An Integrated Simulation-based Generative Design Method for Microelectromechanical Systems

A Dissertation submitted to The University of Cambridge Engineering Department
for the Degree of Doctor of Philosophy

Francesca Bolognini

Jesus College

October 2008

Declaration

Except where otherwise stated and for commonly understood and accepted ideas, this dissertation is the result of my own research and does not include the outcome of work done in collaboration. Chapter Five reports part of the work done for my Master of Philosophy dissertation submitted to this University in October 2004, and that I have considered fundamental to understand further developments of this research. This thesis has not been submitted in whole for any other degree or qualification at this University or any other Institute of Learning. This work contains less than 45,000 words and 73 figures.

Francesca Bolognini,
Jesus College
Cambridge
October 2008

Abstract

This work investigates the development of an automated design method aimed at facilitating synthesis through the generation of a range of feasible and optimally directed design alternatives. The method is aimed at assisting designers in the exploration of performance limits and trade-offs for synthesis tasks as well as reducing design time.

Due to their complexity and their multidomain nature, microelectromechanical systems (MEMS) are the design domain of application chosen for this exploration. MEMS design is still performed by hand, despite a growing interest in facilitating their design and shortening their time to market.

The novel method combines a multicriteria generate-and-test search algorithm, called Burst, with a Connected Node System (CNS) design representation. The method provides automatic links to multiphysics simulation for quantitative evaluation of designs performance, as well as the possibility for designers to choose preferred solutions in archives of Pareto optimal designs.

Different case studies are used in order to test the potential of the method, and extend its capabilities to solve a variety of design tasks. Characteristics of flexibility and scalability of the method are validated through complex MEMS design tasks. The appraisal is provided by practical applications such as microresonators developed for the mobile and satellite industry. The tasks examined range from size optimisation of longitudinal free-free-beam resonators to more complex topology optimisation of sandwich resonators. The design objectives examined are bulk resonant frequencies, motional resistance and quality factor.

The results not only successfully proved the effectiveness, flexibility and extendibility of the novel method, but also offered a better understanding of potential and limits of synthesis methods, as well as the basis for their future improvement.

Contents

List of Figures

List of Tables

GLOSSARY - Explanation of Terms and Acronyms

AC Alternating Current

ANSYS An engineering simulation software

API Application Programming Interface

BL Burst Length

CAD Computer Aided Design (commonly used to refer specifically to computer-aided sketching tools)

CAE Computer Aided Engineering

CAM Computer Aided Manufacturing

CC Clamped-Clamped

CMOS Complementary Metal–Oxide–Semiconductor is a technology is used in microprocessors

CNS Connected Node System

COMSOL COMSOL Multiphysics (formerly FEMLAB) is a finite element analysis and solver software package for various physics and engineering applications, especially coupled phenomena, or multiphysics.

DC Direct Current

E Young's Modulus

EMO Evolutionary Multi-Objective

FDM Finite Difference Method

FEA Finite Element Analysis

FEM Finite Element Method

FF Free-Free

GA Genetic Algorithm

MATLAB A numerical computing environment and programming language.

MEMS Microelectromechanical Systems

ML Machine Learning

MO Multi-Objective

MOGA Multi-Objective Genetic Algorithm

MOSA Multi-Objective Simulated Annealing

MUMPS Multi-User MEMS Process is a MEMS foundry process

ODE Ordinary Differential Equations

PDE Partial Differential Equations

Q Quality Factor

RF Radio Frequency

R_m Motional Resistance

RTB Return to Base

SOGA Single Objective Genetic Algorithm

SUGAR Analysis software for MEMS

TED Thermo-Elastic Damping

V Voltage

V_{DC} DC Voltage

VLSI Very Large Scale integration is the process of creating integrated circuit by combining thousands of transistor-based circuits into a single chip.

CHAPTER 1: INTRODUCTION

1.1 Computational Synthesis in Engineering Design

Through invention of technical artefacts, namely Engineering Design, humans create what does not exist in nature. Engineering design can be subject to errors and failures, and after thousands of years designers are still trying to reach that perfection and maximum efficiency that characterises many natural processes. How can we improve design? And how can we design more effectively and efficiently? More or less satisfactory responses to these questions have been formulated, but the most significant answer was given when computers took a central role in design activities.

Engineering design is a complex process consisting of several stages and in the past many have tried to model it in a systematic way [Pahl, 1996]. The approaches taken to describe it divide it in steps that go from the initial clarification of the design task to the final solution (Figure 1.1). Since the second half of last century CAD/CAM tools have seen a rapid diffusion, and computers have increasingly become partners in the design process, helping automate many of the stages once carried out exclusively by hand. Today computers help designers in any task involving analysis and calculations. Computational tools are also used for modelling, planning, manufacturing and managing knowledge. However, a closer look at the design process shows that there is one phase where the computer still has a peripheral role. This phase is the creative or conceptual one, also known as 'synthesis'. Synthesis can be defined as the act of combining or putting together or 'synthesising' existing components or building blocks to produce a unified system that exhibits at least the required behaviour [Shea, 1997]. Synthesis is also a broader notion, one that encompasses the creation of new things in general, and not only by means of combination

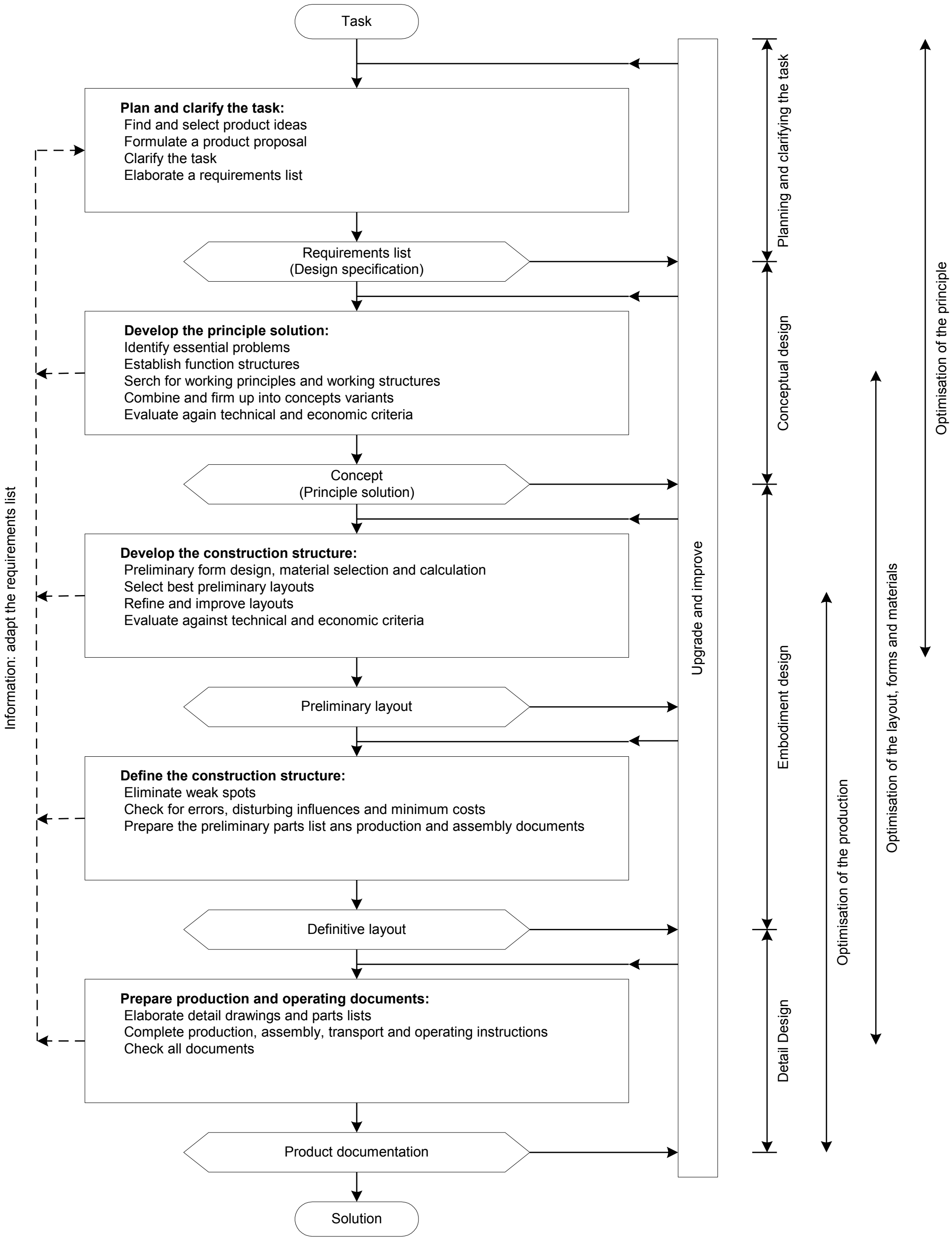


Figure 1.1. Stages of the engineering design process following the systemic approach of Pahl and Beitz [Pahl, 1996].

of existing elements. Antonsson and Cagan define synthesis as the conception and postulation of possibly new solutions to solve a problem [Antonsson, 2001]. This step of the design process is, in most engineering design tasks, performed by creative human minds.

The effort of the scientific community to facilitate the creative stage of the design process has been broad and a variety of methods flourished in the last few decades under the name of computational synthesis or 'formal synthesis'[1]. By formal synthesis we mean the algorithmic creation of designs; the organised, methodological modelling, implementation and execution of design creation on a computer [Cagan, 2005]. The aim is to leverage computational speed and depth of calculation to reduce the tedium of human designers and augment the process of searching the space of alternatives for preferred solutions [Cagan, 2005]. It is the goal of research into formal engineering design synthesis not only to develop procedures that can synthesise novel designs, but also to understand the fundamental principles by which these processess work, so that this understanding can be applied to new systems and design environments [Antonsson, 2001].

The advantages that computational synthesis can bring to designers are several:

- Better search of the design space in an automated way, in order to consider as many good solutions as possible
- Promotion of lateral thinking to aid innovation
- Insight into poorly understood tasks (due to interdependent parameters and constraints, coupled multidisciplinary performances, or emergent behaviours)
- Generation of multicriteria design archives for investigating complex performance trade-offs
- Reduction of design time and cost
- Improved computational support for conceptual design and embodiment
- Possibility for designers to not get 'trapped' into design solutions based only on previous experience and intuition, that are not necessarily beneficial
- Better direction of the overall creative effort.

From this simple list, it is clear why total automated synthesis has been regarded as the most sought after breakthrough in engineering design, promising to bring significant advances to both the design process and resulting products. Unfortunately, as things stand now, the only possibility for designers seems to be working in partnership with computers, as complete automation of synthesis tasks is still far from being commonplace in industry. In most engineering fields, the use of generative design tools is at early stages and

---

[1] In this work the definitions 'computational synthesis methods', 'automated synthesis methods', 'formal synthesis methods' and 'generative design methods' will be considered as synonymous.

designers cannot rely on commercial packages to use in everyday practice. In the development of synthesis methods, the involvement of creativity and other competencies that are unique to human minds, have been an obstacle. This and other implementation difficulties behind the scarce use of computational synthesis have been the motive of this research.

1.2 Research Overview

In Spring 2003 the American Association for Artificial Intelligence organised the first symposium dedicated to computational synthesis issues, titled ‘Computational Synthesis: from Basic Building Blocks to High Level Functionality’. No prior meeting had been dedicated to fundamental questions in automatic synthesis. Although several conferences on design automation existed for many years, they had the tendency to be domain specific and did not systematically address universal issues. The symposium was conceived in the light of a simple consideration: synthesis methods, if compared to other computational design tools, had a peripheral role in the design process. The symposium had the purpose of bringing together experts in the field to discuss major flaws in the development and use of computational synthesis methods, possible solutions and future objectives in the research area.

The main justification for the limited use of computational synthesis, given in the introductory notes of the symposium, was that generative design methods are often restricted in their scope due to implementation issues. The symposium set the objective of identifying fundamental or ‘high level’ properties of design synthesis methods that could allow a larger spread of their use and employment in tasks of ever growing complexity until the point that, ultimately, they could replace human invention.

The first of these properties to be considered was the domain-independent nature of synthesis methods. Many current design automation paradigms focus on specific domains, making use of elaborate domain knowledge and domain-specific algorithms, and are typically limited to a single level optimisation. Multidomain methods are instead universal methods that can address open-ended conceptual synthesis in a largely unconstrained design space.

Scalability is the property that allows expanding and upgrading of synthesis methods in order to handle increasingly difficult tasks. The main challenge is scaling synthesis algorithms so that they can achieve complex functionalities. The investigation of scalability issues starts from some characteristics that are recognised to have an effect on scalability. These characteristics are:

- Modularity: as modules are intended as components of a synthesis method, modularity is the characteristic that allows the modification and expansion of methods through straightforward introduction of new features and functionalities.
- Regularity and hierarchy: another key ingredient is the efficient composition of design building blocks into large complexes. Regular structures, hierarchical structures, symmetry, duplication, self-similarity and minimisation of information contents are important aspects of making complex designs manageable and the synthesis process scalable.
- Abstraction and encapsulation of functionalities: for the synthesis process to scale, it is necessary to be able to abstract and encapsulate lower level design building components, so that the overall number of parameters remain constant as higher levels of complexity are reached.

The ideal synthesis tool, still to be invented, should present universal characteristics such as a multidomain nature, an ability to scale-up to difficult tasks without sacrificing accuracy, flexibility and the possibility to be integrated with other design tools.

The present research stems from the necessity to explore limits and potentials of synthesis methods, in order to improve their level of functionality and use. For this purpose, the properties of synthesis methods listed above represent the starting point of this work. The necessary steps to pursue the investigation are:

- The development of a synthesis method to be implemented according to the envisioned characteristics aforementioned
- Testing of the method in a design field that will allow the exploration of multiple design domains and present sufficient complexity to investigate scalability issues
- Confirmation of the efficacy of the method, when compared with existing synthesis methods: this can be done trough the use of benchmark design problems found in literature

- Application the method to case studies of interest to explore characteristics, potentials and limitations.

Each of these points will be individually tackled in the following chapters.

1.3 Research Method

The implementation of a new computational synthesis method has been considered in this investigation as a fundamental means to explore development issues, limits and potentials of generative tools.

The aim of computational synthesis methods is not just to find the best possible solution to a design problem: their emphasis is rather on creating design solutions that are, possibly, new design alternatives. Synthesis as a method contrasts with traditional optimisation in that the goal of synthesis is more broadly to capture, emulate and/or utilise design decisions made by human designers during the creative process. Ideally, computational design synthesis is evoked in situations in which the human designers are often at a loss of what avenues to pursue, or the best method of achieving a solution requires the generation and evaluation of countless alternatives [Cagan, 2005]. Computational synthesis methods are complex products that do not just automate the optimisation of solutions. Their scope goes much beyond that, extending the concept of search of optimal designs with methods for creating solutions to propose to the optimisation. Also, some intelligence can be integrated in the way synthesis techniques offer solutions. Ultimately, the aim is to get machines to exhibit behaviour, which if done by humans, would be assumed to involve the use of intelligence [Samuel, 1969].

There is no exact formulation to implement synthesis methods, but there is agreement in the computational synthesis community that they can be considered as a set of distinct activities [Cagan, 2005]. Cagan divides synthesis methods in a set of four activities, identifying them as representation, generation, evaluation and guidance [Cagan, 2005]. In order to understand each of these steps, it is useful to think of them as similar to some important activities that humans follow in their design process: creation of a mental model of the object (representation), creation of the parts and the whole (generation), analysis of how well it meets the design goals and constraints (evaluation), and feedback on the improvements to the design for the next iteration (guidance). In this work, the synthesis method implemented was thought as composed by the following activities:

1) Representation: this is the activity of a synthesis method where the form of design solutions is captured. Representation may be implemented very differently, for example using building elements that are more or less faithful reproductions of reality, or using simpler schematics representing just connections between parts. The representation method determines the level of detail of solutions as well as the type of solution that can be generated, and includes relative connection and constraints between design components.
2) Generation: this is the act of putting together design components to form a unified system. For example, a generation methods is constituted by a set of rules or rigorous regulations that govern the generation of new designs through combination of parts. It is the formalism that directs the way solutions can be generated.
3) Evaluation: this is the activity related to the assessment of generated solutions. Each solution is evaluated to determine how well it meets the objectives of the search. The evaluation system makes use of simulation and analysis tools to extract metrics that are used for the evaluation of the design objectives.
4) Search: this is the activity that guides the generation in the preferred direction and determines how different solutions compare to each other. Optimisation and search methods are used in this stage.

Fig. 1.2 presents these activities (and related sub-activities), showing how synthesis develops in an iterative cycle, rather than in a single moment. From this simplistic framework it is understandable how the activities must be thought as interconnected and how complex it is to formulate efficient and innovative methods. The generative technique presented in this research has been developed following the architecture in Figure 1.2. As it will be shown, the choices made for its implementation favoured the exploration of the research targets.

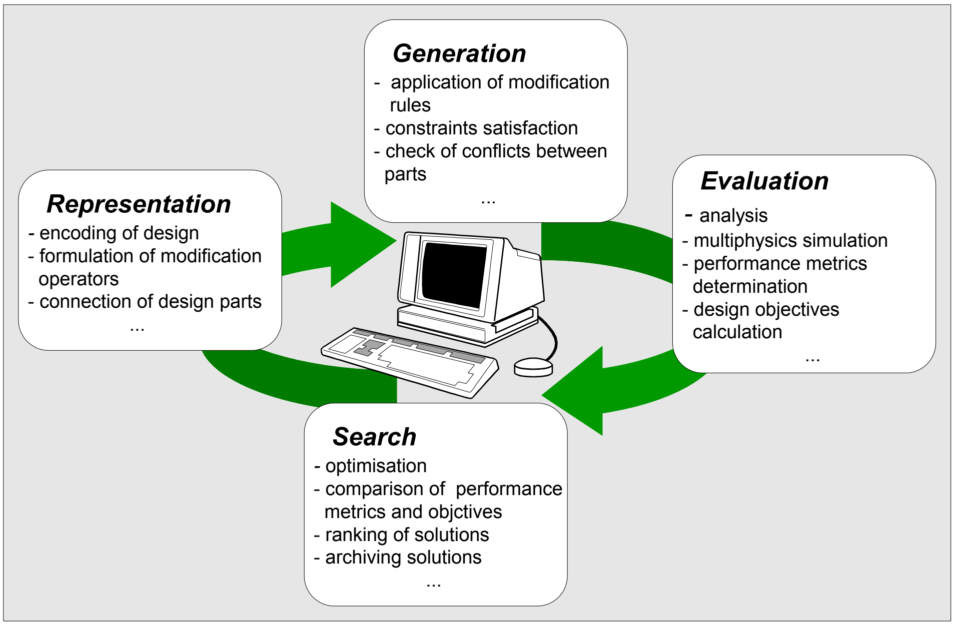


Figure 1.2: Computational synthesis methods architecture: activities interaction.

1.4 Microelectromechanical Systems Design: a Case Study

Microelectromechanical systems (MEMS) are the design domain of application chosen for this work. Although the name ‘MEMS’ suggests the coexistence of mechanical parts with electronics or electricity (or both), the concept of MEMS has grown to encompass many types of small devices, including thermal, magnetic, fluidic and optical systems, with or without moving parts [Senturia, 2001]. In practice, all MEMS share the following characteristics [Senturia, 2001]:

- Minimum feature sizes are of the order of a micron $(10^{-6} m)$
- They can involve both electronic and non-electronic elements
- They perform functions that can include signal acquisition (sensing), signal processing, actuation, display and control, as well as serving as vehicles for performing chemical and biochemical reactions and assaysThey are mainly fabricated using processes based on Integrated Circuit fabrication techniques and material (silicon).

The history of MEMS is a recent one: known since the Seventies and recognised as having great potentials in the Eighties, MEMS saw their commercial success only in the Nineties. They are nowadays commonly employed and well-accepted solutions over a wide range of technical problems [Petersen, 2000]. These days total sales in MEMS market is around

$10bn and are tipped to have a 20% potential annual growth. Markets that make use of MEMS can reach hundreds of billion in sales (for example the printers market).

Among the reasons why MEMS where chosen as a case study for this work are their complexity and their multidomain nature, that allow for exploration of the high-level properties of synthesis methods listed in 1.2. Difficulties in MEMS design are related with scaling effects of physical phenomena at microscale, material losses and damping effects. Also, the integration with electronics and the coupling between several physical effects used to control products behaviour, require interdisciplinary knowledge. Many CAD systems for analysis and manufacturing of MEMS have been commercialised [Senturia, 1998], but development of synthesis methods lags behind for the aforementioned reasons. MEMS design is still performed by hand with the help of simulation tools, despite the growing interest in computational methods that can facilitate their design and shorten their time to market. This interest is motivated by the high investments in fabrication and the limited yield of new experimental processes. In Fedder's words [Fedder, 1997]:

> *'No rapid design process is available today for MEMS. Only one or two CAD iterations involving simple functional simulations are usually attempted during prototype design. As a result, fabrication replaces simulation in the iterative loop. This is very expensive, since fabricated prototypes often do not meet performance specifications and, sometimes, are not even functional. Full verification of designs requires months of effort, and design optimisation is not realistic in all but the simplest cases'.*

Nevertheless, the potential for advanced automated synthesis methods in MEMS design was identified more than a decade ago in the findings and recommendations of the NSF (National Science Foundation) Workshop on Structured Methods for MEMS design, held at Caltech in November 1995 [Antonsson, 1996]. In this occasion, it was recognised that MEMS share many beneficial attributes with other areas that have highly developed and automated design methodologies, such as VLSI (Very Large Scale Integration) [Antonsson, 2001]. Among these attributes, Antonsson identifies the following:

- The fabrication uses methods similar to those used for VLSI
- The systems are assembled from many instances of a small number of component
- Largely planar geometry (21/2 dimensional)

- Limited mechanical coupling between components.

