so-called “most probable point” to get a more accurate estimate of probability of constraint satisfaction, as well as to improve computational efficiency.

CHAPTER 3

SINGLE OBJECTIVE ROBUST OPTIMIZATION

3.1. INTRODUCTION

When optimizing a design or a system, it is important to ensure that its performance at the optimum does not change significantly if some parameters vary uncontrollably. In designing a racecar, for example, it is often desirable that its weight be minimized. At the same time, it is also necessary to guarantee that a change in its weight (due to a change in some uncontrollable parameters) is limited. If the weight of the vehicle increases significantly, the vehicle speed may decrease considerably. If the weight decreases significantly, the lift that the vehicle experiences might result in lost traction. This so-called performance robustness (or objective robustness) of a design is especially critical if the design is part of a larger system. A robotic arm for a cutting tool, for instance, is often optimized for maximum movement speed. However, once an optimum speed is decided, it must be maintained so that it is compatible with other parts of the robotic system that are designed for that speed.

The purpose of this chapter is to present a robust optimization method that can obtain a design solution that is objectively robust for a single objective optimization problem. A design is objectively robust if the variation in its objective function (i.e., variation in its performance) is small, within an acceptable range specified by the designer.

We begin this chapter by presenting a method to measure the sensitivity (or inversely the robustness) of a design using a sensitivity region concept. We then present an approach to use such a measure to obtain a robust optimum design. Several numerical

and engineering comparison studies are given in this chapter to demonstrate the applications of the method. Note that in this chapter we consider objective robustness only, and assume that the robust optimum obtained is feasible when parameter variations occur. We will present our feasibility robustness approach later on in Chapter 5.

3.2. TWO-SIDED SENSITIVITY MEASURE

We start by developing a method to measure the two-sided sensitivity of a design. This sensitivity measure is developed based on the notion that for each design alternative, there is a region in $\Delta\mathbf{p}$-space that can be used to evaluate that design's sensitivity. This measure is a "two-sided" measure because we limit both the increase and decrease of the design performance (unlike feasibility robustness in Chapter 5, which is "one-sided").

3.2.1. Sensitivity Set

Let $\boldsymbol{x}_0$ be a design alternative whose sensitivity we want to measure, and let $\mathbf{p}_0$ be the nominal parameter values for which the objective value of that design is defined, i.e., $f(\boldsymbol{x}_0,\mathbf{p}_0)$. If a subset of $\boldsymbol{x}$ belongs to $\mathbf{p}$, then the $\mathbf{p}_0$ values of this subset are its $\boldsymbol{x}_0$ values. Also let $\Delta f_0 \geq 0$ be the maximum acceptable changes for the design performance as determined by the designer. For this design and given Δf_0, there is a set of $\Delta\mathbf{p}$ values such that the changes in $f(\boldsymbol{x}_0,\mathbf{p}_0)$ due to these $\Delta\mathbf{p}$'s are less than or equal to Δf_0. This set is called the "sensitivity set" ($\mathbf{S}_f$) of the design, and mathematically it is defined by Eq. (3.1). (We use the square of Δf to account for negative values.)

$$\begin{aligned} &\mathbf{S}_f(\boldsymbol{x}_0) = \left\{ \Delta\mathbf{p} \in \mathrm{R}^G : \left[\Delta f(\Delta\mathbf{p})\right]^2 \leq \left[\Delta f_0\right]^2 \right\} \\ &\text{where:} \quad \Delta f(\Delta\mathbf{p}) = f(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f(\boldsymbol{x}_0, \mathbf{p}_0) \end{aligned} \tag{3.1}$$

Let us take a moment to discuss the importance of this set. What exactly is a sensitivity set? As shown in Eq. (3.1), $\mathbf{S}_f$ is a set of $\Delta\mathbf{p}$'s that can be allowed to happen if we want variation in $f(\boldsymbol{x}_0,\mathbf{p}_0)$ to be within Δf_0. So, a sensitivity set is essentially a collection of parameter changes $\Delta\mathbf{p}$ that a design can "absorb" before it violates the acceptable performance variation limit Δf_0. Clearly, a design that can absorb a large amount of $\Delta\mathbf{p}$ is less sensitive (or more robust) than a design that can absorb only a small amount. This observation implies that $\mathbf{S}_f$ is an indicator of a design's sensitivity. As the number of elements in the set $\mathbf{S}_f$ increases, the design can allow more changes in $\mathbf{p}$. This in turn brings up two key observations:

1. Given two designs, the design with a larger $\mathbf{S}_f$ is less sensitive (more robust) to changes in $\mathbf{p}$ than the design with a smaller $\mathbf{S}_f$
2. If we can control $\Delta\mathbf{p}$ such that it is always a member of $\mathbf{S}_f$, then we can guarantee that the Δf_0 limit will always be satisfied.

Provided $f(\boldsymbol{x}_0,\mathbf{p}_0)$ is defined and thus exists, $\mathbf{S}_f$ is guaranteed to exist and is always unique. The existence of $\mathbf{S}_f$ is easy to see. Because f is defined for the pair $(\boldsymbol{x}_0,\mathbf{p}_0)$, then the smallest set possible is $\mathbf{S}_f = \{\mathbf{0}\}$. An empty $\mathbf{S}_f$ implies that $f(\boldsymbol{x}_0,\mathbf{p}_0)$ does not exist; a contradiction to our assumption. The uniqueness of $\mathbf{S}_f$ is also straightforward. Suppose $\boldsymbol{x}_0$ has J non-unique sensitivity sets that satisfy Eq. (3.1): $\mathbf{S}_{f,1},\ldots,\mathbf{S}_{f,J}$. Then the unions of all these sets must necessarily also satisfy Eq. (3.1), and this superset becomes the unique sensitivity set of $\boldsymbol{x}_0$: $\mathbf{S}_f = \{\mathbf{S}_{f,1} \cup \ldots \cup \mathbf{S}_{f,J}\}$.

Let us demonstrate with an example how to use the sensitivity set of a design to measure its robustness.

Example 3.1

Consider a cylindrical piston pin made out of stainless steel (density $\rho = 8.0$ gr.cm^{-3}) whose height and radius are: h = 5 cm and r = 2 cm, respectively. The nominal weight of the pin is W = 502.6 gr, but suppose due to variations in ρ and r, the actual value of the pin's weight varies. If we want the weight to vary by at most $\Delta W_0 = 10$ gr, determine the Δρ and Δr values that can be allowed to occur.

Solution

The weight of the pin is given by $W = \rho\pi r^2 h$. When there are variations Δρ and Δr, the weight becomes $W' = (\rho+\Delta\rho)\pi(r+\Delta r)^2 h$. Setting the square of the weight difference (with and without the variations): $\Delta W^2 = (W'-W)^2$, to be less than $(\Delta W_0)^2$, we obtain a quadratic inequality:

$$t^2 - (2\rho r^2)\, t - \left(\frac{\Delta W_0^2}{(\pi h)^2} - \rho^2 r^4 \right) \le 0 \tag{3.2}$$

where: $t = uv^2$, $u = (\rho+\Delta\rho)$, and $v = (r+\Delta r)$. Solving the inequality in Eq. (3.2) and using backward substitution, we obtain the sensitivity set of the pin design:

$$\mathbf{S}_f = \left\{ (\Delta\rho, \Delta r) : \frac{\rho r^2 - \dfrac{\Delta W_0}{\pi h}}{(r+\Delta r)^2} - \rho \le \Delta\rho \le \frac{\rho r^2 + \dfrac{\Delta W_0}{\pi h}}{(r+\Delta r)^2} - \rho \right\} \tag{3.3}$$

As long as the condition in Eq. (3.3) is satisfied, the pin weight will always remain within ±10 gr. For a simple verification, if (Δρ,Δr) = (-0.4, 0.05), then the pin weight is 501.7 gr and the condition in Eq. (3.3) is satisfied. If (Δρ,Δr) = (1.0, -0.02), then the pin weight is 554.2 gr, and the condition in Eq. (3.3) is violated. If (Δρ,Δr) = (0.495, -0.04), the

condition is actively (as an equality) satisfied, and the pin weight is 512.6 gr (exactly equal to: nominal weight plus ΔW_0).

The reader might have readily observed that there is an apparent relationship between strict satisfaction of Eq. (3.3) and the amount of the ΔW_0 limit being used up. This observation is not a coincidence. In fact, it is an important property that later will become the basis for all of our robust optimization methods. ♦

3.2.2. Sensitivity Region

Because we are dealing with continuous $\Delta\mathbf{p}$, mathematically the size of $\mathbf{S}_f$ is infinite, and as such we cannot explicitly use it as a measure of a design's robustness. However, if we plot $\mathbf{S}_f$ in the $\Delta\mathbf{p}$-space, we obtain a region surrounding the origin that we call the "sensitivity region" (SR) of the design. The size of this region explicitly corresponds to the size of $\mathbf{S}_f$, and therefore the size of a SR is also a measure of a design's sensitivity: the larger the region, the less sensitive (more robust) the design. Figure 3.1 shows a typical example of a SR (for a two-parameter function).

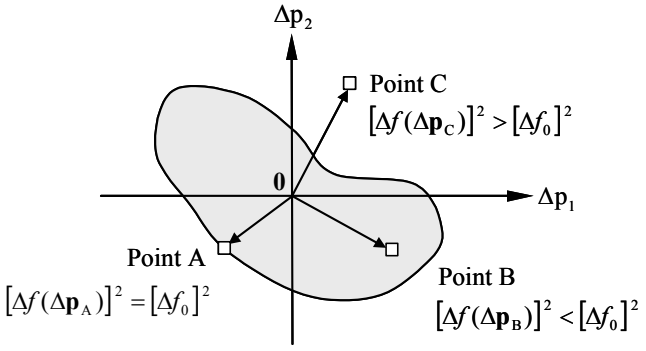


Figure 3.1: Sensitivity region of a design alternative.

Since SR is simply a plot of $\mathbf{S}_f$, the conditions for a $\Delta\mathbf{p}$ point to be inside, outside, or on the boundary of a SR can be derived from Eq. (3.1). It is obvious that if a point is

outside a SR (e.g., point C), then Eq. (3.1) is not satisfied, i.e., $[\Delta f(\Delta \mathbf{p})]^2 > [\Delta f_0]^2$. What about points inside and on the boundary of a SR? It is not obvious from Eq. (3.1) what the conditions for these points are. As it turns out, a point inside a SR (e.g., point B) satisfies $[\Delta f(\Delta \mathbf{p})]^2 < [\Delta f_0]^2$, while a point on the boundary of a SR satisfies Eq. (3.1) with an equality $[\Delta f(\Delta \mathbf{p})]^2 = [\Delta f_0]^2$.

Let us discuss the boundary point condition first. The argument for this condition is as follows. Suppose a point is on the boundary of a SR but Eq. (3.1) is satisfied as an inequality. Then the inequality implies that $\Delta \mathbf{p}$ can change by an infinitesimal amount in any direction, and the condition $[\Delta f(\Delta \mathbf{p})]^2 \le [\Delta f_0]^2$ is still satisfied. But geometrically, if the point is on the boundary of a SR, then there is at least one direction along which the change in $\Delta \mathbf{p}$, no matter how small, will push the point to the outside of SR, i.e., $[\Delta f(\Delta \mathbf{p})]^2 > [\Delta f_0]^2$ (Figure 3.2). This is a contradiction, so we conclude that if a point is on the boundary of a SR, then Eq. (3.1) must necessarily be satisfied with an equality.

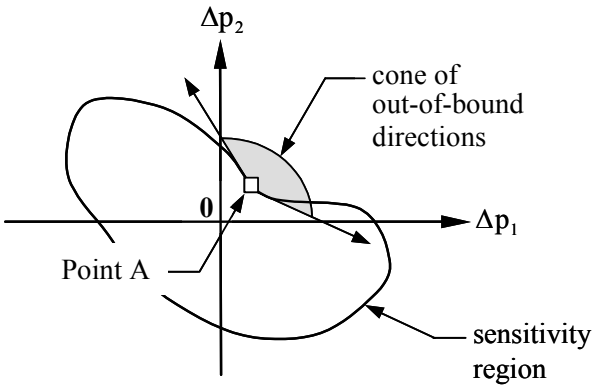


Figure 3.2: Geometric condition for a boundary point.

It should be noted that the above argument is valid only if $f(\boldsymbol{x},\mathbf{p})$ is continuous with respect to $\mathbf{p}$ (but not necessarily differentiable). If $f(\boldsymbol{x},\mathbf{p})$ (hence $\Delta f(\Delta \mathbf{p})$) is discontinuous, the condition $[\Delta f(\Delta \mathbf{p})]^2 = [\Delta f_0]^2$ is still valid as long as the discontinuity does not occur

on the SR boundary (i.e., as long as it is an inner discontinuity). However, even if the discontinuity does occur on the boundary, the condition can still be valid provided the discontinuity is such that $[\Delta f(\Delta \mathbf{p})]^2 = [\Delta f_0]^2$ as the discontinuity is approached from the left or from the right (Figure 3.3). Most engineering functions are continuous, so we rarely have to deal with this situation (recall that we are not dealing with discrete $\Delta \mathbf{p}$). However, if the function of interest happens to be discontinuous, then we assume that the discontinuity observes either one of the two conditions described above.

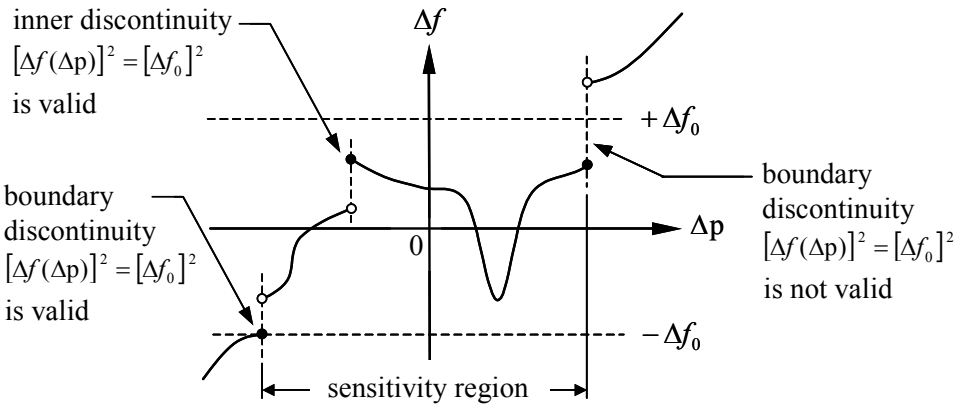


Figure 3.3: One-dimensional SR example of a discontinuous function.

The argument for the condition of a point inside a SR is similar to that of a boundary point condition, but there is a small complication that must be addressed. Strictly speaking, the inner point condition should be $[\Delta f(\Delta \mathbf{p})]^2 \leq [\Delta f_0]^2$ because it is possible that the equality is satisfied by inner points also (as shown in Figure 3.4). However, since we are using the size of SR to measure the robustness of a design, we are interested primarily in the SR boundary. So, to avoid computational problems it is necessary to make a clear distinction between inner and boundary point conditions. To make this distinction, we modify the given Δf_0 to $\Delta \hat{f}_0 = \Delta f_0 + \delta_f$, where δ_f is an infinitesimal positive number.

Using $\Delta\hat{f}_0$, the inner point condition becomes $[\Delta f(\Delta\mathbf{p})]^2 < [\Delta\hat{f}_0]^2$ and the boundary point condition becomes $[\Delta f(\Delta\mathbf{p})]^2 = [\Delta\hat{f}_0]^2$.

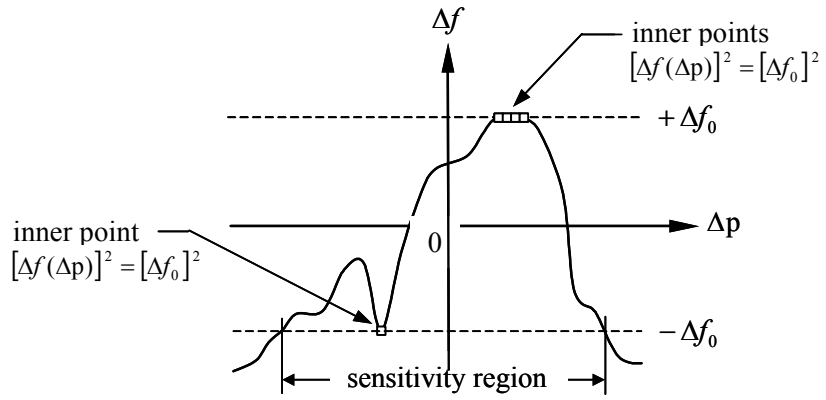
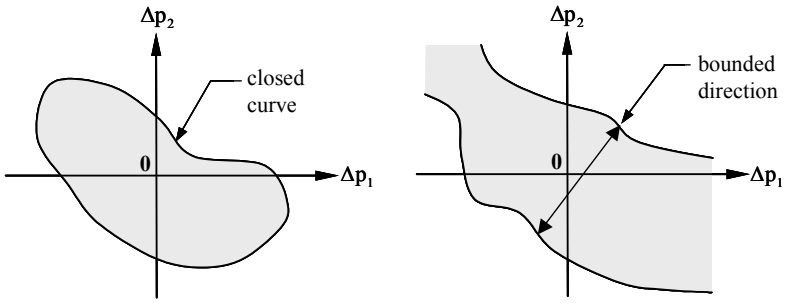


Figure 3.4: One-dimensional example of equality inner point condition.

Mathematically, the boundaries defined by $[\Delta f(\Delta\mathbf{p})]^2 = [\Delta\hat{f}_0]^2$ and $[\Delta f(\Delta\mathbf{p})]^2 = [\Delta f_0]^2$ are slightly different. However, since δ_f is infinitesimal, practically they are the same. For the rest of our discussions, we will use $[\Delta f(\Delta\mathbf{p})]^2 < [\Delta\hat{f}_0]^2$ and $[\Delta f(\Delta\mathbf{p})]^2 = [\Delta\hat{f}_0]^2$ for inner and boundary point conditions, respectively. For simplicity, we will use the notation Δf_0 to refer to $\Delta\hat{f}_0$.

The existence and uniqueness of a SR follows directly from the existence and uniqueness of $\mathbf{S}_f$. One property of SR that should be noted is that it always encloses the origin of the $\Delta\mathbf{p}$-space because the set $\{\mathbf{0}\}$ is always a member of $\mathbf{S}_f$. This property will become important later on. It should also be noted that a SR does not necessarily have to be fully bounded although we showed a fully bounded SR example in Figure 3.1 and 3.2. By fully bounded we mean that in the G-dimensional $\Delta\mathbf{p}$-space, the SR boundary is a closed hyper-surface. However, we do assume that SR is bounded along at least one $\Delta\mathbf{p}$-direction, not necessarily the coordinate axis (Figure 3.5).



(a) Fully bounded SR (b) Not fully bounded SR

Figure 3.5: Boundedness of a sensitivity region.

The assumption for the SR's one-directional boundedness is necessary because if the SR is completely unbounded, then it implies that the design is already inherently robust, and hence there is no need to calculate its robustness. Two conditions can cause a SR to be unbounded. First, the objective function $f(\boldsymbol{x}_0,\mathbf{p}_0)$ is independent of $\Delta\mathbf{p}$, i.e., $f(\boldsymbol{x}_0,\mathbf{p}_0) = f(\boldsymbol{x}_0,\mathbf{p}_0+\Delta\mathbf{p})$, $\forall\Delta\mathbf{p}$. In turn, this means that the objective value of the design $\boldsymbol{x}_0$ is unaffected by changes in $\mathbf{p}$ (i.e., the hyper-surface of $f(\boldsymbol{x}_0,\mathbf{p}_0)$ with respect to $\mathbf{p}$ is flat). Second, Δf_0 is larger than the maximum possible Δf (e.g., the sinus function in Figure 3.6). This also implies that $\boldsymbol{x}_0$ is practically unaffected by $\Delta\mathbf{p}$; at least for the given Δf_0. If either condition holds, then any design alternative is already inherently robust, and there is no need to search for the robust optimum design.

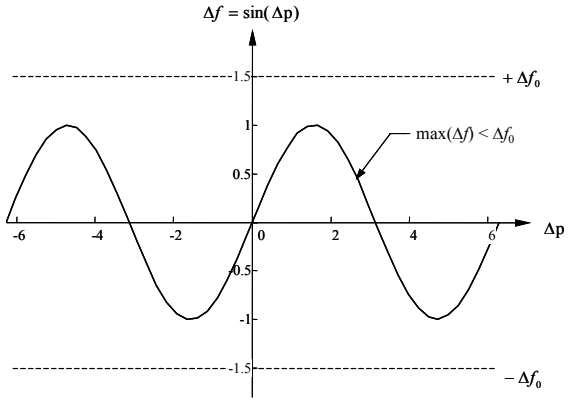


Figure 3.6: Example of a condition that causes SR to be unbounded.

Note also that a SR does not have to be a single connected region. It can be a collection of several disjointed regions in which one of them encloses the origin (Figure 3.7). Each of the disjointed regions can be either fully or partially bounded.

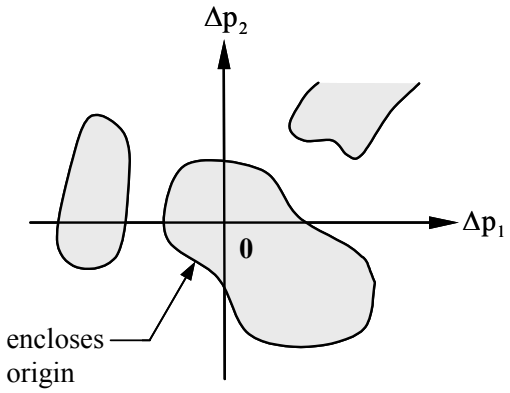


Figure 3.7: Example of a disjointed sensitivity region.

A SR has a geometric significance as well. If we look into the hyper-surface of the function $\Delta f(\Delta \mathbf{p})$, then SR is simply the collection of points sandwiched between two parallel $-\Delta f_0$ and $+\Delta f_0$ hyper-planes, projected onto the $\Delta \mathbf{p}$ hyper-plane (as shown in Figure 3.3 and 3.4 for one dimension). In other words, SR is the region between the contours of $\Delta f(\Delta \mathbf{p})$ for values $\Delta f = -\Delta f_0$ and $\Delta f = +\Delta f_0$. Since contours of a function always exist and are unique, this observation provides yet another justification for the existence and uniqueness of SR. In addition, it also provides an explanation as to why it is possible for a SR to be not fully bounded or even be disjointed.

Example 3.2

Let us revisit our piston pin design example again. In Example 3.1 we derived the sensitivity set of the design when its height and radius are $h = 5$ cm and $r = 2$ cm, respectively. Given $\Delta W_0 = 10$ gr, what is the sensitivity region of this design?

Solution

Since we already have $\mathbf{S}_f$, we can directly use Eq. 3.3 to plot the SR of this design as shown in Figure 3.8(a) below.

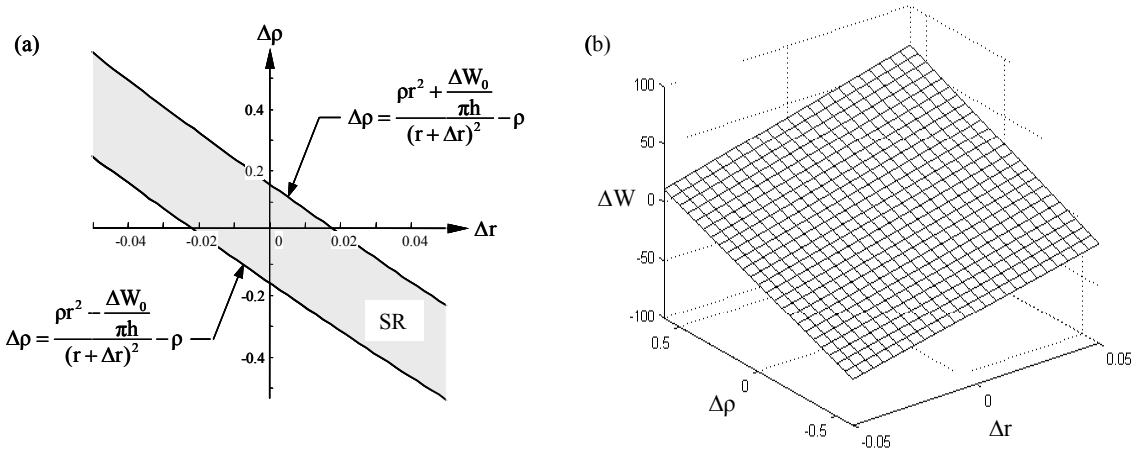


Figure 3.8: (a) Sensitivity region of the piston pin. (b) Surface plot of ΔW(Δρ,Δr).

For comparison, we have also shown the surface plot of the difference function ΔW(Δρ,Δr), as shown in Figure 3.8(b). Notice in Figure 3.8(a) how the SR is simply the projection of points between two parallel planes $\pm\Delta W_0$ cutting through the surface in Figure 3.8(b). Notice also that the SR encloses the origin, but is unbounded.

In Example 3.1 we performed a simple calculation to validate the accuracy of Eq. 3.3. We showed that for the pair (Δρ,Δr) = (-0.4, 0.05) the pin weight is 501.7 gr (less than $W+\Delta W_0$), for (Δρ,Δr) = (1.0, -0.02) the pin weight is 554.2 gr (greater than $W+\Delta W_0$), and for (Δρ,Δr) = (0.495, -0.04), the pin weight is 512.6 gr (equal to $W+\Delta W_0$). Observe how these three (Δρ,Δr) pairs fall inside, outside, and on the boundary of the SR shown in Figure 3.8(a), respectively, and that the changes in the weight are as we predicted. ♦

3.2.3. Directional Sensitivity

Figure 3.8(a) shows a drawback in using SR size as a robustness measure, namely the asymmetry of the SR. Because of this asymmetry, a design might be very robust if $\Delta\mathbf{p}$ moves along certain directions, but very sensitive if it moves along some other directions. Besides, if the SR is not fully bounded (like in Figure 3.8(a)), its size is infinite. Figure 3.9 shows an example of such a directional sensitivity. In this figure, the SR of design A is larger than that of design B and therefore, based on our previous discussions, design A is more robust than design B. However, if $\Delta\mathbf{p}$ happens to vary along the direction shown, design A will violate the Δf_0 limit first before design B does. In other words, for this $\Delta\mathbf{p}$ direction design B should be considered more robust than design A.

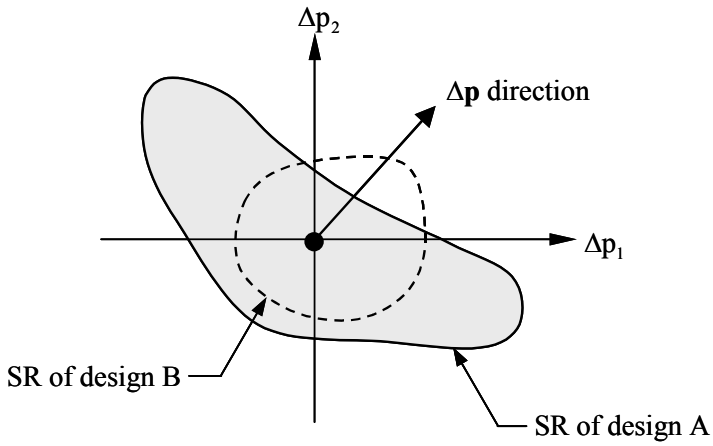


Figure 3.9: Example of a directional sensitivity.

From this simple example it is apparent that the size of SR only measures the overall sensitivity behavior of a design. If design A has a larger SR than design B, then when $\Delta\mathbf{p}$ occurs along all directions, design A will stay within the acceptable Δf_0 bound more often than design B. However, this does not mean that design A will always be more robust

than design B. Therefore, to get a more complete picture of a design's robustness, we need to consider the directional sensitivity of the design as well.

Example 3.3

Let us return to our piston pin example. Suppose the designer wants to decrease the weight of the pin by using an aluminum alloy 6061 (density $\rho = 2.7$ gr.cm^{-3}) instead of stainless steel. To maintain the strength requirement, however, (s)he finds that the pin radius must be increased to $r = 3$ cm. The pin height is kept at $h = 5$ cm. The nominal weight of the aluminum pin is calculated to be $W = 381.7$ gr, so it is indeed lighter than the stainless steel pin. If the designer still wants the weight variation to be within $\pm\Delta W_0 = 10$ gr, determine the sensitivity region of this design. Is the aluminum pin more or less robust than the stainless steel pin?

Solution

The $\mathbf{S}_f$ equation in Eq. 3.3 was derived independent of the nature of the design, so it is also applicable to the aluminum pin design provided we use the appropriate values. Substituting $\rho = 2.7$ and $r = 3$ into Eq. 3.3 and plot the inequalities, we obtain the SR of the aluminum pin as shown in Figure 3.10. For comparison, we have also shown the SR of the stainless steel pin.

As can be seen in Figure 3.10, the aluminum pin SR is narrower than the steel pin SR and is slightly rotated counter-clockwise. Overall, the steel pin SR is larger than the aluminum pin SR, so based on the sensitivity region concept, one will conclude that the steel pin is more robust than the aluminum pin. However, this assertion is only partially accurate because of the SR rotation. If the changes in the pin radius is between

$\Delta r = [-0.017, 0.017]$, then we can safely assert that the steel pin is more robust than the aluminum pin (the steel SR encloses the aluminum SR). However, beyond this range there are $(\Delta\rho,\Delta r)$ pairs that the steel pin can absorb but the aluminum pin cannot, and vice versa. As such, we cannot make a conclusive comparison between the two designs.

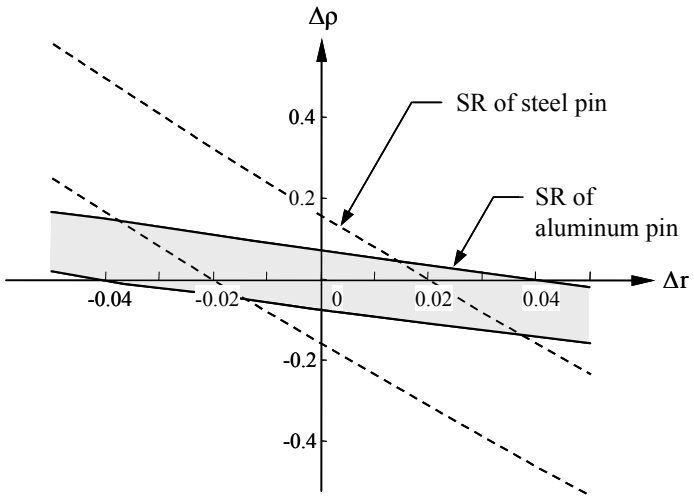


Figure 3.10: SR of the aluminum and stainless steel pins.

Notice in this example how we can easily compare the two SR's even though the ρ and r values of the two designs are different. This is because SR is defined in the Δ**p**-space, and not in the **p**-space. One advantage of working in the Δ**p**-space is that the SR's are automatically translated to enclose the origin of this space. In addition, the **p** value difference among the designs is also automatically taken care of because we are now looking into the changes of those **p** values, and not their absolute values. ♦

An important Δ**p**-direction that must be considered when measuring a design's robustness is the most sensitive (least robust) direction: the Δ**p**-direction along which the design can absorb the least amount of Δ**p** before it violates the Δf_0 bound. In other words, this direction represents the worst-case scenario for this design. Geometrically, this

worst-case direction is depicted as a vector from the origin of the Δ**p**-space to the point on the SR boundary that is closest (in terms of Euclidean distance) from the origin (Figure 3.11).

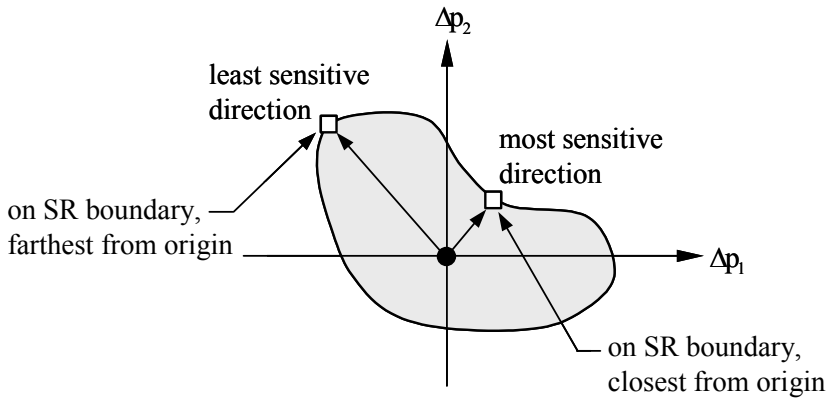


Figure 3.11: Most and least sensitive directions of a SR.

Another Δ**p**-direction that might also be of interest is the least sensitive (most robust) direction: the Δ**p**-direction along which the design can absorb the most amount of Δ**p** before it violates the Δf_0 bound. This direction represents the best-case scenario for the design, and is depicted as a vector from the origin to a point on the SR boundary farthest from the origin (Figure 3.11). If a SR is not fully bounded, the least sensitive direction is the direction along which the SR is unbounded. For the best-case direction to make sense, however, all Δ**p** along this direction must be part of the SR (Figure 3.12).

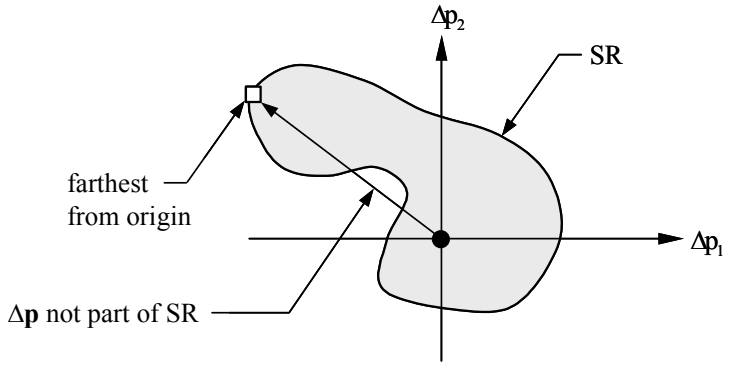


Figure 3.12: An example where best-case direction not contained in a SR.

3.2.4. Worst Case Sensitivity Region

It is impossible to consider all possible $\Delta\mathbf{p}$ directions when comparing the robustness of two or more designs. This is especially true as the number of parameters under consideration increases. For this reason, we choose to measure the robustness of a design in the most important direction only: the worst-case direction. Using this worst-case approach, the robustness of a design is no longer measured by the size of a SR, but by the size of the worst-case approximation to SR. We call this approximation the Worst Case Sensitivity Region (WCSR) of the design.

Geometrically, a WCSR is defined as the smallest hyper-sphere inside a SR that touches the SR boundary on at least one point (Figure 3.13). A WCSR does not have to be tangent to the SR boundary (e.g., when the point of contact is a cusp), but the point of contact between WCSR and SR must necessarily be the point on the SR boundary closest from the origin in the $\Delta\mathbf{p}$-space. This closest-from-origin requirement guarantees that the WCSR is indeed the worst-case estimate of the SR.

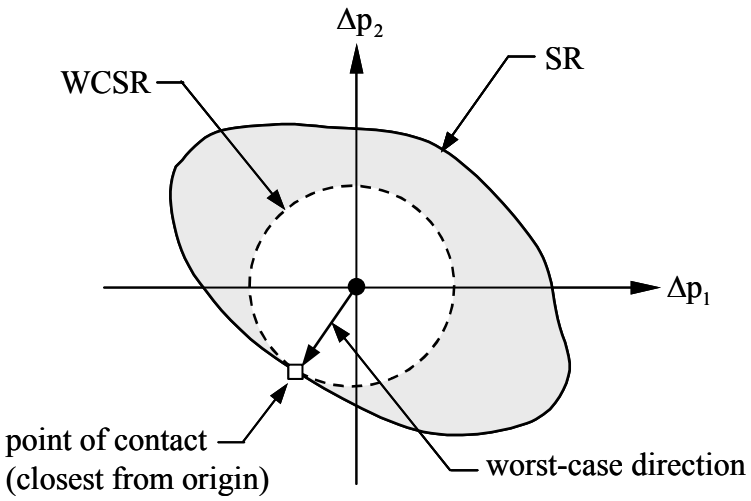


Figure 3.13: Worst Case Sensitivity Region.

Like SR, the size of a WCSR measures how much $\Delta\mathbf{p}$ a design can absorb before Δf_0 is violated, although in the worst-case sense only. So, the size of a WCSR is also a measure of a design's robustness: the larger the size the more robust the design. Unlike SR, however, WCSR is always symmetric because it is a hyper-sphere. As such, when using a WCSR to compare the robustness of two designs, the directional sensitivity of the designs has been taken into account, and the comparison will be conclusive (i.e., one design is definitely more or less or as robust as the other). Unless the SR is a hyper-sphere, using a WCSR to approximate it will always lead to an underestimation of its size. In return, however, we are guaranteed that the design is at least as robust as the approximation predicts it.

In addition to its symmetry, WCSR has a computational advantage over SR as well. Because of its simple geometric shape, size of a WCSR is much easier to calculate than the typically-not-so-simple SR size. In fact, in general the mathematical formulation of a SR shape is not known in closed form, so analytical calculation is often not possible while simulation-based calculation can be prohibitively expensive.

To calculate the size of a WCSR, we need to calculate its radius, i.e., Euclidean distance from the origin to the point of contact. Using the fact that this point of contact is: (i) on the SR boundary and (ii) closest from the origin, the WCSR radius can be obtained by solving the following optimization problem:

$$\begin{aligned} &\underset{\Delta\mathbf{p}}{\text{minimize}} \quad \mathrm{R}_f(\Delta\mathbf{p}) = \left[\sum_{i=1}^{\mathrm{G}} (\Delta \mathrm{p}_i)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \quad 1 - \frac{\left[\Delta f(\Delta\mathbf{p})\right]^2}{\left[\Delta f_0\right]^2} = 0 \\ &\text{where}: \Delta f(\Delta\mathbf{p}) = f(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f(\boldsymbol{x}_0, \mathbf{p}_0) \end{aligned} \tag{3.4}$$

In this optimization problem, the variables are the G elements of $\Delta\mathbf{p}$, and the objective is to minimize the Euclidian distance from the origin. The constraint in the problem reflects the fact that a point on a SR boundary has to satisfy: $[f(\boldsymbol{x}_0,\mathbf{p}_0+\Delta\mathbf{p}) - f(\boldsymbol{x}_0,\mathbf{p}_0)]^2 = [\Delta f_0]^2$. The optimum R_f value for this problem is the WCSR radius that we seek.

The optimization problem in Eq. (3.4) is guaranteed to have a solution. The point $\Delta\mathbf{p} = \mathbf{0}$ is always a member of $\mathbf{S}_f$, so the smallest WCSR radius is $R_f = 0$. In the extreme case where the design's SR is unbounded, there will be no feasible solution to Eq. (3.4) and $R_f = \infty$. However, recall that we require the SR to be bounded along at least one $\Delta\mathbf{p}$-direction. So, based on this requirement: $R_f < \infty$, and in general $0 \le R_f < \infty$.

Note that Eq. (3.4) is not the optimization problem for which we are trying to find a robust optimum. The optimization problem in Eq. (3.4) is used to measure a design's robustness, which later on will be used to obtain the robust optimum of the original optimization problem.

Mathematically, sometimes it is easier to satisfy an inequality constraint than an equality constraint. Since Eq. (3.4) is a minimization problem, its constraint can be changed into an inequality, as shown in Eq. (3.5), without changing the optimum R_f. This inequality constraint relaxes the requirement that the point of contact must be on the SR boundary. The feasible domain now includes points outside the SR as well. However, since we are minimizing a distance function, at the optimum this inequality constraint must be active (i.e., satisfied as an equality). For differentiability purposes, it is also recommended that we optimize R_f^2 instead of R_f. For the rest of our discussions, we will use Eq. (3.4) to calculate a WCSR radius because that is a more general formulation. In our comparison studies, however, Eq. (3.5) is often used as well.

$$\begin{aligned} &\underset{\Delta\mathbf{p}}{\text{minimize}} \quad R_f^2 = \sum_{i=1}^{G} (\Delta p_i)^2 \\ &\text{subject to}: \quad 1 - \frac{[\Delta f(\Delta\mathbf{p})]^2}{[\Delta f_0]^2} \le 0 \end{aligned} \tag{3.5}$$

Once we obtain a WCSR radius, its size can be easily calculated using the following formula for the "hyper-volume" of a G-dimensional hyper-sphere (Apostol, 1957):

$$V(G) = \frac{\pi^{\frac{G}{2}}}{\Gamma[\frac{G}{2}+1]} R_f^G \tag{3.6}$$

where $\Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx$ is the gamma function, and R_f is the WCSR radius. Eq. (3.6) is often broken down into more readily usable equations:

$$V(G) = \begin{cases} \dfrac{\pi^{\frac{G}{2}}}{(\frac{G}{2})!} R_f^G & ; \quad G \text{ is even} \\ 2^G \pi^{\frac{G-1}{2}} \dfrac{(\frac{G-1}{2})!}{G!} R_f^G & ; \quad G \text{ is odd} \end{cases} \tag{3.7}$$

(As a simple check: if G = 2, V = πR_f^2, area of a circle; if G = 3, V = $\frac{4}{3}\pi R_f^3$, volume of a sphere.)

It should be pointed out that as a robustness measure, WCSR size is fundamentally different from conventional robustness measures discussed in the literature. That is in general, our robustness measure cannot determine Δf given $\Delta\mathbf{p}$ (e.g., the gradient measure $\nabla_{\mathbf{p}} f$). The reason is because unlike conventional measures, WCSR size measures the design robustness in "reverse." Instead of giving an answer to the question: "If we have $\Delta\mathbf{p}$, what is Δf?", we provide an answer to the question: "If we can allow Δf_0, what is $\Delta\mathbf{p}$?"

Because the size of a WCSR cannot determine Δf given $\Delta\mathbf{p}$, in this sense WCSR size is an ordinal measure. When comparing robustness of two designs, the actual values of their WCSR sizes are not important, only the ordering of these values is important. This is especially true considering that the actual value of a WCSR size is an abstract indicator that does not explicitly represent a physical quantity. On the other hand, the ordering of WCSR sizes is not the only important aspect of this measure. The relative ratio among WCSR sizes is also important. For instance, if WCSR of say design A is twice as large as design B, then we know that design A can absorb $\left(\frac{R_{f,A}}{R_{f,B}}\right)^G = 2^G$ times more $\Delta\mathbf{p}$ than design B. In this sense, design A is 2^G times more robust than design B. So from this angle, the WCSR size measure is cardinal, although not in the traditional sense of performance robustness. In other words, WCSR size can be said to be a "semi-cardinal" measure.

Because the size of a WCSR is a semi-cardinal measure, we do not need to calculate its actual value. Rather, we can simply use the WCSR radius R_f to measure a design's robustness. Like our previous discussion showed, using only R_f is sufficient to calculate the relative robustness of two designs. For the rest of our discussions, we will use R_f as a robustness measure and will not calculate the actual size of the hyper-volume defined by this radius.

Example 3.4

Determine the WCSR's of our stainless steel and aluminum pin designs using the SR's obtained in Example 3.3. Based on the worst-case notion, which pin design is more

robust? Calculate the radius of each WCSR mathematically using Eq. (3.4). Is it consistent with the WCSR graphs obtained?

Solution

The SR's and WCSR's of the two pin designs are shown in Figure 3.14. We can obtain the WCSR radius graphically from Figure 3.14. For the stainless steel pin, the point of contact between SR and WCSR is approximately at $(\Delta\rho,\Delta r) = (0.11, 0.05)$ for a WCSR radius of $R_f = 0.12$. For the aluminum pin, the point of contact between SR and WCSR is approximately at $(\Delta\rho,\Delta r) = (0.08, 0.001)$ for a WCSR radius of $R_f = 0.08$. Since the R_f of the stainless steel pin is larger than the R_f of the aluminum pin, the stainless steel pin is more robust than the aluminum pin.

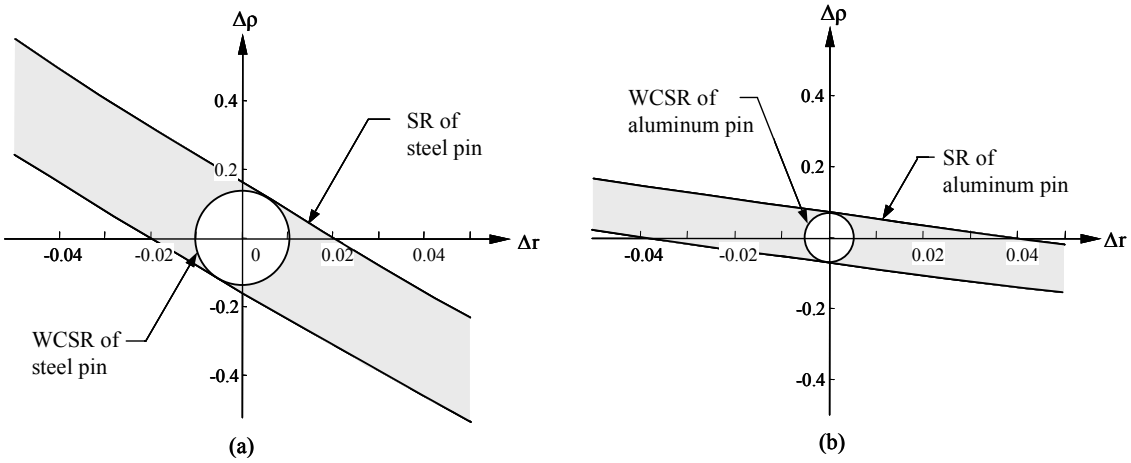


Figure 3.14: SR and WCSR of the (a) steel pin, and (b) aluminum pin.

To obtain the WCSR radius mathematically, we solve the following optimization problem (we use R_f^2 as the objective):

$$\begin{aligned} &\underset{(\Delta\rho,\Delta r)}{\text{minimize}} \quad R_f^2 = \Delta\rho^2 + \Delta r^2 \\ &\text{subject to}: \quad 1 - \frac{1}{(\Delta W_0)^2} \cdot \left[\Delta W(\Delta\rho, \Delta r)\right]^2 = 0 \end{aligned} \tag{3.8}$$

Since we know that the SR boundary is defined by $\Delta\rho = \dfrac{\rho r^2 - \dfrac{\Delta W_0}{\pi h}}{(r+\Delta r)^2} - \rho$ and $\Delta\rho = \dfrac{\rho r^2 + \dfrac{\Delta W_0}{\pi h}}{(r+\Delta r)^2} - \rho$ (recall Figure 3.8(a)), Eq. (3.8) can be broken down into two simpler problems, Eq. (3.9) and Eq. (3.10):

$$\begin{aligned} &\underset{\Delta r}{\text{minimize}} \quad R_{f,1}^2 = [\Delta\rho(\Delta r)]^2 + \Delta r^2 \\ &\text{where:} \quad \Delta\rho(\Delta r) = \frac{\rho r^2 - \dfrac{\Delta W_0}{\pi h}}{(r+\Delta r)^2} - \rho \end{aligned} \tag{3.9}$$

$$\begin{aligned} &\underset{\Delta r}{\text{minimize}} \quad R_{f,2}^2 = [\Delta\rho(\Delta r)]^2 + \Delta r^2 \\ &\text{where:} \quad \Delta\rho(\Delta r) = \frac{\rho r^2 + \dfrac{\Delta W_0}{\pi h}}{(r+\Delta r)^2} - \rho \end{aligned} \tag{3.10}$$

Where the WCSR radius is $R_f^2 = \min\left((R_{f,1}^2)^*, (R_{f,2}^2)^*\right)$, and $(R_{f,1}^2)^*$ and $(R_{f,2}^2)^*$ are the optima of Eq. (3.9) and Eq. (3.10), respectively.

Eq. (3.9) and Eq. (3.10) are both one-dimensional unconstrained optimization problems. Setting the gradient of each function to zero, the optimum Δr is obtained by finding the roots of the following sixth order polynomial:

$$\begin{aligned} &\Delta r(r+\Delta r)^5 + 2A\rho(r+\Delta r)^2 - 2A^2 = 0 \\ &\text{where: } A = \begin{cases} \rho r^2 - \dfrac{\Delta W_0}{\pi h} & \text{for Eq. (3.9)} \\ \rho r^2 + \dfrac{\Delta W_0}{\pi h} & \text{for Eq. (3.10)} \end{cases} \end{aligned} \tag{3.11}$$

The optimum $\Delta\rho$ is obtained by substituting the optimum Δr into the equalities in Eq. (3.9) and (3.10). We do not need to check for sufficiency condition because the function is unimodal minimum (unimodality of Eq. (3.11) can be verified graphically).

Using r = 2 and ρ = 8, the optimum (Δρ,Δr) for the stainless steel pin is found to be (0.0025, 0.0195) for a WCSR radius of R_f = 0.0196. Similarly, using r = 3 and ρ = 2.7, the optimum (Δρ,Δr) value for the stainless steel pin is found to be (0.0166, 0.0297) for a WCSR radius of R_f = 0.034. Notice that the R_f values obtained mathematically are different from the R_f values obtained graphically. In fact, according to the R_f's obtained by solving Eq. (3.8), the aluminum pin is more robust than the stainless steel pin, a contradiction to our graphical observation before!

The reason for this inconsistency is because the magnitudes of Δρ and Δr are different (Δρ is an order of magnitude larger than Δr). Because of this magnitude difference, when minimizing the distance function R_f^2, Δr numerically dominates Δρ. Graphically this numerical domination translates into an elliptical WCSR, and not a circular WCSR (Figure 3.15).

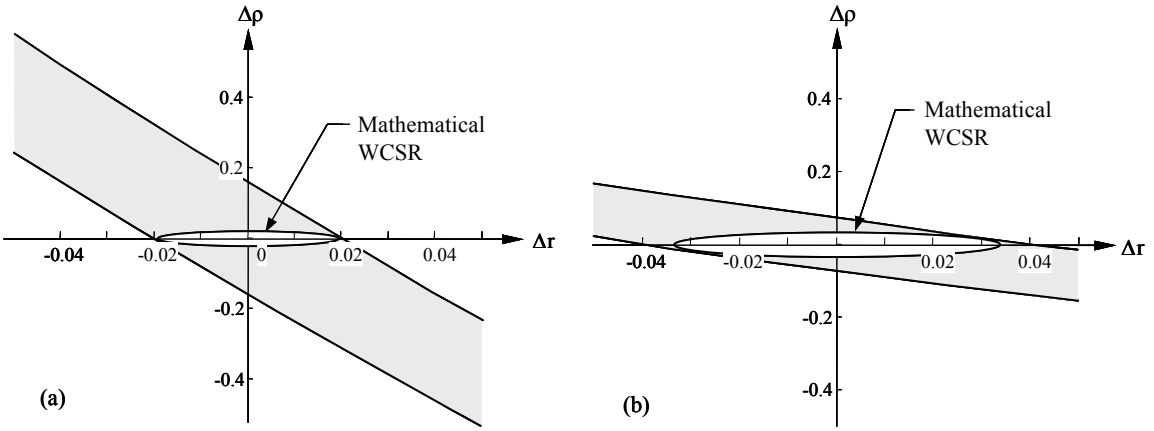


Figure 3.15: Mathematical WCSR of the (a) steel pin, and (b) aluminum pin.

The reader might be surprised to know that actually the mathematical WCSR (Figure 3.15) is a circle and the WCSR in Figure 3.14 is an ellipse. The reason they look the way we showed them is because the scales of the Δρ and Δr axes are different. The

scale of the Δρ axis is much larger than the Δr axis, in effect "compressing" the WCSR. If the WCSR is plotted on an equally scaled coordinate system, the mathematical WCSR will be circular (Figure 3.16), and hence is consistent with Eq. (3.8).

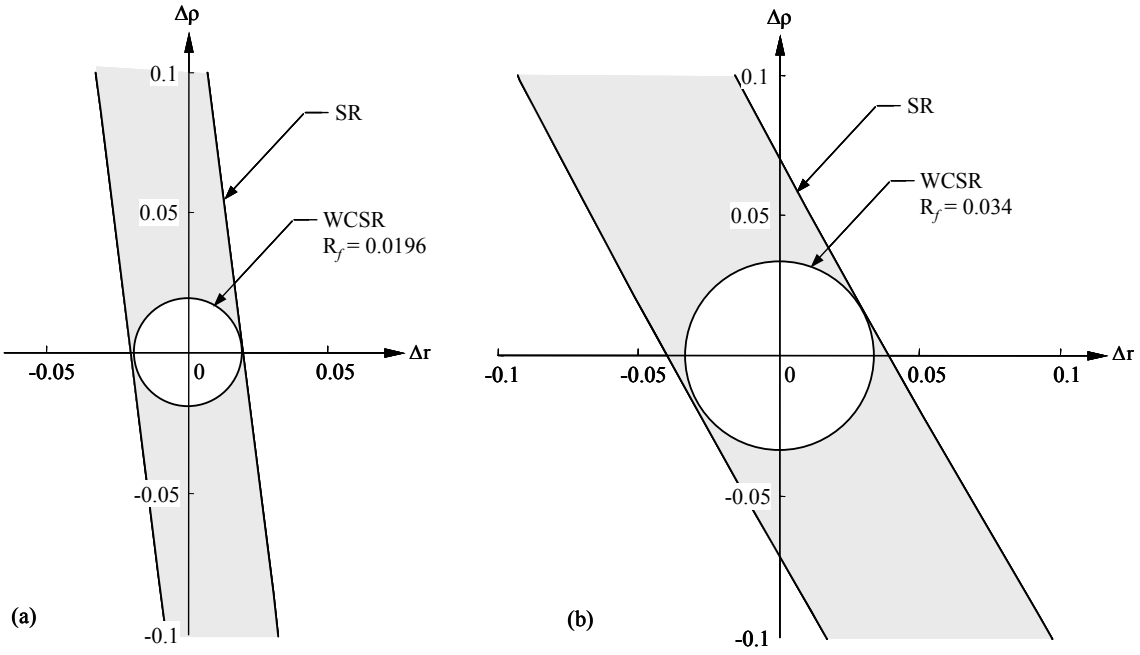


Figure 3.16: Equal-scaled SR and WCSR of (a) steel and (b) aluminum pins.

