Robust Optimization:
Design in MEMS

by

Peter Josef Sedivec

B.S. (University of California, Berkeley) 2000

A report submitted in partial satisfaction of the
requirements for the degree of

Master of Science, Plan II

in

Mechanical Engineering

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Andrew Packard, Chair
Professor George Johnson
Professor Panos Papadopoulos

December 2002

Abstract

Robust Optimization:
Design in MEMS

by

Peter Josef Sedivec
Master of Science, Plan II in Mechanical Engineering

University of California, Berkeley

Professor Andrew Packard, Chair

This report introduces a robust design technique for micro-electro-mechanical systems (MEMS) subject to inherent geometric and material uncertainties. Although design has played an important role in MEMS development, little has been done to account for the uncertainties associated with MEMS fabrication. Current fabrication techniques have poor dimensional tolerances and material properties at the micro-scale are not well characterized. The robust design problem we pose is to minimize the expected variance between the actual and target system performance.

After a few assumptions, the robust design problem can be written as a constrained minimization. We consider a subset of these problems and design an algorithm to minimize rational polynomial functions subject to polynomial inequality constraints. In this report we give a detailed outline of the algorithm, verify its performance on two standard optimization test problems.

Finally, we conclude with two MEMS design problems that illustrate our robust design technique and optimization algorithm. These examples show that the robust designs nominally meet the target performance and are significantly less sensitive to geometric uncertainties than typical designs.

Contents

List of Figures

List of Tables

Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 0087038.

Chapter 1

Introduction

Micro-electro-mechanical systems (MEMS) is a new field that integrates mechanical systems and electronics through a silicon substrate using a microfabrication technique [1]. This new field largely emerged after polycrystalline silicon was discovered to have excellent mechanical properties [2]. This realization prompted experimentation of many simple mechanical devices such as the double-folded flexure resonator [3]. The MEMS community was able to borrow substantially from the relatively mature IC industry and as a result grew quickly. The cutting edge of MEMS technology currently incorporates over 100 different materials and numerous fabrication processes. Some examples of current MEMS devices includes 3-axis accelerometers, microfluidics, folding mirrors, biomedical needles, micromotors, optical switches and gyroscopes. A good overview on the field of MEMS can be found in [4].

While uncertainty is inherent in all systems, it is particularly important in MEMS design for several reasons. First, there is the issue of scale. While most MEMS features are on the order of micrometers, typical geometric tolerances for most processes are on the order of tenths of microns [5]. With minimum feature sizes of 2 microns, the relative uncertainty can reach 5% or more which is embarrassing when compared with macroscopic manufacturing techniques.

Second, the materials used in MEMS devices are poorly characterized. This is largely a result of the infancy of the field and the number of new materials and processes being used. We are well aware that metals, plastics, and other material properties are largely dependent on how they are manufactured, similarly the properties of MEMS materials are dependent on how they are fabricated. With a number of developments occurring in the past 15 years, the characterization of mechanical, electrical, and thermal properties has barely scratched the surface for the overwhelming combinations of materials and processes. Further complicating the characterization process is the difficulty of making measurements at the micrometer scale. For these reasons, most microfabrication materials have large uncertainties associated with their properties. For example, Sharpe and Jackson [6] characterized polycrystalline silicon

in the MUMPs® process and measured values of 162 ± 14 GPa for Young's modulus and 1.56 ± 0.25 GPa for tensile strength, which are relative uncertainties of 9% and 16%, respectively.

Finally, there are several other sources of uncertainty such as surface roughness, side-wall etch angle, and grain size. These factors, however, are typically much more complicated and difficult to characterize than the aforementioned two.

Uncertainty is clearly significant at the micro-scale, however it is often neglected in the MEMS design process. In order to design highly reliable systems that can be fabricated with high yields, it will be necessary to account for the effects of uncertainty especially with the current trends to make more complex and integrated systems. In this report we will address a strategy for robust design through optimization. Our focus will be on dimensional and material uncertainties.

Chapter 2

Robust Design Theory

The class of design problems that we will address are those with rational polynomial objective functions subject to polynomial inequality constraints. Mathematically we can describe this set of problems as

$$f(x) = \frac{N(x)}{D(x)} \quad \textit{subject to} \quad g_i(x) \leq 0 \ \textit{for i=1,...,m} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the design vector, $f(x)$ is the objective function, and $g_i(x)$ are the m polynomial constraints. Both $N(x)$ and $D(x)$ are polynomial in x. The objective, $f(x)$, in its most general form, is a function that anyone might endeavor to robustly design in the presence of uncertainty. Typically, $f(x)$ will describe an aspect of a system's performance. The constraints, $g_i(x)$, are the design requirements that have been imposed and must be satisfied.

2.1 Problem Formulation

To robustly design $f(x)$ we need to understand how uncertainty affects the objective function. Let us augment $f(x)$ to include a description of the uncertainty, which we will define to be $f(x, \delta)$ as

$$f(x, \delta) = \frac{N(x, \delta)}{D(x, \delta)} \tag{2.2}$$

where $\delta \in \mathbb{R}^r$ is the uncertainty, and $N(x, \delta)$ and $D(x, \delta)$ are polynomial in x and δ. The dimension of the uncertainty can be any size. When $\delta = 0$, the nominal performance, $f(x, 0)$, is equal to $f(x)$. Later in this chapter we will show several methods of deriving $f(x, \delta)$ from $f(x)$.

The robust design problem that we aim to solve is to minimize the expected value of the squared error between the actual and target performance. We can write this

as

$$\min_x \mathbf{E}\left(f(x,\delta)-\bar{f}\right)^2 \quad \textit{subject to} \quad g_i(x)\leq 0 \tag{2.3}$$

where $\bar{f}$ is the target performance we would like the system to operate at, and the expectation is taken over the random vector δ. We will assume that the uncertainty, δ, can be characterized as a random vector with the following statistics

$$\begin{aligned}\mathbf{E}(\delta) &= 0_{r\times 1}\\ \mathbf{E}(\delta\delta^T) &= \Sigma\in\mathbb{R}^{r\times r}\end{aligned}$$

where Σ is the covariance matrix and is positive semi-definite. If the uncertainties are uncorrelated then Σ is diagonal, otherwise correlation exists when the off-diagonal entries are non-zero.

The problem posed in (2.3) is a difficult optimization to solve in the general. To simplify the problem we chose to approximate $f(x,\delta)$ with a first-order Taylor series expansion in δ as

$$f(x,\delta)\approx f(x,0)+\nabla_2 f(x,0)\delta \tag{2.4}$$

where $\nabla_2 f(x,0)$ is the gradient of $f(x,\delta)$ with respect to δ. Using this approximation we can expand the expression $\left(f(x,\delta)-\bar{f}\right)^2$, from equation (2.3), into

$$\left(f(x,0)-\bar{f}\right)^2+2\left(f(x,0)-\bar{f}\right)\nabla_2 f(x,0)\delta+\nabla_2 f(x,0)\delta\delta^T\nabla_2^T f(x,0) \tag{2.5}$$

Taking the expectation of equation (2.5) yields

$$\left(f(x,0)-\bar{f}\right)^2+2\left(f(x,\delta)-\bar{f}\right)\nabla_2 f(x,0)\mathbf{E}(\delta)+\nabla_2 f(x,0)\mathbf{E}(\delta\delta^T)\nabla_2^T f(x,0) \tag{2.6}$$

By reducing equation (2.6), based on our assumptions about the mean and covariance of δ, we can write

$$\mathbf{E}\left(f(x,\delta)-\bar{f}\right)^2\approx\left(f(x,0)-\bar{f}\right)^2+\nabla_2 f(x,0)\Sigma\nabla_2^T f(x,0) \tag{2.7}$$

Substituting the approximation in (2.7), back into the original design problem posed in (2.3) yields

$$\min_x\left(\left(f(x,0)-\bar{f}\right)^2+\nabla_2 f(x,0)\Sigma\nabla_2^T f(x,0)\right) \quad \textit{subject to} \quad g_i(x)\leq 0 \tag{2.8}$$

Since $f(x,\delta)$ is a rational polynomial, the objective function we wish to minimize in (2.8) is also a rational polynomial in x. Finally, to non-dimensionalize the cost function, we chose to divide through by $\bar{f}^2$. We will refer to the following expression as our robust design problem

$$\min_x\left(\left(\frac{f(x,0)-\bar{f}}{\bar{f}}\right)^2+\frac{1}{\bar{f}^2}\left(\nabla_2 f(x,0)\Sigma\nabla_2^T f(x,0)\right)\right) \quad \textit{subject to} \quad g_i(x)\leq 0 \tag{2.9}$$

Clearly, the expression we wish to minimize has two distinct terms. For notational convenience, we will label the two terms as

$$\begin{array}{rcl} c^2(x) & := & \left(\frac{f(x,0)-\bar{f}}{\bar{f}}\right)^2 \\ s(x,\Sigma) & := & \frac{1}{\bar{f}^2}\left(\nabla_2 f(x)\Sigma\nabla_2^T f(x)\right) \end{array} \tag{2.10}$$

With the above definitions the robust problem posed in (2.9) becomes

$$\min_x \; c^2(x) + s(x,\Sigma) \quad \textit{subject to} \quad g_i(x) \leq 0 \tag{2.11}$$

The first term, $c^2(x)$, penalizes deviation of the nominal solution, $f(x,0)$, from the target, $\bar{f}$, while the second term, $s(x,\Sigma)$, penalizes the sensitivity of the design with respect to δ. The first term is non-negative and is zero when $f(x,0) = \bar{f}$. The second term is also non-negative, because Σ is positive semi-definite. We therefore see that the objective, $c^2(x) + s(x,\Sigma)$, is lower-bounded by zero. Since the cost function is the sum of two terms, there will typically be a trade-off between finding a design that minimizes the squared error of the nominal design and one that reduces the sensitivity.