For all the aforementioned reasons, MEMS represent the ideal design domain for the investigation of synthesis methods proposed here, as well as offering countless case studies for application. In particular, this work focused on the design of microresonators, for their importance and variety. Finally, a broad literature on computational synthesis of microresonators offered sound benchmarks for the validation of the developed method.

1.5 Thesis Outline

This first chapter has highlighted issues related to the development and use of synthesis methods and has motivated the investigation carried out in this work. Chapter 2 will present background work on computational synthesis, with a special focus on generative tools for MEMS design. Chapter 3 will introduce in detail the novel synthesis method implemented for this research, and will show how ideal and desirable properties such as flexibility, adaptability and modularity have been included in the development of the method. Chapter 4 introduces two MEMS design tasks that will serve as benchmarks to compare the method against existing synthesis techniques. Chapter 5 presents examples that validate the method and test the desired characteristics listed in Chapter 3. Chapter 6 is a discussion on potential applications of the generative tool developed. The thesis concludes with some observations on the future of the method and on of computational synthesis in general (Chapter 7).

CHAPTER 2: BACKGROUND

This chapter presents the necessary background to position this work in the frame of computational synthesis research. The chapter is divided in three sections. A summary of fundamental concepts in optimisation, which are the backbone of generative methods, is given first. Section 2.2 presents a literature review of computational synthesis techniques in engineering design, emphasising recent progresses in the domain of MEMS design. Finally, the last section highlights the contribution of this work to the research field.

2.1 Search and Optimisation

As explained in 1.3, optimisation plays an important role in the computational synthesis approach taken in this research, determining part of its capabilities and effectiveness. The underlying optimisation technique complements the generative mechanism adopted and is a necessary component of the overall synthesis process (Fig. 1.2). Optimisation techniques come in many formulations and are adopted according to the optimisation task considered. This section will explain the reasons behind the use of specific optimisation techniques in computational design synthesis.

2.1.1 Multiobjective Optimisation

An optimisation problem can be briefly described as the search for the minimum (the optimum) of a function that defines it (the objective function). A main difficulty encountered in design optimisation tasks is precisely modelling a problem using a function. The definition of a problem with just one equation can be very complex, if not impossible. When a design problem is modelled instead, the objectives to satisfy are multiple and often contradictory, meaning that a decrease in one objective leads to an increase in another objective.

The goal of a multiobjective (MO) problem is to minimise, in the best way possible, the various objective functions. A MO problem is presented in the following form:

$$\begin{aligned}&\textit{Minimise } \vec{f}(\vec{x}) && \textit{(2.1)}\\ &\textit{subject to } \vec{g}(\vec{x}) \leq 0 \textit{ (m inequality constraints)} &&\\ &\textit{and } \vec{h}(\vec{x}) = 0 \textit{ (p equality constrains)} &&\end{aligned}$$

where $\vec{x} \in R^n$, $\vec{f}(\vec{x}) \in R^k$, $\vec{g}(\vec{x}) \in R^m$, $\vec{h}(\vec{x}) \in R^p$, k is the number of objective functions, $\vec{x}$ is a vector of design variables and $\vec{f}(\vec{x})$ are the set of objective functions to optimise. While the set of constraints $\vec{g}(x)$ and $\vec{h}(x)$ delimits a restricted subspace to be searched for the optimal solution.

When solving a MO problem, the expected outcome is to find one and only optimal solution. In fact, this is an ideal case and there is seldom one 'best solution'. Most of the time, a set of solutions is found, owing to the contradictory objectives. These solutions are optimal but at the same time represent tradeoffs. This means that they do not minimise all the objectives of the problem contemporaneously and rather represent a compromise amongst their values. Of all these solutions, only a small subset will be of interest and a systematic approach is required to rank them. An important idea, which allows defining the solutions that are of better interest, is that of dominance. For a solution to be interesting, there must exist a domination relation between the solution considered and the other solutions. The relation is defined as follows:

We say that a vector $\vec{x}_1$ dominates a vector $\vec{x}_2$ if:
- $\vec{x}_1$ is at least as good as $\vec{x}_2$ for all the objectives and
- $\vec{x}_1$ is strictly better than $\vec{x}_2$ for at least one objective.

Solutions that dominate the others but are not dominated themselves are called non-dominated solutions or optimal solutions in the Pareto sense. The number of solutions selected using the sorting rule based on the definition of domination produces what is called a trade-off surface, also known as Pareto front or archive (Figure 2.1).

The concept of dominance and Pareto optimality in general received considerable attention in design literature since Pareto's initial observations [Pareto, 1896]. As MO searches do not provide a single solution but rather a multitude of solutions, this effective yet simple concept provides designers with a system for: 1) eliminating non-acceptable solutions that

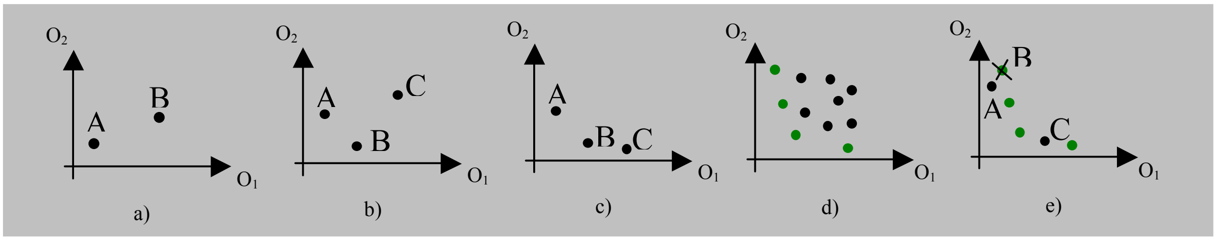


Figure 2.1. Dominance relationships between points in Objective Space. a) Solution A dominates solution B. b) A and B dominates C; A and B are non-dominated solutions. c) Pareto front of non-dominated solutions. d) Solutions in black are cut off the archive. e) Progression of the Pareto front.

are immediately cut off by the surface during the search, 2) rank solutions according to their needs. Especially the graphic results (archives or surfaces) that the ranking system provides is an effective and useful method to pinpoint areas of the design space that are of particular interest to designers and help visualise groups of solutions presenting acceptable trade-offs among design criteria.

It appears clear so far how important it is for designers not only to be able to formulate a synthesis problem through a set of objectives and constraints, but also to have a discriminating method to help them choose the solutions that better fit their needs. Pareto optimality is the powerful concept that helps designers with this goal, as well as presenting another advantage. The variety of solutions in a Pareto archive may in fact offer innovative design alternatives or be inspirational in boosting designers' lateral thinking.

2.1.2 Constrained Optimisation in Design

One of the main issues in dealing with complex optimisation problems is the form of the search space. Optimisation problems for which the variables of the objective function are constrained to evolve in a precisely defined area of the search are called constrained optimisation problems. This is often the case in design optimisation, as approaching an optimum (minimum of the objective function) does not necessarily mean feasibility. If constraints are set appropriately, they should be able to delimit the design space of interest, in order to obtain only viable solutions. At the same time, they should not pose excessive restrictions, so that potentially good solutions are not ruled out. In the set-up of a design optimisation problem, the difficulty in striking the balance between these two requirements is second only to a good definition of the objective functions.

Typical constrained optimisation problems in engineering design are those concerning the geometry of parts (symmetry constraints for example), the space occupied (area and volume constraints), the use of specific materials, movement capability, least energy usage and cost reduction.

We have seen in (2.1) how "hard" constraints $\vec{g}(x)$ and $\vec{h}(x)$ are included in the definition of an optimisation problem. These constraints, considered explicitly within the optimisation, will not be violated during the search of the optimum. In many design problems though, there may be the need to temporarily breach some constraints, in order to explore solutions that lie close to constraints' boundaries. This is particularly necessary in the case of fragmented design spaces, where multiple regions are off-limit for viable solutions. Because of their violable nature, such constraints are called "soft" constraints. These constraints can be transformed into design objectives using penalty functions. In optimisation, penalty functions are used to 'penalise' points that lie outside the feasible regions. Their advantage is that they turn constraint optimisation problems into a series of unconstrained problems whose solutions must converge to the solution of the original constrained problem. In practice, this technique consists in adding a penalty term to the objective function. The penalty term grows when the constraints are violated and is zero in the region where constraints are not violated. In the particular optimisation tasks that will be shown in this work, penalty functions are defined as stand alone objective functions and constraint violation is calculated as the error from the desired value for a specific design constraint. By minimising the penalty function throughout the search process, the constraint violation is minimised to zero and thus the constraint is satisfied. In this implementation, all design objectives and "soft" constraints are assumed to be minimised by the search and must be formulated accordingly. The MO optimisation problem becomes in this case a more general 'multicriteria' problem (as the functions to minimise are both objectives and errors) and if formulated as follows:

$$\begin{aligned} \textit{Minimise } & \vec{f}(\vec{x}) \qquad (2.2) \\ & \vec{\varepsilon}(\vec{x}) = |\vec{e}(\vec{x}) - \vec{e}(0)| \\ \textit{Subject to } & \vec{g}(x) \le 0 \textit{ (m inequality constraints)} \\ \textit{and } & \vec{h}(x) = 0 \textit{ (p equality constrains)} \end{aligned}$$

with $\vec{x} \in R^n$, $\vec{\varepsilon}(\vec{x}) \in R^h$, $\vec{e}(\vec{x}) \in R^h$, $\vec{f}(\vec{x}) \in R^k$, $\vec{g}(\vec{x}) \in R^m$, $\vec{h}(\vec{x}) \in R^p$ (*k*: number of objective functions; *h*: number of soft constraints).
$\vec{f}(\vec{x})$ are the objective functions and $\vec{\varepsilon}(\vec{x})$ are the soft constraints. More specifically, $\vec{\varepsilon}(\vec{x})$ are errors calculated as the difference between the actual values $\vec{e}(\vec{x})$ and the target values $\vec{e}(0)$ for certain design constraints.

Despite the time that may be lost exploring solutions that are not viable, soft constraints offer the possibility to cross over forbidden regions and have been adopted in this work for a more complete exploration of the design spaces.

2.1.3 Design Synthesis and Optimisation: the Role of Heuristic Techniques

A main categorisation of optimisation methods is the one that sees conventional analytical or numerical methods opposed to heuristic methods. The use of optimisation techniques to solve complex engineering design problems often requires resorting to these ‘unconventional’ search approaches. The following subsection explores the advantages of this choice.

In the context of engineering design, the interesting connotation of deterministic methods is their ability to find the optimal solution for a given problem. This result is assured providing that the functions describing the problem satisfy certain mathematical conditions. Deterministic methods (both analytical and search methods) rely in fact on the calculation of first and second derivatives of these functions (gradients and hessians) to direct the optimisation. Unfortunately designers are faced daily with optimisation problems of great complexity and the calculation of these derivatives is not always possible. Objective functions may be non-linear, or may even not have an analytic expression in terms of design parameters. They may be multiple and often contradictory. In the worst scenario, both sources of complications may be present [Siarry, 2003]. Another factor of complexity may be the combinatorial nature of design optimisation problems (i.e. the type of decision variables are permutations on a set of numbers of finite size), leading to very large design spaces and complex objective functions with many local minima. Since gradient-based techniques are likely to struggle in the aforementioned cases [Reeves 1996], design optimisation tasks are more commonly carried out using heuristic search. Heuristic search techniques are methods that seek good (i.e. nearly optimal) solutions at a reasonable computational cost without being able to guarantee optimality, and possibly not feasibility [Reeves, 1996]. Nevertheless, such methods share the following advantages:

1) They can be at least partly stochastic. This characteristic makes them able to handle combinatorial explosions of possibilities and multi-modal problems
2) While gradient-based algorithms tend to get stuck in the local minima of objective functions, typical of synthesis problems, heuristic methods are more likely to find global minima (or get close to them). This is guaranteed by a deeper and thorough exploration of the design space executed through climbing movements that allow a temporary increase of the objective function (for example return to base mechanisms) [Siarry, 2003].
3) They have the advantage of being direct, which means they do not need to compute the derivatives of the objective function to direct the search.
4) They are inspired by specific analogies (with physics or biology for example) but nevertheless adaptable to a large range of problems.
5) They are able to guide, in a particular task, other specialised search methods (for example, another heuristic search or a local exploration method).

These methods have also some disadvantages, which make them not the ideal choice if the intention is to use them to speed up the design process:

1) They show difficulties in tuning the parameters of the method, and require high computation time (although, in the case of complex optimisation tasks, exhaustive explorations of design space with deterministic techniques would also fail in terms of time).
2) Among the advantages seen before, in principle they can find or approach a global optimum that is hard to reach (nearly optimal solutions). In fact heuristic techniques do not guarantee an optimal solution to be found, but rather only ‘optimally directed solutions’ [Cagan, 1993]. Optimally directed design is an approach to design optimisation that directs the design generation towards the numeric range of a global optimum [Shea, 1997].

3) They are often expected to generate solutions from scratch or minimal starting points/designs, requiring also the exploration of procedures that govern the growth and modification of design spaces. As a consequence, search problems become more onerous and time-consuming, revealing the case-sensitive applicability of these methods, whereas it would be desirable to keep them as ‘flexible’ as possible, especially for engineering design purposes [Vale, 2003].
4) Most of the heuristic techniques have been tailored specifically for a single problem.

Nevertheless, the ability of these methods to overcome obstacles that deterministic methods cannot even approach makes them the ideal partners to solve engineering design problems. Of particular interest for synthesis problems are stochastic methods. Stochastic optimisation algorithms incorporate probabilistic (random) elements either in the problem data (objective function, constraints) or in the algorithm itself (through random parameter values, random choices, etc.), or in both [Spall, 2003].

These methods are particularly suited for synthesis not only because of their ability to overcome combinatorial problems and get out of local minima. The randomness introduces in the search a non-prejudicial element, typical of creative designers in action, that brings about novel and alternative designs.

Heuristic search techniques have been found to be not only effective in dealing with complex design problems, but also to be particularly adaptable in dealing with the multiobjective nature of synthesis problems. A number of such procedures have hence emerged as suitable candidates for solving difficult MO design problems, generating indecision when it comes to choosing an optimisation method to solve a design task. At present, there have been few attempts to classify these methods [Miettinen, 1998; Takbi, 1999]. For many of these approaches, their success have been often dependant on the nature of the design problems and objectives. Others are instead particularly adaptable and efficient and have become the core methods used in design synthesis.

Among the latest, procedures have progressed from the use of simple generate-and-test methods, such as Monte Carlo Sampling [Metropolis, 1949], to the popular field of Evolutionary Multiobjective Optimisation [Fonseca, 1993]. EMOs have provided a rich variety of contributions to multiobjective optimisation with algorithms such as MOGAs (MO Genetic Algorithms) [Holland, 1975]. Efforts in adapting other heuristic optimisation techniques have also resulted in the development of MOSA algorithm, which are derived from Simulated Annealing techniques [Suppapitnam 2000; Kirkpatrick, 1983]. Other methods in design synthesis include Tabu search [Glover and Laguna, 1997] and Swarm Particles Optimisation Algorithms [Kennedy, 1995]. All these techniques demonstrated to be particularly useful and effective in the field of design synthesis and amongst the most popular. Nevertheless, in the domain of complex design problems, where multiple design criteria and a desire for more diverse solutions are a necessity, the demand for exploration techniques is always growing. Creating appropriate heuristic search to cope with specific demands is counterbalanced by the strong need for methods that are flexible and that allow designers to use them in different design tasks. In both cases there is room for exploration of new heuristic techniques and for the improvement of existing ones.

2.1.4 Size, Shape and Topology Optimisation

An important classification of optimisation problems in design is based on the nature of the design variables. Size optimisation problems include all those cases where the variables are discrete parameters of the elements used in the design, such as their size. Shape optimisation entails finding the shape of the design elements that optimise the overall design. Whereas topology optimisation is concerned with the optimisation of the entire layout of the design. While size optimisation problems are often simple and solved via deterministic methods, the most important result to test the efficiency of a stochastic method is its ability to supply entire new topologies. Topology optimisation has been of central importance to test the generative method developed in this work, although size optimisation cases have also been included in testing the adaptability of the method to problems of different nature. This work presents also a general outlook at shape optimisation problems and the potentials of the method in such area. The method developed demonstrated to be a versatile instrument for each of these optimisation types.

2.2 General Classification of Computational Synthesis Methods

For almost half a century computational design synthesis has been an active research field [Cagan, 2005]. Already in 1969, Simon was providing the foundation of automated engineering synthesis methods with his paper ‘The Science of Design’, published in his book ‘The Science of Artificial’ [Simon, 1969]. Today, a wide range of problems may be solved through these techniques, based either on deterministic or stochastic optimisation algorithms.

The breadth of methods and results existing under the name of computational synthesis is so vast that it is almost impossible to review them all. The purpose of this section is instead to give an insight into how these methods can differ and be categorised, and mention some interesting fields of application.

As seen in 1.3, computational synthesis methods can be thought of as a set of activities. Cagan [Cagan, 2005] introduces an interesting classification of synthesis methodologies following this division of tasks. Any implemented system that designs automatically must include some semblance of these activities (namely representation, generation, evaluation and guidance). Starting from this division, Cagan explains how the various methods differ. Each of these tasks can in fact be accomplished in different ways, creating a variety of methods. For example, the representation activity can be:

1) Function based
2) Form based
3) Vector based
4) Graph based
5) Grammar based

The generation activity can range from the most random number generator to the most sophisticated agents simulating human thoughts as a design evolves [Cagan, 2005]. For the evaluation, the choice of how to implement it depends exclusively on the design task examined and on the engineering domain of operation. As for the guidance of the search, again a multitude of choices are available, ranging from genetic algorithms (GA) to Simulated Annealing (SA), to name two amongst the most popular.

It is understandable how, from the combination of any four of these activities, a variety of computational synthesis methods can be formulated. A detailed literature review of synthesis methods can be found in [Antonsson, 2001] and [Chakrabarti, 2002].

However, at this point a concise review of some synthesis methods is useful in order to give an idea of the variety of the design domains that have been investigated through computational synthesis, and of the progresses reached in each field. The classification introduced for this brief review is based on the engineering field of application for which the methods have been conceived (structures, mechanics, robotics, electric). This classification allows grouping generative methods developed exclusively for MEMS applications in a stand-alone section.

Structures have been the first field of application for synthesis techniques and a vast academic literature is available in this domain, also called structural topology optimisation. Although there has been a modest use of these methods in industry, this field of application offered better commercial uptake than other engineering design fields. Especially in building design practice, topology synthesis has been investigated for many years with a vast employment of techniques, ranging from the use of discrete and continuous topology optimisation, to evolutionary based searches. A valuable example of computer based conceptual design methods for buildings can be found in Park and Grierson [Park, 1999], which makes use of a multicriteria GA to generate Pareto-optimal conceptual designs of office buildings. Other interesting contributions in this area include Shea's work based on structural shape annealing [Antonsson, 2001] and Baldock's recent work on structural optimisation in building design practice [Baldock, 2007]. A review of the state of the art, drivers and barriers in computer-based conceptual design research for structures can be found in Grierson [Grierson, 2002].

Probably the most successful field of application for computational synthesis methods is the electrical domain. Koza [Koza, 2003] gives a complete review of all the devices synthesised using Genetic Programming, listing the patented circuit types that have been automatically generated, and showing how the electrical domain is also one of the most prolific fields of application for generative design tools.

Compared to the structural domain, fewer computational approaches for mechanical design synthesis exist. The difficulties found in this field relate to the fact that mechanical design covers a large and diverse range of problems where designs consist of many different types of interrelated functions and components. As a consequence, desired performance criteria cannot always be easily translated into quantifiable objectives. Successful research in the mechanical domain includes a variety of applications. To name some examples, Campbell's A-Design [Campbell, 2000] is a design methodology that incorporates multiagent systems into automated component-based design synthesis in the electromechanical domain. Starling [Starling, 2005] automates the synthesis of gear systems using a grammatical, simulation-driven method. Schmidt et al. [Xin-Schmidt, 2000] present a graph-grammatical approach to synthesis of epicyclic gear trains. Rudolf et al. [Rudolf, 2005] introduce a multi-domain Design Compiler able to automatically link the generation of graph-designs to CAD models and FEM analysis tools. A brief outline of methods for automated synthesis of mechanical systems, with primary focus on applications domains involving kinematics and mechanical motions, is given by Kota [Kota, 2003].

Other applications for computational synthesis are multi-domain engineering fields. A successful example in this sense is the work of Lipson and Pollack in the electromechanical domain [Lipson, 2000]. Lipson and Pollack achieve the automatic design and manufacture of robotic lifeforms by evolving electromechanical systems, controlling the generative process with a simple fitness function. This is a particularly successful example of application, although synthesis of elecromechanical and mechatronic systems has received in general much less attention than other fields of engineering, due to the difficulty presented by their multidisciplinary nature. Microelectromechanical systems are an example of multidomain field of application where synthesis techniques have struggled to prove their usefulness. A detailed review of computational synthesis methods for MEMS design is given in the next section.

Some of the synthesis methods mentioned here present similarities with the method introduced in this work, meaning that they provide a complete generation of solutions from their conception to their validation and analysis and none of these steps are executed manually. Some methods may also reach the automated prototyping stage. This is not always the case with synthesis methods, which may easily be incomplete design tools. Especially in cross-discipline fields, it may be difficult, for example, to set up analysis and evaluation activities, and methods are limited to the creation of design alternatives later evaluated by hand.

This incompleteness is only an example of the limitations of synthesis methods that delayed their adoption at large scale by designers. Another limitation arguable from literature review is that many techniques are formulated for a single design domain. In general, difficulties arise from using a method in multiple domains or even for multiple design tasks in a single domain. Although some techniques have began to be used in industrial practice mainly in electronics, yet many open issues and research challenges remain [Cagan, 2005].