According to Figure 3.14, the steel pin is more robust than the aluminum pin, but according to Figure 3.16, the aluminum pin is more robust. Which one is correct? Clearly, when there is a difference in **Δp** magnitudes, there is an ambiguity between the WCSR concept and the mathematics behind it. As it turns out, the optimization problem used to calculate the WCSR radius (Eq. (3.4)) needs to be normalized when the scales of **Δp** are different. ♦

3.2.5. Normalization

When the magnitudes of **Δp** are different, the optimum of Eq. (3.4) will be numerically driven by those Δp_i's whose magnitudes are small. For instance, if **Δp** are the

normal stress (magnitude in the order of 10^3) and deflection (magnitude in the order of 10^{-2}) of a beam, then the search for the optimum R_f will be numerically driven by the deflection factor. However, this is not what we want because this does not reflect the actual robustness of the design. If a beam can absorb say 10^3 MPa of stress variation, an R_f of 10^{-2} will very much distort the actual robustness of this design. In other words, *scale importance* of the stress and deflection parameters is different.

Because of this difference in scale importance, an increase (or decrease) of Δ along one axis is not the same as an increase (or decrease) of the same Δ along another axis. As a consequence, the most sensitive direction that we are interested in may not necessarily be the closest distance from the origin mathematically. Correspondingly, the WCSR defined by this direction may not be geometrically spherical. It should be pointed out that although the WCSR is not spherical, it is still equally sensitive along all directions because now the scale importance is different.

To obtain the WCSR radius that reflects the scale importance, the optimization problem in Eq. (3.4) needs to be normalized. There are many ways to normalize this problem. In this dissertation we use a single-valued normalization where we use the known variation ranges $\Delta\mathbf{p}_0$ as the reference value. The normalized optimization problem to obtain the WCSR radius is as follows:

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \quad \overline{\mathrm{R}}_f(\Delta\overline{\mathbf{p}}) = \left[\sum_{i=1}^{\mathrm{G}} (\Delta\overline{\mathrm{p}}_i)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \quad 1 - \frac{\left[\Delta f(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)\right]^2}{[\Delta f_0]^2} = 0 \\ &\text{where}: \Delta\overline{\mathbf{p}} = \left[\Delta\overline{\mathrm{p}}_1, \ldots, \Delta\overline{\mathrm{p}}_{\mathrm{G}}\right] \quad ; \quad \Delta\overline{\mathrm{p}}_i = \frac{\Delta \mathrm{p}_i}{\Delta \mathrm{p}_{0,i}} \end{aligned} \tag{3.12}$$

In this formulation, the variables are $\Delta\overline{\mathbf{p}}$ (the normalized $\Delta\mathbf{p}$), and the objective and constraint are the same except they are modified for $\Delta\overline{\mathbf{p}}$ instead of $\Delta\mathbf{p}$. The notation "$\otimes$" refers to a vector operation between $\Delta\overline{\mathbf{p}}$ and $\Delta\mathbf{p}_0$ (recall Section 2.2). As shown before we can use $\overline{R}_f^2$ for the objective and '$\leq$' (recall Eq. (3.5)) for the constraint if it makes the problem easier to solve.

One might point out that the normalization in Eq. (3.12) depends on the $\Delta\mathbf{p}_0$ value, and that the optimum $\overline{R}_f$ will change if we use a different $\Delta\mathbf{p}_0$. The actual value of $\overline{R}_f$ will indeed change along with $\Delta\mathbf{p}_0$ changes. However, the ratio between the $\overline{R}_f$'s of different designs will remain the same as long as the change in $\Delta\mathbf{p}_0$ is consistent, i.e., if $\Delta p_{0,1}$ is multiplied by 2, $\Delta p_{0,2}$ is multiplied by 2 also, and so on. This ratio preserving property is important because the ratio of $\overline{R}_f$'s determines the magnitude of relative robustness between different designs. If the $\Delta\mathbf{p}_0$ change is not consistent, then obviously the $\overline{R}_f$ ratio will be different because we have implicitly changed the scale importance, and correspondingly, the WCSR that we are searching for.

Aside from the scale importance issue, normalizing Eq. (3.4) into Eq. (3.12) has other advantages as well. Computationally, it is easier to obtain a more accurate optimum for Eq. (3.12) than for Eq. (3.4). This is because a non-normalized SR might be so stretched (i.e., a very long and thin strip) that solving Eq. (3.4) most likely results in $R_f = 0$ due to round-off errors. Conceptually, normalizing Eq. (3.4) makes a WCSR easier to work with because it is now defined in a unit-less space. A WCSR defined in a space with different units for its axes (e.g., MPa vs. cm) promotes human errors when comparing it to other WCSR's.

In almost all cases, $\Delta\mathbf{p}$ magnitudes will be different and therefore we will have to use the normalized WCSR. Even for cases when the scales are the same, we still recommend using the normalized WCSR for the reasons we just discussed. Strictly speaking, we should use the notation $\overline{R}_f$ to refer to the normalized WCSR radius and R_f for the non-normalized radius. However, since we will be using mostly the normalized radius for the rest of our discussions, we will simply use the notation R_f to refer to the normalized one.

Example 3.5

If the $\Delta\mathbf{p}_0$ of our piston pin design is known to be $(\Delta\rho_0,\Delta r_0) = (0.2, 0.02)$, calculate the normalized WCSR radius of the steel and aluminum pins. Which design (steel or aluminum) is more robust now? Derive and graph the WCSR's of the two designs in the normalized $(\Delta\rho,\Delta r)$-space, and compare them to the WCSR's we obtained in the previous examples.

Solution

The optimization problem to calculate the normalized WCSR radius is as follows (again, we use R_f^2 instead of R_f):

$$\begin{aligned} &\underset{(\Delta\bar{\rho},\Delta\bar{r})}{\text{minimize}} \quad R_f^2 = \Delta\bar{\rho}^2 + \Delta\bar{r}^2 \\ &\text{subject to}: \quad 1 - \frac{1}{(\Delta W_0)^2} \cdot \left[\Delta W(\Delta\rho_0 \cdot \Delta\bar{\rho}, \Delta r_0 \cdot \Delta\bar{r})\right]^2 = 0 \end{aligned} \tag{3.13}$$

We can use a procedure similar to the one in Example 3.4 to solve Eq. (3.13). The optimum $\Delta\bar{r}$ of this problem is obtained by finding the roots of the following sixth order polynomial:

$$(\Delta\rho_0^2)\Delta\bar{r}(r+\Delta r_0\Delta\bar{r})^5 + 2\rho\Delta r_0 A(r+\Delta r_0\Delta\bar{r})^2 - 2\Delta r_0 A^2 = 0$$

$$\text{where}: A = \begin{cases} \rho r^2 - \dfrac{\Delta W_0}{\pi h} & \text{for lower SR boundary} \\ \rho r^2 + \dfrac{\Delta W_0}{\pi h} & \text{for upper SR boundary} \end{cases} \tag{3.14}$$

The optimum $\Delta\bar{\rho}$ is obtained by substituting the optimum $\Delta\bar{r}$ into Eq. (3.15).

$$\Delta\bar{\rho} = \frac{1}{\Delta\rho_0}\left[\frac{A}{(r+\Delta r_0\cdot\Delta\bar{r})^2} - \rho\right] \tag{3.15}$$

Substituting the appropriate values into Eq. (3.14) and solving it, the optimum $(\Delta\bar{\rho},\Delta\bar{r})$ for the steel pin is (0.388, 0.4811) for an R_f = 0.618. The optimum $(\Delta\bar{\rho},\Delta\bar{r})$ for the aluminum pin is (0.0631, 0.342) for an R_f = 0.347. So, according to the normalized WCSR, the steel pin is more robust than the aluminum pin.

The graph of the normalized SR and WCSR in the $(\Delta\bar{\rho},\Delta\bar{r})$-space is shown in Figure 3.17. Notice that except for the axis values, this graph is identical to the one in Figure 3.14, which is what we want. After the normalization, it is verified that our mathematical calculation is consistent with the conceptual WCSR.

For $(\Delta\rho_0,\Delta r_0)$ = (0.2, 0.02), the R_f ratio between the steel and aluminum pins is 1.77. If we change $\Delta\mathbf{p}_0$ to $(\Delta\rho_0,\Delta r_0)$ = (0.1, 0.01), the R_f of the steel and aluminum pin is 1.236 and 0.695, respectively with a ratio of 1.77. If $(\Delta\rho_0,\Delta r_0)$ = (0.5, 0.05), the R_f of the steel and aluminum pin is 0.247 and 0.139, respectively, again with a ratio of 1.77. Notice how the ratios are the same in all three cases. Based on this ratio value, we conclude that the steel pin is about $(1.77)^2$ = 3.13 times more robust than the aluminum pin. ♦

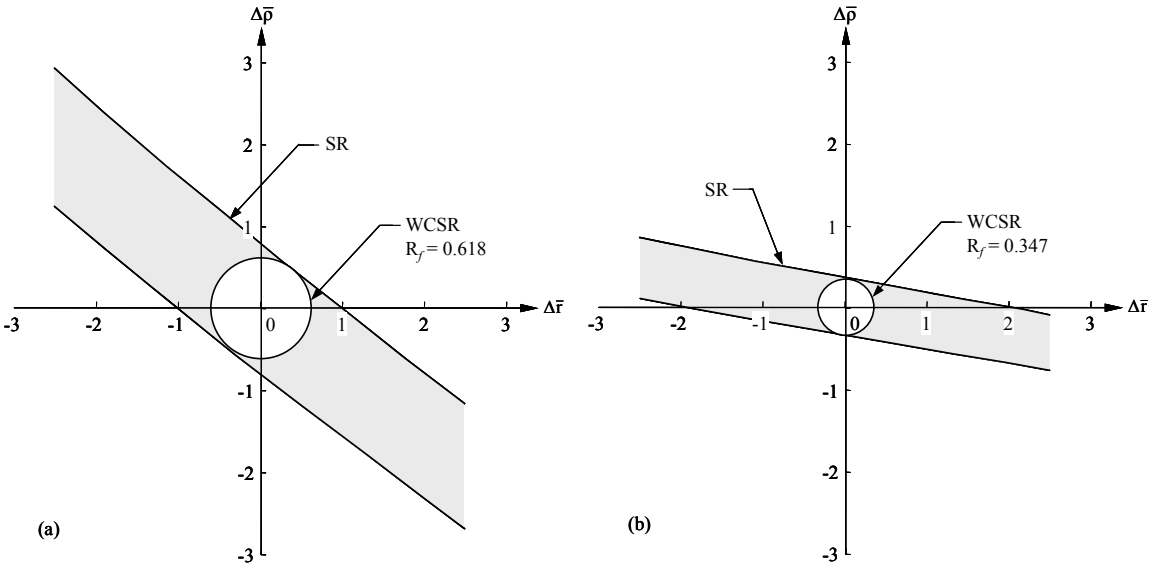


Figure 3.17: Normalized SR and WCSR of (a) steel pin, and (b) aluminum pin.

3.3. ROBUST OPTIMIZATION

Now that we have a measure of a design's robustness, we are ready to tackle the problem of finding a robust optimum design for an engineering optimization problem.

3.3.1. Robustness Index

The quantity R_f determines the ranges of $\Delta\mathbf{p}$ that a design can absorb without violating the Δf_0 limit. In other words, it defines the ranges of $\Delta\mathbf{p}$ that *must* occur if we want our design to be as robust as we specified. On the other hand, $\Delta\mathbf{p}_0$ defines the ranges of $\Delta\mathbf{p}$ that *actually do* occur in the real world. So, by comparing these two $\Delta\mathbf{p}$ ranges, we can see whether or not a design is robust. If the $\Delta\mathbf{p}_0$ ranges are larger than the R_f ranges, it means the design is not robust. There are some $\Delta\mathbf{p}$ variations that will cause the design to violate the Δf_0 limit. Conversely, if the R_f ranges are larger than the $\Delta\mathbf{p}_0$ ranges, it means the design is more robust than we need. It will never violate the Δf_0 limit, and there are

some $\Delta\mathbf{p}$ variations that the design can still absorb. If the R_f ranges are exactly the same as the $\Delta\mathbf{p}_0$ ranges, it means the design is strictly as robust as we want it to be. It strictly satisfies the Δf_0 limit.

Because R_f is defined in the normalized $\Delta\mathbf{p}$-space, the $\Delta\mathbf{p}_0$ ranges must be normalized first before we make any comparison. Since we use $\Delta\mathbf{p}_0$ to normalize R_f, we must also use the same reference to normalize $\Delta\mathbf{p}_0$ ranges. Normalization of $\Delta\mathbf{p}_0$ ranges with itself results in a unit hyper-cube in the normalized $\Delta\mathbf{p}$-space (Figure 3.18). We have also shown the interior and exterior hyper-sphere of the unit hyper-cube. The radius of the interior and exterior hyper-sphere of this hyper-cube is $R_I = 1$ and $R_E = (G)^{1/2}$, respectively. For the two-dimensional example shown in Figure 3.18, $R_I = 1$ and $R_E = 1.414$.

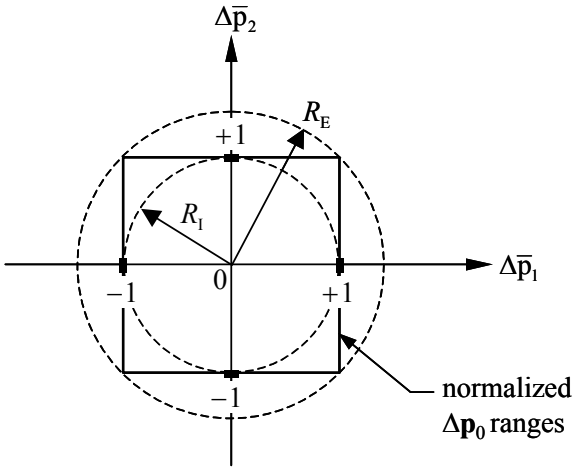


Figure 3.18: Normalized $\Delta\mathbf{p}_0$ ranges.

Direct comparison between R_f and the normalized $\Delta\mathbf{p}_0$ is difficult because R_f is a radius of a hyper-sphere while $\Delta\mathbf{p}_0$ is a range. To facilitate this comparison, we use the radius of $\Delta\mathbf{p}_0$ exterior hyper-sphere (R_E) instead. Because both R_f and R_E define a hyper-sphere, if $R_f \geq R_E$, then R_f encloses the $\Delta\mathbf{p}_0$ ranges, and our design is robust. If $R_f < R_E$, on

the other hand, then our design is not robust. If we define a quantity $\eta = \frac{R_f}{R_E}$, then a design is robust if $\eta \geq 1$, and is not robust if $\eta < 1$. We call η the "robustness index" of a design. Because $R_E = (G)^{1/2}$, the robustness index of a design is $\eta = (G)^{-1/2} R_f$.

A design is more robust if R_f is larger. Consequently, the larger η, the more robust the design. Figure 3.19 shows the comparison between η and the normalized $\Delta\mathbf{p}_0$ ranges in a two-dimensional example. As we can see in this figure, the larger η the more we enclose the $\Delta\mathbf{p}_0$ ranges, and thus the more robust the design.

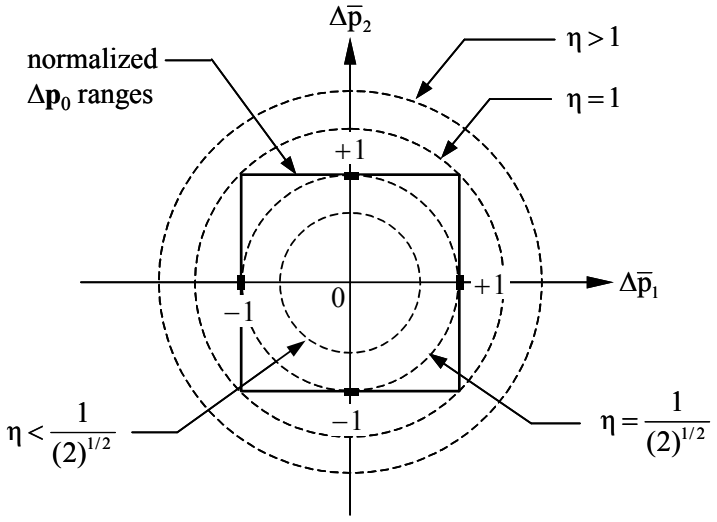


Figure 3.19: Comparison between η and $\Delta\mathbf{p}_0$.

3.3.2. Constrained Robustness Approach

In robust optimization, we want to simultaneously optimize performance of a design and maximize its robustness, where we use the objective function $f(\boldsymbol{x},\mathbf{p}_0)$ to measure the design performance, and η to measure its robustness. By doing so, we have essentially converted the original single objective optimization problem into a two-objective one.

Because of the trade-off between performance and robustness of a design, in general there is a set of optimum solutions to this two-objective problem. It is then up to the designer to select the single most preferred design from this set by trading-off the two objectives. However, the η measure that we use here does not have a physical association with the design itself and as such it would be difficult for the designer to do such a trade-off analysis. This difficulty is compounded further by the fact that η has a semi-cardinal scale.

To avoid these difficulties, in this dissertation we use a constrained approach to robustness. Instead of maximizing η, we constrain it to be greater than or equal to 1 to make sure that the WCSR of the optimum design at least encloses the $\Delta\mathbf{p}_0$ ranges. The formulation for our robust optimization problem is then as follows:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}, \mathbf{p}_0) \\ \text{subject to}: \quad & \mathrm{g}_j(\boldsymbol{x}, \mathbf{p}_0) \le 0 \quad ; \quad j = 1,...,\mathrm{J} \\ & \mathrm{h}_k(\boldsymbol{x}, \mathbf{p}_0) = 0 \quad ; \quad k = 1,...,\mathrm{K} \\ & 1 - \eta \le 0 \end{aligned} \tag{3.16}$$

where the last constraint is the robustness constraint, $\eta = (\mathrm{G})^{-1/2}\mathrm{R}_f$, and R_f is calculated by solving an inner optimization shown in Eq. (3.12) (re-shown here for convenience).

$$\begin{aligned} \underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \quad & \mathrm{R}_f(\Delta\overline{\mathbf{p}}) = \left[\sum_{i=1}^{\mathrm{G}} (\Delta\overline{\mathrm{p}}_i)^2\right]^{\frac{1}{2}} \\ \text{subject to}: \quad & 1 - \frac{\left[\Delta f(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)\right]^2}{[\Delta f_0]^2} = 0 \end{aligned} \tag{3.17}$$

Notice that although not explicitly shown in Eq. (3.16), η is a function of $\boldsymbol{x}$, the design variable, because R_f is a function of $\boldsymbol{x}$. In Eq. (3.17), $\Delta f(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)$ is a function of $\boldsymbol{x}$.

Using Eq. (3.16), our search for a robust optimum design is performed in the N-dimensional $\boldsymbol{x}$-space, in which during this search, we run another search, Eq. (3.17), in the G-dimensional $\Delta\overline{\mathbf{p}}$-space. Although our approach involves two optimization problems, this is not a multi-disciplinary optimization approach. Rather, Eq. (3.17) is simply a tool to obtain the robustness information needed by Eq. (3.16). As such, convergence of this approach depends entirely on the convergence of Eq. (3.16), and has nothing to do with the links between the two problems. If Eq. (3.16) has an optimum solution before the robustness constraint is added, but does not have a feasible solution after it is added, it simply means that there is no design that satisfies our robustness requirement (i.e., the addition of the robustness constraint causes the feasible domain of Eq. (3.16) to become empty).

Eq. (3.17) is a simple single objective optimization problem with one constraint. If the function $f(\boldsymbol{x},\mathbf{p})$ is simple enough, Eq. (3.17) can be solved analytically like in our piston pin examples. Otherwise, traditional gradient-based optimization methods, such as Quasi-Newton methods or the Generalized Reduced Gradient method, can be used to solve it. The choice of an optimizer for Eq. (3.16) depends on the complexity of the problem. Nevertheless, because of the nature of the robustness constraint, we recommend using global optimization methods to solve Eq. (3.16). Unlike conventional constraints, the robustness constraint in Eq. (3.16) may not necessarily divide the search space into two well-defined feasible and infeasible half-spaces. Rather, it may result in many disjointed infeasible domains in the feasible domain, in effect adding "holes" to the original feasible domain (Figure 3.20). Because of this effect, direction-based optimizers may have difficulty converging.

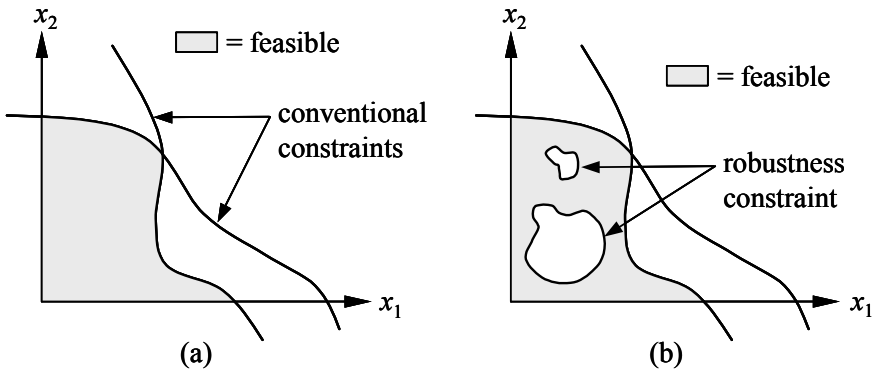


Figure 3.20: Effect of adding robustness constraint.

Sometimes, the constraint $\eta \geq 1$ is too strict, that is no design has that much robustness. In this case, the right hand side value of 1 needs to be lowered. To give the designer the flexibility to change the desired level of robustness, we use a quantity η_0 to replace this value of 1, where η_0 is determined by the designer. The larger η_0, the more robust the designer wants the optimum design to be, and vice versa. Using η_0 in the robustness constraint, our robust optimization formulation becomes:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(\boldsymbol{x}, \mathbf{p}_0) \\ \text{subject to}: \quad & \mathrm{g}_j(\boldsymbol{x}, \mathbf{p}_0) \leq 0 \quad ; \quad j = 1,...,\mathrm{J} \\ & \mathrm{h}_k(\boldsymbol{x}, \mathbf{p}_0) = 0 \quad ; \quad k = 1,...,\mathrm{K} \\ & 1 - \frac{\eta}{\eta_0} \leq 0 \end{aligned} \tag{3.18}$$

where again η is calculated by solving Eq. (3.17).

3.4. COMPARISON STUDY

To demonstrate our robust optimization method, we applied it to four examples: one numerical example and three engineering examples. The purpose of the numerical example is to provide a graphical verification of the results obtained by our method. The

purpose of the three engineering examples is as follows. The welded beam example demonstrates an application of our method to a non-differentiable objective function. The three-bar truss and compression spring examples compare our method to mean-based and worst case analysis-based robust optimization methods, respectively.

3.4.1. Wine-Bottle Function

This example is originally formulated by Van Veldhuizen and Lamont (1998) as a multi-objective optimization problem. We converted it into a single objective problem by significantly modifying and optimizing one of the original objectives. The problem has two variables x_1 and x_2, both continuous, and the objective is to minimize the "wine-bottle" function $f(x_1,x_2)$. There are variations in the variables (Δx_1 and Δx_2), and we want to minimize the sensitivity of the optimum solution with respect to these variations. Because the variability occurs only in the design variables (there are no other noisy factors), in this problem $\mathbf{p} = \boldsymbol{x}$ and $\Delta\overline{\mathbf{p}} = \Delta\overline{\boldsymbol{x}}$. The mathematical formulation of the problem is as follows:

$$\begin{aligned} &\text{minimize } f(x_1,x_2) = \frac{\mathrm{C}+1}{9.32} \\ &\text{where: } \quad \mathrm{C} = \frac{\mathrm{A}^2+\mathrm{B}^2+3}{2} + \sin(\mathrm{A}^2+\mathrm{B}^2+2) \\ &\qquad\quad \mathrm{A} = 6x_1 - 3 \quad ; \quad \mathrm{B} = 6x_2 - 3 \\ &\qquad\quad 0 \le (x_1,x_2) \le 1 \end{aligned} \tag{3.19}$$

Figure 3.21 shows a three-dimensional surface mesh of the objective function. As shown in this figure, the function is axially symmetric and has a dome-like region in the center whose shape resembles the bottom of a wine bottle. The function has an infinite number of global minima circularly located around the center point $(x_1,x_2) = (0.5,0.5)$,

and they are defined by a circle equation: $(x_1-0.5)^2 + (x_2-0.5)^2 = (0.246)^2$. The objective value at the global minima is $f^* = 0.292$. Points on the dome-region of the function have a slightly higher objective value ($f_R = 0.365$) compared to the global minima. However, this region of the function is flat and thus the points in this region are insensitive to the variable variations.

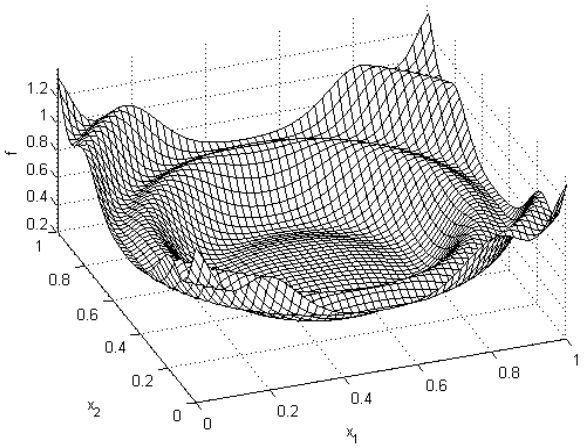


Figure 3.21: Surface plot of the wine-bottle function.

The maximum allowable variation in f is $\Delta f_0 = 0.01$, and the variation ranges are known to be $\Delta\mathbf{p} = (0.05,0.05)$. We added the robustness constraint into Eq. (3.19), and then solve it using a Genetic Algorithm (GA) for the outer optimization problem, and MATLAB's *fmincon* function for the inner optimization problem (Eq. (3.17)). MATLAB's *fmincon* function is an optimizer based on the Sequential Quadratic Programming (SQP) method with BFGS formula for the Hessian estimation. We used a GA for the outer optimization because the objective function has multiple local minima (see Figure 3.21) so that direction-based optimizers may not find the global minima.

Because GA is a stochastic method, the robust optimum obtained might differ from one run to another. To account for this, we solved the problem 50 times (using different

initial population each time). Figures 3.22 and 3.23 show the robust optima obtained from solving the problem 50 times each for three different values of η_0. Figure 3.22 shows the distributions of the optima on the contour plot of the objective function, while Figure 3.23 shows those points of the distributions that lie on the cross-section of the function when $x_2 = 0.5$.

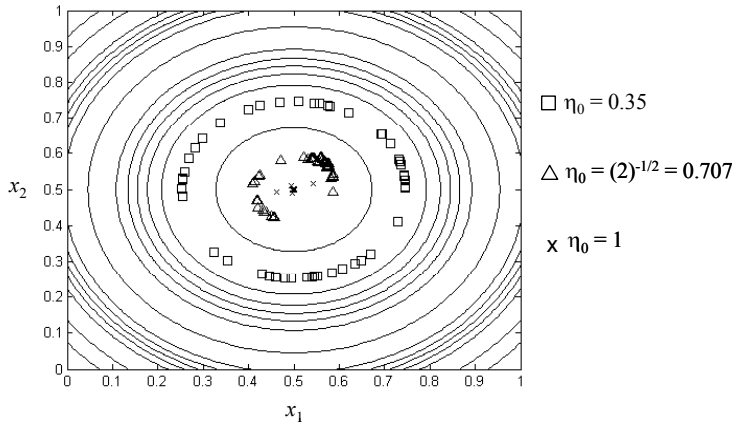


Figure 3.22: Distributions of robust optima on a contour plot.

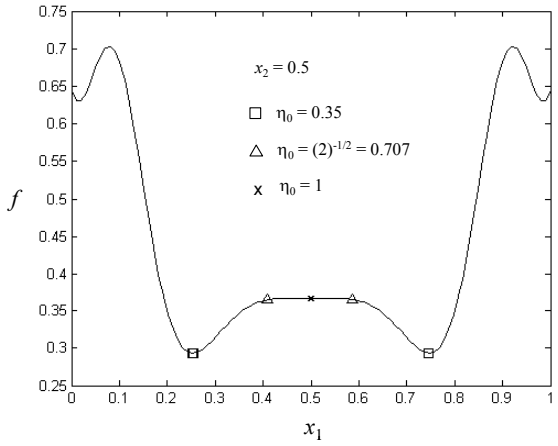


Figure 3.23: Distributions of robust optima on a cross-section plot.

We observe in Figure 3.22 and 3.23 that for $\eta_0 = 1$, the optimum obtained is located at the dome region of the function for all 50 GA runs. Because this region is insensitive to the variations, these results confirm that solving the problem using our robust

optimization method results in a robust optimum. We also observe from Figures 3.22 and 3.23 that as η_0 is reduced (relaxing the robustness constraint), the objective value decreases (gets better). The optimum objective value for $\eta_0 = 1$ is $f_1 = 0.365$, while $f_{0.707} = 0.361$ and $f_{0.35} = 0.292$. When $\eta_0 = 0.35$, the robust optimum obtained is the global optimum. We expect to observe these phenomena because in general there are trade-offs between performance and robustness.

3.4.2. Design of a Three-Bar Truss

This example was first formulated by Schmit (1960), and has been thoroughly studied in (Sun et al., 1975) and (Haug and Arora, 1979). Here, we modified the problem by substituting the original objective with one of the structural constraints and by adding some variations into the problem.

In this problem, we are designing a system of three-bar truss with a constant force P = 100 kN acting at an angle θ at the end of the truss as shown in Figure 3.24 (l = 1.0 m). The truss is symmetric (member 1 and member 3 are identical), and is to be designed for minimum stress in members 1 and 3 (A_1), and the angle θ.

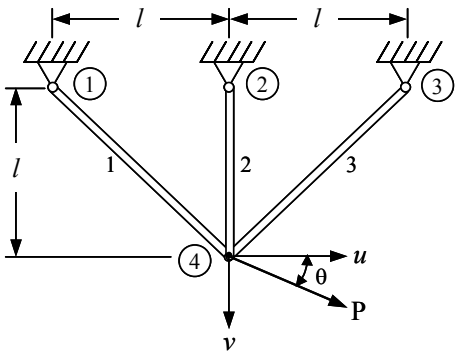


Figure 3.24: A three-bar truss.

To prevent failure, the design is subjected to 6 structural constraints. The horizontal and vertical deflection at node 4 (u and v) must be less than $u_a = 0.5$ cm and $v_a = 0.5$ cm, respectively. The stress on member 2 (σ_2) must be less than $\sigma_a = 140$ MPa. The buckling load on all members must satisfy the buckling constraint. In addition, the total volume of the members is constrained to be less than $V_a = 2000\ \text{cm}^3$. The mathematical formulation of the problem is given in Eq. (3.20).

$$
\begin{aligned}
&\underset{A_1,\theta}{\text{minimize}}\ \ \sigma = \frac{P}{\sqrt{2}}\left[\frac{\cos(\theta)}{A_1} + \frac{\sin(\theta)}{(A_1 + \sqrt{2}A_2)}\right] \\
&\text{subject to}: \quad g_1 \equiv \frac{u}{u_a} - 1 \le 0 \quad ; \quad g_2 \equiv \frac{v}{v_a} - 1 \le 0 \\
&\qquad\qquad g_3 \equiv \frac{\sigma_2}{\sigma_a} - 1 \le 0 \quad ; \quad g_4 \equiv \frac{F_1}{F_{b1}} - 1 \le 0 \\
&\qquad\qquad g_5 \equiv \frac{F_2}{F_{b2}} - 1 \le 0 \quad ; \quad g_6 \equiv \frac{F_3}{F_{b1}} - 1 \le 0 \\
&\qquad\qquad g_7 \equiv \frac{V}{V_a} - 1 \le 0 \\
&\qquad\qquad 30^\circ \le \theta \le 40^\circ \quad ; \quad 2\,\text{cm}^2 \le A_1 \le 10\,\text{cm}^2
\end{aligned}
\tag{3.20}
$$

The constraints are calculated by the following formulas. In calculating the buckling load, the members are considered to be columns with pins at both ends.

$$u = \frac{\sqrt{2}lP\cos(\theta)}{A_1 E} \quad ; \quad v = \frac{\sqrt{2}lPs\sin(\theta)}{(A_1 + \sqrt{2}A_2)E} \tag{3.21}$$

$$\sigma_2 = \frac{\sqrt{2}P\cos(\theta)}{(A_1 + \sqrt{2}A_2)} \quad ; \quad F_1 = -\frac{P}{\sqrt{2}}\left[\frac{\cos(\theta)}{A_1} + \frac{\sin(\theta)}{(A_1 + \sqrt{2}A_2)}\right] \tag{3.22}$$

$$F_{b1} = \frac{\pi^2 E\beta A_1}{2l^2} \quad ; \quad F_2 = -\frac{\sqrt{2}P\sin(\theta)}{(A_1 + \sqrt{2}A_2)} \tag{3.23}$$

$$F_{b2} = \frac{\pi^2 E\beta A_2}{l^2} \quad ; \quad F_3 = -\frac{P}{\sqrt{2}}\left[\frac{\sin(\theta)}{(A_1 + \sqrt{2}A_2)} - \frac{\cos(\theta)}{A_1}\right] \tag{3.24}$$

$$V = l(2\sqrt{2}A_1 + A_2) \tag{3.25}$$

where: E = Young's modulus (= 70 GPa)

A_2 = cross section area of member 2 (= 2 cm^2)

β = non-dimensional constant (= 1.0)

Using MATLAB's *fmincon* to solve Eq. (3.20), we obtained the nominal optimum design $(A_1^*, \theta^*) = (6.36, 40)$, and $\sigma^* = 134.56$ MPa in just 3 iterations. The constraint values of this nominal optimum are: $\mathbf{g} = (-0.5133, -0.7173, -0.2934, -1.6128, -1.716, -0.8375, 0.0)$, where g_7 is active.

The two design variables are known to vary by $(\Delta A_1, \Delta\theta) = (0.1\ cm^2, 5°)$, and we would like to minimize the sensitivity of the optimum design with respect to these variations. The allowable variation in the objective is $\Delta\sigma = 2.75$ MPa. We added the robustness constraint (with $\eta_0 = 1$) into Eq. (3.20) and then solved it using *fmincon* for both the inner and outer optimization problems. On average (we ran the algorithm using many different initial points) the outer optimization converges in 9 iterations. On average, the inner optimization converges in 13 iterations. The robust optimum design obtained is shown in Table 3.1.

For comparison, we also solved Eq. (3.20) for the robust optimum using two conventional robust optimization approaches: (i) minimizing the mean value of the objective function, and (ii) minimizing the sum of mean and standard deviation. The mean and standard deviation are calculated by performing 10,000 Monte Carlo simulations around the design following a uniform pdf of the ranges $(\Delta A_1, \Delta\theta) = (0.1\ cm^2, 5°)$. The optimum designs obtained using these two methods are also shown in Table 3.1.

Table 3.1: Optimum designs of the three-bar truss.

	Nominal Optimum	Robust Optimum	Mean Optimum	Mean+Std Optimum
σ (Mpa)	134.56	135.08	134.69	135.06
η	0.826	1.0	0.832	0.885
A_1 (cm^2)	6.364	6.364	6.36	6.35
θ (degree)	40	36.3	30	38.6
F_{call}	N/A	40	10000	10000

In Table 3.1, the quantity F_{call} is the number of function evaluations needed per design to obtain its robustness information. For our method, this quantity is the number of function evaluations performed in solving the inner optimization. For the mean-std methods, this quantity is the number of samples. In this table we have also shown the η value of each optimum design (remember, each design has a corresponding η value even though it is not obtained by our robust optimization method).

We see in Table 3.1 that the nominal optimum has the lowest σ value (the objective function), but it also has the lowest η value. In contrast, our robust optimum has the highest σ value of the four optima, but it has the highest η. The σ and η values of the other two optima are somewhere in between. This observation is just what we expected. The larger η, the more robust the design, but at the expense of performance degradation (i.e., stress increases). We also see in this table that of the four optima, only the η value of our robust design is equal to 1. So, according to our theoretical development, when the variations occur, our robust design should be the only one that satisfies the requirement $\Delta\sigma \leq 2.75$ MPa. Let us verify this claim.

We randomly perturbed (A_1, θ) 20 times around their nominal values within the given ranges, and calculated the new σ value of the design using these perturbed values.

The difference between this new σ value and the previous value is the Δσ. We did the same analysis for all four optima, using the same (ΔA_1, Δθ) for each. Figure 3.25 shows the graphs of the Δσ of the optima for the 20 random perturbations. In this figure the dashed lines are the $\Delta\sigma_0$ limit (2.75 MPa).

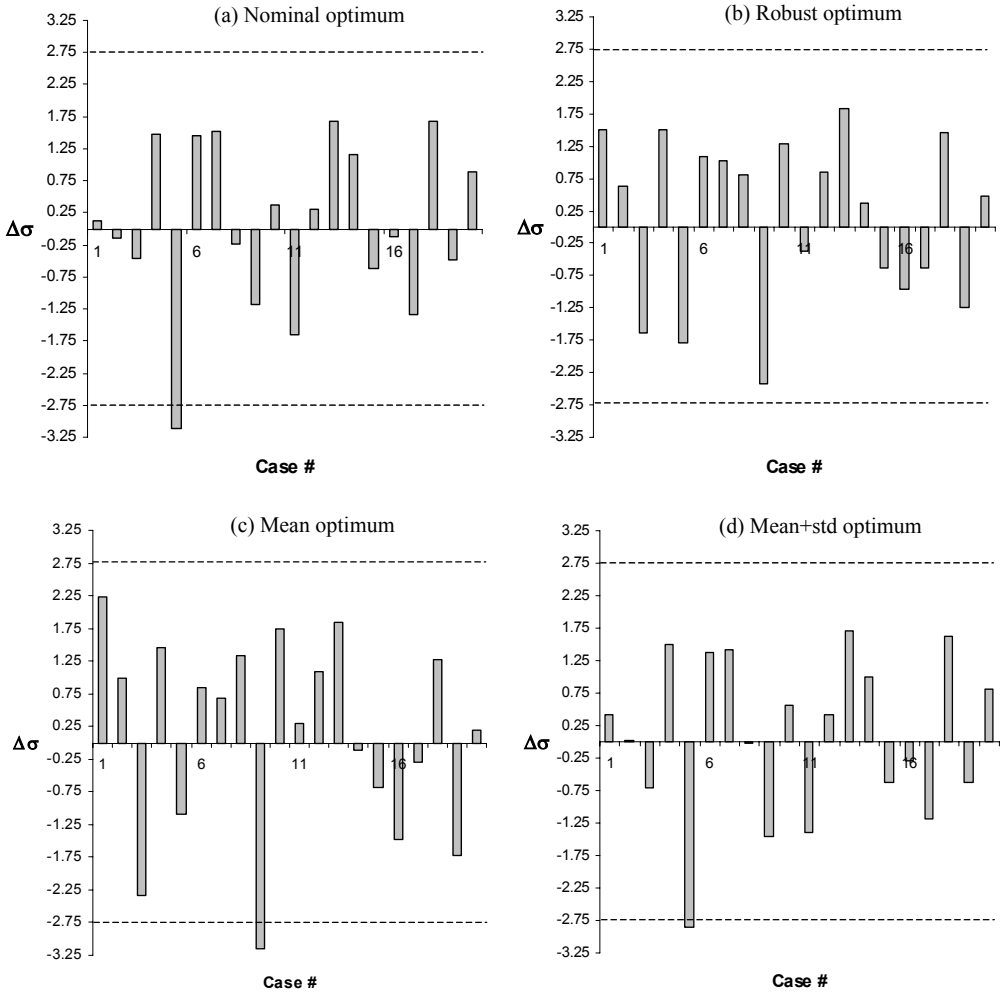


Figure 3.25: Sensitivity analysis of the three-bar truss optima.

We observe in Figure 3.25 that when the variations occur, the nominal optimum design as well as the mean and mean+std optima violates the allowable Δσ limit. Only our robust optimum design stays within the limit, thus showing that this design satisfies our robustness requirement.

To further validate the robustness information provided by the η value, we calculated the SR and WCSR of each of the optima, and compared them to the $(\Delta A_1, \Delta\theta)$ range. Since the objective function involves trigonometric expressions, analytical derivation of SR is impossible, so we derived them numerically instead. We constructed an orthogonal grid in the range $(\Delta A_1, \Delta\theta) = (\pm 0.2\ \text{cm}^2, \pm 10°)$, and calculated the $\Delta\sigma$ of the optimum designs at each junction. The SR is obtained by determining if $(\Delta\sigma)^2 \le (\Delta\sigma_0)^2$ at these junctions, and the WCSR is obtained by finding the point on the SR boundary closest from origin. The SR and WCSR of the four optima are shown in Figure 3.26 through Figure 3.29. In these figures, the dashed rectangles are the known $(\Delta A_{1,0}, \Delta\theta_0)$ range.

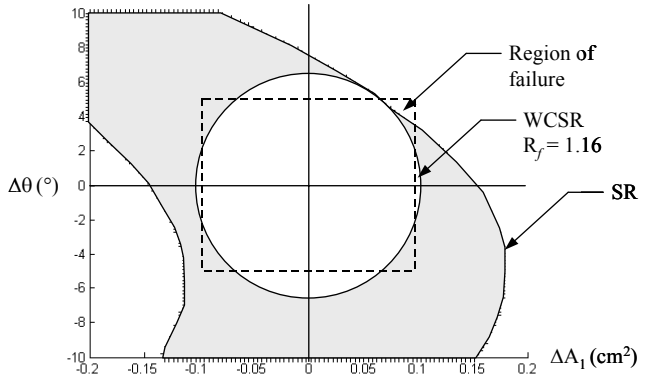


Figure 3.26: SR and WCSR of the nominal optimum.

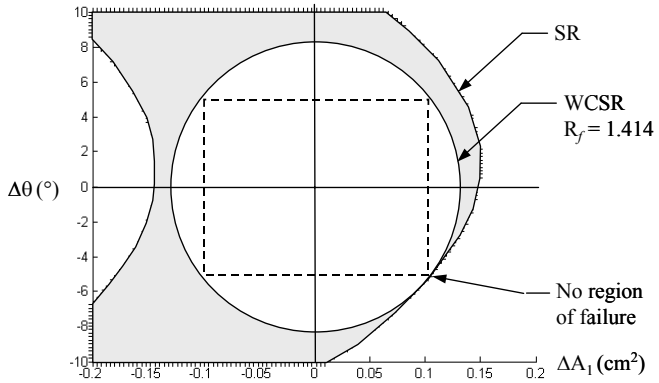


Figure 3.27: SR and WCSR of the robust optimum.

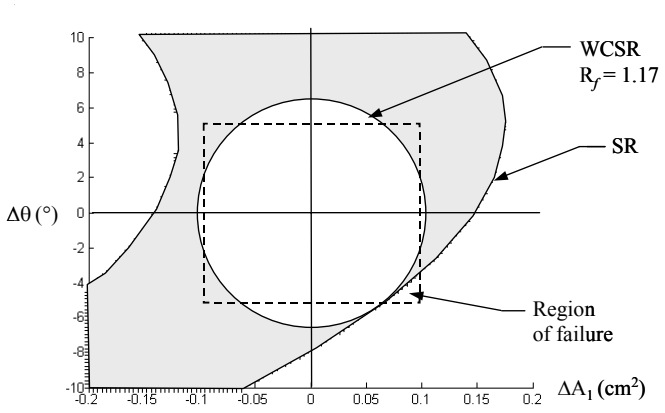


Figure 3.28: SR and WCSR of the mean optimum.

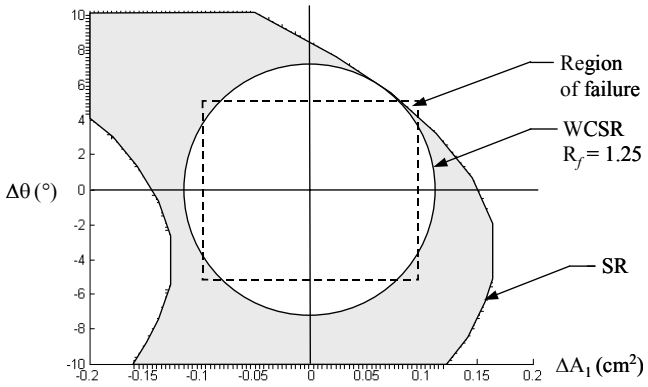


Figure 3.29: SR and WCSR of the mean+std optimum.

As we can see in the above figures, only the SR of our robust optimum design fully encloses the $(\Delta A_{1,0}, \Delta\theta_0)$ range. The other designs have small regions for which the requirement $(\Delta\sigma)^2 \leq (\Delta\sigma_0)^2$ is not satisfied. The WCSR's of the designs also reflect this observation. Only the WCSR of our robust optimum encloses the $(\Delta A_{1,0}, \Delta\theta_0)$ range completely.

3.4.3. Design of A Welded Beam

This example is the well-known welded beam problem originally formulated by Ragsdell and Phillips (1976) and Reklaitis et al. (1983). We slightly modified this

problem by adding variations to two of the parameters, and by making the objective function discontinuous with respect to one of these parameters.

In this problem, a beam A is to be welded to a rigid support member B. The beam has a rectangular cross-section and is to be made out of steel. The beam is designed to support a force F = 6000 lbf acting at the tip of the beam, and there are constraints on the shear stress, normal stress, deflection, and buckling load on the beam. The problem has four (4) continuous design variables, and they are: thickness of the weld (*h*), length of the weld (*l*), thickness of the beam (*t*), and width of the beam (*b*). All variables are in inches. The objective of the problem is to minimize the total cost of making such an assembly. Figure 3.30 shows a diagram of the welded beam assembly.

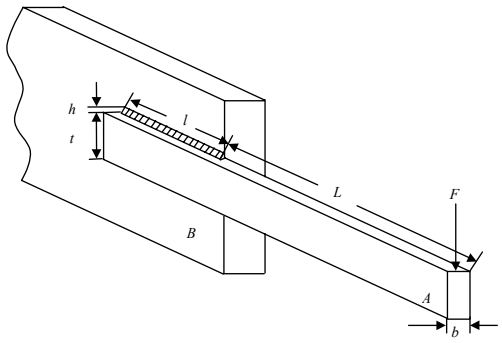


Figure 3.30: A welded beam assembly.

The complete formulation of the problem is shown in Eq. (3.26).

$$
\begin{aligned}
&\text{minimize} \quad f_{\text{cost}} = (1+\mathrm{c}_3)h^2 l + \mathrm{c}_4 tb(\mathrm{L}+l) \\
&\text{subject to}: \mathrm{g}_1 \equiv \frac{\tau}{\tau_d} - 1 \le 0 \qquad \mathrm{g}_2 \equiv \frac{\sigma}{\sigma_d} - 1 \le 0 \\
&\qquad\qquad \mathrm{g}_3 \equiv \frac{\delta}{0.25} - 1 \le 0 \qquad \mathrm{g}_4 \equiv \frac{\mathrm{F}}{P_c} - 1 \le 0 \\
&\qquad\qquad \mathrm{g}_5 \equiv \frac{h}{b} - 1 \le 0 \qquad \mathrm{g}_6 \equiv \frac{0.125}{h} - 1 \le 0 \\
&\qquad\qquad 0.1 \le h \le 2.0 \qquad 0.1 \le l \le 10.0 \\
&\qquad\qquad 0.1 \le t \le 10.0 \qquad 0.1 \le b \le 2.0
\end{aligned}
\tag{3.26}
$$

where:

c_3 = cost of weld material (\$0.1047 /inch^3)
c_4 = cost of beam material (\$0.0481/inch^3)
τ = maximum shear stress in weld (psi)
τ_d = allowable shear stress of weld (13,600 psi)
σ = maximum normal stress in beam (psi)
σ_d = allowable normal stress in beam (30,000 psi)
δ = deflection at beam end (inch)
P_c = allowable buckling load (lbf)
L = length of unwelded beam (14 inch)

The quantities τ, σ, δ, and P_c are calculated as shown in Eq. (3.27) through Eq. (3.33). To calculate δ, it is assumed that the beam is a cantilever beam with length L, and for steel G = 12×10^6 psi and E = 30×10^6 psi.

$$\tau = \sqrt{(\tau')^2 + 2\tau'\tau''\cos\theta + (\tau'')^2} \tag{3.27}$$

$$\tau' = \frac{\mathrm{F}}{\sqrt{2}hl} \quad ; \quad \tau'' = \frac{MR}{J} \quad ; \quad \cos\theta = \frac{l}{2R} \tag{3.28}$$

$$M = \mathrm{F}\left(\mathrm{L} + \frac{l}{2}\right) \quad ; \quad R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2} \tag{3.29}$$

$$J = 2\left\{0.707hl\left[\frac{l^2}{12} + \left(\frac{h+t}{2}\right)^2\right]\right\} \tag{3.30}$$

$$\sigma = \frac{6\mathrm{FL}}{bt^2} \qquad ; \qquad \delta = \frac{4\mathrm{FL}^3}{\mathrm{E}t^3 b} \tag{3.31}$$

$$P_c = \frac{4.013\sqrt{\mathrm{EI}\alpha}}{\mathrm{L}^2}\left(1 - \frac{t}{2\mathrm{L}}\sqrt{\frac{\mathrm{EI}}{\alpha}}\right) \tag{3.32}$$

$$\mathrm{I} = \frac{1}{12}tb^3 \quad ; \quad \alpha = \frac{1}{3}\mathrm{G}tb^3 \tag{3.33}$$

The parameters L and c_3 vary by some amount ΔL and Δc_3, and they affect the total cost of the welded beam assembly as shown in Eq. (3.34) (the notation $\lceil \bullet \rceil$ is for rounding up to the nearest integer). Notice that Eq. (3.34) reduces to the original f_{cost} in Eq. (3.26) when $\Delta L = 0$ and $\Delta c_3 = 0$.

$$f_{cost} = (1 + c_3 + \Delta c_3)h^2 l + c_4 tb(L + l) + c_5 l$$
$$\text{where} \quad c_5 = \$0.01 \left\lceil \frac{|\Delta L|}{0.05} \right\rceil \tag{3.34}$$

We want to find a design that minimizes f_{cost} and is insensitive to ΔL and Δc_3. We added the robustness constraint into Eq. (3.26), and we set $\Delta f_0 = \$0.30$ and $\Delta \mathbf{p}_0 = (0.05, 0.25)$. In this problem, $\mathbf{p} = (c_3, L)^t$ and $\mathbf{p}_0 = (0.1047, 14)^t$. We solved the problem using a GA for the outer and inner problems (with $\eta_0 = 1$), and obtained a robust optimum value of $f_R = \$2.49$. For comparison, we solved Eq. (3.26) using the same GA, and obtained a nominal (non-robust) optimum of $f_{cost} = \$2.39$ (which is very close to that reported by Ragsdell and Phillips (1976): $f_{cost} = \$2.38$, and by Deb (1991): $f_{cost} = \$2.43$).

For further comparison, we also solved Eq. (3.26) for robust optimum by minimizing the mean value of the design. The mean value was calculated using Monte Carlo simulation assuming uniform and normal probability distribution for the parameters. For the uniform assumption, Δc_3 distribution is modeled between [-0.05, 0.05] while ΔL distribution is modeled between [-0.25, 0.25]. For the normal assumption, Δc_3 distribution is modeled to have a mean and standard deviation of [0, 0.05/3] while ΔL distribution is modeled by [0, 0.25/3].

Table 3.2 shows a list of the optimum designs obtained (for the Monte Carlo optima, the f_{cost} shown are the mean values). The quantity F_{call} is the number of function

evaluations needed per design to obtain the sensitivity information. As shown in Table 3.2, making different assumptions about the pdf of the uncertain parameters can lead to significantly different results. We also observe in Table 3.2 that the solution obtained by our method is not the same as the one obtained by the Monte Carlo method for a uniform pdf. This shows that although our method uses a range of parameter variations, it is not the same as assuming a uniform pdf for the parameters. This in fact is an advantage of our method in that it does not require a presumed pdf of the uncertain parameters. In Chapter 7 we will show how to use the probability distribution information, if they are available, with our robust optimization method.

Table 3.2: Optimum designs of the welded beam.

	Nominal Optimum	Robust Optimum	Monte Carlo (Uniform)	Monte Carlo (Normal)
f_{cost}	2.39	2.49	2.63	2.91
h	0.241	0.246	0.257	0.337
l	6.158	5.461	5.8	5.054
t	8.5	9.138	8.267	7.058
b	0.243	0.248	0.257	0.337
F_{call}	N/A	250	100000	100000

Figure 3.31 shows the SR and WCSR of the nominal optimum and the robust optimum obtained using our method. In this figure, the shaded region bounded between the straight line on the left and the step function on the right is the SR, the ellipse is the WCSR, and the dashed rectangle is the $\Delta\mathbf{p}_0$ ranges. We obtain the SR's analytically by substituting the values in Table 3.2 into the f_{cost} function in Eq. (3.34) and obtain the Δf_{cost} as a function of Δc_3 and ΔL. Then using the given Δf_0 value, we solve for Δc_3 at the six discrete ΔL values. Mathematically, the left boundary of the sensitivity region is also a step function similar to that of the right boundary. However, the parameter c_3 represents a

material cost, so in reality it cannot be negative. For this reason we restrict the Δc_3 value to be –0.10471 at the lowest (the straight line left boundary). We obtain the WCSR's by solving the inner optimization problem using the values in Table 3.2. The WCSR's shown are ellipses (instead of circles) because Figure 3.31 shows the regions in the non-normalized $\Delta\mathbf{p}$-space (for clarity of presentation).

