2014-12-01

A Hybrid Method for Sensitivity Optimization With Application to Radio-Frequency Product Design

Abraham Lee
Brigham Young University - Provo

A Hybrid Method for Sensitivity Optimization with Application to Radio-Frequency Product Design

Abraham D. Lee

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Alan R. Parkinson, Chair
Christopher A. Mattson
John S. Lawson

Department of Mechanical Engineering

Brigham Young University

December 2014

ABSTRACT

A Hybrid Method for Sensitivity Optimization with
Application to Radio-Frequency Product Design

Abraham D. Lee
Department of Mechanical Engineering, BYU
Master of Science

A method for performing robust optimal design that combines the efficiency of experimental designs and the accuracy of nonlinear programming (NLP) has been developed, called *Search-and-Zoom*. Two case studies from the RF and communications industry, a high-frequency micro-strip band-pass filter (BPF) and a rectangular, directional patch antenna, were used to show that sensitivity optimization could be effectively performed in this industry and to compare the computational efficiency of traditional NLP methods (using *fmincon* solver in MATLAB R2013a) and they hybrid method *Search-and-Zoom*. The sensitivity of the BPF's S_{11} response was reduced from 0.0666 at the (non-robust) nominal optimum to 0.0182 at the sensitivity optimum. Feasibility in the design was improved by reducing the likelihood of violating constraints from 20% to nearly 0%, assuming RSS (i.e., normally-distributed) input tolerances and from 40% to nearly 0%, assuming WC (i.e., uniformly-distributed) input tolerances. The sensitivity of the patch antenna's S_{11} function was also improved from 0.0208 at the nominal optimum to 0.0115 at the sensitivity optimum. Feasibility at the sensitivity optimum was estimated to be 100%, and thus did not need to be improved. In both cases, the computation effort to reach the sensitivity optima, as well as the sensitivity optima with RSS and WC feasibility robustness, was reduced by more than 80% (average) by using *Search-and-Zoom*, compared to the NLP solver.

Keywords: NLP, Monte Carlo, feasibility robustness, sensitivity optimization, sensitivity robustness, Taguchi method, tolerance, orthogonal array, Search-and-Zoom

ACKNOWLEDGEMENTS

It goes without saying that I couldn't have completed this research on my own. To that end, I'd like to thank the people who assisted me in any way and gave me encouragement during the work of this thesis.

I thank Robert Male and Brady Davies at L-3 Communications who provided me with an industrial opportunity which could directly benefit from this research. I thank my committee chair, Dr. Alan Parkinson for his wisdom, insight, and unwavering encouragement. I also thank my committee members, Dr. Mattson and Dr. Lawson for their insight and advice.

I also thank my wonderful wife and children for their patience, love, and support through these long years of work. And finally, I thank my Heavenly Father for His mercy, patience, and endless love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

In the communications industry, engineers seek to develop high performance Radio Frequency (RF) equipment that pushes technology limits in areas such as antenna range, pointing accuracy and bandwidth (capacity of the data stream). These devices need to work in a wide range of environmental conditions, sometimes including military conditions. Functional requirements push technology to the edges of its capability, so understanding the limitations of a technology and the related manufacturing processes drives this work. A major challenge is dealing with the different sources of variation that creep into a design at its various life stages. For engineers and designers, the process for accounting for this variation, and reducing its effect, is called *robust design*.

When we speak of *robustness*, this can have different meanings to different people. In this thesis, we will define it as *a design's ability to function as intended in the presence of uncontrollable variation*. Most variation is controllable to some degree, but eventually it becomes too expensive or simply impossible to control further. For example, changing from one manufacturing operation to another may allow the designer to specify tighter tolerances, reducing geometric variation. However, further tightening might require a new process and/or manufacturing machine altogether, and this may not be possible because either it is too expensive or another machine with more precision simply doesn't exist. Another example is the variation of material properties. When a company procures a batch of aluminum for machining purposes, there is no guarantee that the each batch will have exactly the same

properties, such as elastic modulus or yield strength. They may be close, but controlling the exact make-up of the aluminum is often beyond the capabilities of the manufacturing processes that produce the aluminum.

In engineering, there are two kinds of robustness that are of usually of most interest: *feasibility robustness* and *sensitivity robustness*. They are related, but have important differences and goals.

1.1 Feasibility Robustness

When a design has *feasibility robustness*, it means that all of the designs constraints or requirements will remain satisfied even when subjected to variation. Most engineers make their designs robust in this way by performing *worst-case analysis*. This involves identifying *worst-case conditions* (such as maximum material condition (MMC) and least material condition (LMC) in an assembly), and then combining them in such a way to give the most extreme case that could possibly happen. If the extreme combinations do not violate the design requirements, then the design is considered acceptable with no further regard for the variation within those limits. At this level, since these are the extreme conditions, no design is ever expected to fail. However, this approach to feasibility robustness can have detrimental financial consequences, usually making the product more expensive than is necessary.

Applying a common statistical approach to characterize the expected variation almost always yields a less stringent design that is often easier to make and cheaper to produce. In this thesis, we will use a reasonable statistical approach for variation analysis (also called *error propagation* or *uncertainty analysis*). More details regarding the usage and background of this approach will be given in Chapter 2.

1.2 Sensitivity Robustness

Assume that an antenna has been made that has a known maximum broadcasting range that can vary from 10 to 50 miles. To the engineer, who was focused on making a design that is able to broadcast a minimum of 10 miles, this may seem acceptable since the design requirement is met in all cases, but to a soldier who needs to transmit important information to a UAV for re-transmission, the antenna that only is capable of broadcasting 10 miles may appear defective compared to the antenna that can broadcast 50 miles. Comparably, it would be more desirable that ALL antennas of this design have a more consistent range of 25 ± 3 miles rather than 30 ± 20 miles because the perceived quality is better and the user can depend on the product's specified capabilities.

If it is discovered that a product's performance or assembly variation is too excessive, then the engineer may need to consider designing for *sensitivity robustness*. This kind of robustness is more concerned with reducing the influence of "input" variations to performance variations. This can be done in a variety of ways, but ultimately, all methods focus on identifying a "location" in the design space where the derivatives (*sensitivities*) are small. If, mathematically, the performance or assembly stack-up is relatively linear, then sensitivity robustness may not be possible. However, many engineering performance metrics are not linear in nature and therefore can likely benefit from this design practice. Not only does this improve product consistency, but it may even allow for the increase of controllable tolerances, which usually makes the product cheaper and easier to produce.

The greatest benefit of feasibility and sensitivity robustness comes from designing for both to exist. The result is a design that is minimally affected by input variation while still satisfying constraints when subject to variation. Unfortunately, this can come at a potentially significant computational cost when using simulation tools to predict product performance. For

engineers, using numerical models that are not analytical in nature is quite common. Finite element analysis (FEA) and computational fluid dynamics (CFD) are well known examples of this kind of model. With the advent of more powerful computers, simulation time has certainly decreased, but this has encouraged engineers and designers to consider more realistic models which, in turn, increases the complexity of their analysis. Using these models in optimization routines presents challenges that have to be weighed between schedule, cost, and performance. However, since robustness is often most effectively realized using optimization methods, it either gets neglected altogether, or a simpler analysis involving only one or two extreme cases instead that helps serve to envelope all other cases.

1.3 Research Objectives and Thesis Outline

There are two main purposes of this thesis. The first is to show how sensitivity optimization may be done in the communications/RF industry. The second is to address the important issue related to how efficiently an engineer or designer can perform robust design on complex, non-analytical models. Although the application focus will be on communications-based designs, modeled numerically, it is equally applicable to other engineering disciplines.

To reach the goal of a more efficient robust design methodology, this thesis will explore two main areas that are used effectively for this purpose: experimental design methods (e.g., Taguchi methods) and nonlinear programming (NLP) or optimization methods. Each has its strengths and weaknesses for the kinds of problems they can solve. Experimental methods work very well for non-continuous, or discrete, variable options (like choosing to use steel or aluminum). They are also usually quite efficient, with a minimum amount of experiments. Then, through statistical techniques, approximation models are used to predict optimal variable combinations that provide the most robustness. However, for many common engineering

problems, some experimental methods don't provide the modeling flexibility to account for design constraints. NLP methods, on the other hand, are known for their flexibility and allow for the analysis of virtually any kind of model. This provides an excellent framework for complex design optimization. Unfortunately, the flexibility and accuracy of the underlying mathematical methods can come at a potentially high computational cost, particularly when trying to design for sensitivity robustness. More details related to these two methods will be explained in Chapter 2.

To overcome the weaknesses of these two methods, we propose a hybrid methodology that combines the efficiencies of experimental methods with the flexibility and accuracy of NLP methods. The details of the development of this method will be described in Chapter 3. In Chapter 4, the practical use of the hybrid method will be demonstrated on two relevant case studies: a micro-strip band-pass filter and a PCB-mounted patch antenna. In these cases, we will compare the efficiency of traditional NLP methods with this hybrid method by seeking a nominal optimal design (no robustness considered), and then combinations of feasibility and sensitivity robustness. The results of these two cases show the benefits for its usage in robust design, particularly when applied to sensitivity robustness. Although any kind of design variation can theoretically be included in this analysis (with proper quantification), we will focus on the sole effects of geometric tolerances on design performance. Chapter 5 will then offer some concluding remarks and recommendations for further work. The end result of this thesis is a method that gives designers and engineers the ability to perform robust design in a way that might not otherwise be possible, given the complexity of the design requirements and the associated time-cost of the design process.

CHAPTER 2 LITERATURE REVIEW

In this chapter, we will briefly discuss two common approaches that designers use to achieve design robustness. We will first discuss a kind of experimental method, called *Taguchi methods,* followed by a discussion of a computational method, called *nonlinear programming* (NLP).

2.1 Taguchi Methods

In order to understand the hybrid method that will be developed in Chapter (3), we must first understand how experimental methods, commonly called *Taguchi methods*, and nonlinear programming (NLP), or *optimization* techniques, are used to do robust design.

2.1.1 Basic Concepts

In Taguchi methods, pioneered by Genichi Taguchi (1987), there are two kinds of factors that help define the system that we are interested in. The first kind of factor is called a *control factor* (CF), for which we will use the notation x. These factors are variables that are input to the system that the engineer can specify freely. Each control factor can take multiple values, called *levels*, which can be continuous or discrete. An example of a control factor might be the diameter of a pipe or the thickness of a beam. The engineer's main job is to determine an appropriate level for each control factor that will allow the design to meet some kind of performance goal.

The second kind of factor is also an input to the system, called a *noise factor* (NF), which we will denote as z. Noise factors are present in all systems and cause the system's performance to deviate from the desired value. This deviation is often the cause of design failure and product unreliability. Input factors may be put in this category when they cannot be controlled directly by the designer or are too expensive to control. In Taguchi methods, part of the intent of the experiments is to understand which noise factors cause the variability in system performance, and how much, so extra efforts are made to control them during the experiments. Common examples of noise factors include manufacturing tolerances, environmental effects, and user error.

The output, y, of the system is called the *response*. There may be multiple system responses of interest to the designer. These will often have pre-specified requirements that the designer is trying to meet and will be used to determine the *quality* of the design. In Taguchi methods, these are called *quality characteristics* (Phadke (1989)). Examples of system responses include weight, gain, cost, speed, strength, and electrical resistance—to name just a few.

Taguchi methods have been used in many industries, but most notably in the improvement of manufacturing processes. An early example is recounted by Phadke (1989) where Taguchi was asked to help the Ina Tile Company because of excess variability in the final dimensions of the tiles it produced. An analysis of the process showed a non-uniform temperature distribution within the kiln. There were two options for solving this problem: 1) redesign the kiln for more uniform temperature, which would be very expensive, or 2) use inexpensive experiments to identify process parameters that would allow the tiles to be less sensitive to the temperature's non-uniformity. Following the second route, Taguchi found that increasing the lime content from 1% to 5% would reduce dimensional variation. Thus, the problem of non-uniform tile size was solved by minimizing the effect of the non-uniform

Table 2.1 - The L-4 orthogonal array.

Trial	Factor A	Factor B	Factor C
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

temperature distribution without changing the kiln design at all. This particular change also turned out to be the least expensive to implement.

The purpose of the experiments in Taguchi methods is two-fold. The first is to determine which control factor levels result in the desired response. The second is to determine which control factor levels reduce variability in the response. Historically, Taguchi methods have been used where the system being analyzed had no analytical model (i.e., there was no $f(x, z) = y$ to relate x, z, and y). The only option was to run a set of experiments, called *trial conditions* or simply *trials*, and construct a statistical model rather than an analytical one. With this statistical model, designers would then hope to be able to determine a robust design. Common models take into consideration first-order (linear) influences of the input factors:

$$y = \beta_1 x + \beta_2 z + \varepsilon \tag{2.1}$$

where β_i are the main effects of x and z, and ε is the error in the statistical model. Other models include second-order (quadratic) influences:

$$y = \beta_1 x + \beta_2 z + \beta_3 x^2 + \beta_4 x z + \beta_5 z^2 + \varepsilon \tag{2.2}$$

The effectiveness of the statistical model depends upon how the various input factor levels are combined and how many experiments are performed. In order to get a "good" model, the designer will run these experiments at carefully chosen values, arranged in a *design matrix*.

For example, the L-4 orthogonal array in Table 2.1 would be appropriate for the linear model in Equation 2.1.

Taguchi methods focus on constructing a design matrix that has the property of *orthogonality* (strength 2), which means you can take any pair of columns and you will see that all combinations of factor levels occur exactly one time. For example, if we look at the columns for Factor *A* and Factor *B* in Table (2.1), we see that (1, 1), (1, 2), (2, 1), and (2, 2) are all the possible ordered pairs of the two element set and each appears exactly once. For an array to be orthogonal, this relation must hold for all column combinations. Orthogonality is a *balancing property* which makes it possible for the designer to mathematically estimate the individual influences of each of the input factors (called *main effects*), and sometimes *interactions* between input factors (requires strength 3 or 4 design). In robust design, it is often important to estimate the interactions between control factors and noise factors and find control factor levels that minimize these interactions.

The other important property of design matrices that Taguchi methods exploit is the minimization of the number of experiments needed to estimate these effects. In statistical experimental design, there are three common kinds of design matrices:

1. **Full-factorial designs**: An experiment is done for all possible factor level combinations. This allows for all CF main effects and all CF-CF interaction effects to be statistically estimated.
2. **Fractional-factorial designs**: A sub-set of trials from a full-factorial is selected. This is done to estimate the CF main effects and only some, if any, CF-CF interaction effects, depending on the resolution of the design matrix.
3. **Response surface designs**: A more complex (and typically longer) design that allows for estimation of first- and second-order CF effects, including interaction effects.

For Taguchi methods, the most common choice are fractional-factorial designs that are still orthogonal because they minimize the number of experiments while allowing at least the first-order effects to be estimated. Taguchi most often referred to these as *orthogonal arrays* (OA).

Since it is important to understand the CF-NF interactions for robust design, Taguchi methods will *cross* a control factor OA (the *inner array*) with a noise factor array (the *outer array*, not necessarily an OA). This means that each trial condition of the inner array is evaluated at each *noise condition* in the outer array, which provides full information about CF-NF interactions.

Once we have a model, we use it to search for values of control factors in order to achieve the two objectives of Taguchi methods: reach some desired response while minimizing variability around the response. To make the optimization easier, we combine these two objectives into a single statistic, called the *Mean Squared Deviation* or *MSD*, often referred to by Taguchi as the *Loss Function*:

$$MSD = (\mu - y_t)^2 + \sigma^2 \qquad (2.3)$$

This concept is convenient because it allows for great flexibility in its definition, depending on what kind of objective y_t we are trying to achieve: *stay on target*, *minimize*, or *maximize*. Mathematically, we can tailor the definition of *MSD* in the following ways. If we wish to keep the response at some *target value* while minimizing variability, we choose CF settings that minimize Equation 2.4, which is mathematically equivalent to Equation 2.3 for large n. If we wish to *minimize* the response (i.e., $y_t = 0$) while minimizing variability, we choose CF settings that minimize Equation 2.5. And finally, if we want to *maximize* the response while minimizing variability, we choose CF settings that minimize Equation 2.6.

$$MSD_T = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_t)^2 \tag{2.4}$$

$$MSD_S = \frac{1}{n}\sum_{i=1}^{n} y_i^2 \tag{2.5}$$

$$MSD_L = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{y_i^2} \tag{2.6}$$

One downside to the basic definitions for MSD_S and MSD_L is that they do not provide natural support for some kinds of responses. For example, if a response y can have both positive and negative values, and we want a value that is as close to -∞ as possible, the above definition for MSD_S doesn't behave as we would expect—it actually penalizes responses closer to -∞. To remedy this, Ku (1998) offers an alternative formulation that works more like the traditional forms of *minimize* (i.e., as close to -∞ as possible) and *maximize* (i.e., as close to +∞ as possible), which also make them more useful in modern computer algorithms:

$$MSD_S = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(1+y_i^2), & y_i > 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{1+y_i^2}\right), & y_i \le 0 \end{cases} \tag{2.7}$$

$$MSD_L = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{1+y_i^2}\right), & y_i > 0 \\ \frac{1}{n}\sum_{i=1}^{n}(1+y_i^2), & y_i \le 0 \end{cases} \tag{2.8}$$

Once the respective *MSD* is calculated, this value is then used to calculate the *signal-to-noise ratio* (*S*/*N*), given by Taguchi (1987), which is measured in decibels (dB):

$$S/N = -10\log_{10}(MSD) \tag{2.9}$$

When S/N is maximized, the corresponding MSD is minimized, which means that the difference between the desired value of the response y_t and the actual value is minimized and the variation about y_t due to the noise is also minimized.