2.2 Uncertainty Models

At the beginning of this section we introduced $f(x)$ and $f(x,\delta)$ as the nominal and uncertain objective functions, respectively. We will now show that given an objective function $f(x)$, there are several ways to incorporate uncertainty while maintaining the rational polynomial structure. Two common uncertainty models for which this holds true are additive and multiplicative models. Design variables subject to additive uncertainty can be written as

$$\tilde{x}_i = x_i + \delta_i$$

where $\tilde{x}_i$ is an uncertain variable and δ_i is the associated uncertainty with x_i. For multiplicative uncertainty the model is

$$\tilde{x}_i = (1+\delta_i)x_i$$

Any polynomial function can be expressed as a sum of monomials, written as

$$h(x) = \sum_{t=1}^{T} \beta_t \left(\prod_{i=1}^{n} x_i^{\alpha_{t,i}} \right) \tag{2.12}$$

where T is the number of monomials, β_t is the coefficient for the t^{th} monomial, $\alpha_{t,i}$ is the power of the x_i in the t^{th} monomial, and n is the dimension of x. Since $h(x)$ is a polynomial, the matrix α contains elements from the set of non-negative integers.

Let us assume that the first n_a variables have additive uncertainty, the next n_m variables have multiplicative uncertainty, and the remaining $(n - n_a - n_m)$ variables have no uncertainty. Substituting the additive and multiplicative models given above into the general polynomial $h(x)$, we find that

$$h(x,\delta) = \sum_{t=1}^{T} \beta_t \left(\prod_{i=1}^{n_a} (x_i + \delta_i)^{\alpha_{t,i}} \prod_{i=n_a+1}^{n_a+n_m} (1+\delta_i)^{\alpha_{t,i}} x_i^{\alpha_{t,i}} \prod_{i=n_a+n_m+1}^{n} x_i^{\alpha_{t,i}} \right)$$

Clearly, $h(x,\delta)$ is polynomial in x and δ, and therefore we see that by using an additive and/or multiplicative uncertainty model does not change the polynomial structure of a function.

In addition to uncertainty in design variables, it is common for it to appear in constants. For example, density, Young's modulus, gravity, and aerodynamic coefficients, to name a few, all have associated uncertainties. Most uncertain constants are expressed as

$$\tilde{\beta}_t = \beta_t + \epsilon_t$$

where $\tilde{\beta}_t$ is the actual value, β_t is the nominal value, and ϵ_t represents the error. Referring to our general polynomial $h(x)$ from equation (2.12), let us assume the first T_u coefficients are uncertain. We can therefore write

$$\begin{aligned} h(x,\delta,\epsilon) &= \sum_{t=1}^{T} \beta_t \left(\prod_{i=1}^{n_a} (x_i + \delta_i)^{\alpha_{t,i}} \prod_{i=n_a+1}^{n_a+n_m} (1+\delta_i)^{\alpha_{t,i}} x_i^{\alpha_{t,i}} \prod_{i=n_a+n_m+1}^{n} x_i^{\alpha_{t,i}} \right) \\ &+ \sum_{t=1}^{T_u} \epsilon_t \left(\prod_{i=1}^{n_a} (x_i + \delta_i)^{\alpha_{t,i}} \prod_{i=n_a+1}^{n_a+n_m} (1+\delta_i)^{\alpha_{t,i}} x_i^{\alpha_{t,i}} \prod_{i=n_a+n_m+1}^{n} x_i^{\alpha_{t,i}} \right) \end{aligned}$$

The expression for $h(x,\delta,\epsilon)$ is polynomial in x, δ, and ϵ. We can define δ_h to be

$$\delta_h = \begin{bmatrix} \delta & \epsilon \end{bmatrix}^T$$

and the dimension of δ_h is $(n_a + n_m + T_u)$. In this section we showed that given a polynomial, it is trivial to add uncertainty to the design variables and constants while maintaining the polynomial structure. This allows us to consider problems of the form posed in (2.1), and then later add uncertainty to the objective and maintain the rational polynomial structure presented in equation (2.2).

Chapter 3

Optimization Algorithm

In this section we present the details of an algorithm designed to minimize rational polynomial functions subject to polynomial inequality constraints. A mathematical formulation for this class of problems is

$$\min_x \frac{N(x)}{D(x)} \quad \textit{subject to} \quad g_i(x) \leq 0 \tag{3.1}$$

where $x \in \mathbb{R}^n$, and $N(x)$, $D(x)$, and $g_i(x)$ are multi-variable polynomials. The underlying principle for the algorithm is that the problem posed in (3.1) is trivial to solve when x is a scalar. We will show that the scalar problem can be computed efficiently and exactly by exploiting the polynomial structure of the objective function and constraints. In addition, we will explain how the algorithm uses the scalar problem to do global line searches in $\mathbb{R}^n$, and how the search directions are chosen.

A brief outline of the algorithm follows:

1. The algorithm is initialized at a starting point x_0.
2. A search direction, v_k, is chosen.
3. A single variable line search is done along $x(t) = x_{k-1} + tv_k$

$$\min_t \frac{N(t)}{D(t)} \;\; s.t. \;\; g_i(t) \leq 0$$

4. When a feasible minimum, t^*, is found, $x_k = x_{k-1} + t^* v_k$, otherwise $x_k = x_{k-1}$.
5. Steps 2-4 are repeated until a stopping criteria is met. The subscript k represents the iteration, hence at the k^{th} iteration x_k represents the current minimum and v_k was the search direction.

3.1 Choosing a Starting Point

The first step is to choose a starting point, x_0, to initialize the search. The algorithm was designed to start from a feasible point. We consider finding a feasible point, given a set of constraints, to be a separate problem, and thus have not addressed it. Once a starting point has been chosen, the algorithm enters the main loop where it repeatedly chooses a direction and does a line search.

3.2 Selecting a Search Direction

Each time the algorithm does a line search, it must first chose a direction to search in. This choice is by far the biggest influence on how well the algorithm performs. Ideally the direction chosen would pass through the constrained global minima and the algorithm would terminate after only a single line search. This however is wishful thinking and hence an interesting problem arises, "how to best chose a direction?" In this section we will describe our approach for choosing a direction, which combines several strategies.

3.2.1 Coordinate-wise Strategy

The first strategy is to choose a direction aligned with a single design variable. This is the simplest strategy, and is accomplished by setting v to the zero vector and then randomly choosing a single entry of v to be one. The result of this is a search along a single design variable. We will refer to this strategy as the coordinate-wise search.

Searching coordinate-wise has several benefits. The most obvious benefit is that searching along a single design variable allows that variable to be optimized, ceteris paribus. This can be very effective on variables that are involved in a few constraints or less. When this is the case, these variables can usually be optimized independently without violating a constraint, whereas random or gradient based searches will likely violate a constraint when other variables are heavily constrained. A second benefit of the coordinate-wise search is that it is unaffected by poor scaling among the design variables. The last main benefit is that the transformation $x = x_{k-1} + tv_k$ is faster to compute and the resulting line-search is generally easier to solve.

3.2.2 Random Strategy

The second strategy is to chose a purely random direction. This is done by picking n random numbers that are uniformly distributed on the interval $[-1, 1]$, and scaling

them according to the relative size of the design variables. We will refer to this as the random search.

It is common for design variables to have different magnitudes and it is necessary to account for this scaling. Simply changing units can affect the scaling of a problem. Poorly scaled optimization problems are usually very difficult to solve [7]. One method for dealing with poorly scaled problems is to normalize the variables and recast the problem into an equivalent problem where all of the variables are in the unit cube. We have chosen to account for the relative scaling of design variables through a diagonal matrix, Ψ. The diagonal entries of Ψ represent the relative scale between design variables. Currently Ψ needs to be set prior to running the optimization, otherwise the scaling of the variables is assumed equal, $\Psi = I$. The search direction, v, for the random strategy can then be written as $v = \Psi w$, where w is a uniformly distributed random vector in $\mathbb{R}^n$ on the interval $[-1, 1]$.

One of the biggest pitfalls of the random search is that it does very poorly on problems with small feasible regions and numerous constraints. As the number of constraints increases, the probability that the random strategy chooses a feasible search direction typically decreases.

3.2.3 Constrained Random Strategy

The third strategy is a variation of the random search because it attempts to choose a direction that does not violate active linear constraints. At first, this strategy might not seem beneficial for problems with only nonlinear constraints; however, since variable bounds of the form $L \leq x \leq U$ are linear constraints, most problems realize some benefit. Since we are searching along an affine function, specifically $x(t) = x_{k-1} + tv_k$, we can choose a random direction in the hyperplane tangent to the active linear constraints. We will refer to this strategy as the constrained random search.

The implementation requires some basic linear algebra with which we assume the reader is familiar. First, the set active linear constraints, Γ, is defined to be

$$\Gamma := \{\alpha_j : g_i(x_{k-1}) = \alpha_j^T x_{k-1} + c_k,\ -\epsilon \leq g_i(x_{k-1}) \leq 0\} \tag{3.2}$$

where $\alpha_j \in \mathbb{R}^n$ are the coefficients for the j^{th} active linear constraint, c_k is a constant, and ϵ is the tolerance used to determine if a constraint is active. Assuming there are l active linear constraints, the next step is to stack the coefficients, α_k, in a matrix H as

$$H := [\alpha_1\ \ \alpha_2\ \ \ldots\ \ \alpha_l]^T \tag{3.3}$$

To choose a direction tangent to the active linear constraints, v needs to be chosen such that it lies in the null-space of H, $\mathcal{N}(H)$, [8]. Such a choice implies that $Hv = 0$, and hence the line search will not violate any active linear constraints. In order to

maintain the scaling discussed above, v is chosen as described for the random search strategy and then projected onto the range of the $\mathcal{N}(H)$. The $\mathcal{N}(H)$ is found by doing a singular value decomposition of H. The final expression used to compute v can be written as

$$v = VV^T \Psi w \tag{3.4}$$

where V is a basis for the $\mathcal{N}(H)$ and $w \in R^n$ is a uniformly distributed random vector in the unit cube. When there are no active linear constraints, equation (3.4) reduces to $v = \Psi w$. When the $\mathcal{N}(H)$ is the empty set, this implies that a tangent hyperplane to the active linear constraints does not exist. In this case, the algorithm defaults to using the random search strategy.

3.2.4 Steepest-descent Strategy

The final strategy is a steepest-decent method. Since the gradient of the objective function represents the direction of steepest ascent, searching along the negative of this vector yields the direction of steepest-decent. The expression for v is then given by

$$v_k = \nabla \frac{N(x)}{D(x)}\bigg|_{x=x_0}$$

We have included this method because it is usually efficient at finding local minima. More information on this technique can be found in [9].

3.2.5 Search Strategies Integration

Finally, we will explain how the strategies are integrated into a single method to choose a direction. Each time the algorithm needs to generate a new direction, one of the strategies is selected randomly with a specified probability. Table (3.1) lists the probabilities associated with each strategy. These probabilities were chosen because they performed best on a set of test problems.

strategy	probability
coordinate-wise	0.2
random	0.3
constrained random	0.3
steepest-decent	0.2

Table 3.1: Probabilities with which a search strategy is chosen.