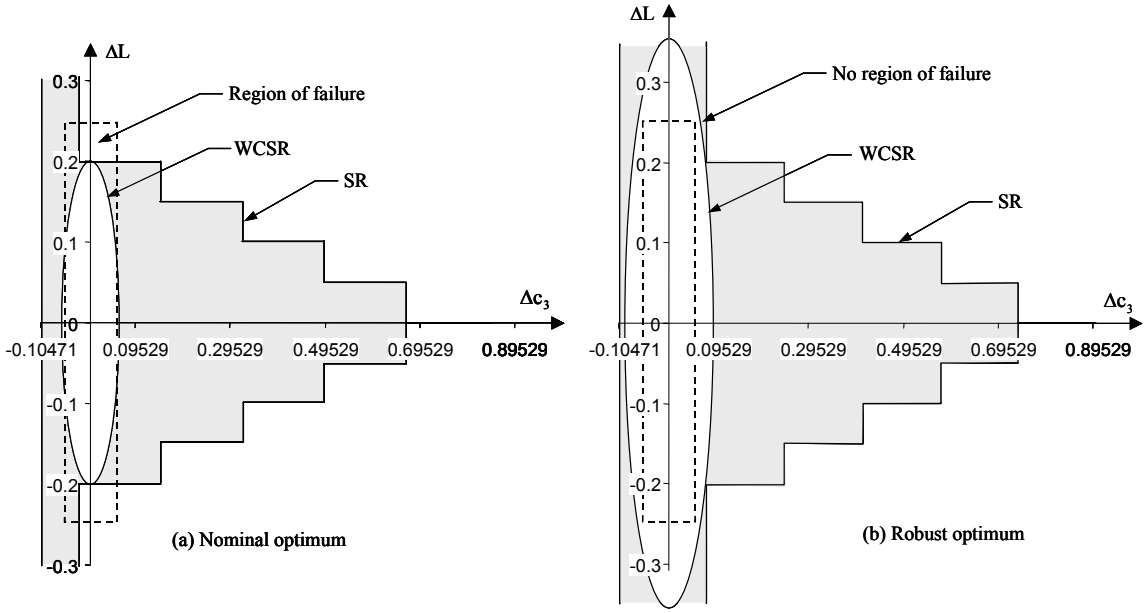


Figure 3.31: SR and WCSR of the optima: (a) nominal, (b) robust.

Notice that the SR's are unbounded. This is because for the range of Δc_3 values where the regions are unbounded, the value $c_5 = \$0.05$ ($|\Delta L|>0.20$) still makes the inequality $(\Delta f)^2 \le (\Delta f_0)^2$ satisfied. Notice also that the SR boundary is discontinuous because f_{cost} is a step function. However, at the discontinuity, $(\Delta f)^2 = (\Delta f_0)^2$ is satisfied either from the left or from the right, so our SR boundary condition $(\Delta f)^2 = (\Delta f_0)^2$ is valid (recall Figure 3.3).

We observe from Figure 3.31 that the SR of the nominal optimum does not fully enclose the $\Delta\mathbf{p}_0$ ranges while the robust optimum's does. This remains true when using

the WCSR's as estimates of the SR's. We conclude that in the worst case sense, the robust optimum is less sensitive to $\mathbf{p} = (c_3, L)^t$ than the nominal optimum.

To further assess the sensitivity of the robust optimum design obtained, we calculated the f_{cost} value of the design for 15 perturbed values of the two uncertain parameters (c_3 and L), and determined how much the f_{cost} differed from the unperturbed f_{cost} value (Table 3.2). For comparison, we performed the same analysis (using the same Δc_3 and ΔL values) to calculate the Δf_{cost} of the nominal and the Monte Carlo optima. In this analysis, the values for Δc_3 and ΔL were randomly sampled from the $\Delta \mathbf{p}_0$ range.

Figure 3.32 and Figure 3.33 show the graphs of the Δf_{cost} of the optimum designs for the 15 cases. The dashed-lines in these figures are the Δf_0. We observe in Figure 3.32 that in all 15 cases the Δf_{cost} of the robust optimum is less than that of the nominal optimum, i.e., $\Delta f_R < \Delta f_N$. This shows that the robust optimum obtained by our method is less sensitive to $\Delta \mathbf{p}$ than the nominal optimum. In addition, we also observe that in all 15 cases, the Δf_{cost} of the robust optimum stays within the acceptable bound, while the nominal optimum violates the bound in cases 6 and 15. In Figure 3.3 we observe that both Monte Carlo optima also violate the bound in cases 6 and 15.

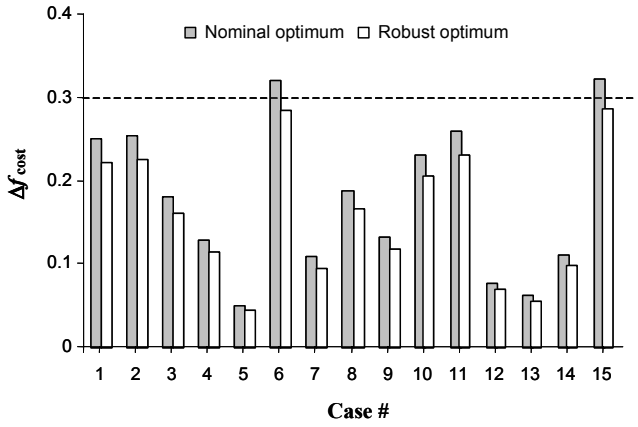


Figure 3.32: Sensitivity analysis of the welded beam optima.

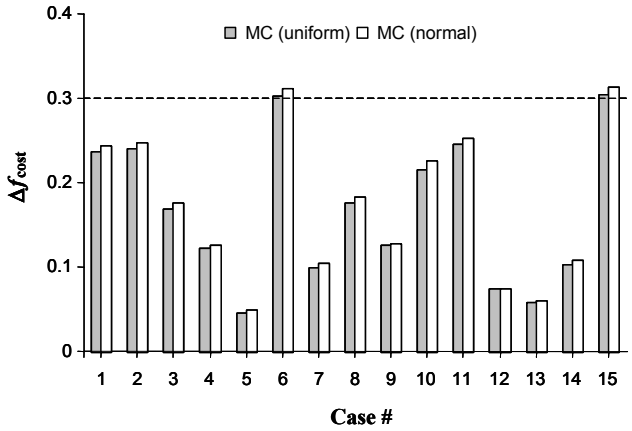


Figure 3.33: Sensitivity analysis of the welded beam optima.

In this chapter, our robust optimization method looks only at the objective robustness of a design, and we implicitly assume that the robust optimum design remains feasible when the variations occur. To verify the validity of this assumption, we calculated the constraints of the robust optimum for the 15 random cases. The results are shown in Table 3.3. In this table we calculated only the constraints g_1, g_2, g_3, g_4 because g_5 and g_6 are independent of c_3 and L. As shown in Table 3.3, in all 15 cases the constraints for the robust optimum are still satisfied ($\mathbf{g} \leq 0$), so our feasibility assumption holds.

Table 3.3: Constraints of the robust optimum design.

Case #	g1	g2	g3	g4
1	-0.016	-0.204	-0.956	-0.134
2	-0.004	-0.190	-0.954	-0.109
3	-0.004	-0.190	-0.954	-0.109
4	-0.014	-0.201	-0.955	-0.128
5	-0.010	-0.196	-0.955	-0.120
6	-0.011	-0.198	-0.955	-0.124
7	-0.008	-0.194	-0.954	-0.116
8	-0.014	-0.201	-0.955	-0.128
9	-0.002	-0.188	-0.953	-0.105
10	-0.017	-0.205	-0.956	-0.135
11	-0.009	-0.195	-0.955	-0.119
12	-0.004	-0.190	-0.954	-0.109
13	-0.016	-0.204	-0.956	-0.134
14	-0.014	-0.201	-0.955	-0.129
15	-0.015	-0.202	-0.956	-0.131

Interestingly, if we look into the constraints of the nominal optimum, we will find that constraints g_1, g_2, g_3, g_4 of this design are active or nearly active, and when c_3 and L change, g_1 and g_4 are violated. So, in some sense the robust optimum provides design robustness not only in terms of the objective, but in terms of feasibility also; although obviously this observation is not general and is valid for this particular example only. We will discuss our feasibility robustness approach in Chapter 5.

3.4.4. Design of a Compression Spring

This example is from Arora (2001) and has been modified to demonstrate the application of our robust optimization method. Consider a coil spring loaded in compression as shown in Figure 3.34. Before the force P = 10 lbf is applied, the spring is at its free length. The spring is to be used as an energy-storing device, and we are designing it to have as large a restoring force as possible by maximizing the axial deflection δ. There are three variables that affect the design: the wire diameter (d), the mean coil diameter (D), and the number of active coils (N). The wire diameter and the mean coil diameter are measured in inches, and the number of active coils must be an integer between 2 and 15.

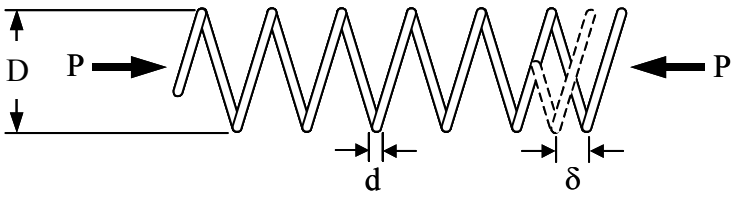


Figure 3.34: A compression spring.

To guarantee a proper design, five (5) constraints are imposed. The wires of a spring under compression experience twisting, so to prevent shear failure, the shear stress due to

this twist is constrained to be less than $\tau_A = 80$ ksi. In case of a dynamic load P, we would like to avoid resonance by requiring that the surge wave frequency of the spring is greater than $\omega_A = 100$ Hz. The outer diameter of the spring is constrained to be less than $OD_A = 1.5$ in, and the total mass of the spring is constrained to be less than $M_A = 2.309 \times 10^{-5}$ lbm. To prevent an unrealistic design, the spring deflection is constrained to be less than $\delta_A = 0.75$ in (objective constraint). In addition to the design constraints, there is also a lower and upper bound constraints on the design variables.

The mathematical formulation of the problem is shown in Eq. (3.34). In this formulation: $G = 1.15 \times 10^7$ psi is the shear modulus, $Q = 2$ is the number of inactive coils, and $\rho = 7.383 \times 10^{-4}$ lbm/in^3 is the mass density. For a more detailed analysis of a spring design problem, consult Spott (1953) or Wahl (1963).

$$
\begin{aligned}
&\underset{(d,D,N)}{\text{maximize}} \quad \delta = \frac{8 P D^3 N}{d^4 G} \\
&\text{subject to}: \quad g_1 \equiv \frac{\tau}{\tau_A} - 1 \le 0 \quad ; \quad g_2 \equiv 1 - \frac{\omega}{\omega_A} \le 0 \\
&\qquad\qquad\quad g_3 \equiv \frac{D + d}{OD_A} - 1 \le 0 \quad ; \quad g_4 \equiv 1 - \frac{M}{M_A} \le 0 \\
&\qquad\qquad\quad g_5 \equiv \frac{\delta}{\delta_A} - 1 \le 0 \\
&\qquad\qquad\quad 0.05 \le d \le 0.20 \quad ; \quad 0.25 \le D \le 1.30 \quad ; \quad 2 \le N \le 15
\end{aligned}
\tag{3.34}
$$

where:

$$
\tau = \frac{8PD}{\pi d^3}\left[\frac{4D - d}{4D - 4d} + \frac{0.615d}{D}\right] \tag{3.35}
$$

$$
\omega = \frac{d}{2\pi\pi^2 N}\sqrt{\frac{G}{2\rho}} \tag{3.36}
$$

$$
M = \frac{\pi^2 D d^2 \rho}{4}(N + Q) \tag{3.37}
$$

We use a GA (Goldberg, 1989) to solve Eq. (3.34) and obtain the nominal optimum design $(d,D,N)^* = (0.0519 \text{ in}, 0.3616 \text{ in}, 11)$ for a maximum deflection of $\delta^* = 0.4985$ in. At the nominal optimum, constraints g_1 and g_4 are active. The lower bound of the variable d is also nearly active. In our implementation we use GA to solve this problem because one of the variables is an integer. However, other mixed-integer programming methods such as Branch and Bound algorithm (Belegundu and Chandrupatla, 1999) may also be used. For comparison, if we relax the variable N to be continuous, the nominal optimum becomes $(d,D,N)^* = (0.0517 \text{ in}, 0.3569 \text{ in}, 11.293)$ for a maximum deflection of $\delta^* = 0.50$ in.

Three of the problem's parameters have variability in them: the wire diameter d varies by $\pm\Delta d_0 = 0.001$ in; the mean coil diameter D varies by $\pm\Delta D_0 = 0.01$ in; and the compression force P varies by $\pm\Delta P_0 = 0.1$ lbf. We want to minimize the sensitivity of the optimum design with respect to these variations. Using $\Delta\delta_0 = 0.075$ in, we added our robustness constraint to Eq. (3.34) and solved it using GA for the inner and outer optimization problems ($\eta_0 = 1$). The robust optimum obtained is $(d,D,N)_R^* = (0.0548 \text{ in}, 0.4219 \text{ in}, 8)$ for a deflection of $\delta^* = 0.4633$ in (a 0.0352 in. difference from the nominal optimum). For comparison, we also solved the problem using the constrained worst-case gradient approach where we added the gradient robustness constraint: $\Delta\delta = \sum_{i=1}^{3} \left| \frac{\partial \delta}{\partial p_i} \right| \Delta p_{i,0} \le \Delta\delta_0$ to Eq. (3.34). The optimum obtained using this worst-case gradient approach is $(d,D,N)_G^* = (0.0560 \text{ in}, 0.3319 \text{ in}, 9)$ for a deflection of $\delta^* = 0.2321$ in. The values of the nominal, robust, and gradient optima are shown in Table 3.4.

Table 3.4: Optimum designs of the compression spring.

	Nominal Optimum	Robust Optimum	Gradient Optimum
δ (in)	0.4985	0.4633	0.2321
η	0.8248	0.9942	1.9109
d (in)	0.0519	0.0548	0.056
D (in)	0.3616	0.4219	0.3319
N	11	8	9
F_{call}	N/A	250	N/A

We see in Table 3.4 that the nominal optimum has the largest deflection, as expected, followed by the robust optimum and then the gradient optimum. If we look at the η value, our robust optimum satisfies the $\eta \geq 1$ requirement with an equality (rounded up), the nominal optimum does not satisfy the requirement, while the gradient optimum over-satisfies it. Although the gradient optimum more than satisfies our robustness requirement, it comes with a significant reduction in performance (less than half of the nominal value).

This observation brings up a very important fact regarding our robust optimization method. Although we label our WCSR measure as a "worst-case" approach, it is not really worst-case per se; at least not in the traditional sense. Traditionally, the term "worst-case" refers to a situation where all the worst possible variabilities occur together simultaneously. Our WCSR measure is different. Instead of blindly using the worst variabilities for each parameters, it has implicitly taken into account the fact that some of these variables might cancel out. This is the reason why our robust optimization method can obtain a design with such a high performance while still satisfying our robustness requirement. This capability comes with a price, however, namely it needs to perform

some function evaluations to gather the robustness information (~250 function calls per design in this example).

To validate the robustness of the optimum designs shown in Table 3.4, we perturbed (d,D,P) 20 times within the (Δd_0,ΔD_0,ΔP_0) range, and each time calculated the deviation of the δ value of each design from its original value. The results of our analysis are shown in Figures 3.35 through 3.37. In these figures the dashed lines are the $\Delta\delta_0$. We can make a couple observations from these figures. We see in Figure 3.35 that the nominal optimum violates the $\Delta\delta_0$ requirement in case 11, while the robust and the gradient optima never violate $\Delta\delta_0$ (Figure 3.36 and Figure 3.37, respectively). We also observe in these figures that on average the $\Delta\delta$ of the nominal optimum is large, while the robust optimum's is less, and that of the gradient optimum is least. This observation verifies the information provided by their η values.

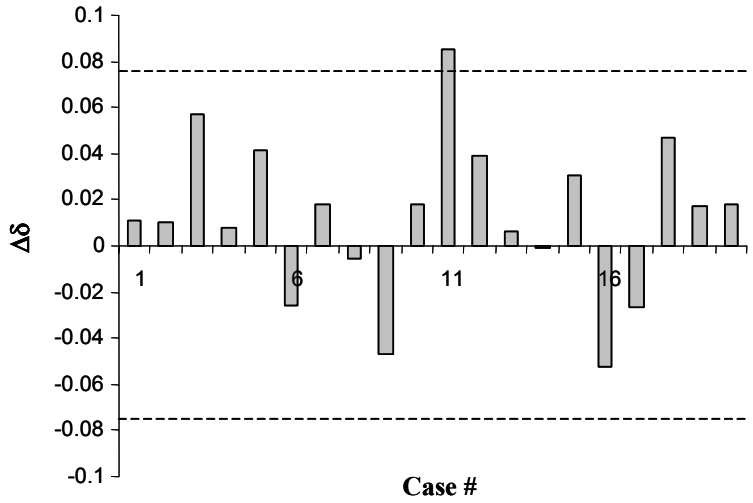


Figure 3.35: Sensitivity analysis of the nominal optimum.

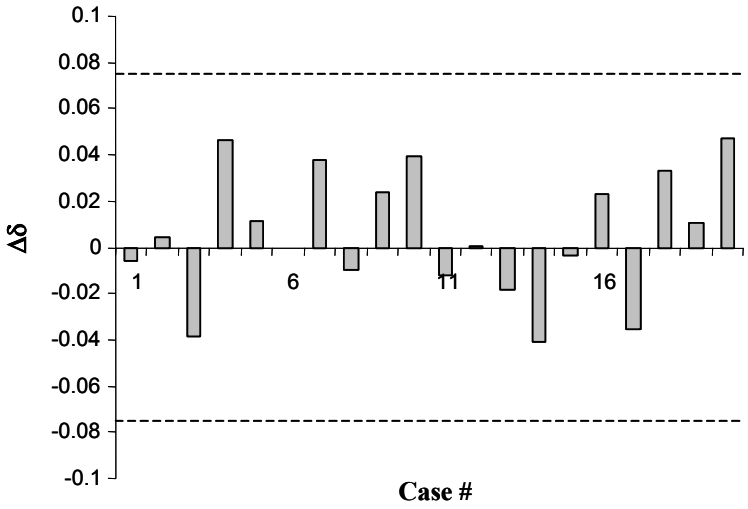


Figure 3.36: Sensitivity analysis of the robust optimum.

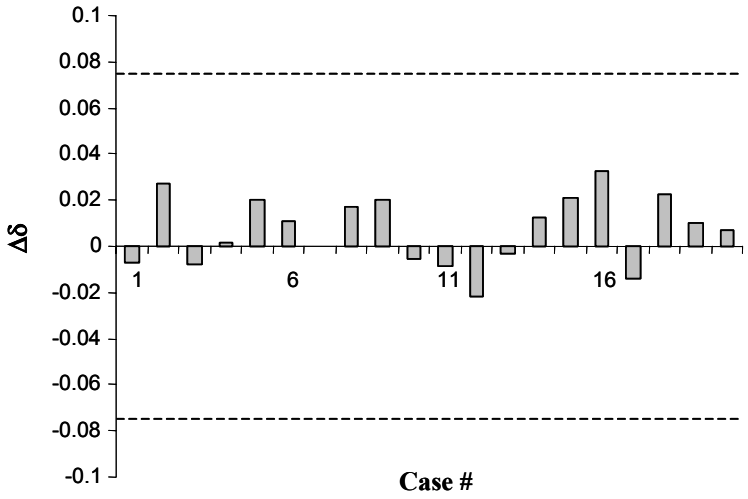


Figure 3.37: Sensitivity analysis of the gradient optimum.

3.5. SUMMARY

- For each design alternative, there is a unique set of $\Delta\mathbf{p}$ that indicates how much variation the design can absorb before the Δf_0 limit is violated. This set is called the sensitivity set ($\mathbf{S}_f$) of that design.
- Size of a $\mathbf{S}_f$ is a measure of how robust a design is: the larger $\mathbf{S}_f$, the more robust the design. In addition, since $\mathbf{S}_f$ measures how much variation a design can absorb, if we

can control the actual variations such that they are always within $\mathbf{S}_f$, then we are guaranteed that the performance of the design is always within $\pm\Delta f_0$.

- The plot of $\mathbf{S}_f$ in the $\Delta\mathbf{p}$-space is called the Sensitivity Region (SR) of the design. The $\Delta\mathbf{p}$ points inside, outside, and on the SR boundary satisfy $[\Delta f(\Delta\mathbf{p})]^2 < [\Delta f_0]^2$, $[\Delta f(\Delta\mathbf{p})]^2 > [\Delta f_0]^2$, and $[\Delta f(\Delta\mathbf{p})]^2 = [\Delta f_0]^2$, respectively.
- The size of a SR is a measure of a design's robustness: the larger SR, the more robust the design. However, since a SR is typically asymmetric, we have to account for the directional sensitivity of the design as well.
- We use the most sensitive direction in approximating a SR of a design to account for the worst-case situation. This worst-case approximation to a SR is called the Worst Case Sensitivity Region (WCSR), and is defined as the smallest hyper-sphere inside the SR that touches the SR boundary on at least one point.
- Size of a WCSR is a worst-case measure of a design's robustness: the larger the WCSR, the more robust the design. The radius of a WCSR (R_f) can be calculated by solving an optimization problem shown in Eq. (3.4). Since WCSR size has a semi-cardinal scale, it is not necessary to calculate its volume. The radius value is sufficient to compare robustness of two or more designs.
- If the magnitudes of the Δp are significantly different, then we need to normalize the optimization problem used to calculate R_f. We use a single-valued normalization with $\Delta\mathbf{p}_0$ as the reference point.
- To determine whether or not a design meets our robustness requirement, we use the robustness index $\eta = (G)^{-1/2} R_f$. A design is robust if $\eta \geq 1$, and is not robust if $\eta < 1$.

- In robust optimization we want to simultaneously maximize performance and robustness. These objectives are often conflicting, so to avoid having to make a trade-off, we use a constrained robustness approach instead.
- Using the constrained robustness approach, our robust optimization method searches for the robust optimum design by solving two optimization problems: an inner and outer optimization. The inner optimization is used to calculate a design's robustness which is then used in the outer optimization.

CHAPTER 4

MULTI-OBJECTIVE ROBUST OPTIMIZATION

4.1. INTRODUCTION

When an optimization problem has only one objective, the notion of a robust optimum is straightforward because the only trade-off present is between the objective and the robustness of the design. When a problem has multiple objectives, however, it is not quite so because the same variability might affect some or all of the objectives, and in turn, a design will have different robustness behavior with respect to different objectives. If we follow the conventional definition of a robust optimum design in the context of a multi-objective optimization, we might end up making trade-offs between the performance of the design in terms of one objective and its robustness in terms of another objective. Clearly, not only this is difficult to do, it is not very useful either.

The purpose of this chapter is to present the concept of multi-objective robustness, a method on how to measure such robustness, and an optimization scheme to obtain a design that is optimum and robust multi-objectively by using this measure. The method to measure a multi-objective robustness of a design presented in this chapter is a generalization of the WCSR measure we presented in Chapter 3, so there will be similarities, but with important differences. Like in Chapter 3, here we look into the robustness of a design in terms of its objectives only. Feasibility robustness will be covered in Chapter 5. Several examples are given at the end of the chapter to demonstrate the applications of the method.

4.2. BASIC CONCEPTS

Before we present the method to obtain robust solutions of a multi-objective problem, we need to first discuss what it means for a design to be multi-objectively robust and optimum.

4.2.1. Multi-Objective Robustness

When there is only one objective, a design is termed "robust" if its objective value is insensitive to parameter variations. When there are multiple objectives, a design has different robustness behaviors depending on the objective in question. So by direct extension, a design is termed "robust" multi-objectively if each of its objective value is insensitive to parameter variations. In other words, a design is robust (or insensitive) if Δf_i is small, for all $i = 1,\ldots,M$. In the $\boldsymbol{f}$-space, the robustness of a design is depicted graphically as the hyper-rectangle constructed from the Δf_i ranges (Figure 4.1); the smaller this hyper-rectangle (i.e., a rectangle in two-dimension, as shown in Figure 4.1), the more robust the design. In Figure 4.1, the solid triangles denote the nominal $\boldsymbol{f}$ values of the designs A, B, C and D shown.

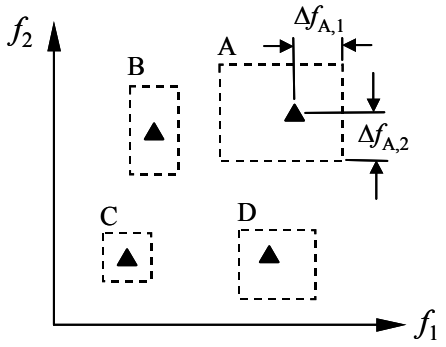


Figure 4.1: Graphical illustration of multi-objective robustness.

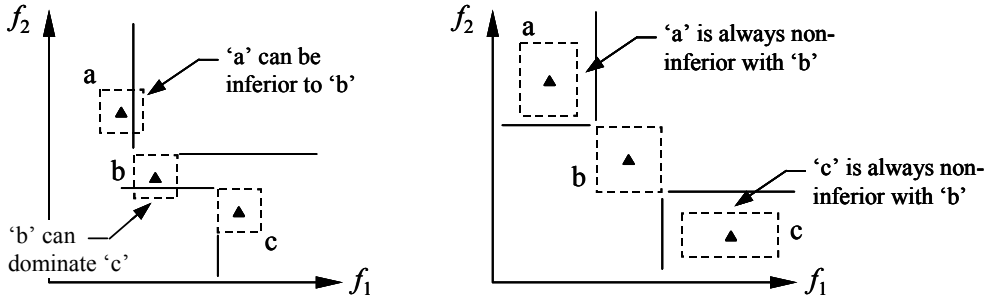
As shown in Figure 4.1, it is very often the case that a design is less sensitive in one objective than the other objective(s) (design B). Also, the negative and positive ranges of Δf_i do not necessarily have to be equal (design A and D).

It should be noted that the term "insensitive" is subjective and depends on the preferences of the decision maker. If the decision maker is risk averse, then for a design to be insensitive, its Δf_i must be very small for all objectives $i = 1,\ldots,M$. On the other hand, if the decision maker is risk prone, then even a rather large Δf_i is acceptable. In this chapter, we deem a design multi-objectively robust if $|\Delta f_i| \leq |\Delta f_{i,0}|$ for all $i = 1,\ldots,M$, where $\Delta \boldsymbol{f}_0 = [\Delta f_{1,0}, \ldots, \Delta f_{M,0}]$ is the ranges of acceptable $\boldsymbol{f}$ variation determined by the decision maker.

The multi-objective robustness described previously is defined for one design only. If we have a set of designs, the overall robustness of the set is determined by looking at the robustness of each and every design, one at a time. In other words, our definition of multi-objective robustness is for *point robustness*. Another type of multi-objective robustness worth mentioning is that of *set robustness*. Unlike point robustness, set robustness is not defined for one design, but rather for a set of designs, and this set must be a trade-off set. A trade-off set is a set of designs in which all designs in the set are non-inferior with respect to each other (recall Section 2.2 that this set is different from a Pareto set). We define a set of trade-off designs to observe set robustness if the set remains a trade-off set when **p** varies.

Figure 4.2 shows a comparison between a set of trade-off designs that observes point robustness and a set that observes set robustness. We see in Figure 4.2(a) that the $\Delta \boldsymbol{f}$ of each design in the set is small, but for some of the $\Delta \boldsymbol{f}$ rectangles, there are portions that lie

on either the dominant or inferior region of another $\Delta \boldsymbol{f}$ rectangle. What this means is that when **p** varies, it is possible that these points are no longer trade-off points (i.e., one dominates or is dominated by others). In contrast, the $\Delta \boldsymbol{f}$ of each design on the set shown in Figure 4.2(b) is rather large (i.e., not a point robustness), but all the $\Delta \boldsymbol{f}$ rectangles lie in the non-inferior region of each other. This means that no matter how the points change, these points will always remain a trade-off set. It can be readily seen in Figure 4.2 that as $\Delta \boldsymbol{f}$ becomes infinitely small, point robustness implies set robustness and vice versa.



(a) Point robustness (b) Set robustness

Figure 4.2: Comparison between point and set robustness.

In multi-objective robust optimization, we are interested in the point robustness of a design only, and not its set robustness. This is because practically, the set robustness of a design is not important. We will discuss this issue further in the next section.

4.2.2. Multi-Objective Robust Optimality

The goal of Multi-Objective Robust Optimization (MORO) is to obtain a set of design alternatives that are: (i) multi-objectively robust (point robustness), and (ii) Pareto optimum for the nominal **p**. Such a set of design alternatives is termed multi-objectively robust and optimum, and we refer to them as robust Pareto solutions. The set of all robust Pareto solutions is called the robust Pareto set.

Ideally one wants the robust Pareto set to be the same as the nominal Pareto set. However, robust Pareto set is generally inferior to the nominal Pareto set as shown in Figure 4.3(a). It is also common for the robust Pareto set and the nominal Pareto set to overlap, as in Figure 4.3(b), or for the robust Pareto set to be a subset of the nominal Pareto set, as in Figure 4.3(c). Obviously, the robust Pareto set cannot be superior or be a superset of the nominal Pareto set. One might argue that in case of Figure 4.3(b), the overlap portion of the robust Pareto frontier is better than the non-overlap portion because not only this portion is robust, but it also belongs to the nominal Pareto frontier. In some sense, it is. However, objective-wise this overlap portion does not dominate the non-overlap portion, so multi-objectively the overlap and non-overlap portions are equally optimum (i.e., incomparable).

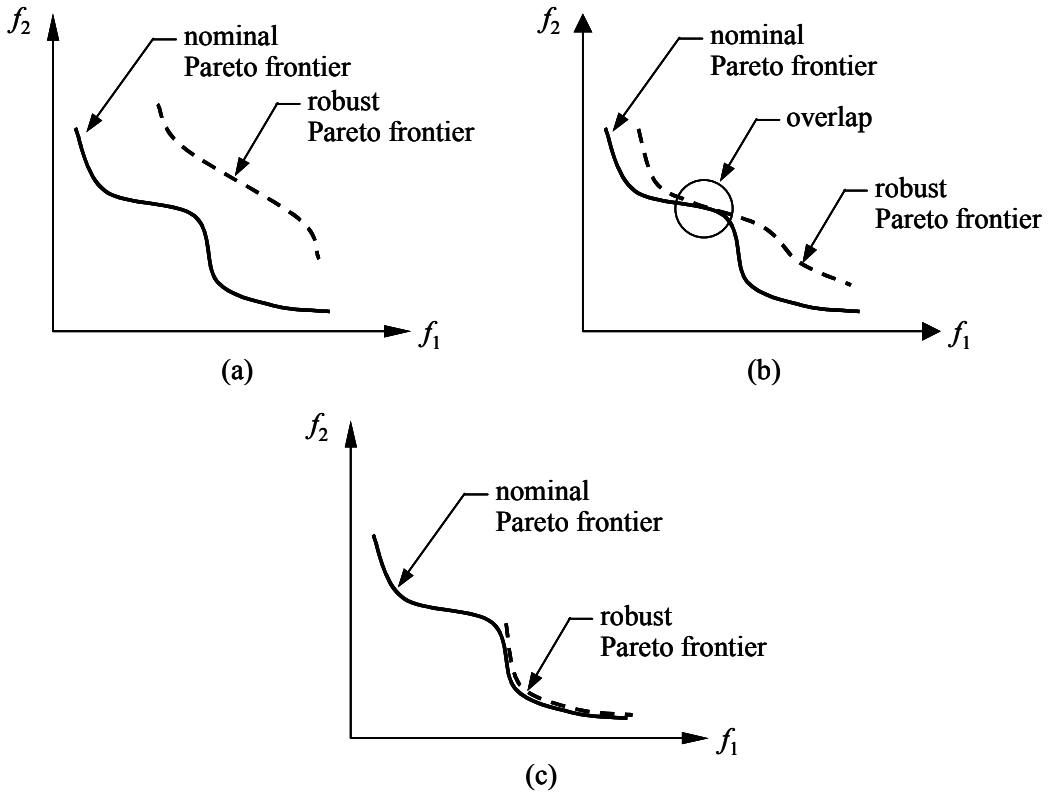


Figure 4.3: Comparison between nominal and robust Pareto set.

When $\mathbf{p}$ varies, $\boldsymbol{f}(\boldsymbol{x},\mathbf{p})$ deviates from its nominal value. Because the robust Pareto set is obtained based on point robustness, it is possible that when $\mathbf{p}$ changes the robust Pareto set is no longer a trade-off set. However, although the solution to a multi-objective optimization problem is a set of trade-off designs, ultimately the designer would choose a single design to implement. The purpose of including a design's robustness as an additional optimization criterion is to ensure that the one design selected is robust with respect to variations in $\mathbf{p}$. As such, it does not matter if the robust Pareto designs as a set are insensitive to $\mathbf{p}$. In other words, the set robustness of the robust Pareto set is of no real-world interest beyond our mathematical curiosity.

4.3. TWO-SIDED SENSITIVITY OF MULTIPLE FUNCTIONS

We are now ready to present our method for measuring multi-objective robustness of a design. The method presented here is a generalization of the WCSR method presented in Chapter 3 for single objective robustness. As before, the robustness measured here is a two-sided robustness, i.e., we account for both the increase and decrease in $\boldsymbol{f}$.

4.3.1. Generalized Sensitivity Set and Sensitivity Region

Let $\boldsymbol{x}_0$ be the design alternative whose sensitivity we want to measure, and let $\mathbf{p}_0$ be the nominal parameter value with respect to which the objective values of this design is defined, i.e., $\boldsymbol{f}(\boldsymbol{x}_0,\mathbf{p}_0)$. Given the acceptable variation range $\Delta\boldsymbol{f}_0 = [\Delta f_{1,0}, \Delta f_{2,0}, \ldots, \Delta f_{M,0}]$, there is a set of $\Delta\mathbf{p}$ values such that the $\Delta\boldsymbol{f}$ due to these $\Delta\mathbf{p}$ values falls within the ranges of $\Delta f_{i,0}$ for all $i = 1,\ldots,M$. This set is the generalized sensitivity set $\mathbf{S}_f$, and it is mathematically defined as shown in Eq. (4.1). As before, we use the square of each Δf_i

values to account for negative values, i.e., a two-sided sensitivity. We do not make a distinction between single and multiple objective $\mathbf{S}_f$ because the single objective $\mathbf{S}_f$ is simply the multi-objective version of $\mathbf{S}_f$ for M = 1.

$$\begin{aligned} \mathbf{S}_f(\boldsymbol{x}_0, \mathbf{p}_0) &= \left\{ \Delta\mathbf{p} \in \mathrm{R}^{\mathrm{G}} : [\Delta f_i]^2 \le [\Delta f_{i,0}]^2, \forall i = 1,...,\mathrm{M} \right\} \\ \text{where:} \quad \Delta f_i &= f_i(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f_i(\boldsymbol{x}_0, \mathbf{p}_0) \end{aligned} \tag{4.1}$$

Similar to the single objective case (recall Chapter 3), $\mathbf{S}_f$ determines how much $\Delta\mathbf{p}$ a design can absorb for the given $\Delta\boldsymbol{f}_0$. So, the larger $\mathbf{S}_f$, the more robust the design, and if we can make sure that $\Delta\mathbf{p}$ is always a member of $\mathbf{S}_f$, then $\Delta\boldsymbol{f}_0$ will always be satisfied. The plot of the generalized $\mathbf{S}_f$ on the $\Delta\mathbf{p}$-space is the generalized Sensitivity Region (SR) of the design.

If we let the notation $\mathbf{S}_{f,i}$ be the set of $\Delta\mathbf{p}$'s such that $[\Delta f_i]^2 \le [\Delta f_{i,0}]^2$, i.e., the sensitivity set with respect to the i-th objective, then it is easily seen that $\mathbf{S}_f$ is really just the intersection of all $\mathbf{S}_{f,i}$'s:

$$\mathbf{S}_f = \mathbf{S}_{f,1} \cap \ldots \cap \mathbf{S}_{f,\mathrm{M}} \tag{4.2}$$

The above observation implies that the generalized SR corresponding to $\mathbf{S}_f$ is simply the intersection of all the SR's of $\mathbf{S}_{f,i}$'s (Figure 4.4). The existence and uniqueness of $\mathbf{S}_f$ also follows directly from the existence and uniqueness of each $\mathbf{S}_{f,i}$.

The fact that the generalized SR is an intersection of each objective's SR's has another important consequence: it defines the requirements that must be met for a point in the $\Delta\mathbf{p}$-space to be inside, outside, or on the boundary of the generalized SR. A point inside the region (shaded region in Figure 4.4) implies that it belongs to all $\mathbf{S}_{f,i}$'s, but it is not on the boundary of any $\mathbf{S}_{f,i}$'s, hence it must satisfy $[\Delta f_i]^2 < [\Delta f_{i,0}]^2$ for all $i = 1,\ldots,\mathrm{M}$. A point outside the region implies that it does not belong to at least one $\mathbf{S}_{f,i}$, thus for this

point there must be at least one i such that $[\Delta f_i]^2 > [\Delta f_{i,0}]^2$. A point on the boundary of the region implies that not only it belongs to all $\mathbf{S}_{f,i}$'s, but it is also on the boundary of at least one $\mathbf{S}_{f,i}$, therefore it must satisfy $[\Delta f_i]^2 \le [\Delta f_{i,0}]^2$ with a strict equality for at least one i.

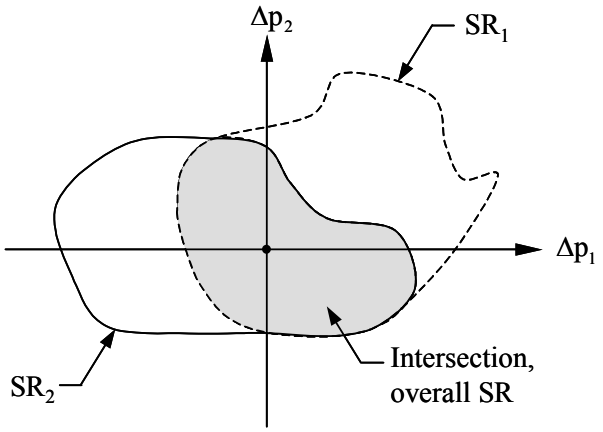


Figure 4.4: Graphical definition of the generalized SR.

Graphically, the points inside, outside, and on the perimeter of the generalized SR correspond to points inside, outside, and on the boundary of the hyper-rectangle formed by the Δf_0 ranges respectively (Figure 4.5).

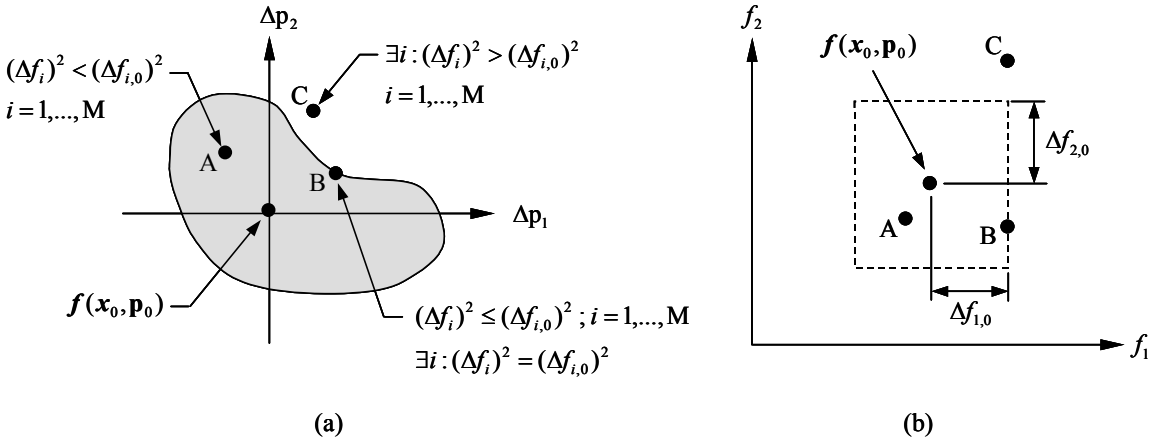


Figure 4.5: (a) Generalized SR, and (b) its corresponding Δf_0 ranges.

Theoretically, it is possible that the SR's intersect at their boundaries only, which graphically translates into a degenerate generalized SR (it is now a G-1 dimensional

hyper-surface). When this situation occurs, the points inside the generalized SR satisfy the condition for a boundary point as well. Practically however, this implies that one of the Δp_j's can be expressed as a function of the other Δp_j's. In reality this situation almost never happens, so for practical purposes we can safely assume that the SR's intersection is a G-dimensional hyper-surface, and that the equations shown in Figure 4.5(a) are valid.

4.3.2. Generalized Worst Case Sensitivity Region

As seen in Figures 4.4 and 4.5, in general a multi objective SR is asymmetric, and as such it is possible that a design is very sensitive along a certain direction of $\Delta\mathbf{p}$, but is much less sensitive along other directions. To account for this directional sensitivity, here too we use a worst-case approximation to the generalized SR to measure a design's robustness. We call this approximation the generalized Worst Case Sensitivity Region (WCSR), which is a symmetric region that approximates the generalized SR along its most sensitive direction. The definition for the generalized WCSR is very much the same as the single objective WCSR except that now it is defined for the generalized SR. So, mathematically the generalized WCSR is defined as a subset of $\mathbf{S}_f$ in which for each point in the subset, its distance from the origin is less than or equal to the distance of the smallest $\Delta\mathbf{p}$ value in $\mathbf{S}_f$ that causes $\Delta\boldsymbol{f}_0$ to be satisfied. Graphically, the generalized WCSR is a hyper-sphere inside the generalized SR that touches the SR boundary at the closest point from the origin, as shown in Figure 4.6.

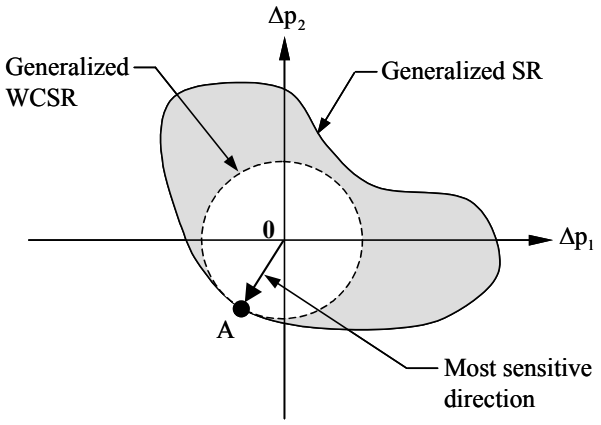


Figure 4.6: Generalized WCSR.

The radius of the generalized WCSR for a design $\boldsymbol{x}_0$ can be calculated by solving the single objective optimization problem shown in Eq. (4.3). In this problem, the design variables are the $\Delta\mathbf{p}$, the objective function is the WCSR radius, and the constraints reflect the fact that the WCSR touches the SR at a boundary point. Notice how the constraints of Eq. (4.3) differ significantly from their single objective counterpart (Eq. (3.4)). This is because the boundary condition for the generalized SR is different than the single objective SR. Despite the difference, however, if M = 1, then Eq. (4.3) collapses into Eq. (3.4), thus shows that Eq. (4.3) is simply the generalization of Eq. (3.4).

$$\begin{aligned} &\underset{\Delta\mathbf{p}}{\text{minimize}} \;\; \mathrm{R}_f(\Delta\mathbf{p}) = \left[\sum_{j=1}^{\mathrm{G}} (\Delta \mathrm{p}_j)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \;\; \frac{[\Delta f_i]^2}{[\Delta f_{i,0}]^2} - 1 \le 0 \quad ; \quad i = 1,\ldots,\mathrm{M} \\ &\qquad\qquad \Delta f_i = f_i(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f_i(\boldsymbol{x}_0, \mathbf{p}_0) \\ &\qquad\qquad \text{with at least one strict equality} \end{aligned} \tag{4.3}$$

Like in single objective WCSR, we can use R_f^2 instead of R_f if it makes the problem easier to solve. However, this time we cannot choose between either "=" or "≤" for the

constraints. The constraints must all be inequalities, and at the optimum, at least one of them must be satisfied as an equality.

Eq. (4.3) is a single objective optimization problem with M inequality constraints (from which at least one is active). Realizing that these constraints simply mean that the optimum must lie on the boundary of the rectangle formed by the $\Delta \boldsymbol{f}_0$ ranges (see Figure 4.5), we can simplify the constraints by aggregating them into a single constraint as follows:

$$\max_{i=1,\ldots,M}\left(\frac{|\Delta f_i|}{\Delta f_{i,0}}\right)-1=0 \qquad (4.4)$$
$$\text{where}: \Delta f_i = f_i(\boldsymbol{x}_0, \mathbf{p}_0+\Delta\mathbf{p}) - f_i(\boldsymbol{x}_0, \mathbf{p}_0)$$

The simplified constraint in Eq. (4.4) above restricts the maximum ratio of $\frac{|\Delta f_i|}{\Delta f_{i,0}}$ to be unity (we use the absolute of Δf_i to account for negative values). Graphically, this constraint means that the resultant vector of Δf_i's must touch the rectangle formed by the $\Delta \boldsymbol{f}_0$, which is again equivalent to saying that $\Delta\mathbf{p}$ must be on the generalized SR boundary (Figure 4.7).

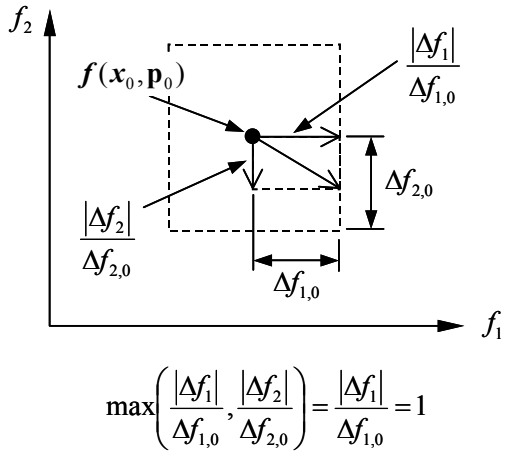


Figure 4.7: Graphical interpretation of the simplified constraint.

Substituting the constraints in Eq. (4.3) with the simplified constraint in Eq. (4.4), the optimization problem to obtain the radius of the generalized WCSR becomes:

$$\begin{aligned} &\underset{\Delta\mathbf{p}}{\text{minimize}} \;\; \mathrm{R}_f(\Delta\mathbf{p}) = \left[\sum_{j=1}^{\mathrm{G}} (\Delta \mathrm{p}_j)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{i=1,\ldots,\mathrm{M}} \left(\frac{|\Delta f_i|}{\Delta f_{i,0}}\right) - 1 = 0 \\ &\text{where}: \Delta f_i = f_i(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f_i(\boldsymbol{x}_0, \mathbf{p}_0) \end{aligned} \tag{4.5}$$

As in the single objective case, Eq. (4.5) is always guaranteed to have a feasible solution. At the very least $\Delta\mathbf{p} = \{\mathbf{0}\}$ is guaranteed to be a feasible solution to Eq. (4.5) and $\mathrm{R}_f = 0$. It is also possible that the generalized WCSR is completely unbounded, and $\mathrm{R}_f = \infty$. However, in general $0 < \mathrm{R}_f < \infty$, and solving Eq. (4.5) results in a finite radius of the generalized WCSR.

4.3.3. Normalization

Again, when the magnitudes of $\Delta\mathbf{p}$ are different, Eq. (4.5) needs to be normalized to account for the difference in scale importance. Using a single-valued normalization where we use the known variation ranges $\Delta\mathbf{p}_0$ as the reference value, the normalized optimization problem to obtain the generalized WCSR radius is obtained:

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \;\; \overline{\mathrm{R}}_f(\Delta\overline{\mathbf{p}}) = \left[\sum_{j=1}^{\mathrm{G}} (\Delta\overline{\mathrm{p}}_j)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{i=1,\ldots,\mathrm{M}} \left(\frac{|\Delta f_i(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)|}{\Delta f_{i,0}}\right) - 1 = 0 \\ &\text{where}: \Delta\overline{\mathbf{p}} = \left[\Delta\overline{\mathrm{p}}_1, \ldots, \Delta\overline{\mathrm{p}}_{\mathrm{G}}\right] \quad ; \quad \Delta\overline{\mathrm{p}}_j = \frac{\Delta \mathrm{p}_j}{\Delta \mathrm{p}_{0,j}} \end{aligned} \tag{4.6}$$

In this formulation, the objective and constraint are the same as in Eq. (4.5) except that they are modified for $\Delta\overline{\mathbf{p}}$ instead of $\Delta\mathbf{p}$. As before we can use $\overline{R}_f^2$ for the objective if it makes the problem easier to solve.

To demonstrate the application of Eq. (4.6), let us revisit our stainless steel piston pin example from Chapter 3. Previously, we measured the robustness of this pin based on its weight only. Here we will add a second objective, the stress acting on the pin.

Example 4.1

As part of the engine assembly, our stainless steel pin is attached to a piston arm at the middle of the pin (Figure 4.8). The length of the pin in contact with the arm is 3 cm, and there is a constant force P = 50 kN acting on the piston arm as shown in the figure. Determine the normal stress on the pin. If the density and radius of the pin vary by $(\Delta\rho_0,\Delta r_0) = (0.2, 0.02)$, determine also the normalized radius of the two-objective WCSR (weight and stress). The acceptable stress variation is given to be $\Delta\sigma_0 = 0.5$ MPa.

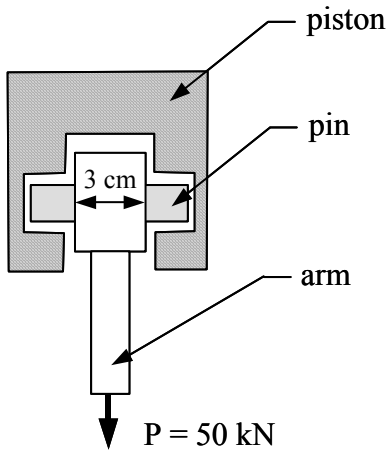


Figure 4.8: A piston-pin-arm assembly.

Solution

The pin is cylindrical, so the amount of area used to calculate stress is the projection of a cylindrical area, i.e., a rectangle. The length of the projection area is the length of contact (l = 30 mm), while the width is the diameter of the pin (= 2r). So, the normal stress acting on the pin is $\sigma = \dfrac{P}{(30)(2r)}$ = 41.6 MPa. (Notice that this value is much less than the yield stress of stainless steel ~200 MPa, so our pin design is fine.)

If we use Eq. (4.6) to calculate the normalized WCSR radius of this problem, we obtain R_f = 0.618. Let's verify this result. The generalized SR of the pin design is the intersection of the weight SR and the stress SR. We have obtained the weight SR from the previous examples in Chapter 3, so all we need now is the stress SR. Setting $(\Delta\sigma)^2 \le (\Delta\sigma_0)^2$, we obtain the following quadratic inequality, where $u = 1/(r+\Delta r)$.

$$\left(\frac{P^2}{4l^2}\right)u^2 - \left(\frac{83.2P}{2l}\right)u + 1730.31 \le 0 \tag{4.7}$$

Substituting the value P = 50 kN and l = 3 cm, and solving the inequality in Eq. (4.7), we obtain the stress $\mathbf{S}_f$ to be as follows (in cm). Notice that the stress $\mathbf{S}_f$ is independent of $\Delta\rho$ because σ is independent of ρ.

$$\mathbf{S}_f = \{(\Delta\rho, \Delta r) : -0.0197 \le \Delta r \le 0.0267\} \tag{4.8}$$

The generalized SR and WCSR (based on weight and stress) of this pin design are shown in Figure 4.9 (shown in the normalized region). In this graph, the region bounded by the solid lines is the weight SR, while the region bounded by the dashed lines is the stress SR. The shaded region is the overall SR, while the white circle is the overall WCSR.

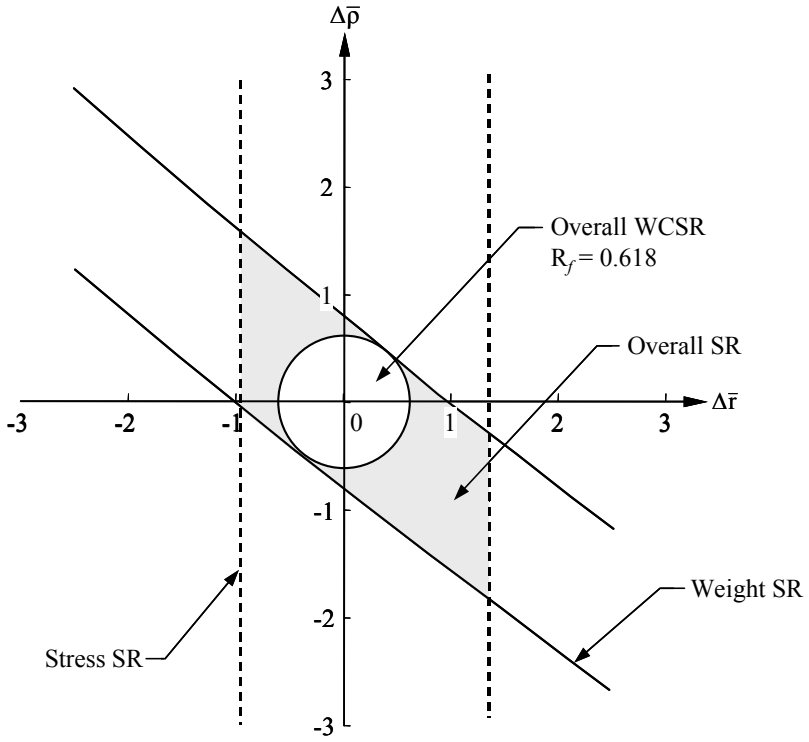


Figure 4.9: Overall SR and WCSR of the stainless steel pin.