2.1.2 A Simple Example

We now illustrate how Taguchi methods can be used to optimize, for example, a contrived machining process. The goal here is to improve (i.e., minimize) the surface finish of the metal being worked, in microns, by selecting optimal settings for three factors, each with two levels:

Table 2.2 - Control factor levels for example manufacturing process to be optimized.

Factor	Level 1	Level 2
A: Tool type	High Carbon	Carbide Tip
B: Cutting speed	1500 rpm	2000 rpm
C: Feed rate	2 mm/sec	5 mm/sec

The smallest OA we can use for three 2-level factors is the L-4 array found in Table 2.1. Each trial condition is carried out, repeated three times each to estimate the variability in the manufacturing process (the noise factor), using the level combinations indicated. Since we want the surface finish to be *as small as possible*, we use Equation 2.5 to calculate MSD_S and we end up with the et of results in Table 2.3.

In order to determine the optimal level combination, we calculate the S/N averages at each of the respective factor levels. For example, the level averages for factor A (tool type) are calculated as follows:

Table 2.3 – Experimental results of example manufacturing process.

	Factor			Surface Finish [micron]				
Trial #	A	B	C	Rep.1	Rep. 2	Rep. 3	MSD	S/N [dB]
1	1	1	1	15	16	20	293.6667	-24.6785
2	1	2	2	13	11	12	144.6667	-21.6037
3	2	1	2	13	17	18	260.6667	-24.1609
4	2	2	1	22	19	19	402	-26.0423

$$A_1 = \frac{(-24.6785)+(-21.6037)}{2}$$

$$= -23.1411$$

$$A_2 = \frac{(-24.1609)+(-26.0423)}{2}$$

$$= -25.1016$$

The factor level averages are summarized in Table 2.4:

Table 2.4 – Factor level averages of S/N for example manufacturing process.

Level	*A*	*B*	*C*
1	−23.1411	−24.4197	−25.3604
2	−25.1016	−23.8230	−22.8823

To determine the optimal configuration of levels from each factor, we simply select the factor levels that have the largest average S/N (i.e., levels A_1, B_2, and C_2), as shown in Figure 2.1. This assumes that the differences between factor levels is not due to chance, that there is no difference in the cost of each level, and that there are no interactions between A, B, and C.

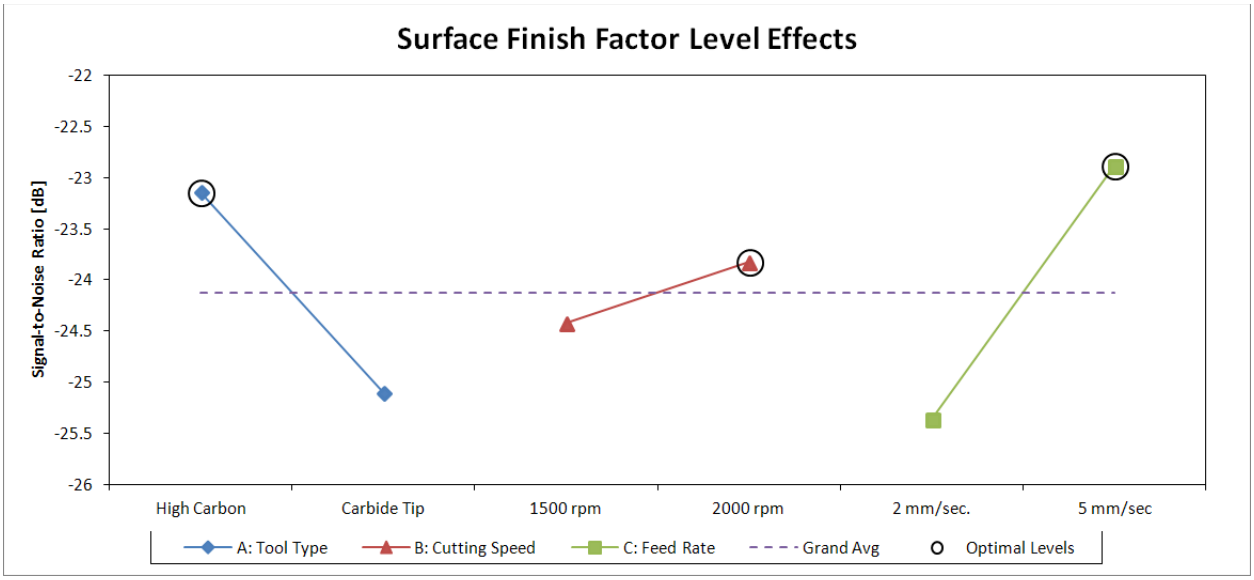

Figure 2.1 - Main effects plot for example manufacturing process factors, with optimal levels indicated.

Even though we didn't construct an experiment for every possible combination of the factor levels, we can estimate the expected performance in surface finish at the optimum condition. To do this, we need the grand average of all four trial conditions, T = −24.1213 (shown as the horizontal dashed line in Figure 2.1). Then, assuming an additive model, the predicted optimum S/N is calculated as

$$\begin{aligned} Y_{opt} &= T+(A_1-T)+(B_2-T)+(C_2-T) \\ &= (-24.1213)+[(-23.1411)-(-24.1213)] \\ &\quad +[(-23.8230)-(-24.1213)] \\ &\quad +[(-22.8823)-(-24.1213)] \\ &= -21.6037 \end{aligned}$$

In a real situation, a confirmation experiment (or multiple experiments) should be done at the optimal factor levels to verify the prediction. At this point, the design would be considered *optimized.*

It is sometimes found that certain factors exhibit a strong influence on the mean value of the response while having a weak influence on S/N, and vice versa. When this is the case, Taguchi recommends a two-step method for system optimization:

1. **Maximize *S/N***: In this step, we choose factor levels that maximize S/N while ignoring the target objective y_t.
2. **Adjust the mean *on target***: During this step, we utilize those factors that have less effect on S/N and more effect on the mean to adjust the mean to be closer to its target objective y_t without changing S/N. These factors are called *adjustment factors*.

2.1.3 Multiple Objectives: The Desirability Function

In real-world design problems, we may find that there is more than a single objective to be optimized. For example, a mechanical engineer might want to minimize the stresses in a truss, but is also concerned with minimizing the total weight of the truss. To do this in experimental methods, we can utilize *desirability functions*, as explained by Derringer and Suich (1980). Desirability functions are used to translate the designer's intent of what is and isn't acceptable in the response metrics and also how that acceptability changes between them. In other words, each objective is given a range of acceptable values that allows the designer to find a suitable compromise when any objectives compete with each other.

In mathematical terms, we define a design's desirability as follows. For some response y_i, a desirability function $d_i(y_i)$ assigns a value between 0 and 1 to the possible values of y_i. When d_i=0, the corresponding y_i is considered completely unacceptable. When d_i=1, the corresponding y_i is considered to be completely acceptable. Then, to get the overall design's desirability, D, for k objectives, we take each objective's individual desirability d_i and combine them using a geometric mean:

$$D=\left(\prod_{i=1}^{k} d_i\right)^{\frac{1}{k}} \tag{2.10}$$

From Equation 2.10, we observe that when any $d_i = 0$, then the overall $D = 0$ as well, implying that if any of the k response functions is *completely undesirable*, the whole design is also.

Like the MSD equations, Derringer and Suich classify the desirability functions into three classes: *nominal-is-best* (NTB), *smaller-the-better* (STB), and *larger-is-better* (LTB). Each is defined by at least two of the following values: a lower bound L, a target value T, and an upper bound U, as shown in Figure 2.2.

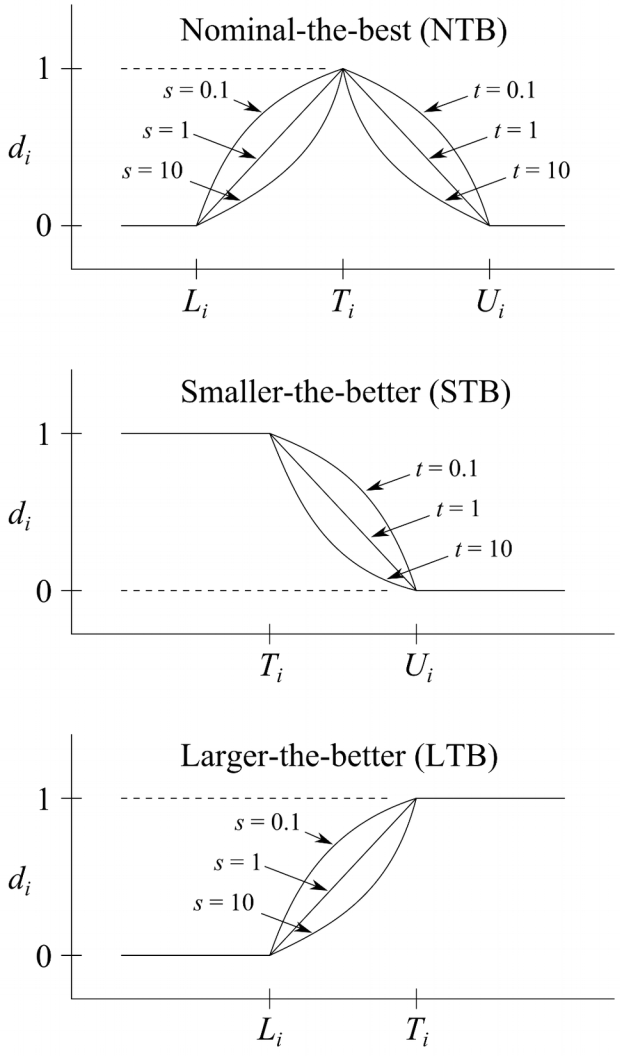

Figure 2.2 - Graphical representation of desirability functions.

If we want to achieve some target value (a NTB kind of quality characteristic), then the desirability function is:

$$d_{NTB}=\begin{cases}\left(\frac{y-L}{T-L}\right)^{s} & L\le y\le T\\ \left(\frac{y-U}{T-U}\right)^{t} & T\le y\le U\\ 0 & otherwise\end{cases} \tag{2.11}$$

with exponents s and t determining how important it is to hit the target value T (not to be confused with the grand average T in the Taguchi example above). For $s = t = 1$, d_i changes linearly towards T. For $s, t < 1$, the function is convex. For $s, t > 1$, the function is concave. If we want to minimize a response (STB), we define d as:

$$d_{STB}=\begin{cases}0 & y>U\\ \left(\frac{y-U}{T-U}\right)^{t} & T\le y\le U\\ 1 & y<T\end{cases} \tag{2.12}$$

with T denoting a small enough value to be acceptable. In contrast, if we want to maximize the response (LTB), then we define d as:

$$d_{LTB}=\begin{cases}0 & y<L\\ \left(\frac{y-L}{T-L}\right)^{s} & L\le y\le T\\ 1 & y>T\end{cases} \tag{2.13}$$

with T denoting a large enough value for y to be acceptable. A downside to these equations is that the designer must be able to provide appropriate values for L, T, and U, which may not be known. Wu and Hamada (2000) suggest a double-exponential formulation for NTB, STB, and LTB, shown in Equations 2.14 – 2.16, respectively:

$$d_{NTB}=\begin{cases}\exp\{-c_1|y-m^{\alpha}|\}, & -\infty<y\le m\\ \exp\{-c_2|y-m^{\alpha}|\}, & m\le y<\infty\end{cases} \tag{2.14}$$

$$d_{STB}=\exp\{-c|y-a^{\alpha}|\}, \quad a\le y<\infty \tag{2.15}$$

$$d_{LTB}=\begin{cases}1-\dfrac{\exp\{-c\,y^{\alpha}\}}{\exp\{-c\,L^{\alpha}\}}, & L<y\le\infty\\ 0, & y<L\end{cases} \tag{2.16}$$

where c is the scale constant of the desirability function. Wu (2008) notes that, in practical applications, L, U, and a can be treated as the lower specification limit (LSL), upper specification limit (USL) and 0, respectively, and m is the ideal target value for y.

2.1.4 Weaknesses of Taguchi Methods

In addition to manufacturing applications, Taguchi methods have also been used successfully in electronic circuit design, heat exchanger design, cash flow optimization, and many others. With so many benefits from utilizing Taguchi methods for robust design, we must ask the question, why wouldn't we? Although Taguchi methods are experimentally efficient and allow the simultaneous consideration of many more variables than other methods, there are some mathematical and statistical problems that arise from their use.

Interactions are part of the real world. It is often criticized that Taguchi methods tend to neglect the CF-CF interactions because the usage of the more common OA doesn't provide enough information from the experiments to estimate them. In statistical terms, this means that the CF-CF interactions are *confounded* with the CF main effects.

Critics of Taguchi methods also tout the inefficiency of using outer arrays to quantify the noise conditions. Crossing the inner array (with N_x trial conditions) with the outer array (with N_z noise conditions) requires a total number of $N = N_xN_z$ experiments. N can be

prohibitively large, especially when dealing with physical experiments. To be more efficient, Buyske (2000) argued that we should first perform a *screening design* for the purpose of identifying a subset of noise factors that exhibit a more significant contribution to the overall system variation. A similar screening design would also be done to eliminate less significant control factors. Once we have identified the smaller set of control factors and noise factors, a single, simpler array that allows us to estimate the relevant interactions can then be used to drive the experiments for determining the final robust design.

Another situation that makes Taguchi methods awkward to use is the need to account for system constraint functions. Ku (1998) used penalty functions in the place of constraints, which converted the problem to an unconstrained optimization problem. This, however, eliminates the ability to estimate a design's feasibility. When desirability functions are used to represent constraints, they impose "soft" boundaries, again making it difficult to estimate feasibility.

Thus, we can conclude that Taguchi methods have excellent qualities, particularly when experiments are physical in nature, and when there isn't an underlying analytical relation that is understood. However, when we do have an understanding of the analytical relationship between the input factors and the response $f(\mathbf{x}, \mathbf{z}) = y$, and we need to account for design constraints, then using other methods (specifically, NLP methods) may provide a more suitable means for doing robust design.

2.2 Nonlinear Programming

Another method that has been used for robust design is Nonlinear Programming (NLP). NLP, in the general sense, is the mathematical process of solving an optimization problem where the model exhibits nonlinearity in the relationship between the input and output values.

The designer creates a model that interacts with the NLP optimizer, as shown in Figure 2.3. The model accepts input values from the NLP optimizer, computes the outputs of the model, then returns these values so the optimizer can determine what to do next.

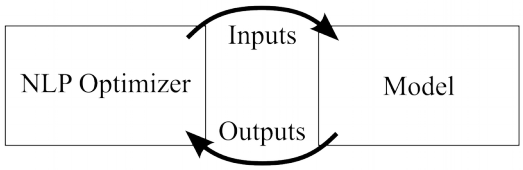

Figure 2.3 - Interaction between optimizer and model.

The inputs to the model are comprised of *design variables* and *design parameters*. Design variables are free to be set by the designer to improve the design (e.g., the length of a beam). These are the optimization analog to Taguchi's control factors. Design parameters, on the other hand, are also inputs to the model, but remain constant throughout the optimization process. These are held constant for a variety of reasons, but usually it is because the designer has little control over what values they can take on (e.g., material properties like elastic modulus and density). The set of all unique designs defined by the inputs constitutes the *design space*.

The input values are used by the optimization routines to calculate two kinds of output function values: *objective* function values and *constraint* function values. The objective function (there can be more than one) is the output that the designer is trying to improve as much as possible. The constraint functions determine the subset of designs in the design space that are considered *feasible*. The values of the design variables that yield the best objective function value within the feasible region give the optimal design.

The design problem can be stated in mathematical terms of a multiple objective optimization problem of the form:

Minimize	$f_k(\mathbf{x}, \mathbf{p})$	$k = 1, \ldots, q$
subject to	$g_i(\mathbf{x}, \mathbf{p}) \le b_i$	$i = 1, \ldots, r$
	$\mathbf{L} \le \mathbf{x} \le \mathbf{U}$	

where $\mathbf{x}$ = n-dimensional vectors of design variables
$\mathbf{L}, \mathbf{U}$ = lower and upper limits on $\mathbf{x}$, respectively
$\mathbf{p}$ = m-dimensional vector of fixed design parameters
f_k = k^{th} objective function
g_i = i^{th} inequality constraint function
b_i = i^{th} inequality constraint allowable value

In words, this means that our optimization problem has q objective functions f_k we wish to minimize. There are r inequality constraint functions, g_i, and allowable values, b_i, which we will simply denote as the vector **b.** Since we are focused on applying NLP methods to develop robust designs, we do not include equality constraint functions since they are virtually guaranteed to never be satisfied in the presence of variation. Also, although we specify the goal of *minimizing* the objectives with *less-than* inequality constraints, we maintain definition generality since any optimization problem, with *minimize/maximize/target* objectives and *less-than/greater-than* inequality constraints, can be represented in the above form. The vector $\mathbf{x}$ is an n-dimensional vector of design variables whose values are selected within the range of the lower and upper bounds, given by $\mathbf{L}$ and $\mathbf{U}$, respectively. These variables comprise the variables that can be directly controlled and modified in order to obtain the optimal design. The vector $\mathbf{p}$ is an m-dimensional vector of fixed parameters. These are considered constants to the system.