2.3 Automated Synthesis of MEMS

The most complete review of synthesis methods for MEMS design has been presented by Ananthasuresh [Ananthasuresh, 2003]. This thorough investigation lists the methods according to the actuation principle of the devices synthesised, reflecting the well-known MEMS characteristic of operating in multiple energy domains. The approach followed in this review gives a broad vision of where the research efforts have been directed in automating MEMS synthesis [Achiche, 2006]. MEMS design is a complex process and automation can be introduced at any step. For this reason the systematic modelling subdivision proposed by Senturia [Senturia, 2001] is used here as a classification of design levels. Such categorisation provides the stages where different design paradigms occur. Four modelling levels are proposed: System, Device, Physical and Process, as shown in Figure 2.2 [Senturia, 2001].

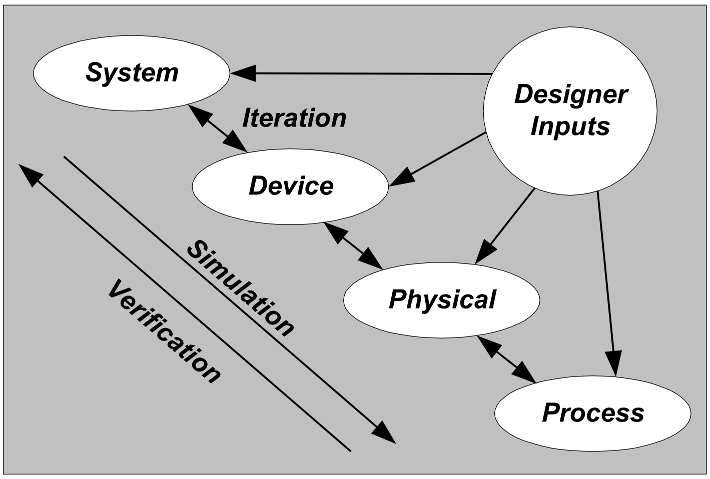


Figure 2.2. Different Modelling Levels for MEMS [Senturia, 2001]. While designing MEMS devices from top to bottom is called Simulation, the inverse process is called Verification.

The design levels give a clear framework for structuring the review of automated synthesis of MEMS. For each level there has been research carried out on how to automate design. Each of these levels will be reviewed here, although particular relevance will be given to works on automated synthesis at the Device level, which is the focus of the present work. Most of the works described in this section deal only with one level, as automated synthesis methods for MEMS are still far from being able to embrace the complete design process.

System Level

The interactions of a micro-component with its environment and electronics is designed and simulated at this level. This is the home of block-diagram descriptions and lumped-element circuit models, either of which can be used, and both of which lead to a coupled set of ordinary differential equations (ODE) [Senturia, 2001]. Due to the multi-domain nature of each device, design and analysis at system level is very complex and requires access to simulation tools able to deal with entire systems and their associated circuitry. The construction of behavioural models using these tools is a manual activity, inefficient and prone to error, and computation cost is typically very high.
Therefore a common representation that encompasses multiple energy domains is needed for analysing the whole system.

The most popular approach to performing synthesis at system level has been based on the integration of ‘macro-models’ in the system-level model. ‘Macro-models’ or ‘reduced order models’ are a form that captures essential physical behaviours of the components of the system (device level), and simultaneously is directly compatible with a system-level description. The macro-models can be rapidly calculated and directly inserted into a system-level simulator. The most typical way of creating macro-models is to use the lumped-element approach. Various techniques for automatically extracting MEMS macro-models have been proposed. Although these techniques are arguably device level synthesis methods, they are reported here because of their specific use and integration at system level synthesis.

The development of the AutoMM CAD tool, which supports automatic generation of dynamic macro-models for a broad class of MEMS devices such as the Analog Devices

ADXL76 airbag accelerometer, was proposed by Swart et al. [Swart, 1998]. For linear or weakly non-linear systems, the macromodels can be easily obtained by reducing the finite-element (FEM) or finite-difference (FDM) formulation using the Arnoldi Algorithm [Silveira, 1999] or the quadratic techniques [Bechtold, 2003; Chen, 2000; Wang, 1998; Yang, 2004]. These approaches are very effective and accurate for linear systems with complicated geometries, but are incapable of capturing the dynamical behaviour for non-linear models. An approach used for highly non-linear system over a wide range of operating conditions was proposed by Rewienski et al. [Yang, 2001; Rewienski, 2001], where the algorithm presented has been demonstrated to be effective and accurate for highly non-linear systems, but its performance still depends on a few parameters, which need to be adjusted more or less arbitrarily for a given application example. The reduced-order models generated by the Karhunen-Loeve-Galerkin approach [Hung, 1997; Park, 1998], on the other hand, are based on the basis functions extracted from an ensemble of the snapshots of the physical fields (e.g., pressure distribution or temperature distribution) under certain actuation conditions, and have also been proved to be effective and accurate for macromodelling non-linear systems. However, this approach requires expensive coupled-domain FEM/FDM runs to provide enough snapshot data for extracting basis functions. Furthermore, the efforts required for implementing and simulating the FEM/FDM runs increase exponentially if any new physics domain is introduced into the system [Yang, 2004]. Other automatic extraction methods using control theoretical approaches, Krylov's subspace-based approaches and commercially available reduced order modelling can be found in work conducted by Antoulas [Antoulas, 1999], Ostergaard [Ostergaard, 2000] and Antonsson [Antonsson, 2001] respectively.

Although giving good results mostly using artificial intelligence techniques, these approaches were mainly applied to existing system models or very simple combinations of MEMS components. Applications with more complex devices are needed to validate the level of accuracy and robustness of such techniques.

Device Level

Already at the end of the last decade the investigations carried out by Fedder and Cagan's research groups actively contributed to the field. The work developed at Carnegie Mellon University by Fedder, Iyer and Muhkerjee [Fedder, 1997; Mukherjee, 1998; Iyer, 1998] is the first successful implementation of a computational synthesis technique for MEMS design. This approach consists in modelling the design task as a formal numerical optimisation problem, and then solving it with powerful optimisation techniques. The process of modelling the design task involves determining the design variables and the design constraints. Different objective functions are implemented in order to drive the synthesis towards preferred design types. The synthesis of solutions is achieved through an optimisation algorithm, which seeks to minimise the objective functions while satisfying the constraints. Once the optimisation results in a valid design, i.e. a set of values for the variables satisfying all the design task specifications, these values are fed into a parameterised layout generation tool called CaMEL (Consolidated Micromechanical Element Library). The outcome of this synthesis procedure is an optimal synthesis tool that allows rapid exploration of micromechanical design issues and objectives through rapid generation of design solutions. The synthesis technique has been successfully applied to the synthesis of accelerometers [Mukherjee, 1999] and folded-flexure electrostatic-comb-drive microresonators. For this last task resonator layouts were successfully synthesised for minimum area, minimum amplitude of the comb drive voltage and maximum displacement at resonance, using various input frequency specifications ranging from 3kHz to 300 kHz [Mukherjee, 1998].

Agarwal and Cagan’s research, focusing on shape-based representation theories behind automated synthesis, offers a unique example of grammar for generation of MEMS [Agarwal, 1999; 2000]. However, the proposed architecture for the creation of a resonating structure is not coupled with any simulation tool.

All the approaches mentioned so far require prior specification of building blocks by the designer. As an attempt not to restrict these methods to specific types of MEMS devices, a high-level MEMS conceptual design synthesis model was proposed by Gibson [Gibson, 1999], where a case study of an optical processor manually designed using the MUMPs process is presented. This model uses MEMS devices’ behaviours as building blocks. These blocks are specified at a higher level of abstraction than in the case where structural components are used as building blocks. However, there is no elaboration, nor any new example that illustrates how the building blocks can be constructed for design synthesis.

A novel approach to computational synthesis of MEMS is the one developed by Campbell [Campbell, 2000]. In this work, an automation technique called A-Design is implemented combining shape grammar with nodal simulation, and is applied successfully to the synthesis of ADXL accelerometers manufactured by Analog Devices. The algorithm used in this method is founded on optimisation techniques but incorporates software agents with an adaptive search for new design solutions. Starting from the definition of a library of components, a description of inputs and outputs of the design process and of the objectives to optimise, the method is able to create novel configurations for MEMS problems where a predetermined configuration is unknown.

A synthesis tool for the layout design of complex or arrayed MEMS devices that consists of a number of similar components is proposed by Juneidi [Juneidi, 2001]. In order to use such tool, the method requires designers to model topologies of MEMS devices composing the arrays first.

Ongkodjodjo [Ongkodjodjo, 2002] proposes an optimisation and design of MEMS devices (using a microgyroscope as an example) based on a simulated annealing algorithm. The results of the simulation algorithm were verified and validated with the finite element method and the boundary element method of IntelliSuiteTM [IntelliSense, 1991].

The method developed at Berkeley by Zhou et al. is one of the most successful examples of automated synthesis at the device level [Zhou, 2001, 2002; Kamalian, 2002, 2004]. This work proposes the use of Multi-Objective Genetic Algorithms (MOGAs) to automate synthesis of MEMS devices. This non problem-specific technique guarantees high robustness and capability to optimise for multiple design objectives. MOGAs use an evolutionary approach to develop a population of optimal solutions. Given a higher-level description of the device's desired behaviour, an initial population of candidate designs is generated randomly from a number of parameterised MEMS components such as anchors, beams, electrostatic gaps and combs. Both the topology and size of the device are generated by the algorithm. Each design is checked for geometrical validity and its performance evaluated. This process continues until a set of 'Pareto optimal' solutions are synthesised. As described in section 2.1.1, Pareto optimality provides MEMS designers with a family of 'equally good' or 'non-dominated' solutions. This method incorporates a MEMS simulator SUGAR [Zhou et al., 1998] for the evaluation of designs. SUGAR is a

MEMS simulation package developed at the University of California at Berkeley. Zhou and Kamalian also compare MOGAs and the Single Objective Genetic Algorithm (SOGA) against Simulated Annealing (SA) optimisation [Kamalian, 2002], showing that SA can in some cases synthesise valid designs faster then Genetic Algorithms. However its dependence on a single objective function and the difficulty in finding the global optimum indicate that it is a less robust method for many MEMS synthesis problems. Kamalian has also explored the role that geometric constraints play in a computer-aided MEMS design system based on genetic algorithms [Kamalian, 2004]. The case studies investigated so far at Berkeley's research are electrostatic actuators [Zhou, 2001] and meandering resonators [Zhou, 2002]. The latter example was used as a benchmark problem for this research, due to the similarities that the MOGA approach has with the technique used in the present work. Both methods are, in fact, stochastic non problem-specific, and adaptable to a large variety of design tasks. The example is detailed in Chapter 4.

Another design synthesis method using behavioural building blocks is the one proposed by Deng at al. [Deng, 2004], accomplished by both forward and backward synthetic search strategies. Here again, the method was applied to existing MEMS devices but new device concepts were neither proposed nor optimised.

An evolutionary approach using genetic programming over a bond graph representation to achieve automatic design of a MEMS system-level lumped element parameter model is proposed by Fan et al. [Fan, 2003], where an initial (starting) design is necessary to start the evolutionary algorithm. The method has been successfully applied to an automated synthesis of an RF MEM device, namely, a micro-mechanical band pass filter.

Much attention has been devoted to the automated synthesis of microcompliant mechanisms in the past. Compliant mechanisms are used to transfer or transform motion, force or energy, but do not consist of rigid links connected by movable joints, and rather gain their mobility from the deflection of flexible members. These mechanisms have been used in this work to benchmark the method and will be presented in further detail in Chapter 4. Work done in computational synthesis of compliant mechanisms is almost exclusively in the area of continuous structural topology optimisation methods. These numerical techniques are a versatile synthesis method that can automatically generate structural forms from function-level specifications. A mechanical synthesis problem is posed by defining arbitrary regions within which a device to be designed must fit. Forces and fixed space portions are known. The region is then discretised into finite elements. The task of finding a device's layout is reduced to selective retainement or removal of material in different regions. This is done by assigning a design variable to each element in the discretised model. This variable determines the state of the elements. If the variable's value is '1', the material exists. While if the value is '0', it creates a hole. In order to use continuous optimisation methods, the state is smoothly varied between '0' and '1'. This variable multiplies a material property such as Young's modulus, at the corresponding site. Performance-related requirements of a design determine the objectives and constraints, which are defined in terms of this element variable. For compliant mechanisms, the typical design criteria are related to stiffness, flexibility and strength. But several different formulations exist, combining one or more of these criteria. Numerical optimisation algorithms search the design space to find the minimum of the objective function. Feasibility (i.e. satisfaction of the constraints) is also maintained throughout. Starting from a design given by the user and called the initial guess, the algorithm searches the design space for better designs until an optimum is found. At every design point a search direction is identified and the step along that direction is computed. Determination of the search direction distinguishes one algorithm from another [Ananthasuresh, 2003]. A more general approach consists in varying the artificial material density ρ_i at every point i in the design domain (continuous material density parameterisation). When ρ_i reaches very small values, it implies that the element is made of an artificially very soft material and thus making it virtually absent from the structure. If it reaches high values, that element forms the solid portion of the resulting optimal compliant mechanism. The artificial density approach was developed on the basis of a more rigorous method called the 'homogenisation method' developed by Bensdøe and Kikuchi in 1988 [Bensdøe, 1988]. Another factor that distinguishes one kind of topology optimisation approach from another is the parameterisation of the design space. The easiest topology optimisation design parameterisation is to use an exhaustive set of truss or beam/frame elements in the design domain to approximate the continuum domain and to vary their individual cross-section dimension by defining them as design variables. When the area of the cross-section of an element goes to zero, that element is removed. Thus, after the optimisation procedure converges, some elements will be removed from the original exhaustive set. The remaining elements will define the topology and shape for the compliant mechanism. The exhaustive set of structural elements is known as the 'ground structure' [Howell, 2001].

Many approaches to the synthesis of compliant mechanisms using structural topology optimisation have been implemented, their differences consisting mainly in the formulation of the optimisation problem. The first paper to appear on the application of topology optimisation for the design of compliant mechanisms was by Ananthasuresh [Ananthasuresh, 1994]. Ananthasuresh and Saxena apply synthesis of compliant mechanisms to AND logic gates and amplifiers [Saxena, 1998, see Howell], using ground structures of linear trusses and a flexibility-stiffness formulation for the optimisation problem. PennSyn, the topology synthesis software developed at University of Pennsylvania by Ananthasuresh et al. provides, through a graphical user interface, an automated design route from perfomance specifications to fabrication of fully compliant mechanisms [Saxena, 2000].

Sigmund [Sigmund, 1997] introduces a continuous topology optimisation formulation to make the optimal mechanism depend on the stiffness of the workpiece, including also a limitation on the amount of material to be used. The method is a general and versatile technique able to find solutions to many structural synthesis tasks. The result of this work is TOPOPT (www.topopt.dtu.dk), a publicly available software. Sigmund applies his method to synthesise micro-grippers, force inverters and displacement amplifiers [Sigmund, 1997]. The latter will be detailed in the next section. The work developed by Frecker [Frecker, 1999] focuses on a multi-criteria optimisation formulation for topology design of compliant mechanisms with multiple output requirements, to produce truss and continuum type mechanisms in two dimensions and truss type mechanisms in three dimensions. At the University of Michigan's design laboratory, the work developed by Kota et al. consists in adapting a formal structural optimisation technique based on the homogenisation method by Bensdøe and Kikutchi [Bensdøe, 1988]. An improved and robust objective function and its implementation for a network of linear beam elements [Kota, 2000] is applied to the synthesis of crimping mechanisms. An interesting comparative study of topology optimisation techniques used for compliant mechanisms synthesis can be found in Deepak [Deepak, 2008].

Although many of the aforementioned works are successful examples of automated design techniques, none of them has demonstrated to be general or robust enough to be commercialised. Antonsson [Antonsson, 2001] has given a complete overview of the different approaches undertaken at the Device level.

Physical Level

The Physical level of MEMS design process is probably the one that makes most use of computational tools. This level addresses the behaviour of real devices in three-dimensional continuum. The governing equations are typically partial differential equations (PDE) [Senturia, 2001]. A variety of analysis tools using finite-element, boundary-element, or finite-difference methods are available for simulation at the physical level. This refers to field solvers for any physics such as solving Finite Element Models for mechanics, Navier-Stokesfor fluids, or Maxwell equation for electromagnetic devices, to name a few. Other discretisation techniques are Boundary Element methods, Finite Volume and Volume of Flow Methods [Van Kuijk, 2005]. IntelliSuite by Intellisense, ANSYS, COMSOL are just few of the many commercial packages used to perform behavioural simulation of MEMS. Some of these tools may also present an in-built optimisation toolbox that allows for local parametric optimisation. However it is not the scope of this review to report the various algorithms that exist for automatic meshing of 3D models.

Process Level

Process simulation consists of 3D numerical simulation of process, chemistry and physics to produce accurate models after material addition and subtraction (mostly depositing and etching). Typically process simulation for MEMS is limited to the simulation of wet chemical anisotropic etching of crystal silicon and calculation of doping profiles created by implantation and diffusion. The considerable variety of process steps in MEMS fabrication, the large number of unknown inputs, and the complexity of the calculations, render the process simulation extremely time consuming or in most cases not possible due to the lack of adequate simulation tools [Senturia, 2001]. For this reason, MEMS CAD tools used at this level favour the process emulation approach, which takes 2D masks and a description of the fabrication process to create a geometric 3D solid model. These models are built rather quickly and cannot subsequently be used for physical modelling. Other recent techniques such as the ‘voxel-based emulation’ [Tushar, 2005] do not have this limitation and are robust to 2D-mask errors, making it possible to build highly detailed, realistic-looking virtual prototypes. Due to the complexity of the MEMS process, few attempts on automatic mask generation and automatic process planning have been developed.

An automatic method for synthesizing MEMS mask-layouts is proposed by Antonsson et al. [Li, 1998]. This method incorporates a forward simulation of fabrication into a general evolutionary algorithm loop, and was further developed by introducing robustness and some process variables in the learning algorithm [Ma, 2000]. The evolutionary techniques presented were applied to simple 2D ½ (extrusions) models. A design tool that calculates the required 2D mask set producing a given 3D model by investigating the vertical topology to the model through a trial mask set was proposed by Schmidt et al. [Schiek and Schmidt, 2005-2006]. This work was based on the development introduced by Cho et al. [Cho, 2002], where a new process planning technique that uses a three-dimensional surface micromachined structure as input is proposed. The method decomposes an imported surface micromachined model into a set of three-dimensional models, each of which has geometry compatible with the fabrication process, and then groups them for efficient layer generation. Subsequently, the fabrication order and the masks for all the layers of the structure are generated. A systematic mask synthesis method for surface micromachined MEMS was proposed by Ananthakrishnan et al. [Ananthakrishnan, 2003]. This method generates the mask automatically given a 3D geometric model of the MEMS device. The process sequence is referred to as an 'inverse problem'. This necessitates a systematic solution of the direct problem, which involves automatically generating a geometric model of the MEMS device given the masks. This work presents a systematic and implementation-independent framework for the geometric modelling of MEMS in order to solve both the direct and inverse problems for general surface-micromachined devices.

From a general point of view, although the aforementioned works approach the task of automating the fabrication process in a satisfactory manner, it must be said that these are rare examples of success. Moreover, they only address simple geometries and use few process variables. A more standard way of MEMS processing is still very much needed in order to facilitate this costly stage and streamline the entire design process.

2.4 Thesis Contributions

Implementation issues and obstacles to a widespread use of synthesis methods have been highlighted both in Chapter 1 and in 2.2. This work stems from the analysis of the gaps left by previous research on the development of synthesis methods. Rather than focusing on the comparison of a newly defined synthesis method with existing generative techniques, this research investigates possible improvements to these methods through consideration of fundamental properties of synthesis methods for multidomain and multitask use.

The first contribution of this work is in the way synthesis methods must be devised before their actual implementation: the inclusion of high level properties must be decided a priori, during the early stages of a method's definition, and before its main functionalities are outlined. The definition of these properties and their presence influence in fact the definition of any other features of the method. The properties explored in this research, often neglected in other works, are flexibility, scalability, adaptability to different design domains and, most importantly, the possibility of integration with other design tools and evaluation methods to use in a big variety of scenarios.

A second contribution introduced in this work is the use of a novel generate-and-test search method for the exploration of the design space. This choice was dictated by the need for a search method that would not hinder the ability of the method to generate new and original solutions. As seen in 2.1.3, generate-and-test methods have demonstrated the ability to improve exploration capability and efficiency of a search, where efficiency is defined as the ratio of design solutions' quality to the number of objective functions' evaluations used. Rule based design methods, such as the one developed for this work, require a rather large degree of flexibility in the exploration engine. Generate-and-test implementation is perhaps the most flexible type of search in this respect, as it offers the advantage of covering the entire design space. Small jumps allow topology to be investigated freely [Vale, 2003]. The approach does not require problem-specific tuning and is very easy to apply to multiobjective search (and to a wide range of different design problems) without modification. In its simplest form, where generation is completely stochastic and acceptance mandatory, the optimisation is not guided by any criteria, but machine learning features can be easily integrated, guaranteeing a less wasteful use of resources during the search [Vale, 2003]. The choice of a generate-and-test method gave the possibility to have an adaptable and expandable search, bringing further flexibility into the synthesis method.

Lastly, another novelty introduced in this work is its application to complex MEMS design tasks. MEMS are the design domain of interest chosen to test the developed method. Nevertheless, the results obtained set an important contribution to the automated design of MEMS. MEMS applications used to test synthesis methods are often quite simplistic and unrealistic. In this work the novelty consisted in the application of the method to design problems that are of common interest to designers and that showed a high degree of complexity. Again, here the goal was to explore capabilities and limitations of automated MEMS design. As it will be shown, the method demonstrated both its enhanced characteristics and its capacity to synthesise a range of performance-driven and novel designs.