From the graph, the point of contact is approximately at $(\Delta\bar{\rho}, \Delta\bar{r}) = (0.4, 0.5)$, for a WCSR radius of 0.64. This value is in good agreement with the one obtained analytically ($R_f = 0.618$). Notice also in this graph that the R_f value is governed by the weight SR. ♦

4.4. ROBUST OPTIMIZATION

We will now use the generalized R_f measure developed previously to obtain robust solutions to a multi-objective optimization problem.

As before, by considering the robustness of a design, we have essentially added an additional objective to an optimization problem that already has multiple objectives. Unlike other "real" objectives, however, robustness is a rather abstract concept, and therefore it is hard to make trade-offs with robustness as one of the objectives. To avoid

this difficulty, we again use a constrained approach to MORO where we treat the robustness of a design as a constraint, and not as an objective.

With this constrained approach, our MORO method obtains the robust Pareto set by first eliminating all non-robust designs and then selecting the best set of trade-off designs from the designs that are not eliminated. These optimization steps are illustrated with an example in Figure 4.10. In this figure, the triangles indicate the nominal $\boldsymbol{f}$ values of a few designs, and the dashed rectangles show the possible $\Delta\boldsymbol{f}$. Initially, there are five feasible design alternatives (Figure 4.10(a)), but designs A and C do not meet the robustness constraint so they are eliminated (recall that in the objective space, the smaller the possible $\Delta\boldsymbol{f}$ range, the more robust the design). From the remaining designs, design E is inferior to design B so design E too is eliminated. Designs B and D are our robust Pareto designs, as shown in Figure 4.10(b).

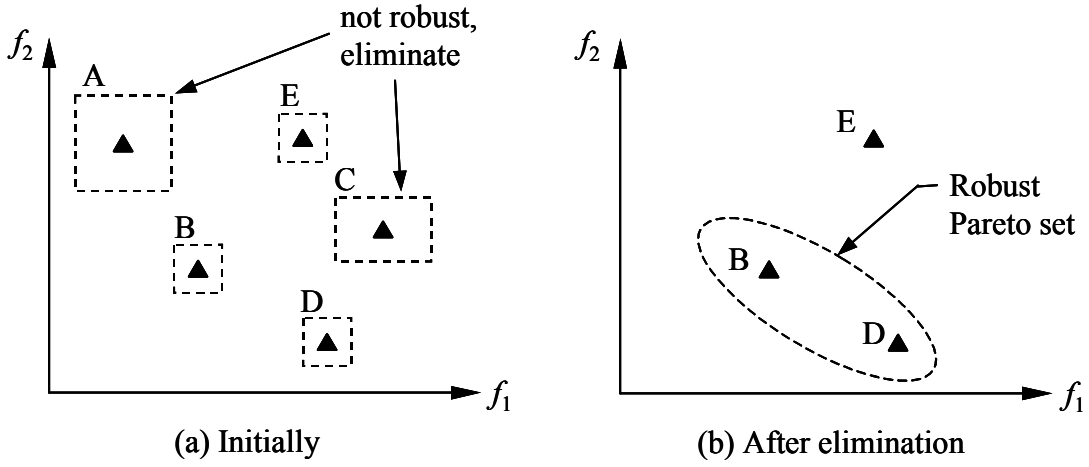


Figure 4.10: Constrained MORO approach.

If we define $\eta = (\mathrm{G})^{-1/2}\mathrm{R}_f$ to be the generalized robustness index, the problem to obtain robust solutions to a multi objective problem becomes as follows.

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & \left[f_1(\boldsymbol{x},\mathbf{p}_0),...,f_{\mathrm{M}}(\boldsymbol{x},\mathbf{p}_0)\right] \\
\text{subject to}: \quad & \mathrm{g}_j(\boldsymbol{x},\mathbf{p}_0) \le 0 \quad ; \quad j = 1,...,\mathrm{J} \\
& \mathrm{h}_k(\boldsymbol{x},\mathbf{p}_0) = 0 \quad ; \quad k = 1,...,\mathrm{K} \\
& 1 - \frac{\eta}{\eta_0} \le 0
\end{aligned}
\tag{4.9}
$$

Here, η_0 is the desired level of robustness and is determined by the designer, and R_f is calculated using Eq. (4.6).

Eq. (4.9) can be solved using any traditional multi-objective optimization methods: no preference (e.g., global criterion), a priori (e.g., weighting method), or a posteriori (e.g., Multi-Objective Genetic Algorithm, Multi-Objective Simulated Annealing). However, again due to the "hole-inducing" nature of the robustness constraint, we do not recommend those methods that solve multi-objective problems by converting them into a single-objective form. In our implementation, we use Multi-Objective Genetic Algorithm (MOGA) as our optimizer (see for example, Deb (2001) or Coello Coello et al. (2002)).

4.5. COMPARISON STUDY

To demonstrate our MORO method, we applied it to four examples: one numerical and three engineering examples. The purpose of the numerical example is to demonstrate the behavior of the robust Pareto solutions as we relax our robustness constraint. The purpose of the vibrating platform and the speed reducer examples is to demonstrate the application of our method to a two-objective optimization problem. The power electronic example is presented to show the application of our method to a three-objective optimization problem.

In all examples, the parameter variability of the problems is large (more than 5% of the nominal value). In two of the engineering examples, one of the design variables is

discrete. For implementation reasons, we use a GA to solve the inner optimization. Thus, in each problem each robustness constraint calculation require ~250 function evaluations.

4.5.1. Numerical Example

This relatively simple example has three variables, two objectives, and three constraints. There are also lower and upper bounds on the variables. All variables are continuous, and two of the constraints are objective constraints. Both the objectives and the constraints are non-symmetric with respect to the variables, and they are non-linear and non-convex. The mathematical formulation of the problem is shown in Eq. (4.10).

$$
\begin{aligned}
&\underset{(x_1,x_2,x_3)}{\text{minimize}} \quad f_1 = 1000\left\{x_1\left(16+x_3^2\right)^{0.5} + x_2\left(1+x_3^2\right)^{0.5}\right\} \\
&\underset{(x_1,x_2,x_3)}{\text{minimize}} \quad f_2 = \frac{20\left(16+x_3^2\right)^{0.5}}{1000x_1x_3} \\
&\text{subject to}: \quad g_1 \equiv \frac{f_1}{100} - 1 \le 0 \quad ; \quad g_2 \equiv \frac{f_2}{100} - 1 \le 0 \\
&\qquad\qquad g_3 \equiv \frac{f_3}{100} - 1 \le 0 \quad , \text{where}: f_3 = \frac{80\left(1+x_3^2\right)^{0.5}}{1000x_1x_3} \\
&\qquad\qquad 0.0001 \le x_1 \le 0.25 \\
&\qquad\qquad 0.0001 \le x_2 \le 0.25 \\
&\qquad\qquad 1.0 \le x_3 \le 3.0
\end{aligned}
\tag{4.10}
$$

In this formulation, the objectives and constraints have been numerically scaled to help expedite the optimization search. The nominal Pareto solutions to this problem are shown in Figure 4.11. These solutions are obtained using the NSGA method (see Srinivas and Deb, 1995).

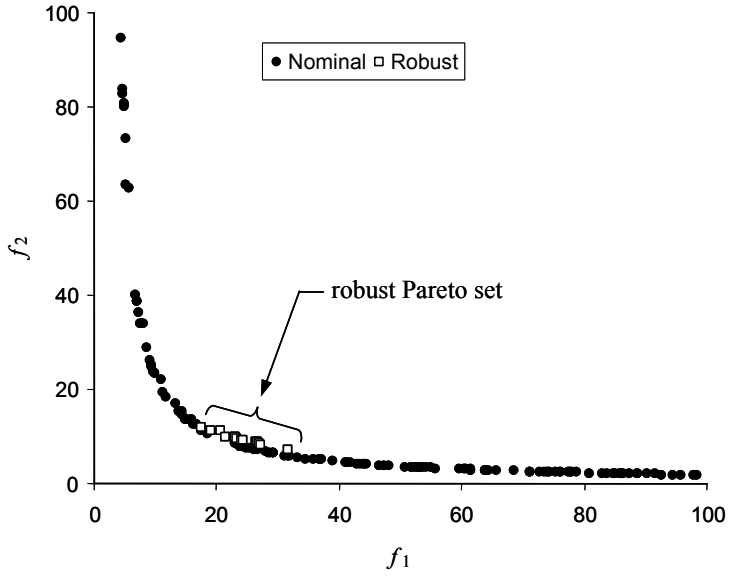


Figure 4.11: Nominal and robust Pareto set of the numerical example.

All three variables have variations in them, and we want to obtain robust solutions to this problem. The variations in each variable are given as $(\Delta x_1, \Delta x_2, \Delta x_3)_0 = (10^{-4}, 10^{-4}, 0.1)$, while the acceptable variation of each objective is specified to be $(\Delta f_{1,0}, \Delta f_{2,0}) = (1.0, 1.0)$. Adding the robustness constraint $1 - \frac{\eta}{\eta_0} \leq 0$ to Eq. (4.10) and solving it (with $\eta_0 = 1$), we obtain the robust Pareto solutions as shown in Figure 4.11. We see in Figure 4.11 that the robust Pareto set is essentially a subset of the nominal Pareto set (recall Figure 4.3(c)).

If we relax the robustness constraint slightly by using $\eta_0 < 1$, we will observe an interesting behavior. Figure 4.12 shows the robust Pareto set of the problem for $\eta_0 = 0.8$ (left) and for $\eta_0 = 0.5$ (right). As we see in this figure, as the robustness constraint is relaxed, the robust Pareto set "spreads out" along the nominal Pareto frontier, i.e., it includes those nominal Pareto solutions that were not robust enough before. This behavior is expected because there is a trade-off between the robustness of a design and its performance.

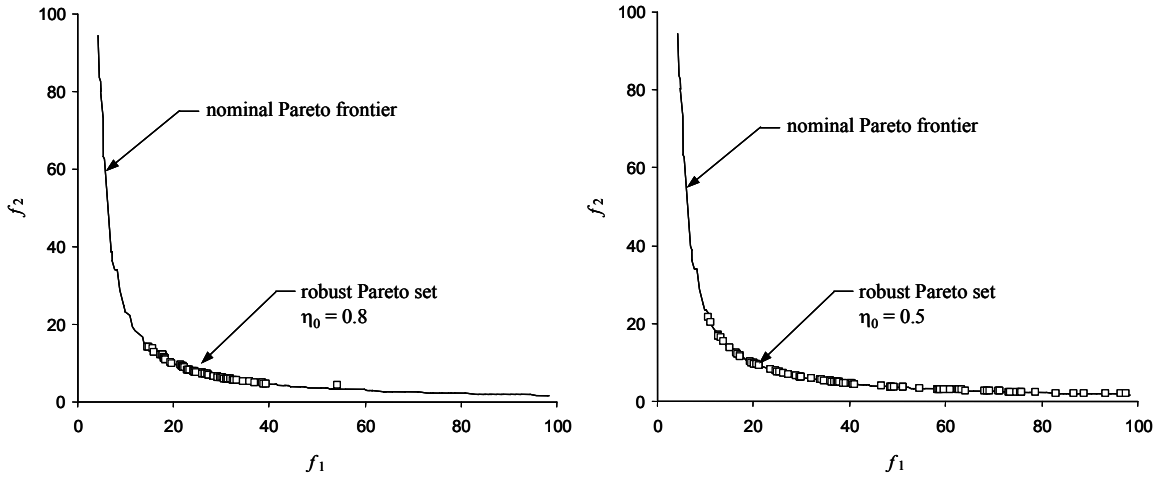


Figure 4.12: Behavior of robust Pareto set as constraint is relaxed.

Behavior similar to that observed in Figure 4.12 can also be expected for a robust Pareto set that is not a subset of the nominal Pareto set. For a robust Pareto set inferior to the nominal Pareto set (Figure 4.3(a)), the robust Pareto set will move closer to the nominal one as the robustness constraint is relaxed. For a robust Pareto set that overlaps with the nominal Pareto set (Figure 4.3(b)), the non-overlap part of the robust set will move closer to the nominal set while the overlap part will remain the same.

For comparison, we also solved Eq. (4.10) using a probabilistic approach by minimizing both the mean values of f_1 and f_2. The mean values are calculated using 100000 runs of Monte Carlo simulations assuming two different probability distributions of $\Delta \boldsymbol{x}$: uniform and normal. For the uniform assumption, the pdf is modeled as an equal probability distribution between $[-10^{-4}, +10^{-4}]$ for Δx_1, between $[-10^{-4}, +10^{-4}]$ for Δx_2, and between $[-0.1, +0.1]$ for Δx_3. For the normal assumption, the pdf is modeled to have a mean value of 0, and a standard deviation of $10^{-4}/3$ for Δx_1 and Δx_2, and 0.1/3 for Δx_3. In both models, the variations are assumed to be independent. The solutions obtained using this approach are shown in Figure 4.13.

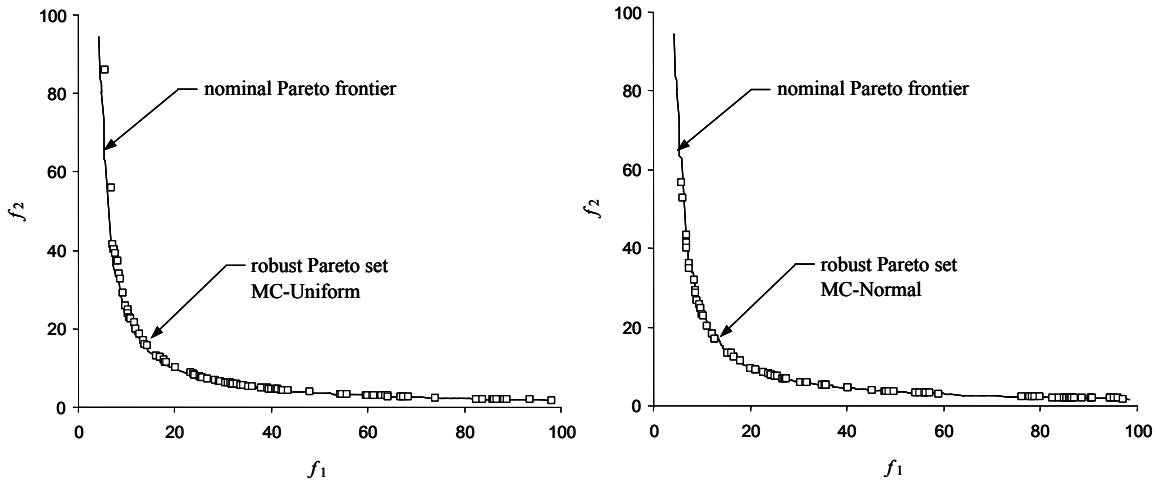


Figure 4.13: Robust solutions using Monte Carlo simulations.

We see in Figure 4.13 that solutions obtained using the Monte Carlo approach are really just the nominal Pareto solutions. In other words, minimizing the means of the objectives does not result in robust solutions. This is only an intuitive argument, however. Let us verify this claim.

Showing the three-dimensional SR of the designs is difficult, and projecting the SR onto a two-dimensional plane does not provide much information. For this reason, we will forgo analyzing the SR and WCSR of the solutions obtained, and instead simply perform a numerical verification of their robustness. We verify the results by randomly perturbing (x_1,x_2,x_3) around their nominal values according to $(\Delta x_1,\Delta x_2,\Delta x_3)_0$, and calculating the difference between the perturbed $\boldsymbol{f}$ value and the nominal $\boldsymbol{f}$ value. A design is robust if $\Delta\boldsymbol{f}$ is within $\Delta\boldsymbol{f}_0$. Figures 4.14 and 4.15 show the $\Delta\boldsymbol{f}$ of two of the robust designs (denoted by "robust-1" and "robust-2") obtained by our MORO method. The figure on the left is for Δf_1, and the one on the right is for Δf_2. The dashed lines in the figures are the $\Delta\boldsymbol{f}_0$. We see in these figures that the $\Delta\boldsymbol{f}$ of both designs is within $\Delta\boldsymbol{f}_0$. This shows that these designs are indeed robust.

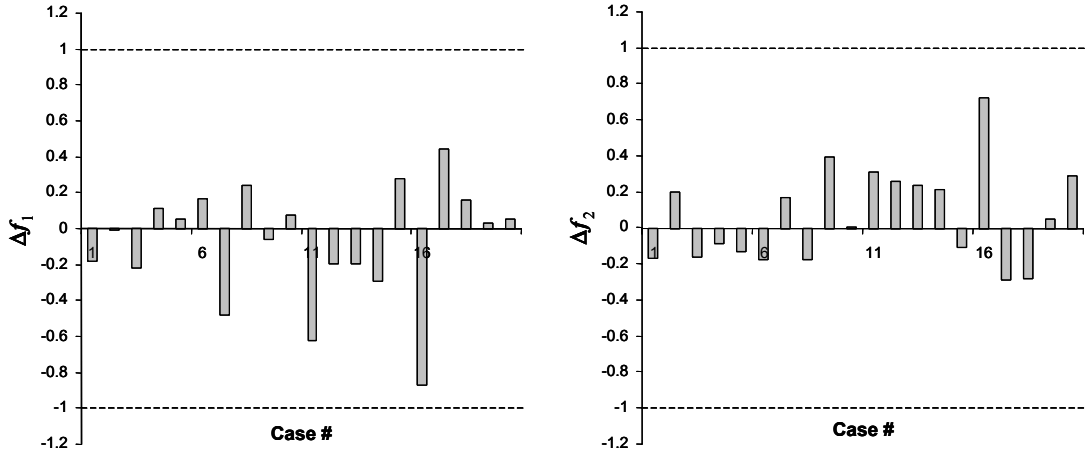


Figure 4.14: Sensitivity analysis of the robust-1 design.

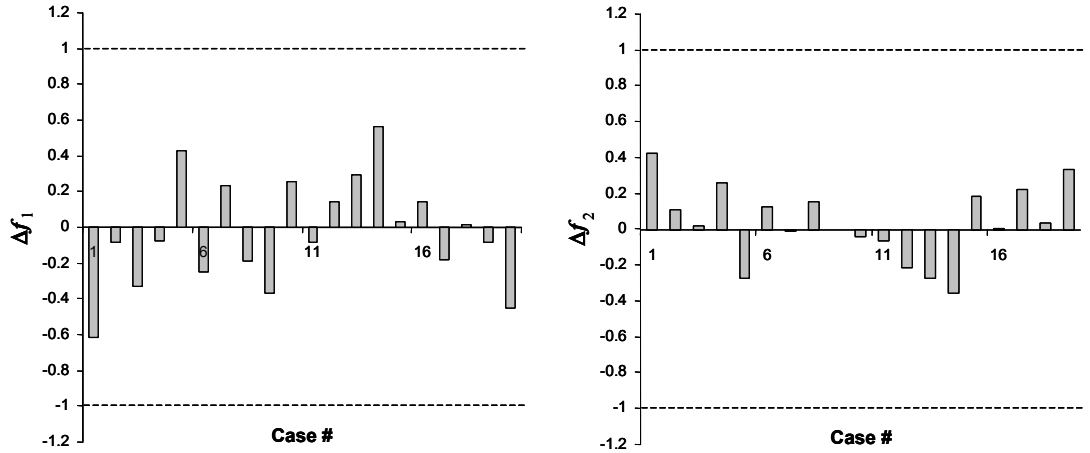


Figure 4.15: Sensitivity analysis of the robust-2 design.

For comparison, we performed the same analysis for the nominal designs. This time we picked three designs from the nominal Pareto set: one from the overlap region (nominal-1), one from the region where the f_1 value is high (nominal-2), and one from the region where the f_2 value is high (nominal-3). The $\Delta \boldsymbol{f}$ of the three designs due to perturbations is shown in Figures 4.16, 4.17, and 4.18 for the nominal-1, 2, and 3 designs, respectively. As before, the figure on the left is for Δf_1, while the figure on the right is for Δf_2.

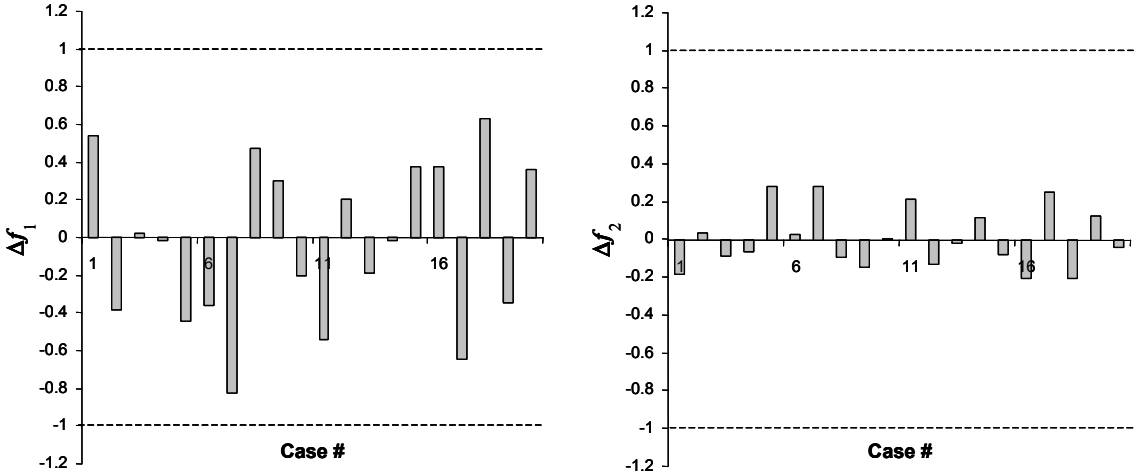


Figure 4.16: Sensitivity analysis of the nominal-1 design.

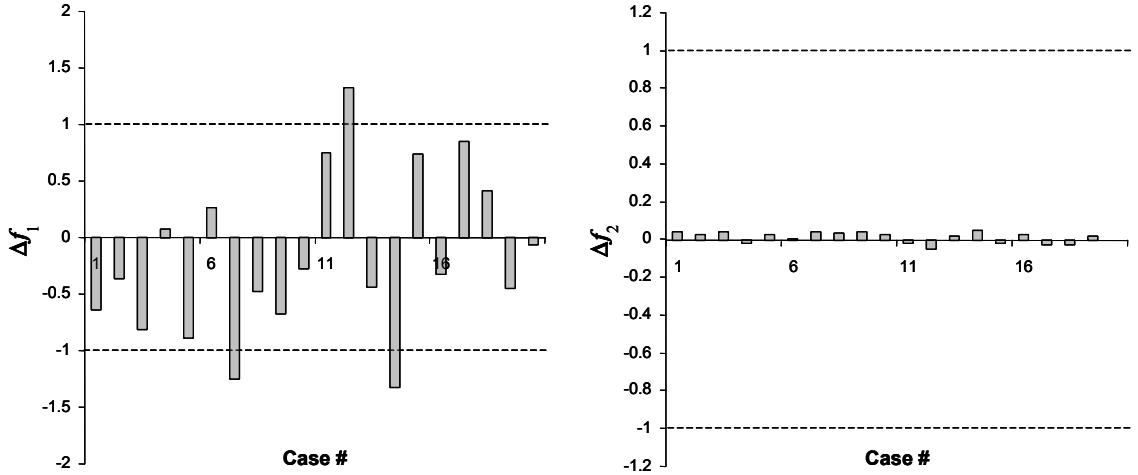


Figure 4.17: Sensitivity analysis of the nominal-2 design.

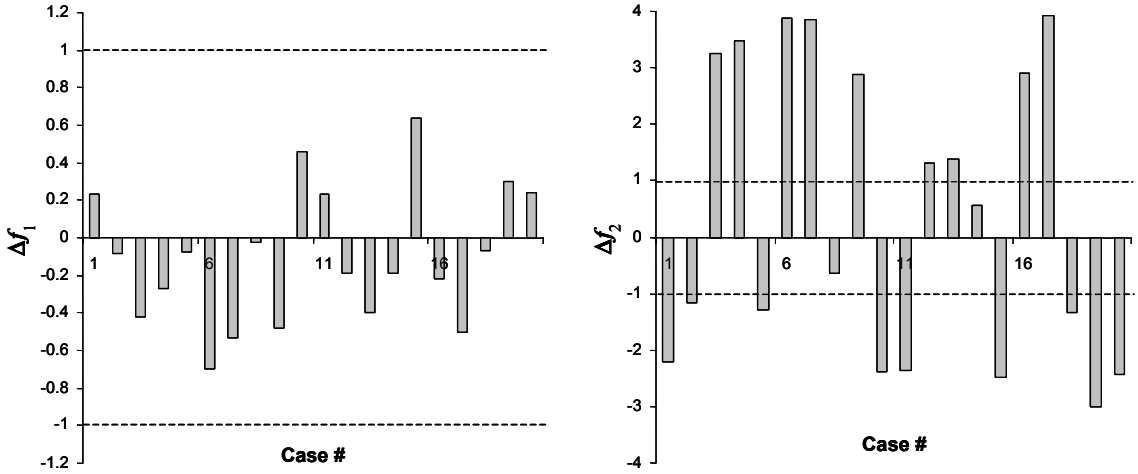


Figure 4.18: Sensitivity analysis of the nominal-3 design.

We see in Figure 4.16 that the nominal-1 design satisfies the Δf_0 limit, so this further shows that the designs in the overlap region are indeed robust. In contrast, Figure 4.17 shows that the nominal-2 design violates the $\Delta f_{1,0}$ limit, so this design is not robust with respect to f_1. Similarly, the nominal-3 design violates the $\Delta f_{2,0}$ limit, so this design is not robust with respect to f_2.

From Figure 4.13, the solutions obtained using a Monte Carlo approach are essentially the same as the nominal Pareto solutions, so robustness of these solutions are the same as the robustness of the nominal designs (Figure 4.18), and there is no need to redo the sensitivity analysis.

4.5.2. Design of a Vibrating Platform

For the second example, we applied our MORO approach to a two-objective constrained optimization problem as given by Narayanan and Azarm (1999). We modified the problem slightly by adding variations to two of the problem's parameters.

In this problem, we want to optimize the design of a vibrating platform modeled as a pinned-pinned sandwich beam with a vibrating motor on top, as shown in Figure 4.19.

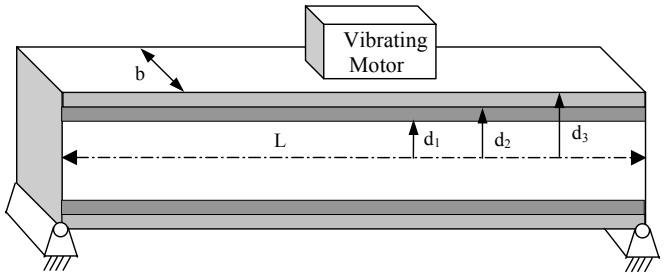


Figure 4.19: A pinned-pinned vibrating platform.

The platform has three layers of material (the inner layer, two middle layers sandwiching the inner layer, and two outer layers sandwiching the inner and middle

layers), and they are to be made out of three different materials: materials A, B, and C. The choice of materials for the layers must be mutually exclusive, i.e., no two layers can use the same material. However, the layers are allowed to have zero thickness (i.e., there is no layer). The properties of each of the materials are shown in Table 4.1. In this table, ρ is the mass density, E is the modulus of elasticity, and c is the cost of the material per volume.

Table 4.1: Material properties.

	Material A	Material B	Material C
ρ (kg/m^3)	100	2770	7780
E (GPa)	1.6	70	200
c (\$/m^3)	500	1500	800

We want to minimize the total cost of making such a platform and maximize its natural frequency by controlling five sizing variables (continuous) and one combinatorial variable (discrete). The sizing variables are the width of the platform (b), the length of the beam (L), and the thicknesses of the three layers (d_1, d_2, and d_3). The thicknesses of the middle and outer layers are represented as a difference between two sizing variables (e.g., thickness of the middle layer is equal to (d_2-d_1)). The combinatorial variable is the choice of materials for the layers (M). Since there are three possible material types, there are six possibilities for M (starting from the inner layer outward): {A,B,C}, {A,C,B}, {B,A,C}, {B,C,A}, {C,A,B}, and {C,B,A}. The platform design is subjected to five constraints: the maximum weight of the platform and the lower and upper limits on the thickness of the middle and outer layers.

The optimization formulation for this example is shown in Eq. (4.11).

$$
\begin{aligned}
\text{minimize} \quad & \text{cost} = 2b\left[c_1 d_1 + c_2\left(d_2 - d_1\right) + c_3\left(d_3 - d_2\right)\right] \\
\text{maximize} \quad & f_n = \left(\frac{\pi}{2L^2}\right)\left(\frac{EI}{\mu}\right)^{0.5} \\
& EI = \left(\frac{2b}{3}\right)\left[E_1 d_1^3 + E_2\left(d_2^3 - d_1^3\right) + E_3\left(d_3^3 - d_2^3\right)\right] \\
& \mu = 2b\left[\rho_1 d_1 + \rho_2\left(d_2 - d_1\right) + \rho_3\left(d_3 - d_2\right)\right] \\
\text{subject to:} \quad & g_1 \equiv \mu L - 2800 \le 0 \\
& g_2 \equiv \left(d_1 - d_2\right) \le 0 \\
& g_3 \equiv \left(d_2 - d_1\right) - 0.15 \le 0 \\
& g_4 \equiv \left(d_2 - d_3\right) \le 0 \\
& g_5 \equiv \left(d_3 - d_2\right) - 0.01 \le 0
\end{aligned}
\tag{4.11}
$$

In Eq. (4.11), the notations (ρ_1, ρ_2, ρ_3), (E_1, E_2, E_3), and (c_1, c_2, c_3) refer to the density, modulus of elasticity, and material cost for the inner, middle, and outer layer of the platform, respectively. The lower and upper bounds for the sizing variables are: $0.05 \le d_1 \le 0.5$, $0.2 \le d_2 \le 0.5$, $0.2 \le d_3 \le 0.6$, $0.35 \le b \le 0.5$, and $3 \le L \le 6$.

There are variations in the density and cost of "Material A" (ρ_A and c_A), and we want our optimum solutions to be insensitive to these variations. More specifically, we want to obtain the robust Pareto solutions of this problem for the nominal parameter values: $\rho_A = 100$ kg/m^3 and $c_A = 500$ \$/m^3.

We add the robustness constraint $1 - \frac{\eta}{\eta_0} \le 0$ to Eq. (4.11), and solve it (using $\eta_0 = 1$).

For the sensitivity requirements, we set the acceptable $\Delta \boldsymbol{f}$ to be ($\Delta f_{1,0}$, $\Delta f_{2,0}$) = (\$5, 5Hz) and the parameter variations are known to be ($\Delta\rho_{A,0}$, $\Delta c_{A,0}$) = (5 kg/m^3, 25 \$/m^3); 5% of the nominal ρ_A and c_A values. For the optimizer, we use Multi-Objective Genetic Algorithm (MOGA) of Fonseca and Fleming (1993) combined with the method of Kurapati et al. (2002) to handle constraints.

Figure 4.20 shows the robust Pareto set obtained (shown as a min-min plot by taking the negative of the frequency value). The average number of function evaluations performed in calculating the sensitivity constraint is 250. For comparison, Figure 4.20 also shows the nominal Pareto set of the problem (without the sensitivity constraint), and the Pareto set obtained using the probabilistic approach. In the probabilistic approach, we minimize the sum of mean and standard deviation value for each objective. These values are calculated by performing 100,000 Monte Carlo simulations.

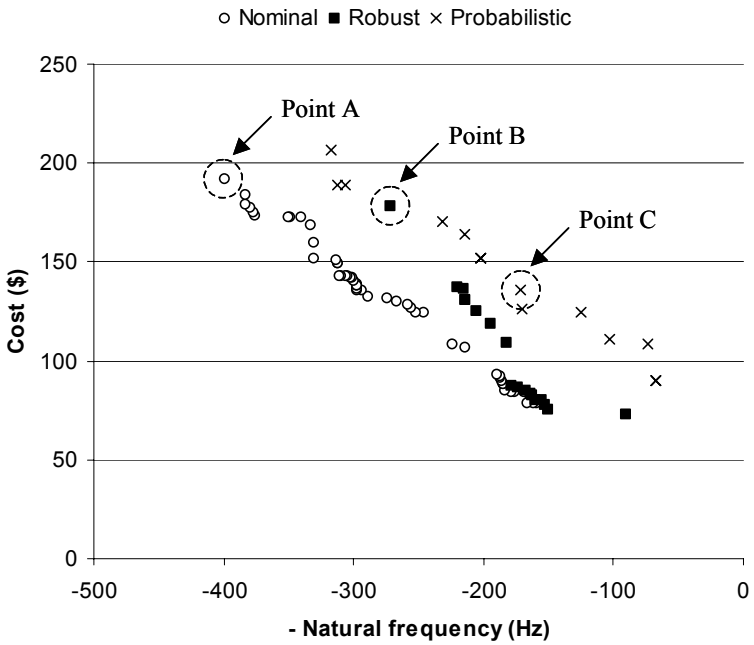


Figure 4.20: Pareto sets of the vibrating platform example.

We observe in Figure 4.20 that for the given sensitivity requirements, the robust Pareto set obtained is mostly inferior to the nominal Pareto set. So in designing the platform, we must sacrifice some performance to achieve higher robustness. However, when the frequency and cost values of the designs are in the range of (150 Hz, 200 Hz) and ($75, $100), the nominal and robust Pareto set overlap. Thus, those platform designs

within these ranges are not only insensitive to changes in ρ_A and c_A, but are also Pareto optimum for the nominal ρ_A and c_A values. Naturally, when making a selection for the final design, the designs in the overlap region are good candidates. The Pareto set of the probabilistic approach is inferior overall to both the nominal and robust Pareto sets. So, based on their multi-objective optimality these solutions are conservative.

Figure 4.20 shows the optimality of the robust Pareto set obtained, but we have yet to make a case for its robustness. To verify the robustness of the obtained robust Pareto set, we performed a sensitivity analysis on the design points by arbitrarily perturbing the parameters (ρ_A,c_A) around their nominal values. We then computed the changes in the objective values of the designs due to these perturbations and compared those changes to the acceptable values. Due to space limitations, we present the results of such sensitivity analysis only for three randomly selected design points, one point each from the nominal, robust, and probabilistic Pareto set (points A, B, and C in Figure 4.20, respectively). Figure 4.21 shows the plots of the $\Delta \boldsymbol{f}$ values of the three design points.

In Figure 4.21, the inner rectangle is the range of acceptable objective value $\Delta \boldsymbol{f}_0$, and the square points are the (Δfreq, Δcost) value of the design when the parameters (ρ_A,c_A) are perturbed. It is clearly shown in Figure 4.21(a) that when (ρ_A,c_A) are perturbed around their nominal values, the changes in objective values of the nominal Pareto design (point A) are larger than the acceptable range. Thus, according to the sensitivity requirements set by the designer, this design is not robust enough. On the other hand, the $\Delta \boldsymbol{f}$ of the robust Pareto design, Figure 4.21(b), is within the acceptable range. So this design (point B) is indeed robust. The objectives variation of the probabilistic Pareto

design (Figure 4.21(c)) violates the sensitivity requirements as well, so not only this design (point C) is inferior, but it is also not robust.

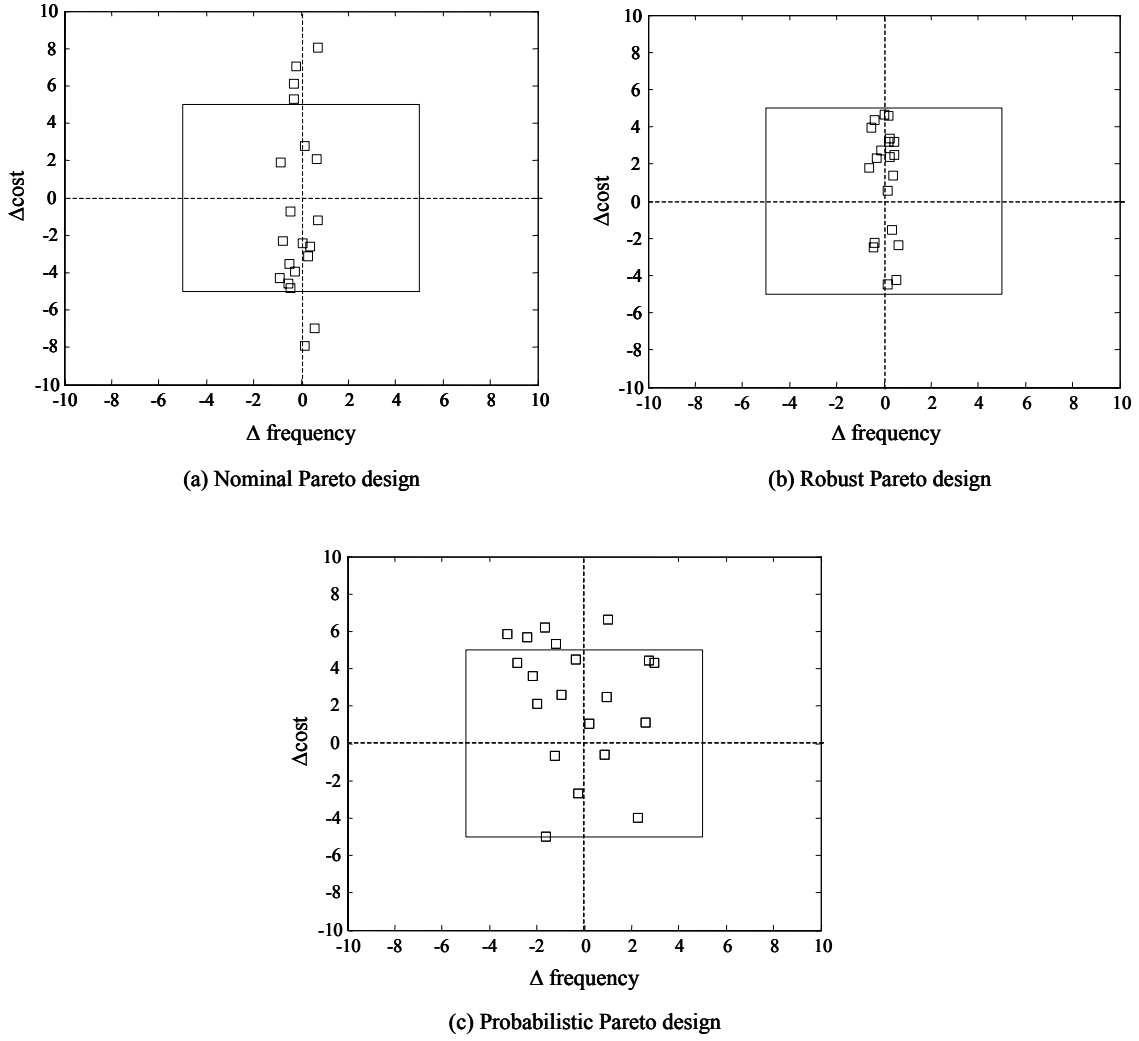


Figure 4.21: Sensitivity of: (a) nominal, (b) robust, and (c) probabilistic designs.

To further assess the sensitivity of the designs, we calculated their sensitivity regions and WCSR's. This is done by substituting their design variable values (b,L,d_1,d_2,d_3,M) into the cost and frequency function of Eq. (4.11), and then using the specified Δf_0 value, solve for $\Delta\rho_A$ and Δc_A. Figure 4.22(a), (b), and (c) show the SR's and WCSR's of the nominal (point A), robust (point B), and probabilistic (point C) Pareto design, respectively.

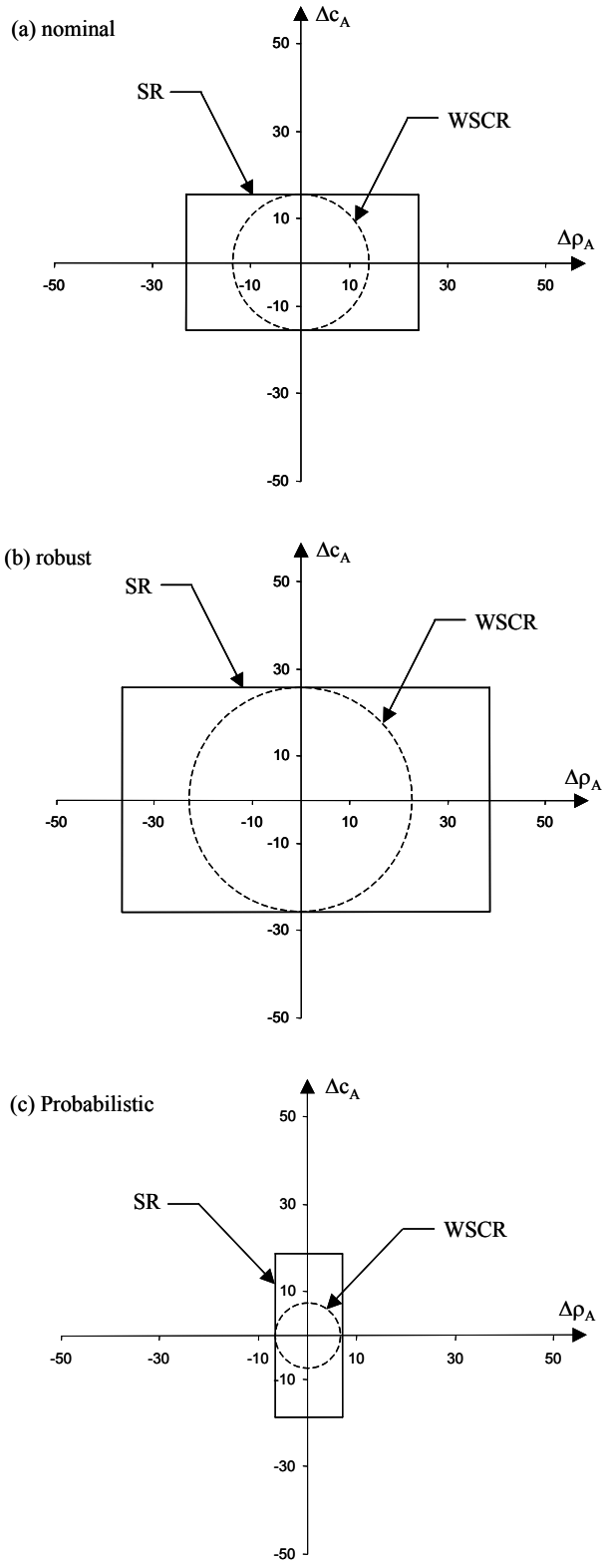


Figure 4.22: SR and WCSR of: (a) nominal, (b) robust, and (c) probabilistic designs.

The rectangles around the origin in Figure 4.22 are the SR's of the designs, while the circles are their WCSR's. The SR's are rectangles because each of the objective functions is a function of only one parameter, i.e., cost(Δc_A) and freq($\Delta \rho_A$). Because of this, each objective will be sensitive only to a range of one parameter value but not the other, and these ranges of values form a rectangle in the Δ**p**-space. The WCSR's are easily formed by finding the point on the SR boundary closest from the origin.

We can make a few observations from Figure 4.22. First, we observe that both the SR and WCSR of the robust Pareto design are larger than that of the nominal and probabilistic Pareto designs, verifying the sensitivity analysis shown in Figure 4.21. Second, we also observe that the robust design satisfies the ($\Delta \rho_{A,0}$, $\Delta c_{A,0}$) = (5 kg/m^3, 25 \$/m^3) range, while the nominal and probabilistic designs do not.

In our MORO approach, we have implicitly assumed that the constraints are still satisfied when the parameters change. To show the validity of this assumption, we performed a sensitivity analysis on the constraints of the robust Pareto design under study. We conducted the analysis by again arbitrarily perturbing the parameters around their nominal values and then calculating the constraints of the design for these perturbed parameters. Table 4.2 shows the values of the g_1 constraint for 20 randomly perturbed ρ_A values (c_A does not affect the constraints, and g_2, g_3, g_4, and g_5 are independent of ρ_A). We see from the table that g_1 is still satisfied when ρ_A changes, so our assumption holds.

Table 4.2: Sensitivity of the constraints.

Case #	g_1	Case #	g_1
1	-2196.5	11	-2199.67
2	-2199.8	12	-2195.41
3	-2198.2	13	-2195.33
4	-2198.5	14	-2198.57
5	-2198.6	15	-2197.84
6	-2198.0	16	-2198.77
7	-2198.1	17	-2199.42
8	-2196.9	18	-2195.66
9	-2196.3	19	-2197.14
10	-2195.4	20	-2196.81

4.5.3. Design of a Speed Reducer

Our third example is the well-known problem of designing a two-shaft speed reducer formulated by Kurapati et al. (2002). Here, we modified it by adding variations to two of the design variables.

Figure 4.23 shows the configuration of the speed reducer to be optimized. The objectives of the problem are to minimize the total weight of the speed reducer as well as the normal stress on the first gear shaft. Since the speed reducer is to be made of the same material throughout, the first objective is the same as minimizing the total volume. The problem has seven design variables: the gear face width (x_1), the teeth module (x_2), the number of teeth pinion (x_3), the distance between bearings on the first shaft (x_4) and on the second shaft (x_5), and the diameter of the first shaft (x_6) and second shaft (x_7). All design variables are continuous except for x_3 (the number of teeth), which is an integer.

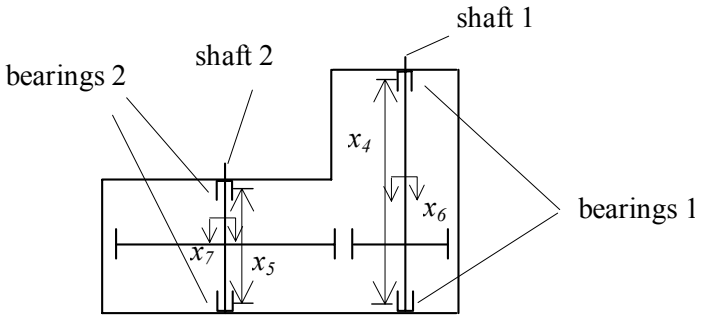


Figure 4.23: A speed reducer.

A lower and upper bound are imposed on each of the design variables. In addition, the design is subject to 11 inequality constraints as required by gear and shaft design practices. The constraints are: upper bound on the bending stress of the gear tooth (g_1), upper bound on the contact stress of the gear tooth (g_2), upper bound on the transverse deflection of the first shaft (g_3) and the second shaft (g_4), dimensional restrictions based on space and/or experience (g_5, g_6, and g_7), design requirements on the shaft based on experience (g_8 and g_9), and upper bound on the normal stress on the first shaft (g_{10}) and on the second shaft (g_{11}). The constraint g_{10} is an objective constraint. The mathematical formulation of the problem is as follows. The units for all the variables are cm (except for x_3 – the integer variable). The unit for the first and second objective is cm^3 and kPa, respectively.

$$\begin{aligned}\text{minimize} \quad f_1 &= 0.7854x_1x_2^2\left(\frac{10x_3^2}{3}+14.933x_3-43.0934\right)\\ &\quad -1.508x_1\left(x_6^2+x_7^2\right)+7.477\left(x_6^3+x_7^3\right)\\ &\quad +0.7854\left(x_4x_6^2+x_5x_7^2\right) \qquad (4.12a)\end{aligned}$$

$$\text{minimize} \quad f_2 = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2+1.69x10^7}}{0.1x_6^3}$$

$$\text{subject to}: \quad g_1 \equiv \frac{1}{x_1 x_2^2 x_3} - \frac{1}{27} \le 0 \quad ; \quad g_2 \equiv \frac{1}{x_1 x_2^2 x_3^2} - \frac{1}{397.5} \le 0$$

$$g_3 \equiv \frac{x_4^3}{x_2 x_3 x_6^4} - \frac{1}{1.93} \le 0 \quad ; \quad g_4 \equiv \frac{x_5^3}{x_2 x_3 x_7^4} - \frac{1}{1.93} \le 0$$

$$g_5 \equiv x_2 x_3 - 40 \le 0 \quad ; \quad g_6 \equiv \frac{x_1}{x_2} - 12 \le 0 \tag{4.12b}$$

$$g_7 \equiv 5 - \frac{x_1}{x_2} \le 0 \quad ; \quad g_8 \equiv 1.9 - x_4 + 1.5 x_6 \le 0$$

$$g_9 \equiv 1.9 - x_5 + 1.1 x_7 \le 0 \quad ; \quad g_{10} \equiv f_2 - 1300 \le 0$$

$$g_{11} \equiv f_3 - 1100 \le 0$$

$$\text{where}: \quad f_3 = \frac{\sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 1.575 x 10^8}}{0.1 x_7^3} \tag{4.12c}$$

$$\begin{aligned} &2.6 \le x_1 \le 3.6 \qquad 0.7 \le x_2 \le 0.8 \\ &17 \le x_3 \le 28 \qquad 7.3 \le x_4 \le 8.3 \\ &7.3 \le x_5 \le 8.3 \qquad 2.9 \le x_6 \le 3.9 \\ &5.0 \le x_7 \le 5.5 \end{aligned} \tag{4.12d}$$

Solving Eq. (4.12) using NSGA, we obtain the nominal Pareto solutions as shown in Figure 4.24. We see in this figure that the nominal Pareto frontier is composed of two regions: a weakly Pareto region (the horizontal portion) and a normally Pareto region (the non-horizontal portion). For multi-objective minimization, a design $\boldsymbol{x}_a$ is termed weakly Pareto if there is no feasible design $\boldsymbol{x}_b$ such that $f_i(\boldsymbol{x}_b) < f_i(\boldsymbol{x}_a)$ for all i=1,…,M. Some might argue that a weakly Pareto frontier is not really a Pareto frontier. However, to be general we include it in our discussion.

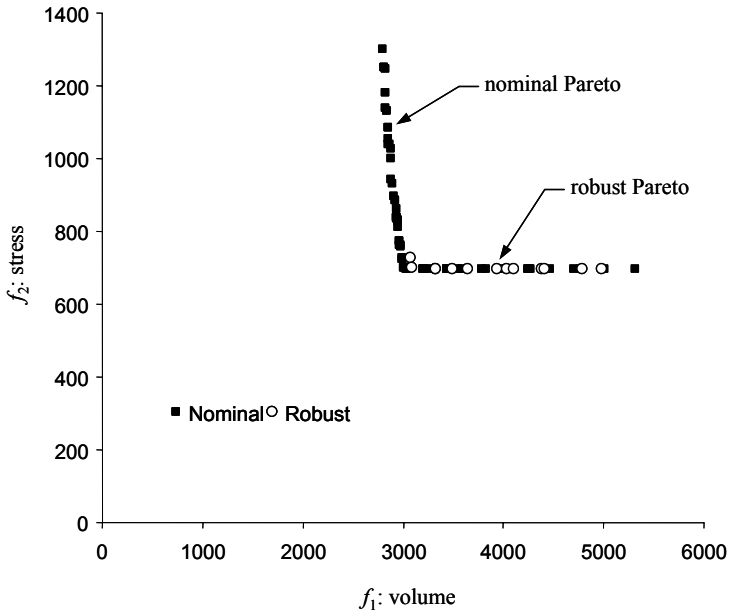


Figure 4.24: Nominal and robust Pareto solutions of the speed reducer problem.

Two of the variables, the teeth module (x_2) and the first shaft diameter (x_6), vary by $(\Delta x_2, \Delta x_6) = (0.01, 0.1)$, and we want to obtain robust optimum solutions to this problem. The acceptable $\boldsymbol{f}$ variation is given to be $(\Delta f_{1,0}, \Delta f_{2,0}) = (150, 75)$. Using $\eta_0 = 1$, we added the robustness constraint $1 - \frac{\eta}{\eta_0} \leq 0$ to Eq. (4.12) and solved it using NSGA. The robust Pareto solutions obtained are also shown in Figure 4.24. We see in this figure that the robust Pareto set is a subset of the nominal Pareto set, and it is located on the weakly Pareto portion of the nominal Pareto frontier.

To verify the robustness of the designs obtained, we randomly perturbed the variables x_2 and x_6 around their values following $(\Delta x_2, \Delta x_6) = (0.01, 0.1)$, and observed the changes in (f_1, f_2) due to these perturbations. To ensure the validity of the analysis, we performed the analysis on two designs from the robust Pareto set. For reference, we call them robust-1 and robust-2 designs. The results of the sensitivity analysis of the two

designs are shown in Figure 4.25. In this figure, the small rectangles are the $\Delta \boldsymbol{f}$ values of the design when x_2 and x_6 are perturbed, while the large inner rectangle is the $\Delta \boldsymbol{f}_0$ limit.

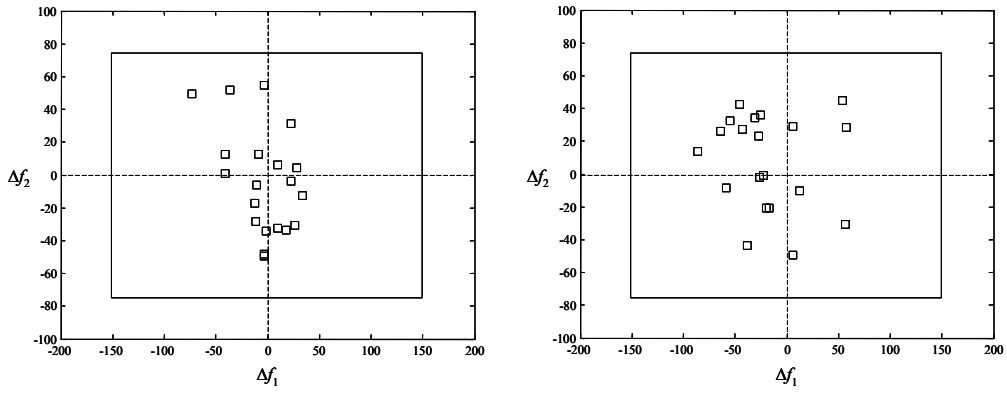


Figure 4.25: Sensitivity analysis of the robust-1 (left) and robust-2 (right) design.

We observe in Figure 4.25 that for both designs, the variations in f_1 and f_2 are always within the $\Delta \boldsymbol{f}_0$ limit, thus verifying that these designs are indeed robust.