2.2.1 Including Model Tolerances

Conventional optimization algorithms help find the nominal optimum, but this doesn't take into account the presence of variability. To make the optimization problem more realistic, we need a way to incorporate variation. Thus, in addition to the nominal values of $\mathbf{x}$, $\mathbf{p}$, and $\mathbf{b}$, we include corresponding tolerance values $\Delta\mathbf{x}$, $\Delta\mathbf{p}$, and $\Delta\mathbf{b}$ that will be used to represent the expected variation for each kind of value. These tolerances can come from manufacturing

tolerances or any other sources of variation that are generally beyond the designer's control, but still need to be considered. For example, a material's density is usually specified with a nominal value, but in actuality has natural variation around that value. For the sake of this thesis, we will assume that the designer can determine appropriate tolerance values for **x**, **p**, and **b**.

We will specify tolerances in two different ways: *worst-case* (WC) and *statistical* (or root-sum-squared, RSS). In both cases we will assume the tolerances are symmetric about the nominal values, making it the *mean* value. In WC analysis, we assume that the tolerances $\Delta\mathbf{x}$, $\Delta\mathbf{p}$, and $\Delta\mathbf{b}$ are represented by *Uniform* distributions. Thus, for design variables, the minimum and maximum values are $\mathbf{x} - \Delta\mathbf{x}$ and $\mathbf{x} + \Delta\mathbf{x}$, respectively. The goal for WC analysis is to create a design that *never* violates the constraints (i.e., 100% feasibility). For the RSS analysis, the designer must know the variance (denoted by $\sigma_{\mathbf{x}}^2$, $\sigma_{\mathbf{p}}^2$, $\sigma_{\mathbf{b}}^2$) or standard deviation (denoted by $\sigma_{\mathbf{x}}$, $\sigma_{\mathbf{p}}$, $\sigma_{\mathbf{b}}$) of the distribution of the inputs. It is most common to assume, for RSS analysis, the distribution is a *Normal* or *Gaussian* distribution. The specified tolerance values are then chosen to represent $\pm1\sigma$, $\pm2\sigma$, or $\pm3\sigma$ of the distribution (i.e., $\pm\Delta\mathbf{x}=\pm3\sigma$). In this thesis, where RSS tolerances are concerned, we will use $\pm3\sigma$ to represent the tolerance limits. RSS tolerance conditions tend to be more realistic and are less conservative than WC tolerances. Figure 2.4 shows how RSS and WC tolerances may be specified.

Much work has been done to show how to include tolerances as part of design optimization. Balling (1986) and Michael and Siddall (1981, 1982) devised a method that placed a "tolerance box" around the design variables using primarily WC assumptions. The design was then adjusted until the tolerance box fit completely within the feasible region. This method can run into trouble when it isn't possible to fit the entire box within the feasible region. The methods also have trouble extending the problem to include design parameter tolerances.

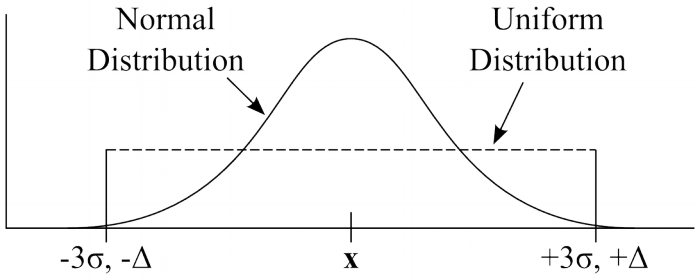

Figure 2.4 - Illustration of input tolerance distributions for RSS (Normal) and WC (Uniform) variables.

Gunawan and Azarm (2005) avoid this problem of distinction between design *variables* and design *parameters* by including all variables that have uncontrollable variation into the **p** term, even if they are part of *x*. Then, the design's feasibility robustness is determined via a sub-optimization, performed at each design point. This helps identify the worst-case sensitivity region (WCSR) within the sensitivity region (SR, i.e., the feasible region). Figure 2.5 illustrates that this WCSR is defined as the largest hypersphere around the point *x* that remains completely feasible. The sub-optimization solves for the radius of this hypersphere caused by all the contributing **Δp**. Although this can be an excellent tool for designing for feasibility robustness, calculating the WCSR is also too computationally expensive because it requires many calculations of the constraint functions at each design point. It doesn't, however, require any gradient evaluations.

Parkinson (1993) has done work on a linear tolerance model that works well with NLP methods. In his research, there are two main assumptions made. The first is that, for the RSS method, the transmitted variation to the design functions is normally distributed. This is mostly for the sake of simplicity since statistical error propagation is well suited to symmetrically distributed variables. This is a direct result of the Central Limit Theorem, which states that under certain (fairly common) conditions, sums or differences of random variables

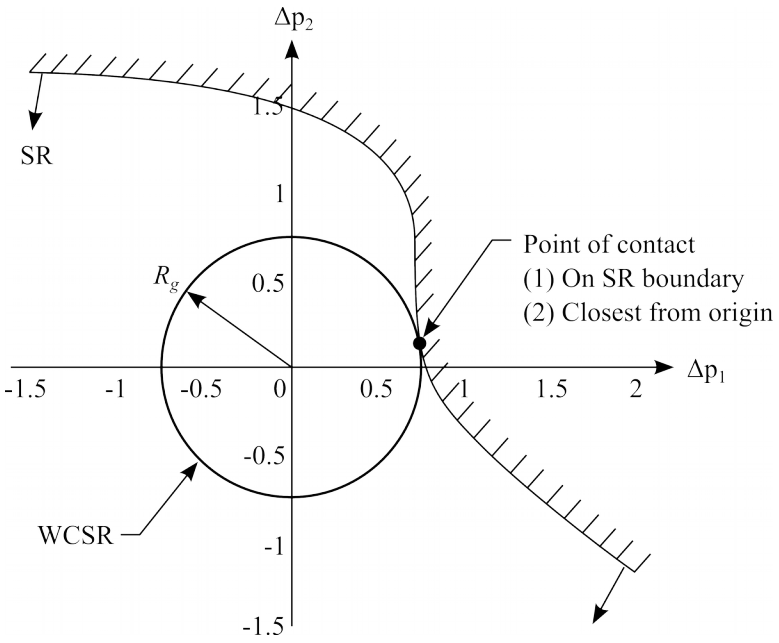

Figure 2.5 - Worst case representation of a sensitivity region (SR).

will have an approximately normal distribution. The second assumption is that second order effects are small (i.e., a Taylor series linear approximation for the mean and variance of the constraint functions is adequate for analysis). This is generally accurate enough when the tolerances are small (i.e., 5% or less of the nominal values of the design variables) or when the input-output relationship is relatively linear. However, when tolerances get larger on nonlinear models, this assumption breaks down and higher order tolerance methods may need to be used, such as the second-order method developed by Lewis (1994).

Lee and Park (2001) suggest a multi-objective formulation to balance the objective's optimality (or mean value, μ) when tolerances are *not* considered and the objective's robustness (or standard deviation, σ) when the tolerances *are* considered. This is done by combining the two objectives into a weighted sum, $\Phi(x)$:

$$\Phi(x) = \alpha \cdot \frac{\mu_f(x)}{\mu'_f} + (1-\alpha) \cdot \frac{\sigma_f(x)}{\sigma'_f}, \quad 0 \le \alpha \le 1 \tag{2.17}$$

where μ'_f, σ'_f are the mean and standard deviation of the objective *at the nominal optimum* and $\mu_f(x)$, $\sigma_f(x)$ are the mean and standard deviation of the objective evaluated at x. α is used as the weight factor to control how much the designer seeks a more optimal objective value or a more robust objective. Because this formulation requires the evaluation of $\sigma_f(x)$ at each design point, unless the underlying model can be evaluated easily, this method can be too computationally expensive for many modern engineering problems.

Another field that is growing in popularity is *stochastic optimization*. Stochastic optimization refers to directly incorporating statistically distributed design variables rather than deterministic (i.e., a single-value) design variables as part of the optimization problem definition. Once defined non-deterministically, the problem is then transformed into an equivalent deterministic one to be used with other more common numerical methods that can solve the transformed problem. One benefit of stochastic optimization is that the optimum will be found that will satisfy the constraints to a specified percentage, (e.g., 95%). This allows the designer to determine the feasibility robustness of the design from the start. A good introduction to stochastic optimization is given by Rao (1979). Many researchers have applied this technique to the optimization of a variety of mechanical designs, including four-bar mechanisms, cam design, aircraft wing structures, etc. (Rao, 1986b; Rhyu and Kwak, 1988; Agarwal, 1981; Beohar and Rao, 1980; Rao and Gavan, 1980). This research has tended to take two main avenues, as noted by Eggert (1990), with one avenue being Monte Carlo simulation and the other being analytical methods. Monte Carlo simulation is very popular due to its modeling flexibility and accuracy, but it, as well as other types of simulation, can be computationally expensive (Sundaresan et. al., 1991). Although powerful, stochastic optimization hasn't been used to actively control transmitted variation.

Another technique for optimizing for robustness is through the dual-response surface approach first suggested for computer-based experiments by Sacks (1989). Alternate proposals were given more recently by Lehman (2004), Bates (2005), and Giovagnoli (2008). A surrogate model is developed that is simpler to model and easier to evaluate for the underlying response function's mean value and its variance, then used simultaneously in an optimization model, like the following simple model:

Minimize $Var(y)$
subject to $\mathrm{E}[y] \leq y_{spec}$

where $Var(y)$ is the variance regression function, $\mathrm{E}[y]$ is the *Expectation* or mean-value regression function and y_{spec} is some design specification for the response function that should not be exceeded. This is done using response-surface techniques which provide accurate approximations, provided the underlying response function isn't too nonlinear. To determine the variance regression parameters, the response function's variance (or standard deviation) must be calculated at multiple points within the design space. This can be estimated stochastically using Monte Carlo simulation. The usefulness of this method comes at a much increased computational cost from the need to evaluate the actual response function at more than its mean value. For an in-depth discussion of response surface techniques, see Myers and Montgomery (2002).

2.2.2 Linear Uncertainty Propagation

Since we will be considering relatively small tolerances in this thesis, the linear approximation for calculating the variance of the constraint functions should be sufficient. Cox (1986) and Bjorke (1989) present the theory of *linear uncertainty propagation* (or tolerance analysis), which is the general statistical term for describing how variation in the input values

translates to variation in the output values. We will now examine how to consider both the WC and RSS input tolerance cases.

WC tolerance analysis assumes that all the input variations may occur simultaneously in the worst possible combinations. The effect of variation on the constraint functions is estimated with the first-order Taylor series:

$$\Delta_{g\,i}=\sum_{j=1}^{n}\left|\frac{\partial g_i}{\partial x_j}\Delta x_j\right|+\sum_{j=1}^{m}\left|\frac{\partial g_i}{\partial p_j}\Delta p_j\right| \tag{2.18}$$

where Δg_i represents the transmitted variation to the constraint function g_i for a WC analysis. We see that the variation only adds positively (as shown by the absolute value signs). Although Δg_i is almost always overly conservative, there are instances, such as with thermal expansion, that it is appropriate for use. However, if the tolerances on the inputs are independent of each other, it is very unlikely they will simultaneously occur in the worst possible combinations.

In a statistical RSS analysis, we allow for a small number of *rejects*, i.e. designs which are not feasible, out of a theoretical population set of designs. This allows the designer to use larger tolerances or back away from the optimal design a smaller amount than a WC analysis. For a linear RSS analysis, the variations in **x** and **p** sum, in terms of independent component variances, to result in an output variance σ_{gi}^2 for function g_i according to the expression

$$\sigma_{g\,i}^2=\sum_{j=1}^{n}\left(\frac{\partial g_i}{\partial x_j}\sigma_{x\,j}\right)^2+\sum_{j=1}^{m}\left(\frac{\partial g_i}{\partial p_j}\sigma_{p\,j}\right)^2 \tag{2.19}$$

Equations 2.18 and 2.19 show how variation from the input design variables and parameters are propagated to the constraint functions. In addition, we can include variation from the constraint values, b_i, to get a total transmitted variation

$$\Delta_i = \Delta_{bi} + \Delta_{gi} \tag{2.20}$$

where Δ_i is the total WC variation for the i^{th} constraint function and Δb_i is the tolerance on the constraint right hand side (RHS) value. For RSS analysis, we have a similar expression,

$$\sigma_i^2 = \sigma_{bi}^2 + \sigma_{gi}^2 \tag{2.21}$$

where σ_i^2 is the total statistical variance for the i^{th} constraint function and σ_{bi}^2 is the variance on the constraint RHS value. In this thesis, we will neglect σ_{bi}^2 for the most part and refer to Δg_i and σ_{gi}^2 as the *constraint variation* or simply *transmitted variation* and Δ_i and σ_i^2 as the *total constraint variation.*

2.2.3 Robust Design Method for Linear Analysis

With an understanding of how variation in the input design variables and parameters is transmitted to the constraint and objective functions, we are now prepared to discuss how the designer can use this to control the number of designs that will be infeasible when variability is present. For WC design, the designer would like to have 100% feasibility. For RSS optimization, the designer chooses the desired probability level that the robust optimum is to remain feasible. Because the assumptions for the RSS method are not always valid (i.e., our functions are normally distributed and second-order effects can be ignored), it is important to realize that the goal of RSS analysis to allow the designer to estimate the order of magnitude of the number of expected infeasible designs (i.e., *rejects*), such as 10%, 1%, 0.1%, etc., for a given set of input tolerances. This level of accuracy is adequate for many design situations and is usually consistent with the tentative nature of most information available during the design stage regarding the actual statistical distribution types and variances.

The "order of magnitude" range calculation is easily demonstrated. For example, if the designer wants the actual number of rejects to about 3%, we first calculate $\log_{10}(3)=0.477$. Then, to get the range lower bound by $0.477 - 0.5 = -0.023$. For the upper bound we calculate $0.477 +$

0.5 = 0.977. Finally, we take the antilog of these two values to get $10^{-0.023}$ = 0.948% and $10^{0.977}$ = 9.48% respectively. Thus, if the predicted percentage of rejects falls between 0.948% and 9.48%, we have order of magnitude agreement.

2.2.3.1 Feasibility Robustness

During design optimization, it is common to have the nominal optimum lie on one or more constraint boundaries. We would like to ensure that the input variable and parameter tolerances do not cause the design to be infeasible. A design is said to have *feasibility robustness* when it can be characterized by a definable probability, set by the designer, to remain feasible, given the variations in **x**, **p**, and **b**.

Feasibility robustness can be achieved by reducing the feasible region to account for the tolerances. Specifically, we increase the constraint right-hand-side for less-than constraints, or decrease the constraint right-hand-side for greater-than constraints by an amount chosen by the designer, typically equal to the total constraint variation. This adjustment will *always* make the feasible region smaller and, if the optimum is constrained, will make the objective worse. For WC analysis, this shift for less-than constraints is represented by

$$g_i + \Delta_i \leq b_i \tag{2.23}$$

An equivalent expression is made by adjusting the constraint right-hand-side (RHS) instead

$$g_i \leq b_i - \Delta_i \tag{2.24}$$

We can apply a similar shift for linear RSS analysis by simply changing Δ_i to $k\sigma_i$. The value of k is a constant that reflects the probability that the design will remain feasible with respect to the i^{th} constraint.

For example, since we assume the constraint variation to be approximately normally distributed, k = 3 means that for a large number of sampled designs, the constraint should be satisfied approximately 99.86% of the time if the constraint is currently binding. Other values for k and the corresponding percentages are shown below in Table 2.5. These values are based on a one-sided percentile calculation of the standard normal distribution since, usually, a design only violates a constraint on one side. The $k\sigma_i$-shift of the constraint to maintain feasibility is illustrated in Figure 2.6. When more than one constraint is binding at the nominal optimum, the total predicted feasibility becomes the product of the feasibility of each $k\sigma_i$-shift. For example, if we apply a 3σ-shift to two binding constraints, the total predicted feasibility changes to approximately 0.9986*0.9986 = 0.9971 or 99.71%.

Table 2.5 – Relation of k to constraint feasibility

k (number of standard deviations)	Percentage of Feasible Designs (based on standard normal distribution)
1	84.13
2	97.73
3	99.856
4	99.9968
5	99.999971
6	99.999999

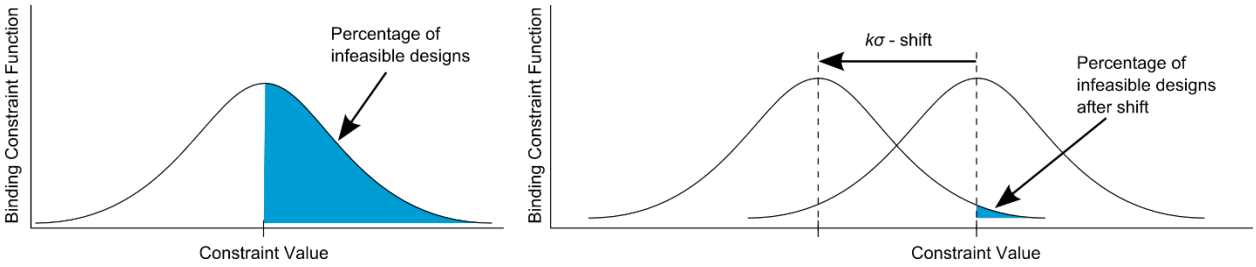

Figure 2.6 - Application of $k\sigma$-shift to binding constraint function to control feasibility.

2.2.3.2 Two-Step Solution for Feasibility Robustness

Gradient-based optimization algorithms require the calculation of first derivatives of the objective function and constraint functions. When the transmitted variation equation, which is comprised of derivatives, is used in optimization models, this requires calculating the derivatives of derivatives, or second derivatives of the original functions . For many problems, calculating second derivatives can be computationally expensive. A *two-step method* can be performed to reduce the need for continuous evaluation of transmitted variation and function second derivatives.