CHAPTER 3 – THE CNS-BURST METHOD

This chapter gives a detailed description of the developed generative method. CNS-Burst is a simulation-based, object-oriented multicriteria computational synthesis technique that makes use of basic stochastic optimisation to perform design space searches. Details of the method's structure are given, highlighting the importance of its modular architecture. The method is entirely implemented in Matlab.

3.1 The CNS-Burst Method: Overview

In 1.1 it has been discussed how the aim of automated synthesis is to develop computational tools that give designers a broader support in automating the design process. Not only do these tools provide assistance in the analysis and evaluation phases, but help designers in the creative phase and in the decision process. In order to develop such tools, a systematic approach is needed. A framework will provide the sequence of necessary steps for transforming the design process into an algorithmic solution. Many different approaches have been taken to define a framework for computational design synthesis [Cagan, 2005]. Among the many, Starling [Starling, 2003] and Campbell [Campbell, 2003] present the closest approaches to the one here introduced. Starling proposes a method based on four steps: investigate, generate, evaluate and mediate. Campbell's approach offers a generalisation of many computational synthesis methods, based on a review of general approaches by Antonsson and Cagan [Antonsson, 2001].

Although the approach taken in this work stems from the abovementioned works and is in line with the general framework proposed by Campbell, here the method used is reformulated following a general system theory approach to design, as illustrated in Pahl and Beitz [Pahl, 1996]. This choice offers a sequential representation of the design process (Figure 3.1), as well as showing its iterative aspect. Its sequential aspect also facilitates a more intuitive parallel between its steps and the formal algorithmic formulation of the method (later shown in Figure 3.2). In this framework, the logic behind the design process has been divided into main blocks. Whether this framework is followed by a designer or by an automated process, these steps remain the fundamental and necessary milestones of synthesis. The first step is the analysis of the task, which includes the mathematical formulation of the problem and search objectives. The search for solutions can start from an initial design, which can be an existing well-known solution or an estimated idea of what the solution is going to be like. The search can also start from a basic component of the design, or even from nothing in some cases.

The next step is the generation of a novel solution: new ideas and past knowledge are recalled at this stage, together with some insights on how to modify design components and behaviours that do not meet the desired objectives. In order to come out with the best possible solution, designers usually do not stop their search after their first intuition. Behaviour and design objectives for each generated solution are compared with existing solutions, in order to verify whether the new design has come any closer to the desired objectives of the search. This generative-and-test mechanism (iterative cycle in Figure 3.1) will be repeated until the solution gets close enough to the objectives of the search or until time limitations for the search are reached.

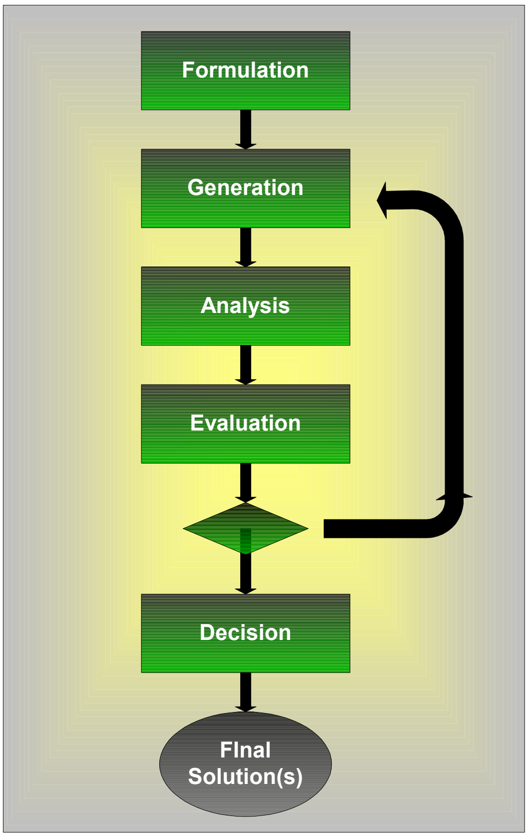


Figure 3.1: Systematic Approach to synthesis: steps of the method.

According to the framework just described, it is understandable how the implementation of an automated synthesis method must follow a general architecture. As anticipated in 1.3, a synthesis method can be though of as a set of activities that follow this framework. Each of these activities is constituted by specific components (or set of components) that, in this particular work, will be called ‘modules’. The main modules of the architecture of this automated synthesis tool are the following:

1) A concise and flexible design representation method: the generated designs needs to be embodied in a virtual object. A spatial representation of designs is essential to perform behavioural analysis of the solution. In this work, a Connected Node System (CNS) design representation has been employed. The method makes use of basic building blocks, called primitives, and nodes to build systems and subsystems of interconnected primitives.
2) A generative mechanism to generate alternative designs: the main idea behind the search method presented is to iteratively modify an initial design (represented by a connected-node system) using a library of modification operators that generate new solutions by combining primitives and altering their connectivity and internal geometry.
3) Integrated simulation for quantitative analysis of designs’ behaviour: generated designs are simulated using external analysis packages. Multiphysics analysis software offers the possibility to analyse designs that span more than one design domain, or just to reuse their capabilities for other design tasks concerned with different engineering domains.
4) An Evaluation mechanism to evaluate design objectives for each generated design: generated designs are evaluated using defined objective functions and constraints. This evaluation makes use of results returned by the analysis module mentioned above. The analysis module is in fact used by the evaluation module itself.
5) An Integrated multicriteria search: this module is the search method used to find feasible and optimised design alternatives. The module directs the search making use of all the modules mentioned above: it applies modifications to existing designs, decides how to apply the modifications, evaluates and compares designs. The search module used in this work has been called Burst, a simple multicriteria generate-and-test algorithm that provides, at the end of the search, a final archive of Pareto-optimal solutions.

All the components of this architecture need to be integrated into a general algorithm that directs the synthesis process. The modules are linked together through of the main algorithm that directs the search and calls the required different modules following the logic expressed by the framework in Figure 3.1. Each module can be thought of as a black box with its own inputs and outputs and completely independent from the other modules. The main algorithm directing the search is represented in Figure 3.2. The synthesis task is formulated as a design optimisation task consisting of design parameters, constraints and objectives. Inputs required to start the search include:

- A maximum number of design evaluations that can be sustained (for computational reasons or for time limitations)
- An initial design
- An optimisation model expressed in terms of objectives to minimise and any constraints to respect.

The first step of the search algorithm consists of validating (in terms of design constraints) and evaluating (in terms of design objectives) the initial design and creating an archive where the initial design and any other Pareto optimal design generated by the search will be stored. Next, the main loop of the search method starts, and stops only if the maximum number of evaluations set initially has been reached. The main module of the search is called Burst algorithm (green steps in Figure 3.2). This search advisor acquires a design from the archive and selects modification operators to apply to this design. The design is then modified and validated. Each candidate design successfully generated by the design modifications is automatically modelled in the chosen simulation environment, where its behaviour is analysed. Using feedbacks from this analysis, the design is then evaluated according to the predefined design objectives and constraints. Finally, the evaluated design is tested for inclusion in the design archive. Designs that satisfy the Pareto optimality criteria are stored to evolve a Pareto-optimal front of non-dominated solutions, according to all the solutions seen throughout the optimisation process. As explained in 2.1.1, a solution is ‘non-dominated’ when, through pair wise comparison, it is superior to any other in the design archive for at least one objective. The Pareto-optimal front develops throughout the synthesis and search process so that the outcome is a set of Pareto-optimal designs. This search loop will be repeated at each iteration until the maximum number of evaluations set initially is reached or the termination criteria are met.

The method described here has been called CNS-Burst method. In the following sub-sections a description of each module will be given.

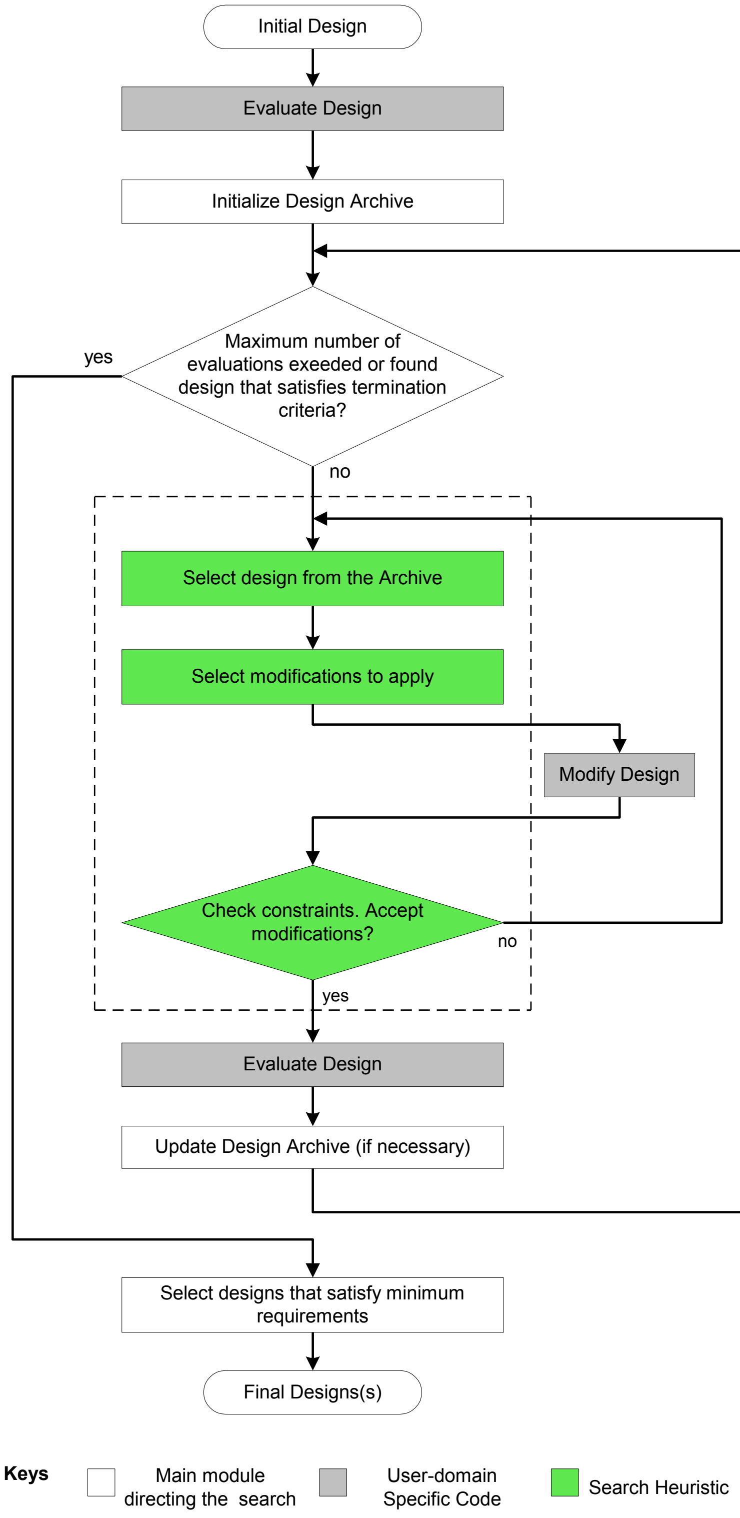


Figure 3.2: Overview of the main algorithm directing the search.

3.2 Representation: the Connected Node System Design

Representing something we want to design can be complex. This is especially true if, like in the case of automated design processes, it is necessary to know a priori all the types of building blocks needed for the final assembly.

The problem of how to integrate a representation method into a computational generative tool is usually solved by implementing a library of parts (or elements, or building blocks) to be accessed during the generation of the solutions. Represented Parts can be more or less similar to what they are in reality, stylised or just symbolic representations of reality.

Even hypothesising that all the parts necessary for the design layout are known a priori, designers' ability to represent a solution is also based on some knowledge of how these parts can be connected and how they can work together for representing a solution to the design task. Hence, any automated approach making use of libraries cannot be exempted from including similar knowledge. In general, necessary characteristics of a representation will include:

- Generality (in the way parts are implemented and treated)
- Easy integration of new parts, if needed: this can be done by keeping the structure of the representation method modular
- Integration of some sort of rules and constraints-checks in the way parts can be assembled.

The representation method chosen for this work has been named Connected-Node System Design Representation. The representation module is itself constituted by modules, each of which will be analysed in the following sections.

The Connected-Node System (CNS) design representation is a generalised representation for a class of design problems specifically oriented towards interconnected systems, such as trusses, MEMS, and electrical circuits. Vale [Vale, 2003b] underlines how the principle of connected-node systems is not claimed to be an original proposal and might also be found under other names in the literature. The formalism was chosen in order to have a flexible design representation that could be used in different design synthesis tasks. Inspiration for this application originates from the MEMS analysis software, SUGAR [SUGAR, 1998], developed at UC Berkeley, which uses very similar constructs to define

MEMS models for numerical analysis. This representation is simple, well known in many design domains (for example circuit representation) and very effective for keeping the structure of the code modular. The main advantage of the representation method is its generality and systems approach.

The CNS design representation employs basic building blocks, called primitives, and nodes to build systems and subsystems of interconnected primitives. Nodes serve as connection points for primitives and subsystems that form the complete connected node system. Nodes are defined by:

1) Their position
2) Their degree of freedom (i.e. floating or anchored nodes)
3) A flag determining whether it is possible or not to change their coordinates
4) A flag determining whether it is possible or not to remove the node itself.
5) A range of value for their coordinates
6) Maximum and minimum number of primitives they can be connected to.

Optional properties for the nodes can be mass, voltage level, force applied to the node and possibility for interacting with primitives only (internal nodes), or with primitives and subsystems (port-nodes).

The implementation of the representation consists of a number of user-coded primitive and subsystem modules. Primitives and subsystems must be written as a separate module and according to a specific format, such that they fulfil the requirements of the CNS definitions. The next sections examine in more detail how systems, primitives and subsystems are built and their characteristics.

3.2.1 Primitives

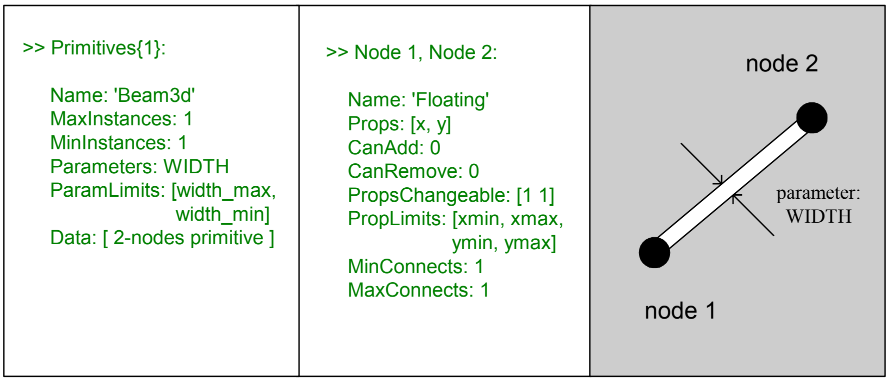
Primitives are basic elements that constitute a system, such as beams, circuit elements (for example resistors), masses or elements of any shapes. The essence of a primitive type is that it should be able to self-organize its internal structure and properties as a function of external specifications of:

- The nodes that the primitive instantiation is connected to in a system

- A list of properties defined as ‘primitive’s parameters’ (such as width for a beam, resistance for a resistor, or mass), which can be either static in the synthesis process or variable.

Hard constraints for the primitive are defined within the primitive object class together with functions that check for violation of constraints specific to the primitive alone and with other primitives. An example of a simple primitive is a solid, round truss element in a truss structure. The primitive type has two nodes to connect to the outside world, and a variable that defines the cross-sectional area. The primitive type object must be written in such a way that, given only the external node connections and the single variable area, the particular instantiation of the primitive type will be fully defined. In this example, it seems obvious to simply define the beam as having end-points coinciding with the node locations, and having a cross-section equal to the externally specified cross-section. In this case, the ‘free’ primitive parameter is the cross-sectional area. Constraints for the primitive (hard constraints) are defined within the primitive description, together with functions that check for violation of constraints specific to the primitive alone and related with other primitives. As it is understandable, each primitive can be seen as an object, with its own properties and applicability. Figure 3.3 shows an example of beam primitive and the description that Matlab uses to instantiate the primitive.

The beam primitive is fully defined when the two nodes it is connected to are defined (node 1 and node 2 in this case). The links between primitives and nodes are implemented through a connectivity matrix.



```
>> Primitives{1}:

    Name: 'Beam3d'
    MaxInstances: 1
    MinInstances: 1
    Parameters: WIDTH
    ParamLimits: [width_max,
                  width_min]
    Data: [ 2-nodes primitive ]
```

```
>> Node 1, Node 2:

    Name: 'Floating'
    Props: [x, y]
    CanAdd: 0
    CanRemove: 0
    PropsChangeable: [1 1]
    PropLimits: [xmin, xmax,
                 ymin, ymax]
    MinConnects: 1
    MaxConnects: 1
```

Figure 3.3: Implementation of a beam primitive in Matlab. Design features for the primitive are

1) *Name*
2) *MaxInstances: max number of instantiations of this primitive that it is possible to use*
3) *MinInstances: min number of instantiations of this primitive that it is possible to use*
4) *Width*
5) *ParamLimits: max and min possible values for width*
6) *Data: number of nodes of the primitive*

Each primitive is implemented as a stand-alone entity or module, so that the system remains flexible and new parts can be added, if necessary. Any new primitive can be implemented independently and "plugged" into the code. Primitives do not need to be all of the same nature to be implemented. This offers the possibility for creating a library of components to be used according to the design domain explored and offering the possibility to generate multidisciplinary designs, for example mechatronic devices.

In Chapter 6 it will be shown how the library of primitives can be extended to include many more building blocks than the one used here as an example. Primitives can also possess different properties, as it will be shown in section 5.2.3 with the introduction of the mass primitive.

3.2.2 Systems

A system is represented as a graph that consists of nodes, primitives and subsystems. Figure 3.4 shows an example of these graphs, where each symbol represents a primitive. Individual primitives within a connected-node system, such as the ones in Figure 3.4, are each instantiations of a specific primitive type. These representations are symbolic, but in the examples described in this work the primitives are perfect images of the real parts. Each primitive has a certain number of ports to connect it to the rest of the system. Each port is connected at a node point. Nodes that are not connected to anything represent anchors, i.e. they are connected to the ground and represent a rigid constraint for the system.

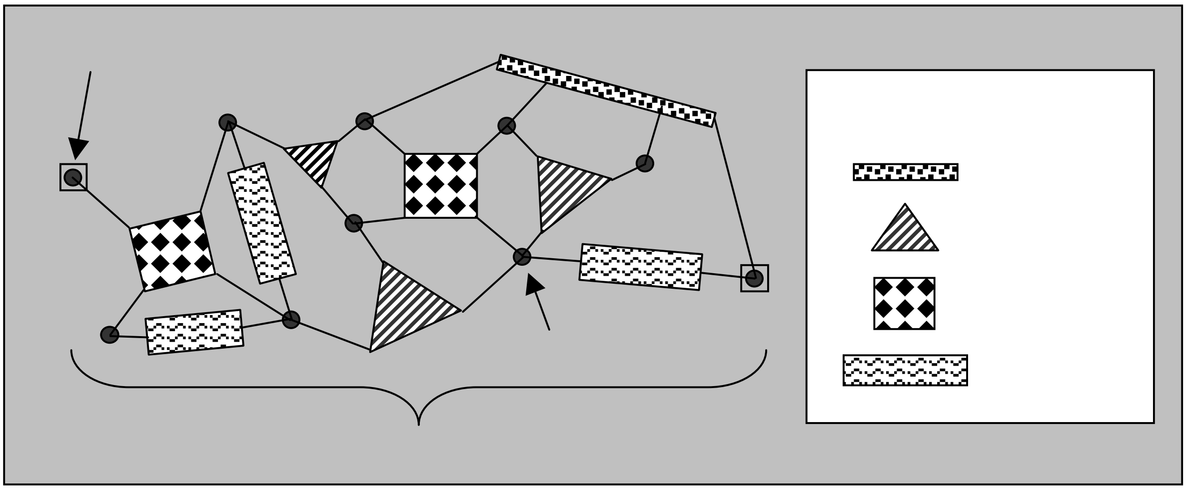
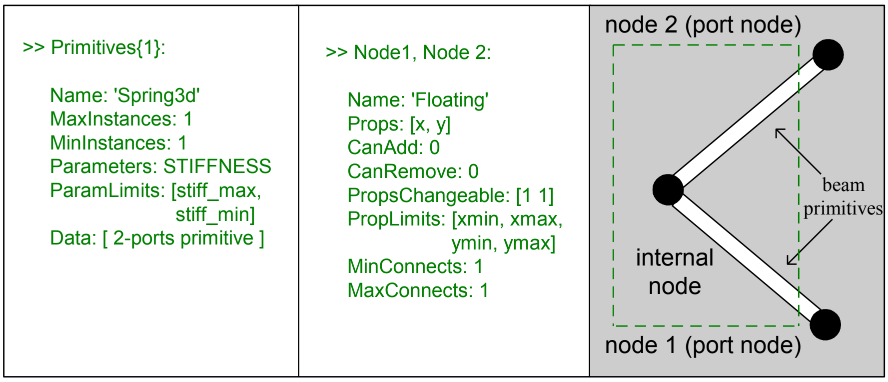


Figure 3.4: Example of a connected-node system with polygonal primitives.