For comparison, we performed the same sensitivity analysis for two of the nominal Pareto designs. We chose one optimum design from the non-horizontal portion, and another design from the horizontal portion to be analyzed. For reference, we call them nominal-1 and nominal-2 design, respectively. The results of the sensitivity analysis are shown in Figure 4.26. As before, in this figure the small rectangles are the $\Delta \boldsymbol{f}$ values of the designs, while the large inner rectangle is the $\Delta \boldsymbol{f}_0$ limit.

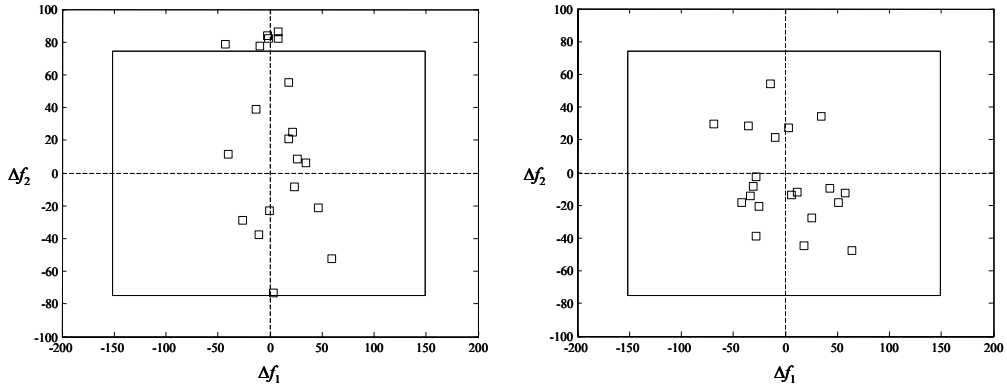


Figure 4.26: Sensitivity analysis of the nominal-1 (left) and nominal-2 (right) design.

We observe in Figure 4.26 that for the nominal-1 design, the variations in f_2 violate the $\Delta f_{2,0}$ limit, while the $\Delta \boldsymbol{f}$ of the nominal-2 design is within the $\Delta \boldsymbol{f}_0$ limit. This observation shows that the designs in the horizontal portion of the nominal Pareto frontier are robust, while those in the non-horizontal portion are not.

To further confirm the results in Figures 4.25 and 4.26, we computed the SR and WCSR of the designs. Since the robustness of the designs has been verified experimentally, here we only show the SR and WCSR of the nominal-1 and robust-1 designs. Analytical derivation of the designs SR is very difficult, if at all possible. So, we performed an exhaustive analysis instead. We partition the ranges $(\Delta x_2, \Delta x_6) = (0.04, 0.4)$ into an equally spaced orthogonal grid, and calculate the $\Delta \boldsymbol{f}$ of the designs on each intersection point. If $(\Delta \boldsymbol{f})^2 \leq (\Delta \boldsymbol{f}_0)^2$, then this point is in the SR. The set of such points is the SR of the design. To be able to derive the WCSR of the designs graphically, we plot the SR in the normalized space. The SR and WCSR of the nominal-1 and robust-1 designs are shown in Figures 4.27(a) and 4.27(b), respectively (in the normalized space).

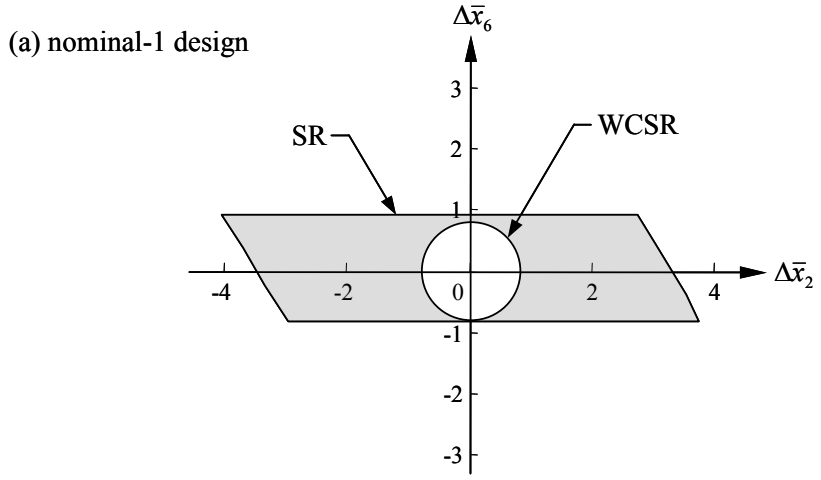


Figure 4.27(a): SR and WCSR of the nominal-1 design.

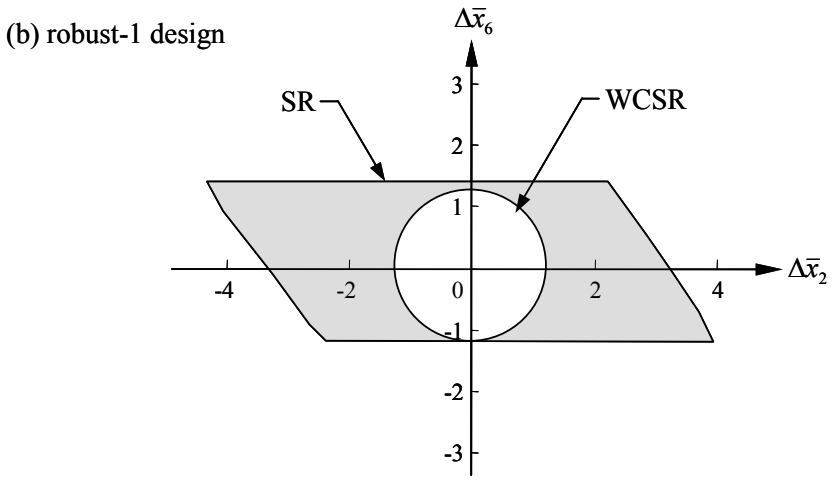


Figure 4.27(b): SR and WCSR of the robust-1 design.

We see in Figure 4.27 that the SR of the robust-1 design encloses the normalized range of variable variation $(\Delta\bar{x}_2, \Delta\bar{x}_6) = (1,1)$, while that of the nominal-1 design does not. This observation again confirms the robustness of each design (or lack thereof).

If we solve the problem for robust optimum solutions using probabilistic methods (by minimizing mean of each objective), we will observe that the probabilistic Pareto solutions are essentially the same as the nominal Pareto solutions. So, to avoid repetition we do not re-show these results in the dissertation (a summary of this result can be found in Table 7.3 in Chapter 7).

4.5.4. Design of a Power Electronic Module

For our last example, we look into the design optimization of a power electronic module as formulated by Palli et al. (1998). The main components of the power module to be optimized are shown in Figure 4.28. In this configuration, a chip/die is attached to a substrate using solder. The substrate is made of alumina and is coated with copper.

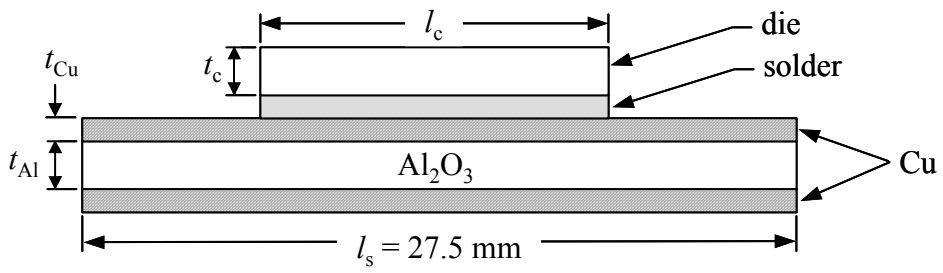


Figure 4.28: A power electronic module.

The four design variables of the problem are: the length of the chip (l_c), the thickness of the chip (t_c), the thickness of the alumina substrate (t_{Al}), and the thickness of the copper coating (t_{Cu}). The units for all design variables are mm.

The power module experiences stresses in the solder layer due to thermal mismatch between the die and the substrate. In addition, the solder also undergoes fatigue due to cyclic stresses created by thermal changes during power switches. The objectives of the optimization problem are to: (1) minimize the maximum shear stress in the solder layer, (2) maximize the life expectancy of the module under chip fatigue, and (3) maximize the life expectancy of the module under substrate fatigue.

The shear stress model used in this problem is based on elastic, plastic, and creep behavior and is governed by the following differential equation:

$$\frac{d^2}{dx^2}\left[\frac{\tau}{G}+C_1\left(\frac{\tau}{G}\right)^{n_p}+\int_0^t \frac{AG}{T}\left(\frac{\beta\tau}{G}\right)^{n_c} e^{\frac{-\Delta H}{kT}}\,dt\right]=\left(\frac{1}{E_c t_c}+\frac{1}{E_s t_s}\right)\frac{\tau}{t_{ca}} \tag{4.13}$$

In this equation, τ is the shear stress, G is the shear modulus, x is the distance from the center of the chip, ΔH is the activation energy, k is the Boltzmann's constant, and the quantities A, β, n_p, and n_c are constants. The boundary conditions for Eq. (4.13) are: (1) the shear stress at the center of the chip is zero, i.e., $\tau(0) = 0$, and (2) the compressive

force caused by the solder displacement is equal to the tensile force caused by the thermal mismatch.

The shear stress is a function of the position from the center of the chip. However, because we are interested only in the maximum shear stress, we need only to calculate the shear stress at the edge of the chip, i.e., at $x = \pm l_c/2$. Following the boundary condition, the shear stress at this position can be estimated using the following function. Here, we have looked into the shear stress at a particular instance of time.

$$\tau_{\max} = \left(\frac{l_c}{2}\right)^2 + \frac{4\left[\mathrm{A} - \frac{1}{3}\left(\frac{l_c}{2}\right)^3\right]}{l_c}$$
$$\text{where : } \mathrm{A} = \frac{\alpha_s - \alpha_c}{\left(\frac{1}{E_c t_c} + \frac{1}{E_s t_s}\right)}\Delta\mathrm{T} \tag{4.14}$$

The expected life of the module under chip fatigue is estimated using Eq. (4.15) following the physics of failure model of Suhir (1987).

$$life_{\text{chip}} = \left[\left(1 - \frac{\varepsilon_{pc}}{2}\right)\left(C_{pc}\right)^{\varepsilon_{pc}}\left(\sigma_{\text{tensile}}\right)^{\varepsilon_{pc}}\pi^{\frac{\varepsilon_{pc}}{2}}\right]^{-1} \times \left[A_{vfc}^{(1-\varepsilon_{pc}/2)} - A_{vic}^{(1-\varepsilon_{pc}/2)}\right] \tag{4.15a}$$

$$\sigma_{\text{tensile}} = \left[\frac{\frac{1}{t_c} + \frac{3(t_s + t_c)E_c t_c}{12(1-\upsilon_c^2)F_1}}{\frac{1-\upsilon_s}{E_s t_s} + \frac{1-\upsilon_c}{E_c t_c} + \frac{t_c t_s}{4F_1}}\right] \times \left[1 - \frac{1}{\cosh(A_{factor} l_c)}\right](\alpha_s - \alpha_c)\Delta\mathrm{T} \tag{4.15b}$$

$$F_1 = \left[\frac{(E_s t_s)^3}{12(1-\upsilon_s^2)}\right] + \left[\frac{(E_{ca} t_{ca})^3}{12(1-\upsilon_{ca}^2)}\right] + \left[\frac{(E_c t_c)^3}{12(1-\upsilon_c^2)}\right] \tag{4.15c}$$

$$A_{factor} = \left[\frac{\left(\frac{1-\upsilon_s}{E_s t_s}\right) + \left(\frac{1-\upsilon_c}{E_c t_c}\right) + \left(\frac{(t_c + t_s)^2}{4F_1}\right)}{\left(\frac{t_c}{3G_c}\right) + \left(\frac{t_s}{3G_s}\right) + \left(\frac{2t_{ca}}{3G_{ca}}\right)}\right]^{1/2} \tag{4.15d}$$

Similarly, the expected module life under substrate fatigue is formulated as follows:

$$life_{\text{substrate}} = \left[\left(1-\frac{\varepsilon_{ps}}{2}\right)\left(C_{ps}\right)^{\varepsilon_{ps}}\left(\sigma_{\text{tensile}}\right)^{\varepsilon_{ps}}\pi^{\frac{\varepsilon_{ps}}{2}}\right]^{-1} \times \left[A_{vfs}^{(1-\varepsilon_{ps}/2)} - A_{vis}^{(1-\varepsilon_{ps}/2)}\right] \tag{4.16a}$$

$$\sigma_{\text{tensile}} = \left[\frac{\frac{1}{t_s}+\frac{3(t_{case}+t_s)E_s t_s}{12(1-\upsilon_s^2)F_2}}{\frac{1-\upsilon_{case}}{E_{case}t_{case}}+\frac{1-\upsilon_s}{E_s t_s}+\frac{t_{case}t_s}{4F_2}}\right] \times \left[1-\frac{1}{\cosh(A_{factor}l_s)}\right](\alpha_{case}-\alpha_s)\Delta\text{T} \tag{4.16b}$$

$$F_2 = \left[\frac{(E_{case}t_{case})^3}{12(1-\upsilon_{case}^2)}\right]+\left[\frac{(E_{sa}t_{sa})^3}{12(1-\upsilon_{sa}^2)}\right]+\left[\frac{(E_s t_s)^3}{12(1-\upsilon_s^2)}\right] \tag{4.16c}$$

$$A_{factor} = \left[\frac{\left(\frac{1-\upsilon_s}{E_s t_s}\right)+\left(\frac{1-\upsilon_{case}}{E_{case}t_{case}}\right)+\left(\frac{(t_{case}+t_s)^2}{4F_2}\right)}{\left(\frac{t_{case}}{3G_{case}}\right)+\left(\frac{t_s}{3G_s}\right)+\left(\frac{2t_{sa}}{3G_{sa}}\right)}\right]^{1/2} \tag{4.16d}$$

The constants used in Eq. (4.14), (4.15), and (4.16) are as follows:

ΔT = temperature difference between chip and substrate (= 10 °K)

A_{vfc} = Length of final vertical crack in chip (= 5.08 x 10^{-5} m)

A_{vfs} = Length of final vertical crack in substrate (= 5.08 x 10^{-5} m)

A_{vic} = Length of initial vertical crack in chip (= 6.35 x 10^{-6} m)

A_{vis} = Length of initial vertical crack in substrate (= 6.35 x 10^{-6} m)

C_{pc} = Paris coefficient for chip fatigue (= 7.53 x 10^{-18})

C_{ps} = Paris coefficient for substrate fatigue (= 5.15 x 10^{-15})

E_c = elastic modulus of chip (= 120 GPa)

E_{ca} = elastic modulus of chip attach (= 11.85 GPa)

E_{case} = elastic modulus of case (= 110 GPa)

E_s = elastic modulus of substrate (= 327 GPa)

E_{sa} = elastic modulus of substrate attach (= 29.8 GPa)

G_c = shear modulus of chip (= 45.1 GPa)

G_{ca} = shear modulus of chip attach (= 4.2 GPa)

G_{case} = shear modulus of case (= 42.3 GPa)

G_s = shear modulus of substrate (= 130.8 GPa)

G_{sa} = shear modulus of substrate attach (= 10.6 GPa)

l_s = length of substrate (= 27.5 mm)

t_{ca} = thickness of chip attach (= 0.08 mm)

t_{case} = thickness of case (= 4.0 mm)

t_s = thickness of substrate (= $t_{Cu} + t_{Al}$)

t_{sa} = thickness of substrate attach (= 0.5 mm)

α_c = coefficient of thermal expansion of chip (= 2.6 x 10^{-6}/°K)

α_{case} = coefficient of thermal expansion of case (= 17 x 10^{-6}/°K)

α_s = coefficient of thermal expansion of solder layer (= 25 x 10^{-6}/°K)

ε_{pc} = Paris exponent for chip fatigue (= 3)

ε_{ps} = Paris exponent for substrate fatigue (= 3.3)

ν_c = Poisson's ratio of chip (= 0.3)

ν_{ca} = Poisson's ratio of chip attach (= 0.4)

ν_{case} = Poisson's ratio of case (= 0.3)

ν_s = Poisson's ratio of substrate (= 0.25)

ν_{sa} = Poisson's ratio of substrate attach (= 0.4)

In addition to the three objectives, there are lower and upper bounds on the design variables. The complete formulation of the optimization problem is as follows:

$$\begin{aligned} &\text{minimize} \quad f_1 = \tau_{\max} \\ &\text{maximize} \quad f_2 = life_{\text{chip}} \\ &\text{maximize} \quad f_3 = life_{\text{substrate}} \end{aligned} \tag{4.17a}$$

$$\begin{aligned} \text{subject to}: \; & 5.0 \le l_c \le 10.0 \\ & 0.1 \le t_{\text{Cu}} \le 1.0 \\ & 0.3 \le t_{\text{Al}} \le 1.5 \\ & 1.0 \le t_c \le 2.0 \end{aligned} \tag{4.17b}$$

The nominal Pareto solutions of the optimization problem, Eq. (4.17), are shown in Figure 4.29 (these solution points are obtained using the NSGA method (Srinivas and Deb, 1995)). Because we have three objectives, we have shown the points projected onto two-dimensional planes. We have also shown the three objectives as all minimization by taking the negative of f_2 and f_3.

Three of the design variables and four of the design parameters have variability in them. The three design variables are (t_c, t_{Al}, t_{Cu}), and their variations are (0.05, 0.1, 0.2), respectively (all in mm). The four design parameters are (α_s, α_c, E_c, E_s), and their variations are (2 x 10^{-6}/°K, 0.1 x 10^{-6}/°K, 6 GPa, 10 GPa), respectively. It is required that the objective values are bounded within ($\Delta f_{1,0}$, $\Delta f_{2,0}$, $\Delta f_{3,0}$) = (3 MPa, 1.0 x 10^{31} cycles, 1.5 x 10^{26} cycles). Adding the robustness constraint $1 - \frac{\eta}{\eta_0} \le 0$ to Eq. (4.17) and solving it with $\eta_0 = 1$, we obtain the robust Pareto solutions as shown in Figure 4.29.

From Figure 4.29, we see that the robust Pareto solutions are a subset of the nominal solutions. We also observe that the robust Pareto solutions are concentrated around the region close to the origin.

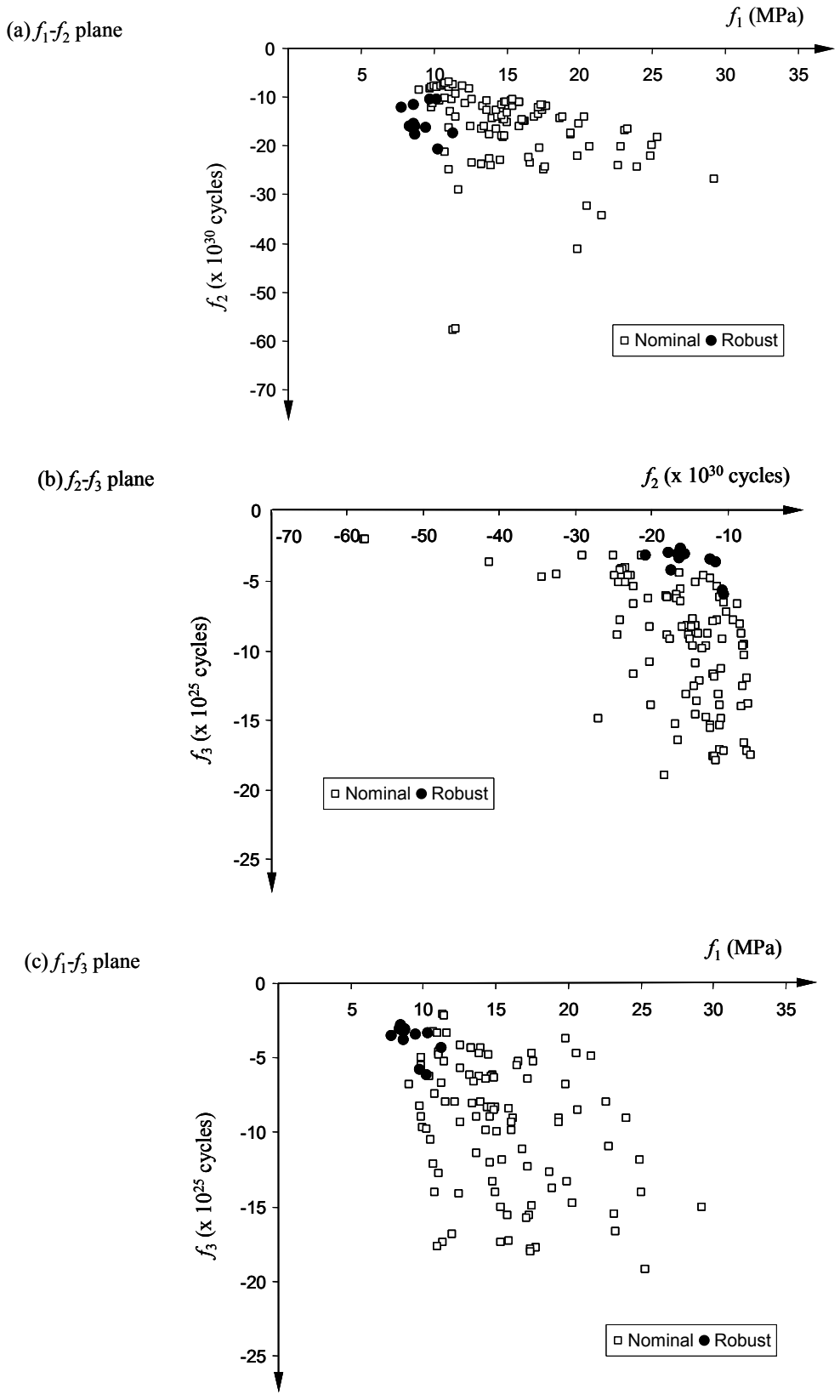


Figure 4.29: Nominal and robust Pareto solutions of the power electronic problem.

Let us now verify the robustness of the solutions obtained. Since we have a total of 7 design factors with variability, it is impossible to show their SR and WCSR on a two-dimensional plane. So, we will perform a numerical sensitivity analysis instead. We analyze the sensitivity of one design each from the nominal and robust solution set. For the nominal design, we choose a design not located in the region near the origin (the robust region). The two designs to be analyzed are shown in Table 4.3. For comparison, we have also shown their η values. Notice in this table that $\eta_{robust} \geq 1.0$ while $\eta_{nominal}$ is not. So according to our WCSR prediction, the robust design satisfies the $\Delta \boldsymbol{f}_0$ limit while the nominal design does not.

Table 4.3: Nominal and robust designs to be analyzed.

	Nominal	Robust
l_c (mm)	6.071	9.754
t_{Cu} (mm)	0.596	0.153
t_{Al} (mm)	1.005	1.073
t_c (mm)	1.962	1.205
f_1 (MPa)	23.978	9.764
f_2 (x 10^{30} cycles)	24.478	10.552
f_3 (x 10^{25} cycles)	9.013	5.789
η	0.418	1.075

We randomly perturb the value of the 7 design factors within their variation range, and then observe the changes in the objective values of the design. A design is robust if the objective changes are within the given limit, for all three objectives. The graphs of the $\Delta \boldsymbol{f}$ of the two designs are shown in Figure 4.30. In this figure, the left three graphs are the Δf_1, Δf_2, and Δf_3 of the nominal design, while the right three graphs are the Δf_1, Δf_2, and

Δf_3 of the robust design. The horizontal axis shows the case number, while the dashed lines are the $\Delta \boldsymbol{f}_0$ limits.

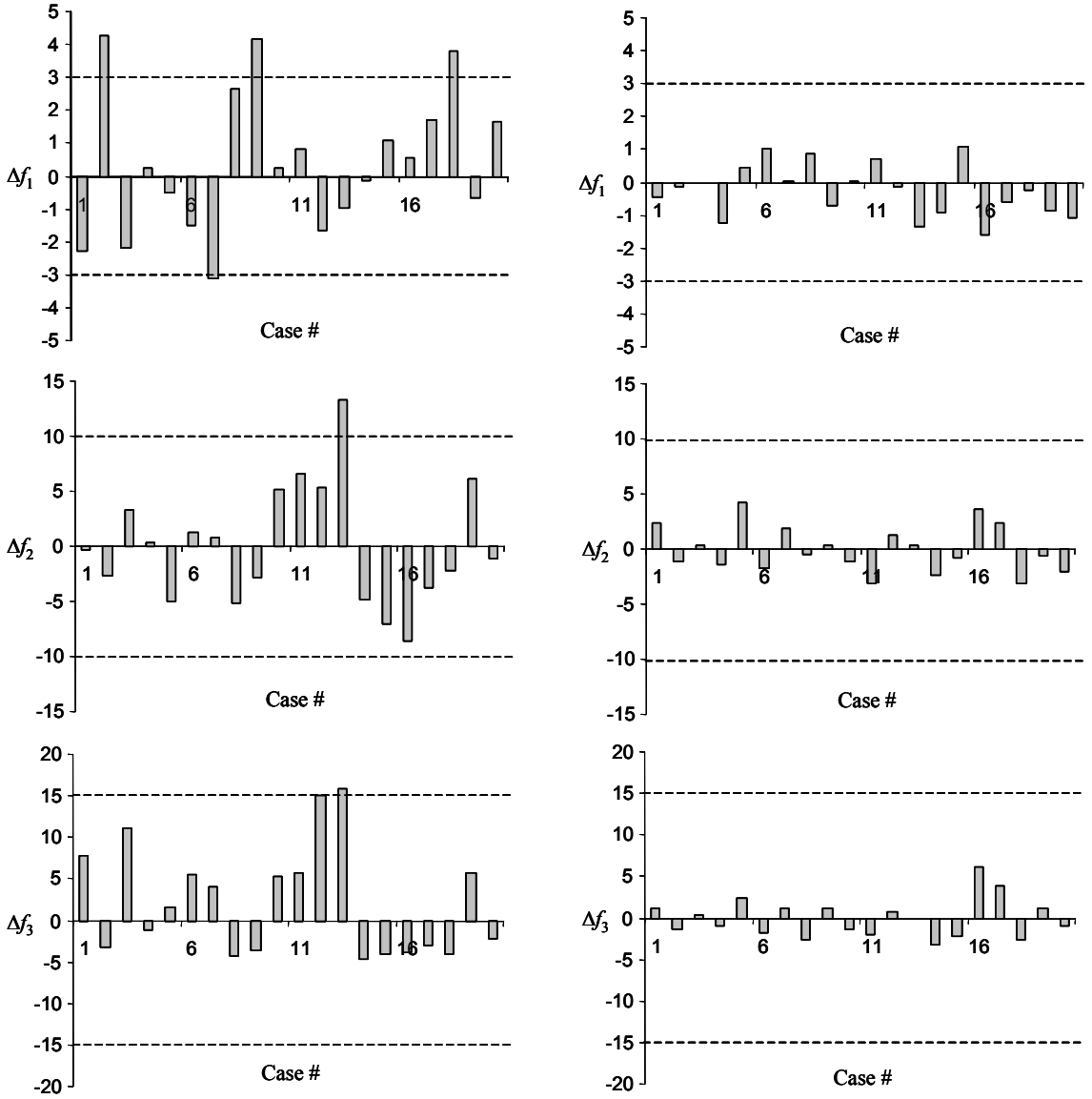


Figure 4.30: Sensitivity analysis of the nominal (left) and robust (right) designs.

We see in Figure 4.30 that the Δf of the robust design is always within the limits for all three objectives. In contrast, the Δf of the nominal design violates the given limits for all three objectives. This observation verifies the robustness of the two designs as claimed.

As in the previous example, solving this problem using probabilistic approach (Monte Carlo simulation with 100,000 samples) gives us essentially the same results as the nominal optimization approach. As such, the optimality and robustness of these solutions are the same as the nominal Pareto solutions, and we do not re-perform the sensitivity analysis here for clarity.

4.6. SUMMARY

- A design is termed "multi-objectively robust" if each of its objective value is insensitive to parameter variations.
- For a set of designs, the overall robustness of the set is determined by the robustness of each of the designs in the set. This type of robustness is called *point robustness*.
- In Multi Objective Robust Optimization (MORO), we are interested in the point robustness of the solutions. Those designs that are multi-objectively optimum (for nominal values of parameters) and also multi-objectively robust are called robust Pareto solutions.
- A robust Pareto set is generally inferior to the nominal Pareto set. It is also common for the robust Pareto set and the nominal Pareto set to overlap, or for the robust Pareto set to be a subset of the nominal Pareto set.
- The multi objective $\mathbf{S}_f$ of a design is the intersection of the $\mathbf{S}_{f,i}$ of each objective. When the number of objectives M = 1, the multi-objective $\mathbf{S}_f$ reduces to the single objective $\mathbf{S}_f$.
- The generalized SR is the plot of the generalized $\mathbf{S}_f$ on the $\Delta\mathbf{p}$-space. A point inside the SR satisfies $[\Delta f_i]^2 < [\Delta f_{i,0}]^2$ for all $i = 1,\ldots,M$. A point outside satisfies $[\Delta f_i]^2$

$> [\Delta f_{i,0}]^2$ for at least one i. A point on the boundary satisfies $[\Delta f_i]^2 \leq [\Delta f_{i,0}]^2$ with a strict equality for at least one i.

- The generalized WCSR is a hyper-sphere inside the generalized SR that touches the SR boundary at the closest point from the origin. The radius of the WCSR can be calculated by solving a single-objective optimization problem with an equality constraint. If the magnitudes of $\Delta\mathbf{p}$ are different, this optimization problem needs to be normalized to account for scale importance.
- To avoid difficulties in making trade-offs between the objectives, we use a constrained approach to MORO where we treat the robustness of a design as a constraint instead of an objective.
- Using the constrained approach, our MORO method obtains the robust Pareto set by first eliminating all non-robust designs and then selecting the set of non-inferior designs from those designs that are not eliminated.

CHAPTER 5

FEASIBILITY ROBUST OPTIMIZATION

5.1. INTRODUCTION

In the previous two chapters we presented a method to measure objective robustness of a design, and a scheme to use that measure to obtain a robust optimum design. In those chapters, we implicitly assumed that the design will always remain feasible when the variations occur. Obviously, this assumption may not be valid in general. Feasibility robustness of a design must be explicitly accounted for and enforced if we want the design to be always feasible even as the uncontrollable parameters vary.

The purpose of this chapter is to develop a method to measure the feasibility robustness of a design, and develop an optimization scheme to use this measure to guarantee the feasibility of a design. Similar to Chapters 3 and 4, the robustness measure presented here is also based on the sensitivity region concept, but with a significant difference. Unlike objective robustness, which was "two-sided", feasibility robustness of a design with respect to constraints is "one-sided". This one-sided feasibility robustness implies that we are interested in limiting the constraint deviation along one direction only, either increase or decrease but not both. In our case, we limit the increase in the constraints because we use the notation $\mathbf{g}(\boldsymbol{x},\mathbf{p}) \leq \mathbf{0}$ for constraints. We will point out this important difference in more detail in this chapter.

For clarity and simplicity, in this chapter we will focus entirely on the feasibility robustness of a design, and we will not account for objective robustness. We will discuss the combined objective and feasibility robust optimization approach in the next chapter.

In the next few sections we develop the concept of one-side sensitivity region for single and multiple constraints, and then present an approach to use it in an optimization routine. At the end of the chapter we give a demonstration of the applications of our method to numerical and engineering examples.

5.2. ONE-SIDED SENSITIVITY MEASURE

We begin our discussions by presenting a method to measure the sensitivity (robustness) of a design in terms of constraint functions using the sensitivity region concept presented in previous chapters. We present our approach for a single constraint case first and then extend it to a more general case of multiple constraints.

5.2.1. Single Constraint

Let $\boldsymbol{x}_0$ be the design alternative whose sensitivity we want to measure, and let $g(\boldsymbol{x}_0,\mathbf{p})$ be the constraint whose change in value is of interest. Constraint $g(\boldsymbol{x}_0,\mathbf{p})$ depends on two factors, the design $\boldsymbol{x}_0$ itself and parameter $\mathbf{p}$. Our goal is to get a measure of how g changes when $\mathbf{p}$ varies by some $\Delta\mathbf{p}$.

Suppose the value of $g(\boldsymbol{x}_0,\mathbf{p})$ is allowed to decrease indefinitely, but is allowed to increase only by some non-negative amount Δg_0 (i.e., a one-sided sensitivity). (We choose to limit the increase in g, instead of the decrease, to be consistent with our constraint notation "$\leq$".) For this Δg_0 increase, there is a set of $\Delta\mathbf{p}$'s such that:

$$\mathbf{S}_g(\boldsymbol{x}_0) = \left\{ \Delta\mathbf{p} \in R^G : g(\boldsymbol{x}_0, \mathbf{p}+\Delta\mathbf{p}) \leq g(\boldsymbol{x}_0, \mathbf{p}) + \Delta g_0 \right\} \tag{5.1}$$

This set of $\Delta\mathbf{p}$'s is called the "feasibility sensitivity set" ($\mathbf{S}_g$) of design $\boldsymbol{x}_0$. Notice that the feasibility sensitivity set $\mathbf{S}_g$ does not have squared terms as in the objective sensitivity set

$\mathbf{S}_f$ that was presented in Chapters 3 and 4. This is because the feasibility sensitivity set is a one-sided sensitivity measure.

Rearranging the inequality in Eq. (5.1), we obtain:

$$g(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p}) - g(\boldsymbol{x}_0,\mathbf{p}) \le \Delta g_0 \tag{5.2}$$

$$\Delta g(\Delta\mathbf{p}) \le \Delta g_0 \tag{5.3}$$

Eq. (5.3) shows an important property of $\mathbf{S}_g$: for all $\Delta\mathbf{p}$'s in $\mathbf{S}_g$, the changes in g due to $\Delta\mathbf{p}$'s are always less than or equal to an allowable increase Δg_0. This implies that $\mathbf{S}_g$ is an indicator of how much $\Delta\mathbf{p}$'s design $\boldsymbol{x}_0$ can "absorb" for it to remain within the allowable limit. As the number of elements in $\mathbf{S}_g$ increases, the design can allow more changes in $\mathbf{p}$. So, like the objective $\mathbf{S}_f$, the larger $\mathbf{S}_g$, the more feasibly robust the design, and if we can make sure that $\Delta\mathbf{p}$ is always a member of $\mathbf{S}_g$, then we are guaranteed that Δg_0 will always be satisfied.

Plotting $\mathbf{S}_g$ in $\Delta\mathbf{p}$-space, we obtain the Feasibility Sensitivity Region (FSR) of the design. The size of FSR corresponds to the size of $\mathbf{S}_g$, and therefore FSR size is also a measure of a design's robustness: the larger it is, the more feasibly robust the design. Based on Eq. (5.3), a $\Delta\mathbf{p}$ point inside, outside, and on the boundary of FSR must satisfy $\Delta g(\Delta\mathbf{p}) < \Delta g_0$, $\Delta g(\Delta\mathbf{p}) > \Delta g_0$, and $\Delta g(\Delta\mathbf{p}) = \Delta g_0$, respectively.

Example 5.1

Let us demonstrate an application of the above concept with a simple constraint function. Suppose we have an inequality constraint: $g(\boldsymbol{x},\mathbf{p}) \equiv x_1^{(10p_1)} + (p_2)^{x_2} \le 0$, where the parameters are $\mathbf{p} = [0, -0.5]$ and the allowable increase for the constraint is

$\Delta g_0 = 1.125$. If the design of interest is $\boldsymbol{x}_0 = [1.1, 3.0]$, determine the feasibility sensitivity set $\mathbf{S}_g$ for this design, and show its FSR in $(\Delta p_1, \Delta p_2)$-space.

Solution

The nominal value of the constraint is $g(\boldsymbol{x}_0, \mathbf{p}) = -0.125$. Using $\Delta g_0 = 1.125$, we obtain: $\mathbf{S}_g(\boldsymbol{x}_0) = \left\{ (\Delta p_1, \Delta p_2) : (1.1)^{10\Delta p_1} + (\Delta p_2 - 0.5)^3 \le 2.0 \right\}$. Plotting this inequality in the $(\Delta p_1, \Delta p_2)$-space, we obtain the FSR of the design as shown in Figure 5.1. As a quick verification, for $(\Delta p_1, \Delta p_2) = (0,0)$, inside the FSR, the inequality is $0.875 < 2.0$. For $(\Delta p_1, \Delta p_2) = (1.0,1.0)$, outside the FSR, the inequality is $2.718 > 2.0$. For $(\Delta p_1, \Delta p_2) = (0,1.5)$, on the FSR boundary, the inequality is $2.0 = 2.0$.

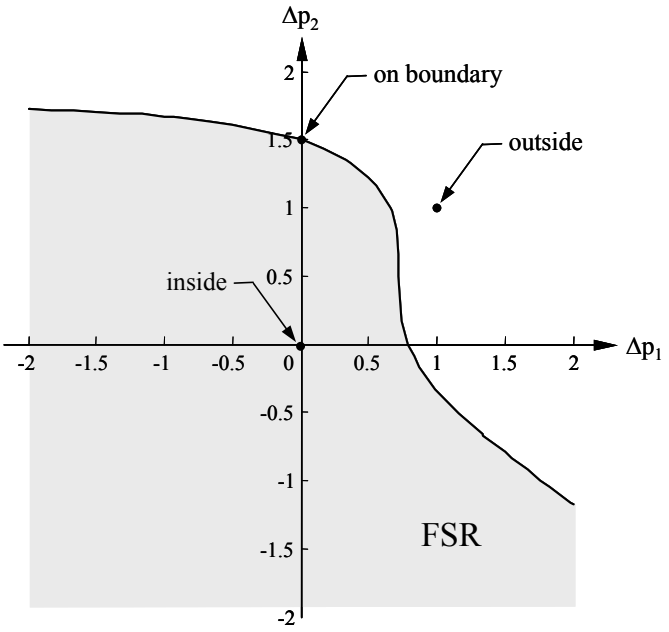


Figure 5.1: Feasibility Sensitivity Region for the inequality constraint.

Notice that the FSR in Figure 5.1 is unbounded and only has one boundary. This is in contrast to the objective SR where even when it is unbounded, it still has two boundaries (recall Chapters 3 and 4). Again, the reason for this difference is because FSR is a one-

sided sensitivity measure. The FSR boundary corresponds to the Δg_0 allowable increase, but since there is no limit for the decrease, there is no other boundary. ♦

As seen in Figure 5.1, like objective SR, FSR also has a drawback of being asymmetric. Because of this asymmetry, the directional sensitivity of a design becomes an important issue. To account for this directional sensitivity, we once again use the worst-case representation of FSR, hereafter called "Feasibility Worst Case Sensitivity Region" (FWCSR), as a measure of a design's feasibility robustness. Graphically, FWCSR of a design is a hyper-sphere inside the FSR of that design that touches the boundary at the closest point from the origin in $\Delta\mathbf{p}$-space. The FWCSR is typically, but not necessarily, tangent to the FSR at the point of contact (e.g., when there is a cusp at the point of contact).

Example 5.2

Using the FSR obtained in Example 5.1, determine the FWCSR of the design $\boldsymbol{x}_0 = [1.1, 3.0]$ for the constraint $g(\boldsymbol{x}, \mathbf{p}) \equiv x_1^{(10p_1)} + (p_2)^{x_2} \leq 0$.

Solution

The FSR of the design is shown in Figure 5.1. The FWCSR of it is simply the smallest hyper-sphere inside it that touches it at the point closest to the origin as shown in Figure 5.2. In this example, the FWCSR is derived from the non-normalized $(\Delta p_1, \Delta p_2)$-space, but it is valid because the scale of Δp_1 and Δp_2 is the same. We will discuss normalization in Section 5.3. ♦

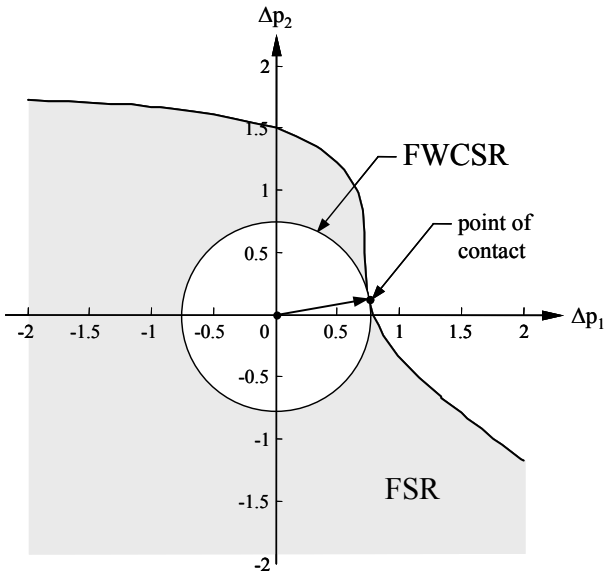


Figure 5.2: Feasibility WCSR for the inequality constraint.

The point of contact between FWCSR and FSR is a point on FSR boundary that is closest from the origin. Since the point of contact satisfies $\Delta g(\Delta \mathbf{p}) = \Delta g_0$, the FWCSR radius can be calculated by solving the following optimization problem:

$$\begin{aligned} &\underset{\Delta \mathbf{p}}{\text{minimize}} \quad R_g(\Delta \mathbf{p}) = \left[\sum_i^G \Delta p_i^2 \right]^{\frac{1}{2}} \\ &\text{subject to}: \; g(\boldsymbol{x}_0, \mathbf{p} + \Delta \mathbf{p}) - g(\boldsymbol{x}_0, \mathbf{p}) = \Delta g_0 \end{aligned} \tag{5.4}$$

Using the radius calculated from Eq. (5.4), we can then calculate the size of FWCSR. However, like in the objective robustness case, there is really no need to perform this calculation. The R_g value alone is sufficient for calculating the feasibility robustness of the design.

Example 5.3

Use Eq. (5.4) to calculate the R_g in Example 5.2. Compare this value to the FWCSR graph shown in Figure 5.2.

Solution

The problem is simple enough to be solved analytically. For ease of differentiation, we use R_g^2 instead of R_g as objective. Taking the first derivative of the Lagrangian with respect to Δp_1 and Δp_2, we obtain the following equations (λ is the Lagrange multiplier):

$$2\Delta p_1 + \lambda(0.953)(1.1)^{10\Delta p_1} = 0 \tag{5.5}$$

$$2\Delta p_2 + 3\lambda(\Delta p_2\text{-}0.5)^2 = 0 \tag{5.6}$$

Combined with the equality constraint: $(1.1)^{(10\Delta p_1)} + (\Delta p_2 - 0.5)^3 - 2.0 = 0$, we have three linearly independent equations with three unknowns (Δp_1,Δp_2,λ). Solving these equations, we obtain the point of contact to be (Δp_1,Δp_2) = (0.749, 0.147) for a FWCSR radius of $R_g = 0.763$ (the Lagrange multiplier is $\lambda = -0.786$). The Hessian of the Lagrangian at this (Δp_1,Δp_2) point is positive definite, so the sufficiency condition is satisfied.

Graphically (see Figure 5.2), the point of contact is approximately at (Δp_1,Δp_2) = (0.75, 0.15) for a FWCSR radius of $R_g = 0.765$. This value is in agreement with the analytic value, thus confirming that solving Eq. (5.4) does indeed give us the FWCSR radius that we are looking for. ♦

5.2.2. Multiple Constraints

We can easily extend the FWCSR concept described in the last section to measure robustness of a design when there are multiple constraints. Let $\mathbf{g}(\boldsymbol{x},\mathbf{p}) = [g_1,\ldots,g_J]$ be the constraint functions of interest, and $\Delta\mathbf{g}_0 = [\Delta g_{1,0},\ldots, \Delta g_{J,0}] \geq \mathbf{0}$ be the vector of acceptable increments. Then there is a set of $\Delta\mathbf{p}$'s such that:

$$\mathbf{S}_{\mathrm{g}}(\boldsymbol{x}_0) = \left\{ \Delta\mathbf{p} \in \mathrm{R}^{\mathrm{G}} : \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}+\Delta\mathbf{p}) \le \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}) + \Delta\mathrm{g}_{j,0} \ , \forall j = 1,...,\mathrm{J} \right\} \tag{5.5}$$

$$\Leftrightarrow \mathbf{S}_{\mathrm{g}}(\boldsymbol{x}_0) = \left\{ \Delta\mathbf{p} \in \mathrm{R}^{\mathrm{G}} : \Delta\mathrm{g}_j(\Delta\mathbf{p}) \le \Delta\mathrm{g}_{j,0} \ , \forall j = 1,...,\mathrm{J} \right\} \tag{5.6}$$

This set is the generalized feasibility sensitivity set $\mathbf{S}_{\mathrm{g}}$. Notice how Eq. (5.6) collapses to Eq. (5.1) when J=1.

Recall from Chapter 4 that the generalized $\mathbf{S}_f$ is really just the intersection of all $\mathbf{S}_{f,i}$'s. The same property also exists for the generalized $\mathbf{S}_{\mathrm{g}}$. If we let the notation $\mathbf{S}_{\mathrm{g},j}$ be the set of $\Delta\mathbf{p}$'s such that $\Delta g_j(\Delta\mathbf{p}) \le \Delta g_{j,0}$, then it is easily seen that the overall $\mathbf{S}_{\mathrm{g}}$ is simply the intersection of all $\mathbf{S}_{\mathrm{g},j}$'s:

$$\mathbf{S}_{\mathrm{g}} = \mathbf{S}_{\mathrm{g},1} \cap \ldots \cap \mathbf{S}_{\mathrm{g,J}} \tag{5.7}$$

Consequently, the overall FSR of a design is then formed by the intersections of the FSR of each constraint, and the overall FWCSR is then defined for this overall FSR.

Utilizing the fact that the overall FSR is an intersection of all constraints' FSRs, we can define the requirements for a $\Delta\mathbf{p}$ point to be inside, outside, or on the boundary of the overall FSR. Similar to the generalized SR, a point inside the overall FSR satisfies $\Delta g_j(\Delta\mathbf{p}) < \Delta g_{j,0}$ for all j=1,…,J. For a point to be outside of the FSR, there must exist at least one j such that $\Delta g_j(\Delta\mathbf{p}) > \Delta g_{j,0}$. A point on the boundary of the FSR satisfies $\Delta g_j(\Delta\mathbf{p}) \le \Delta g_{j,0}$ with a strict equality for at least one j.

Example 5.4

Going back to our inequality constraint from Examples 5.1-5.3. Suppose there is a second constraint $\mathrm{g}_2(\boldsymbol{x},\mathbf{p}) \equiv (2.471\,x_1)^{\left(\frac{\mathrm{p}_2+0.5}{2}\right)} - (x_2 + \mathrm{p}_1) + 2 \le 0$ with the same parameter

values $\mathbf{p} = [0,-0.5]$. If the increment limit for this new constraint is $\Delta g_{2,0} = 1.0$, determine the overall FSR and FWCSR of the design $\boldsymbol{x}_0 = [1.1, 3.0]$.

Solution

We already have the FSR of the first constraint, so to get the overall FSR we only need to calculate FSR of the second constraint and then find the intersection of the two FSRs. From previous examples, the feasibility sensitivity set of the first constraint is $\mathbf{S}_{g,1}(\boldsymbol{x}_0) = \left\{ (\Delta p_1, \Delta p_2) : (1.1)^{10\Delta p_1} + (\Delta p_2 - 0.5)^3 \le 2.0 \right\}$. The nominal value of the second constraint is $g_2(\boldsymbol{x}_0,\mathbf{p}) = 0$, so the feasibility sensitivity set of the second constraint is $\mathbf{S}_{g,2}(\boldsymbol{x}_0) \equiv \left\{ (\Delta p_1, \Delta p_2) : (2.718)^{\Delta p_2/2} - (\Delta p_1 + 2) \le 0 \right\}$. Plotting these inequalities in the $(\Delta p_1,\Delta p_2)$-space, we obtain the overall FSR of the design as shown in Figure 5.3.

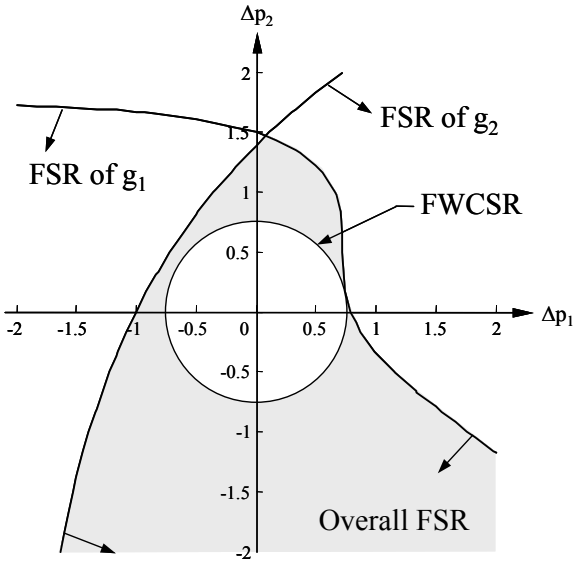


Figure 5.3: Overall FSR and FWCSR for the two constraints.

As a verification, the point $(\Delta p_1,\Delta p_2) = (0,0)$ is inside the overall FSR, and it satisfies both $\mathbf{S}_{g,1}$ inequality ($0.875 < 2.0$) and $\mathbf{S}_{g,2}$ inequality ($-1 < 0$). The point $(\Delta p_1,\Delta p_2) = (1,1)$ is outside the overall FSR. It satisfies $\mathbf{S}_{g,2}$ inequality ($-1.35 < 0$), but not $\mathbf{S}_{g,1}$ inequality

(2.718 > 2.0). The point $(\Delta p_1,\Delta p_2) = (0,2)$ is also outside the overall FSR, but it does not satisfy either $\mathbf{S}_{g,1}$ inequality (4.375 > 2.0) or $\mathbf{S}_{g,2}$ inequality (0.718 > 0). The point $(\Delta p_1,\Delta p_2) = (-1,0)$ is on the boundary of the overall FSR. It satisfies $\mathbf{S}_{g,1}$ inequality (0.26 < 2.0), and satisfies $\mathbf{S}_{g,2}$ inequality with a strict equality (0 = 0).

The overall FWCSR of the design is also shown in Figure 5.3. We see in this figure that the overall FWCSR is the same as the FWCSR of the first constraint. This is because the worst-case scenario for this particular design is governed by the first constraint. ♦

Using the requirement for a point to be on the overall FSR boundary, the FWCSR radius of a multiple-constraint design can be calculated by solving the optimization problem shown below (notice again how Eq. (5.8) collapses into Eq. (5.4) when J=1):

$$\begin{aligned} \underset{\Delta\mathbf{p}}{\text{minimize}} \quad & \mathrm{R_g}(\Delta\mathbf{p}) = \left[\sum_{i}^{\mathrm{G}} \Delta \mathrm{p}_i^2\right]^{\frac{1}{2}} \\ \text{subject to}: \quad & \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}+\Delta\mathbf{p}) - \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}) \le \Delta \mathrm{g}_{j,0} \\ & j = 1,\ldots,\mathrm{J} \\ & \exists j : \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}+\Delta\mathbf{p}) - \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}) = \Delta \mathrm{g}_{j,0} \end{aligned} \tag{5.8}$$

Notice in Eq. (5.8) that at least one of the J inequality constraints must be active. Based on this information, we can simplify these constraints.

If we rearrange each inequality constraint, we obtain the following:

$$\begin{aligned} & \frac{\mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}+\Delta\mathbf{p}) - \mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p})}{\Delta \mathrm{g}_{j,0}} - 1 \le 0 \\ & j = 1,\ldots,\mathrm{J} \end{aligned} \tag{5.9}$$

For a feasible $\Delta\mathbf{p}$ point, the maximum of each of the modified constraint is 0, and for a constraint to be active, it has to be at its maximum. Therefore, for a feasible $\Delta\mathbf{p}$ point with at least one active constraint, Eq. (5.10) below must be satisfied:

$$\max_{j=1,\ldots,J}\left[\frac{g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})-g_j(\boldsymbol{x}_0,\mathbf{p})}{\Delta g_{j,0}}\right]-1=0 \tag{5.10}$$

Substituting Eq. (5.10) into Eq. (5.8), the optimization problem to calculate the FWCSR radius of a multiple-constraint design becomes as shown in Eq. (5.11). (Notice again how Eq. (5.11) simplifies to Eq. (5.4) when J=1.)

$$\begin{aligned}&\underset{\Delta\mathbf{p}}{\text{minimize}} \quad R_g(\Delta\mathbf{p})=\left[\sum_{i=1}^{G}\Delta p_i^2\right]^{\frac{1}{2}}\\ &\text{subject to}: \max_{j=1,\ldots,J}\left[\frac{\Delta g_j(\Delta\mathbf{p})}{\Delta g_{j,0}}\right]-1=0\\ &\text{where}: \quad \Delta g_j(\Delta\mathbf{p})=g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})-g_j(\boldsymbol{x}_0,\mathbf{p})\end{aligned} \tag{5.11}$$

The constraint in Eq. (5.11) is identical to the constraint used in calculating the WCSR radius of the objective SR (Eq. (4.5)) except that this time we do not take the absolute value of the nominator, the $\Delta g_j(\Delta\mathbf{p})$ term. This is again because this constraint is for a one-sided sensitivity measure. In the objective WCSR, the absolute value is necessary to guarantee that the $\Delta\boldsymbol{f}$ value is bounded in both increasing and decreasing directions. In the feasibility WCSR, we only need to guarantee that the $\Delta\mathbf{g}$ value is bounded in the increasing direction, so there is no need for the absolute value.