Once a direction, v, has been chosen, the objective function and constraints are transformed from multi-variable polynomials to single variable polynomials using

$$x(t) = x_{k-1} + tv \tag{3.5}$$

where x_{k-1} was the current minimum at the last iteration. The single variable optimization problem is

$$\min_t \frac{N(t)}{D(t)} \quad s.t. \quad g_i(t) \leq 0 \tag{3.6}$$

We will refer to this problem as a line search in $\mathbb{R}^n$. This rational polynomial minimization in one variable, subject to polynomial constraints is easy to solve in closed form.

3.3 Exact Line Search Minimization

The solution to the problem posed in equation (3.6) is presented here. We begin by defining the set of feasible points, Ω_i, for each constraint, $g_i(t)$, as

$$\Omega_i := \{t : \ g_i(t) \leq 0\} \tag{3.7}$$

Computing Ω_i is done by finding the real roots of $g_i(t)$, and then checking the intervals between the roots to identify where $g_i(t)$ is negative.

Assuming that the roots of $g_i(t)$ can be computed numerically, the real roots, $\{\lambda_{r1}, \lambda_{r2}, \ldots, \lambda_{rp}\}$, satisfy $g_i(\lambda_{rk}) = 0$ for $k = 1, ..., p$. Once the real roots are identified, the set Ω_i can be found by simply checking the sign of $g_i(t)$ on either side of λ_{rk} for $k = 1, ..., p$. Double roots are easily handled because the algorithm computes the $sgn(g_i(t))$ on both sides of λ_{rk}. If no real roots exist, then the function $g_i(t)$ is either strictly positive, in which case Ω_i is the empty set, or strictly negative which implies $\Omega_i \in \mathbb{R}$. In this case, evaluating $g_i(0)$ is sufficient to determine the sign of $g_i(t)$ for all t. Storing Ω_i is can be done by maintaining a list of numbered pairs that represent the intervals of feasible regions.

Once all of the feasible sets have been found, the next step is to find their intersection. We will define the intersection to be

$$\Omega := \Omega_1 \cap \Omega_2 \cap ... \cap \Omega_p \tag{3.8}$$

Ω is the set of feasible points for which all the constraints are satisfied. If Ω is the empty set, then the search has failed because no where along the line $x = x_{k-1} + tv$ can all the constraints be satisfied. Otherwise, the constrained minimum, t^*, must satisfy $t^* \in \Omega$.

Assuming the objective function is continuous on the set Ω, it can be differentiated to find local minima and maxima. If the objective function is not continuous on Ω,

this implies that there exists a $\hat{t} \in \Omega$ such that $D(\hat{t}) = 0$. Clearly this implies that the minimum is $-\infty$ and hence the optimization problem was poorly posed. Setting the first derivative of $\frac{N(t)}{D(t)}$ to zero yields

$$N'(t)D(t) - D'(t)N(t) = 0 \tag{3.9}$$

The real roots of (3.9) correspond to the points at which local minima and maxima of $\frac{N(t)}{D(t)}$ are achieved. The real roots that are not in the set Ω are discarded because they are infeasible. Finding the constrained minimum requires evaluating $\frac{N(t)}{D(t)}$ at the feasible real roots of (3.9) and the endpoints that define the set Ω.

If the line search is successful, then the current optimal design vector, x_k, is calculated

$$x_k = x_{k-1} + t^* v_k$$

Otherwise, when the line search fails the current minimum persists for the next iteration, $x_k = x_{k-1}$. This is the end of a single iteration and the algorithm continues until a stopping criteria is met.

3.4 Stopping Criteria

We have put little focus into the stopping criteria of the algorithm and thus the three conditions we implemented are trivial. The first is to limit the maximum number of iterations. This is common among optimization packages. The second condition compares the value of the objective function against a user supplied minimum. When a minimum is known, or when a lower-bound can be computed, this can be supplied to the algorithm as a stopping criteria.

The last condition is a maximum number of iterations that the algorithm should attempt without improving the cost. This can actually be viewed, and implemented, in two different ways. One is to keep track and limit the total number of unsuccessful searches. A second method would be to limit the number of sequential searches that fail.

3.5 Algorithm Summary

To summarize, we were able to exploit the polynomial structure of the objective function and constraints which allowed us to find the feasible regions and minimize the objective function easily for the scalar problem. For problems in $\mathbb{R}^n$, we used an affine transformation to do global line searches. The heart of the algorithm lies in the choosing the search direction, which is done each iteration. Because of the randomness involved in choosing search directions, the algorithm is not deterministic.

Chapter 4

Optimization Examples

We will present several example problems in this section to illustrate our rational polynomial algorithm presented in chapter 3. These are pure optimization examples in constrained global minimization and do not directly relate to the robust design problem posed in chapter 2.

4.1 Inventory Control Problem

The first example is an inventory control problem taken from [10]. The problem has three non-negative design variables and a single constraint. We will define the design vector to be $x = [x_1 \;\; x_2 \;\; x_3]^T$.

Objective Function

$$\min_x \;\; f(x) = \frac{5x_1^2x_2x_3 + 50,000x_2x_3 + 20x_1x_2^2x_3 + 72,000x_1x_3 + 10x_1x_2x_3^2 + 144,000x_1x_2}{x_1x_2x_3}$$

Variable Bounds

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \end{aligned}$$

Constraints

$$g(x) = 4x_2x_3 + 32x_1x_3 + 120x_1x_2 - x_1x_2x_3 \leq 0$$

For this problem, it is rather easy to choose a feasible starting point. Setting $x_0 = [160 \; 160 \; 160]^T$ will satisfy the constraint and variable bounds. Choosing the scaling, Ψ, is less intuitive. With no a priori knowledge of the design variables, we

assumed a uniform scaling by letting $\Psi = I_3$. The maximum number of iterations was set to 500.

The solution found was

$$\begin{aligned} x^* &= \begin{bmatrix} 108.73 & 85.15 & 204.29 \end{bmatrix}^T \\ f(x^*) &= 6299.8 \\ g(x^*) &= 0 \end{aligned}$$

This result closely agrees with the published result given in [11]. From the value of $g(x^*)$ it is clear that the constraint is active at the solution. In fact, the constraint was active during 97% of the optimization. The solution was found after 297 iterations which took approximately 1 second on a P4 1.8Ghz Linux workstation. Figure (4.1) contains two plots that illustrate results from this example.

The plot on the left demonstrates two characteristics of our algorithm. First, the value of the objective function will always monotonically decrease. Second, decreasing the cost function is typically trivial initially, but becomes increasingly difficult. Of the 297 iterations required, there were 31 successful line searches where the algorithm succeed in finding a better cost. Sixteen of the successful searches occurred in the first 50 iterations of the optimization.

The plot on the right in figure (4.1) illustrates how the design variables changed throughout the optimization. After the first 50 iterations, the constraint was active and the cost decreased by roughly 1/2%. The small change in the value of the objective function after the initial iterations, and from inspection of the problem, one can conclude that the minimum should occur along the constraint. Because the constraint is nonlinear, the algorithm is not able to search along the constraint. This is one main limitation of the algorithm.

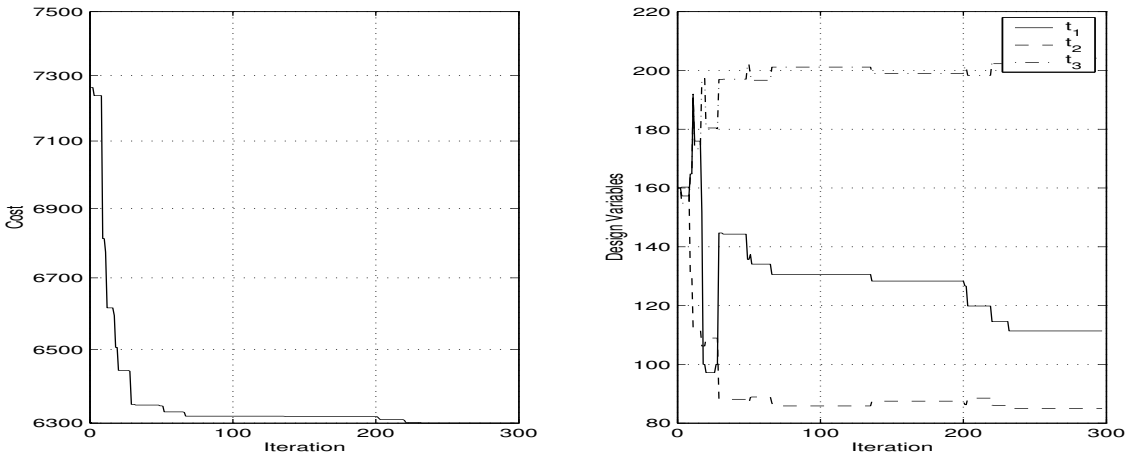

Figure 4.1: Inventory Control Minimization Results.

4.2 Alkylation Process Design

The next optimization problem was taken from "Handbook of Test Problems in Local and Global Optimization." This example is a chemical engineering design problem of an alkylation unit. The objective is to improve the octane number of some olefin feed by reacting it with isobutane in the presence of acid [12].