3.2.3 Subsystems

Within the representation method, primitives and subsystem types are considered as different classes. A subsystem can be considered as a connected-node system with a set of ‘port nodes’ that connect the subsystem to the larger system and allow it to interact with the complete interconnected system as if it was a simple primitive. The requirements for subsystems are exactly the same as for primitives, except more complex, and perhaps even incorporating internal iterative design loops. Figure 3.5 shows a simple example of a spring subsystem and its Matlab implementation. The subsystem is a small connected-node system itself, with two beam primitives connected at an internal node. As a subsystem, however, the port nodes are nodes 1 and 2, and they are all that is ‘seen’ by the connected-node system using the spring subsystem. Parameters for the subsystem primitive would be, in this case, design criteria such as mass or stiffness. The spring subsystem would be fully defined by its port node locations (i.e. node 1 and 2) and a stiffness parameter. The effect of representing such subsystems is to remove from the global search part of the details of the connected-node system included in the subsystem. In a representation method, encapsulation of design parameters and variables is an important feature that allows the synthesis process to easily scale up. Also, CNS representations are freely hierarchical, giving the possibility of system decomposition in different ways. In Chapter 6 new ‘combined primitives’ will be introduced as a special instantiation of subsystems.



```
>> Primitives{1}:

    Name: 'Spring3d'
    MaxInstances: 1
    MinInstances: 1
    Parameters: STIFFNESS
    ParamLimits: [stiff_max,
                  stiff_min]
    Data: [ 2-ports primitive ]
```

```
>> Node1, Node 2:

    Name: 'Floating'
    Props: [x, y]
    CanAdd: 0
    CanRemove: 0
    PropsChangeable: [1 1]
    PropLimits: [xmin, xmax,
                 ymin, ymax]
    MinConnects: 1
    MaxConnects: 1
```

Figure 3.5: Implementation of a ‘spring’ subsystem consisting of two beam primitives
Design features for the primitive are:

1. *Name*
2. *MaxInstances: max number of instantiations of this primitive that it is possible to use*
3. *MinInstances: min number of instantiations of this primitive that it is possible to use*
4. *Stiffness of the spring*
5. *ParamLimits: max and min possible values for the stiffness*
6. *Data: number of port-nodes of the primitive*

3.3 Generation: the Modification Module

For the purposes of automatically synthesising a wide range of connected-node systems that represent feasible design alternatives, it is necessary to modify CNS representations in a robust, consistent and purposeful way. The Modifier Module encompasses a set of modification operators that are applicable to all connected-node systems, subject to user-defined constraints provided in the CNS definition. The modification operators guarantee that when a CNS is modified, the result will always be a valid CNS.

The modification operators reflect valid modifications of a spatial graph. Since the modification operators are domain independent, the effort of producing modification operators for specific design synthesis tasks is removed. Five general modification operators have been defined (but others can be added, according to needs and design domain):

1) The node property modification operator (Figure 3.6.I: the location of the node is modified)
2) The primitive parameter modification operator (Figure 3.6.II: the width of the beam is modified)
3) The primitive addition operator (Figure 3.6.III)
4) The connection swapping operator (Figure 3.6.IV)
5) The primitive removal operator (Figure 3.6.V).

While modifying a design, the Modifier Module also provides built-in ways to check for constraints, assuring that not only the design is feasible, but also that is valid. For every system generated the following types of checks can be made:

- Number of instantiations of a particular primitive not to exceed a specified limit
- Conflicts between primitives
- Correct node connectivity.

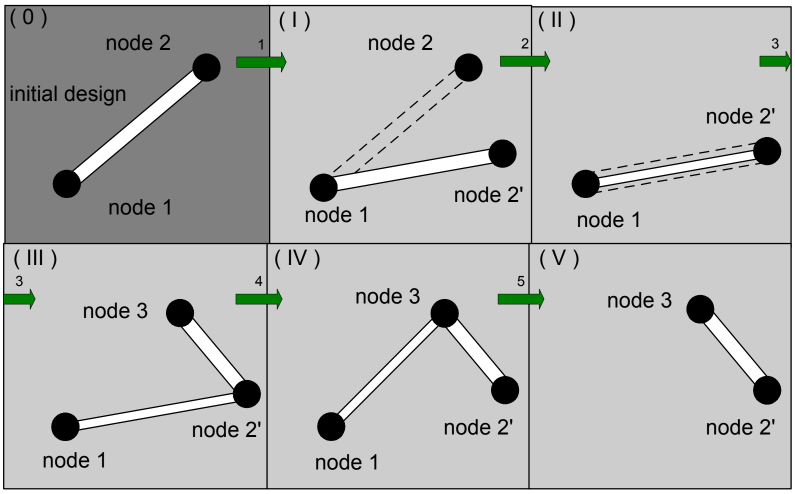


Figure 3.6: The five modification operators. Top row form left to right:
0) Initial design,
I) Modification of the position of node 2
II) Modification of the width of the primitive.
Bottom row form left to right:
III) Addition of a primitive,
IV) Swapping connection of a primitive,
V) Removal of a primitive.

3.4 Search Module: The Burst Algorithm

The search for solutions is driven by an iterative loop according to the design objectives and constraints defined in the optimisation model. Since the synthesis tasks under investigation have multiple objectives, a multicriteria generate-and-test search termed the 'Burst Algorithm' [Vale, 2002] has been adopted to test the synthesis method.

A flowchart of the search loop is shown in Figure 3.7. The algorithm relies on the modification module (seen in 3.3) and on an evaluation module (3.5). The Burst algorithm starts with the selection of a random design from the archive of solutions and chooses the modification operators to apply in short 'bursts'. The modification module is called and is expected to return a modified design variant of the input design. The evaluator is given the modified design and is expected to return the corresponding objectives for that specimen [Vale, 2002], but always accepting the valid modifications. The maximum length of each burst (BL) is a parameter set by the user, usually of the order of ten. Should any design that emerges be a new non-dominated solution to the problem, the design is archived. After each burst, a 'return-to-base' (RTB) is carried out by selecting a new starting design from the existing archive of non-dominated solutions for the next burst of design modifications. The algorithm terminates after a specified number of evaluations is executed. The CNS representation and the modification operators detailed in sections 3.2 and 3.3 are easily integrated in the loop. This flexible structure allows the search algorithm to be applicable to a wide variety of possible exploration scenarios. Burst can be seen as a derivative of MOSA algorithms with mandatory acceptance of solutions and a random return to base heuristic. In this sense it can be considered a naïve stochastic search algorithm that results in optimally directed solutions.

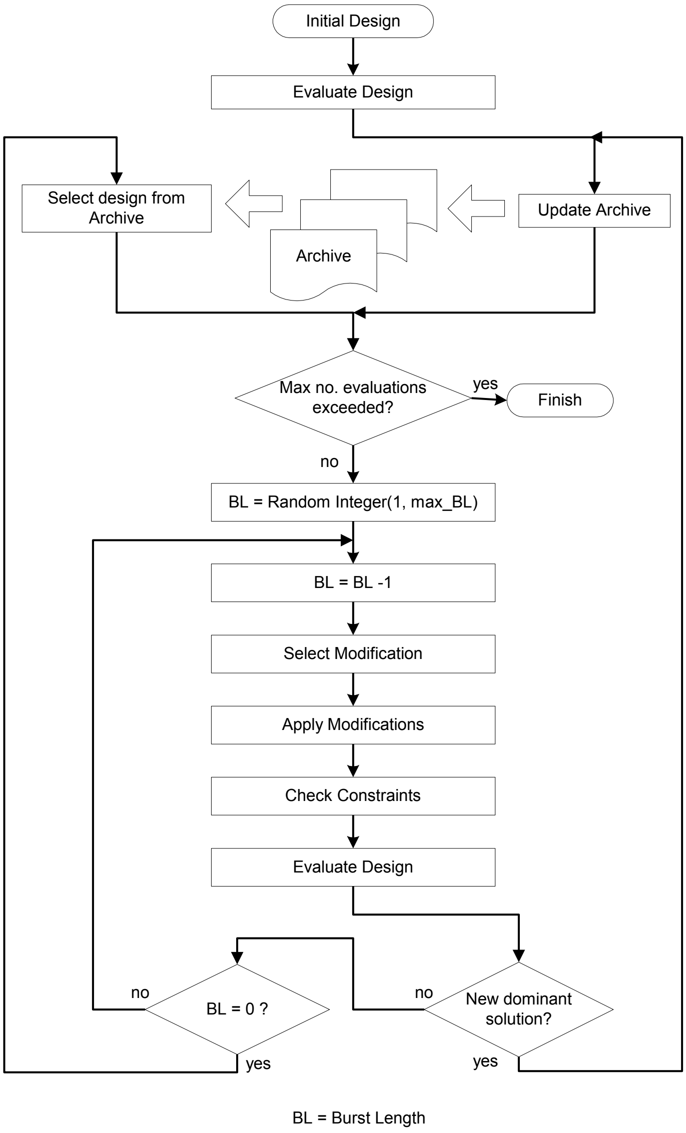


Figure 3.7: The Burst Search Algorithm.

The advantages of using stochastic optimisation methods in design synthesis problems like the one described here, has been explained in 2.1.3. However, any generate-and-test type of algorithm would be suitable for the synthesis method and easily ‘plugged’ into the synthesis loop like any other module. This is due to the fact that the method has been implemented in a way that a great variety of options can be integrated in terms of search heuristics, from using standard search methods to experimenting with new approaches.

The reason why Burst was chosen is the range of possibilities it offers for synthesis methods. The algorithm was created with the intent of aiding modification-based computational design synthesis tools, facilitating the search with the incorporation of machine learning (ML) techniques [Vale, 2002]. Learning from the ongoing development of the search, ML-based techniques define intelligent modification strategies to use in scenarios where the possible design modifications can be classified as a finite set. ML gives the possibility of a more directed and less wasteful use of resources during the search.

The reason for the naivety of Burst lays in the fact that the algorithm was implemented as a basis to incorporate ML in the search. The simple structure of the algorithm allows the method to improve, as well as scaling up for more difficult tasks. Burst in conjunction with ML has been tested with success in a variety of tasks, ranging from truss structures to bitmap problems [Vale, 2003d].

No use of ML techniques was added at any stage of this research, although the adoption of Burst and the modular structure of the method leave the possibility for future investigation of these aspects. The simple version of Burst presented here has been used in most of the synthesis tasks investigated, while an enhanced version of the algorithm has been used for topology optimisation tasks in Chapter 5. In the Burst algorithm described in Figure 3.7, the modification module would typically apply one of the associated modification rules at random. An enhanced version of Burst algorithm, called ‘Weighted Burst’, is able to assign a weight to modification rules, allowing them to be applied according to preferences.

A second reason for choosing a simple algorithm like Burst was dictated by the need for flexibility. The search algorithm is highly flexible and can be applied to both single and MO problems. The approach does not require problem-specific tuning, allowing to be applied to a wide range of different problems without modification [Vale, 2002].

Although it is difficult to review the performance of MO algorithms in terms of archive progression, Burst has been also successfully tested for convergence. This test was initially set up to compare different ML techniques and the improvement they could bring in speeding up the convergence of the archive. The testing of the archive's convergence is based on storing archives of solutions every few iterations and calculating hypervolumes of objective space that is dominated by the Pareto archive. Burst was tested for different design tasks and in all cases showed to converge reliably. Further description of the convergence study of the algorithm can be found in [Vale, 2002]. In the examples tackled in this work the study of the convergence was not investigated, as the Pareto fronts showed fast convergence to desired solutions. It is always possible though to apply these techniques to any design tasks, in order to determine the accurate number of iterations needed for the convergence of the algorithm in each case.

In the following chapters it will be shown how the Burst algorithm was able to produce good quality results, confirming the validity of this choice.

3.5 Evaluating the Connected-Node Systems: the Evaluation Module

The evaluation of connected-node system designs synthesised by the modification operators is carried out within the Evaluator Module. This is, by nature, a domain specific task, depending on the type of device being evaluated and the design optimisation model. Nevertheless, the Evaluator module is expected to adhere to a set of simple conventions. Its essential features include:

- Decoding the connected-node system and encoding it into a simulation model: the Evaluator Module is in charge of passing the design to a simulation module, i.e. a commercial software package able to provide quantitative feedback on the design behaviour
- Calling simulation to perform analysis of the behaviour of system, tipically done via FEA tools
- Reading analysis results returned from simulation
- Calculating all design performance metrics (both design objectives and soft constraints): within the Evaluator module the design specifications are defined in the form of objective functions to be minimised by the search algorithm. The definition of

the objectives would typically involve using the results returned from behavioural analysis.

The returned data for any design evaluation consists of a set of values for performance metrics, namely “soft” constraint metrics and design objectives. As mentioned in Chapter 2, “soft constraint” is used here to describe design constraints that are transformed into design objectives with penalty functions that are minimised to zero. In the implementation of the Evaluator, all design objectives are assumed to be minimised by the search algorithm.

In the case studies that will be introduced, the simulation is performed using either SUGAR, a software developed at Berkeley for MEMS analysis [SUGAR, 1998], or COMSOL Multiphysics [COMSOL, 2007], a more general Matlab-based multiphysics simulation tool. Either simulation modules can be used according to the requirements of the synthesis task and other simulation tools can be integrated as needed. In the following chapters it will be shown how the possibility to easily modify the design objectives according to the design task and to plug into the code different simulation software according to needs are the base of CNS-Burst’s flexibility.

3.6 Method Implementation

In 1.2 the description of a new generation of computational synthesis tools has been outlined. The present chapter will further analyse these features and will show how they have been taken into account in the implementation of CNS-Burst.

The development of a generative method’s architecture would ideally include higher-level functionalities that allow it to be extended and adapted to many different design tasks with the minimum effort. These characteristics, although not directly influencing the efficiency of the search method, have an impact on its capability to be used on a larger scale and be readapted to multiple tasks and design domains. From literature observation, it is clear that a number of difficulties prevent synthesis methods from exhibiting these characteristics and consequently being used in every-day design practice. Some of them are related with:

- The development of tools that do not include the possibility to be expanded in terms of functionalities (for example adding new parts or new objectives to the search). The majority of generative design methods are oriented at solving a single design task.
- The development of tools that do not include the possibility to be extended to multiple design domains and used in multi-disciplinary areas of design.
- The development of tools that are not easy to integrate with third party simulation/analysis/visualisation tools or any other tool needed in the design process.
- The development of tools with no API (application program interface), not easy to use, if not impossible to customise and adapt to designers' needs.

This chapter analyses these difficulties and highlights the directions taken on these issues in the development of CNS-Burst. In particular, two aspects of generative methods implementation are examined here:

1) The overall vision of generative tools' architecture. A possible approach to the development of a comprehensive tool will be examined, with particular attention to how the architecture of a method can influence general aspects such as its adaptability, reusability and flexibility.
2) The integration of generative methods with external simulation and analysis packages as well as any other tool needed during the search. This integration is a source of practical difficulties, if the adaptability and extendibility of the search method has been an aspect neglected during its implementation. More specifically, the integration of the COMSOL analysis package with CNS-Burst will be analysed.

3.6.1 Advantages of a Modular Implementation Paradigm

Some approaches to the development of generative design tools have been described in Chapter 2, each of them being implemented using a different paradigm. Either function-oriented, or task-oriented, or object-oriented, these paradigms are dictated by implementors' requirements and knowledge. If some paradigms may be advantageous in terms of implementation time (see for example task-oriented approaches), somehow they have represented an obstacle to the advance and commercialisation of synthesis methods.

Many successful generative design methods have been developed with the aim of covering a single design case in a specific design domain, making them not reusable for other design tasks. Koza [Koza, 2003] introduces the concept of 'routineness' as a way of assessing synthesis techniques. A synthesis technique has a high degree of routineness if it is applicable to a wide range of problems, within a single domain, with minimal human effort required to adapt the method and implementation to new design scenarios. Most of the approaches mentioned in Chapter 2 have been demonstrated to be effective in specific design tasks, but cases of techniques oriented at multiple design tasks are rare. This lack of adaptability has two important consequences in the development of a synthesis method:

1) The first one is the loss of knowledge in the development of computational synthesis methods for the design domain examined. A task-oriented approach leads to the solution of a single problem, and the method developed for this aim will probably not be used for any other design task or extended further, reducing the benefit of the research effort to implement it.
2) The consequent problem is the waste of time spent developing techniques that will not have future applications, resulting in a delay of commercial use of synthesis tools. Every time a new design task in the same design domain needs to be solved, extra resources will be spent in setting up a novel method to solve it.

Another typical tendency in synthesis methods development is the practice of covering a single design domain. The majority of computational synthesis methods implemented have been developed for a single design domain and examples of 'multi-domain' techniques, i.e. adaptable to different design domains, are rare. As engineering design requires knowledge of many disciplines, current limitations of computational synthesis methods often include the necessity of supporting multidisciplinary considerations as well as the integration of automated simulation modelling to create multiphysics simulation-driven synthesis tools. Straightforward adaptability of multi-domain synthesis methods creates instead the potential to provide customisable tools for automated design synthesis and optimisation.

In order to overcome the limitations set by the paradigms described above, an interesting approach in the development of a generative design tool - and the one followed in this work - is the possibility to focus the implementation on its reusability and extendibility. Developers of commercial design tools have already embraced this approach, considering the advantage that these two characteristics can bring to engineering design in terms of variety of applications and design tasks. This relatively new implementation paradigm, emerged in the mid-Nineties (Herbsleb and Moitra, 2001), offers agility in design by basing software development on the recombination of reusable components, as well as on the possibility to rapidly expand product variations. It also guarantees significant improvements in terms of shortening time to market and reducing development costs (Kotlarsky, 2007). This approach has not involved closely synthesis tools so far. It is useful at this stage to explore how the inclusion of this implementation practice could influence a widespread use and a faster commercialisation of synthesis tools.

In order to put these considerations into practice, CNS-Burst has been implemented following a modular approach rather than a functional one. Each of the fundamental components of the method (building blocks, evaluator, modifier) has been implemented as a stand-alone module, independent from the others. This allows for desired changes in each module without affecting the rest of the code, enabling also straightforward inclusion of new modules. The modules are linked together through the main algorithm that directs the search and calls the required different modules (Figure 3.2). Figure 3.8 shows a modular representation of the method where each module can be thought of as a black box with its defined inputs and outputs, completely independent from the other modules. The picture also highlights the cascade order in which each module is executed during the search. The method has been entirely implemented in Matlab, a numerical computing environment nowadays used also as programming language for science.

It is understandable how modularity represents the real strength of the method and is an important step in the development and adoption of generative tools in design. The choice of a modular paradigm enables desirable features for a synthesis method such as reusability and extendibility. New modules can be easily plugged-in to extend the method. The flexibility is due to the possibility to independently change, add or remove single modules.

It will be shown in the next chapters how this approach is advantageous for adapting the method to multiple tasks. Flexibility will be demonstrated in terms of task setting, search objectives definition and new primitives inclusion. Also, potential capabilities of the method to be extended to multiple design domains will be presented.

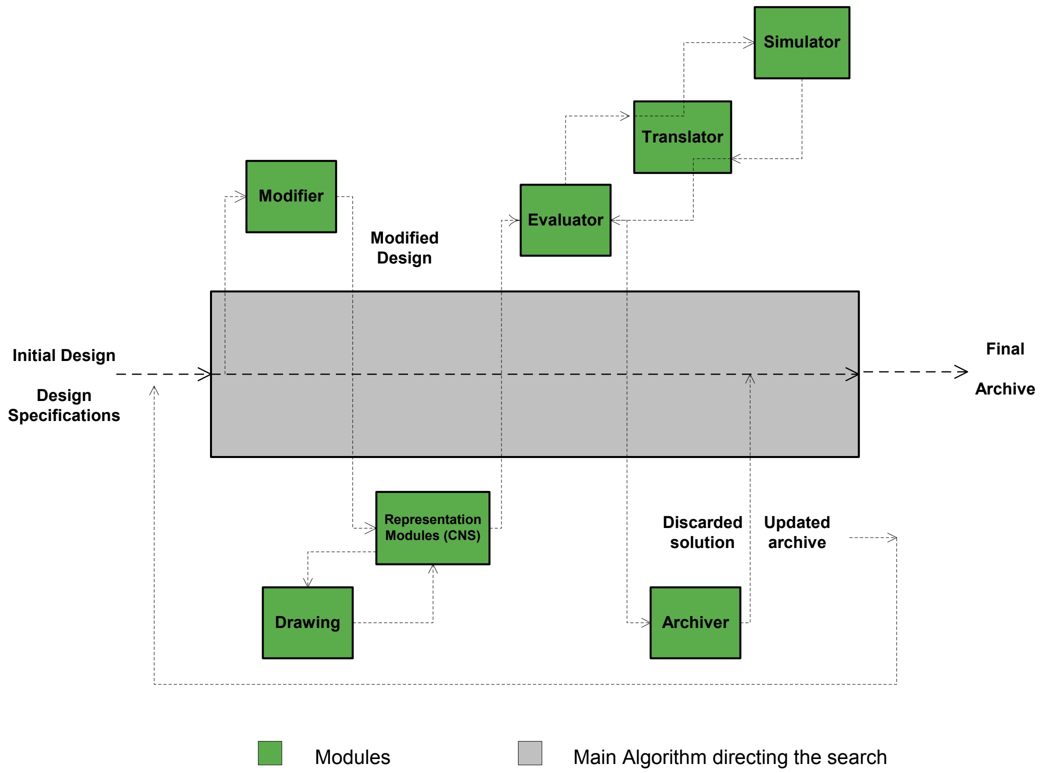


Figure 3.8: Modular structure of the method.

3.6.2 Integrated Simulation

Computational synthesis methods often find another obstacle to their development in the difficulty of integrating in the automated search external packages for the analysis/simulation of the solutions. As pointed out by Farouki [Farouki, 1999], the major difficulty in automating a CAE optimisation procedure, is the proper linking of the various stages of the design process (namely design generation, geometrical model and related simulation model creation) by means of a CAD tool, setting of properties and boundary conditions, numerical analysis and performance evaluation, and finally being able to optimise the shape, or other functional parameters of the system.