The *first step* is to optimize the design as usual, subject to the un-shifted constraints. At this point, the transmitted variation for each constraint is calculated according to Equations 2.20 or 2.21. If the designer doesn't have access to the partial derivatives from the optimization routine, then they must be calculated separately using finite difference equations, automatic differentiation, or any other suitable method.

The *second step* is to shift each constraint by $k\sigma_i$ or Δ_i and re-optimize subject to the new constraints, starting at the nominal optimum. This step assumes that the transmitted variation is constant through the shift (for RSS), which should be adequate provided the tolerances are small. The nominal optimum will always be infeasible with respect to the shifted binding

constraints, so an algorithm that can start at an infeasible design should be used, such as sequential quadratic programming (SQP).

2.2.4 Example of Feasibility Optimization

Parkinson (1993) demonstrates the two-step method with several design problems. For each problem, the robust optimum was identified and verified with Monte Carlo simulation. For the case where tolerances were assumed to be $\pm 3\sigma = \pm 5\%$ of the nominal value, the amount of predicted infeasibility of the linear tolerance analysis matched the simulated number of rejects from the Monte Carlo simulation within an order of magnitude. In fact, typically the Monte Carlo simulation showed a significantly better feasibility than was predicted using the two-step method.

Recall the previous two-bar truss example. This was optimized for minimum weight, subject to two stress constraints and one deflection constraint. At the nominal optimum, the weight was calculated to be 56.9 N, with the yield stress and buckling stress constraints binding. After calculating the transmitted variation and shifting each of the constraints, the optimal weight increased to 63.0 N. With two binding constraints, the estimated feasibility for the design was 99.7%. Using Monte Carlo simulation, the actual feasibility was 99.8%, well within the order of magnitude limit. Similar results were obtained for several other design problems.

2.2.5 Sensitivity Optimization

Sometimes the designer not only wants to stay feasible when the design is subject to variation, but also wishes to reduce the sensitivity of the objective to variation. The designer can perform *sensitivity optimization* for the purpose of finding a design that has minimum

sensitivity to the input variation and still satisfies the design constraints. The goal of this form of optimization is to find a design where the objective is insensitive to variation (i.e., where the transmitted variation is small). By examination of Equation 2.21, we see that this will happen when the partial derivatives of the variables and parameters are small, when the tolerances themselves are small, or a combination of the two. Since tolerance reduction can become costly, requiring more expensive manufacturing processes and more quality control, it is desirable to find a design with less sensitivity to given tolerances without changing the tolerances themselves. Indeed, if the sensitivity is reduced enough, it may even be possible to *increase* the tolerances.

Formulation of the design problem for sensitivity optimization will generally follow two options: make the transmitted variation of the objective *another* objective, or define a reasonable limit for the original objective function (turning it into a constraint function) and setting the transmitted variation as the only objective. The latter option will be used throughout this thesis to maintain the simplicity of a single objective. Then, it becomes a regular optimization design problem; if we wish, feasibility robustness can also be included in the process.

2.2.6 Benefits and Drawbacks of NLP

We have discussed many uses and benefits for using NLP methods, but they will be summarized here. The requirement of gradient-based optimizers to have derivatives is easily met if we assume the functions are continuous. These derivatives can be calculated using a variety of techniques. Unlike Taguchi methods, NLP methods can easily accommodate problems with constraints. NLP methods can also handle single and multiple objectives with flexibility in the kind of objectives allowed (i.e., minimize, maximize, or target).

However, NLP methods have drawbacks. Because NLP methods require the calculation of derivatives at each iteration for each function with respect to each input variable, the computational effort required to determine the optimum can grow considerably as the number of input design variables grows. When performing sensitivity optimization, the number of required function evaluations compounds on the order of $O([n + m]^2)$ for n design variables and m parameters with tolerances. For analytical functions (i.e., "simple" mathematical equations), this may not be a hindrance, but when the designer is using numerical simulation tools, such as finite element analysis (FEA), computational fluid dynamics (CFD), etc., the computational cost of calculating second derivatives can be very prohibitive. Other well understood issues are characteristic of NLP methods, including the potential for sub-optimal designs if the objective's topology has multiple minima within the design space.

2.3 Conclusions

We have addressed the development and use of experimental methods in Section 2.1 and NLP methods in Section 2.2 and discussed why both of these methodologies may not be suitable for robust optimal design (i.e., feasibility and sensitivity optimization). In Chapter 3, a hybrid optimization method will be presented that combines some of the efficiencies of experimental methods and the accuracies of NLP methods.

CHAPTER 3 PROPOSED SOLUTION

In order to address the aforementioned limitations and utilize the benefits of both DOE and NLP methods, we propose the following solution. This solution is a hybrid algorithm that blends NLP and DOE methods. When computational expense is relatively low, we use the NLP approach to robust design; when it is high, we build a statistical model using DOE methods and apply NLP methods to the statistical model.

3.1 The Search-and-Zoom Algorithm

The hybrid algorithm, which is designated the *Search-and-Zoom* algorithm for reasons that will be described shortly, is of most practical use when performing sensitivity optimization (i.e., minimizing transmitted variation). It maximizes the use of the statistical model for the calculation of derivatives rather than by using the actual model. The *Search-and-Zoom* iterative algorithm is outlined in Figure 3.1 and described in the following steps:

1. Start from a convenient point. This can be the nominal optimum or something else since there are no presumptions about the proximity of the sensitivity and nominal optimums.
2. Set the starting variable bounds to be approximately 20% of the full variable bounds. This is somewhat arbitrarily chosen, but is designed to give the algorithm a "head-start" and should still provide opportunities for the algorithm to move around the design space.

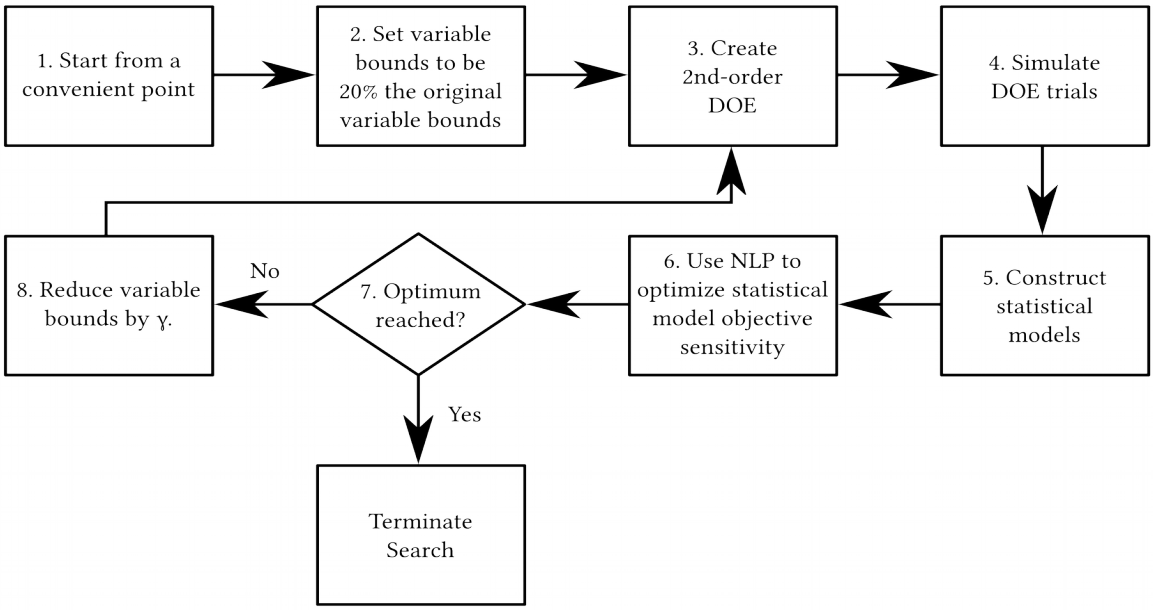

Figure 3.1 - Flowchart diagram of the *Search-and-Zoom* optimization algorithm.

3. Construct a combinatorial set of input variable values that allows for the construction of a statistical model that includes the estimation of second-order effects. These could be Central-Composite Designs (CCD), Box-Behnken designs (BB), etc.
4. Simulate each trial condition using the actual model and collect all outputs.
5. Construct statistical models to approximate the system for all output functions.
6. (*Search*) Perform an NLP optimization on the statistical model where the goal is to minimize the objective's sensitivity using either Equation 2.20 or 2.21, subject to any constraints, over the input variable bounds.
7. Evaluate the optimal design from step 6 using the actual model (f_{opt}) and check for convergence between the optimum of the previous iteration and the current one. If the change in f_{opt} is below the tolerance, stop. The optimal design has been reached.

8. (*Zoom*) If convergence has not been reached, reduce the allowable search space around the current iteration's optimum using a *zoom factor* ($0<\gamma<1$), typically between 0.5 and 0.85, with the current iteration's optimum at the center of the new ranges. If necessary, apply shifts to the new variable ranges in order to maintain the original variable bounds. For example, if the original bounds of a variable were [1, 3] and the new iteration bounds were calculated to be [2.75, 3.25], then, in order to remain entirely within the original design space, we would shift BOTH values down by 0.25 to get [2.5, 3] to keep the range of the bounds. This is checked for each variable before moving on.
9. Repeat steps 3-8 until the optimum converges at step 7.

Once the sensitivity optimum is found with this process, it may be necessary to further optimize for feasibility robustness. Since the feasibility optimums tend to be near the nominal optimum, we can continue using *Search-and-Zoom* to our advantage:

1. Estimate the transmitted variation to the constraint functions using the actual model (3σ for RSS conditions, Δ for WC conditions).
2. Shift constraints by their respective transmitted variation.
3. Increase the variable bounds by $\alpha = 1/\gamma^3$. Since γ is less than 1.0, this makes α have an expansion effect on the variable bounds, once again giving the algorithm more opportunity to search.
4. Re-instate *Search-and-Zoom* starting at the sensitivity optimum until it converges to the RSS and WC feasibility optimums.

The algorithm will now be explained via a simple analytical example using the design of the two-bar truss. We start with just two design variables, truss height H and tube diameter d. A suitable DOE for two variables that allows for excellent second-order approximation comes from a Central Composite Design (CCD). These designs are comprised of *corner points*, *star*

points, and *center points*. In physical experiments, the center point is replicated numerous times to capture the variability in the system. However, analytical equations provide consistent output for a given set of input values, so we only need a single center point. We will also scale the design down so the star points are located at the design variable bounds rather than traditionally extending outside them, creating an *inscribed CCD*.

The design matrix for this set of trials is outlined in Table 3.1. We notice that there are 9 total trial conditions. The first four trials are the corner points, trials 5-8 are the star points, and trial 9 is the center point. These values (X_{scaled}), ranging from −1 to +1, are related to the actual lower and upper bound values of the design variable ranges ($X_{unscaled}$) using the following equations,

$$X_{scaled} = \frac{X_{unscaled} - X_{average}}{X_{hw}} \tag{3.1}$$

or

$$X_{unscaled} = X_{average} + X_{hw} X_{scaled} \tag{3.2}$$

where,

$$X_{average} = \frac{X_{max} + X_{min}}{2} \tag{3.3}$$

$$X_{hw} = \frac{X_{max} - X_{min}}{2} \tag{3.4}$$

Table 3.1 – Inscribed central composite design for two variables and no variability (values are normalized)

Trial	x_1	x_2
1	−0.70710 678	−0.70710 678
2	0.70710 678	−0.70710 678
3	−0.70710 678	0.70710 678
4	0.70710 678	0.70710 678
5	−1	0
6	1	0
7	0	−1
8	0	1
9	0	0

The min/max values for H and d are [10, 30] and [1, 3], respectively. Using Equations 3.3 and 3.4, we first calculate the *average* and *half-width* (*hw*) values. These are then used in Equation 3.2 to transform each value in the design matrix to be real values that are supplied to the design equations. Table 3.2 shows the same design matrix but with the values transformed into the actual design values using their min/max values. The output functions we will use in this analysis are *weight*, *stress*, *buckling stress*, and *deflection*, as defined in Equations 3.5 – 3.8.

Table 3.2 – Inscribed CCD using actual two-bar truss design variable values

Trial	H	d
1	12.9289	1.2929
2	27.0711	1.2929
3	12.9289	2.7071
4	27.0711	2.7071
5	10	2
6	30	2
7	20	1
8	20	3
9	20	2

$$Weight = 2 \cdot \pi \cdot \rho \cdot d \cdot t \cdot \sqrt{\left(\frac{B}{2}\right)^2 + H^2} \tag{3.5}$$

$$Stress = \frac{P \cdot \sqrt{\left(\frac{B}{2}\right)^2 + H^2}}{2 \cdot \pi \cdot d \cdot t \cdot H} \tag{3.6}$$

$$Buckling\ Stress = \frac{\pi^2 \cdot E \cdot \left(d^2 + t^2\right)}{8\left[\left(\frac{B}{2}\right)^2 + H^2\right]} \tag{3.7}$$

$$Deflection = \frac{P \cdot \left[\left(\frac{B}{2}\right)^2 + H^2\right]^{\frac{3}{2}}}{2 \cdot \pi \cdot d \cdot t \cdot H^2 \cdot E} \tag{3.8}$$

The parameters other than H and d are held constant at the following values:

$t = 0.15$ $\rho = 0.3$ $E = 30\,000$ $B = 60$ $P = 66$

If we start from the nominal optimum, we know we have both a feasible and favorable design. Thus, we define the starting optimization model as

Minimize	$f = Weight$		
Subject to	$Stress \le 100$		$g_s = 100 - Stress \ge 0$
	$Stress \le Buckling\ Stress$	or	$g_b = Buckling\ Stress - Stress \ge 0$
	$Deflection \le 0.25$		$g_d = 0.25 - Deflection \ge 0$

In this example, we will use f and g_x to simplify the notation for the objective function and the design constraint functions, respectively. We can now begin the *Search-and-Zoom* algorithm. **(Step 1)** Using NLP methods, the nominal optimum is discovered at the following design (we will consider any constraint value that is relatively close to zero a *binding* constraint, shown in *italics*):

Table 3.3 - Two-bar truss nominal optimum before sensitivity optimization

Design Variable	Value	Design Function	Value
H	14.214895	f	15.868277
d	1.690574	g_s	3.261734
		g_b	*7.3941e-6*
		g_d	*5.7482e-4*

Now that we have the nominal optimum, we reformulate the optimization model to improve the design's *sensitivity robustness* of the *Stress* function. Thus, we add a new constraint on *Weight* (via g_w) to stay near the same performance level as the nominal optimum:

Minimize	$f = \sigma_{Stress}$		
Subject to	$Weight \leq 17$		$g_w = 17 - Weight \geq 0$
	$Stress \leq 100$	or	$g_s = 100 - Stress \geq 0$
	$Stress \leq Buckling\ Stress$		$g_b = Buckling - Stress \geq 0$
	$Deflection \leq 0.25$		$g_d = 0.25 - Deflection \geq 0$

(Step 2) We will assume the tolerances for H and d are ±0.05 and ±0.005, respectively (equivalent to ±Δ and ±3σ, here). The starting variable bounds around the nominal optimum are then set to have a width of approximately 20% of the full original bounds. Thus, we get: [14.214895 − 2.0, 14.214895 + 2.0] = [12.214895, 16.214895] for H and [1.690574 − 0.2, 1.690574 + 0.2] = [1.490574, 1.890574] for d. **(Step 3)** Following the same procedure as described above, we construct a set of experiments that provide the right information to construct quadratic regression functions, in this case, an inscribed CCD. **(Step 4)** Each of the output functions is then evaluated at each trial condition, except for f, since this will be estimated using the corresponding regression equation (i.e., g_s). Table 3.4 shows each of the calculated output values (the response matrix) to the right of the given input trial conditions of H and d.

Table 3.4 – Two-bar truss starting design matrix and response matrix

Trial	H	d	g_w	g_s	g_b	g_d
1	12.8007	1.5492	2.7134	−15.1827	−30.9096	−0.0691
2	15.6291	1.5492	2.1833	2.1618	−19.4873	0.0112
3	12.8007	1.8320	0.1050	2.6004	20.1438	−0.0198
4	15.6291	1.8320	−0.5219	17.2671	26.5503	0.0481
5	12.2149	1.6906	1.5170	−9.8446	−8.2326	−0.0645
6	16.2149	1.6906	0.6995	12.8836	4.5593	0.0417
7	14.2149	1.4906	3.0090	−9.7182	−34.3467	−0.0335
8	14.2149	1.8906	−0.7455	13.4955	34.2872	0.0264
9	14.2149	1.6906	1.1317	3.2618	0.0001	0.0000

(Step 5) We construct a quadratic approximation y_k for each output function's responses. For two factors, this function is comprised of a constant, two linear terms, two pure quadratic terms, and one interaction term,

$$y_k = \beta_{0k} + \beta_{1k} H + \beta_{2k} d + \beta_{3k} H^2 + \beta_{4k} H d + \beta_{5k} d^2 \tag{3.9}$$

The coefficients β_i are determined using statistical regression techniques. Using a least-squares fit, for example, we get the coefficients for each function's quadratic approximation as found in Table 3.5.

During sensitivity optimization, we could include more factors that have tolerances and create a regression model to support that number of factors, but we will keep it to two factors in this example for simplicity.