Objective Function

$$\min_x \quad c_1x_1 + c_2x_1x_6 + c_3x_3 + c_4x_2 + c_5 - c_6x_3x_5 \tag{4.1}$$

Variable Bounds

$$\begin{aligned}
1500 &\leq x_1 \leq 2000 \\
1 &\leq x_2 \leq 120 \\
3000 &\leq x_3 \leq 3500 \\
85 &\leq x_4 \leq 93 \\
90 &\leq x_5 \leq 95 \\
3 &\leq x_6 \leq 12 \\
145 &\leq x_7 \leq 162
\end{aligned}$$

Constraints

$$\begin{aligned}
c_7x_6^2 + c_8x_1^{-1} - c_9x_6 &\leq 1 \\
c_{10}x_1x_3^{-1} + c_{11}x_1x_3^{-1}x_6 - c_{12}x_1x_3^{-1}x_6^2 &\leq 1 \\
c_{13}x_6^2 + c_{14}x_5 - c_{15}x_4 - c_{16}x_6 &\leq 1 \\
c_{17}x_5^{-1} + c_{18}x_5^{-1}x_6 + c_{19}x_4x_5^{-1} - c_{20}x_5^{-1}x_6^2 &\leq 1 \\
c_{21}x_7 + c_{22}x_2x_3^{-1}x_4^{-1} - c_{23}x_2x_3^{-1} &\leq 1 \\
c_{24}x_7^{-1} + c_{25}x_2x_3^{-1}x_7^{-1} - c_{26}x_2x_3^{-1}x_4^{-1}x_7^{-1} &\leq 1 \\
c_{27}x_5^{-1} + c_{28}x_5^{-1}x_7 &\leq 1 \\
c_{29}x_5 + c_{30}x_7 &\leq 1 \\
c_{31}x_3 - c_{32}x_1 &\leq 1 \\
c_{33}x_1x_3^{-1} + c_{34}x_3^{-1} &\leq 1 \\
c_{35}x_2x_3^{-1}x_4^{-1} - c_{36}x_2x_3^{-1} &\leq 1 \\
c_{37}x_4 + c_{38}x_2^{-1}x_3x_4 &\leq 1 \\
c_{39}x_1x_6 + c_{40}x_1 - c_{41}x_3 &\leq 1 \\
c_{42}x_1^{-1}x_3 + c_{43}x_1^{-1} - c_{44}x_6 &\leq 1
\end{aligned}$$

The parameters $c_i,\ i = 1, \ldots, 44$ are listed in appendix A. This problem has seven design variables, each with upper and lower bounds. In addition, there are two linear and 12 non-linear constraints. The constraints are not written explicitly in the form $g_i(x) \leq 0$, but through algebraic manipulations they can easily be converted.

The variable bounds also need to be transformed into the form $g_i(x) \leq 0$. Each upper and lower bound becomes a single linear constraint. This effectively adds 14 linear constraints, which brings the total number of constraints to 28. The objective function is a rational polynomial with $D(x) = 1$.

Choosing a feasible starting point for this problem is nontrivial. We used a simple approach to generate a set of 100 unique points. This was done by choosing random vectors within the variable bounds, and then checking to see if the constraints were satisfied. This approach took an average of 20,000 guesses to generate a single feasible starting point. This is an indication that the problem has a small feasible design space.

The next step was to set the scaling. From looking at the variable bounds, it is clear that there are several different length scales. With no a priori knowledge of the design variables, other than the bounds, the most obvious scaling appeared to be

$$\Psi_{ij} = \left\{ \begin{array}{ccc} U_i - L_i & , & i = j \\ 0 & , & otherwise \end{array} \right\}$$

where U_i and L_i are the upper and lower bound for x_i, respectively. This scaling is similar to normalizing the variables. This approach expects that the change in a variable, relative to the other variables, will be proportional to the size of it's feasible region. While this certainly does not have to be true, it is usually a good approximation. The last step was to set the maximum number of iterations. We ran the optimization twice, on a set of 100 unique, feasible starting points. The first time the maximum number of iterations was set to 1200, and the second time to 15,000.

For the 100 initial conditions the best solution found by our algorithm was

$$\begin{array}{rcl} x^* & = & \left[\begin{array}{ccccccc} 1697.95 & 53.73 & 3031.12 & 90.12 & 95 & 10.48 & 153.54 \end{array} \right]^T \\ f(x^*) & = & 1226.76 \end{array}$$

This solution almost exactly matches the result published in [12]. Computationally, 1200 iterations took approximately 17 seconds on a P4 1.8Ghz Linux workstation, while 15,000 iterations took just over 200 seconds. This is a difficult optimization problem and therefore our algorithm found a number of local minima.

$\frac{f-f_p}{f_p}$	1200 iter	15000 iter
0.01	9%	60%
0.02	18%	73%
0.05	36%	82%
0.10	57%	92%

Table 4.1: Solutions within a given tolerance of the published solution, f_p.

Table (4.1) is a comparison of success rates between the 1200 and the 15,000 iteration cases. Success was defined as a solution being within one, two, five or ten percent of the published solution, f_p. Clearly, allowing the algorithm to iterate longer improved the success rates. From the table we see that after 15,000 iterations 60% of our solutions were within 1% of the published result.

Similar to most non-convex optimization packages, when our algorithm is started from different initial conditions, it is likely to end up at different solutions. In addition, our algorithm is not deterministic because of the randomness used in choosing a search direction. This results in different solutions being found from the same initial condition, whereas deterministic algorithms will always follow the same path to a solution.

Chapter 5

Robust Design Examples

5.1 Two Degree of Freedom MEMS Resonator

The first example we will consider is a simple resonator with two degrees of freedom. Our objective is to robustly design the resonant frequency of this structure. This is not a difficult problem to solve in closed form, but is a good example that we will use to illustrate our robust design technique presented in chapter 2. Additional motivation for studying this problem is that with only two design variables it is easy to visualize.

The two degree of freedom (DOF) resonator, illustrated in figure (5.1), consists of a proof mass attached by two parallel beams to a fixed anchor. To simplify the problem we assume that the proof mass has a fixed area A, and that the thickness of the entire device is t. The two design variables are the beam width, h, and length, L, as shown in figure (5.1). We will define the design vector to be $x = [h\ L]^T$.

The natural frequency, w_n, of the simple resonator can be approximated as $\sqrt{k/M}$, where k is the combined stiffness of the two parallel beams and M is the mass of the proof mass. The expressions for k and M are

$$\begin{aligned} k &= \frac{2Eth^3}{L^3} \\ M &= \rho A t \end{aligned}$$

where E is Elastic Modulus and ρ is the density of the material used to fabricate the resonator. We will assume additive uncertainty in the design variables, and therefore we define an uncertain design vector to be

$$\tilde{x} = x + \delta \tag{5.1}$$

where $\delta = [\delta_h\ \ \delta_L]$. Since our algorithm is designed for rational polynomial functions, we choose the uncertain objective function, $f(x, \delta)$, to be w_n^2. The expression for w_n^2,

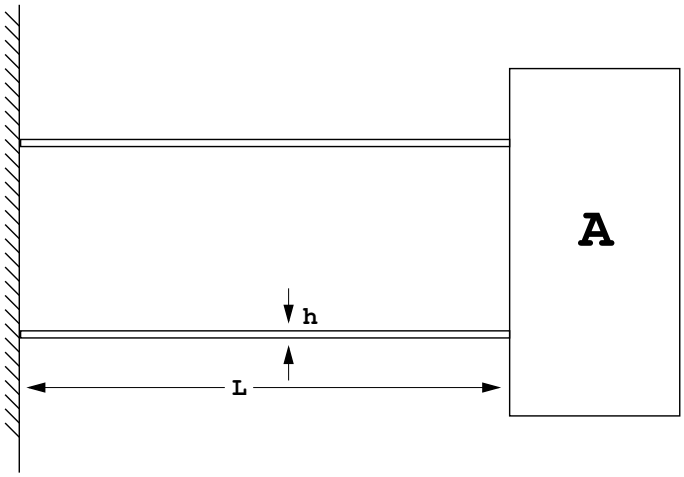

Figure 5.1: Diagram of Two DOF Resonator.

in terms of x and δ, is given by

$$f(x,\delta) = w_n^2 = \left(\frac{2E}{\rho A}\right)\left(\frac{(h+\delta_h)^3}{(L+\delta_L)^3}\right) \tag{5.2}$$

We will assume that the design is subject to the following constraints

$$\begin{array}{c} h_{\min} \leq h \leq h_{\max} \\ 10h \leq L \leq L_{\max} \end{array} \tag{5.3}$$

These constraints have been imposed for illustrative purposes, however they could be the result of MEMS design rules, die size requirements, fabrication limitations or a number of other design factors.

Now that we have a rational polynomial objective function and a set of constraints, we can pose the robust design problem that we would like to solve. This is essentially a restatement of equation (2.9)

$$\min_x \left(\left(\frac{f(x,0) - \bar{f}}{\bar{f}} \right)^2 + \frac{1}{\bar{f}^2} \left(\nabla_2 f(x,0) \Sigma \nabla_2^T f(x,0) \right) \right) \quad s.t. \quad g_i(x) \leq 0 \tag{5.4}$$

where $\bar{f}$ is the w^2_{design}, Σ is the uncertainty covariance matrix, and $g_i(x)$ are the constraints. The constraints given in (5.3) can be written in a vector $g(x)$ as

$$g(x) = \begin{bmatrix} h_{\min} - h \\ h - h_{\max} \\ 10h - L \\ L - L_{\max} \end{bmatrix} \tag{5.5}$$

For simplicity we will assume that only the beam width, h, is uncertain. The covariance matrix Σ is therefore

$$\Sigma = \begin{bmatrix} \sigma_h^2 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.6}$$

where σ_h is the standard deviation of the uncertainty in h. The gradient of $f(x, \delta)$ with respect to δ is given by

$$\nabla_2 f(x, 0) = \begin{bmatrix} \frac{6Eh^2}{\rho AL^3} & -\frac{6Eh^3}{\rho AL^4} \end{bmatrix} \tag{5.7}$$

For this example we choose a target resonant frequency of 20kHz, approximately 125,000 rad/s. The relevant design parameters are listed in table (5.1). For notational convenience, we once again refer to the two terms that make up the cost function in equation (5.4) as

$$\begin{aligned} c^2(x) &= \left(\frac{f(x,0)-\bar{f}}{\bar{f}}\right)^2 \\ s(x, \Sigma) &= \frac{1}{\bar{f}^2}\left(\nabla f(x)\Sigma\nabla^T f(x)\right) \end{aligned}$$

The term $c^2(x)$ is a measure of the deviation of the nominal design, $f(x, 0)$, from the target, $\bar{f}$, while $s(x, \Sigma)$ is a measure of the sensitivity at the nominal design. Minimizing $c^2(x)$ is equivalent to reducing the spread between the nominal and target frequency. By choosing the design variables such that $f(x, 0) = \bar{f}$, the $c^2(x)$ cost can be eliminated.

Design Parameter - Description	Value
A - area of the proof mass	$10,000\,\mu m^2$
t - thickness of proof mass and beams	$2.0\,\mu m$
ρ - density of silicon ($2330\ kg/m^3$)	$2.3 \times 10^{-12}\,gm/\mu m^3$
E - Elastic Modulus of silicon	$1.6 \times 10^8\,gm/\mu m s^2$
$h_{\min}$ - minimum dimension	$2.0\,\mu m$
$h_{\max}$ - maximum beam width	$20.0\,\mu m$
$L_{\max}$ - maximum beam length	$500\,\mu m$
σ_h - standard deviation of the uncertainty in h	$0.2\,\mu m$

Table 5.1: Design parameters for 2 DOF resonator.