In order to find feasible and optimised designs, synthesis techniques are built on a search algorithm, integrated with an evaluation method. The generated designs are evaluated according to desired design performance criteria, stated as objectives of the search. The evaluation might require complex analysis to simulate design behaviour, which is often executed through external FEM software embedded in the search code. Analysis tools must be able to perform rapid simulation of the designs' behaviour and pass to the search code accurate feedback on the designs' performance. Also, the design simulation must be performed automatically after each design generation. These requirements often encounter limitations due to the impossibility of finding commercial packages that are able to perform simulation in the design domain analysed. The choice may often be limited to packages that are not optimal for the design performance being analysed or do not offer the level of accuracy needed. A second problem in integrating external tools is the difficulty of supporting multiple design criteria based on multidisciplinary considerations, hence the necessity of finding compatible multidisciplinary analysis packages. A third important problem is due to the impossibility of integrating analysis tools easily and time-efficiently.

All these reasons often make the evaluation of the solutions a neglected step, being considered too expensive to perform both in terms of time and effort. Synthesis methods are left sometimes incomplete: solutions are generated but not analysed and evaluated, leaving the creative effort of the search without validation.

The integration of analysis packages can be executed in different ways. Batch processing offers a direct link with the Operating System, creating the possibility for using shells to execute operations in pipeline. These scripts have the function to communicate output results of each operation to the following operations that will use them as an input. This is exactly the case with results returned from an analysis package and passed to the evaluation method using an external script. Scripting languages can also be used to provide the link between incompatible pieces of code [Ouststerhout, 1998]. These scripts are able to mediate between results generated using a code written in a certain language that must be used as input by a piece of code written in another language (or between two pieces of code operating on different Operating Systems). An example of application of this technique used to implement a generative tool for synthesis of mechanical systems can be found in [Starling, 2004].

Another effective approach is the direct plug-in of an external package into the search code. This is a far more complex and time-consuming method, and sometimes even impossible to apply. Again, writing modular synthesis methods instead of monolithic ones makes this approach easier to implement. In the next section it will be shown how the latest approach has been used to integrate external analysis in CNS-Burst.

3.6.2.1 CNS-Burst and COMSOL Integration

Complexity of MEMS devices, requiring accurate multiphysics simulation and post-processing, is an obstacle to the development of generative tools able to design them. The necessity of integrating specific analysis packages makes this task particularly difficult. The use of hand design coupled with hand analysis and in some cases with external optimisation packages is still an usual practice for MEMS designers.

This section discusses the choice made to automate simulation of MEMS in CNS-Burst and its motivation. The option chosen for this work has been command line integration of analysis software. In particular, the external packages used have been two. The initial benchmark problems described in Chapter 4 have been solved using SUGAR, a software for MEMS analysis developed at Berkeley and entirely implemented in Matlab. SUGAR is based on a Modified Nodal Analysis methodology, which can be easily integrated with the CNS representation system. Nevertheless its lack of precision and limited capabilities, added to the impossibility to perform multiphysics and non-linear analysis, make its use not ideal for more complex design tasks.

The necessity to perform accurate analysis in order to obtain solutions as ready as possible to be post-processed, brought the substitution of SUGAR with COMSOL. COMSOL is a general multiphysics FEM simulation package, able to perform both linear and non-linear analysis. In particular, the COMSOL toolbox used in this work is the MEMS toolbox for 3D eigenfrequency and static analysis. COMSOL is usually used from user interface but can also be used from API. COMSOL API is written in Matlab, which makes it straightforward to couple it with CNS-Burst.

The major difficulty in integrating this analysis package has been to translate the design generated by the search code - a geometrical model represented by CNS design representation - into a simulation model that could be evaluated by COMSOL. A CNS design is constituted by a set of primitives (Matlab objects) and matrices: the node position matrix and the connectivity matrix (setting connections between nodes and primitives).

The simulation model analysed by COMSOL is a COMSOL object (Figure 3.9). A second problem encountered in the integration process has been to find a way of automating the translation from a geometrical model to a simulation model at each design generation. These problems have been solved using COMSOL from API, i.e. not through user interface. The search method, also implemented in Matlab, is able to call COMSOL functions from the Matlab command line, once the entire path of COMSOL functions has been loaded into Matlab environment.

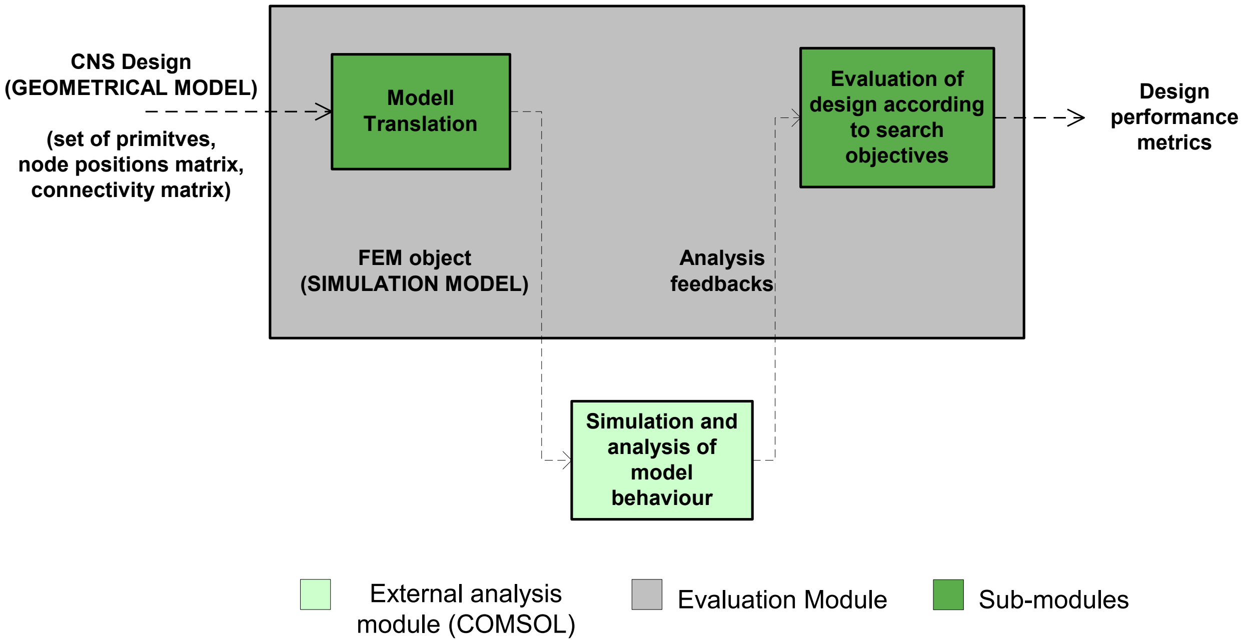


Figure 3.9: The Evaluator Module.

Once the analysis package has been integrated in the search method, the evaluation works as follows: the search algorithm creates new design solutions in the form of connected-node systems. In most of the examples tackled, only a quarter (or half) of the design geometry is generated, while the entire geometrical model is created automatically by symmetry. A translation routine transforms the connected-node system (geometrical model) given in input into a COMSOL object (simulation model) in output (Figure 3.9). The translation routine transforms each primitive of the connected-node system in a COMSOL geometry part, reading the nature of the primitive (beam, disk, etc) and transforming it in the correspondent COMSOL shape (e.g. block, cylinder, etc). The node position and connectivity matrix are transformed in coordinates and orientation angles for the correspondent COMSOL block. Loads and constraints defined in the nodes description are translated by the routine into COMSOL boundary, edge and point settings (load and constraints). The COMSOL object is then passed into COMSOL functions directly from Matlab command line for its behaviour to be analysed. The outputs of the COMSOL functions are read by the evaluation function of the search that uses them to calculate the desired objectives of the search for that specific design (Figure 3.9). The entire process, completely automated from the creation of the novel design to the output of the design objectives, will be repeated each time a new design is generated. The strength of the method is reinforced by the possibility to complete an entire design process with accuracy, as the results in Chapter 5 will show.

On a final note, it has to be highlighted how the possibility to include any appropriate analysis package for a specific design task is again due to the modular and flexible structure of CNS-Burst.

3.7 Conclusions

The method introduced is an object-oriented, simulation-based, multicriteria computational synthesis technique, not formulated for a specific design domain. Necessary components of this synthesis tool comprise a design representation method, a generative mechanism for the generation of alternative designs, integrated evaluation of performance criteria, and a search method to find feasible and optimised design alternatives.

This chapter has also explained how several ideal characteristics for a generative tool, such as adaptability and reusability, can be implemented in practice. The fundamental move towards these characteristics have been considered in the architecture of CNS-Burst, which is based on a modular programming paradigm. Each of the fundamental components of the method has been implemented as a stand-alone module, independent from the others. This allows for desired changes in each module without affecting the rest of the code, enabling also straightforward inclusion of new domain knowledge and desirable analysis and simulation external packages.

The examples tackled in Chapter 4 and 5 will be a validation of the implementation choices described so far. The case studies presented in these chapters will also be helpful in exploring further issues related to generality and lack of reliability of synthesis methods. One problem concerns the effectiveness of synthesis techniques in complex design domains, where it is essential to include sufficient details in both synthesis and optimisation models in order to generate practical results. The second problem deals with the perplexity of whether it is possible to obtain detailed and accurate solutions to design tasks with complex constraints and objectives. The examples will also demonstrate that computational synthesis methods with extended capabilities create potential for application in everyday design practice, leading in the future to extended use of such tools in industry.

CHAPTER 4: BENCHMARKING THE METHOD

This chapter reports results obtained against benchmark problems in automated synthesis of MEMS. Benchmarking the method has been a necessary step in order to prove its potentials. The results obtained in this process also set the basis for more complex synthesis tasks.

Benchmarking the method consisted of three steps: 1) comparing its efficiency against similar existing methods for automated synthesis of MEMS, 2) evaluating whether it could reach at least the same level of satisfactory results for some specific tasks, 3) and identifying unique benefits compared to existing published methods. The examples tackled do not introduce new tasks, but reproduce ones that have already been used in literature. These are simplistic cases that often do not pursue practical design objectives. As well as comparing the method with other synthesis techniques, the goals of these experiments were also to illustrate the basic capabilities and extendibility of the method, investigating its potentials for topology optimisation tasks and proving its adaptability to different case studies.

Two different case studies have been investigated. The first example on micro-compliant mechanisms has been chosen because its structural topology is particularly suitable for CNS representation and for the application of the method. Also, due to their complexity, compliant mechanisms are usually designed by hand, representing an ideal case for automation. The second example investigated here are microresonators. The case is also an introduction to the work of the following chapters, where resonators will be designed under more stringent and practical specifications.

These simple case studies perfectly illustrate the aim of the method to automatically generate spaces of alternative designs that trade-off multiple design objectives. The solutions found, although topologically innovative, are highly conceptual and far from being ready for manufacturing, as they omit several important behavioural aspects of MEMS. Both examples though, show how the flexibility and generality of the method is not an obstacle to finding novel solutions.

4.1 Automated Synthesis of Micro-Compliant Mechanisms

As seen in Chapter 2, compliant mechanisms are one of the most common case studies investigated using automated synthesis methods. This is due in part to the complexity of their topologies, but also to the fact that their design covers exclusively the mechanical domain, differently from the multi-domain nature of most MEMS.

A complete background of computational synthesis works on compliant mechanisms has been given in 2.3.2. The following sections will explore exclusively a research work on a particular case of compliant mechanism that has been used as a benchmark against CNS-Burst.

4.1.1 Use of Compliant Mechanisms in MEMS

Compliant mechanisms, exactly like rigid-body mechanisms, are used to transfer or transform motion, force or energy, but do not consist of rigid links connected at movable joints, and rather gain their mobility from the deflection of flexible members. These mechanisms offer great promise in providing new and better solutions to many mechanical design problems. Their application can be considered for a variety of reasons. More specifically, the advantages of substituting rigid-body mechanisms with compliant mechanisms can be classified into two categories [Howell, 2001]:

1) Lower costs, mainly due to the reduction of the total number of parts required to accomplish a specific task and also to the simplified manufacturing process;
2) Increased performance (increased precision, increased reliability, reduced wear, reduced weight and reduced maintenance).

For the aforementioned reasons, compliant mechanisms are also widely applied in the field of microelectromechanical systems, and are often coupled with actuators. Common methods for their fabrication at microscale use planar layers of material, such as polycrystalline silicon or polysilicon. The constraints introduced by the planar nature of MEMS fabrication and their scale, makes assembly of parts difficult, and offers a number of challenges in the construction of mechanical micro-devices. According to Howell, these fabrication issues make the employment of compliant mechanisms in MEMS even more convenient. Other advantages at the micro level are that they:

1) Can be fabricated in a plane,
2) Are monolithic and do not require assembly,
3) Require less space and are less complex,
4) Have reduced friction,
5) Have less clearance due to pin joints, resulting in higher precision.

Unfortunately, though they offer a number of advantages, compliant mechanisms present also several difficulties related to their design, requiring knowledge of mechanism analysis and deflection of flexible members. As the complex and unpredictable elastic behaviour of such devices makes their design complicated, many compliant mechanisms are still designed by time-consuming trial and error methods, which are mainly applied to very simple systems. All of these difficulties result in a strong need for good automated synthesis methods in order to improve and facilitate their design. Among all the techniques employed, stochastic synthesis algorithms have not found yet a successful application in this field, even though the modularity of mechanisms and the easy mathematical formulation of many design objectives make compliant mechanisms a suitable and challenging field to explore with such methods.

4.1.2 Description of the Benchmark Problem: Synthesis of a Force Inverter and Displacement Amplifier through Continuous Topology Optimisation

The first case study used here as a benchmark is based on the results obtained by Sigmund [Sigmund, 1997] in automating the synthesis of compliant devices. As mentioned in 2.3.2, Sigmund makes use of continuous topology optimisation. Homogenisation techniques have proved to be a successful approach to synthesis of micro-devices, providing a sound benchmark to demonstrate the capabilities of the method introduced. This section is a detailed review of Sigmund's work on compliant mechanisms. Final conclusions and results from the use of continuous topology optimisation for the synthesis of these devices are also reported.

The devices analysed by Sigmund are micro-displacement force inverters and amplifiers. Force inverters are structures where a force applied in input is converted in an output force directed in the opposite direction. Another important issue in MEMS is the lack of actuators with high output displacement and, at the same time, high output force. This obstacle may be overcome coupling actuators with efficient displacement amplifiers. Such devices can convert the output displacement of a large-force, small-displacement actuator into a larger output displacement with minimum loss of force [Sigmund, 1997]. This section explains these cases and gives an idea of how automated synthesis can be beneficial in this design task.

Sigmund's research starts with the investigation of a simple force inverter [Sigmund, 1997]. The goal here is to design a structure, limited in a design domain of $300x300\ \mu m^2$ (Figure 4.1), that is able to convert an input force (F_{in}) applied to the left edge of the design domain into an output force at the right edge (F_{out}). F_{in} and F_{out} must be directed in opposite ways. Input and output points where the forces are exerted are aligned. The application of this input force results in a displacement of the input point (Δ_{in}) and in a consequent displacement of the output point (Δ_{out}). Hence, the first design specification of the design task is to have a $180°$ angle phase between Δ_{in} and Δ_{out} (being input and output displacement on the same axis). The displacement of the structure in input is also constrained. The force inverter synthesised in this task is shown in Figure 4.2.

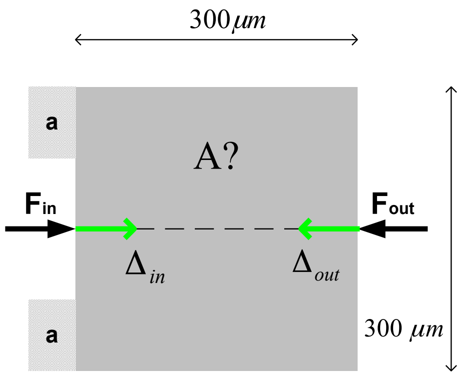


Figure 4.1: Design domain for force inverter
A: design area;
a: anchor points;
F_{in} : input force;
F_{out} : output force;
Δ_{in} : input displacement;
Δ_{out}: output displacement.

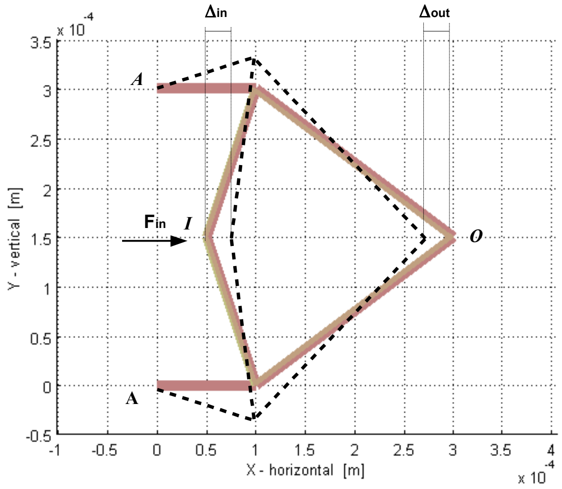


Figure 4.2: Force inverter synthesised by Sigmund through topology optimisation technique (initial position). In dotted line the displaced position. Δ_{in} and Δ_{out} are also highlighted. 'I' is the input point (node where the force is applied), 'O' is the output point, A' are the anchor points.

Sigmund extends further his work to the synthesis of a displacement amplifier [Sigmund, 1997; Pedersen 2001]. In addition to the aforementioned characteristics of being inverters, these devices are also able to magnify the output displacement for a given input. The optimal design obtained in the previous experiment (Figure 4.2) has the following characteristics:

$$\Delta_{in} = 5\ \mu m \qquad (4.1)$$

$$\Delta_{out} = 4{,}8\ \mu m$$

$$GA = \frac{\Delta_{out}}{\Delta_{in}} = 0.96$$

In this example the displacement in input is not amplified in output by the mechanism (GA=0.96). It might be the case though, that designers are interested in having an amplification of the input displacement (GA>1). An inverter having a GA>1 is also called an amplifier. According to Sigmund, one of the most important objectives in compliant mechanisms synthesis is to be able to control the ratios between output and input displacements or output and input forces, which are described by the geometrical advantage (GA). Unfortunately, the complex and unpredictable elastic behaviour of compliant mechanisms, coupled with the demand of a specific performance ratio, complicates the design of inverters. It is therefore understandable how synthesis techniques, like the one used here, can facilitate design tasks of this kind by setting these ratios as objectives of the search.

This second experiment consists then in synthesising designs for a maximum output displacement. Here the design specifications are again a *180°* angle phase between Δ_{in} and Δ_{out} and an input displacement constrained to be less than $5\mu m$. The additional specification for the inverter to be also an amplifier is given by the requirement of having Δ_{out} bigger than Δ_{in}. For this example the force applied in input is $F_{in} = 1000\ \mu N$.

The optimal design resulting from this search has been obtained using linear displacement analysis theory. However, in the conclusions of his article, the author clarifies that the assumption of linear displacement is violated at the micro-scale, leading to inaccurate results, and mentions the necessity for further exploration of the case study with a non-linear optimisation approach. This suggested investigation is then tackled by Pedersen and

Sigmund [Pedersen, 2001]. Using the same example of force-inverter/displacement amplifier examined by Sigmund (1997), the device is first synthesised through linear and then through non-linear displacement analysis theory. The optimised solutions are shown in Figure 4.3a and 4.3b respectively and are both able to amplify the input displacement about ten times in output. These solutions have completely different layouts. Comparing the performance of designs obtained through the two different kinds of analysis, Pedersen's research shows how mechanisms optimised using non-linear displacement analysis are more accurate than those optimised using linear displacement.

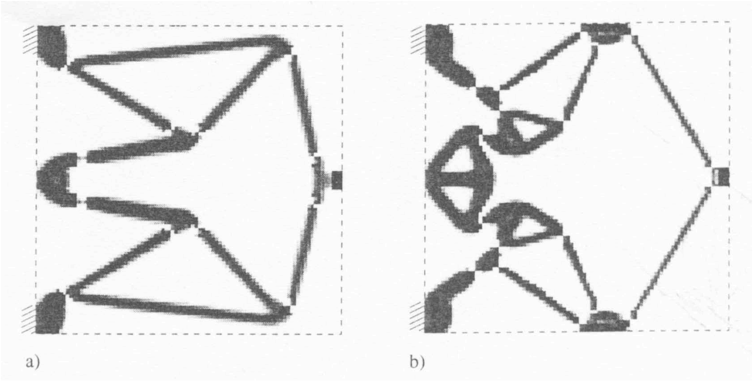


Figure 4.3: Topology synthesised force-inverter mechanisms:a) Optimised using linear finite element analysis, b) Optimised using non-linear finite element analysis [Pedersen, 2001].

From Sigmund's research work, two important facts related to the use of continuous topology optimisation in compliant mechanisms synthesis emerge. The first observation is that the solutions obtained through this method have 'lumped' designs. In the literature, compliant mechanisms have been categorised as 'distributed' or 'lumped'. Subject to an input force, distributed compliant mechanisms bend throughout the structure, whereas lumped compliant mechanisms bend in flexural hinges (Figure 4.4). Sigmund shows that both distributed and lumped compliant mechanisms can be optimal, depending on the task considered [Sigmund, 1997].

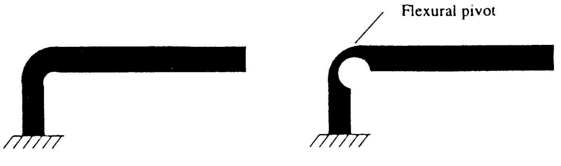


Figure 4.4: A distributed compliant mechanism (left) and a lumped compliant mechanism (right). Figure from [Sigmund, 1997].