Table 3.5 – Initial quadratic regression coefficients for the two-bar truss output functions along with the corresponding goodness-of-fit values (R^2)

y_k	$k = 1$, g_w	$k = 2$, g_s	$k = 3$, g_b	$k = 4$, g_d
β_0	15.816867	−441.642133	−580.375424	−1.827579
β_1	0.167213	23.685287	26.802116	0.133267
β_2	−7.670076	221.415521	262.936181	0.668574
β_3	−0.005879	−0.434615	−0.458242	−0.002838
β_4	−0.120979	−3.347281	−6.269800	−0.015501
β_5	−0.000006	−34.233217	−0.651297	−0.087873
R^2	0.999999	0.999997	0.999998	0.999952

(Step 6) Substituting the coefficients from Table 3.5 into Equation 3.9 for each function, we then run an NLP optimization on the *transmitted variation function of the regression objective*. That is, we use Equation 3.9 to approximate g_s near the iteration optimum and Equation 2.20 or 2.21 estimate the WC or RSS transmitted variation, respectively, and minimize that function. Here, we will use the RSS transmitted variation (σ) as the objective function. For the starting value, we will use the mean value of the variable bounds, x_0 = [14.2149, 1.6906], which, in this case, is the nominal optimum found using NLP methods in Step 1. This leads us to the sensitivity optimum of the first iteration, with the design variable and *actual* response values shown in Table 3.6.

Table 3.6 - Two-bar truss sensitivity optimum after the first iteration

Design Variable	Value	Design Function	Value
H	16.214886	f	0.398832
d	1.763179	g_w	*–5.9286e-4*
		g_s	16.470860
		g_b	16.127150
		g_d	*5.0311e-2*

(Step 7) At this point, we check for convergence between consecutive iteration optimums, which we can calculate using Equation 3.10:

$$\left| f_{opt,i} - f_{opt,i-1} \right| \leq \varepsilon \quad , \tag{3.10}$$

where, again, $f_{opt,i}$ is the objective value at the optimum for the current iteration and $f_{opt,i-1}$ and is that of the previous iteration. For this problem, we'll assume ε = 1e-4. At the end of this iteration, $f_{opt,i}$ = 0.1032 and $f_{opt,i-1}$ = 0.1329. Since |0.1032-0.1329| = 0.0297 > ε, we continue on to Step 8.

(Step 8) Using a *zoom factor* of γ = 0.75, we reduce the search-able space, noting that the prior iteration's optimal factor values act as the center point of the next DOE. Thus, the new bounds for H are [16.214886 – 2.0*0.75, 16.214886 + 2.0*0.75] = [14.714886, 17.714886] and d are [1.763179 – 0.2*0.75, 1.763179 + 0.2*0.75] = [1.613179, 1.913179]. Since these bounds fall completely within the original bounds ([10, 30] for H and [1, 3] for d), there is no need to make any further adjustments to the new ranges. Thhe new design matrix is found in Table 3.7.

Table 3.7 – Design matrix for second iteration of two-bar truss sensitivity optimization.

Trial	H	d
1	15.154226	1.657113
2	17.275546	1.657113
3	15.154226	1.869245
4	17.275546	1.869245
5	14.714886	1.763179
6	17.714886	1.763179
7	16.214886	1.613179
8	16.214886	1.913179
9	16.214886	1.763179

This process continues with steps 3 through 8 until we either meet the convergence criteria in step 7 or a maximum number of allowable iterations pre-set by the designer. If we continue the above steps for 9 iterations (111 total actual function calls), we converge to the following design:

Table 3.8 - Two-bar truss sensitivity optimum

Design Variable	Value	Design Function	Value
H	21.186979	f	0.254791
d	1.637075	g_w	*2.1082e-7*
		g_s	25.847952
		g_b	*–2.6757e-6*
		g_d	9.2635e-2

We see that the sensitivity function f has been reduced from 0.398832 to 0.254791, a 36% decrease, as estimated using the first-order Equation 2.21. As we will see in Sections 3.2.1 and 3.2.2, the improvement is actually greater, in this case, when we validate this design using Monte Carlo simulation.

3.2 Feasibility Optimization

Because at least one constraint is *binding*, we know that variation around the input variable nominal values will cause the design to become infeasible. The *Search-and-Zoom* algorithm will now be used for improving the design's *feasibility robustness* while still maintaining the sensitivity robustness we've already achieved. This is done by following the two-step method described in Section 2.2. We will first estimate the transmitted variation to the constraints.

3.2.1 Calculation of Transmitted Variation

To estimate the transmitted variation for each constraint, we utilize Equation 2.21 for RSS tolerances and Equation 2.20 for WC tolerances, both of which require the calculation of first derivatives. These are estimated using first-order forward-difference derivatives at the nominal optimum using the actual model. Table 3.9 shows the partial derivatives for each constraint function.

Table 3.9 – Partial derivatives for constraint functions at the nominal optimum

	g_w	g_s	g_b	g_d
H	−0.267019	2.335169	0.005759	0.007440
d	−10.384374	45.292682	135.132102	0.096120

For this calculation, we assumed a perturbation of 1e-4 in the finite difference derivatives. Recall that the two design variables H and d have the tolerances of ±0.05 and

±0.005, respectively. The RSS transmitted variation for the *Stress* constraint (assuming *tolerance* = 3σ or rather, σ = tolerance/3) can then be calculated with $\sigma_H = 0.016667$ and $\sigma_d = 0.001667$,

$$\begin{aligned} \sigma_{gs}^2 &= \left(\frac{\partial g_s}{\partial H}\sigma_H\right)^2 + \left(\frac{\partial g_s}{\partial d}\sigma_d\right)^2 \\ &= \left((2.335170)(0.016667)\right)^2 + \left((45.292685)(0.001667)\right)^2 \\ &= 0.029969 \end{aligned} \tag{3.11}$$

or,

$$\begin{aligned} 3\sigma_{gs} &= 3\sqrt{0.029969} \\ &= 0.254791 \end{aligned} \tag{3.12}$$

The value for $3\sigma_{gs}$ represents the amount of statistical transmitted variation we expect from the input tolerances. Likewise, we calculate WC transmitted variation for *Stress* as,

$$\begin{aligned} \Delta_{gs} &= \left|\frac{\partial g_s}{\partial H}\Delta_H\right| + \left|\frac{\partial g_s}{\partial d}\Delta_d\right| \\ &= \left|(2.335170)(0.05)\right| + \left|(45.292685)(0.005)\right| \\ &= 0.343222 \end{aligned} \tag{3.13}$$

One thing to notice is that Δ_{gs} is greater than $3\sigma_{gs}$. This will always be the case, and as more variables are added to the transmitted variation equations, the difference becomes more pronounced. Because of this, it becomes more obvious that WC analysis can be overly conservative.

Following the same procedure, we can calculate similar transmitted variation values for g_w, g_b and g_d. Table 3.10 shows the tabulated values for both RSS and WC transmitted variation.

With these values calculated, the original model constraint right-hand-sides are adjusted to take into account the transmitted variation, according to the desired feasibility.

Table 3.10 – RSS and WC transmitted variation to constraint functions at the sensitivity optimum

Constraint	RSS, 3σ	WC, Δ
g_w	0.053611	0.065273
g_s	0.254791	0.343222
g_b	0.675661	0.675948
g_d	0.000608	0.000853

3.2.2 Statistical Feasibility Optimization

The new statistical (RSS) feasibility optimization model is then re-formulated to include the amount of expected 3σ transmitted variation on each constraint function g_x:

Minimize $f = \sigma_{Stress}$

Subject to

$Weight \le 17 - 0.053611$		$g_w = 16.946389 - Weight \ge 0$
$Stress \le 100 - 0.254791$	*or*	$g_s = 99.785209 - Stress \ge 0$
$Stress \le Buckling\ Stress - 0.67566$		$g_b = (Buckling - 0.67566) - Stress \ge 0$
$Deflection \le 0.25 - 0.000608$		$g_d = 0.249392 - Deflection \ge 0$

The optimization is now re-run, starting at the sensitivity optimum with the new constraints. Since we expect the feasibility optimum to be close to the sensitivity optimum, rather than use the full variable bounds as the starting range for constructing the design matrix, we simply expand the final bounds. After the 9 iterations it took to reach the sensitivity optimum, and starting at a half-width of 20% or 0.2, each variable range has been reduced to $0.2*0.75^9 = 0.015017$, or roughly 1.5% the original bound widths. If we zoom-out three steps, we get $\alpha = 1/(0.75^3) = 2.3704$, so $0.015017*\alpha = 0.015017*(2.3704) = 0.035596$ or about 3.6% the width of the original bounds. Again, shifts to the new bounds may be necessary to

keep them within the original bounds, but since the newly calculated bounds lie entirely within the original, this is not the case here.

The *Search-and-Zoom* algorithm is re-instated at this point, subject to the new constraints and leads to the following new robust optimum after 3 iterations (39 total actual function calls):

Table 3.11 - Two-bar truss sensitivity optimum with RSS feasibility

Design Variable	Value	Design Function	Value
H	20.787097	f	0.086134
d	1.642161	g_w	*1.7554e-7*
		g_s	24.870996
		g_b	*1.2314e-5*
		g_d	8.9452e-2

To confirm the feasibility robustness, we will use Monte Carlo simulation. Since this is an RSS analysis, we assume the tolerances are *normally distributed*, notated as $N(\mu, \sigma)$ with a mean μ and standard deviation σ, and that the tolerance values represent $\pm 3\sigma$. Thus, we assume that $H \sim N(20.787097, 0.016667)$ and $d \sim N(1.642161, 0.001667)$. Taking 10 000 random samples from these distributions (since this is a computationally "cheap" model) and evaluating the model against the *original* constraints (from when we started the sensitivity optimization), and knowing that there are two *binding constraints* (g_w and g_b) at the sensitivity optimum, we expect the feasibility to be roughly the product of the feasibilities of the number of binding constraints. For each constraint's 3σ-transmitted variation we estimate the one-sided feasibility to be 0.9986 or 99.86%. Multiplying together gives us a total feasibility of 0.9986*0.9986 = 0.997

or 99.7%. From the 10,000 samples, only 23 samples violated any constraints, which indicate an approximate 1 – 23/10 000 = 99.77% feasibility—almost exactly the desired amount.

3.2.3 Worst-Case Feasibility Optimization

We expect the result of RSS feasibility optimization to have a small percentage of infeasibility, but for worst-case (WC) feasibility optimization, we use a more conservative representation of the input tolerances in hopes that we can find an optimal design that is *always feasible*. Following the same process as we did for RSS feasibility optimization, we start by applying the WC transmitted variation, Δ, from Table 3.10 to the constraint right-hand-sides to get the following model:

Minimize $f = \sigma_{Stress}$

Subject to

$Weight \le 17 - 0.065273$		$g_w = 16.934727 - Weight \ge 0$
$Stress \le 100 - 0.343222$	*or*	$g_s = 99.656778 - Stress \ge 0$
$Stress \le Buckling\ Stress - 0.67595$		$g_b = (Buckling - 0.67595) - Stress \ge 0$
$Deflection \le 0.25 - 0.0.000853$		$g_d = 0.249148 - Deflection \ge 0$

The starting input values to the S*earch-and-Zoom* algorithm, again, come from the sensitivity optimum, where we calculated the transmitted variation. We set the starting variable bounds to be the same as for the RSS feasibility optimization which, in turn, takes us to the WC robust optimum after 3 iterations (39 function calls):

Table 3.12 - Two-bar truss sensitivity optimum with WC feasibility

Design Variable	Value	Design Function	Value
H	20.741812	f	0.086327
d	1.642191	g_w	*2.9907e-8*
		g_s	24.673457
		g_b	*2.2614e-5*
		g_d	8.8852e-2

To simulate WC variability conditions in the confirmation Monte Carlo simulations, we assume the input tolerances are *uniformly distributed*, and that the lower and upper bounds of the distributions are $[\mu - \Delta, \mu + \Delta]$, respectively, where μ is the nominal value and Δ is the tolerance. Thus, we define the design variables as $H \sim U(20.741812 - 0.05, 20.741812 + 0.05) = U(20.691812, 20.791812)$ and $d \sim U(1.642191 - 0.005, 1.642191 + 0.005) = U(1.637191, 1.647191)$. After simulating 10,000 samples from these two distributions at the WC feasibility optimum, against the original constraints, the number of designs that violate at least one constraint is 70 for a feasibility of 99.3%. This is not quite what we would expect from a WC feasibility optimization. Figure 3.2 shows a zoomed-in contour plot of the two-bar truss's design space, with the feasible region shaded for clarity. Since the WC sensitivity optimum is so close to an acute intersection of the *Weight* and *Buckling* constraints, this is the most likely cause of the feasibility being less than 100%. This is because the 2-step method for feasibility optimization makes the assumption that the constraints are independent of each other (i.e., they are perpendicular), which is clearly not the case here. Table 3.13 summarizes the feasibility of the sensitivity, RSS and WC optimums.

Table 3.13 – Monte Carlo estimated feasibility of sensitivity optimum, and sensitivity optima with RSS and WC feasibility using RSS and WC tolerances on input variables after *Search-and-Zoom* optimizations.

Tolerance Type	Sensitivity Optimum (SO)	SO with RSS Feasibility	SO with WC Feasibility
RSS	4.04%	99.77%	–
WC	3.10%	–	99.3%

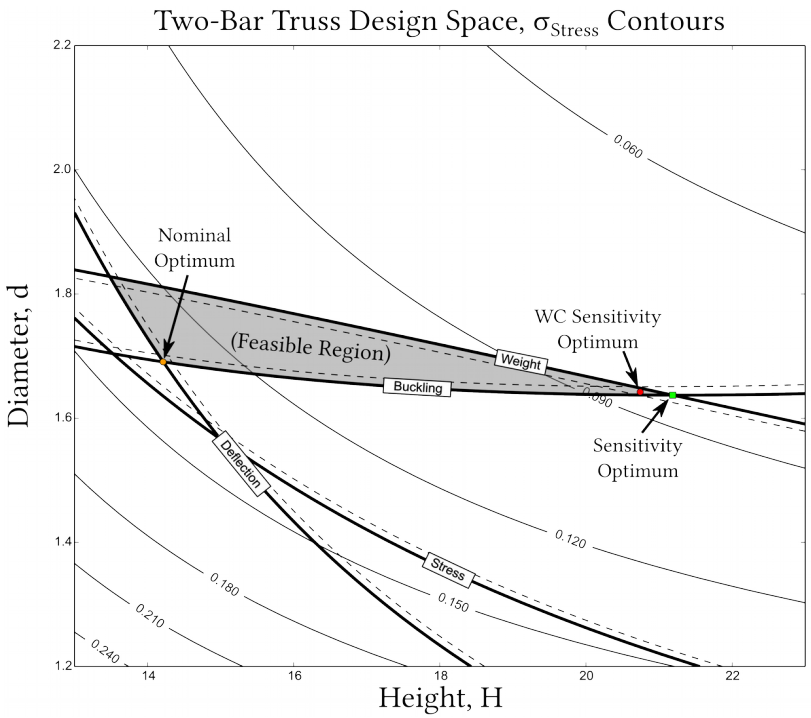

Figure 3.2 – 2D contour plot of the two-bar truss design space. The dashed lines indicate the feasible side of the constraints (only the original constraints are shown).

It is interesting and important to see how the variability in *Stress* was reduced from 0.132943 at the nominal optimum and to that in the three sensitivity optima. This can be seen in the histograms in Figure 3.3 from the Monte Carlo simulations at each of the respective RSS tolerances and WC tolerances. Because the mean values are so different from the nominal optimum to the sensitivity optima, the histograms in Figure 3.3 have been centralized for easier comparison. Although we can see the difference in the numeric values, the histogram makes it much more apparent how much the variability has been reduced.

3.3 Conclusions

The *Search-and-Zoom* algorithm has been presented with an example for performing robust optimal design. Efficient methods for achieving feasibility robustness and sensitivity

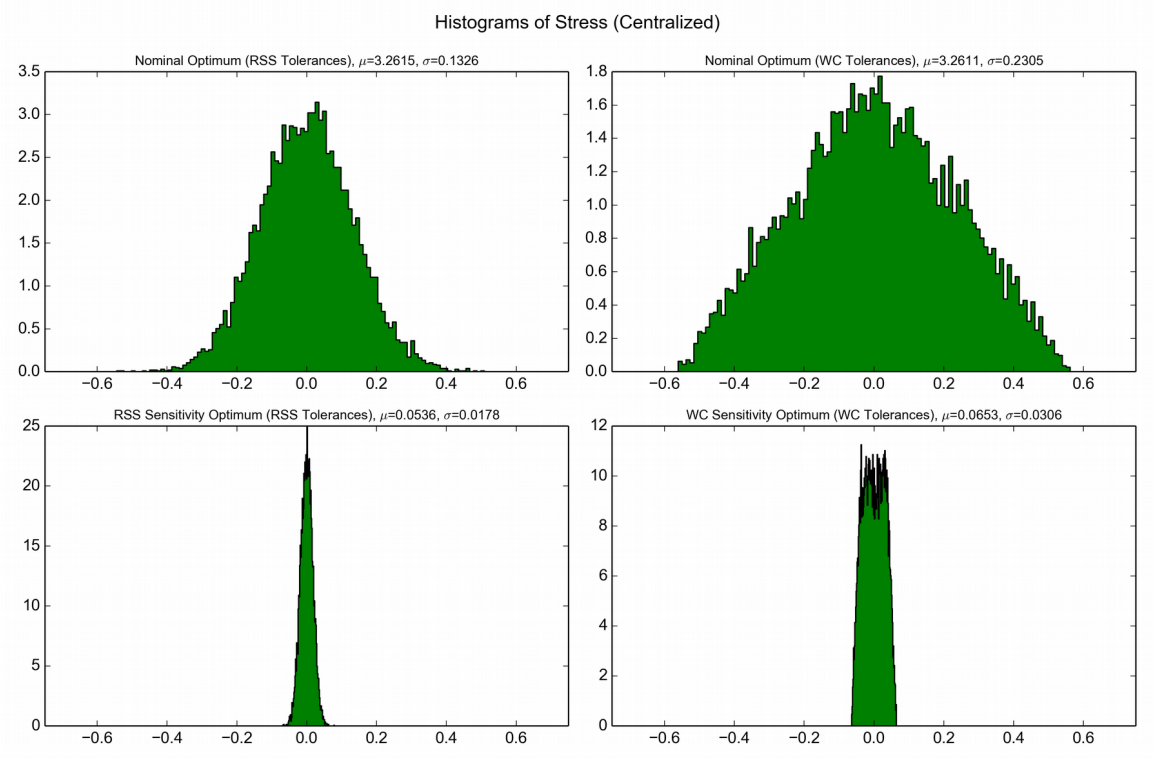

Figure 3.3 - Centralized histograms of *Stress* comparing the nominal optimum (top row) and sensitivity optima with RSS and WC feasibility (bottom row) using RSS tolerances (left column) and WC tolerances (right column).

robustness are also explained. Although not directly addressed previously, the efficiency of the *Search-and-Zoom* algorithm will diminish with an increase in input design variables. This is because the full CCD's number of trial conditions required for the quadratic approximations increases exponentially with the number of design variables, as shown in Figure 3.4. There may be other experimental designs that provide better efficiency for larger numbers of variables, but that is not addressed in this thesis.