We will begin the analysis of this problem by looking at the feasible design space. Figure (5.2) illustrates the four linear constraints given in equation (5.5). The feasible region is inside the quadrilateral and is clearly convex. The dashed line in figure (5.2) represents the set of designs that have a resonant frequency of 20 kHz. Mathematically, we can say that it is the set $\{x \mid f(x, 0) = \bar{f}\}$, and therefore the associated

$c^2(x)$ cost term is zero along this line. All designs above and to the left of the dashed line have resonant frequencies less than 20 kHz while those that lie to the right and below have higher resonant frequencies.

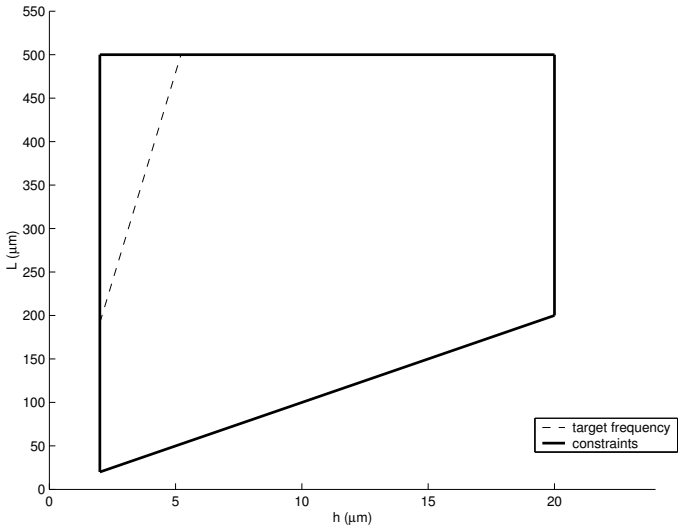

Figure 5.2: Feasible design space for 2 DOF resonator and target frequency.

Since the $c^2(x)$ term is invariant along the dashed line in figure (5.2), we can look at just the $s(x, \Sigma)$ term to find the best design for this set. Figure (5.3) is a plot of $s(x, \Sigma)$ as a function of h along the line $c^2(x) = 0$. Clearly the sensitivity of the nominal solution decreases as h increases. Intuitively this makes sense because increasing the nominal dimension will make h less sensitive to uncertainty. The best design along the line $c^2(x) = 0$ is $x = [5.3\mu m \;\; 500\mu m]^T$, which is along the constraint $L \leq L_{\max}$. This design however is not optimal with respect to the robust design problem posed in (5.4).

Figure (5.4) is a 3-D plot of $\log_{10}(c^2(x) + s(x, \Sigma))$ as a function of the design variables. The valley seen in the 3-D plot roughly corresponds to the dashed line in figure (5.2). This valley is a result of the $c^2(x)$ term being nearly zero where $f(x, 0)$ is close to $\bar{f}$. By writing out $s(x, \Sigma)$

$$s(x, \Sigma) = \left(\frac{6E\delta_h}{\rho A \bar{f}}\right)^2 \left(\frac{h^4}{L^6}\right)$$

it is obvious that $s(x, \Sigma)$ increases with increasing h or decreasing L. As a result of this, the valley occurs slightly to the left and above the dashed line in figure (5.2). In addition, the cost associated with the $s(x, \Sigma)$ term dominates when h is large and L is small, which can be seen in figure (5.4).

Visually, from looking at figure (5.4), it is easy to see that a unique minimizer exists for this constrained optimization problem. The minimum is located along the

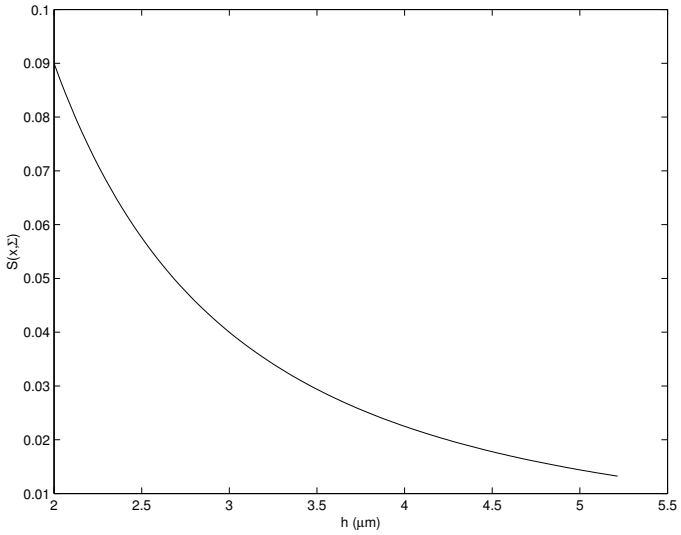

Figure 5.3: Sensitivity cost term, $s(x, \Sigma)$, on the invariant set $c^2(x) = 0$ for the 2 DOF resonator.

$L \leq L_{\max}$ constraint at $x = [5.20\mu m \;\; 500\mu m]^T$. The associated cost is 1.316×10^{-2} and the resonant frequency is $19.9kHz$. The reason the resonant frequency of our optimal design is 1/2% below the target is because of the cost trade-off between the $c^2(x)$ and $s(x, \Sigma)$. The minimum $s(x, \Sigma)$ cost found in figure (5.3) is 1.323×10^{-2}, which is larger than the combined cost, $c^2(x) + s(x, \Sigma)$, of our optimal solution.

Figure (5.5) illustrates how our optimization algorithm converged on the solution from six different initial conditions. The starting points for each path are label on the plot with a unique number that corresponds to the legend. Some of the solutions were found after only a few iterations while others chatter down the valley taking a couple hundred iterations. In figure (5.5) all of the paths converge to the correct global solution indicated on the plot. The average number of iterations to find a solution, based on 2000 runs from different feasible random initial conditions, was approximately 100.

In this section we illustrated the robust design of a two DOF resonator. We considered only uncertainty in the beam width and as expected, increasing h reduced the sensitivity cost term $s(x, \Sigma)$. The result was a solution where L was driven to it's maximum value and h was selected such that the nominal resonant frequency was close to the target.

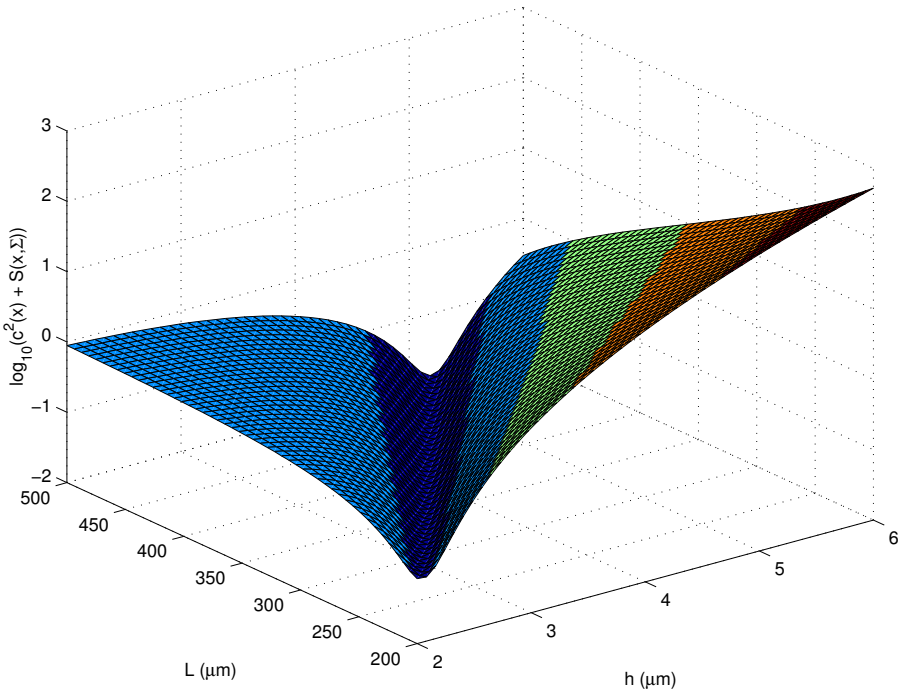

Figure 5.4: 3D plot of total cost as a function of design variables.

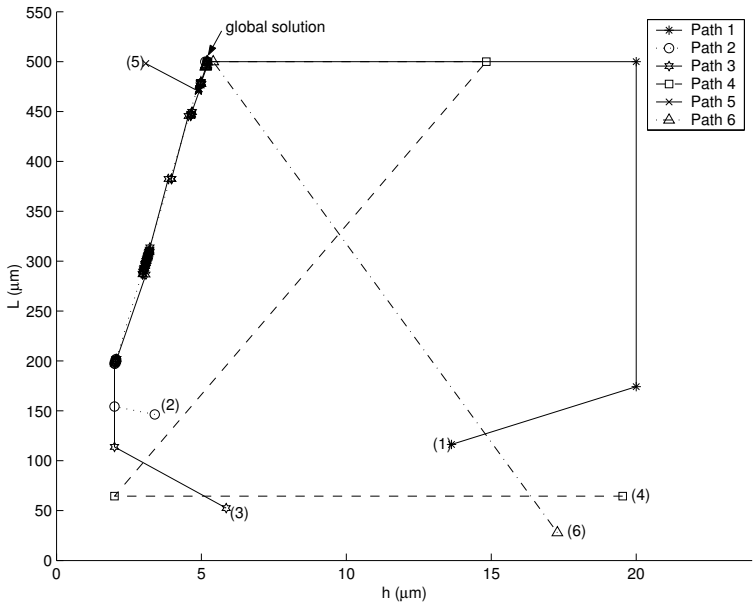

Figure 5.5: Several different paths taken to find the solution from different initial conditions. The starting points are label with numbers on the plot.

5.2 Crab-Leg Resonator

The second example that we will discuss is a six variable crab-leg resonator. This problem was originally taken from [13] and has been modified and extended. The goal is identical to that for the problem presented in section 5.1. We would like to robustly design the resonant frequency in the presence of geometric process variations. The design for the two DOF resonator was rather intuitive, however this problem is sufficiently complex that numerical optimization is necessary to find a robust design. There is significant freedom in how the design variables can be chosen and we will show that the optimized solution is superior to the average design.