The second consideration is related to an important issue when using continuous topology optimisation. The solutions obtained, in fact, are always lumped designs and usually need to be submitted to an extensive post-processing. This mainly involves the definition of flexural hinges contours around the irregular geometry that emerges from the optimisation procedure. Figure 4.5a shows an example of an optimised inverter topology obtained by applying the method and its irregular profile. Figure 4.5b shows the same design solution after post-processing.

The amplifier example is the benchmark problem chosen for the present research and will be discussed further in the next section.

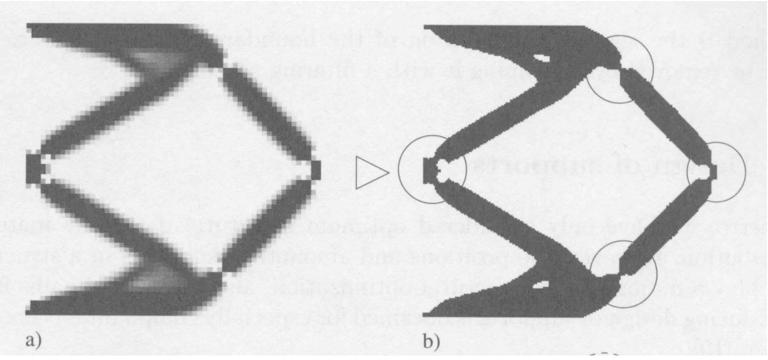


Figure 4.5: Post-processing in topology optimisation: a) optimal inverter topology, b) same topology after post-processing (from [Bensdøe, 2003]).

4.1.3 Solution of the Task Using CNS-Burst Method

Sigmund's work on the displacement amplifier will be used as a benchmark. In this experiment the CNS-Burst method, which involves a discrete representation, is compared with the continuum-type topology optimisation technique used by Sigmund. The synthesis task has been described in the previous section and will be formulated here as an optimisation task. For practical reasons, the type of analysis used in this synthesis tasks is linear. Although, in the concluding remarks of their work, Sigmund and Pedersen [Pedersen, 2001] underline the importance of using non-linear analysis theory for the synthesis of compliant mechanisms, results will be compared with the design showed in

Figure 4.3a. The input displacement for this design is $\Delta_{in} = 5\ \mu m$ and the output displacement, calculated through linear analysis, is $\Delta_{out} = 48.6\ \mu m$.

The reason behind this choice is determined by the initial use of SUGAR [SUGAR, 1998] for the automatic simulation and analysis of the results presented. The SUGAR analysis package integrated in the search code at this stage is limited to linear analysis. The choice of this analysis package was driven initially by the simplicity of integrating it in the search code for the benchmark studies. The SUGAR version adopted here uses a derivation of the linear Euler-Bernoulli beam model [Reddy, 1995]. In this model, the axial and transverse deflections are decoupled (which is part of the reason why this is appropriate only for small deflections). The only place the model deviates from the standard description given by Reddy is in the inclusion of a damping term from the fluid film between the beam and the substrate [SUGAR Implementor's guide, 2002].

Although SUGAR has been the analysis software of choice used for this benchmark, it has to be reminded that the search generates solutions independently from the analysis software, which is called at the evaluation level and after the design has been synthesised. The external analysis package does not influence the search capabilities of the method, which can potentially be integrated with any evaluation packages executing linear but also non-linear analysis, as it will be shown in Chapter 5.

The Initial Design

The first step of the present investigation has been to reproduce the optimal device synthesised by Sigmund. The device has been used then as a starting point to explore the possibility of synthesising amplifiers with a higher level of performance than the one obtained by Sigmund. Initially it has been fundamental to create an initial design with performance close to Sigmund's optimal solution, in order to validate whether the method could provide better results than this starting point. The initial design has been modelled to be similar to the lumped structure of the amplifier proposed by Sigmund. In order to obtain the same flexible structure, micro-joint beams have been used. A first version of the initial design model is shown in Figure 4.6.

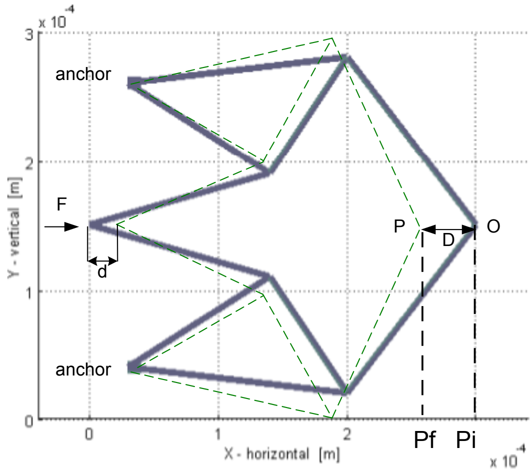


Figure 4.6:Displacement amplifier: initial design. Solid line: initial position. Dotted line: finalposition. 'F': force applied at the input point. 'O': output point, where the output force is released. 'd': input displacement. 'D': output displacement.

In this model, the size of the design domain is $300\times300\ \mu m^2$. An input force of *1mN* is applied at the centre of the left edge (Figure 4.1). The inverter is built from Polysilicon (Young's Modulus *160GPa*). Finally this initial design exhibits geometry and characteristics similar to those of Sigmund's amplifier shown in Figure 4.3a ($\Delta_{in} = 5\ \mu m$, $\Delta_{out} = 48.6\ \mu m$).

Formulation of the Optimisation task and Performance Evaluation

Design specifications for the search were to obtain:

- A *180*-degree phase angle between the input and the output displacements (input and output displacement on the same axis but in different directions). This is the constraint specification that guarantees that the device is an inverter. This result was obtained introducing axial symmetry of designs to guarantee perfect horizontal displacement of the output node without vertical displacement. Designs were synthesised using half symmetry. To address this issue, the first step considered was to synthesise one half of each design alternative, minimising both the vertical displacement of the output and the input node. The other half of the design was obtained by mirroring the design across the horizontal symmetry line.
- An input displacement $\Delta_{in} \leq 5\ \mu m$. This was modelled by constraining the input displacement with a soft constraint (described in 2.2.4). $|\Delta_{in} - 5\mu m|$ is the error in input

displacement (the difference between the actual input displacement and the desired one). This objective was introduced because the simple minimisation of the input displacement often led to solutions with minimum Δ_{in} possible. The solutions were non-dominated by any other design of the archive, although they did not present competitive output displacement.

- The maximum output displacement possible. This was modelled initially by minimising $1/GA$ (inverse of the geometric advantage), but the search did not pursue this minimisation in parallel with the maximisation of the output displacement. A small value of $1/GA$ can in fact also not correspond to small values of the input displacement ($\Delta_{in} \leq 5\ \mu m$). Since the constraint on input displacements is a soft constraint, the resulting designs exhibited unwanted input and output displacements (for example both too low), although presenting a high GA. The choice was then to minimise the difference between the initial position of the output node ($P_i = 300\mu m$) and its displacement yielding the final position after the displacement (P_f, see Figure 4.6):

$$P_f = |P_i - \Delta_{out}| = |300 - \Delta y_{out}| \quad [\mu m] \tag{4.2}$$

This objective, again obtained through a soft constraint, finally guaranteed the expected results. Since Sigmund's device has an output displacement of $48.6\mu m$ ($P_f = 252\mu m$), designs with $\Delta_{out} \geq 48.6\mu m$ ($P_f \leq 252\mu m$) are considered solutions with good performance.

Considering the aforementioned specifications, the optimisation model for the design task was formulated as follows:

$$\begin{gathered} \min P_f, |\Delta_{in} - 5\mu m| \\ S.t. 0 \leq x_j \leq 300\mu m, 0 \leq y_j \leq 300\mu m, \Delta_{in} \leq 5\mu m \\ \textit{Design Constraints (see Table IV.I)} \end{gathered} \tag{4.3}$$

Where j is the number of nodes, (x_j, y_j) are the coordinates of the nodes. The design constraints for the beams are listed in Table IV.1. Design variables considered for this task synthesis task were:

- the number of beam primitives: i
- length and width of the beams: l_i, w_i
- the connectivity of the beams (defining the design topology)
- the geometric position of all the nodes (x_j, y_j)

Table IV.I: Design constraints for the beam primitive.

Specific to primitives	Conflicts with other Primitives	Node connectivity
▪ Maximum length = 300 μm ▪ Maximum width = 20 μm ▪ Minimum length = 5 μm ▪ Minimum width = 5 μm ▪ Thickness = 5 μm ▪ Number of instances N≤5	▪ No overlap allowed ▪ Minimum separations between any two of non-connected beams = 50μm ▪ Angle between the beams θ > 30°	▪ Can share normal (floating) node with maximum five other beams ▪ If connected to anchor type node, cannot share that node with any other primitive.

Results

The algorithm was run five times at *12,000* iterations each. Figure 4.7 shows each archive from five optimisation runs overlaid on the same axes. The three coordinates of the plot represent the objectives of the search: final position of the output node, input displacement, and error in the input displacement. The trend of convergence to the constraint value (*5μm*) for the input displacement and the effectiveness of soft constraints are shown in Figure 4.8, that represents the *XY* view of the same archive. The plots give an impression of the convergence of the solutions for an input displacement of *5μm* or less (highlighted by the darkening trend of the solutions). The lower right corner of the plot encloses the number of solutions with correct input displacement in the overall archive: *64 %* of the solutions have $\Delta_{in} \leq 5\ \mu m$. Figure 4.9 (*XZ* view of the plot) shows how many solutions have reached the targets and can be considered high-performing solutions: the highlighted square limits all the solutions of the overall archive that met the modelled constraints (*5.2 %* of the solutions). As mentioned in 3.4, the Burst algorithm have been tested a length for convergence through the search [Vale, 2002], although this has not been necessary for this simple example and in any other presented in this chapter. The Pareto archive resulted unchanged immediately after the first iterations, meaning that the algorithm had converged rather quickly to the final solutions and before the maximum number of iterations set was reached. Moreover, the designs observed after few iterations demonstrated to have better performance than the benchmark solution, confirming the efficiency of the method.

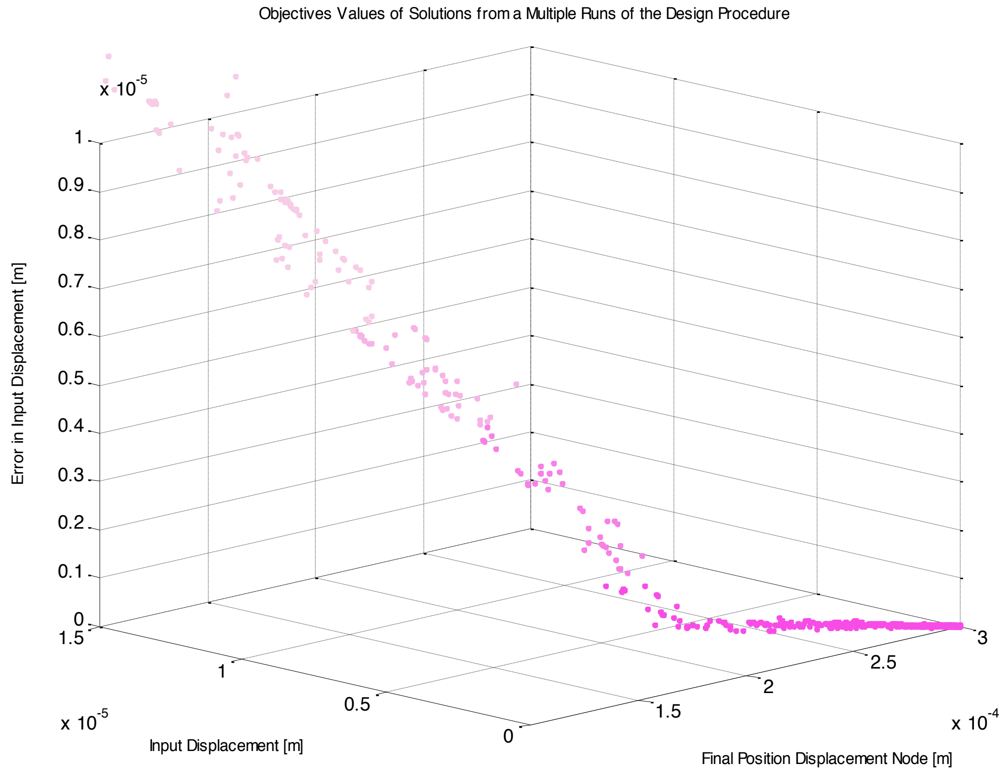


Figure 4.7: Archive of solutions that resulted from 5 optimisation runs at 12000 iterations.

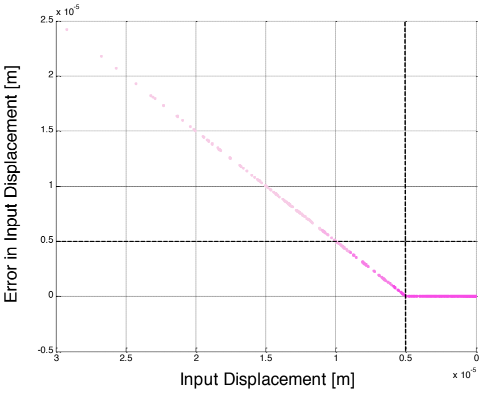


Figure 4.8: YZ view of the archive in Figure 4.7

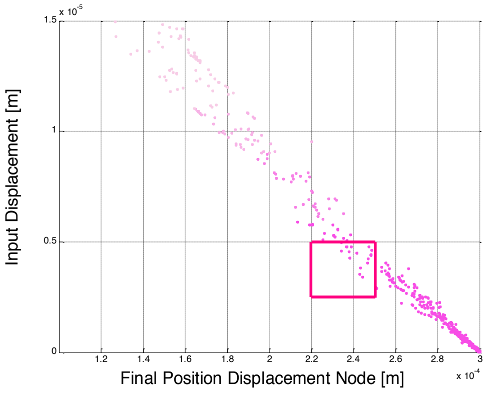


Figure 4.9: XY view of the archive in Figure 4.7.

Figures 4.10-4.11 show some examples of these valid solutions. These solutions are scaled, meaning that the thickness of beams in the image does not correspond to their real thickness. Values for the input and output displacement are listed, as well as the geometric advantage *GA*. All these designs have better performance than Sigmund's optimal design ($\Delta_{in} = 5\ \mu m$, $\Delta_{out} = 48.6\mu m$, $GA = 9.6$). Another important result is that both designs do not present a lumped configuration. The lumped structures obtained by Sigmund always represented a limitation of the method, which only a recent research on manufacturing constraints has solved [Hedge, 2008]. Although starting from an initial design formed by micro-joint beams, with characteristics similar to the lumped structure synthesised by Sigmund, the solutions found have a distributed compliance structure. Through modifications of the initial design, most of the synthesised solutions evolved into distributed structures that rivalled the benchmark solution.

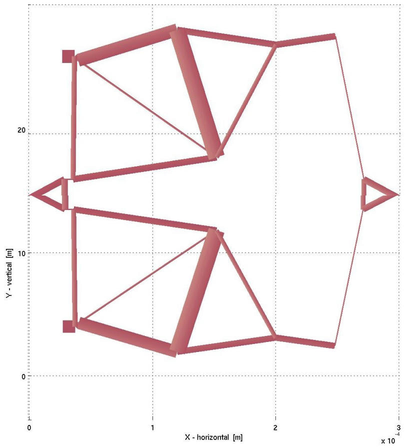


Figure 4.10: Design solution: Δin= 5μm, Δout= 57μm

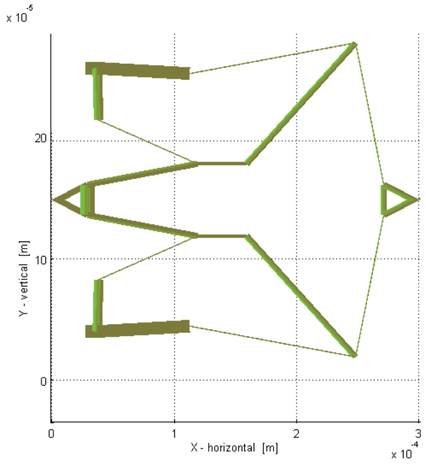


Figure 4.11: Design solution: Δin= 5μm, Δout= 75μm

4.2 Synthesis of Meandering Resonators

The second benchmark problem considered here is a meandering resonator. This type of MEMS device was chosen for its modular structure, that is particularly suitable to be synthesised through CNS-Burst method, and also for the many design objectives covering multiple design domains that can be optimised. This is the same example tackled in the research on automatic synthesis of MEMS conducted by Zhou and Kamalian [Zhou, 2001, 2002; Kamalian, 2002, 2004] and mentioned in Chapter 2.

In that work, starting from a high-level description of the device's desired behaviour, both topology and sizing were generated by the combination of parameterised basic MEMS building blocks, using MOGAs (Multi-objective Genetic Algorithms) to produce a Pareto-optimal set of feasible designs. The general procedure of the stochastic technique just mentioned, not so different in fact from the method introduced in the present work, makes this case study the ideal benchmark. The possibility to benchmark results obtained for the same synthesis problem using a different stochastic synthesis technique is advantageous because it provides a measure of the capabilities and efficiency of the CNS-Burst method.

Again, this is a simple case study that shows the potentials of the method in topology optimisation problems and in problems making use of spatial constraints, typical of MEMS design. Also, it represents an introductory study for the synthesis of more complex problems on microresonators like the one that will be presented in Chapter 5.

4.2.1 Use of Resonators in MEMS

A resonator is a device with a vibratory natural response that, when properly excited, resonates at a certain frequency or over a range of frequencies. The device is actuated by the application of a force electrically generated. The structure resonates when the force reaches a certain frequency. A resonator can be considered a second order mass-spring-damper mechanical system consisting of a central resonant mass suspended by spring structures. The central resonant shuttle serves as a proof-mass, the ambient air as the damper and suspension elements as springs. Given the good sensitivity of vibrating sensors, resonators are commonly used for accelerometers and gyroscopes. Another major use is as time-keeping/frequency reference (filters, clocks). For this purpose, mechanical resonance is widely applied in high precision oscillators (i.e. electromechanical systems formed by a resonant structure plus an external circuit that provides energy to sustain steady-state oscillations). Quartz-crystal resonators have been so far used in a multitude of time-keeping and frequency reference applications.

Resonant structures are core to a large number of products commonly used nowadays, such as wireless or communication devices. The recent growth of interest in mobile personal communication systems has driven this sector in the direction of two primary goals: to lower costs and minimise the size of mobile devices. Miniaturisation is either concerned with new transmitting and receiving architectures, or miniaturisation of individual components. Although quartz-crystal resonators are very stable and free form extraneous effects, they present a major drawback due to their size. Also, the crystal needs to be mounted between conduction electrodes interlaced with the elements of an electrical circuit (Bilic, 2001). These considerations initiated numerous studies that focuses on resonating devices, both oriented at improving quartz-crystals performance and size, as well as introducing new manufacturing techniques. Due to their compact size, microelectromechanical resonators appear to open exceptional possibilities for creating miniature-scale devices for wireless communication and mobile technologies (Mattila, 2002). Micro-scale mechanical structures also allow enhanced mass sensitivity, which is beneficial in their application as mass sensors [Lee, 2008]. In addition, silicon micromechanical resonators are fabricated using SOI processes and, unlike quartz-crystals, can be directly integrated on CMOS chips.

Since the acknowledgement of microresonator potentials, these devices have attracted the interest of a research community committed to automate their design process. Microresonators have fast become the micro-device that is more often explored using innovative generative methods. Automated techniques can facilitate their design in many aspects. Firstly, their design is done predominantly by hand. Automated synthesis can speed up their design, especially considering its complex multiobjective nature. The shortage of design space on chips offers the challenge of generating design alternatives that fit a designated space: a typical constrained optimisation design task easily approachable with automated synthesis. The fabrication process does not put limits to the shape of these resonators, offering the possibility to generate new shapes and new solutions for reduced design spaces. Silicon resonators can in fact have many shapes and vibration modes: beams (bending mode), plates (contour mode), rings (elliptical mode), bars (shear mode), squares (Lamé and bulk mode) and any combinations of the above mentioned. Finally, the development of new sensors and actuators to be fabricated by silicon micromachining techniques requires many prototypes to be tested (Hubbard, 1996). The primary method for creating MEMS by trial-and-error is a slow process, not robust to design variation, and produces only informal guidance. Synthesis can be employed in order to reduce the design time necessary to generate the high number of prototypes required for experimentation.

Chapter 2 references many attempts to synthesise such devices, including the research work done by Zhou et al. These works will not be described here again, except the case study of interest regarding the synthesis of a particular type of microresonators called 'meandering' resonator. Fedder was the first to introduce meandering resonators [Fedder, 1995], naming them as a consequence of the fact that the resonating mass in these devices is supported by four serpentine springs. Serpentine springs get their name from the meandering snake-like pattern of the beam segments (Figure 4.12). In his work on modelling and simulation of microresonators, Fedder focuses his attention on the model of meander springs and underlines their advantages. Meander suspensions have spring constants that are linear over a relatively large displacement and are less sensitive to residual stress than straight-beam fixed-fixed suspensions, even though sizing them accurately is a lengthy process. The model developed by Fedder gives an idea of the onerous calculations related to the design of meandering springs. Since these structures are still manually designed in a physical layout and analysed using electromechanical and mechanical finite-element simulation, this type of resonator is a good case study for which the application of automated synthesis can lead to interesting solutions with a reduction of design and development time.