In Chapter 4, we will demonstrate the benefit of using the S*earch-and-Zoom* algorithm by way of two applications in the communications engineering industry: a high-frequency micro-strip band-pass filter and a small rectangular patch antenna.

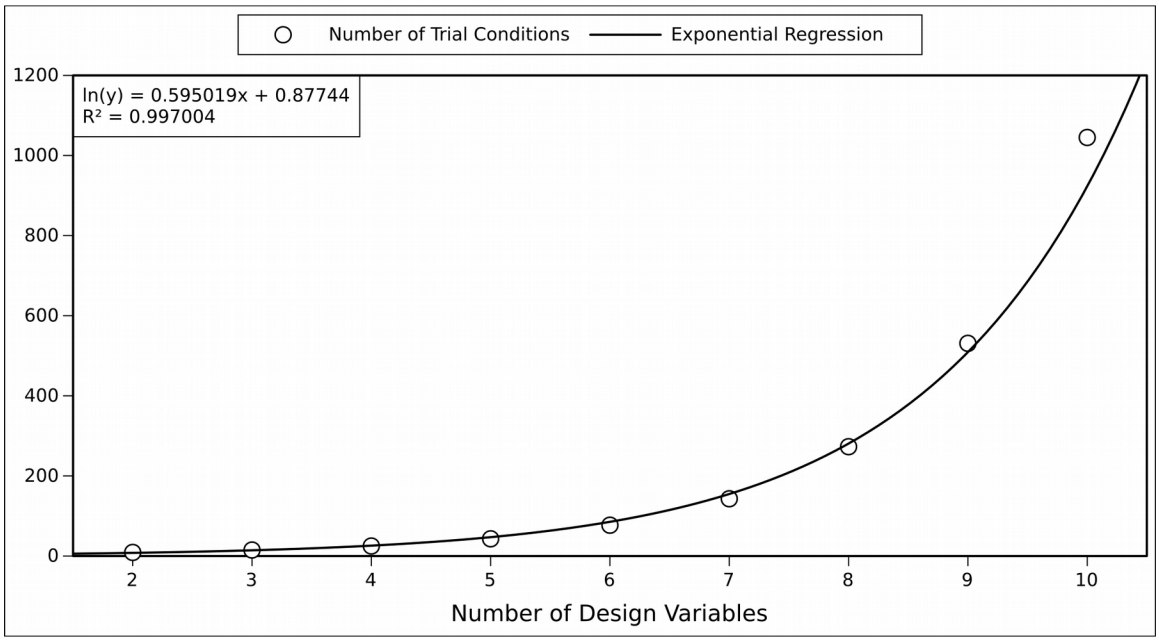

Figure 3.4 - The relationship between the number of design variables and the number of trial conditions of an inscribed CCD required for a quadratic regression approximation for up to 10 variables.

CHAPTER 4 CASE STUDY RESULTS

In the previous chapter, we introduced the *Search-and-Zoom* algorithm for solving robust optimal design problems. In this chapter, we will demonstrate the algorithm's computational benefit by applying it to two design problems from the communications/radio (RF) industry. The first problem is for a high-frequency micro-strip filter and the second is a rectangular patch antenna. It is helpful, first, to understand some aspects of RF design, which we will discuss now.

4.1 Introduction to RF Design

The purpose of products in RF design is to manage the transmission of electromagnetic (EM) waves that propagate through materials and space. When an EM wave impacts a material, the signal can either reflect back or continue to propagate through the material. The effect that a material has on this reflection or propagation is dependent upon material properties and the geometry of the material.

These effects are modeled in the form of a *transmission line model*, which helps determine how a material's *in*-port and *out*-port behave when an EM wave is introduced at each port. Figure 4.1 shows a 2-port network that has four metrics, called scattering parameters, or *S-parameters*. The notation S_{ij} refers to the S-parameter of the i^{th} out-port and the j^{th} in-port of the material. Thus, S_{21} refers to the proportion of the EM wave that enters port 1 and exits from port 2 and S_{11} is the amount of the signal that enters at port 1 and is reflected

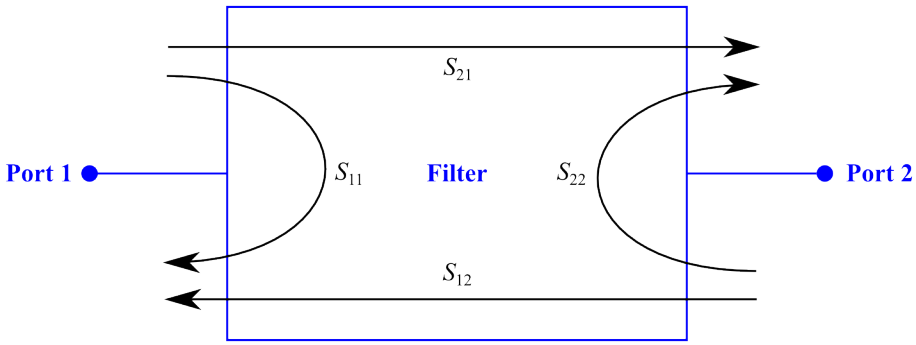

Figure 4.1 - Scattering parameters for 2-port RF network.

back out of port 1. When an RF network is symmetric, like the micro-strip filter presented later in Section 4.2, it generally behaves symmetrically as well, which means $S_{11}=S_{22}$ and $S_{21}=S_{12}$. The only possible values for S-parameters are between 0 and 1, on a linear scale, but they are usually reported in decibels (dB) with the transformation $y_{dB} = 20 \log_{10}(y_{linear})$. The basic theory for how S-parameters are calculated can be found in any introductory Microwave RF textbook, so this will not be addressed here. In the case studies in this chapter, these values are calculated automatically by simulation software.

RF products, in general, are designed to work at one of two conditions: at a specific, target EM frequency (like 2.4 GHz) or over a frequency range (like 12GHz to 14GHz in the Ku band). In the case studies below, the micro-strip filter is designed to work over a frequency range and the patch antenna is designed to work at a target frequency. In addition to *frequency* requirements, the patch antenna, as with all other antennas, also has spatial, or *angular*, requirements that define an acceptable transmission profile for the EM waves as they propagate through space away from the antenna.

4.2 Case Study 1 – High Frequency Micro-strip Band-pass Filter

4.2.1 Design Background

The first case we will present is a high-frequency micro-strip band-pass filter. To create the micro-strip transmission line, metal is chemically deposited onto a dielectric substrate material. Then the geometry is formed using a manufacturing process called photo-lithography, which uses light to etch away the unwanted metal strip geometry. A cross-sectional view is shown in Figure 4.2. The precision of the photo-lithography process is highly dependent on factors that control the focus of the light and the duration the light is allowed to etch away material.

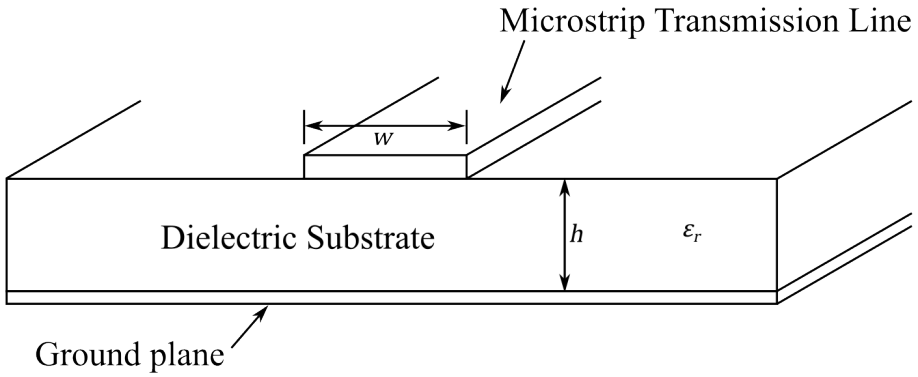

Figure 4.2 - Cross-section view of micro-strip transmission line.

In this case study, we will assume that the manufacturing tolerance for the photo-lithography process is approximately ±0.1 mils (1/10,000 inch). We will see that even with this kind of precision, which is beyond the capability of many manufacturing processes, the performance of the filter can be sensitive to very high frequency EM waves. Figure 4.3 shows a bird's-eye view of the filter's topology. Even though the metal strips are not in physical contact with each other, the EM waves can propagate when the strips are caused to electrically resonate. This is one of the design challenges of micro-strip devices—to determine the

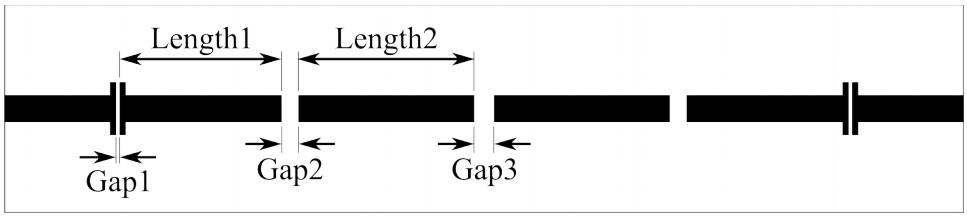

Figure 4.3 - Topology of micro-strip band-pass filter.

appropriate geometry that controls the strips' resonant frequencies.

A band-pass filter normally has at least four performance metrics based on the four major constraint zones of influence for the filter. Figure 4.4 shows where these four zones apply, corresponding to the respective frequency ranges. These zones define the desired behavior of certain S-parameters. For example, S_{21} has a zone *below* the pass-band frequencies (*S21 lowstop*) which is minimized or constrained to be below some minimum value, one *within* the pass-band frequencies (*S21 pass*) which is maximized, and one *above* the pass-band frequencies (*S21 highstop*) which is also minimized or constrained, as shown in Figure 4.4a. Within the pass-band frequencies, EM waves are supposed to be able to propagate through the filter. Outside the pass-band, EM waves are not supposed to propagate. The corresponding S_{11} *within* the passband frequencies (*S11 pass*) is constrained to be as small low as possible, which

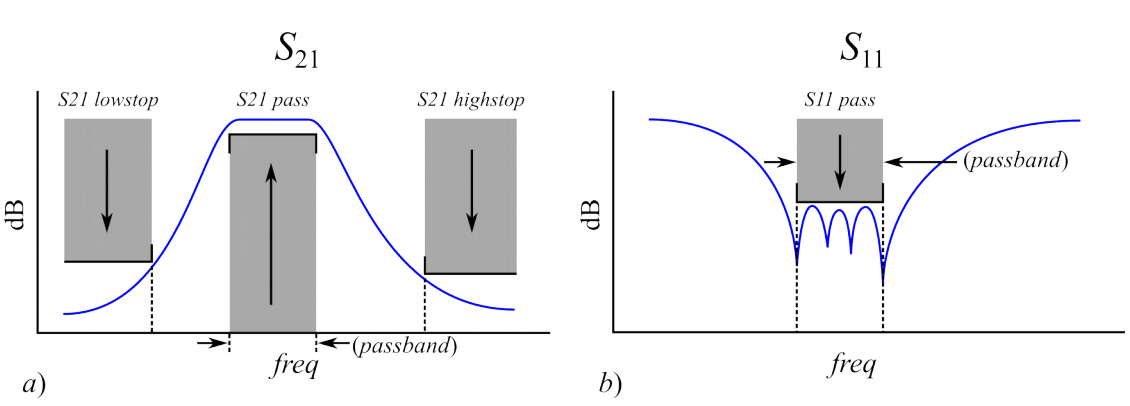

Figure 4.4 - Typical band-pass filter constraint zones.

correlates to better transmission of the EM waves. Each section of the filter provides a filtering effect at some target frequency. By combining the sections in a configuration like Figure 4.3 we can then create a filtering effect over a frequency range.

The dimensions of the strips are defined with five design variables, as well as values for some other dimensions (assumed constant), as shown in Figure 4.3. These five variables, with their respective nominal value and lower and upper bounds (all in units of mils), are:

Table 4.1 – Micro-strip band-pass filter design variables and variable bounds

Variable Name	Nominal Value	Lower Bound	Upper Bound
Length1	62.2	61.0	64.0
Length2	67.5	67.0	68.0
Gap1	1.7	1.0	2.5
Gap2	6.8	6.0	8.5
Gap3	8.0	7.0	10.0

4.2.2 Constraint Formulation

We now need to define suitable constraint and objective functions for the optimization problem. Since the filter is designed to deal with EM waves over a frequency range, creating a multiplicity of output values, we need a metric that can account for each of the above metrics at each sampled frequency point within their respective ranges. Equation 4.1 shows the formulation for constraint functions used in this thesis:

$$Y_{con,j} = \min_{i \in n} \left(\alpha_j \frac{y_{i,j} - y_{d,j}}{|y_{d,j}|} \right) \tag{4.1}$$

where α_j is −1 for less-than constraints and +1 for greater-than constraints, $y_{i,j}$ is the constraint function value at the i^{th} frequency point of the j^{th} constraint over n frequency points, and $y_{d,j}$ is

the j^{th} constraint RHS. The actual values for y are normalized by the respective reference constraint value since values for y are given in units of dB. By inspection, we see that Equation 4.1 calculates the *worst* value within the j^{th} constraint's frequency range, where feasibility is defined by $Y_{con,j} \geq 0$ (i.e., more negative values indicate greater constraint violation and more positive values indicate greater design constraint feasibility). For example, if we have the constraint $Y_{con,1} \leq -5$ (i.e., $\alpha = -1$, $y_{d,1} = -5$) and $y_{i,1} = \{-3, -5.1, -6.2\}$ and , then we calculate:

$$\begin{aligned} Y_{con,1} &= min\left((-1)\cdot\frac{\{-3,-5.1,-6.2\}-(-5)}{|(-5)|}\right) \\ &= min\{-0.4,0.02,0.24\} \\ &= -0.4 \end{aligned} \tag{4.2}$$

We see that the function for $Y_{con,1}$ determined the worst value of the set. Since the result is a negative value, we know that the constraint is violated for that data point.

4.2.3 Objective Formulation

Equation 4.3 is an alternative form that is useful for representing an objective function that applies over a range of values:

$$Y_{obj} = \frac{\alpha}{n}\sum_{i=1}^{n}\left(\frac{y_i - y_d}{|y_d|}\right) \tag{4.3}$$

It calculates the average scaled value of all output values within the objective's frequency range. For example, if the above three values were used, we would get the following for the objective function, which we then minimize. For example, using the same values as above:

$$
\begin{aligned}
Y_{obj} &= \frac{(-1)}{3}\sum\left(\frac{\{-3,-5.1,-6.2\}-(-5)}{|(-5)|}\right) \\
&= -\frac{1}{3}\sum\{0.4,-0.02,-0.24\} \\
&= -0.047
\end{aligned}
\tag{4.4}
$$

This result provides an average value over the sampled points, normalized by the reference value of −5. The nominal optimization model for this design problem is therefore defined as follows:

Maximize $f = Y_{obj}\,(S21\ pass)$
subject to $g_1 = Y_{con,1}\,(S21\ lowstop \le -45\ \text{dB})$
$g_2 = Y_{con,2}\,(S21\ pass \ge -3\ \text{dB})$
$g_3 = Y_{con,3}\,(S21\ highstop \le -40\ \text{dB})$
$g_4 = Y_{con,4}\,(S11\ pass \le -10\ \text{dB})$

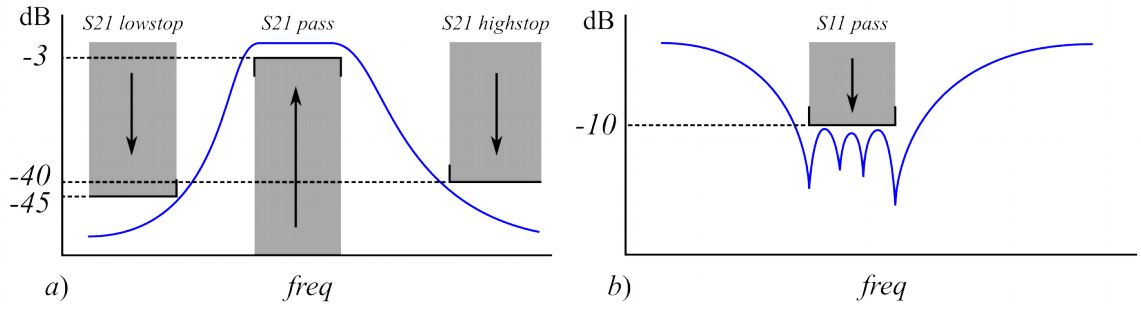

Figure 4.5 – Actual band-pass filter optimization constraint zones.