The crab-leg resonator is shown in figure (5.6). The structure exhibits four-fold symmetry and thus there are six design variables of interest. The design variables h_m and b_m are the height and width of the proof mass, respectively. The legs that support the proof mass are anchored to the substrate. The design variables for the four legs are h_1, h_2, L_1 and L_2. We are assuming that this resonator will be fabricated using a surface micro-machining technology and therefore the entire structure will have a fixed thickness t. The variables X and Y represent the maximum dimensions of the overall structure. Finally, we will define k_{x} and k_{y} to be the stiffness of the structure in the x and y-direction, respectively.

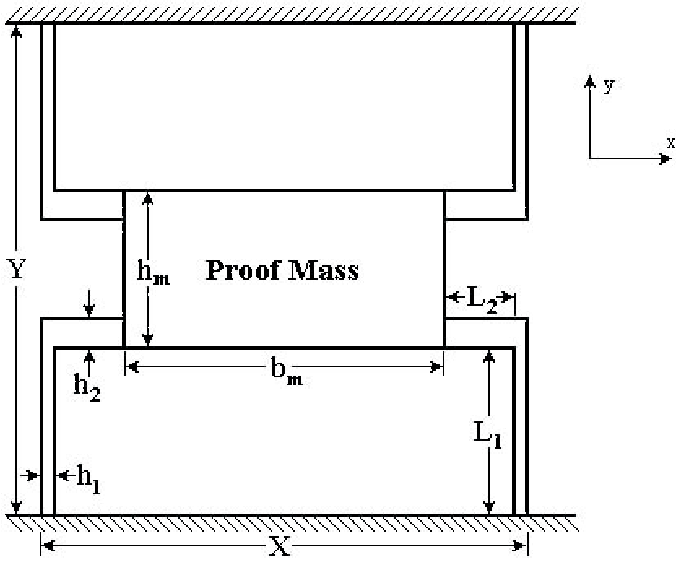

Figure 5.6: Diagram of Six Variable Crab-leg Resonator.

We will use the same approximation for natural frequency as in the preceding section, assuming that $w_n = \sqrt{k_{\mathrm{x}}/M}$. Our goal is to design the resonant frequency in the x-direction, and thus k_{x} is the stiffness of interest. We will also neglect the mass

of the four legs and therefore the expression for M is

$$M = \rho h_m b_m t$$

where ρ is the density. The expressions for k_{x} and k_{y} are slightly more complex but are given by

$$\begin{aligned} k_{\mathrm{x}} &= 16Et\left(\frac{h_1}{L_1}\right)^3\left(\frac{h_1^3L_2+h_2^3L_1}{4h_1^3L_2+h_2^3L_1}\right) \\ k_{\mathrm{y}} &= 16Et\left(\frac{h_2}{L_2}\right)^3\left(\frac{h_1^3L_2+h_2^3L_1}{h_1^3L_2+4h_2^3L_1}\right) \end{aligned}$$

where E is the Young's Modulus of the material. The expressions for k_{x} and k_{y} were derived based on the several assumptions. The derivations can be found in [13]. It follows that the expression for w_n^2 is given by

$$w_n^2 = \frac{16Eh_1^3(h_1^3L_2 + h_2^3L_1)}{\rho h_m b_m L_1^3(4h_1^3L_2 + h_2^3L_1)} \tag{5.8}$$

Similar to the two DOF example, the expression for w_n^2 is a rational polynomial in the design variables, however here the numerator is seventh order and the denominator is ninth order. We will define the design vector for this problem to be $x = [L_1 \;\; L_2 \;\; h_1 \;\; h_2 \;\; h_m \;\; b_m]^T$.

Now that we have defined the variables and expressed natural frequency of the crab-leg as a rational polynomial, we will explain the 10 constraints imposed on the design. These constraints are given in table (5.2).

#	Constraint	#	Constraint
1.	$h_1 \geq h_{min}$	6.	$L_2 \geq 10h_2$
2.	$h_2 \geq h_{min}$	7.	$X \geq 2L_2 + 2h_1 + b_m$
3.	$b_m \geq h_{min}$	8.	$Y \geq 2L_1 + h_m$
4.	$h_m \geq 2h_2 + h_{min}$	9.	$k_{\mathrm{y}} \geq \alpha k_{\mathrm{x}}$
5.	$L_1 \geq 10h_1$	10.	$\beta_{max} \geq \beta$

Table 5.2: Design constraints for crab-leg resonator.

The first four constraints were imposed because MEMS processes, such as MUMPs®, typically have design rules that specify minimum line and space requirements [14]. The fifth and sixth constraints ensure that the legs can be modelled as thin-beams. The next two are requirements imposed by the designer on the overall size of the structure. The ninth constraint is in place to separate the two resonant frequencies. We would like the parasital y-direction resonant frequency to be much greater than that in the x-direction to reduce motion in the y-direction. The last constraint is to

enforce that the stress, β, at the joint (where the two beams join to form a leg) does not exceed a maximum. The nonlinear expression for β is given by

$$\beta = \frac{12Eh_1^3h_2^3L_1D}{h_2^2L_1^2(4h_1^3L_2 + h_2^3L_1)}$$

where D is the maximum allowable deflection of the structure in the x-direction. Once again, through simple algebraic manipulation the constraints can be written as polynomials in the form $g_i(x) \leq 0$. Table (5.3) contains the values of constants used for this example. All of the design parameters were chosen to be realistic.

Design Parameter - Description	Value
t - thickness of proof mass and beams	$2.0\,\mu m$
ρ - density of silicon (2330 kg/m^3)	$2.3 \times 10^{-12}\,gm/\mu m^3$
E - Elastic Modulus of silicon (160 GPa)	$1.6 \times 10^{8}\,gm/\mu ms^2$
h_{min} - minimum dimension	$2.0\,\mu m$
D - maximum deflection in the x-direction	$2.0\,\mu m$
β_{max} - maximum allowable stress (1.6 GPa)	$1.6 \times 10^{6}\,gm/\mu ms^2$
X - maximum size of structure in the x-direction	$600\mu m$
Y - maximum size of structure in the y-direction	$600\mu m$
α - minimum stiffness ratio $k_{\mathrm{y}}/k_{\mathrm{x}}$	16

Table 5.3: Design parameters for crab-leg resonator.

5.2.1 Determining the set of feasible resonant frequencies

Currently we have a rational polynomial that describes the resonant frequency of the crab-leg and a set of constraints. There is no notion of uncertainty yet, but the problem is well-defined. Before a designer tackles the robust problem, a good first question to address is "what range of performance can be achieved subject to the constraints?" For this example, we would like to find lower and upper bounds on the achievable resonant frequencies.

Finding the lower-bound requires minimizing w_n^2 subject to the constraints. Conversely, the upper-bound can be found by maximizing w_n^2. Maximizing the resonant frequency however is equivalent to minimizing $\frac{1}{w_n^2}$, or $-w_n^2$. Solving for the two bounds requires minimizing a rational polynomial subject to polynomial constraints; which is precisely the type of problem we designed our optimization algorithm for.

The lower, $w_{\min}$, and upper, $w_{\max}$, bound were found to be 13 kHz and 11.8 MHz, respectively. Computationally, the calculations typically took less than 1500 iterations, or 12 seconds, to compute on a P4 1.8Ghz Linux workstation. If the

target resonant frequency is outside of the feasible range, then either the design or constraints need to be modified.

5.2.2 Choosing a target frequency

With an understanding of the system's achievable performance, given the design and constraints, we chose a target resonant frequency, w_{target}, of 200 kHz. Before tackling the robust problem, we would like to find the set of designs that have a resonant frequency of w_{target} when there is no uncertainty. There are two important reasons which motivate us to explore this set. First, in studying this set we hope to demonstrate that there is considerable freedom in how the design variables are chosen. Secondly, by finding a set of designs that achieve the desired performance under nominal conditions ($\delta = 0$), we will have a set for which we can compare the robust design. We will refer to this set as Φ

$$\Phi = \{x : w_n^2 = w_{\text{target}}^2,\ g_i(x) \leq 0 \ for\ i = 1, \dots, m\} \tag{5.9}$$

Taking w_{target} to be 200 kHz, the set Φ contains an infinite number of designs. While equation (5.9) is easily written, finding the set Φ is an impractical task. Instead, we chose to find a subset, which we will refer to as Φ_{1000}, that contains 1000 unique designs.

We were able to find Φ_{1000} by minimizing the squared error between w_n^2 and w_{target}^2 from different initial conditions. This can be written as the following optimization

$$\min_x \left(w_n^2 - w_{\text{target}}^2\right)^2 \ \ s.t.\ \ g_i(x) \leq 0 \tag{5.10}$$

This problem is equivalent to minimizing the $c^2(x)$ term in the robust design problem of equation (2.9). The optimization problem posed in (5.10) is rational polynomial with polynomial constraints, and thus we were able to use our optimization algorithm to find unique designs. Figure (5.7) illustrates four of the more extreme designs from Φ_{1000}. Each of these structures has a resonant frequency of 200 kHz and satisfies the constraints. They demonstrate that there is significant freedom in how the design variables are chosen. The question we will now ask is "which design is the most robust to geometric uncertainties, and how can we find that design?"