Alternatively, we can also justify the simplified constraint in Eq. (5.11) graphically. In the **g**-space, the quantity $\Delta\mathbf{g}(\Delta\mathbf{p})$ is represented as a vector from the point $\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})$ to the point $\mathbf{g}(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})$. Based on the definition of $\mathbf{S}_g$, if a point $\Delta\mathbf{p}$ is on the FSR boundary, then the vector $\Delta\mathbf{g}(\Delta\mathbf{p})$ must touch the boundary of $\Delta\mathbf{g}_0$, which implies that at least one of its component vectors must touch this boundary. Coupled with the fact that $\Delta\mathbf{g}_0 \geq \mathbf{0}$, this

implies that $\max_{j=1,\ldots,J}\left[\frac{\Delta g_j(\Delta \mathbf{p})}{\Delta g_{j,0}}\right]=1$, which is the constraint in Eq. (5.11). Figure 5.4 shows an illustration of this derivation.

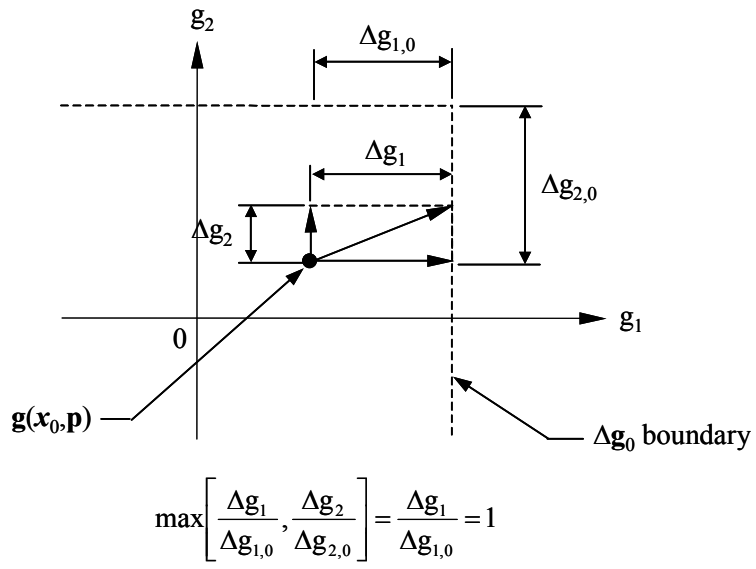


Figure 5.4: Graphical derivation of the simplified constraint.

Since the single-constraint FSR and FWCSR is just a simplified version of the multiple-constraint case, from hereon our discussion will focus only on the multiple-constraint case.

5.3. ROBUST OPTIMIZATION

We will now show how to use the sensitivity measure presented earlier in a feasibility robust optimization scheme.

5.3.1. Determination of Increment Limit

In developing the FWCSR concept, we have assumed there exists a quantity $\Delta\mathbf{g}_0$ that defines how much $\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})$ is allowed to increase, but we have not yet discussed how this

quantity is determined. Unlike the objective $\Delta \boldsymbol{f}_0$ limit, the $\Delta \mathbf{g}_0$ limit is not determined by the designer. Rather it is determined by the position of the design relative to the constraints boundary.

Suppose a design $\boldsymbol{x}_0$ is feasible, i.e., $\mathbf{g}(\boldsymbol{x}_0,\mathbf{p}) \le 0$. Then the value $|\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})|$ shows how much $\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})$ must increase for it to become active. In other words, the quantity $|\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})|$ represents the maximum allowable increment in $\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})$ for the design to remain feasible, i.e., $\Delta \mathbf{g}_0 = |\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})|$. If $\boldsymbol{x}_0$ is infeasible, then there is no need to calculate the feasibility robustness of this design (feasibility cannot be guaranteed). Here, we must use the absolute value because we use the convention $\mathbf{g}(\boldsymbol{x},\mathbf{p}) \le \mathbf{0}$, i.e., a feasible $\boldsymbol{x}$ has a negative $\mathbf{g}$ value. If we use the convention $\mathbf{g}(\boldsymbol{x},\mathbf{p}) \ge \mathbf{0}$ for a feasible design, then the absolute value is not necessary (but then we have to limit the decrease in $\mathbf{g}$).

Substituting $\Delta \mathbf{g}_0 = |\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})|$ into the constraint in Eq. (5.11), we obtain:

$$\max_{j=1,\ldots,J}\left[\frac{g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})-g_j(\boldsymbol{x}_0,\mathbf{p})}{\left|g_j(\boldsymbol{x}_0,\mathbf{p})\right|}\right]-1=0 \tag{5.12}$$

Since for a feasible design, we have $g_j(\boldsymbol{x}_0,\mathbf{p}) \le 0$ and $|g_j(\boldsymbol{x}_0,\mathbf{p})| \ge 0$ for all j:

$$\max_{j=1,\ldots,J}\left[\frac{g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})}{\left|g_j(\boldsymbol{x}_0,\mathbf{p})\right|}\right]=0 \tag{5.13}$$

Let us assume for a moment that $|\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})| > \mathbf{0}$, i.e., when $\boldsymbol{x} = \boldsymbol{x}_0$ no constraint is active. Then Eq. (5.13) simplifies into:

$$\max_{j=1,\ldots,J}[g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})]=0 \tag{5.14}$$

Recall that the constraint in Eq. (5.11) is the requirement for a point $\Delta\mathbf{p}$ to be on the FSR boundary. Eq. (5.14) further simplifies this requirement. Eq. (5.14) states that if a point

$\Delta\mathbf{p}$ causes any $g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p})$ to be active, then it is on the FSR boundary. This simplified requirement is intuitively obvious. Because $\Delta g_{j,0} = |g_j(\boldsymbol{x}_0,\mathbf{p})|$, if $\Delta g_j(\Delta\mathbf{p}) = \Delta g_{j,0}$, then $g_j(\boldsymbol{x}_0,\mathbf{p}+\Delta\mathbf{p}) = 0$. This FSR boundary condition is still valid even when $|\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})| = \mathbf{0}$. So, Eq. (5.14) is valid for $|\mathbf{g}(\boldsymbol{x}_0,\mathbf{p})| \geq \mathbf{0}$.

Substituting Eq. (5.14) into Eq. (5.11), the optimization problem to calculate the FWCSR radius of a design becomes:

$$\begin{aligned} &\underset{\Delta\mathbf{p}}{\text{minimize}} \quad \mathrm{R_g}(\Delta\mathbf{p}) = \left[\sum_{i=1}^{\mathrm{G}} \Delta \mathrm{p}_i^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{j=1,\ldots,\mathrm{J}} \left[\mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}+\Delta\mathbf{p})\right] = 0 \end{aligned} \tag{5.15}$$

5.3.2. Normalization and Feasibility Robustness Index

To account for the scale difference among the parameters, Eq. (5.15) needs to be normalized. Even if the scale of the parameters is the same, we still recommend normalizing Eq. (5.15) to help with convergence. As in objective robustness, we use the known ranges of variations $\Delta\mathbf{p}_0$ to normalize Eq. (5.15). Since $\Delta\overline{\mathrm{p}}_i = \dfrac{\Delta \mathrm{p}_i}{\Delta \mathrm{p}_{i,0}}$ for all i=1,…,G, then $\Delta \mathrm{p}_i = \Delta\overline{\mathrm{p}}_i \cdot \Delta \mathrm{p}_{i,0}$; or in vector notation $\Delta\mathbf{p} = \Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0$. Substituting this equality into Eq. (5.15), the normalized optimization problem to calculate the FWCSR radius becomes as follows (for simplicity, we have forgone using the $\overline{\mathrm{R}}_\mathrm{g}$ notation).

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \quad \mathrm{R_g}(\Delta\overline{\mathbf{p}}) = \left[\sum_{i=1}^{\mathrm{G}} \Delta\overline{\mathrm{p}}_i^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{j=1,\ldots,\mathrm{J}} \left[\mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p}+\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)\right] = 0 \end{aligned} \tag{5.16}$$

If we define $\eta_g = (G)^{-1/2} R_g$ to be the feasibility robustness index of the design, then the value $\eta_g \geq 1$ implies that the design is always feasible even when the variations occur, i.e., it is feasibly robust. In contrast, the value $\eta_g < 1$ implies that there will be instances during the changes in parameters when the design will become infeasible. Adding the robustness constraint $\eta_g \geq 1$ to an optimization problem guarantees that the optimum design is feasibly robust. Eq. (5.17) shows the overall feasibility robust optimization problem.

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}),..., f_M(\boldsymbol{x})] \\ &\text{subject to}: \ g_j(\boldsymbol{x}, \mathbf{p}) \leq 0 \quad j = 1,..., J \\ &\qquad 1 - \frac{\eta_g(\boldsymbol{x})}{\eta_{g,0}} \leq 0 \end{aligned} \tag{5.17}$$

Here $\eta_g(\boldsymbol{x})$ is the feasibility robustness index calculated from Eq. (5.16), and $\eta_{g,0}$ is the desired level of robustness as determined by the designer.

There are a few important things needs to be pointed out about Eq. (5.17). First, the robustness constraint guarantees feasibility robustness of an optimum design with respect to inequality constraints only. Equality constraints are hard constraints in the sense that unless $\Delta\mathbf{p}$ variations are such that $\mathbf{h}(\boldsymbol{x},\mathbf{p}+\Delta\mathbf{p}) = \mathbf{0}$, there is no way to guarantee these constraints will always be satisfied. Second, although we have presented our feasibility robust optimization for a multi-objective optimization problem, our approach is also applicable to single objective problems. The feasibility robustness constraint in Eq. (5.17) does not depend on the number of objectives. Third, implementation-wise, solving Eq. (5.17) as it is will give us a feasibly robust optimum design(s). However, we can improve the efficiency of the approach, i.e., reduce the number of evaluations, simply by not calculating the robustness constraint when the other constraints (both equalities and

inequalities) are not satisfied. A design can be guaranteed to be always feasible (when parameter variations occur) only if it is feasible in the first place (i.e., as a nominal design). If a nominal design is infeasible, then it does not even meet our design requirements, so there is no need to calculate its robustness.

5.4. COMPARISON STUDY

As a demonstration of our feasibility robust optimization method, we applied it to one numerical and three engineering examples. In the numerical example, we show the applicability of our method to problems whose parameter variations are large. In the explosive actuator and control valve linkage examples, we provide comparison between our method and a probabilistic method. In the Belleville spring example, we compare our method to the min-max method developed by Hirokawa and Fujita (2002).

5.4.1. Numerical Example

For our first example, we solve a two-dimensional numerical example presented in Hirokawa and Fujita (2002). The problem is a single objective optimization problem with two inequality constraints. The mathematical formulation of the problem is as follows:

$$\begin{aligned} \underset{\boldsymbol{x}=[x_1,x_2]}{\text{minimize}} \quad & f(\boldsymbol{x}) = x_1^3 \sin(x_1+4) + 10x_1^2 + 22x_1 + 5x_1x_2 + 2x_2^2 + 3x_2 + 12 \\ \text{subject to}: \quad & g_1(\boldsymbol{x}) \equiv x_1^2 + 3x_1 - x_1 \sin x_1 + x_2 - 2.75 \le 0 \\ & g_2(\boldsymbol{x}) \equiv -\log(0.1x_1 + 0.41) + x_2 e^{-x_1+3x_2-4} + x_2 - 3 \le 0 \end{aligned} \tag{5.18}$$

There are no restrictions on the design variables $\boldsymbol{x} = [x_1,x_2]$, but they observe some variations $\Delta\boldsymbol{x}_0 = [0.4,0.4]$. Our goal is to obtain a feasibly robust optimum design that will always remain feasible when the variations in $\boldsymbol{x}$ occur.

Figure 5.5 shows a contour graph for the objective and constraint functions of the problem. The two solid lines are the constraint boundaries with the feasible directions indicated by the arrows. The dashed ellipses are the contours of the objective function, and the arrow on the objective contours indicates the decreasing direction. From this graph we can easily see that the nominal optimum of this problem occurs at the point where the g_2 constraint boundary is tangent to the objective contour (indicated in the graph).

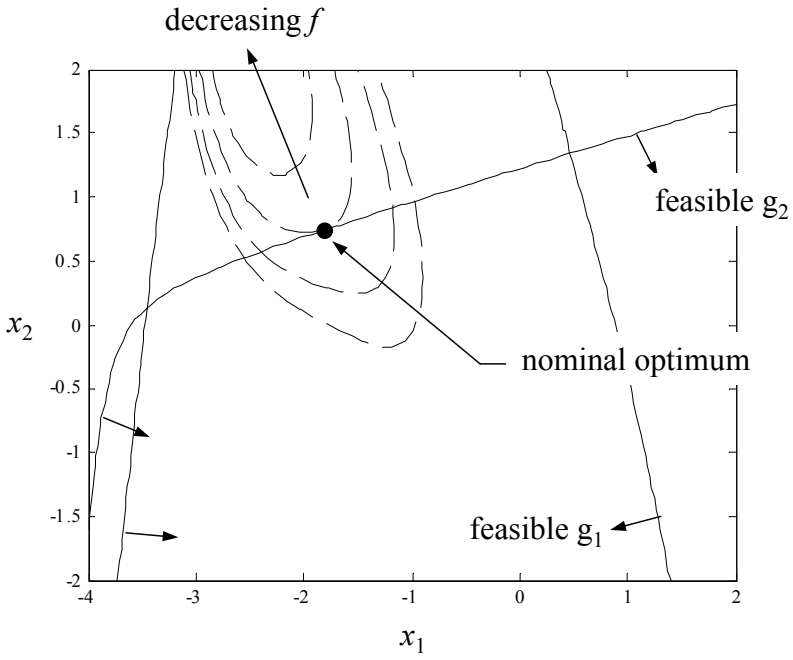


Figure 5.5: Objective and constraint contours of the numerical example.

Solving the problem numerically using MATLAB's *fmincon* function, the nominal optimum is found to be at $\boldsymbol{x}^* = [-1.825, 0.741]$ and the optimum objective is $f^* = -3.287$. The constraint value of the nominal optimum is $\mathbf{g}^* = [-5.919, 0]$. This solution is obtained in 8 iterations. We see that numerically, at the nominal optimum the constraint g_2 is active just as observed in Figure 5.5.

Since the constraint g_2 is active, obviously the nominal optimum is not feasibly robust. When the variables $[x_1,x_2]$ vary, the g_2 constraint can be violated. Adding the feasibility robustness constraint $1-\frac{\eta_g(\boldsymbol{x})}{\eta_{g,0}} \le 0$ to Eq. (5.18) with $\eta_{g,0} = 1.0$, and then solving it using *fmincon*, we obtain the robust optimum design to be $\boldsymbol{x}_R^* = [-1.394, 0.272]$, $f_R^* = -1.552$, and $\mathbf{g}_R^* = [-6.089, -1.374]$. This robust optimum is obtained in 7 iterations. The inner optimization problem used to calculate η_g is also solved using *fmincon* and converges on average in 6.28 iterations.

For comparison, let's also find the feasibly robust optimum design using the conventional robustness methods in the literature. More specifically, let us solve the problem for robust optimum using: (1) the worst-case gradient method where the constraint $g_j(\boldsymbol{x}) \le 0$ is replaced by $g_j(\boldsymbol{x}) + \sum_{i=1}^{N} \left| \frac{\partial g_j}{\partial x_i} \right| \Delta x_i \le 0$, and (2) the moment matching method where the analytic constraint is replaced by a probabilistic constraint $g_j(\mu_{\boldsymbol{x}}) + k\sigma_{g_j} \le 0$, where the standard deviation of g_j is estimated using a Taylor series expansion: $\sigma_{g_j}^2 = \sum_{i=1}^{N} \left[\frac{\partial g_j}{\partial x_i} \right]^2 \sigma_{x_i}^2$, where $\sigma_{x_i}^2$ is the variance of x_i. The factor k in the probabilistic constraint is specified to be equal to 3.0. According to Parkinson et al. (1993), a factor k=3.0 will provide more than 0.99 probability of constraint satisfaction. For the probabilistic constraint, the $\Delta\boldsymbol{x}$ variation is assumed to be normally distributed with a $[\mu_x,\sigma_x] = [0, 0.4/3]$.

The robust optimum designs obtained using the two methods are shown in Table 5.1. For comparison, we have also re-listed the nominal optimum and the robust optimum obtained by our method.

Table 5.1: Optimum designs of the numerical example.

	Nominal	Robust (FWCSR)	Robust (Gradient)	Robust (Moment)
x_1	-1.825	-1.394	-1.557	-1.599
x_2	0.741	0.272	0.477	0.517
f	-3.287	-1.552	-2.266	-2.419
g_1	-5.919	-6.089	-6.077	-6.071
g_2	0	-1.374	-0.98	-0.874

If we consider Table 5.1, we see that the nominal optimum has the smallest f value, while our robust optimum has the largest. The other two robust optima have roughly the same f value, and they are somewhere between the nominal and our robust optimum. However, if we look at the value of the g_2 constraint, we also see that our robust optimum provides that largest amount of 'cushion' for the constraint to vary. The other two robust optima provide some amount of variation cushion, while the nominal optimum provides no room for variation at all ($g_2 = 0$).

At a first glance, it might seem that our robust optimum is too conservative and is an overly non-optimal design. However, let's verify the robustness of each of the optimum designs in Table 5.1. The variations occur in the design variables, so we can use the feasible region graph in Figure 5.5 to see what happens when $\boldsymbol{x}$ varies. For each optimum design, we add a rectangle of ±0.4 around it to indicate the $\Delta\boldsymbol{x}$ variation. An optimum design is truly feasibly robust only if the entire rectangle is inside the feasible region. Figure 5.6 shows the plots of the robustness of each optimum design.

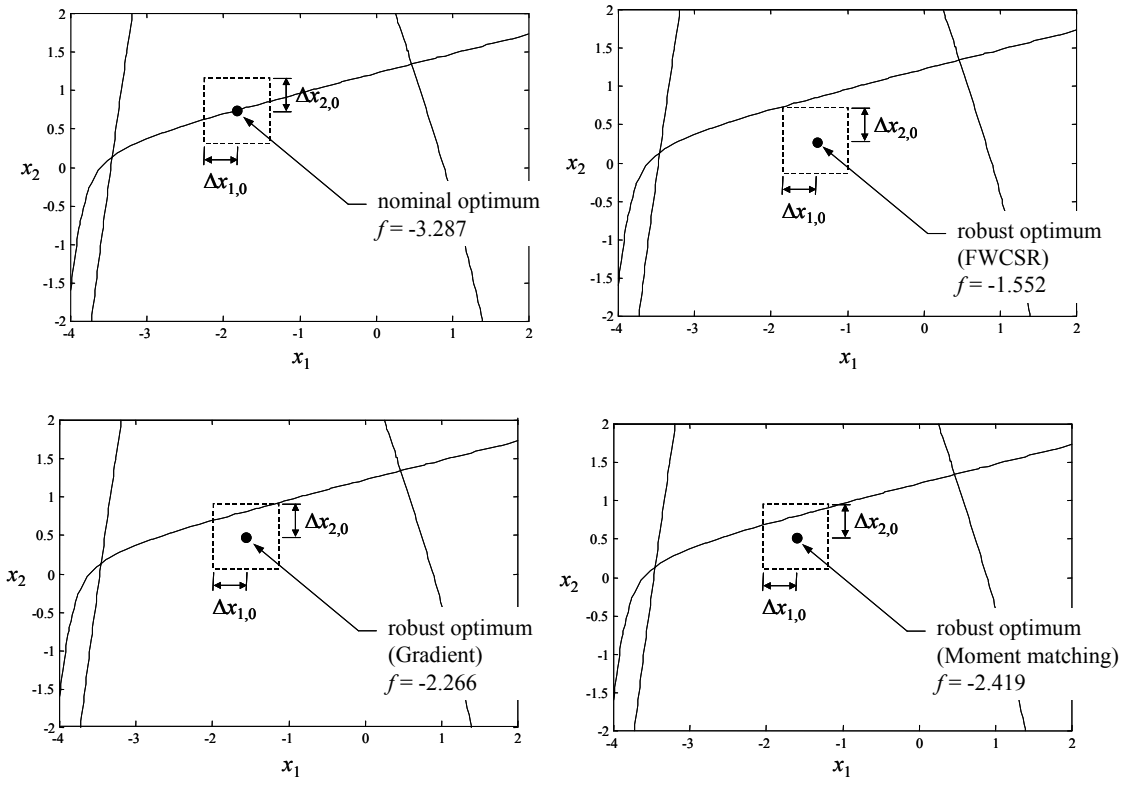


Figure 5.6: Sensitivity analysis of the optimum designs.

We see in Figure 5.6 that of the four optimum designs, only our robust optimum can completely absorb the $\Delta \boldsymbol{x}_0$ variation, i.e., the dashed rectangle is fully inside the feasible region. For the other three optima, there are $\Delta \boldsymbol{x}$ variations that cause the design to fall into the infeasible region. This shows that it is not that our robust optimum is too conservative; rather it is the other optima that are not robust enough. This also shows that the gradient and moment matching methods do not guarantee a feasibly robust optimum.

One reason the gradient and moment matching method do not result in a feasibly robust optimum in this problem is because the $\Delta \boldsymbol{x}_0$ variation is too large for the Taylor series expansion to remain valid. This shows the advantage of our robust optimization method: it is applicable to problems where the variations are large. Let us verify this claim. We re-solve the same problem using four methods: (1) our FWCSR method with

$\eta_{g,0} = 1.0$, (2) our FWCSR method with $\eta_{g,0} = 0.8$, (3) gradient method, and (4) moment matching method. We solve the problem using these methods for 6 increasing amount of $\Delta \boldsymbol{x}_0$ variations (the same for both Δx_1 and Δx_2): [0.01, 0.1, 0.2, 0.4, 0.8, 1.0]. For each optimum obtained, we calculate the probability of constraint satisfaction (P_s) of the design by performing 1000 runs of Monte Carlo simulation assuming a normal pdf of the variations. The plot of the $\Delta \boldsymbol{x}_0$ vs. P_s for each method is shown in Figure 5.7 and 5.8.

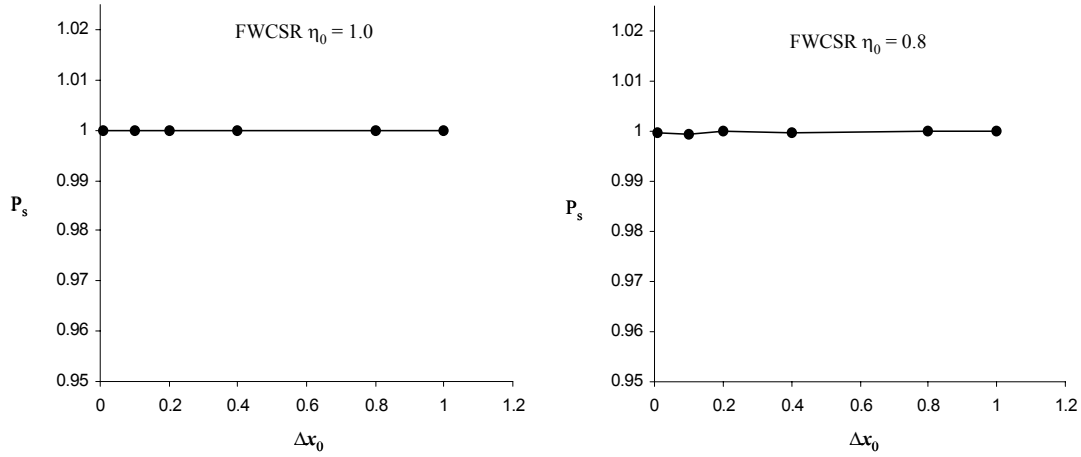


Figure 5.7: Plot of $\Delta \boldsymbol{x}_0$ vs. P_s of the FWCSR optima.

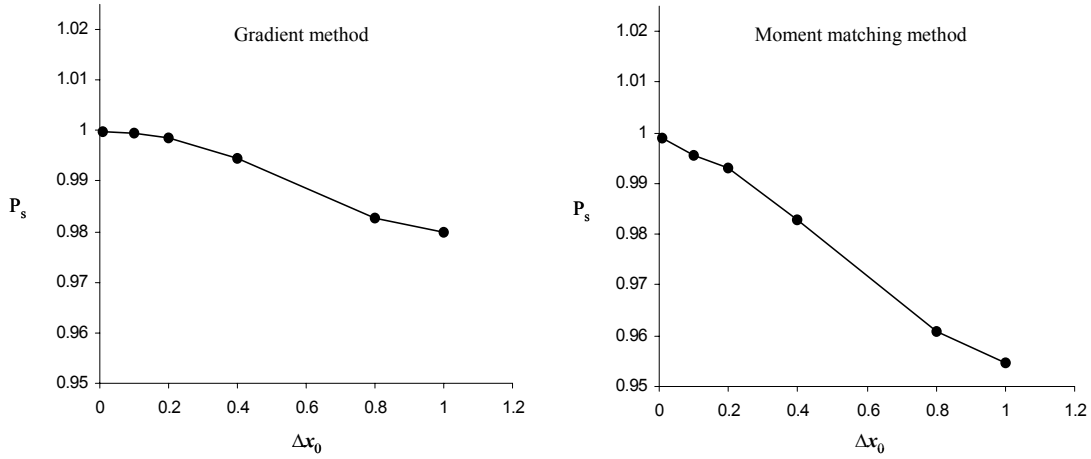


Figure 5.8: Plot of $\Delta \boldsymbol{x}_0$ vs. P_s of the gradient and moment matching optima.

We see in Figure 5.7 that for the FWCSR method with $\eta_{g,0} = 1.0$, P_s is always 1.0 regardless of how large $\Delta \boldsymbol{x}_0$ is. This shows that our method will always obtain a feasibly robust optimum even when the variations are large. Similarly, for the FWCSR method with $\eta_{g,0} = 0.8$, P_s is relatively the same (between 0.9995 and 1.0) even as $\Delta \boldsymbol{x}_0$ increases. This again indicates that our method will still obtain a feasibly robust optimum even as the variations grow beyond the linear range (range in which Taylor series expansion is valid).

In contrast, as seen in Figure 5.8, P_s of the gradient method is very high for small value of $\Delta \boldsymbol{x}_0$, but decreases as $\Delta \boldsymbol{x}_0$ increases. This indicates that this method fails to obtain a robust optimum as the variations grow large. Even worse behavior is observed for the moment matching method. When is $\Delta \boldsymbol{x}_0$ small, P_s is high, but it decreases dramatically (to as low as 0.954) as $\Delta \boldsymbol{x}_0$ increases. This also indicates that this method is good only for problems with small variations.

If we plot the optimum f^* value obtained by each method vs. the increasing $\Delta \boldsymbol{x}_0$, we will obtain a graph as shown in Figure 5.9. We see once again in this figure that for small value of $\Delta \boldsymbol{x}_0$, the optimum f^* of all methods are relatively the same, but they diverge as $\Delta \boldsymbol{x}_0$ increases. The FWSR method with $\eta_{g,0} = 1.0$ is most conservative, but as shown in Figure 5.7 this method guarantees $P_s = 1.0$. FWSR method with $\eta_{g,0} = 0.8$ guarantees P_s of at least 0.999, and its f^* value is better than when $\eta_{g,0} = 1.0$. The f^* value of the gradient and moment matching methods are better than the FWCSR method, but this is misleading since the P_s of these two methods drops significantly as $\Delta \boldsymbol{x}_0$ increases.

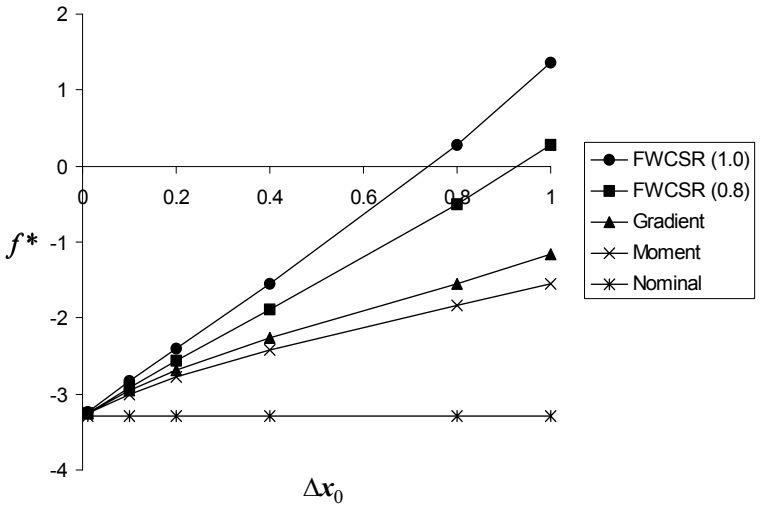


Figure 5.9: Plot of Δx_0 vs. f^* of the four robust optimization methods.

5.4.2. Design of an Explosive Actuated Cylinder

For our second example, we solve the optimization problem of designing an explosive actuated cylinder as formulated by Papalambros and Wilde (1980). In this problem, we are interested in optimizing the actuated cylinder shown in Figure 5.10 for minimum total length by controlling 5 design variables: the unswept cylinder length (x_1 - inch), the working stroke of the piston (x_2 – inch), the outside diameter of the cylinder (x_3 – inch), the initial pressure of the combustion (x_4 – ksi), and the piston diameter (x_5 – inch). All design variables are restricted to be positive.

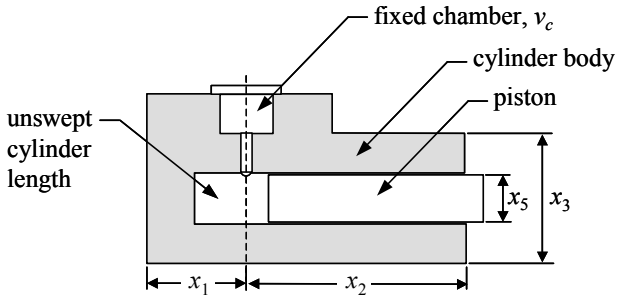


Figure 5.10: An explosive actuated cylinder.

In addition, there are constraints on the kinetic energy produced, the maximum piston force, and the stress on the cylinder wall, as well as geometric constraints. The mathematical formulation of the problem is as follows:

$$\begin{aligned}
&\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) = x_1 + x_2 \\
&\text{subject to}: \ \mathrm{g}_1(\boldsymbol{x}) \equiv \frac{(10^3)x_4 v_1^{\gamma}}{1-\gamma}(v_2^{1-\gamma} - v_1^{1-\gamma}) \geq \mathrm{W}_{\min} \\
&\qquad \mathrm{g}_2(\boldsymbol{x}) \equiv \left(\frac{10^3\pi}{4}\right) x_4 x_5^2 \leq \mathrm{F}_{\max} \\
&\qquad \mathrm{g}_3(\boldsymbol{x}) \equiv S_{yt} \leq (S_y/3) \\
&\qquad \mathrm{g}_4(\boldsymbol{x}) \equiv x_3 \leq \mathrm{D}_{\max} \\
&\qquad \mathrm{g}_5(\boldsymbol{x}) \equiv x_1 + x_2 \leq \mathrm{L}_{\max} \\
&\qquad \mathrm{g}_6(\boldsymbol{x}) \equiv x_5 + 0.001 \leq x_3
\end{aligned} \tag{5.19}$$

$$\text{where}: \ v_1 = v_c + (\pi/4)x_1 x_5^2 \quad ; \qquad v_2 = v_1 + (\pi/4)x_2 x_5^2 \tag{5.20}$$

$$S_{yt} = (\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2)^{1/2} \tag{5.21}$$

$$\sigma_1 = \frac{x_4(x_3^2 + x_5^2)}{(x_3^2 - x_5^2)} \quad ; \quad \sigma_2 = -x_4 \tag{5.22}$$

The constants used in the problem are as follows:

$\mathrm{D}_{\max}$ = maximum allowable cylinder outside diameter (1.0 in)

$\mathrm{F}_{\max}$ = maximum piston force (700 lb)

$\mathrm{L}_{\max}$ = maximum cylinder total length (2.0 in)

S_y = cylinder material yield stress (125 ksi)

v_c = fixed chamber volume (0.084 in^3)

$\mathrm{W}_{\min}$ = minimum kinetic energy for satisfactory performance (600 lb-in)

γ = specific heat ratio (1.2)

Solving Eq. (5.19) using MATLAB's *fmincon*, we obtain the nominal optimum to be $\boldsymbol{x}^*$ = [0, 1.042, 1.0, 23.12, 0.196] and f^* = 1.042. For this nominal optimum, the

constraints g_1, g_2, g_3, and g_4 are active. This optimum value is very close to that reported in Papalambros and Wilde (1980), $f^* = 1.036$, obtained using monotonicity analysis (actually, their optimum point is slightly infeasible. When $f^* = 1.036$, $[g_1, g_2, g_3] = [0.0026, 0.0017, 0.0049] > 0$).

There are variations in three of the design variables, and they are $[\Delta x_{3,0}, \Delta x_{4,0}, \Delta x_{5,0}] = [0.01, 1.0 , 0.01]$. We need to guarantee that the optimum design is feasibly robust with respect to these variations. Adding the robustness constraint $1 - \frac{\eta_g(\boldsymbol{x})}{\eta_{g,0}} \le 0$ to Eq. (5.19) with $\eta_{g,0} = 1.0$, and then solving it using *fmincon*, we obtain the robust optimum design to be $\boldsymbol{x}_R^* = [0.0097, 1.669, 0.823, 17.8, 0.201]$ and $f_R^* = 1.679$. For this robust optimum, all constraints are inactive. The inactivity of the constraints is one indication of the feasibility robustness of this optimum. If the constraints are active, then there is no 'cushion' for them to vary as $[x_3, x_4, x_5]$ vary. By moving slightly inside the feasible region, our robust optimum design provides some safety margin to absorb the $\Delta\boldsymbol{x}$. The inner optimization problem used to calculate the robustness of a design is also solved using *fmincon*, and on average converges in 15 iterations.

For comparison, we also solved the problem using a probabilistic method (Du and Chen, 2000). We replaced all of the constraints in the problem with a probabilistic one and constrained it to be greater than some predetermined value (= 0.99 in this case), $P[\mathbf{g}(\boldsymbol{x}) \le \mathbf{0}] \ge 0.99$. The probability is calculated using 100,000 runs of the Monte Carlo method assuming two different pdf models for the variations: uniform and normal. For the uniform model, the pdf is assumed to be uniformly distributed between $[-\Delta x_{i,0}, \Delta x_{i,0}]$. For the normal model, the pdf is assumed to be normally distributed with a mean and

standard deviation value of $[0, \Delta x_{i,0}/3]$. Table 5.2 shows the robust optimum designs obtained using the probabilistic method. For ease of comparison, we have again listed the values of the nominal optimum and our robust optimum with $\eta_{g,0} = 1.0$, and $\eta_{g,0} = 0.8$. Table 5.3 shows the constraint values of these optimum designs.

Table 5.2: Optimum designs of the explosive actuated cylinder.

	Nominal	Robust (Normal)	Robust (Uniform)	Robust ($\eta_g = 0.8$)	Robust ($\eta_g = 1.0$)
x_1	0	0.025	0.003	0	0.010
x_2	1.042	1.283	1.501	1.580	1.669
x_3	1.000	0.686	0.534	0.507	0.823
x_4	23.123	20.381	16.882	15.963	17.805
x_5	0.196	0.201	0.214	0.218	0.202
f	1.042	1.307	1.505	1.580	1.679

Table 5.3: Constraints of the optimum designs.

	Nominal	Robust (Normal)	Robust (Uniform)	Robust ($\eta_g = 0.8$)	Robust ($\eta_g = 1.0$)
g_1	0.0	-0.087	-0.114	-0.126	-0.167
g_2	0.0	-0.075	-0.136	-0.150	-0.188
g_3	0.0	-0.057	-0.147	-0.168	-0.199
g_4	0.0	-0.314	-0.466	-0.493	-0.177
g_5	-0.479	-0.346	-0.248	-0.210	-0.160
g_6	-0.803	-0.705	-0.598	-0.569	-0.754

We see in Table 5.2 that the nominal optimum has the lowest f value, while our robust optimum with $\eta_{g,0} = 1.0$ has the highest. The f value of the probabilistic optima and that obtained by our method using $\eta_{g,0} = 0.8$ are in between these two values. However, we observe in Table 5.3 that the constraints of the nominal optimum are closest to the feasible region boundary (i.e., its constraint values are largest on average) while those of our robust $\eta_{g,0} = 1.0$ optimum are the furthest. The constraints of the other

optima are somewhere in between. These constraint values indicate that our robust $\eta_{g,0} = 1.0$ optimum can absorb the most $\Delta \boldsymbol{x}$ variation, while the nominal optimum can absorb the least. Let us numerically verify the robustness of each of the optimum designs.

We perturbed the $[x_3, x_4, x_5]$ of each optima 20 times by adding some $\Delta \boldsymbol{x}$ value randomly sampled from the $\Delta \boldsymbol{x}_0$ range, and then calculate the new $\mathbf{g}(\boldsymbol{x})$ value of the design. A design is feasibly robust if all the new constraint values are still feasible, i.e., $\mathbf{g}(\boldsymbol{x}) \leq \mathbf{0}$. We performed the analysis only for g_1, g_2, g_3, and g_4 constraints because constraint g_5 is independent of $[x_3, x_4, x_5]$, and it is easily observed from Table 5.2 that constraint g_6 will always be satisfied for all optima.

Figures 5.11 – 5.13 show the graphs of the g_1, g_2, g_3, and g_4 constraints of each optimum under perturbation. For clarity, but without loss of information, the graphs only show the max[g_1, g_2, g_3, g_4] of each perturbation. In these graphs, a design is feasibly robust if all the bars are below the horizontal axis, i.e., $\max[g_j(\boldsymbol{x})] \leq 0$.

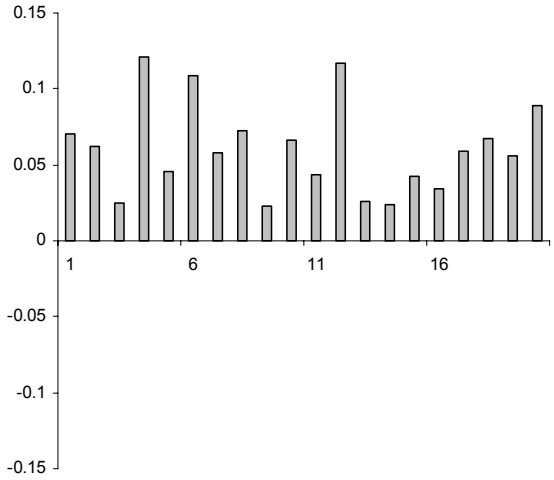


Figure 5.11: Sensitivity analysis of the nominal optimum.

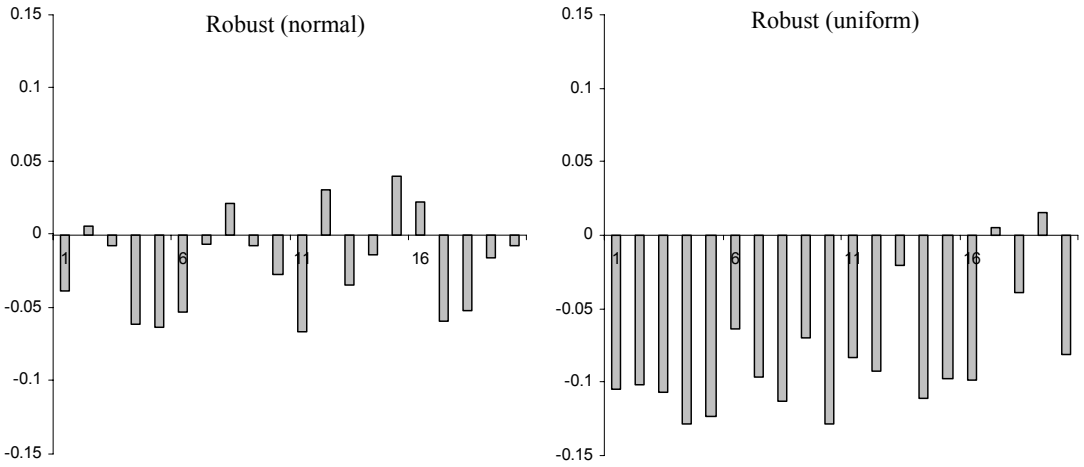


Figure 5.12: Sensitivity analysis of the probabilistic optima.

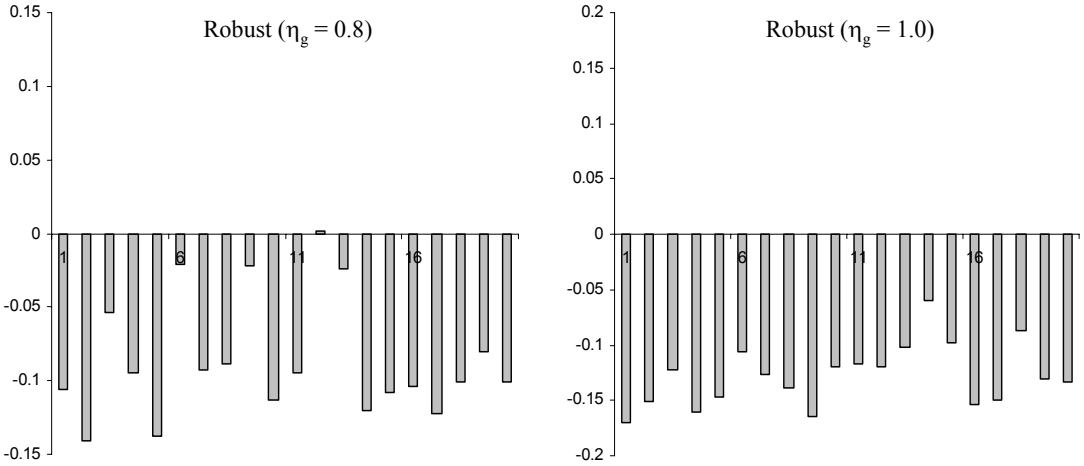


Figure 5.13: Sensitivity analysis of the robust optima.

We see in Figures 5.11 and 5.12 that the nominal optimum becomes infeasible in all 20 cases, while the probabilistic optima become infeasible in five cases for the robust (normal) optimum and in two cases for the robust (uniform) optimum. Our robust $\eta_{g,0} = 0.8$ optimum becomes infeasible in one case, while the robust $\eta_{g,0} = 1.0$ optimum is always feasible. These observations verify that our robust $\eta_{g,0} = 1.0$ optimum is the only

design that is feasibly robust. The robust (uniform) and robust $\eta_{g,0} = 0.8$ optima are close, but not quite feasibly robust.

To further verify the robustness of the optima, we calculate their probability of constraint satisfaction using 100,000 runs of Monte Carlo simulations. The simulations are performed using the uniform and normal $\Delta \boldsymbol{x}$ pdf models as before. The P_s value of each design is shown in Table 5.4.

Table 5.4: Probability of constraint satisfaction of the optima.

	P_s	
	Uniform pdf	Normal pdf
Nominal	0	0
Robust (Normal)	0.815	0.982
Robust (Uniform)	0.983	0.9997
Robust ($\eta_g = 0.8$)	0.996	0.9999
Robust ($\eta_g = 1.0$)	1.0	1.0

We see in Table 5.4 that the nominal optimum has a $P_s = 0$, which implies that it has no robustness towards variations at all. In contrast, the P_s of our robust $\eta_{g,0} = 1.0$ optimum is 1.0 confirming that this optimum design is indeed feasibly robust. The P_s's of the other optima confirm the robustness information shown in Figures 5.12 and 5.13 as well. If we compare Table 5.4 with Table 5.2, we also see that P_s increases as f increases (i.e., the performance vs. robustness trade-off).

One last item of interest is the computational efficiency of each method. For the nominal optimum, we did not perform any calculation of a design's robustness at all. For our FWCSR method, we solved an inner optimization using *fmincon* to calculate a design's robustness. On average the optimization converged in about 15 iterations. Because *fmincon* performed some function calls to estimate gradients, the actual number

of function calls used by our method is approximately close to 50. The probabilistic method calculated the design's robustness by performing 100,000 runs of Monte Carlo simulations. So we see that our robust optimization method is more efficient in terms of number of function calls.

5.4.3. Design of a Belleville Spring

Our third example is the problem of optimizing a Belleville spring originally formulated by Siddall (1982), and later modified by Hirokawa and Fujita (2002) as a robust optimization problem. The problem presented in this chapter is the one formulated by Hirokawa and Fujita (2002).

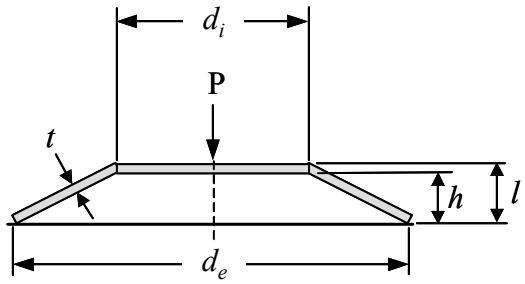


Figure 5.14: A Belleville spring.

In this problem, we want to optimize the Belleville spring (made out of steel) shown in Figure 5.14 for maximum rated load P. The design variables of the problem are: the external diameter (d_e), the internal diameter (d_i), the thickness (t), and the free height (h). The variables are all continuous, and their units are meter. The optimization is constrained by two design constraints, allowable stress and maximum mass, and five geometric constraints. The mathematical formulation of the problem is as follows:

$$
\begin{aligned}
&\underset{\boldsymbol{x}=[d_e,d_i,t,h]}{\text{maximize}} \quad f(\boldsymbol{x}) = \mathrm{P} \\
&\text{subject to}: \ \mathrm{g}_1(\boldsymbol{x}) \equiv \sigma_{\max} - \sigma_{aw} \le 0 \\
&\qquad\qquad \mathrm{g}_2(\boldsymbol{x}) \equiv m - m_{\max} \le 0 \\
&\qquad\qquad \mathrm{g}_3(\boldsymbol{x}) \equiv h_{\min} - h \le 0 \\
&\qquad\qquad \mathrm{g}_4(\boldsymbol{x}) \equiv (h+t) - l \le 0 \\
&\qquad\qquad \mathrm{g}_5(\boldsymbol{x}) \equiv d_e - d_{\max} \le 0 \\
&\qquad\qquad \mathrm{g}_6(\boldsymbol{x}) \equiv 1.25 d_i - d_e \le 0 \\
&\qquad\qquad \mathrm{g}_7(\boldsymbol{x}) \equiv h - 0.3(d_e - d_i) \le 0
\end{aligned}
\tag{5.23}
$$

Here P is the rated load (N), $\sigma_{\max}$ is the maximal stress (Pa), and m is the spring mass (kg). The quantities P, $\sigma_{\max}$, and m are calculated using the following equations.

$$
\mathrm{P} = \frac{E\,\delta_{\max}}{(1-\upsilon^2)\,\alpha\left(\frac{d_e}{2}\right)^2}\left[\left(h - \frac{\delta_{\max}}{2}\right)(h - \delta_{\max})t + t^3\right] \tag{5.24}
$$

$$
\sigma_{\max} = \frac{E\,\delta_{\max}}{(1-\upsilon^2)\,\alpha\left(\frac{d_e}{2}\right)^2}\left[\beta\left(h - \frac{\delta_{\max}}{2}\right) + \gamma t\right] \tag{5.25}
$$

$$
m = \frac{\pi}{4}\rho\,(d_e^2 - d_i^2)\,t \tag{5.26}
$$

$$
\text{where}: \ \alpha = \frac{6}{\pi \ln K}\left(\frac{K-1}{K}\right)^2 \quad ; \qquad \beta = \frac{6}{\pi \ln K}\left(\frac{K-1}{\ln K} - 1\right) \tag{5.27}
$$

$$
\gamma = \frac{6}{\pi \ln K}\left(\frac{K-1}{2}\right) \quad ; \qquad K = \frac{d_e}{d_i} \tag{5.28}
$$

The constants for this problem are:

$d_{\max}$ = maximum allowable diameter (0.3 m)

E = Young's modulus (210 GPa)

$h_{\min}$ = minimum height (0.005 m)

l = maximum allowable total height including t (0.02 m)

$m_{\max}$ = maximum spring mass (2.0 kg)

δ_{max} = maximum allowable deflection (= h)

ν = Poisson's ratio (0.3)

ρ = mass density (7850 kg/m^3)

σ_{aw} = allowable stress (1200 MPa)

The nominal optimum of the problem obtained using *fmincon* is $[d_e, d_i, t, h]^* = [0.3, 0.211, 7.273, 5.0]$ and the objective value is (in kN) $f^* = 42.106$. Here the values of the variables $[t, h]$ are in mm. The constraints of this optimum are $\mathbf{g}^* = [0, 0, 0, -0.386, 0, -0.112, -4.214]$. This nominal optimum f value is very close to that obtained by Hirokawa and Fuijta (2002), $f^* = 41.9$.

Due to manufacturing errors, all design variables $[d_e, d_i, t, h]$ are subject to variations. In addition, two of the material properties $[\sigma_{aw}, E]$ are also subject to variations. So there are six uncontrollable parameters in this problem $\mathbf{p} = [d_e, d_i, t, h, \sigma_{aw}, E]$. The variation ranges of the parameters are $\Delta\mathbf{p}_0 = [8.67 \times 10^{-5}, 7.67 \times 10^{-5}, 3.33 \times 10^{-5}, 3.33 \times 10^{-5}, 4.0 \times 10^{5}, 6.67 \times 10^{7}]$ (as given in Hirokawa and Fujita, 2002). The $\Delta\mathbf{p}_0$ values of the variables $[d_e, d_i, t, h]$ are in m. The $\Delta\mathbf{p}_0$ values of the material properties $[\sigma_{aw}, E]$ are in Pa.

To obtain a robust optimum, we added the robustness constraint to Eq. (5.23). The addition of this constraint, however, causes the *fmincon* algorithm to fail to obtain a solution. One reason of this failure is because the feasible region of the problem is already very small as it is. Adding the robustness constraint reduces the feasible region even further, making it very hard for an SQP algorithm to converge. Because *fmincon* failed to obtain a solution, we used a Genetic Algorithm (GA) (Goldberg, 1989) instead.

Due to implementation reasons, we used the same GA to solve the inner optimization problem. Solving the inner optimization using GA takes ~300 function calls.

The robust optimum obtained ($\eta_{g,0} = 1.0$) is shown in Table 5.5. For comparison purposes, we have also shown the robust optimum of the problem for $\eta_{g,0} = 0.8$, 0.6 and 0.3. Table 5.5 also shows the robust optimum reported by Hirokawa and Fujita (2002). Their robust optimum is obtained by a min-max strategy where they use the maximum of the constraints within a so-called "variation pattern" of the parameters as the constraints of the problem. The constraint values of these optima are shown in Table 5.6.

Table 5.5: Optimum designs of the Belleville spring.

	Nominal	Min-Max	Robust (0.3)	Robust (0.6)	Robust (0.8)	Robust (1.0)
d_e (m)	0	0.300	0.299	0.285	0.298	0.292
d_i (m)	0.213	0.212	0.207	0.187	0.208	0.196
t (mm)	7.273	7.083	6.994	6.938	6.738	6.780
h (mm)	5.0	5.1	5.219	5.063	5.187	5.152
f (kN)	42.106	37.190	37.662	35.677	34.253	33.488
η_g	0.075	0.032	0.248	0.601	0.824	1.430

Table 5.6: Constraint values of the Belleville spring optima.

	Nominal	Min-Max	Robust (0.3)	Robust (0.6)	Robust (0.8)	Robust (1.0)
g_1	0	0	-0.002	-0.017	-0.018	-0.036
g_2	0	0	-0.008	-0.006	-0.060	-0.018
g_3	0	-0.020	-0.044	-0.013	-0.038	-0.030
g_4	-0.386	-0.391	-0.389	-0.400	-0.404	-0.403
g_5	0	0	-0.003	-0.049	-0.006	-0.025
g_6	-0.112	-0.116	-0.132	-0.181	-0.124	-0.160
g_7	-4.214	-4.176	-4.256	-4.834	-4.163	-4.592

We see in Table 5.5 that the nominal optimum has a very low η_g, but has the highest f value (recall that this is a maximization problem). The value η_g progressively increases as f decreases (the performance vs. robustness trade-off). This progressive increase in robustness can also be observed in the decrease in the constraint values of the optima, Table 5.6 (i.e., the optimum is further away from the constraint boundary).

One important thing to notice in Table 5.5 is that the min-max optimum of Hirokawa and Fujita is inferior to the nominal and robust (0.3) optima (i.e., in terms of η_g and f, it is dominated). If we plot the value f vs. η_g, this inferiority is immediately apparent (Figure 5.15). The performance vs. robustness trade-off of the optima can also be observed in this figure.

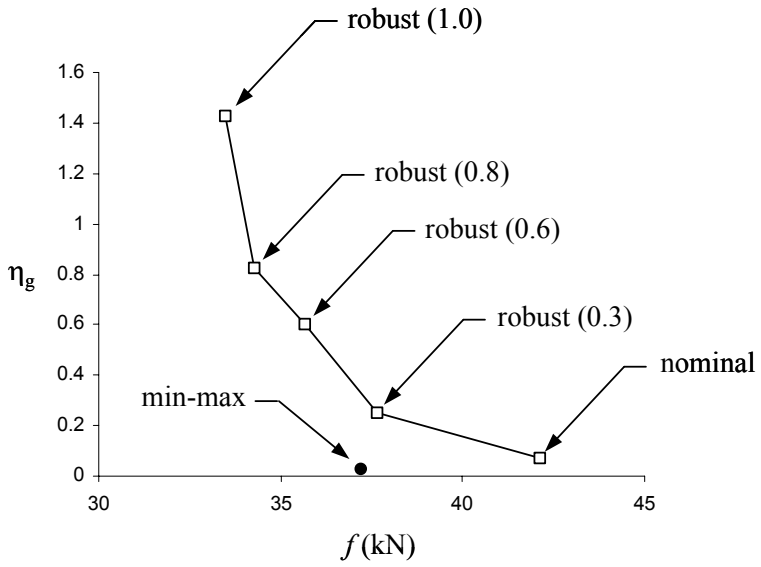


Figure 5.15: Plot of f vs. η_g of the Belleville spring.

To verify the robustness of the optima, we performed a numerical sensitivity analysis by perturbing the parameter values following the given ranges. A design is feasibly robust if all the constraints remain feasible. Figures 5.16, 5.17, and 5.18 show the graphs

of the max[**g**(***x***,**p**)] of each optimum. In these graphs, a design is feasibly robust if all the bars are below the horizontal axis.