For the case studies, the nominal optimum was found using NLP methods. Then, in order to reduce the variability in filter performance and keep *S21 pass* high across the band-width, both NLP methods (e.g. Matlab's fmincon solver with the SQP algorithm) and the *Search-and-Zoom* algorithm were used to perform a sensitivity optimization on the filter, including optimizing for both RSS and WC feasibility using the manufacturing tolerances in

Section 4.1. The simulations were performed using the software *Advanced Design System* (ADS) 2011. All optimization results are given in Table 4.2.

4.2.4 Optimization Results

The nominal optimum was found, using fmincon, at the design:

Table 4.2 - Nominal optimum of micro-strip band-pass filter

Design Variable	Optimal Value	Design Function	Optimal Value
Length1	62.9497	Y_{obj}(*S21 pass*)	0.9984
Length2	67.3525	*S21 lowstop*	−45.0000
Gap1	1.6546	*S21 pass*	−0.0428
Gap2	8.1796	*S21 highstop*	−40.4556
Gap3	9.8275	*S11 pass*	−20.0894

which improved Y_{obj}(*S21 pass*) from 0.6572 to 0.9984. This correlates to an improvement in *S21 pass* from −2.5479 dB to −0.0428 dB, which is nearly a 2x improvement on a linear scale (a perfect filter's *S21 pass* is 0 dB). However, since *S11 pass* exhibited the most transmitted variation at the nominal optimum ($\sigma_{S11\ pass}$=0.0666), this was chosen to be the objective for the sensitivity optimization, with the following model:

Minimize $f = \sigma_{S11\ pass}$
subject to $g_1 = S21\ lowstop \leq -45$ dB
$g_2 = S21\ pass \geq -3$ dB
$g_3 = S21\ highstop \leq -40$ dB
$g_4 = S11\ pass \leq -10$ dB

Because we already have a constraint on *S21 pass*, a new constraint is not necessary.

There were three steps in the sensitivity optimization: first, minimize the transmitted variation of *S11 pass* ($\sigma_{S11\ pass}$), then do one additional step for adding feasibility robustness of

RSS tolerances and one additional step for WC tolerances. As shown in Table 4.3, we can see that the variation in *S11 pass* was reduced from 0.0666 to 0.0162 (fmincon) and 0.0182 (*Search-and-Zoom*), nearly a 75% decrease in sensitivity in both cases. Then, after adjusting the constraints for RSS feasibility, we see that the sensitivity objective (and the original) changed only slightly with a large jump in feasibility (from 80% to 100%). The Search-and-Zoom algorithm actually improved slightly, partially because it didn't have any binding constraints to begin with at the sensitivity optimum. Similar results were achieved for WC feasibility.

We see that there is considerable difference between the number of required actual simulator calls between using fmincon and using *Search-and-Zoom*. Starting with the same design as the nominal optimization, fmincon required 1749 calls to the simulator while *Search-and-Zoom* required only 559—a 68% reduction in computational effort—to reach the sensitivity optimum ($\sigma_{S11\ pass}$ = 0.0182). We note that even though the starting design has a comparable objective value, it is unacceptable because the constraints are not satisfied.

Of the two feasibility optimization steps, we see that *Search-and-Zoom* took just under 1000 simulator calls less than fmincon for RSS constraints—a 61% reduction—and over 1800 simulator calls less than fmincon for WC constraints—an 89% effort reduction in computational effort. The feasibility is also approximately what we would hope from the shifted constraints, even though the WC constraints weren't enough to drive to 100% feasibility. The feasibility was estimated using 4,000 samples in a Monte Carlo simulation based on the original constraints.

Comparing the actual Monte Carlo simulation data of the Nominal Optimum and the RSS and WC Feasible Sensitivity Optimums, we see that the variation of *S11 pass* has been successfully reduced, as shown in Figure 4.6. The x-axis is in linear units, which makes the constraint RHS value of −10dB ≈ 0.316, so the majority of the histogram should lie below this value, and each histogram does.

Table 4.3 - Sensitivity optimization results for micro-strip band-pass filter. Binding constraints are in italics.

			Sensitivity Optimum (SO)		SO with RSS Feasibility		SO with WC Feasibility	
Parameter	Starting Design	Nominal Optimum	fmincon	*Search-and-Zoom*	fmincon	*Search-and-Zoom*	fmincon	*Search-and-Zoom*
Length1	62.2000	62.9497	63.0236	62.9779	63.0027	63.0927	63.0438	63.0748
Length2	67.5000	67.3525	67.2924	97.3549	67.2685	67.3545	67.3244	67.3603
Gap1	1.7000	1.6546	2.1257	1.7790	2.1695	2.4301	2.1500	2.2732
Gap2	6.8000	8.1796	7.8920	8.1841	8.0925	8.5000	8.2125	8.5000
Gap3	8.0000	9.8275	9.9198	9.7776	9.9527	10.0000	9.9920	10.0000
Y_{obj}(*S21 pass*)	0.6572	0.9984	0.9403	0.9975	0.9621	0.9750	0.9752	0.9864
$\sigma_{S11\ pass}$	0.0183	0.0666	0.0162	0.0182	0.0167	0.0180	0.0171	0.0180
S21 lowstop	−38.1974	***−45.0000***	−45.7390	−45.2467	−46.9662	−48.8475	−46.9945	−48.2940
S21 pass	−2.5479	−0.0428	−0.4573	−0.3538	−0.3225	−0.2207	−0.2285	−0.1248
S21 highstop	−29.8537	−40.4556	−40.3262	−40.6974	***−40.8552***	−43.5627	***−41.7989***	−43.2723
S11 pass	−3.5279	−20.0894	***−10.0021***	−23.1251	***−11.4530***	−13.0506	***−12.9027***	−15.4783
Actual Simulator Calls	--	356	1749	559	1419	559	2024	215
RSS Feasibility	--	--	40%	80%	99%	100%	--	--
WC Feasibility	--	--	29%	61%	--	--	99%	99.8%
Constraint RHS	Direction							
S21 lowstop (dB)	≤	−45	−45	−45	−45.8371	−45.8782	−46.9195	−47.0198
S21 pass (dB)	≥	−3	−3	−3	−2.8031	−2.9532	−2.6403	−2.9177
S21 highstop (dB)	≤	−40	−40	−40	−40.8552	−40.7948	−41.7989	−41.6678
S11 pass (dB)	≤	−10	−10	−10	−11.4529	−11.6506	−12.9026	−13.1555

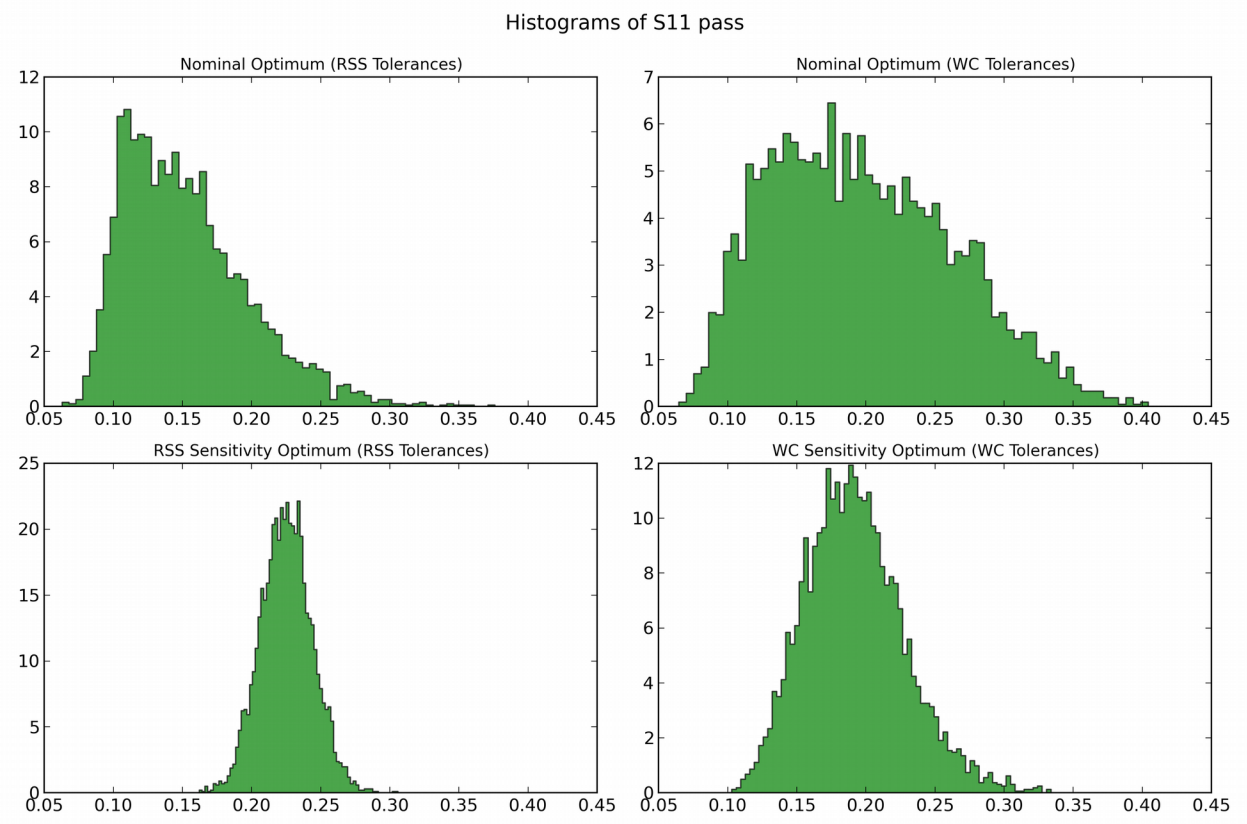

Figure 4.6 - Monte Carlo histograms of *S11 pass* at filter *Search-and-Zoom* nominal optimum (top row) and sensitivity optimums (bottom row) using RSS tolerances (left column) and WC tolerances (right column). Values less than 0.316 are "feasible".

4.3 Case Study 2 – Rectangular Patch Antenna

4.3.1 Design Background

We now present the second case—a rectangular patch antenna. Similar to the filter above, a metal patch is cut to shape using any suitable process and then mounted to a dielectric slab and electrically connected via a feed wire through the back of the dielectric. A grounding plane is mounted on the opposite side of the dielectric and is insulated from the feed wire, as shown in Figure 4.7. When an EM wave makes contact with the metal patch through the feed, the patch resonates and radiates the EM wave away from the dielectric into the surrounding medium. The geometry of the patch and the location of the feed determines the frequency at

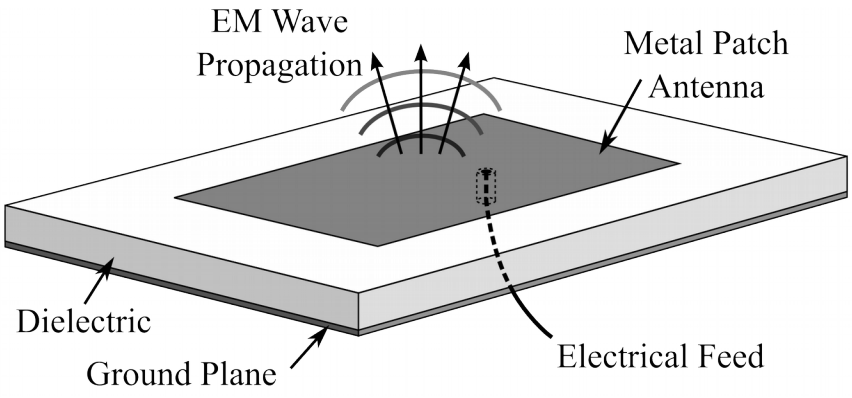

Figure 4.7 - Components of a simple patch antenna design.

which the patch will resonate.

In this case study, we will use the four design variables shown in Figure 4.8 to optimize a patch antenna for maximum transmission at a target frequency of f_0 = 2.98 GHz. The nominal values, as well as the lower and upper bounds for each design variable, are listed in Table 4.4. Like the filter in Section 4.2, this device also has an S-parameter constraint. To attain the largest proportion of the signal out of the patch, we will constrain the output value of S_{11} *@ 2.98 GHz*. A typical frequency response curve for a patch antenna's S_{11} is shown in Figure 4.9. Since antennas radiate into the surrounding medium, it is common to also have directional radiation constraints. When an antenna is designed to broadcast in all directions (e.g., omni-directional, like from a radio tower), the constraints are less restrictive. When an antenna is designed to transmit a signal in a specific direction only, this is called a *directional antenna* (like a satellite dish antenna). Directional antennas are designed for a particular radiation profile that maximizes the energy transmitted in the desired directions and minimizes the energy transmitted in all other unwanted directions.

The patch antenna in this case study is a kind of directional antenna. Figure 4.10 shows a 360° slice of a directional antenna's radiation *gain* profile, measured in dB. We notice two main features of the gain profile. The *main beam* is the region that is maximized (specifically,

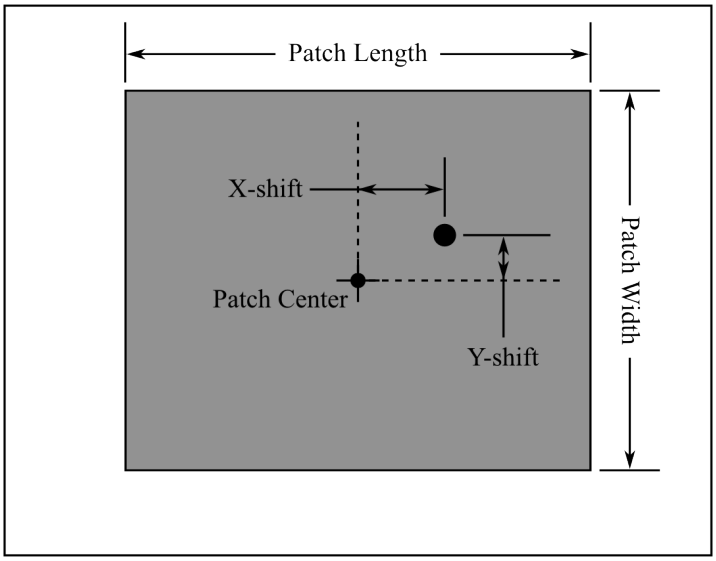

Figure 4.8 - Design variables for a rectangular patch antenna

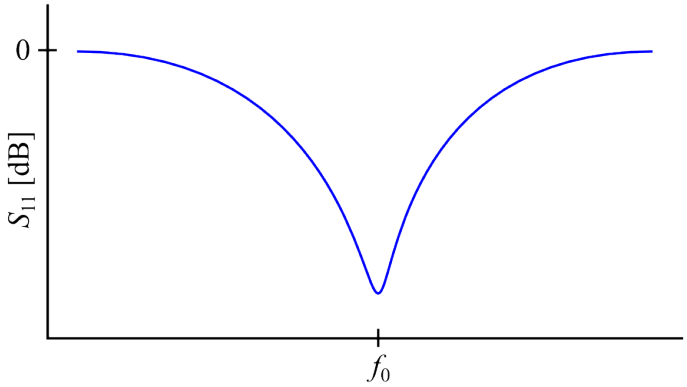

Figure 4.9 – Typical S_{11} response curve for a patch antenna with target frequency, f_0

Table 4.4 - Patch antenna design variables and variable bounds (all in cm).

Variable Name	Nominal Value	Lower Bound	Upper Bound
Patch Length	3.0	1.75	3.5
Patch Width	3.0	1.75	4.0
X-shift	0.5	0.0	1.5
Y-shift	0.5	0.0	0.8

the peak value), where the desired transmission peak is located at 0°. On either side of the main beam are numerous *side lobes*. Side lobes are radiation in undesired directions, which can never be completely eliminated. Thus, they are designed so that the maximum side lobe is at a minimum acceptable level.

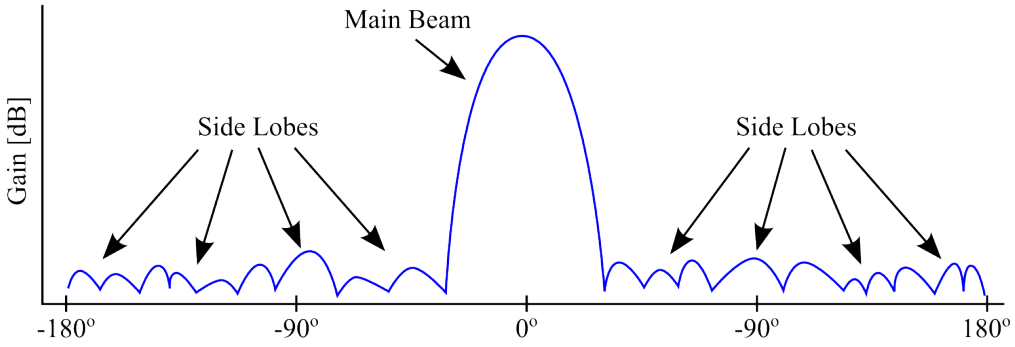

Figure 4.10 - Typical gain profile for a directional antenna.