5.2.3 Robust Design

We will use the method developed in chapter 2 to find a robust design, and answer the question posed in the preceding section. All that is necessary is to assume a model for the uncertainty and write an expression for squared natural frequency as a function

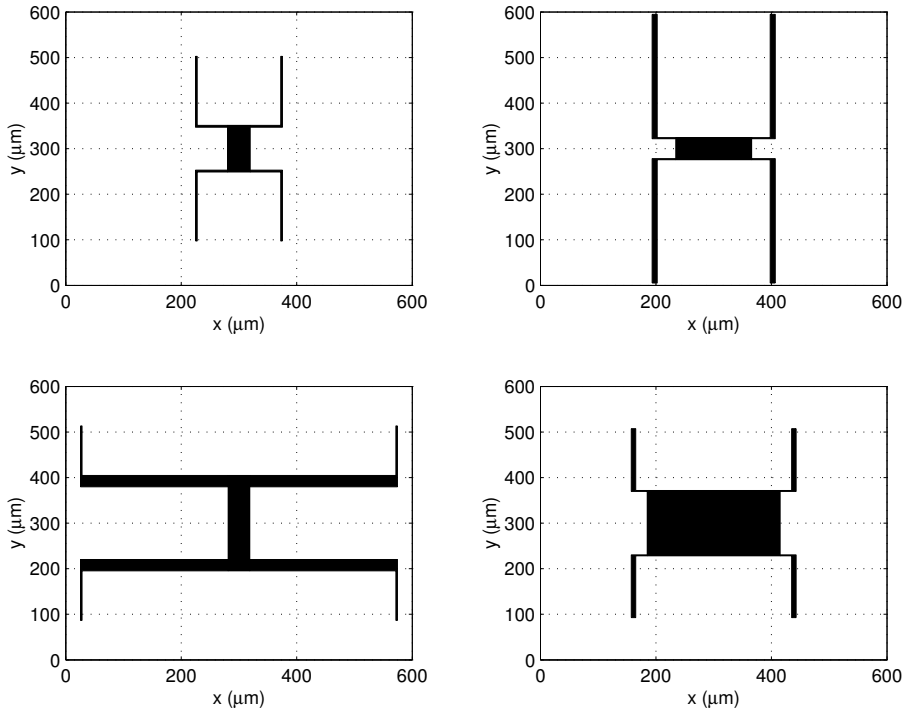

Figure 5.7: Four unique crab-leg resonator designs from Φ_{1000}.

of the design variables and uncertainty. This is easily done by modelling the geometric process variations with the following additive uncertainty model

$$\tilde{x} = x + \delta$$

where $\tilde{x}$ is the uncertain (actual) design vector and $\delta = [\delta_{L_1} \;\; \delta_{L_2} \;\; \delta_{h_1} \;\; \delta_{h_2} \;\; \delta_{h_m} \;\; \delta_{b_m}]^T$. Substituting the uncertain design variables into the expression for w_n^2, equation (5.8), allows us to write

$$f(x,\delta) = \tilde{w}_n^2 = \frac{16E\tilde{h}_1^3\left(\tilde{h}_1^3\tilde{L}_2 + \tilde{h}_2^3\tilde{L}_1\right)}{\rho\tilde{h}_m\tilde{b}_m\tilde{L}_1^3\left(4\tilde{h}_1^3\tilde{L}_2 + \tilde{h}_2^3\tilde{L}_1\right)} \tag{5.11}$$

where $f(x,\delta)$ is a rational polynomial in x and δ. We will restate the robust optimization problem that was formulated in chapter 2 for simplicity

$$\min_x \left(\left(\frac{f(x,0) - \bar{f}}{\bar{f}} \right)^2 + \frac{1}{\bar{f}^2} \left(\nabla_2 f(x,0) \Sigma \nabla_2^T f(x,0) \right) \right) \quad s.t. \quad g_i(x) \leq 0 \tag{5.12}$$

The target performance, $\bar{f}$, is simply w_{target}^2. We will label the cost function in (5.12) as $F(x,\Sigma)$. We remind the reader that the covariance matrix, Σ, represents the

correlation between the uncertainty in the design variables. We will study the robust design problem using two different models for Σ.

The first model assumes that there is no correlation. In this case, Σ is diagonal with the entries representing the variance of each term in δ. Since we are considering geometric process variations, we will assume that the standard deviation of each term in δ is equal. We will define the uncorrelated covariance matrix, Σ_U, as

$$\Sigma_U = \sigma^2 I_6 \tag{5.13}$$

where σ is the standard deviation of the geometric dimensional uncertainty. The second model we will consider is the case when the structure is etched uniformly. Figure (5.8) illustrates the two uniform etch scenarios on a beam with an end mass. The vector $\xi = [-1\ -1\ \ 1\ \ 1\ \ 1\ \ 1]^T$ represents a uniform etch for the crab-leg structure. If we take λ to be a gaussian random variable, with standard deviation σ, then we can write the correlated uncertainty vector as $\delta = \lambda\xi$. When λ is positive, this corresponds to the design, $x + \lambda\xi$, having been underetched, and conversely, negative λ indicates the structure was overetched. We can therefore write the correlated covariance matrix, Σ_C, as

$$\Sigma_C = \sigma^2 \xi\xi^T \tag{5.14}$$

We will proceed by looking at the results for these two uncertainty models.

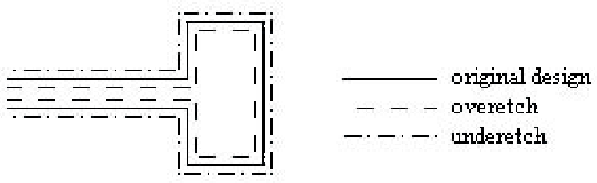

Figure 5.8: Illustration depicting under and overetch of a beam with an end-mass.

Uncorrelated Results

Using our rational polynomial optimization algorithm to solve the problem posed in (5.12), we found the following robust design for the uncorrelated uncertainty case

$$\begin{aligned}
x_U^* &= \left[\ 241.68 \quad 20.56 \quad 14.49 \quad 2.03 \quad 87.84 \quad 442.71\ \right]^T \\
w_n &= 199.960\ kHz \\
F(x_U^*, \Sigma_U) &= 4.2362 \times 10^{-4} \\
c^2(x_U^*) &= 1.6 \times 10^{-7} \\
s(x_U^*, \Sigma_U) &= 4.2346 \times 10^{-4} \\
\nabla_x F(x_U^*, \Sigma_U) &= -\left[\ .057 \quad .025 \quad 4.780 \quad .408 \quad .056 \quad .011\ \right] \times 10^{-5}
\end{aligned}$$

The robust design, x_U^*, for the uncorrelated uncertainty case is shown in figure (5.9). The functions $c^2(x)$ and $s(x, \Sigma)$ were defined in equation (2.10). Clearly, the nominal ($\delta = 0$) resonant frequency, w_n, of this design almost nails the target. This is also evident by noting that the $c^2(x)$ term is small. From looking at the breakdown of the cost function $F(x_U^*, \Sigma_U)$, we see that the sensitivity, $s(x_U^*, \Sigma_U)$, is the dominant term.

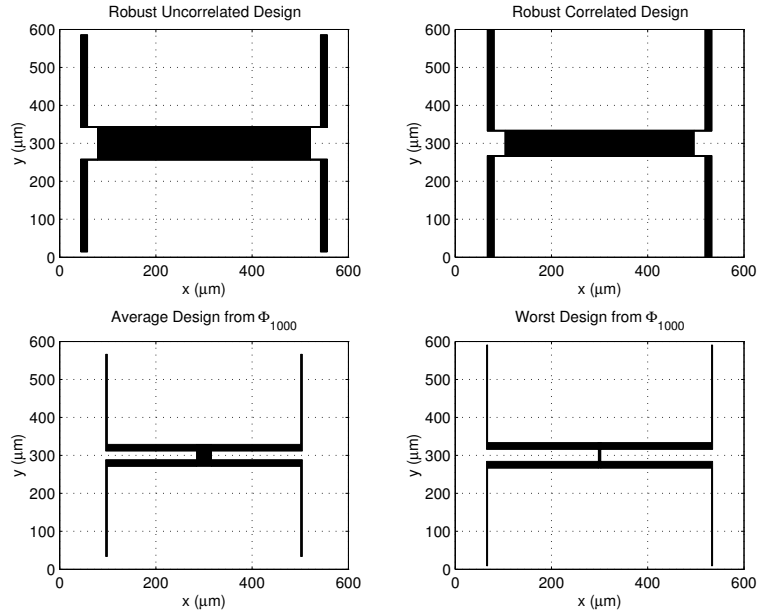

Figure 5.9: Illustration of the robust designs and two designs from Φ_{1000}.

None of the constraints are active at the robust design x_U^*. This would imply that the solution is a local minima, and hence the gradient should be zero. Referring to the solution we see that the gradient, $\nabla_x F(x_U^*, \Sigma_U)$, is essentially zero. The reason the gradient is not identically zero is because of the numerical precision with which the minimum can be calculated.

The most obvious approach to evaluate the robust design is to compare it against designs that nominally meet the target frequency. To do this, let us evaluate the cost function, $F(x, \Sigma)$, for the set Φ_{1000}. Since each design in the set Φ_{1000} has a nominal resonant frequency equal to the target frequency, the term $c^2(x)$ is zero and thus the cost function reduces to $s(x, \Sigma)$.

Figure (5.10) is a histogram of the cost $s(x, \Sigma)$ for all of the designs in Φ_{1000}. The minimum cost is 6.28×10^{-4}, which is approximately 50% larger than the robust design. The average cost is 8.7×10^{-3} and the worst-case cost from the set Φ_{1000} was 2.27×10^{-2}. Both the average and worst designs from Φ_{1000} are illustrated in figure (5.9). Clearly the average and worst designs differ significantly from the robust solutions. Comparing the costs associated with the designs in Φ_{1000} against our robust

solution is useful in verifying that the optimization was successful, but less helpful in understanding how the robustness improved.

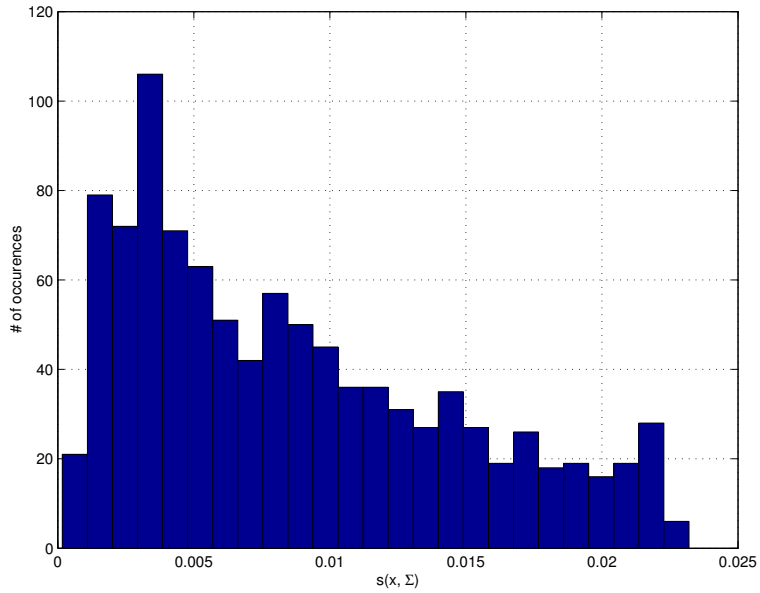

Figure 5.10: Histogram of the sensitivity cost term, $s(x, \Sigma_U)$, for the set Φ_{1000}.