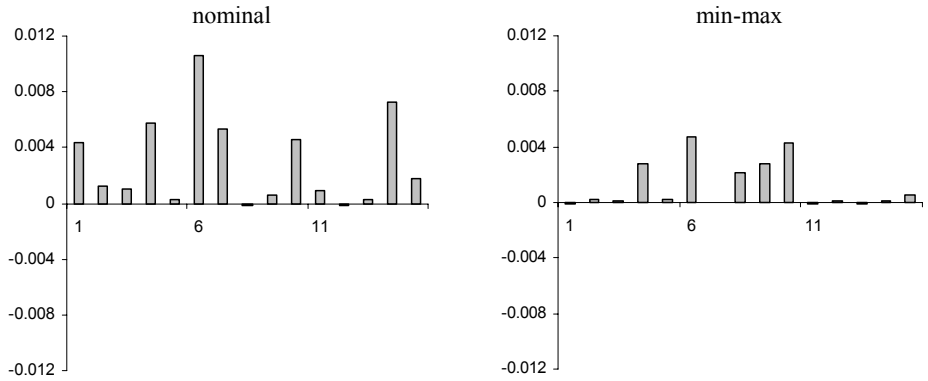


Figure 5.16: Sensitivity analysis of the nominal and min-max optima.

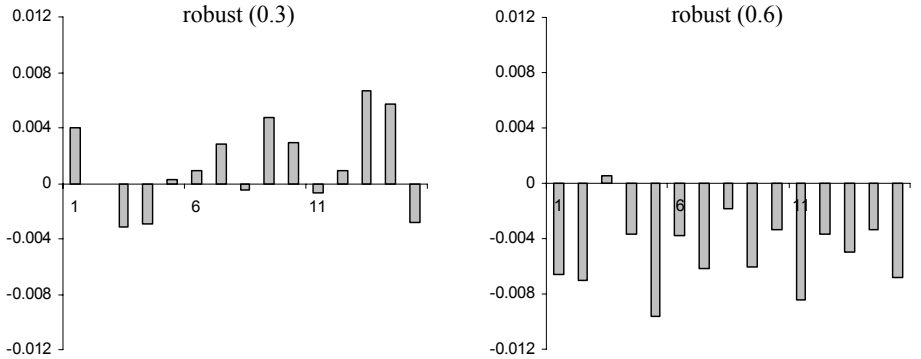


Figure 5.17: Sensitivity analysis of the robust (0.3) and robust (0.6) optima.

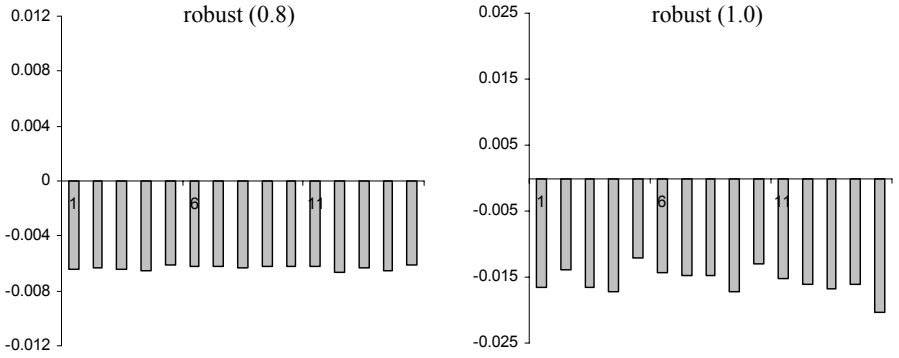


Figure 5.18: Sensitivity analysis of the robust (0.8) and robust (1.0) optima.

We see in Figures 5.16 – 5.18 that the nominal, min-max, and robust (0.3) optima are not feasibly robust. The robust (0.6) optimum is almost feasibly robust except in one case where it becomes infeasible. The robust (0.8) and robust (1.0) optima are always feasible regardless of the perturbations, thus they are feasibly robust. These observations confirm the η_g values of each design shown in Table 5.5.

5.4.4. Design of a Control Valve Actuator Linkage

For our last example, we applied our feasibility robust optimization method to the engineering design of a control valve actuator linkage. This example is adapted from the example in Balling et al. (1986) with some modifications.

The control valve actuator linkage to be designed is shown in Figure 5.19.

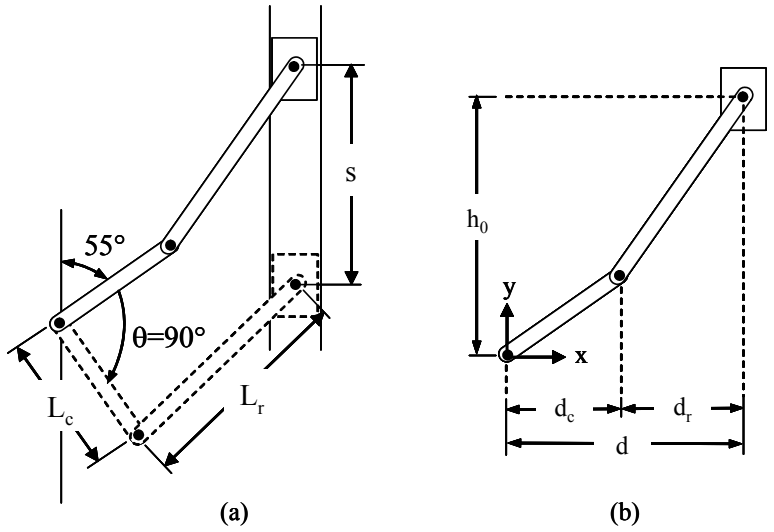


Figure 5.19: A control valve actuator linkage.

The linkage mechanism has two members: a crank and a rod that are connected by a pin joint. The other end of the crank is held stationary, while the other end of the rod is pinned to a slider. Originally the crank was at a 55° angle from the vertical axis as shown in Figure 5.19(a). The dimensions of the linkage are shown in Figure 5.19(b).

There is a constant force F = 1425.5 lbs (6340 N) acting at the end of the rod that causes the mechanism to turn (see Figure 5.20). The design objective is to maximize the torque (T) at the end of the crank as it turns from $\theta = 0°$ to $\theta = 90°$, averaged over 10° interval (Figure 5.20). The weights of the crank and rod are assumed negligible.

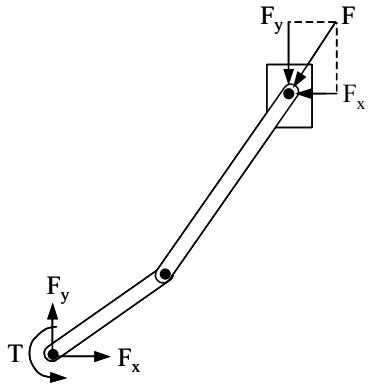


Figure 5.20: Forces acting on the linkage.

The design variables in this problem are the crank length (L_c), the rod length (L_r), and the center distance (d). The crank length and the rod length are constrained to be within 0 and 10 inch (0 and 25.4 cm), while the center distance is constrained to be within 5 and 7 inch (12.7 and 17.78 cm). The problem has three design constraints: (1) the vertical position of the slider when $\theta = 0°$ (h_0) is constrained to be less than 6.5 inch (16.51 cm), (2) the movement of the slider (s) is constrained to be less than 4.5 inch (11.43 cm), and (3) the side force (F_x) averaged over 10° interval is less than 800 lbs (3558 N). In addition, the problem has two geometric constraints: (1) the horizontal length of the crank (d_c) must not be greater than the center distance (d), and (2) the difference between d and d_c must be less than or equal to the rod length (L_r). The mathematical formulation of the problem is shown below (for English units).

$$
\begin{aligned}
&\text{maximize} \quad f(L_c, L_r, d) = \frac{1}{9}\sum_{\theta=0^\circ}^{90^\circ} T(\theta) \\
&\text{subject to}: \; g_1 \equiv \frac{h_0}{6.5} - 1 \le 0 \qquad ; \qquad g_2 \equiv \frac{s}{4.5} - 1 \le 0 \\
&\qquad\qquad g_3 \equiv \frac{\frac{1}{9}\sum_{\theta=0^\circ}^{90^\circ} F_x(\theta)}{800} - 1 \le 0 \quad ; \quad g_4 \equiv \frac{d_c}{d} - 1 \le 0 \\
&\qquad\qquad g_5 \equiv \frac{d_r}{L_r} - 1 \le 0 \\
&\qquad\qquad 0 \le L_c, L_r \le 10 \quad ; \quad 5 \le d \le 7
\end{aligned}
\tag{5.29}
$$

The quantities $T(\theta)$, $F_x(\theta)$, s, h_0, d_c, and d_r are calculated as follows:

$$
d_c = L_c \cdot \cos(\theta - 45) \quad ; \qquad d_r = d - d_c \tag{5.30}
$$

$$
\begin{aligned}
&h_0 = h(\theta = 0^\circ) \\
&\quad h(\theta) = L_r \cdot \sin\beta - L_c \cdot \sin(\theta - 45) \\
&\quad \beta = \cos^{-1}\left(\frac{d_r}{L_r}\right) \quad \text{(in radians)}
\end{aligned}
\tag{5.31}
$$

$$
s = h(\theta = 90^\circ) - h(\theta = 0^\circ) \tag{5.32}
$$

$$
F_x(\theta) = F \cdot \cos\beta \tag{5.33}
$$

$$
\begin{aligned}
&T(\theta) = d.Fy - h(\theta) \cdot F_x(\theta) \\
&\quad F_y = F \cdot \sin\beta
\end{aligned}
\tag{5.34}
$$

Because of manufacturing tolerances, L_c and L_r vary by ±0.1 inch (±0.254 cm), and we need to guarantee the feasibility of the optimum design under these variations. We add the sensitivity constraint $1 - \frac{\eta_g(\boldsymbol{x})}{\eta_{g,0}} \le 0$ to Eq. (5.29) and then optimize it. In this problem $\Delta\mathbf{p}_0 = [0.1, 0.1]$ and $\eta_{g,0} = 1.0$. For comparison, we also optimize the original problem (nominal optimum), and Eq. (5.29) with its constraints replaced by a probabilistic constraint $P(g_j \le 0) \ge 0.99$, j=1,…,5. The probabilistic constraint is

calculated using Monte Carlo simulation assuming two probability distribution models: uniform and normal. For the uniform model, the lower and upper bound of the distribution are specified to be –0.1 and 0.1, respectively. For the normal model, the distribution is specified to have a mean of 0 and standard deviation of 0.033 (= 0.1/3).

The results obtained from this comparison study are shown in Table 5.7. In this table the quantity F_{call} is the number of function evaluations needed per design to calculate its feasibility robustness. Table 5.7 also shows the η_g value of each optimum design.

Table 5.7: Optimum designs of the control valve actuator linkage.

	Nominal	Robust	Monte Carlo (Uniform)	Monte Carlo (Normal)
Torque (in.lb)	3592	3363	3429	3453
L_c (in)	3.182	3.03	3.078	3.093
L_r (in)	5.061	5.009	4.994	4.999
d (in)	5.0	5.0	5.0	5.0
F_{call}	N/A	250	100000	100000
η_g	0.005	1.02	0.73	0.62

We observe from Table 1 that the nominal optimum has the highest torque but the lowest η_g. In contrast, the robust optimum obtained by our method has the lowest torque but the highest η_g. The Monte Carlo optima are somewhere in between the two. This observation is expected because generally we have to sacrifice some performance to gain an increase in robustness. In fact, if we solve the optimization problem as a two-objective problem where we maximize both torque and η_g, we will obtain a trade-off frontier as shown in Figure 5.21. Notice in this figure how torque decreases as η_g increases. Points corresponding to the optima in Table 5.7 are also shown.

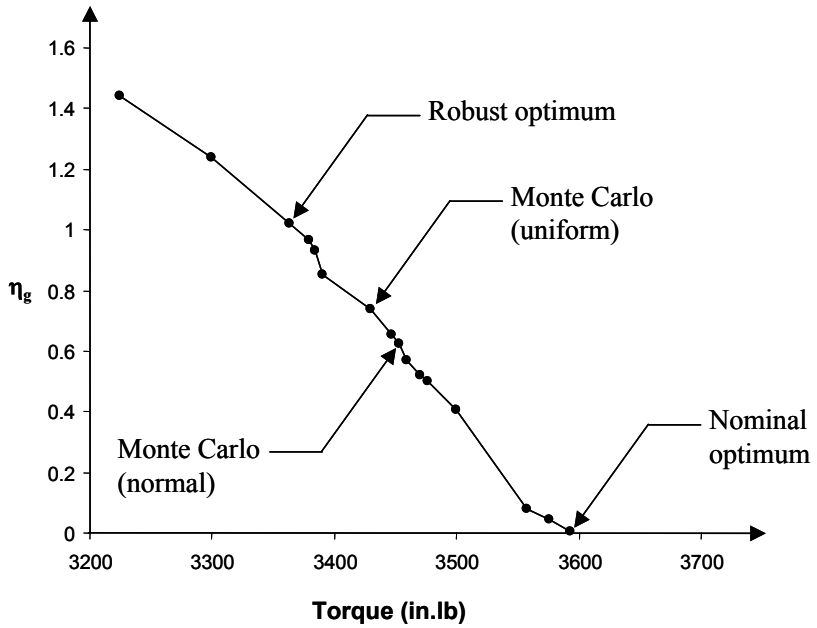


Figure 5.21: Trade-off frontier of the linkage problem.

Note also in Table 5.7 that although the two Monte Carlo optima were obtained by enforcing a 0.99 probability of constraints satisfaction, their η_g value is less than 1.0. In fact, the η_g values of the two optima are quite different. This is because the robustness of the optimum design is sensitive to the assumed probability distributions, and the 0.99 constraints satisfaction probability is valid only if the assumed distribution is valid. We will further discuss this important issue next.

Table 5.7 (and Figure 5.21) also shows that the robust optimum is different than the Monte Carlo optimum using a uniform distribution. This shows that our method does not presume a uniform probability distribution of the parameters (although it may seem so). The fact is our method does not presume any distribution at all, and this is reflected in the fact that we cannot provide probability information for the obtained optimum. We lack the information to do so. Our robustness constraint guarantees that the optimum design will remain feasible if the parameter variations are as specified. If the parameter

distribution changes, the probability of constraint satisfaction will change, but this guarantee still holds.

We also observe in Table 5.7 that Monte Carlo method requires 100,000 function evaluations to calculate a design's robustness. In contrast, our method requires only ~250 evaluations, comparable to those more efficient probabilistic methods (MPP for instance – Du and Chen, 2000). It should be noted, however, that this number (i.e., 250) is an upper bound value because we used GA to solve the inner optimization problem. GA is an optimizer that needs a lot of function evaluation, but it is applicable to a wide range of optimization problem and does not require gradient information. If gradient information is available, we can use a more efficient optimizer to solve our inner optimization problem, and the number of function evaluations our method needs would be much lower (in the order of 10^1).

To validate the results in Table 5.7, we conducted a sensitivity analysis on each optimum design. We performed 100,000 Monte Carlo simulations on each design, and based on the result, calculate its probability of constraint satisfaction. The simulations are performed using two probability distribution models of the parameters: uniform and normal. In the uniform distribution model, ΔL_c and ΔL_r are jointly uniformly distributed in the interval [-0.1,0.1]. In the normal distribution model, ΔL_c and ΔL_r are bi-normally distributed with a mean and standard deviation of [0, 0.033]. In both models, ΔL_c and ΔL_r are assumed independent. The results of this sensitivity study are shown in Table 5.8. For comparison purposes, we have also re-listed the η_g value of each optimum.

Table 5.8: Sensitivity analysis of the optima.

	Probability of constraint satisfaction		η_g
	Uniform model	Normal model	
Nominal	0.376	0.381	0.005
Robust	1.0	1.0	1.02
Monte Carlo (Uniform)	0.989	0.999	0.73
Monte Carlo (Normal)	0.925	0.996	0.62

We observe in Table 5.8 that the nominal optimum has a poor probability of constraint satisfaction in both models. This is not surprising since this optimum is obtained by strict optimization of the linkage's torque, neglecting the variations in L_c and L_r. The robust optimum obtained by our method, on the other hand, has a 1.0 probability of constraint satisfaction in both models, much more robust than the nominal optimum. This observation confirms the information provided by the η_g values of the two optima. The robust optimum has a much larger η_g value (1.02) than the nominal optimum (0.005).

The Monte Carlo (uniform) optimum has a 0.989 probability of constraint satisfaction for the uniform distribution model (same value as imposed by the probabilistic constraint). But this probability value increases to 0.999 for the normal model. The same pattern is also observed for the Monte Carlo (normal) optimum. It has a 0.996 probability of constraint satisfaction for the normal model (same as imposed by the constraint), but this value reduces to 0.925 for the uniform model. This observation shows that the information provided by the probabilistic constraints is dependent on the accuracy of the assumed distribution model. As such, unless the distribution of the uncertain parameters is known with relative certainty, such probability information must be used with utmost caution since it can be misleading.

To further validate our method, we calculated and compared the FSR and FWCSR of the optima in Table 5.7. For simplicity, we only show the comparison for the nominal and robust optima. We form the FSR ($\mathbf{S}_g$) of each design by first forming the FSR of each constraint ($\mathbf{S}_{g,j}$) and then forming the intersection. $\mathbf{S}_{g,j}$ is obtained by constructing the difference function $\Delta g_j(\Delta L_c, \Delta L_r) = g_j(L_c+\Delta L_c,\ L_r+\Delta L_r) - g_j(L_c, L_r)$ from Eq. (5.29), substituting the L_c, L_r, and d values of the design into this function, and then setting it to $\Delta g_j \le \Delta g_{j,0}$. The $\Delta g_{j,0}$ is obtained by taking the absolute value of the *j*-th constraint of the design. The constraint values of the nominal and robust optima are shown in Table 5.9.

Table 5.9: Constraint values of the nominal and robust optima.

	Nominal	Robust
g_1	-0.00026	-0.037
g_2	-0.000031	-0.048
g_3	-0.173	-0.169
g_4	-0.366	-0.396
g_5	-0.457	-0.429

Observe in Table 5.9 that only g_1 and g_2 are active (or nearly active) for the nominal optimum while g_3, g_4, and g_5 are relatively the same for the two optima. This implies that the Δg_1 and Δg_2 functions are critical components of $\mathbf{S}_f$, while Δg_3, Δg_4, and Δg_5 are not. So, in constructing the FWCSR, using only Δg_1 and Δg_2 suffices. From Eq. (5.29), we obtain the Δg_1 and Δg_2 difference functions:

$$\begin{aligned}
\Delta g_1 &= \frac{1}{6.5}\left[(L_r+\Delta L_r)\sin\hat{\beta} - L_r\sin\beta + 0.707\,\Delta L_c\right] \\
\Delta g_2 &= -0.314\,\Delta L_c \\
\hat{\beta} &= \cos^{-1}\left(\frac{\hat{d}_r}{L_r+\Delta L_r}\right) \quad \text{(in radians)} \\
\hat{d}_r &= d - 0.707\,(L_c+\Delta L_c)
\end{aligned} \tag{5.35}$$

Figure 5.22 shows the FSR and FWCSR of the nominal and robust optima. In this figure only the critical components, i.e., Δg_1 and Δg_2, are shown. Note that $\mathbf{S}_{g,1}$ is not a linear function (although it may seem so from the figure). $\mathbf{S}_{g,2}$ is a linear function, while $\mathbf{S}_{g,1}$ is a trigonometric function as shown in Eq. (5.35). Note also that Figure 5.22 shows the regions in the non-normalized space (but has same scale). We do so to better relate to the actual values of L_c and L_r.

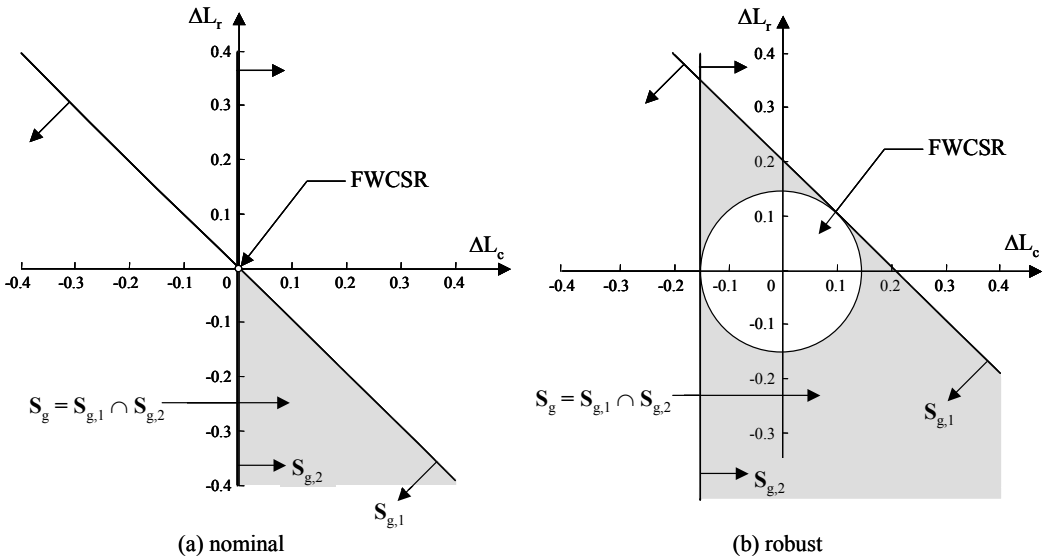


Figure 5.22: FSR and FWCSR of nominal and robust optima.

We observe in Figure 5.22 that the FWCSR of the nominal optimum is very small, close to a zero radius. This is because g_1 and g_2 are almost active for this optimum, and as such there is very little "cushion" for L_c and L_r variation (in the worst case sense). In contrast, the FWCSR of the robust optimum is much larger ($R_g = 1.44$), and it allows for more variations in L_c and L_r. This is also reflected by the larger g_1 and g_2 values of the robust optimum in Table 5.9. Observe also that the FWCSR of the robust optimum covers the area bounded by [-0.1, 0.1] as specified.

5.5. SUMMARY

- The feasibility robustness of a design is indicated by its feasibility sensitivity set ($\mathbf{S}_g$). Like the objective sensitivity set ($\mathbf{S}_f$), the feasibility sensitivity set $\mathbf{S}_g$ shows how much $\Delta\mathbf{p}$ a design can absorb before it violates a prescribed limit. Thus, the larger $\mathbf{S}_g$, the more robust the design.
- However, unlike $\mathbf{S}_f$, $\mathbf{S}_g$ is a one-sided sensitivity measure because we only need to limit the increase, but not the decrease, in the constraints.
- The plot of $\mathbf{S}_g$ in the $\Delta\mathbf{p}$-space is the Feasibility Sensitivity Region (FSR) of a design. To account for directional sensitivity, we use the worst-case estimate of FSR, the Feasibility Worst Case Sensitivity Region (FWCSR) as a robustness measure. The FWCSR radius (R_g) can be calculated by solving a single-objective optimization with one equality constraint.
- Unlike objective robustness, the increment limits for constraints are determined by how far a design is from the constraint boundary, and not by the designer.
- In addition, a design has to be feasible nominally for the FWCSR measure to make sense. If a design is infeasible, then feasibility robustness cannot be guaranteed.
- The inner optimization to calculate R_g must be normalized if the scale of $\Delta\mathbf{p}$ is different. After the normalization, a design is guaranteed to be feasibly robust if its feasibility robustness index, $\eta_g = (G)^{-1/2}R_g$, is greater than or equal to 1.0.

CHAPTER 6

DISCUSSIONS

6.1. INTRODUCTION

In the last three chapters, we presented methods for objective robust optimization and feasibility robust optimization separately. The purpose of this chapter is to show how to combine these methods for both objective and feasibility robust optimization. In addition, this chapter also aims to address those issues in our robust optimization methods that we have not yet addressed, or so far only briefly discussed. More specifically, this chapter will discuss issues regarding: (i) use of one-sided sensitivity measure for objective robustness, (ii) asymmetrical two-sided sensitivity measure, (iii) asymmetrical parameter variations, and (iv) a comparison between robustness index and robustness probability.

6.2. OBJECTIVE AND FEASIBILITY ROBUST OPTIMIZATION

In Chapters 3 and 4, we introduced an index η to measure objective robustness of a design, where $\eta = (G)^{-1/2} R_f$, and R_f calculated by solving an optimization problem in Eq. (4.6), restated here for convenience:

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \;\; R_f(\Delta\overline{\mathbf{p}}) = \left[\sum_{j=1}^{G} (\Delta\overline{p}_j)^2 \right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{i=1,\ldots,M} \left(\frac{\left| \Delta f_i(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0) \right|}{\Delta f_{i,0}} \right) - 1 = 0 \end{aligned} \tag{6.1}$$

Similarly, in Chapter 5 we introduced an index η_g to measure feasibility robustness of a design, where $\eta_g = (G)^{-1/2} R_g$, and R_g calculated by Eq. (5.16), restated below.

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \quad \mathrm{R_g}(\Delta\overline{\mathbf{p}}) = \left[\sum_{i=1}^{\mathrm{G}} \Delta\overline{\mathrm{p}}_i^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{j=1,\ldots,\mathrm{J}}\left[\mathrm{g}_j(\boldsymbol{x}_0, \mathbf{p} + \Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)\right] = 0 \end{aligned} \tag{6.2}$$

Adding the constraint $1 - \frac{\eta}{\eta_0} \le 0$ to an optimization problem guarantees objective robustness of an optimum design, while adding the constraint $1 - \frac{\eta_\mathrm{g}}{\eta_{\mathrm{g},0}} \le 0$ guarantees its feasibility robustness. Accordingly, adding both of these robustness constraints to an optimization problem will guarantee both objective and feasibility robustness of an optimum design. Eq. (6.3) shows the overall formulation of a general robust optimization problem, where η and η_g are calculated by Eq. (6.1) and (6.2), respectively.

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \left[f_1(\boldsymbol{x}, \mathbf{p}_0), \ldots, f_\mathrm{M}(\boldsymbol{x}, \mathbf{p}_0)\right] \\ &\text{subject to}: \quad \mathrm{g}_j(\boldsymbol{x}, \mathbf{p}_0) \le 0 \quad ; \quad j = 1, \ldots, \mathrm{J} \\ &\qquad\qquad \mathrm{h}_k(\boldsymbol{x}, \mathbf{p}_0) = 0 \quad ; \quad k = 1, \ldots, \mathrm{K} \\ &\qquad\qquad 1 - \frac{\eta}{\eta_0} \le 0 \qquad \text{(objective robustness constraint)} \\ &\qquad\qquad 1 - \frac{\eta_\mathrm{g}}{\eta_{\mathrm{g},0}} \le 0 \qquad \text{(feasibility robustness constraint)} \end{aligned} \tag{6.3}$$

Notice in Eq. (6.1) and (6.2) that R_f and $\mathrm{R_g}$ are defined in the same space ($\Delta\overline{\mathbf{p}}$-space), and are of the same scale (normalized by $\Delta\mathbf{p}_0$). Based on this observation, then the two robustness constraints in Eq. (6.3) will be satisfied if the larger of the two constraints is satisfied, i.e., if $\max\left[1 - \frac{\eta}{\eta_0}, 1 - \frac{\eta_\mathrm{g}}{\eta_{\mathrm{g},0}}\right] \le 0$. Using this fact, the robustness constraints in Eq. (6.3) can be stated more compactly as: $1 - \min\left[\frac{\eta}{\eta_0}, \frac{\eta_\mathrm{g}}{\eta_{\mathrm{g},0}}\right] \le 0$.

The advantage of keeping the objective and feasibility robustness constraints separate like shown in Eq. (6.3) (they are still separate in the compact form as well) is that it provides flexibility for a designer to specify his/her preference towards the two types of robustness. By setting different values for η_0 and $\eta_{g,0}$, a designer can specify that (s)he considers one type of robustness (i.e., objective robustness or feasibility robustness) more important than the other type. For example, if a designer specifies $\eta_0 = 0.8$ but $\eta_{g,0} = 1.0$, then it implies that the feasibility robustness of an optimum design is considered to be more important than its objective robustness.

If a designer is indifferent towards either the objective or the feasibility robustness (i.e., $\eta_0 = \eta_{g,0}$), then the robustness constraint $1 - \min\left[\frac{\eta}{\eta_0}, \frac{\eta_g}{\eta_{g,0}}\right] \le 0$ simplifies into $1 - \left(\frac{G^{-1/2}}{\eta_0}\right)\min\left[R_f, R_g\right] \le 0$. This last inequality is of particular interest to us. In this inequality, we only need to determine the smaller of R_f and R_g, but not both. Let's discuss the meaning of $\min[R_f, R_g]$ in more detail next.

Recall from previous chapters that conceptually R_f is the normalized WCSR radius of a design, and WCSR is the worst-case estimate of the corresponding SR. Likewise, R_g is the normalized FWCSR radius of the design, and FWCSR is the worst-case estimate of the FSR. Since SR and FSR are defined in the same $\Delta\overline{\mathbf{p}}$-space and are of the same scale, $\min[R_f, R_g]$ implies that we are looking for the radius of worst-case estimate of an intersection of SR and FSR, as shown in Figure 6.1. In Figure 6.1, the region between the two solid lines is the SR, and the region bounded by the dashed line is the FSR. The R_f

and R_g of the SR and FSR are also shown in Figure 6.1. The shaded region is the intersection of SR and FSR.

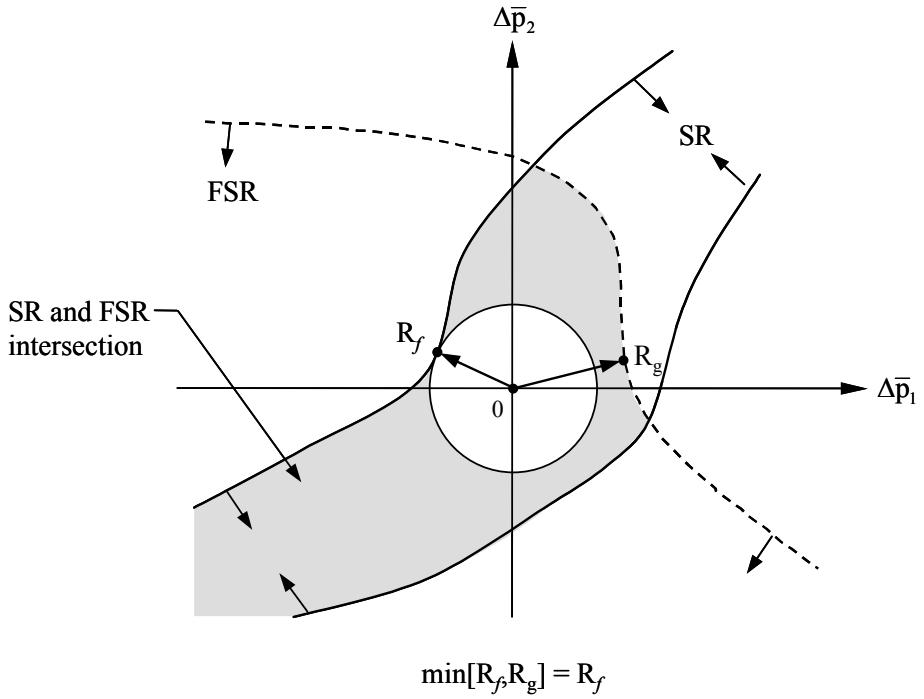


Figure 6.1: Intersection of SR and FSR.

Mathematically, a $\Delta\mathbf{p}$ point is inside the intersection region if it satisfies all of the following inequalities: $\begin{cases}[\Delta f_i]^2 < [\Delta f_{i,0}]^2 & i = 1,...,\mathrm{M} \\ \Delta \mathrm{g}_j < \Delta \mathrm{g}_{j,0} & j = 1,...,\mathrm{J}\end{cases}$. A $\Delta\mathbf{p}$ point is outside the intersection region if either $[\Delta f_i]^2 > [\Delta f_{i,0}]^2$ or $\Delta \mathrm{g}_j > \Delta \mathrm{g}_{j,0}$ is true for at least one $i = 1,\ldots,\mathrm{M}$ or $j = 1,\ldots,\mathrm{J}$, respectively. For a $\Delta\mathbf{p}$ point to be on the boundary of the intersection region, it needs to satisfy $[\Delta f_i]^2 \le [\Delta f_{i,0}]^2$ and $\Delta \mathrm{g}_j \le \Delta \mathrm{g}_{j,0}$, with at least one strict equality (one i or j from $i = 1,\ldots,\mathrm{M}$ or $j = 1,\ldots,\mathrm{J}$, respectively). Using the simplified condition for SR and FSR boundaries developed previously (recall Section 4.3.2 and 5.2.2), the condition for a $\Delta\mathbf{p}$ point to be on the boundary of the intersection region can be

simplified into: $\max\left\{\max_{i=1,\ldots,M}\left(\frac{|\Delta f_i|}{\Delta f_{i,0}}\right), \max_{j=1,\ldots,J}\left(\frac{\Delta g_j}{|g_j(\boldsymbol{x}_0,\mathbf{p})|}\right)\right\}-1=0$, where we have used the fact that $\Delta g_{j,0} = |g_j(\boldsymbol{x}_0,\mathbf{p})|$. Notice also that in this simplified condition we have used $\frac{\Delta g_j}{|g_j(\boldsymbol{x}_0,\mathbf{p})|}$ instead of $g_j(\boldsymbol{x}_0,\mathbf{p})$ to define the FSR boundary (recall Section 5.3). Using the ratio $\frac{\Delta g_j}{|g_j(\boldsymbol{x}_0,\mathbf{p})|}$ is recommended because we are comparing it to a normalized Δf_i value. Using $g_j(\boldsymbol{x}_0,\mathbf{p})$ in the formulation may create difficulty in terms of numerical comparison.

Using the above mathematical definitions, the radius (R) of the worst-case estimate of the intersection region can be calculated by solving the following optimization problem (normalized):

$$\begin{aligned}
&\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \quad R(\Delta\overline{\mathbf{p}}) = \left[\sum_{i=1}^{G}\Delta\overline{p}_i^2\right]^{\frac{1}{2}} \\
&\text{subject to:} \\
&\max\left\{\max_{i=1,\ldots,M}\left[\frac{|\Delta f_i(\mathbf{p}+\Delta\overline{\mathbf{p}}\otimes\Delta\mathbf{p}_0)|}{\Delta f_{i,0}}\right], \max_{j=1,\ldots,J}\left[\frac{\Delta g_j(\mathbf{p}+\Delta\overline{\mathbf{p}}\otimes\Delta\mathbf{p}_0)}{|g_j(\boldsymbol{x}_0,\mathbf{p})|}\right]\right\}-1=0
\end{aligned} \tag{6.4}$$

Recall from Figure 6.1 that this radius value (R) is equal to $\min[R_f,R_g]$. So, Eq. (6.4) shows that we can find $\min[R_f,R_g]$ by solving just one inner optimization problem instead of two (i.e., Eq. (6.1) and (6.2)).

If we define a quantity $\eta_{min} = (G)^{-1/2}\min[R_f,R_g]$, then the overall objective and feasibility robust optimization problem becomes as shown in Eq. (6.5) in which $\min[R_f,R_g]$ is calculated by solving Eq. (6.4). Here, we call η_{min} the overall robustness index. We have used η_0 for the robustness constraint in Eq. (6.5), but since $\eta_0 = \eta_{g,0}$, we could have used $\eta_{g,0}$ as well.

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \left[f_1(\boldsymbol{x},\mathbf{p}_0),...,f_M(\boldsymbol{x},\mathbf{p}_0)\right] \\ \text{subject to}: \quad & g_j(\boldsymbol{x},\mathbf{p}_0) \le 0 \quad ; \quad j = 1,...,J \\ & 1 - \frac{\eta_{min}}{\eta_0} \le 0 \qquad \text{(overall robustness constraint)} \end{aligned} \tag{6.5}$$

One advantage of having to solve just one inner optimization problem is that practically the optimization algorithm to solve Eq. (6.5) (the outer and inner problems and the interface between them) will be much easier to implement. In addition, it helps the outer problem to converge faster because it has one fewer constraint to satisfy. Solving only one inner problem also helps reduce numerical errors transmitted from the inner problem to the outer problem. One potential disadvantage of combining Eq. (6.1) and (6.2) into a single inner problem is that solving Eq. (6.4) might be less efficient computationally than solving Eq. (6.1) and (6.2) separately. For instance, if Eq. (6.1) and Eq. (6.2) can be solved in say T iterations each, it is possible that Eq. (6.4) may need more than T iterations to converge because its constraint is more complex.

Before we continue further, it is important to point out again that Eq. (6.4) and (6.5) are valid only if the designer is indifferent to either objective or feasibility robustness (i.e., $\eta_0 = \eta_{g,0}$). If one type of robustness is preferred to the other type, then we must solve Eq. (6.1) and (6.2) separately.

6.2.1. Design of a Payload for an Undersea Autonomous Vehicle (UAV)

To demonstrate our combined objective and feasibility robust optimization method, we apply it to an engineering example: the design of a UAV payload. The description of the problem is as follows.

Typically, the payload of a UAV must be effective in several different uses, called "scenarios." Effectiveness in a scenario is measured by the probability of success, P_S, of payload delivery in that scenario. The design goal is to simultaneously maximize the individual P_S's for all scenarios. The payload design is constrained by upper limits on the weight of the payload and on the radiated noise generated by the payload.

There are six design variables: the payload length (PL), the hull diameter (DH), the material of the hull (HM), the payload type (PT), the first inner material type (I1), and the second inner material type (I2). Four of the variables are discrete: HM, PT, I1, and I2. The choices for HM, PT and I1 are [6061AL, 7075AL], [BULK, MULTI_MISS], and [TYPE_1A, TYPE_1B], respectively. For discrete variable I2, the options available are [TYPE_2A, TYPE_2B, TYPE_1B], but I2 can be TYPE_1B only if the variable I1 is TYPE_1B also. The other two variables are continuous and they are bounded as: $6.0 \leq DH \leq 12.75$ and $1.0(DH) \leq PL \leq 5.0(DH)$. In addition to the six design variables, there is a fixed continuous design parameter, the maximum depth (= 3000 ft), at which the payload operates. Unlike our other design examples, there are no closed-form relationships to map the design variables to the constraints and to the P_S's. Rather, we are provided with a design analyzer (a computer program) that maps the design variables to the payload weight, the radiated noise, and the P_S's for the scenarios.

In this example, we address a two objective payload design optimization with two constraints. The two objectives are to maximize P_{S1} and P_{S2} for two different scenarios. The two constraints are an 85 lb upper bound on the payload weight and a 0.16 Watt/m^2 upper bound on the radiated noise generated. The problem is mathematically formulated as follows.

$$
\begin{aligned}
&\text{maximize} \quad P_{S1}(\text{PL, DH, HM, PT, I1, I2}) \\
&\text{maximize} \quad P_{S2}(\text{PL, DH, HM, PT, I1, I2}) \\
&\text{subject to: } \text{Weight}(\text{PL, DH, HM, PT, I1, I2}) - 85 \le 0 \\
&\qquad\qquad \text{Noise}(\text{PL, DH, HM, PT, I1, I2}) - 0.16 \le 0
\end{aligned}
\tag{6.6}
$$

There are some uncertainties in the formulation of the problem, and it is modeled by assuming that a parameter (internal to the design analyzer), C_{eq}, which is used in calculating the weight of the payload has an uncontrollable variation. Since the value of C_{eq} depends on the discrete combination of [PT, I1, I2], its variation is taken to be a percentage of the actual value: $\Delta C_{eq,0} = (0.10)C_{eq}$. In addition, we also assume that two internal parameters $[A_1, A_2]$ used in calculating P_S vary by 0.01 each. The problem we are solving, Eq. (6.6), has two P_S objectives, so we have a pair of $[A_1, A_2]$ variations in calculating P_S. It is also assumed that two of the design variables have uncontrollable variations as well: $[\Delta PL_0, \Delta DH_0] = [0.01, 0.01]$. In total, there are 7 uncertain parameters in this problem: $[Ceq, A_{1,s1}, A_{2,s1}, A_{1,s2}, A_{2,s2}, PL, DH]$. The maximum allowable variations in the P_S's are specified to be $[\Delta P_{S1,0}, \Delta P_{S2,0}] = [0.025, 0.025]$.

For the designer, the objective and feasibility robustness of the payload designs are equally important, and the desired robustness is specified to be: $\eta_0 = \eta_{g,0} = 1.0$. Since the designer is indifferent towards the two types of robustness, we can use the combined inner problem, Eq. (6.4), to search for the robust Pareto optima of this problem. Adding our overall robustness constraint $1 - \frac{\eta_{\min}}{\eta_0} \le 0$ to Eq. (6.6) and solving it, we obtain the robust Pareto optima of the problem as shown in Figure 6.2. For comparison, Figure 6.2 also shows the nominal Pareto optima of the problem (i.e., Eq. (6.6) without the robustness constraint), and the Pareto optima obtained from solving the problem using a

probabilistic approach. In the probabilistic approach, we minimize the worst-case P_S of each objective in the form of the sum of mean and standard deviation of the PS's. The mean and standard deviation values are calculated by running 10,000 Monte Carlo simulations assuming uniform probability distribution.

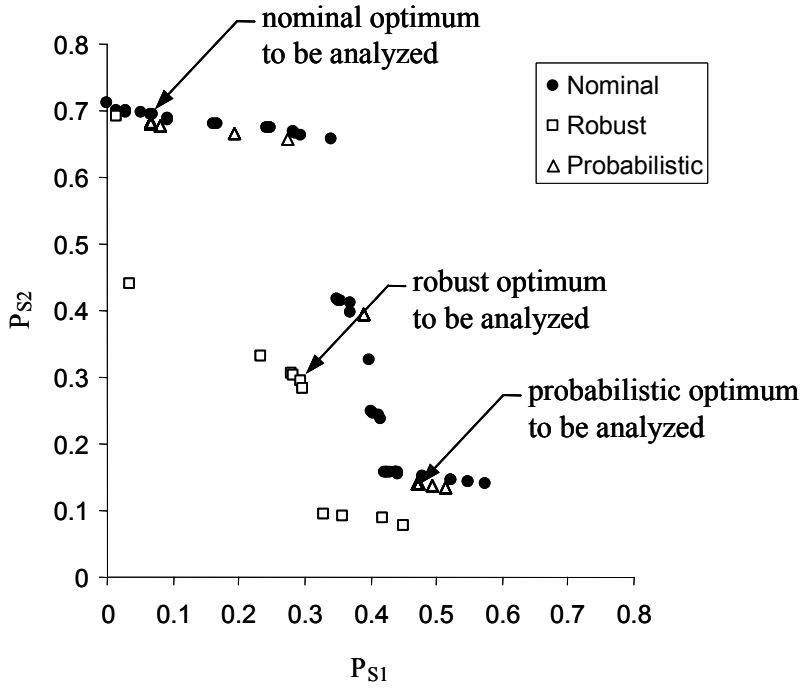


Figure 6.2: Pareto sets of the payload problem.

We see in Figure 6.2 that overall the robust Pareto optima are inferior to the nominal Pareto optima (this is a maximization problem), although there seems to be some overlap between the two Pareto frontiers. This observation is expected because there is a trade-off between the performance of an optimum and its robustness. We also see that the probabilistic Pareto solutions are very close to the nominal Pareto frontier, suggesting that these points do not meet our robustness requirement.

To verify the robustness of the designs, we performed a sensitivity analysis on three of the Pareto optimum designs obtained, one each from the nominal, robust, and

probabilistic Pareto set (randomly selected). The objective and constraint values of the three designs are shown in Table 6.1. Table 6.2 shows their design variable values.

Table 6.1: Objective and constraint values of the optima.

	Nominal	Robust	Probabilistic
P_{S1}	0.067	0.295	0.514
P_{S2}	0.695	0.295	0.135
Weight (lb)	85.000	84.433	84.95
Noise (W/m^2)	0.158	0.158	0.157

Table 6.2: Design variables of the optima.

	Nominal	Robust	Probabilistic
PL (inch)	19.679	24.262	24.237
DH (inch)	9.683	9.048	10.216
HM	7075AL	7075AL	7075AL
PT	BULK	MULTI_MISS	MULTI_MISS
I1	TYPE_1B	TYPE_1B	TYPE_1B
I2	TYPE_2A	TYPE_1B	TYPE_2B

We perform the sensitivity analysis by perturbing the 7 uncertain parameters [Ceq, $A_{1,s1}$, $A_{2,s1}$, $A_{1,s2}$, $A_{2,s2}$, PL, DH] around their original values, and then observing the changes in the objective and constraint values of the two designs. The perturbation values used in this analysis are randomly sampled from the given ranges of the parameter variations. A design meets the robustness criterion if these changes are within the specified limits. The results of the analysis are shown in Figure 6.3, 6.4, and 6.5 for the nominal, robust, and probabilistic optimum, respectively. In these figures, the dashed lines are the variation limits, and a design is robust if the bars do not cross these lines.

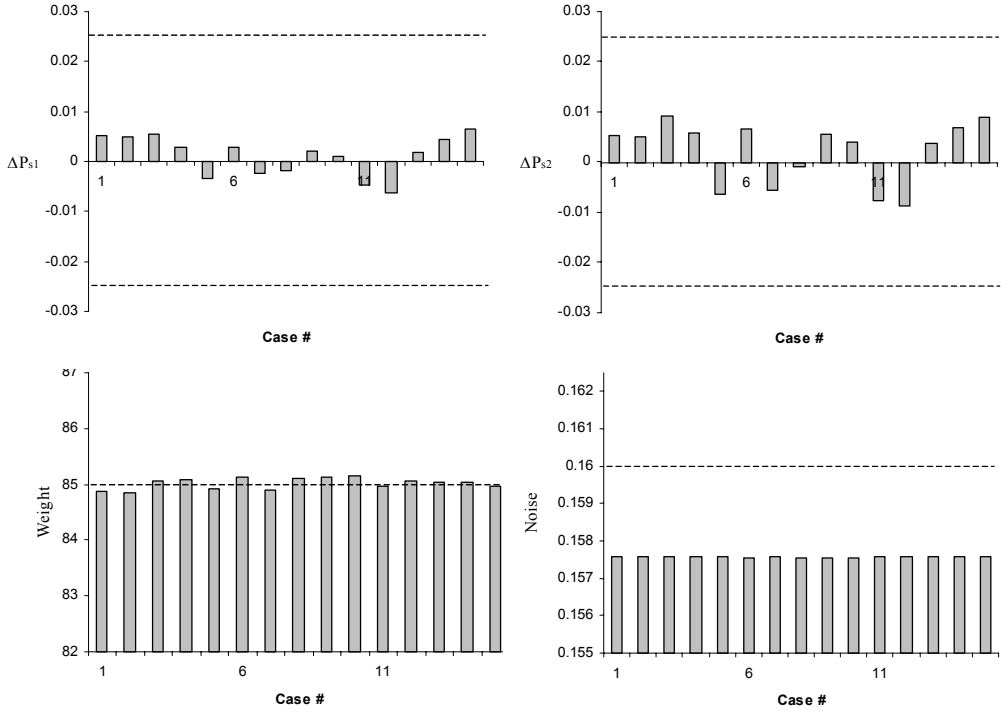


Figure 6.3: Sensitivity analysis of the nominal optimum payload.

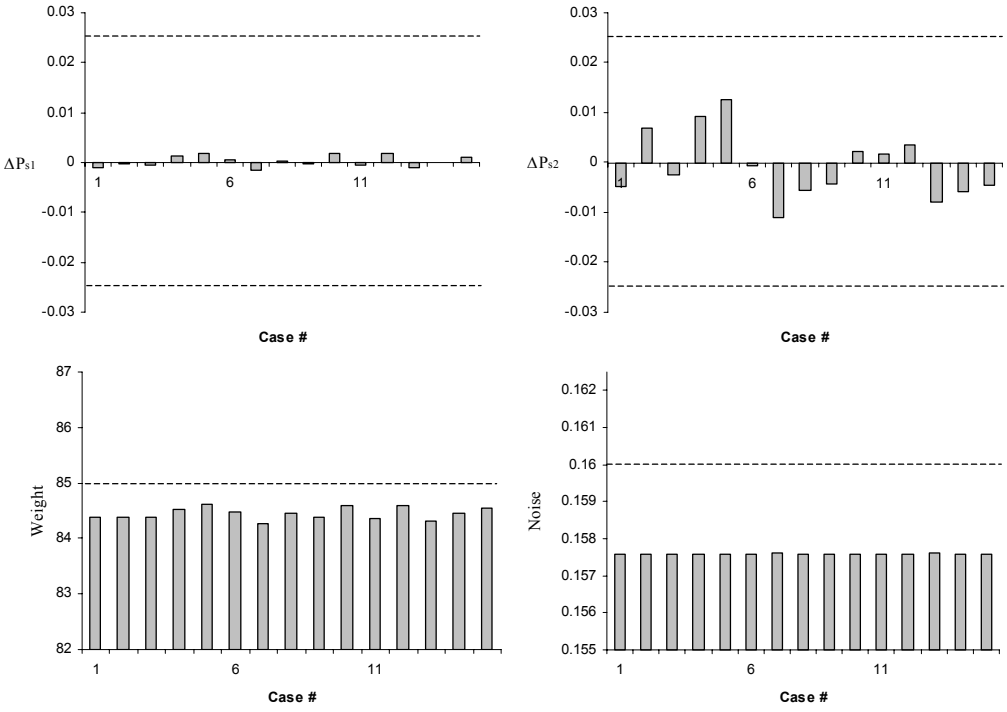


Figure 6.4: Sensitivity analysis of the robust optimum payload.

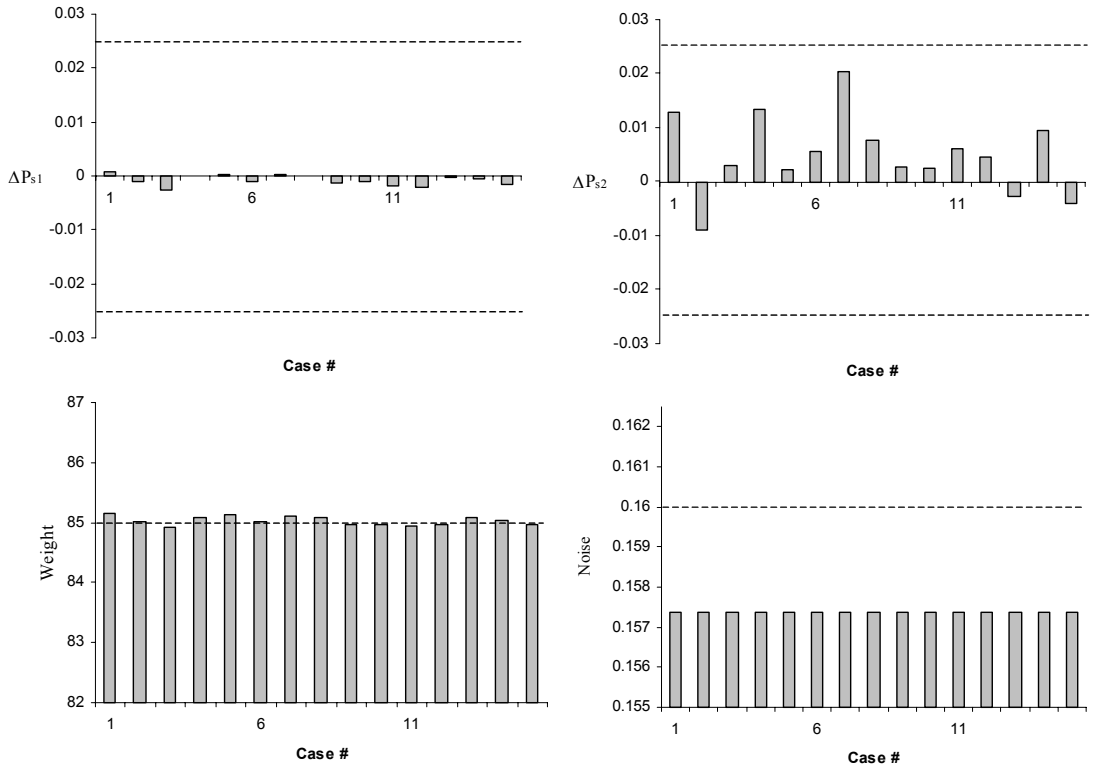


Figure 6.5: Sensitivity analysis of the probabilistic optimum payload.

We observe in Figure 6.3 that the nominal design satisfies both of the objective variation limits and the noise constraint. However, it does not satisfy the weight constraint. Similarly, the probabilistic design also satisfies the objective and noise robustness requirements, but not the weight constraint robustness (Figure 6.5). In contrast, we observe in Figure 6.4 that the robust design satisfies the variation limits for both objectives and constraints. This shows that the nominal and probabilistic designs do not meet the robustness criteria specified while the robust design does. In turn, these observations verify that using our robust optimization method to solve Eq. (6.6) indeed results in optimum designs that are robust with respect to both objectives and constraints.

6.3. ONE-SIDED SENSITIVITY FOR OBJECTIVE ROBUSTNESS

Up to this point, we have used a two-sided sensitivity measure to calculate objective robustness of a design. The comparison studies presented in Chapters 3 and 4 showed applications of this two-sided objective robustness measure to several examples. However, in some cases it may be more appropriate to use a one-sided sensitivity measure to calculate objective robustness of a design. For instance, if the objective of an optimization problem is to minimize the total cost of a design, then we are only interested in preventing the cost increase, but not the decrease, due to parameter variations. For this type of optimization problems, we should use a one-sided sensitivity measure to calculate objective robustness of a design alternative, and not a two-sided one.