The patch antenna in this case study thus has the following design goals:

Maximize f = *Peak Main Beam Gain* (dB)
subject to $g_1 = S_{11}$ *@ 2.98 GHz* ≤ -10 dB
g_2 = *Max Side Lobe Level* ≤ -15 dB

4.3.2 Optimization Results

We proceed with the sensitivity optimizations in a similar manner as the filter in Section 4.2, assuming a general tolerance of ±0.1 cm for each design variable. The nominal optimization was first carried out to give a baseline design using fmincon, resulting at the following design:

Table 4.5 - Nominal optimum of rectangular patch antenna

Design Variable	Optimal Value	Design Function	Optimal Value
Patch Length	3.1446	f(*Peak Main Beam Gain*)	9.2651
Patch Width	4.0000	g_1(S_{11} *@ 2.98 GHz*)	−9.9999
X-shift	0.8423	g_2(*Max Side Lobe Level*)	−18.7387
Y-shift	0.1589		

At this design, we see an improvement in f(*Peak Main Beam Gain*) from 6.0496 to 9.2651 dB. However, similar to the filter problem, this design was most sensitive to the S_{11} constraint (g_1) at the nominal optimum ($\sigma_{S11\ @\ 2.98\ GHz} = 0.00693$). Thus, we chose to formulate the sensitivity optimization model by making the transmitted variation of S_{11} *@ 2.98 GHz* the objective and also add a constraint to the *Peak Main Beam Gain* to keep it high, if possible:

Minimize $f = \sigma_{S11\ @\ 2.98\ GHz}$
subject to $g_1 = S_{11}$ *@ 2.98 GHz* ≤ -10 dB
$g_2 =$ *Max Side Lobe Level* ≤ -15 dB
$g_3 =$ *Peak Main Beam Gain* ≥ 8.75 dB

Both RSS and WC feasibility optimizations were also performed on the sensitivity optimization model. The feasibility of all six cases was confirmed using 500 Monte Carlo simulations at each optimum. The number of simulations was smaller than the filter problem in Section 4.2 because each call to the antenna simulator took much more computation time than with the filter simulator. Table 4.6 gives the results of the three sensitivity optimizations, comparing the efficiency of fmincon and *Search-and-Zoom*.

We see that the sensitivity optimum must be relatively close to the nominal optimum because none of the subsequent design variables changed significantly and the original objective maintained a high margin above the g_3 constraint. The main difference between the two algorithms is seen in the amount of required simulator calls, with *Search-and-Zoom* only

Table 4.6 - Sensitivity optimization results for the patch antenna. The binding constraints are in *italics*.

			Sensitivity Optimum (SO)		SO with RSS Feasibility		SO with WC Feasibility	
Parameter	Starting Design	Nominal Optimum	fmincon	*Search-and-Zoom*	fmincon	*Search-and-Zoom*	fmincon	*Search-and-Zoom*
Patch Length	3.0000	3.1486	3.1437	3.1106	3.1400	3.1101	3.1437	3.0971
Patch Width	3.0000	4.0	4.0	3.9994	3.9950	3.9130	4.0	3.9535
X-shift	0.5000	0.8423	0.8386	0.7123	0.8235	0.7089	0.8264	0.6534
Y-shift	0.5000	0.1589	0.2886	0.1522	0.2836	0.1519	0.2564	0.1519
$\sigma_{S11 @ 2.98\ GHz}$	--	0.020822	0.008385	0.01153	0.008684	0.01156	0.008665	0.01429
S_{11} *@ 2.98 GHz* (dB)	−2.1696	***−9.9999***	***−10.0025***	−14.8018	−10.3436	−13.5993	***−10.3618***	−17.3586
Side Lobe Level (dB)	−206.0496	−18.7387	−18.7243	−18.7059	−18.7129	−18.6800	−18.7216	−18.6447
Max Main Beam Gain (dB)	6.0496	9.2651	9.2628	9.2212	9.2576	9.1988	9.2587	9.1824
Actual Simulator Calls			1880	305	1251	65	1637	365
RSS Feasibility			32.6%	100.0%	96.4%	100%	–	–
WC Feasibility			23.2%	100.0%	–	–	71.8%	100.0%
Constraint RHS	Direction							
S11 @ 2.98 GHz (dB)	≤	−10	−10	−10	−10.2334	−10.3225	−10.3618	−10.4597
Side Lobe Level (dB)	≤	−15	−15	−15	−15.0102	−15.0099	−15.0148	−15.0160
Max Main Beam Gain (dB)	≥	--	8.75	8.75	8.7575	8.7582	8.7602	8.7612

requiring 305 actual simulator calls to drive to the sensitivity optimum, where fmincon required 1880 actual simulator calls—a 84% reduction in computation cost. However, it should be noticed that the sensitivity optimum found by *Search-and-Zoom* ($\sigma_{S11\ @\ 2.98\ GHz}$=0.0115) is not as good a design as that found by fmincon ($\sigma_{S11\ @\ 2.98\ GHz}$=0.0087). Once again, the *Search-and-Zoom* algorithm likely terminated too early, indicated by the lack of binding constraints at any of its optimums. And although the percent feasibility at each optimum is better than fmincon's, none of the $\sigma_{S11\ @\ 2.98\ GHz}$ values are as good.

Using the Monte Carlo simulation data, we can build histograms to compare the variation in S_{11} *@ 2.98 GHz* at the starting and final designs. In Figure 4.11 we see a reduction in variability, comparing the RSS and WC tolerances at the nominal optimum and the RSS and WC feasibility sensitivity optimums.

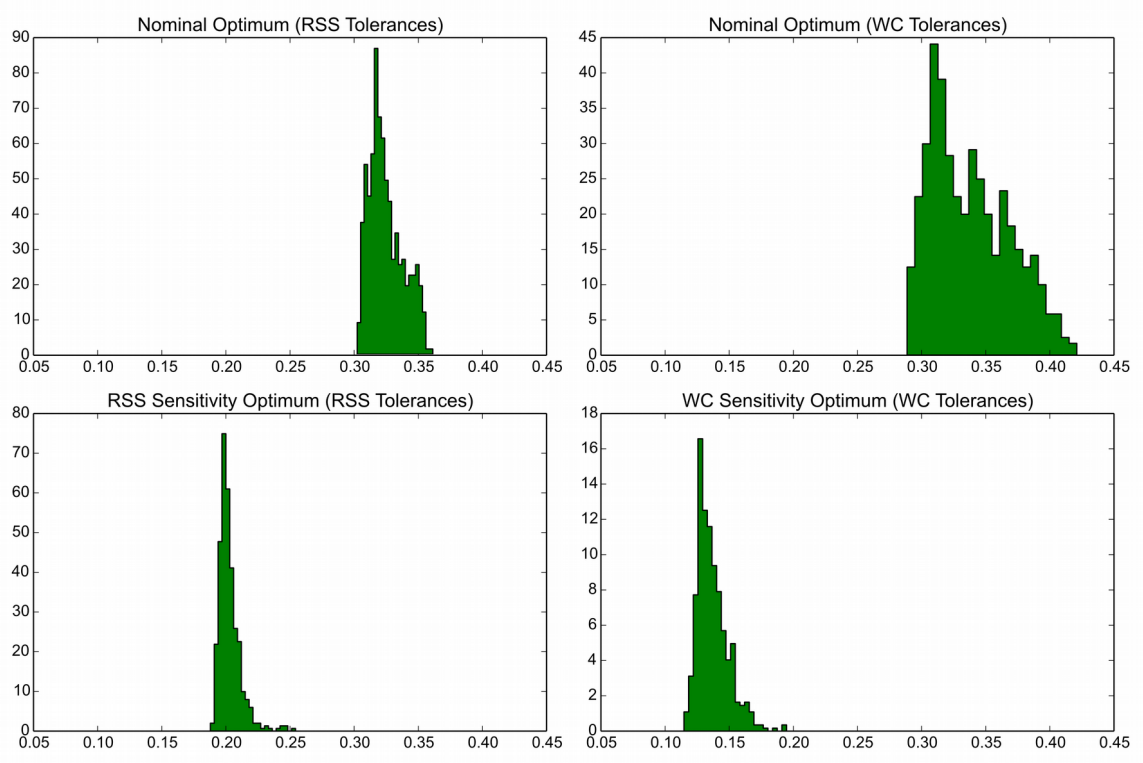

Figure 4.11 – Monte Carlo histograms of S_{11} *@ 2.98 GHz* at patch antenna *Search-and-Zoom* nominal optimum (top row) and sensitivity optimums (bottom row) using RSS tolerances (left column) and WC tolerances (right column).

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In this thesis, our primary objective was to apply robust design methodology to the communications (RF/EM) industry. As a result, we have shown how to develop robustness against variation, wherever it comes from, be it material properties, manufacturing tolerances, etc., while maintaining good nominal performance. Two main methods were explored that are commonly used for this purpose: Taguchi Methods and Nonlinear Programming. Taguchi Methods, though they can be efficient, proved difficult to work with since they couldn't handle constraints appropriately. Nonlinear Programming provided the accuracy and flexibility to formulate any problem with constraints, but suffered from potential excessive computation cost. This led to research into a hybrid method that combined the efficiency of Taguchi Methods with the flexibility and accuracy of Nonlinear Programming. The result is called the *Search-and-Zoom* algorithm.

It is shown that the *Search-and-Zoom* algorithm for robust optimal design can be used as a computationally cost-effective method to optimize for sensitivity objectives for cases with a small number of variables. In the two case studies in Chapter 4, the algorithm found designs that were comparable to, though not as good as, those found using Matlab's fmincon (SQP) algorithm, with a significant reduction in computational cost. When optimizing for statistical (RSS) feasibility, both algorithms successfully found designs that provided over 99% feasibility,

but the computation requirement of the *Search-and-Zoom* algorithm required, on average, 78% less calls to the actual simulator. For all optimization cases (Sensitivity, Sensitivity with RSS Feasibility, and Sensitivity with WC Feasibility), the average reduction in computation cost was 79%, ranging from 61% to 95%. Although this thesis does not exercise the algorithm on more cases to prove the consistency of such reductions, having applied it on two different kinds of problems gives us confidence that this algorithm can be effective at reducing the computation cost of sensitivity optimization design problems.

It is also shown that a confirmation set of Monte Carlo simulation calls verified the predicted amount of transmitted variation when optimizing for feasibility robustness for both objective and constraint functions. During the optimization of the patch antenna, for example, the WC tolerances at the fmincon sensitivity optimum for WC feasibility caused many more *infeasible* designs than expected—approximately 28% (or 71.8% feasibility, as shown in Table 4.4). We expected that number to be close to zero. If the infeasibility were close to zero, we could still accept the design, but this result may hint to more nonlinearity in the model's tolerance region than the first-order transmitted variation equation takes into account. We could use a higher-order tolerance model for more accurate estimation, but this would further increase the computational expense. We could use Stochastic optimization techniques for more accurate feasibility estimates, but this would also add to the computational cost with hundreds or thousands of extra simulation calls that we are trying to avoid here.

5.2 Future Work

The *Search-and-Zoom* algorithm appears to work well when a design problem has a small number of design variables, but more work should be done to determine its effectiveness when the number of design variables is increased. As was shown in Chapter 3, when using

Central-Composite experimental designs to construct the response surface, the number of unique experiments required tends to increase exponentially, which may mean that there may be a limit when the cost of running the experiments becomes prohibitive, but this surely is also dependent upon the actual computation cost of running the actual simulations. We think that there will likely be a point of intersection between the effort required for NLP methods and the *Search-and-Zoom* algorithm. Other experimental designs such as *saturated* second-order designs, etc. may provide an even greater reduction in computational cost than the full Central-Composite designs (like those used in this thesis), but may not provide suitable regression approximations which are necessary for estimating the derivatives in the transmitted variation equation.

Another way to construct response surfaces is by using *space-filling* designs (e.g., *Kriging* methods) for modeling data. These have the benefit of not requiring any specific structure or polynomial "order" within the underlying experiments, but, just like Monte Carlo simulation, becomes more accurate as the number of experiments increases. Using Kriging methods when a normal response surface experimental design becomes too costly may prove beneficial, but was not explored.

A method that is rising in popularity for calculating derivatives is called *automatic differentiation*. This method has the accuracy of symbolic differentiation, but with the computational effort of numerical differentiation. Using a knowledge of the calculation steps and how derivatives propagate through them using the chain rule, derivatives of any arbitrary order can be calculated to machine precision. This, however, requires access to the source code functions and doesn't work with "black-box" functions. In this thesis, since we used commercial applications to perform the simulations, this wasn't an available option.

A comparison with other optimization routines could also be informative. The SQP algorithm was chosen due to its well-known efficiency, but it is not suitable for all kinds of optimization problems. An understanding of how other optimization algorithms, such as *Interior-Point method*, *Genetic Algorithms* (or other global-optimization routines), etc., could show that the *Search-and-Zoom* algorithm is more useful only for a few select cases or that it is generally more useful than other more common routines.

The focus of the application of the *Search-and-Zoom* algorithm in this thesis has been specifically within the RF industry. Further study could be done to understand the industrial applicability of the algorithm in structural, fluid dynamics, heat transfer, etc. design applications. As the algorithm has been effective in RF design, we expect it to also prove effective in other industries to reduce computation time, while yielding robust designs.

REFERENCES

Advanced Design System (ADS) 2011. Agilent Technologies, Santa Rosa, California, United States. http://www.keysight.com/find/eesof-ads.

Agarwal M. M. (1981). "Optimal Synthesis of Tolerance and Clearance in Function Generating Mechanism—A Parametric Programming Problem." *ASME Proceedings*, 81-DET-5, pp. 1-5.

Balling R. J., Free J. C., Parkinson A. R. (1986). "Consideration of Worst Case Manufacturing Tolerances in Design Optimization,. *ASME J. of Mechanisms, Transmission and Automation in Design*, Vol. 108, No.4, p. 438.

Bates R. A., Kenett R. S., Steinberg D. M., Wynn H. P. (2006). "Achieving robust design from computer simulation." *Quality Technology & Quantitative Management*, Vol. 3, pp. 161-177.

Beohar S. B. L. and Rao A. C. (1980). "Optimum Stochastic Synthesis of Four Bar Spatial Function Generators." *ASME Proceedings*, 80-DET-32, pp. 1-5.

Buyske S. and Trout R. (2000). "Robust design and Taguchi methods." Lecture notes 10 of International conference on parametric study, New York, pp. 22-26.

Derringer G. and Suich R. (1980). "Simultaneous Optimization of Several Response Variables." *Journal of Quality Technology*, Vol. 12, No. 4.

Eggert R. J. and Mayne R. W. (1990). "Probabilistic Optimization Using Successive Surrogate Probability Density Functions." *Proc. ASME 16th Design Automation Conf.*, Chicago, IL, DE-Vol. 23-1, p. 129.

Giovagnoli A., Romano D. (2008). "Robust design via simulation experiments: A modified dual response surface approach." *Quality and Reliability Engineering International*, Vol. 24, pp. 401-416.

Gunawan S., Azarm S. (2005). "A feasibility robust optimization method using sensitivity region concept." *Journal of Mechanical Design*, No. 5, pp. 858-865.

Ku K. J., Rao S. S., and Chen L. (1998). "Taguchi-Aided Search Method for Design Optimization of Engineering Systems." *Engineering Optimization*, Vol. 30, No. 1, pp. 1-23.

Lee K. H., Park G. J. (2001). "Robust optimization considering tolerances of design variables." *Computers and Structures*, Vol. 79, pp. 77-86.

Lehman J. S., Santner T. J., Notz WI (2004). "Designing computer experiments to determine robust control variables." *Statistica Sinica*, Vol. 14, pp. 571-590.

Lewis L. and Parkinson A. (1994). "Robust optimal design using a second-order tolerance model." *Research in Engineering Design*, Vol. 6, pp. 25-37.

MATLAB and the Optimization Toolbox Release 2013b. The MathWorks, Inc., Natick, Massachusetts, United States. http://www.mathworks.com.

Michael W. and Siddall J. N. (1982). "The Optimal Tolerance Assignment with Less Than Full Acceptance." *Trans. ASME J. of Mech. Design*, vol. 104, pp. 855-860.

Myers R. H., Montgomery D. C. (2011). *Response Surface Methodology*, John Wiley & Sons: New York.

Parkinson, A. R., Sorensen C., and Pourhassan N. (1993). "A General Approach for Robust Optimal Design." *J. Mech. Des.*, Vol. 115, pp. 74-80.

Phadke M. S. (1989). *Quality Engineering using Robust Design*, Prentice Hall, Englewood Cliffs, New Jersey.

Rao S. S. (1979). *Optimization Theory and Applications*, John Wiley Eastern Limited.

Rao S. S. (1986b). "Automated Optimum Design of Wing Structures: A Probabilistic Approach." *Computers & Structures*, Vol. 24, No.5, pp.799-808.

Rhyu J. H. and Kwak B. M. (1988). "Optimal Stochastic Design of Four-Bar Mechanisms for Tolerance and Clearance." *J. of Mech., Trans., and Automation in Design*, Vol. 110, pp. 255-262.

Sacks J., Welch W., Mitchell T. J., Wynn H. P. (1989). "Design and analysis of computer experiments." *Statistical Science*, Vol. 4, pp. 409-435.

Sundaresan S., Ishii K., and Houser D. R. (1991). "Design Optimization for Robustness Using Performance Simulation Programs." *Advances in Design Automation*, DE-Vol. 32-1, pp. 249-256.

Taguchi G. (1987). *System of Experimental Design*, UNIPUB Kraus International Publications, New York.

Weng W. C., Yang F., and Elsherbeni A. (2008). *Electromagnetics and Antenna Optimization Using Taguchi's Method*, Morgan & Claypool, San Rafael, California.

Wu C. F. J. and Hamada M. (2000). *Experiments: Planning, analysis, and parameter design optimization*, New York: Wiley-Interscience.

Wu F. C. (2008). "Robust design of nonlinear multiple dynamic quality characteristics." *Computers & Industrial Engineering*, 56, pp 1328-1332.