To demonstrate the robustness of our solution we ran a monte carlo simulation to model an uncertain MEMS fabrication process. To represent the variation in the process we generated 400,000 gaussian random vectors with a standard deviation, σ, of $0.1\,\mu m$. We then calculated the natural frequencies of the robust design and the average and worst design from Φ_{1000} subject to these variations. The results are displayed in figure (5.11). The distribution for the robust design has a much tighter spread than both the average and worst-case designs from Φ_{1000}. This is an excellent indication that the optimized design is much less insensitive to geometric uncertainties in the design variables.

Correlated Uncertainty

The robust design for the correlated uncertainty model, equation (5.14), is given here

$$\begin{aligned}
x_C^* &= \begin{bmatrix} 265.64 & 22.84 & 14.09 & 2.00 & 68.71 & 392.60 \end{bmatrix}^T \\
w_n &= 199.950\ kHz \\
F(x_C^*, \Sigma_C) &= 5.700 \times 10^{-4} \\
c^2(x_C^*) &= 3.0 \times 10^{-7} \\
s(x_C^*, \Sigma_C) &= 5.697 \times 10^{-4} \\
\nabla_x F(x_C^*, \Sigma_C) &= \begin{bmatrix} -.003 & -.066 & -.902 & 1.34 & .002 & -.001 \end{bmatrix} \times 10^{-4}
\end{aligned}$$

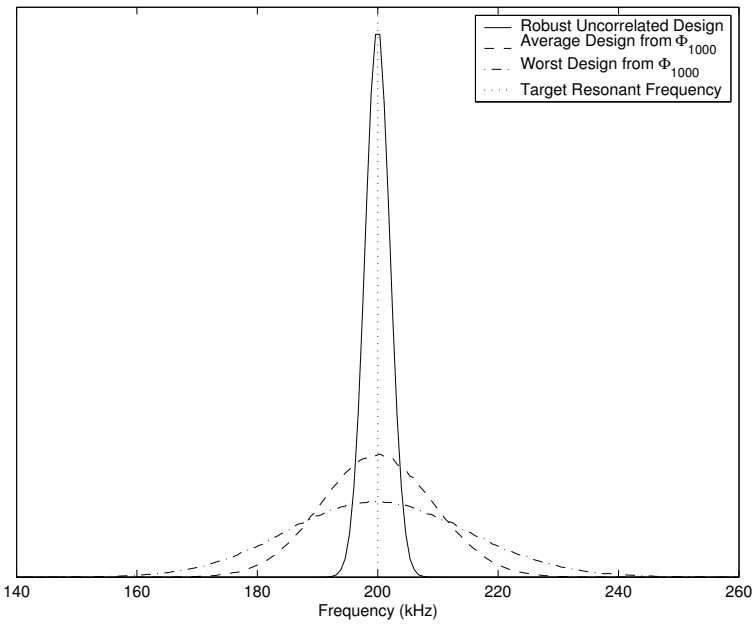

Figure 5.11: Distributions of crab-leg resonant frequencies subject to uncorrelated uncertainty.

Constraints 2, 8, and 9 from table (5.2) are active at the solution. Aside from the fact that there are active constraints, the main difference between the uncorrelated and correlated designs are in the variables L_1, h_m, and b_m. The size of the proof mass for the correlated design is about 70% of the size for the uncorrelated design and the legs are slightly longer. Figure (5.9) illustrates the robust design x_C^*, and figure (5.12) shows a distribution of resonant frequencies when this design is subjected to correlated uncertainty. Although the correlated and uncorrelated designs are physically similar and have similar distributions, they are each optimal for the uncertainty model for which they were designed.

In this chapter we presented the robust design of a six variable crab-leg resonator. We showed how it was possible to determine lower and upper bounds on the resonant frequency given the design and constraints. We then chose a target frequency of 200 kHz, and used our optimization to find a set of 1000 designs that at nominal have a resonant frequency equal to the target. Two uncertainty models, an uncorrelated and correlated, were developed and used in solving the robust problem. We then compared the cost, $F(x, \Sigma)$, between our robust designs and the designs in Φ_{1000}. Finally, we illustrated frequency distributions, generated from a monte carlo simulation of process variation, which showed that our robust designs were less sensitive to uncertainty.

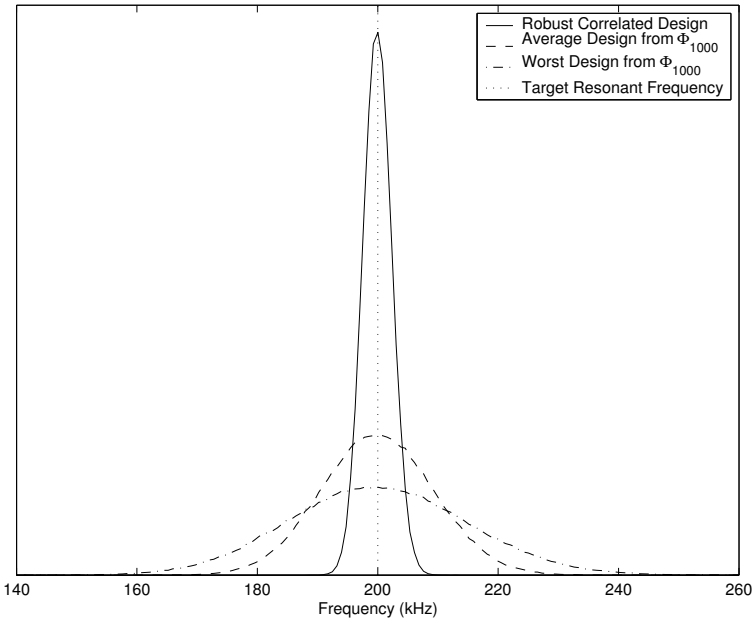

Figure 5.12: Distributions of crab-leg resonant frequencies subject to correlated uncertainty.

Chapter 6

Conclusion

In this report we presented a robust design technique for MEMS systems. The problem posed was to minimize the expected variance between actual and target performance in the presence of uncertainty. The two key assumptions were: (1) the mean and covariance of the uncertainty was known, and (2) the use of a first-order Taylor series expansion to approximate the actual performance. With these assumptions, the robust design problem was reduced to a constrained minimization.

We focused on solving a subset of constrained minimization problems, and gave a detailed outline of an algorithm to minimize rational polynomial functions subject to polynomial inequality constraints. The algorithm uses line searches by choosing a direction and then through a linear transformation converts the optimization into a scalar problem during each iteration. It was shown that the scalar problem was trivial to solve by exploiting the polynomial structure of the objective function and constraints. The algorithm was demonstrated on two optimization problems and was shown to be successful.

Finally, we applied the robust design technique and used the optimization to design two MEMS resonant structures. The first example was a simple two-beam resonator that was easy to visualize. The second example was a more realistic design problem of a crab-leg resonator with six design variables and 10 constraints. We designed this structure to have a resonant frequency of 200 kHz and considered two models for uncertainty, correlated and uncorrelated. Both designs were optimal for their respective uncertainty models and were shown to be far more robust to uncertainty when compared with 1000 random designs which met the constraints and nailed the target frequency.

Bibliography

[1] *What is MEMS Technology?* MEMSnet. November 27, 2002 <http://www.memsnet.org/mems/what-is.html>.

[2] Petersen, Kurt E. "Silicon as a Mechanical Material." *Proceedings of the IEEE* vol. 70 no. 5 (May 1982): pp. 420-457.

[3] Tang, W.C. PhD Thesis, University of California, Berkeley, CA (1990).

[4] Maluf, Nadim. *An Introduction to Microelectromechanical Systems Engineering.* Boston: Artech House, 2000.

[5] Pister, Kris. *Introduction to MEMS Design and Fabrication.* EECS 245 Reader University of California, Berkeley, CA (Fall 2001). pp. 9-45.

[6] Sharpe, W. N., K.T. Turner, and R.L. Edwards. "Tensile Testing of Polysilicon." *Experimental Mechanics* vol. 39 no. 3 (September 1999): pp. 162-170.

[7] Nocedal, Jorge, and Stephen J. Wright. *Numerical Optimization.* New York: Springer-Verlag, 1999.

[8] Strang, Gilbert. *Linear Algebra and its Applications.* Orlando: Harcourt Brace Jovanovich College Publishers, 1988.

[9] Rao, Singiresu S. *Engineering Optimization.* New York: John Wiley & Sons, Inc., 1996.

[10] Kochenberger, G.A., R.E.D. Woolsey, and B.A. McCarl. "On the Solution of Geometric Programs via Separable Programming." *Operation Research Quarterly* vol. 24 no. 2 (June 1973): pp. 285-294.

[11] Rijckaert, M.J. and X.M Martens. "Comparision of Generalized Geometric Programming Algorithms." *Jornal of Optimization Theory and Applications* vol. 26 no. 2 (October 1978), pp. 205-242.

[12] Floudas, Christodoulos A., et al. *Handbook of Test Problems in Local and Global Optimization* Boston: Kluwer Academic Publishers, 1999.

[13] Pisano, Albert P. ME219 Lecture Notes, University of California, Berkeley, CA (Spring 2001).

[14] Koester, David, et al. *PolyMUMPs Design Handbook.* revision 8 2002. Research Triangle Park, NC: Cronos Integrated Microsystems.

Appendix A

Alkylation Constants

i	c_i	i	c_i	i	c_i	i	c_i
1	1.715	12	.0066033	23	.25125634e2	34	.819672
2	0.035	13	.66173269e-3	24	.16118996e3	35	24500
3	4.0565	14	.17239878e-1	25	5000	36	250
4	10	15	.56595559e-2	26	.48951e6	37	.10204082e-1
5	3000	16	.19120592e-1	27	.44333333e2	38	.12244898e-4
6	0.063	17	.5685075e2	28	.33	39	.0000625
7	.59553571e-2	18	1.08702	29	.022556	40	.0000625
8	.88392857	19	.32175	30	.007595	41	.00007625
9	.1175625	20	.03762	31	.00061	42	1.22
10	1.1088	21	.006198	32	.0005	43	1
11	.1303533	22	.24623121e4	33	.819672	44	1

Table A.1: Constants c_i for $i = 1 \ldots 44$ for the Alkylation Process Design