A one-sided sensitivity measure for objective robustness is essentially the same as the one-sided sensitivity measure for feasibility robustness (recall Chapter 5) except that now we are looking at objective functions instead of constraint functions. Suppose $\Delta \boldsymbol{f}_0 = [\Delta f_{1,0}, \Delta f_{2,0}, \ldots, \Delta f_{M,0}]$ is the maximum allowable increase in the objective values of a design $\boldsymbol{x}_0$ due to parameter variations. The one-sided $\mathbf{S}_f$ of $\boldsymbol{x}_0$ is then as follows (notice the absence of the square terms):

$$\begin{aligned} &\mathbf{S}_f(\boldsymbol{x}_0) = \left\{\Delta\mathbf{p} \in \mathrm{R}^{\mathrm{G}} :\ \Delta f_i(\Delta\mathbf{p}) \le \Delta f_{i,0}, \forall i = 1,\ldots,\mathrm{M}\right\} \\ &\text{where}: \quad \Delta f_i(\Delta\mathbf{p}) = f_i(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f_i(\boldsymbol{x}_0, \mathbf{p}_0) \end{aligned} \tag{6.7}$$

In Eq. (6.7) we have constrained the increase in Δf_i and not the decrease because we are minimizing f_i. Since the smaller f_i the better, it is only logical that an increase in f_i is undesirable.

The plot of this $\mathbf{S}_f$ in the $\Delta\mathbf{p}$-space is the one-sided SR of $\boldsymbol{x}_0$, and the worst-case estimate of this SR is the one-sided WCSR of $\boldsymbol{x}_0$. The radius of this one-sided WCSR

(R_f^+) is obtained by solving the inner optimization problem shown in Eq. (6.8). Notice that Eq. (6.8) is the same as the inner problem to calculate feasibility robustness (Eq. (5.11)), except that now the equality constraint involves f_i instead of g_j.

$$\begin{aligned} &\underset{\Delta\mathbf{p}}{\text{minimize}} \quad R_f^+(\Delta\mathbf{p}) = \left[\sum_{i=1}^{G} \Delta p_i^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{i=1,\ldots,M}\left[\frac{\Delta f_i(\Delta\mathbf{p})}{\Delta f_{i,0}}\right] - 1 = 0 \end{aligned} \tag{6.8}$$

Eq. (6.9) shows the normalized version of Eq. (6.8).

$$\begin{aligned} &\underset{\Delta\bar{\mathbf{p}}}{\text{minimize}} \quad R_f^+(\Delta\bar{\mathbf{p}}) = \left[\sum_{j=1}^{G} (\Delta\bar{p}_j)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{i=1,\ldots,M}\left(\frac{\Delta f_i(\Delta\bar{\mathbf{p}} \otimes \Delta\mathbf{p}_0)}{\Delta f_{i,0}}\right) - 1 = 0 \end{aligned} \tag{6.9}$$

Using the R_f^+ value obtained from Eq. (6.9), the one-sided objective robustness of $\boldsymbol{x}_0$ is then indicated by the one-sided robustness index $\eta^+ = (G)^{-1/2} R_f^+$, where a value $\eta^+ \geq 1.0$ shows that $\boldsymbol{x}_0$ is robust, while a value $\eta^+ < 1.0$ shows that $\boldsymbol{x}_0$ is not robust. Formulating the one-sided objective robustness constraint $1 - \frac{\eta^+}{\eta_0^+} \leq 0$, where η_0^+ is the desired level of robustness, the optimization problem to obtain an optimum design that is objectively robust one-sidedly is as shown in Eq. (6.10).

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \left[f_1(\boldsymbol{x},\mathbf{p}_0),\ldots,f_M(\boldsymbol{x},\mathbf{p}_0)\right] \\ &\text{subject to}: \quad g_j(\boldsymbol{x},\mathbf{p}_0) \leq 0 \quad ; \quad j = 1,\ldots,J \\ &\qquad\qquad h_k(\boldsymbol{x},\mathbf{p}_0) = 0 \quad ; \quad k = 1,\ldots,K \\ &\qquad\qquad 1 - \frac{\eta^+}{\eta_0^+} \leq 0 \qquad \text{(one-sided objective robustness)} \end{aligned} \tag{6.10}$$

6.4. ASYMMETRICAL TWO-SIDED SENSITIVITY MEASURE

In calculating the two-sided sensitivity of a design's objectives, we have used a single positive value $\Delta f_{i,0}$ to limit both the increase and decrease in f_i, i.e., a symmetrical two-sided sensitivity. Sometimes, however, it is desired to have different limits for the increase and decrease in f_i, i.e., an asymmetrical two-sided sensitivity. The calculation of the asymmetrical robustness index of a design is a straightforward extension of the symmetrical one.

Let $\Delta \boldsymbol{f}_0^- = [\Delta f_{1,0}^-,...,\Delta f_{\mathrm{M},0}^-]$ and $\Delta \boldsymbol{f}_0^+ = [\Delta f_{1,0}^+,...,\Delta f_{\mathrm{M},0}^+]$ be the maximum allowable decrease and increase in $\boldsymbol{f}$, respectively. Here we assume that $\Delta f_{i,0}^-, \Delta f_{i,0}^+ > 0$ and $\Delta f_{i,0}^- \neq \Delta f_{i,0}^+$ for all i=1,…,M. If $\Delta \boldsymbol{f}_0^- = \Delta \boldsymbol{f}_0^+$ then we are essentially looking at a symmetrical two-sided sensitivity (recall Chapter 4). Using these two allowable limits, the asymmetrical $\mathbf{S}_f$ of a design $\boldsymbol{x}_0$ is then as follows:

$$\begin{aligned} &\mathbf{S}_f(\boldsymbol{x}_0) = \left\{\Delta\mathbf{p} \in \mathrm{R}^{\mathrm{G}} : -\Delta f_{i,0}^- \le \Delta f_i(\Delta\mathbf{p}) \le \Delta f_{i,0}^+, \forall i = 1,...,\mathrm{M}\right\} \\ &\text{where}: \quad \Delta f_i(\Delta\mathbf{p}) = f_i(\boldsymbol{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}) - f_i(\boldsymbol{x}_0, \mathbf{p}_0) \end{aligned} \tag{6.11}$$

Notice in Eq. (6.11) that the square terms are now replaced by two inequalities.

The plot of this $\mathbf{S}_f$ in the $\Delta\mathbf{p}$-space is the asymmetrical SR of $\boldsymbol{x}_0$, while the worst-case estimate to this SR is the asymmetrical WCSR. The inner optimization problem to calculate the radius of the asymmetrical WCSR ($\mathrm{R}_f^{-/+}$) is shown in Eq. (6.12) (this is the normalized version).

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \;\; \mathrm{R}_f^{-/+}(\Delta\overline{\mathbf{p}}) = \left[\sum_{j=1}^{\mathrm{G}} (\Delta\overline{\mathrm{p}}_j)^2\right]^{\frac{1}{2}} \\ &\text{subject to}: \; \max\left\{\max_{i=1,...,\mathrm{M}}\left(\frac{|\Delta f_i(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)|}{\Delta f_{i,0}^-}\right), \max_{i=1,...,\mathrm{M}}\left(\frac{\Delta f_i(\Delta\overline{\mathbf{p}} \otimes \Delta\mathbf{p}_0)}{\Delta f_{i,0}^+}\right)\right\} - 1 = 0 \end{aligned} \tag{6.12}$$

The equality constraint in Eq. (6.12) shows the asymmetry of this two-sided measure. The first and second terms in the equality constraint in Eq. (6.12) correspond to the decrement and increment limits in f_i, respectively.

The asymmetric robustness index of $\boldsymbol{x}_0$ is then calculated as $\eta^{-/+} = (\mathrm{G})^{-1/2}\,\mathrm{R}_f^{-/+}$ where $\mathrm{R}_f^{-/+}$ is obtained by solving Eq. (6.12). As before, the value $\eta^{-/+} \geq 1.0$ shows that design $\boldsymbol{x}_0$ is robust while $\eta^{-/+} < 1.0$ shows that it is not robust. To obtain an optimum design that is objectively robust asymmetrically, we only need to add the robustness constraint $1 - \dfrac{\eta^{-/+}}{\eta_0^{-/+}} \leq 0$ to the optimization problem of interest ($\eta_0^{-/+}$ is the desired level of robustness, and is specified by the designer).

6.5. ASYMMETRICAL PARAMETER VARIATIONS

One of the assumptions of our robust optimization method is that the parameter variation ranges are symmetric: $-\Delta p_{0,i} \leq \Delta p_i \leq \Delta p_{0,i}$, i=1,…,G ($\Delta p_{0,i} > 0$). This assumption is necessary because we are using a single point normalization to account for the scale difference of the $\Delta\mathbf{p}$ when measuring the robustness of a design. Sometimes, however, the variation ranges are not symmetric, and when this occurs, our robustness measure must be modified accordingly to account for this asymmetry.

Let $\Delta\mathbf{p}_0^- = [\Delta p_{0,1}^-, ..., \Delta p_{0,\mathrm{G}}^-]$ and $\Delta\mathbf{p}_0^+ = [\Delta p_{0,1}^+, ..., \Delta p_{0,\mathrm{G}}^+]$ be the lower and upper bounds of the parameter variation ranges, respectively. We assume that $\Delta p_{0,i}^- \neq \Delta p_{0,i}^+ > 0$ for all i=1,…,G (otherwise it is a symmetric range). The purpose of normalizing Δp_i is to map it to the [-1,1] range. When the parameter variation ($\Delta\mathbf{p}_0$) is symmetric, the

normalization $\Delta\overline{p}_i = \dfrac{\Delta p_i}{\Delta p_{0,i}}$ provides such a mapping. When $\Delta\mathbf{p}_0$ is not symmetric, however, there are two cases to consider: the negative and positive Δp_i. If Δp_i is negative, the normalization must be performed with respect to $\Delta\mathbf{p}_0^-$. If Δp_i is positive, the normalization must be performed with respect to $\Delta\mathbf{p}_0^+$. Based on this fact, the normalization for the asymmetric $\Delta\mathbf{p}$ is then: $\Delta\overline{p}_i = \begin{cases} \dfrac{\Delta p_i}{\Delta p_{0,i}^-}, \Delta p_i \le 0 \\ \dfrac{\Delta p_i}{\Delta p_{0,i}^+}, \Delta p_i > 0 \end{cases}$ or $\Delta p_i = \begin{cases} (\Delta p_{0,i}^-)(\Delta\overline{p}_i), \Delta\overline{p}_i \le 0 \\ (\Delta p_{0,i}^+)(\Delta\overline{p}_i), \Delta\overline{p}_i > 0 \end{cases}$. Using the ternary operator $\langle\bullet\rangle$ defined in Chapter 2, this normalization can be written in a compact form as: $\Delta\mathbf{p} = \langle \Delta\overline{\mathbf{p}}, \Delta\mathbf{p}_0^-, \Delta\mathbf{p}_0^+ \rangle$.

Because now the $\Delta\mathbf{p}_0$ ranges have been normalized to be [-1,1], the radius of the exterior hyper-sphere of these ranges are still = $(G)^{-1/2}$. So, the calculation of the objective and feasibility robustness index remain the same as before, i.e., $\eta = (G)^{-1/2} R_f$ and $\eta_g = (G)^{-1/2} R_g$, respectively. However, the calculation of the radius of the WCSR and FWCSR have to be modified as shown in Eq. (6.13) and (6.14), respectively.

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \;\; R_f(\Delta\overline{\mathbf{p}}) = \left[\sum_{j=1}^{G} (\Delta\overline{p}_j)^2 \right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{i=1,\ldots,M} \left(\frac{\left| \Delta f_i(\langle \Delta\overline{\mathbf{p}}, \Delta\mathbf{p}_0^-, \Delta\mathbf{p}_0^+ \rangle) \right|}{\Delta f_{i,0}} \right) - 1 = 0 \end{aligned} \tag{6.13}$$

$$\begin{aligned} &\underset{\Delta\overline{\mathbf{p}}}{\text{minimize}} \;\; R_g(\Delta\overline{\mathbf{p}}) = \left[\sum_{i=1}^{G} \Delta\overline{p}_i^2 \right]^{\frac{1}{2}} \\ &\text{subject to}: \max_{j=1,\ldots,J} \left[g_j(\boldsymbol{x}_0, \mathbf{p} + \langle \Delta\overline{\mathbf{p}}, \Delta\mathbf{p}_0^-, \Delta\mathbf{p}_0^+ \rangle) \right] = 0 \end{aligned} \tag{6.14}$$

6.6. ROBUSTNESS INDEX VS. ROBUSTNESS PROBABILITY

Our robust optimization method measures the robustness of a design based on the values of its robustness index: η and η_g for objective and feasibility robustness, respectively. The magnitude of a robustness index tells us the degree of robustness of a design: the larger η (and/or η_g), the more robust the design. However, since we are only provided with ranges of $\Delta\mathbf{p}_0$, η (and/or η_g) does not tell us the actual robustness probability of the design. (For objective robustness, the robustness probability is the probability that the objective values of the design stay within the acceptable limits. For feasibility robustness, it is the probability of constraint satisfaction of the design.) Nevertheless, if we know the probability distribution of $\Delta\mathbf{p}_0$, we can use the value of η and/or η_g to calculate a lower bound on the robustness probability of the design.

The procedure to calculate the robustness probability is described next. Here we only show the calculation for independent uniform and normal distributions. Probability calculations for other distributions and when there are correlations will be somewhat more involved, but the basic procedure is the same.

Uniform Distribution

Suppose $\Delta\mathbf{p}$ is distributed uniformly within [**-1**,+**1**] and there is no correlation among its Δp_i components. If the distribution is not within [**-1**,+**1**], it can be normalized to fall into this range (it has to be normalized anyway since we are going to compare it with the normalized robustness index). The probability density function of this distribution is $f_x(\Delta\mathbf{p}) = (2)^{-G}$, and the probability that a random variable $\mathbf{X}$ is between $[\Delta\mathbf{p}_1,\Delta\mathbf{p}_2]$ is:

$$P[\Delta\mathbf{p}_1 \le \mathbf{X} \le \Delta\mathbf{p}_2] = \int_{\Delta p_{1,1}}^{\Delta p_{2,1}} \cdots \int_{\Delta p_{1,G}}^{\Delta p_{2,G}} f_x(\Delta\mathbf{p})\, d(\Delta p_1) \ldots d(\Delta p_G) \qquad (6.15)$$

Substituting $f_x(\Delta\mathbf{p}) = (2)^{-G}$ into Eq. (6.15), $P[\Delta\mathbf{p}_1 \le \mathbf{X} \le \Delta\mathbf{p}_2] = (2)^{-G}\prod_{i=1}^{G}(\Delta p_{2,i} - \Delta p_{1,i})$.

A value of $\eta = 1.0$ indicates that the WCSR of a design encloses the [**-1**,+**1**] range in $\Delta\mathbf{p}$-space. (In our discussion we use η, but it is applicable to η_g as well.) More generally, a robustness index of η tells us that the WCSR of the design encloses the [$-\boldsymbol{\eta}$,$+\boldsymbol{\eta}$] range in $\Delta\mathbf{p}$-space. Recall that the $\Delta\mathbf{p}$ range defined by a WCSR tells us the $\Delta\mathbf{p}$ that *must* occur if we want the $\Delta\boldsymbol{f}_0$ limit to be satisfied, while those $\Delta\mathbf{p}$ defined by the probability distributions are the $\Delta\mathbf{p}$ that actually *does* occur. In other words, the probability that the design will satisfy the $\Delta\boldsymbol{f}_0$ limit is equal to the probability that a random $\Delta\mathbf{p}$ falls in the [$-\boldsymbol{\eta}$,$+\boldsymbol{\eta}$] range, i.e., $P[-\boldsymbol{\eta} \le \Delta\mathbf{p} \le +\boldsymbol{\eta}]$. Using the formula obtained previously, the probability that a design will satisfy the $\Delta\boldsymbol{f}_0$ limit is then: $P[-\boldsymbol{\eta} \le \mathbf{X} \le +\boldsymbol{\eta}] = (2)^{-G}\prod_{i=1}^{G}(\eta - (-\eta)) = \eta^G$. Table 6.3 shows the probability values for several instances of η and G.

Table 6.3: Probability values for uniform distribution.

η	G			
	1	2	3	4
0.01	0.01	0	0	0
0.1	0.10	0.01	0.001	0
0.5	0.5	0.25	0.125	0.063
0.8	0.8	0.64	0.512	0.410
0.9	0.9	0.81	0.729	0.656
1.0	1.0	1.0	1.0	1.0

It is very important to point out that the probability calculation above is valid only if each Δp_i is uniformly distributed and they are independent. In addition, the formula $P[.] = \eta^G$ is only a lower bound (worst case) of the actual robustness probability value of

the design. This is because we are comparing the uniform distribution (which is a hyper-cube) with a WCSR (which is a hyper-sphere). So, there are some $\Delta\mathbf{p}$ values that are part of WCSR (and hence η), but are not included in the probability calculation. Another reason our probability calculation is only a lower bound is because WCSR is a worst-case estimate of the actual SR, so again, there are some $\Delta\mathbf{p}$ values that in reality the design can absorb, but are not included in our calculation.

Normal Distribution

Suppose $\Delta\mathbf{p}$ is multi-variate normally and independently distributed with a mean $\boldsymbol{\mu} = \mathbf{0}$ and a covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} 1/9 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/9 \end{bmatrix}$. We use a standard deviation of $\sigma_{ii} = 1/3$ so that the range [-1,1] covers $3\sigma_{ii}$ of the distribution. The probability density function of this distribution is $f_x(\Delta\mathbf{p}) = \left(\frac{1}{(2\pi)^G |\boldsymbol{\Sigma}|}\right)^{1/2} \exp\left(-\frac{1}{2}(\Delta\mathbf{p}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\Delta\mathbf{p}-\boldsymbol{\mu})\right)$ where $|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$ are the determinant and inverse of $\boldsymbol{\Sigma}$, respectively. The probability that a random variable $\mathbf{X}$ is between $[\Delta\mathbf{p}_1, \Delta\mathbf{p}_2]$ is obtained by substituting this density function into Eq. (6.15) and then performing the integration. There is no closed-form solution to this integration, so we must numerically calculate it.

As with the uniform distribution, given the robustness index η, the robustness probability of a design is equal to the probability that a random $\Delta\mathbf{p}$ falls into the $[-\boldsymbol{\eta}, +\boldsymbol{\eta}]$ range: $P[-\boldsymbol{\eta} \le \Delta\mathbf{p} \le +\boldsymbol{\eta}]$. Table 6.4 shows the probability values for several instances of η and G (these values are obtained numerically).

Table 6.4: Probability values for normal distribution.

η	G			
	1	2	3	4
0.01	0.024	0	0	0
0.1	0.236	0.056	0.013	0.004
0.5	0.866	0.742	0.639	0.551
0.8	0.984	0.950	0.926	0.903
0.9	0.993	0.968	0.952	0.936
1.0	0.997	0.975	0.963	0.951

Keep in mind again that the values in Table 6.4 are valid only if the $\Delta\mathbf{p}$ distribution is normal. Also, these values are the lower bounds of the actual values of the robustness probability of the design. Unlike the uniform distribution, however, the lower bound for $\eta = 1.0$ is not 1.0, rather it decreases as G increases. The reason for this is because we only use $\sigma_{ii} = 1/3$ for the distribution. As G increases, the "tail" region of the normal distribution is becoming larger, and this region is not included in our probability calculation.

Example 6.1

Let us revisit the control valve actuator linkage example from Chapter 5. Previously, we obtained a set of optimum designs for various values of η (Figure 5.21). Using the probability values in Tables 6.3 and 6.4, estimate the probability of constraint satisfaction (P_s) of each design assuming uniform and normal $\Delta\mathbf{p}$ distribution. Calculate the actual P_s of the designs, and compare them to the estimated values.

Solution

We can easily estimate the P_s of the designs using the strategy explained previously. To calculate the actual P_s of the designs, we performed 100,000 runs of Monte Carlo

simulations for each design, for each distribution model. Figure 6.6(a) and (b) show the plots of the estimate and actual P_s of the designs for the uniform and normal distribution models, respectively.

We see in both Figure 6.6(a) and (b) that the estimate P_s values are always lower than the actual P_s values. This is because the estimated values are only a lower bound of the actual values. Figure 6.6 also shows that the difference between the estimate and the actual P_s values decreases as η increases. Intuitively, this observation is expected. The discrepancy between the estimated and the actual values is mainly caused by the fact that we have used a worst-case estimate of the FSR to measure a design's robustness. As the actual P_s approaches 1.0, η increases, and the FWCSR of the design will become more and more like the actual $\Delta\mathbf{p}$ distribution. Therefore, as η increases, the lower bound P_s will approach that of the actual P_s. ♦

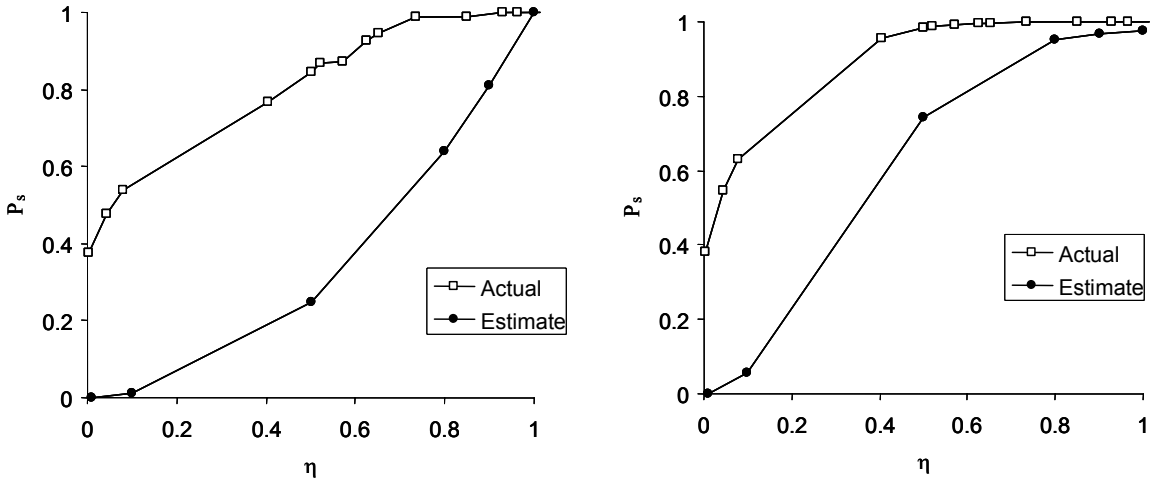


Figure 6.6: Comparison between the estimate and actual P_s.

6.7. SUMMARY

- Adding objective and feasibility robustness constraints to an optimization problem results in optimum designs that are robust both objectively and feasibly.
- The advantage of keeping the objective and feasibility robustness separate is that it provides the designer the flexibility to state his/her preference towards either of the two types of robustness (i.e., by changing η_0 and $\eta_{g,0}$).
- When a designer is indifferent towards the two types of robustness, the overall robustness of the design can be calculated by solving just one inner optimization problem instead of two.
- Practically, it is easier to solve just one inner optimization problem (to calculate the design's robustness) than to solve two inner problems. Also, it helps the outer problem converges faster as well as reduces the error transmitted from the inner to the outer problem. However, solving two inner problems might be computationally more efficient.
- Sometimes it is more appropriate to use a one-sided sensitivity measure for objective robustness. This one-sided measure is the same as the one-sided measure used in feasibility robustness, except that in this case we use objective functions instead of constraint functions in the formulation.
- For a two-sided objective robustness, sometimes it is desired to have different allowable limits for the decrease and increase in f_i, i.e., an asymmetric two-sided sensitivity. This asymmetry can be easily incorporated into our robustness formulation by changing the square terms in $\mathbf{S}_f$ by two inequalities.

- In our robust optimization method, we had made a necessary assumption that parameter variations are symmetric. However, in the event they are not symmetric, our symmetric normalization can be easily modified to account for it.
- Since our method does not use a presumed probability density function (pdf), the value of the robustness index does not provide us with the probability information of the degree of robustness of the design. However, if the pdf of $\Delta\mathbf{p}$ is known, we can use the η value to calculate a lower bound of the robustness probability of the design.

CHAPTER 7

CONCLUSIONS

7.1. CONCLUDING REMARKS

In this dissertation, we have presented a step-by-step development of a novel method for robust design optimization. After presenting our research objective and review of previous work in Chapters 1 and 2, we developed our method for objective robust optimization for a single objective problem in Chapter 3, and then extended it to multi-objective problems in Chapter 4. In Chapter 5, we developed a method for feasibility robust optimization of a design. Chapter 6 presented our combined objective and feasibility robust optimization method.

The essence of our robust optimization method is the robustness measurement of a design alternative using a sensitivity region concept. A sensitivity region is an inherent property of a design that shows how much parameter variations the design can absorb given a limit on its performance variation. The more parameter variations a design can absorb (i.e., the more we can allow the parameters to vary), the more robust the design is. In the method, we use the worst-case estimate of the sensitivity region as a measure of a design's robustness. Based on this worst-case estimate, we calculate a robustness index for the design, which we then constrain and add to the original optimization problem to guarantee the robustness of the optimum design solution obtained.

In Chapters 3-6, we demonstrated the application of our robust optimization method to several numerical and engineering examples. For comparison, we also solved some of the problems using several other well-known robust optimization methods. We showed in

these examples that our method indeed obtains a design that is robust and optimum, and that our method is computationally efficient.

In the next few subsections, we provide some additional concluding remarks regarding the results of our research.

7.1.1. Verification

In Chapters 3 through 6 we solved several numerical and engineering examples and analyzed the results obtained to verify our robust optimization methods.

In the wine-bottle function example (Section 3.4.1), we can graphically verify the validity of our robust optimization method. The function has a unique property in that there is a flat region around the middle at which the function is insensitive to the variable variations, i.e., a robust region. A robust optimization method is valid if its solutions are within this flat region. We observe that the solutions from our method indeed fall into the flat region, thus verifies that our method is a valid robust optimization method for this example.

For the other comparison studies, graphical verification is difficult or impossible to do. So instead, we performed a sensitivity analysis on the results obtained to see if they are indeed robust. The results of our sensitivity analysis are summarized in Table 7.1. In this table, the symbol "√" means that the optimum designs obtained are robust, while the symbol "×" means that they are not. We show in Table 7.1 four categories of robust optimization methods. "Nominal" method refers to a regular optimization method without robustness consideration. "Robust" method refers to our robust optimization method with $\eta_0 = \eta_{g,0} = 1.0$. "Sampling" method refers to those methods that perform a

localized sampling around the nominal parameter values (e.g., Monte Carlo). "Gradient" method refers to those methods that use the gradient of the functions in calculating a design's robustness (e.g., worst-case gradient, moment matching).

Table 7.1: Summary of sensitivity analysis results.

	Nominal	Robust	Sampling	Gradient
Three-bar truss	×	√	×	
Welded beam	×	√	×	
Compression spring	×	√		√
Numerical (multi-objective)	×	√	×	
Vibrating platform	×	√	×	
Speed reducer	×	√	×	
Power electronic module	×	√	×	
Numerical (feasibility)	×	√		×
Explosive actuated cylinder	×	√	×	
Belleville spring	×	√		×
Control valve actuator linkage	×	√	×	
Payload for an UAV	×	√	×	

We see in Table 7.1 that optimum designs obtained by our robust optimization method always satisfy the robustness requirements. This observation verifies that our method is indeed a valid robust optimization method. In contrast, optimum designs obtained by a regular optimization (the "Nominal" method) are not robust for these examples. This is expected since this method does not account for a design's robustness. The optima of the "Sampling" method also do not satisfy the robustness requirement. This is because the method uses probability to measure a design's robustness, so there is a chance (albeit small) that the optimum design will violate the requirement. In the sensitivity analysis, a design is termed robust only if it never violates the requirement. The optimum of the "Gradient" method is robust in the compression spring example, but it has a very poor objective value (recall Section 3.4.4). In the numerical example in

Chapter 5 and Belleville spring design examples, the "Gradient" method fails to obtain a robust optimum design.

7.1.2. Computational Efficiency

Our robust optimization method calculates the robustness of a design by solving an inner optimization problem, which is a single objective problem with an equality constraint. When the objective/constraint functions of the outer problem (the original optimization problem) are simple enough, analytic solutions to the inner problem may be possible, in which case our method does not need to perform any function evaluations. When analytic solutions are not possible, the inner problem may be solved by a gradient-based optimization algorithm such as Sequential Quadratic Programming. The computational cost of such algorithms is generally in the order of 10^1. When gradient-based algorithms are not applicable (e.g., when the functions are non-differentiable), stochastic algorithms such as GA may be used instead. For high solution accuracy, the computational cost for stochastic algorithms is generally in the order of more than 10^3. However, throughout our comparison studies, we found that GA can solve the inner optimization problem using only ~200-300 function evaluations. This is because the inner problem is not too difficult an optimization problem to solve. Its objective function is convex and unimodal, and the search space is not large.

Table 7.2 shows a summary of the number of function evaluations (F_{call}) performed by the four methods to calculate the robustness of one design alternative.

Table 7.2: Summary of average number of function evaluations.

	Nominal	Robust	Sampling	Gradient
Wine-bottle	N/A	30		
Three-bar truss	N/A	39	10000	
Welded beam	N/A	250	100000	
Compression spring	N/A	250		0
Numerical (multi-objective)	N/A	250	100000	
Vibrating platform	N/A	250	100000	
Speed reducer	N/A	300	100000	
Power electronic module	N/A	300	100000	
Numerical (feasibility)	N/A	24		0
Explosive actuated cylinder	N/A	45	100000	
Belleville spring	N/A	300		0
Control valve actuator linkage	N/A	250	100000	
Payload for an UAV	N/A	300	10000	

In Table 7.2, the F_{call} for the "Nominal" method is 0 because this method does not calculate the design's robustness. The F_{call} for the "Gradient" method is also 0, but this is because the gradient of the functions in the examples are known in closed form. If the gradient has to be numerically estimated, the F_{call} will be non-zero, depending on the dimension of the inner problem. The F_{call} of our robust optimization method varies from 24 to 300. The F_{call} is either 250 or 300 when we used GA to solve the inner problem, and it is less than 50 when we used *fmincon*. Overall, the F_{call} of our method is much lower than the "Sampling" method, whose F_{call} ranges from 10,000 to 100,000.

Table 7.3 shows a summary of the computational cost of the four methods in terms of absolute time ("hr" for hour, "m" for minute, and "s" for second), and the optimization algorithms used in each problem. These time values are obtained using a Pentium III 866 MHz computer with 384 RAM.

Table 7.3: Summary of computational time.

	Nominal	Robust	Sampling	Gradient	Algorithm
Wine-bottle	2 s	30 m 43 s			GA
Three-bar truss	0.25 s	3 s	3 m 17 s		*fmincon*
Welded beam	18 s	8 m 25 s	1 hr 15 m		GA
Compression spring	3 s	8 m 14 s		3 s	GA
Numerical (multi-objective)	4 s	18 m 3 s	2 hr 3 m		NSGA
Vibrating platform	4 s	19 m 8 s	1 hr 32 m		MOGA
Speed reducer	5 s	12 m 58 s	1 hr 33 m		NSGA
Power electronic module	5 s	33 m 6 s	7 hr 42 m		NSGA
Numerical (feasibility)	0.22 s	2.5 s		4 s	*fmincon*
Explosive actuated cylinder	1 s	18 s	1 m 28 s		*fmincon*
Belleville spring	4 s	12 m 47 s		N/A	GA
Control valve act linkage	11 s	16 m 17 s	2 hr 28 m		GA
Payload for an UAV	5 s	22 m 29 s	1 hr 52 m		NSGA

The values in Table 7.3 confirm the data shown in Table 7.2. Overall, the "Nominal" method is the fastest, requiring at maximum only 18 sec to solve the problem. This is not surprising since this method does not perform any additional function evaluations. The "Gradient" method is also very fast since the gradient information is available analytically. The "Sampling" method is the slowest and very computationally extensive. For the power electronic module example, it took more than 7 hours to complete the optimization process. In contrast, the computation time of our robust optimization method is much less. It ranges from a few seconds to several minutes depending on the algorithms used to solve the inner optimization problem (*fmincon* and GA, respectively).

7.1.3. Advantages and Disadvantages

The main difference between our robust optimization method and the other methods is that our method calculates the robustness of a design in a "reverse" mode. That is,

instead of calculating $\Delta \boldsymbol{f}$ (or $\Delta \mathbf{g}$ for constraints) for a given $\Delta \mathbf{p}$, our method calculates $\Delta \mathbf{p}$ for a given $\Delta \boldsymbol{f}$ (or $\Delta \mathbf{g}$). The advantage of working in this reverse mode is that the robustness information provided by our method does not depend on the actual $\Delta \mathbf{p}$. Should the actual uncontrollable $\Delta \mathbf{p}$ change, the sensitivity region of the design will not change, so we can still use this information to determine the design's robustness with respect to the new $\Delta \mathbf{p}$. In contrast, for the conventional "forward" methods, if $\Delta \mathbf{p}$ changes, then the design's robustness previously calculated is no longer valid, and we will have to re-evaluate it.

Because our robustness calculation does not depend on the actual $\Delta \mathbf{p}$, our method is independent of the probability distribution of $\Delta \mathbf{p}$. As long as $\Delta \mathbf{p}$ falls within the sensitivity region of the design, the design is guaranteed to satisfy the $\Delta \boldsymbol{f}_0$ (or $\Delta \mathbf{g}_0$) requirement. If the probability distribution of $\Delta \mathbf{p}$ changes, the robustness probability of the design will change, but this guarantee stays the same.

Another advantage of our method is that it does not use the gradient information of the objective/constraint functions. As a result, our robustness calculation is valid even if the $\Delta \mathbf{p}$ variations are large, beyond the linear range in which gradient estimation is valid. This is in contrast with those methods that use gradient calculations, such as a Taylor series expansion, to calculate a design's robustness. The numerical example in Chapter 5 (Section 5.4.1) showed how gradient-based robustness methods fail when $\Delta \mathbf{p}$ becomes large, while our method is still valid. Since our method does not use gradient information, it is also applicable to optimization problems whose objective/constraint functions are not differentiable everywhere with respect to $\Delta \mathbf{p}$. This is demonstrated in the welded beam

example in Section 3.4.3, where the objective function of the problem is a step-function with respect to $\Delta\mathbf{p}$.

Our method is also computationally efficient. As experimentally showed, the upper bound for the number of function evaluations needed by our method is in the order of 10^2 when stochastic algorithms are necessary to solve the inner problem. Our method uses more F_{call} than the gradient-based methods. However, the applications of the gradient-based methods are limited to small range of $\Delta\mathbf{p}$ variations. Our method is more efficient than sampling-based methods whose F_{call} is in the order of 10^3 or more. Even for the more efficient sampling-based methods (such as MPP (Du and Chen, 2000)), the F_{call} is still in the order of 10^2 or above. Besides, our method does not need a presumed probability distribution to calculate a design's robustness.

One shortcoming of our method is that it is conservative because it uses only the worst-case estimate of the sensitivity region to determine a design's robustness. So, there are some $\Delta\mathbf{p}$ variations that in reality the design can absorb, but they are not included in the calculations. Our method also does not provide probability information regarding a design's robustness. However, this is not because we cannot calculate the probability, but rather because we do not assume a probability distribution of the $\Delta\mathbf{p}$. If the pdf of $\Delta\mathbf{p}$ is known, we can numerically calculate a lower bound of the probability (recall Section 6.6). If the actual probability of the design is necessary, it may be interpolated experimentally. We first solve the problem for the robust optimum using several values of η_0, and then calculate the actual probability of these optima. The probability of an optimum for other values of η_0 can then be interpolated from the results (recall Figure 6.5).

7.2. CONTRIBUTIONS

In this dissertation, we have introduced and developed several new and innovative concepts for robust optimization of a design. The contributions of the research presented in this dissertation are summarized below.

- Introduced and developed the notion of "reverse" robustness measure of a design alternative. This robustness measure does not require a presumed probability distribution of parameter variations. It also does not use gradient information so that it is valid for large variations of parameters, and is applicable to non-differentiable objective/constraint functions.
    - Introduced and developed the concept of "one-sided" and "two-sided" sensitivity set and sensitivity region of a design alternative for single and multiple objective/constraint functions. The concept of an asymmetrical two-sided sensitivity of a design has also been introduced and developed.
    - Introduced the notion of directional sensitivity of a design, and developed an approach to account for it using the worst-case estimation of sensitivity region.
    - Developed a mathematical formulation to calculate the radius of worst-case sensitivity region, and an approach to normalize the formulation to account for scale importance of parameters.
    - Developed an approach to calculate the lower bound of the robustness probability of a design when the probability distribution of the uncertain parameters is known.

- Developed an efficient constraint-based robust optimization method using the sensitivity region concept. The method is applicable to both single and multi-objective optimization problems, and can account for both objective and feasibility robustness of an optimum design.
    - Introduced and developed the concept of a robustness index for a design alternative, which is a measure of robustness calculated based on the radius of worst-case sensitivity region.
    - Developed an inner-outer optimization framework to efficiently search for design alternatives that are optimum and robust.
    - Introduced and developed the concepts of multi-objective robustness and multi-objective robust optimality of a design alternative.

7.3. FUTURE RESEARCH DIRECTIONS

The robust optimization method presented in this dissertation addresses many of the shortcomings of previous works in robust optimization. However, there are still many important research issues left unresolved. In this last section we briefly discuss some of these issues and provide some general research directions to address them. Some of the discussions presented here are based on our experience during the development of this dissertation. Some others are based on the inputs and comments from colleagues and active researchers from other institutions.

- One very important issue that has received little attention so far is in determining if a robust optimization is needed in the first place. We have assumed in our research that there is a trade-off between performance and robustness of a design.

However, it is not uncommon that robustness of a design increases as its performance increases. For a situation like this, robust optimization is not needed since the optimum design is already guaranteed to be also the most robust. To avoid wasting time and resources, we need some sort of indicators that can tell us from the beginning if the performance vs. robustness trade-off exists. The gradient of a function may be such an indicator. If the gradient of a function is monotonically decreasing with the function's value, then as the function is minimized the gradient is minimized as well, so there is no performance-robustness trade-off. Other inherent properties, such as the concavity or modality of the function, may also indicate such a trade-off.

- One shortcoming of our robust optimization method is that it is conservative. This is because we have used the worst-case estimate of the sensitivity region of a design as a measure of the overall robustness of the design. If we can incorporate those portions of the sensitivity region that are not included in the worst-case estimate into our robustness calculations, we can obtain a more accurate description of the design's robustness. An experiment-based regression analysis potentially can be used to numerically approximate the sensitivity region of a design so that we have the entire region as a robustness measure, and not just the worst-case region.
- In this dissertation, we assume that parameter variations are continuous. In many engineering design problems, parameters such as temperature variations or dimensional tolerances are continuous. However, in some cases the variations might be discrete, and one wants to find a design alternative that is optimum and

robust over a range of discrete scenarios. Examples of discrete variations include changes in material type, or number of teeth in a gear for a power tool. Theoretically, the sensitivity set concept should still be applicable to the discrete variations case; however, the notion of a sensitivity region may no longer apply. The possibility that the parameter variations have both continuous and discrete elements should also be investigated.

- An important issue that has not been addressed in this dissertation is the fact that the notion of a robust design is a subjective matter. A design that is considered robust by one designer may not satisfy the robustness preference or requirements of another designer. A design that is considered robust for each designer in a group may not be robust enough for the group collectively. The topic of preferences and decision-making is a very active area of research by itself. Nevertheless, if we are somehow able to incorporate some of the preference capturing methods into our robust optimization method, it will make the method more practical.
- Practically speaking, design optimization should be fitted within an iterative and collaborative process with various disciplines in which the information regarding the design is constantly updated and improved after each iteration. The robust optimization method presented in this dissertation does not account for this collaborative model. Integration of our method with some sort of systematic design techniques might be beneficial to improve the applicability of the method to more practical and real-world problems.

REFERENCES

1. Apostol, T.M., 1957, *Mathematical Analysis: A Modern Approach to Advanced Calculus*, Addison-Wesley, Reading, MA.
2. Arakawa, M, and Yamakawa, H., 1998, "Robust Design Using Fuzzy Numbers," Proc. of the Design Technical Conf., DETC98/DAC-5803, Atlanta, GA, Sept 13-16.
3. Arora, J.S., 2001, *Introduction to Optimum Design*, McGraw-Hill, New York.
4. Badhrinath, K., and Rao, J. R., 1994, "Bi-Level Models for Optimum Designs which are Insensitive to Perturbations in Variables and Parameters," Advances in Design Automation, **69**(2), 15-23.
5. Balling, R.J., Free, J.C., and Parkinson, A.R., 1986, "Consideration of Worst-Case Manufacturing Tolerances in Design Optimization," Trans. of the ASME, Journal of Mech. Design, **108**, 438-441.
6. Belegundu, A.D., and Chandrupatla, T.R., 1999, *Optimization Concepts and Applications in Engineering*, Prentice Hall, Upper Saddle River, NJ.
7. Belegundu, A.D., and Zhang, S., 1992, "Robustness of Design Through Minimum Sensitivity," Trans. of the ASME, Journal of Mech. Design, **114**, 213-217.
8. Box, G., 1988, "Signal-to-Noise Ratios, Performance Criteria, and Transformations," Technometrics, **30**(1), 1-18.
9. Branke, J., 1998, "Creating Robust Solutions by Means of Evolutionary Algorithms," Proc. of the 5th Conf. On Parallel Problem Solving from Nature, 119-128, Amsterdam, The Netherlands, Sept 27-30.

10. Branke, J., 2001, "Reducing the Sampling Variance when Searching for Robust Solutions," Proc. of the Genetic and Evolutionary Computation Conf., 235-242, San Francisco, CA, July 7-11.
11. Chandu, S.V.L., and Grandhi, R.V., 1995, "General Purpose Procedure for Reliability Based Structural Optimization Under Parametric Uncertainties," Advances in Engineering Software, **23**, 7-14.
12. Charnes, A., and Cooper, W.W., 1959, "Chance-Constrained Programming," Management Science, **6**(1), 73-79.
13. Charnes, A., and Cooper, W.W., 1963, "Deterministic Equivalents for Optimizing and Satisficing Under Chance Constraints," Operations Research, **11**, 18-39.
14. Chen, W., Allen, J.K., Tsui, K.L., and Mistree, F., 1996, "A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors," Trans. of the ASME, Journal of Mech. Design, **118**, 478-485.
15. Chen, W., and Yuan, C., 1999, "A Probabilistic-Based Design Model for Achieving Flexibility in Design," Trans. of the ASME, Journal of Mech. Design, **121**, 77-83.
16. Chen, W., Sahai, A., Messac, A., and Sundararaj, G.J., 2000, "Exploration of the Effectiveness of Physical Programming in Robust Design," Trans. of the ASME, Journal of Mech. Design, **122**, 155-163.
17. Choi, K.K., and Youn, B.D., 2002, "On Probabilistic Approaches for Reliability-Based Design Optimization (RBDO)," AIAA-2002-5472, Proc. of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, Sept.4-6.

18. Choi, K.K., Tu, J., and Park, Y.H., 2001, "Extensions of Design Potential Concept for Reliability–Based Design Optimization to Nonsmooth and Extreme Cases," Structural and Multidisciplinary Optimization, **22**, 335-350.
19. Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B., 2002, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, Boston.
20. Deb, K., 1991, "Optimal Design of a Welded Beam via Genetic Algorithms," AIAA Journal, **29**(11), 2013-2015.
21. Deb, K., 2001, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd, New York.
22. Du, X., and Chen, W., 2000, "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design," Trans. of the ASME, Journal of Mech. Design, **122**, 385-394.
23. Elsayed, E.A., and Chen, A., 1993, "Optimal Levels of Process Parameters for Products with Multiple Characteristics," Int. Journal of Production Research, **31**(5), 1117-1132.
24. Fernandez, F.R., Nickel, S., Puerto, J., and Rodriguez-Chia, A.M., 2001, "Robustness in the Pareto-Solutions for the Multi-Criteria Minisum Location Problem," Journal of Multi-Criteria Decision Analysis, **10**, 191-203.
25. Fiacco, A.V., 1983, *Introduction to Sensitivity and Stability Analysis in Non Linear Programming*, Vol 165 in *Mathematics in Science and Engineering*, Academic Press, New York.

26. Fonseca, C. M., and Fleming, P. J., 1993, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization," Proc. of the 5th Int. Conf. on Genetic Algorithms, 416-423, Urbana-Champaign, IL, July 17-21.
27. Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
28. Haug, E.J., and Arora, J.S., 1979, *Applied Optimal Design*, Wiley-Interscience Publication, New York.
29. Hirokawa, N., and Fujita, K., 2002, "Mini-max Type Formulation of Strict Robust Design Optimization Under Correlative Variation," Proc. of Design Engineering Technical Conf., DETC2002/DAC-34041, Montreal, Canada, Sept 29-Oct 2.
30. Hwang, K.H., Lee, K.W., and Park, G.J., 2001, "Robust Optimization of an Automobile Rearview Mirror for Vibration Reduction," Structural Multidisciplinary Optimization, **21**, 300-308.
31. Jung, D.H., and Lee, B.C., 2002, "Development of a Simple and Efficient Method for Robust Optimization," Int. Journal for Numerical Methods in Engineering, **23**, 2201-2215.
32. Kackar, R.N., 1985, "Off-Line Quality Control, Parameter Design and the Taguchi Method (with discussions)," Journal of Quality Technology, **17**, 175-209.
33. Kalsi, M., Hacker, K., and Lewis, K., 2001, "A Comprehensive Robust Design Approach for Decision Trade-offs in Complex Systems Design," Trans. of the ASME, Journal of Mech. Design, **123**, 1-10.
34. Kunjur, A., and Krishnamurty, S., 1997, "A Robust Multi-Criteria Optimization Approach," Mechanisms and Machines Theory, **32**(7), 797-810.

35. Kurapati, A., Azarm, S., and Wu, J., 2002, "Constraint Handling Improvements for Multi-Objective Genetic Algorithms," Structural and Multidisciplinary Optimization, **23**(3), 204-213.

36. Lee, K., and Lee, T.H., 2001, "Fuzzy Multi-Objective Optimization of an Automotive Seat Using Response Surface Model and Reliability Method," Proc. of the 4th World Congress of Structural and Multidisciplinary Optimization (WCSMO), Dailan, China, June 4-8.

37. Leon, R.V., Shoemaker, A.C., and Kacker, R.N., 1987, "Performance Measures Independent of Adjustment," Technometrics, **29**(3), 253-265.

38. Messac, A., and Yahaya, I.A., 2002, "Multiobjective Robust Design Using Physical Programming," Structural and Multidisciplinary Optimization, **23**, 357-371.

39. Miettinen, K. M., 1999, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston.

40. Narayanan, S., and Azarm, S., 1999, "On Improving Multiobjective Genetic Algorithms for Design Optimization," Structural Optimization, **18**, 146-155.

41. Otto, K.N., and Antonsson, E.K., 1993, "Tuning Parameters in Engineering Design," Trans. of the ASME, Journal of Mech. Design, **115**, 14-19.

42. Palli, N., Azarm, A., McCluskey, P., and Sundararajan, R., 1998, "An Interactive Multistage ε-Inequality Constraint Method for Multiple Objectives Decision Making," Trans. of the ASME, Journal of Mech. Design, **120**, 678-686.

43. Papalambros, P., and Wilde, D.J., 1980, "Regional Monotonicity in Optimum Design," Trans. of the ASME, Journal of Mech. Design, **102**(3), 497-500.

44. Papalambros, P.Y., and Wilde, D.J., 2000, *Principles of Optimal Design: Modeling and Computation*, 2nd ed, Cambridge University Press, New York.

45. Parkinson, A., Sorensen, C., and Pourhassan, N., 1993, "A General Approach for Robust Optimal Design," Trans. of the ASME, Journal of Mech. Design, **115**, 74-80.

46. Parkinson, D.B., 1998, "Simulated Variance Optimization for Robust Design," Quality Reliability Engineering International, **14**, 15-21.

47. Parkinson, D.B., 2000, "The Application of a Robust Design Method to Tolerancing," Trans. of the ASME, Journal of Mech. Design, **122**, 149-154.

48. Phadke, M.S., 1989, *Quality Engineering Using Robust Design*, Prentice Hall, Englewood Cliffs, NJ.

49. Pignatiello, J.J., Jr., 1993, "Strategies for Robust Multiresponse Quality Engineering," IIE Transactions, **25**(3), 5-14.

50. Pignatiello, J.J., Jr., and Ramberg, J.S., 1985, "Discussion of Off-Line Quality Control, Parameter Design, and the Taguchi Method," Journal of Quality Technology, **17**, 198-206.

51. Pignatiello, J.J., Jr., and Ramberg, J.S., 1987, "Discussion of Performance Measures Independent of Adjustment," Technometrics, **29**, 274-277.

52. Ragsdell, K. M., and Phillips, D. T., 1976, "Optimal Design of a Class of Welded Structures Using Geometric Programming," Journal of Engineering for Industries, Series B, **98**(3), 1021-1025.

53. Ramakrishnan, B., and Rao, S.S., 1996, "A General Loss Function Based Optimization Procedure for Robust Design," Engineering Optimization, **25**, 255-276.

54. Rao, S.S., 1984, "Multiobjective Optimization in Structural Design with Uncertain Parameters and Stochastic Processes," AIAA Journal, **22**(11), 1670-1678.

55. Rao, S.S., and Cao, L., 2002, "Optimum Design of Mechanical Systems Involving Interval Parameters," Trans. of the ASME, Journal of Mech. Design, **124**, 465-472.

56. Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M., 1983, *Engineering Optimization Methods and Applications*, Wiley-Interscience Publication, New York.

57. Schmit, L.A., 1960, "Structural Design by Systematic Synthesis," Proc. of the 2nd ASCE Conf. On Electronic Computations, 105-122, Pittsburgh, PA, Sept 8-9.

58. Shelokar, P.S., Jayaraman, V.K., and Kulkarni, B.D., 2002, "Ant Algorithm for Single and Multiobjective Reliability Optimization Problem," Quality and Reliability Engineering Int., **18**, 497-514.

59. Siddall, J.N., 1982, Optimal Engineering Design: Principles and Applications, Marcel Dekker, New York.

60. Simpsons, T.W., Allen, J.K., Mistree, F., and Chen, W., 1997, "Designing Ranged Sets of Top-Level Design Specifications for a Family of Aircraft: An Application of Design Capability Indices," SAE World Aviation Congress and Exposition, AIAA-97-5513, Anaheim, CA, October 13-16.

61. Spotts, M.F., 1953, *Design of Machine Elements*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ.

62. Srinivas, N., and Deb, K., 1995, "Multiobjective Function Optimization Using Nondominated Sorting Genetic Algorithms," Evolutionary Computation Journal, **2**(3), 221-248.

63. Suhir, E., 1987, "Die Attachment Design and Its Influence on Thermal Stresses in Die and the Attachment," Proc. of the 37th Electronic Components Conf., 508-517, Boston, MA, May 11-13.

64. Sun, P.F., Arora, J.S., and Haug, E.J., 1975, "Fail-Safe Optimal Design of Structures," Technical Report No. 19, Department of Civil and Environmental Engineering, The University of Iowa, Iowa City.

65. Sundaresan, S., Ishii, K., and Houser, D., 1992, "Design Optimization for Robustness Using Performance Simulation Programs," Engineering Optimization, **20**, 163-178.

66. Sundaresan, S., Ishii, K., and Houser, D.R., 1993, "A Robust Optimization Procedure with Variations on Design Variables and Constraints," Advances in Design Automation, **69**(1), 379-386.

67. Taguchi, G., 1978, "Performance Analysis Design," International Journal of Production Research, **16**, 521-530.

68. Taguchi, G., and Phadke, M.S., 1984, "Quality Engineering Through Design Optimization," Proc. of the IEEE Globecom Conference, 1106-1113, Atlanta, GA, Nov 26-29.

69. Tsui, K.L., 1999, "Robust Design Optimization for Multiple Characteristics Problems," Int. Journal of Production Research, **37**(2), 433-445.

70. Tsutsui, S., 1999, "A Comparative Study on the Effects of Adding Perturbations to Phenotypic Parameters in Genetic Algorithms with a Robust Solution Searching Scheme," Proc. of IEEE Systems, Man, and Cybernetics Conf., 585-591, Tokyo, Japan, Oct 12-15.

71. Tsutsui, S., and Gosh, A., 1997, "Genetic Algorithms with a Robust Solution Searching Scheme," IEEE Trans. on Evolutionary Computation, **1**(3), 201-208.

72. Tu, J., Choi, K.K., and Park, Y.H., 1999, "A New Study on Reliability-Based Design Optimization," Trans. of ASME, Journal of Mech. Design, **121**, 557-564.

73. Van Veldhuizen, D. A., and Lamont, G. B., 1998, "Multiobjective evolutionary algorithm research: A history and analysis." Technical Report TR-98-03, Air Force Institute of Technology, Wright-Patterson AFB, OH.

74. Wahl, A.M., 1963, *Mechanical Springs*, 2nd ed., McGraw-Hill, New York.

75. Wang, H.T., Liu, Z.J., Chen, S.X., and Yang, J.P., 1999, "Application of Taguchi Method to Robust Design of BLDC Motor Performance," IEEE Transactions on Magnetics, **35**(5), 3700-3702.

76. Wu, Y.T., Millwater, H.R., and Cruse, T.A., 1990, "Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions," AIAA Journal, **28**(9), 1663-1669.

77. Youn, B.D., Choi, K.K., and Park, Y.H., 2003, "Hybrid Analysis Method for Reliability-Based Design Optimization," Trans. of ASME, Journal of Mech. Design, **125**, 221-232.

78. Yu, J.C., and Ishii, K., 1994, "Robust Design by Matching the Design with Manufacturing Variation Patterns," Advances in Design Automation, **69**(2), 7-14.

79. Yu, J.C., and Ishii, K., 1998, "Design for Robustness Based on Manufacturing Variation Patters," Trans. of the ASME, Journal of Mech. Design, **120**, 196-202.

80. Zhu, J., and Ting, K.L., 2001, "Performance Distribution Analysis and Robust Design," Trans. of the ASME, Journal of Mech. Design, **123**, 11-17.