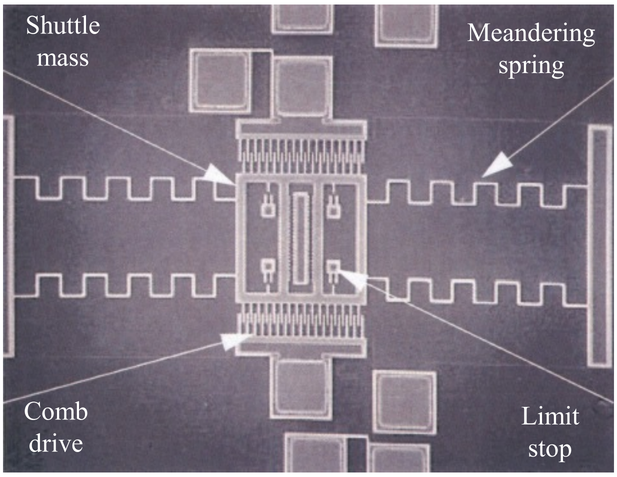


Figure 4.12: Scanning electron micrograph of a lateral microresonator suspended by four meandersprings [Fedder, 1995].

4.2.2 Description of the Benchmark Problem: Synthesis of a Meandering Resonator using MOGAs

This section gives a description of the case study on meandering resonators tackled by Zhou et al. that is used as a benchmark. A meandering resonator is made of a central mass supported by four serpentine spring structures (Figure 4.13). The device's vibration response is driven in the preferred direction by electrostatic comb drives. The goal of this synthesis task is to design a meandering resonator matching the assigned design specifications for a specific natural frequency and stiffness in the x and y-direction. This

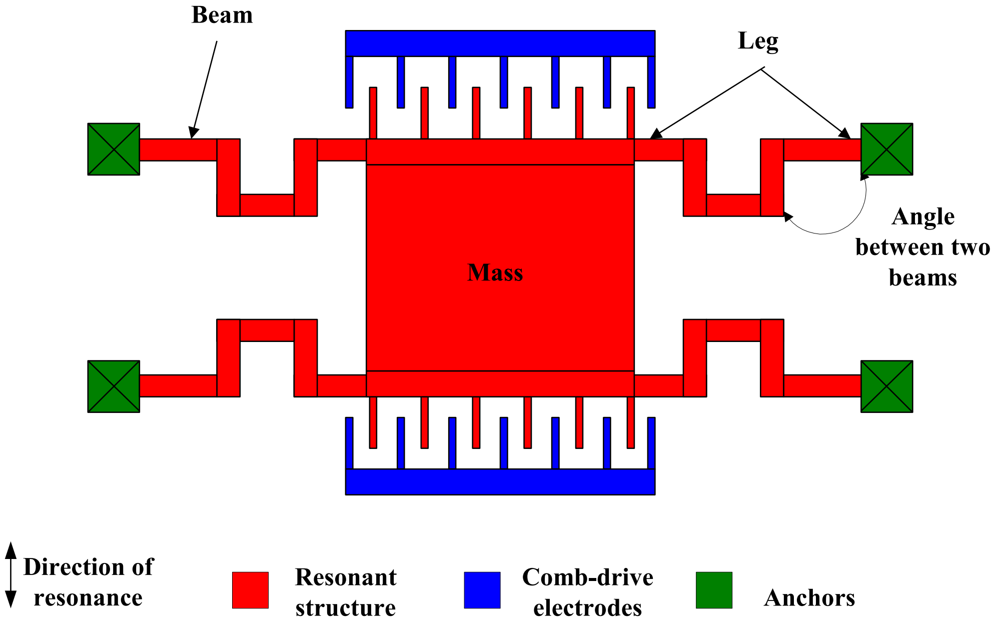


Figure 4.13: Geometry of the resonator.

was done by combining five parameterised building blocks: one centre of mass and four meandering springs. Each spring is a cluster of one anchor and N beams connected sequentially, and is defined by the number of beams *N* and length, width and angle of each beam. Upper and lower constraints are assigned for each of these parameters. Each spring building block has only one node connected to the centre of mass. The centre of mass is defined as a four-node plate made by beams with fixed length and width. The design task is to choose the right building blocks and the numerical values of the associated parameters to achieve combinations that satisfy the specified design objectives and constraints. In this particular case, design variables were (for a maximum of *92* variables and a minimum of *12* variables):

- length and width of the beams
- angles between the beams and
- the number of beams constituting each spring (*6* maximum)

More specifically, the numerical value of the objectives of the synthesis task were:

- to match a lowest natural frequency $f_0 = 93{,}723\ rad/s$

- to match the spring stiffness in *x*-direction $Kx_0 = 1.90\ N/m$
- to match the spring stiffness in *y*- direction $Ky_0 = 0.56\ N/m$.

Each soft constraint value used in the multiobjective search was calculated by taking the absolute value of the difference between the target and actual value for each equality constraint. Starting from a population size of *400*, in *30* generations the MOGA was able to find a set of *26* ‘equally good’ Pareto solutions with different configurations [Zhou, 2002]. These results show the capability and potential of MOGAs in finding viable solutions to MEMS design tasks.

4.2.3 Solution of the Task Using CNS-Burst Method

Formulation of the Optimisation Task and Performance Evaluation

The case study developed by Zhou et al. described in the previous section has been reproduced in this research in order to synthesise the same meandering resonator using the CNS-Burst method and to compare its results and efficacy with MOGAs. Here the simulation and analysis of the solution is once again performed using SUGAR and limited to linear analysis. This choice is dictated by the fact that SUGAR is also used by Zhou and can guarantee a direct comparison of CNS-Burst with MOGAs, so that results are not influenced by the type of analysis adopted. The objectives of the search have been expressed in the previous section. The formulation of the optimisation model used here is the following:

$$\min\{|\Delta f|, |\Delta Kx|, |\Delta Ky|\}, \qquad (4.4)$$

S.t. *Design Constraints (see Table IV.II, IV.III)*

where $\Delta f = (f_0 - f)$ is a “soft constraint” representing the error in natural resonant frequency from the target $f_0 = 93{,}723\ rad/s$, $\Delta Kx = (Kx_0 - Kx)$ is a “soft constraint” representing the error in spring stiffness in the x-direction from the target $Kx_0 = 1.90\ N/m$, and $\Delta Ky = (Ky_0 - Ky)$ is a “soft constraint” representing the error in spring stiffness in the y-direction from the target $Ky_0 = 0.56$ N/m. Design variables considered for the synthesis task were:

- the number of beams: *i*

- the connectivity of the beams (defining the design topology)
- length and width of the beams: l_i, w_i (the thickness of the beams was considered constant)
- the geometric position (*x, y*) of all the nodes.

This is essentially a constraint satisfaction task modelled as an optimisation task. The only hard constraint considered, which must be met throughout the search, are the ones listed in table IV.II and V.III. Due to fabrication constraints, the width of the beam elements (*w*) is limited to $1\mu m$. In order to reproduce the results obtained by Zhou, the same design constraints were initially used (listed in Table IV.II and IV.III for beam and mass primitives respectively).

Table IV.II: Design constraints for the beam primitive.

Specific to primitives	**Conflicts with other Primitives**	**Node connectivity**
▪ Maximum length = 400μm ▪ Maximum width = 20 μm ▪ Minimum length = current width ▪ Minimum width = 1 μm ▪ Number of instances N ≤ 6	▪ No overlap allowed ▪ Minimum separations between any two of non-connected beams = 50μm ▪ Angle between the beams -90°<θ< 90°	▪ Can share normal (floating) node with only one other beam, or mass primitive. ▪ If connected to anchor type node, cannot share that node with any other primitive.

Table IV.III: Design constraints for the mass primitive.

Specific to primitives	**Conflicts with other Primitives**	**Node connectivity**
▪ Dimensions of the square mass: 125μm X 12.5 μm X 2 μm ▪ Maximum value of the x- and y-OFFSET = 800 μm ▪ Minimum value of the x- and y-OFFSET = - 800 μm ▪ No node may lie inside the mass ▪ Minimum angle formed between any beam and the side of the mass = 30° For each of the 4 beams attached to the mass square: ▪ Maximum length = 400 μm ▪ Maximum width = 20 μm ▪ Minimum length = current width ▪ Minimum width = 2 μm ▪ Number of instances N = 1	▪ No overlap allowed ▪ Angle between the beams θ > 90°	▪ Can share normal (floating) node with only one other beam, or mass primitive. ▪ If connected to anchor type node, cannot share that node with any other primitive.

The angles between beams - *90°<θ < 90°* implies that each element must radiate away from the centre of mass. In order to understand the meaning of constraints for the mass primitive design, the CNS mass primitive is illustrated in Figure 4.14. The fixed mass primitive consists of four beams and a fixed square of beams forming the mass. The mass used is *100 μm X 20μm X 2μm* and is fixed.

```
 >> Primitives{1}:
      Name: 'Mass'
      MaxInstances: 1
      MinInstances: 1
      Parameters: MASS
                  WIDTH1
                  WIDTH2
                  WIDTH 3
                  WIDTH4
                   X-OFFSET
                   Y-OFFSET
 ParamLimits:
[ WIDTHi_max, WIDTHi_min;
X-OFFSETmax,X-OFFSETmin;
Y-OFFSETmax,Y-OFFSETmin]

      Data: [ 4-ports primitive ]
```

```
>> Node1, Node 2,
     Node 3, Node  4


    Name: 'Floating'
    Props: [x, y]
    CanAdd: 0
    CanRemove: 0
    PropsChangeable: [1 1]
    PropLimits: [xmin, xmax,
                 ymin, ymax]
    MinConnects: 1
    MaxConnects: 1
```

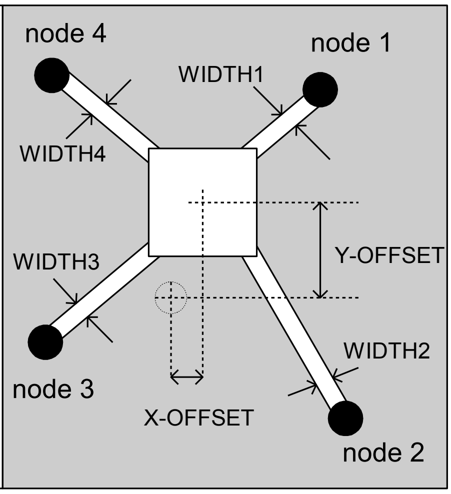


Figure 4.14. The Fixed Mass Primitive. X-OFFSET and Y-OFFSET represent the offsets from the position of the centre of gravity of the four nodes.

Also, in addition to the constraints on primitives, a specific size for the design domain is given (*X= 0 - 1000μm, Y= 0 - 600μm*), considering that in MEMS fabrication is always spatially constrained. The positions of nodes were limited to lie in the specified design domain).

The Initial Design

The Connected-Node System representation has the potential for allowing synthesis of designs from scratch or minimal starting points. This implies that the initial design used to start the search is not necessarily close to the desired design. The meandering resonator considered here as an initial design for the search is a minimal design and the meander shapes of the spring are not formed. Also, the natural resonant frequency and values of spring stiffness are far from the objective values (f_0 = *1,563,372*, Kx_0 = *145 N/m*, Ky_0 = *137 N/m* respectively). The initial design consists of a centre mass supported by four springs, which are made of connected beams extending from an anchor point to one of the four port-nodes of the mass. The choice of the width of each beam is random. As seen in 3.2.1, the beam primitive is defined as having two nodes, where the beam endpoints coincide with the nodes. An outline of the initial design is shown in Figure 4.15.

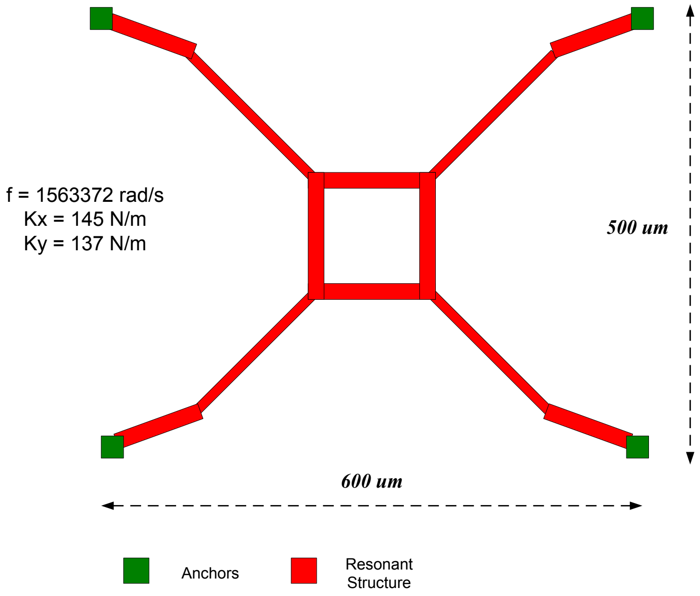


Figure 4.15: The starting resonator design.

Results of Standard Search

Starting from the initial design (Figure 4.15), the algorithm was run ten times using *11,400* iterations each, for a total time of three hours (using an Athlon XP *2.8GHz*, *1GB* RAM). As mentioned in 4.1.3, the convergence of the Pareto archive appeared rather quickly and the high number of iterations used was introduced only for the purpose of having a more direct comparison with results obtained by Zhou using *12,000* iterations. Each run resulted in an archive of non-dominated solutions distributed on a concave Pareto front. A representative archive is shown in Figure 4.16. The coordinates of the plot represent two of the objectives of the search, the spring stiffness in the *x*- and *y*- direction respectively), while the colour of the solutions represents the value of their resonant frequency. In this plot a trade-off is shown between the competing objectives of minimising the error in spring constants and the resonant frequency. Solutions having a low error in resonant frequency (indicated by a diamond in Figures 4.16) have also a higher error in the desired spring constant and are located mostly in the upper right part of the distribution. While solutions with a low error in spring constants (indicated by a circle) often suffer from a higher error in resonant frequency.

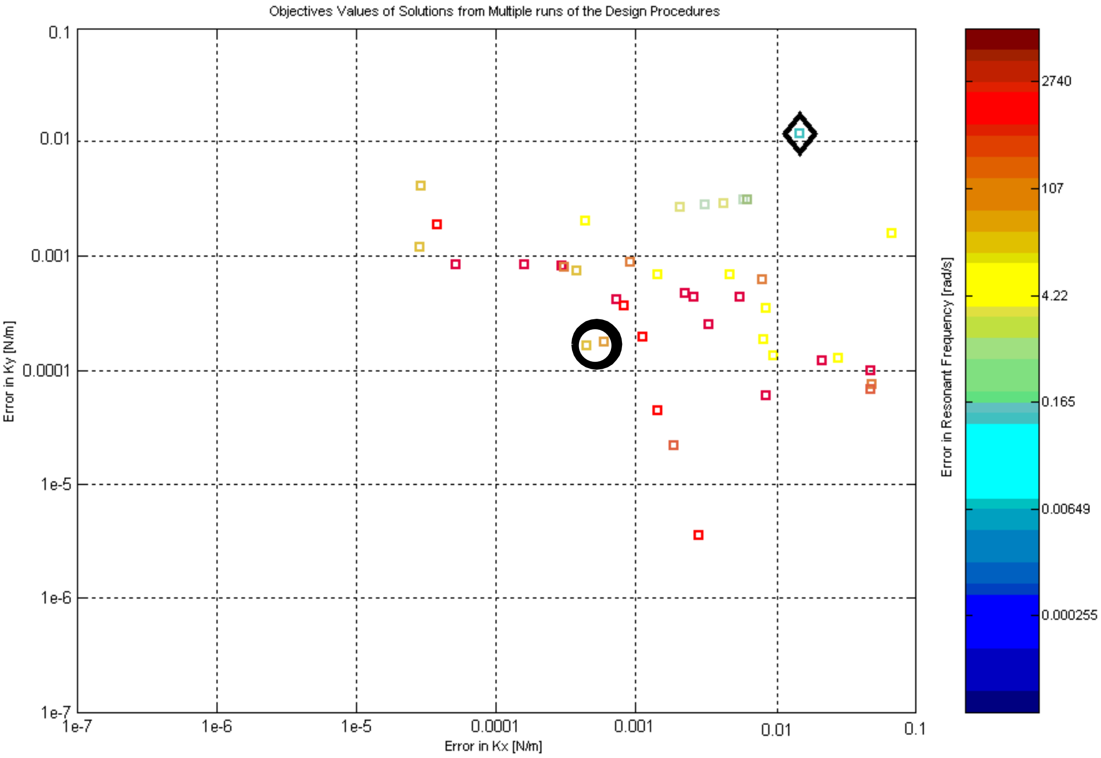


Figure 4.16: Result of a single run from the present work: plot of the objective values of the archive members with the third objective, error in frequency, indicated by colour.

Figure 4.17 shows each archive from ten runs overlaid on the same axes to give an impression of the search results. The diagram confirms that the results of each run are all distributed within the area *X= [1e-5; 10] N/m, Y=[1e-6; 0.1] N/m* (except in a few rare cases). Concerning the error in resonant frequency, solutions are distributed in the range *[3e-1; 1.6e4] rad/s*. Also, there appears to be an abundance of results in the region between *0.0001* and *0.01N/m* in the *x*-direction and *y*-direction. In general, most of the solutions of the overall archive show a good performance, considering that from *1%* downward errors in MEMS design can be considered acceptable. The maximum, minimum and average values for each objective function are listed in Table IV.IV.

Table IV.IV: Maximum, minimum and average values of the objective functions over an archive of ten algorithm runs (Figure 4.7), at 11400 iterations each.

Error in spring-stiffness in the x-direction K_x (initial value: 143.1 N/m)			**Error in spring-stiffness in the x-direction K_y (initial value 136.44N/m)**			**Error in the resonant frequency Δf (initial value 1,469,649 rad/s)**		
Maximum K_x	**Minimum K_x**	**Average K_x**	**Maximum K_y**	**Minimum K_y**	**Average K_y**	**Maximum Δf**	**Minimum Δf**	**Average Δf**
2.87 N/m	5.3e-7 N/m	0.17 N/m	0.29 N/m	1.8e-7 N/m	0.013 N/m	13951 rad/s	1.89e-4 rad/s	1291rad/s

Four of the designs that emerged from the searches are shown in Figures 4.18 through 4.21. In each of these Figures the size of the design domain is indicated by a dotted line. The Figures include each design's actual resonant frequency and *x*- and *y*-oriented spring constants. The location of the design solutions 1-4 in the overall archive is indicated in Figure 4.17. The results obtained with MOGAs using a higher number of evaluations, are indicated on the same plot.

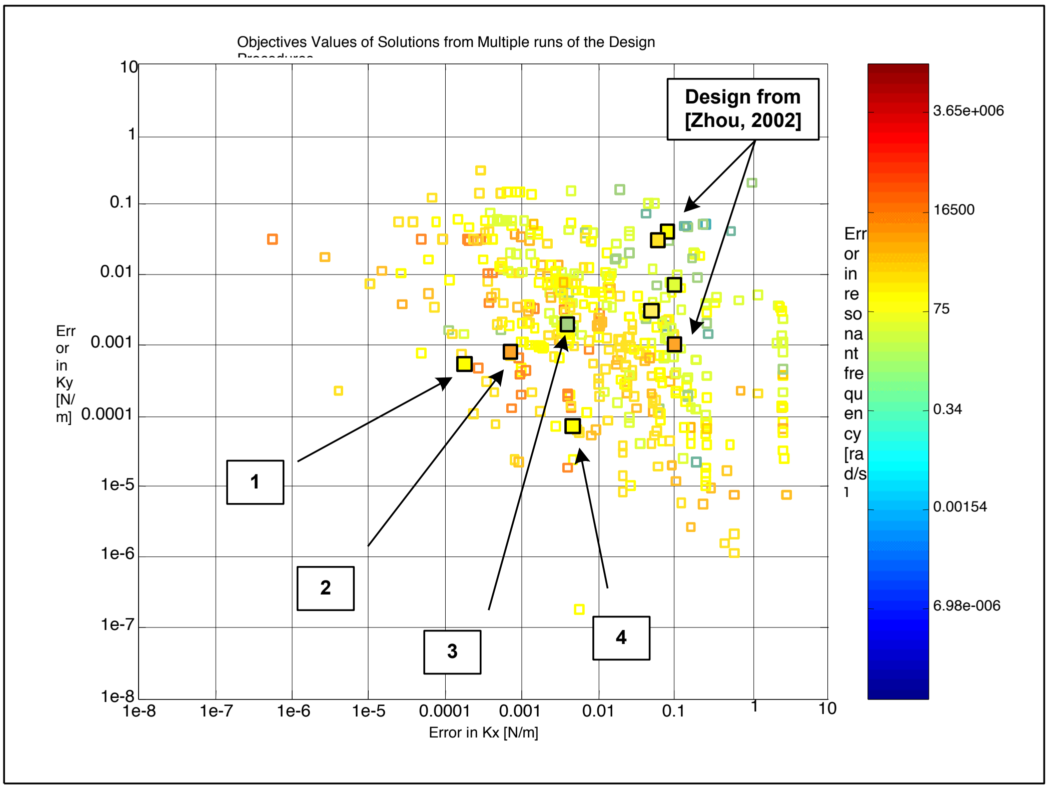


Figure 4.17: Solutions that resulted from 10 runs of the search algorithm.

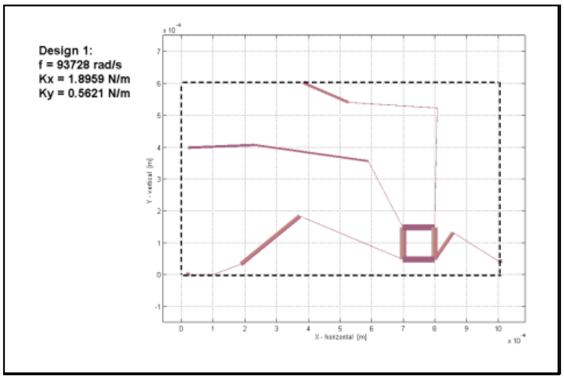


Figure 4.18: Design 1.

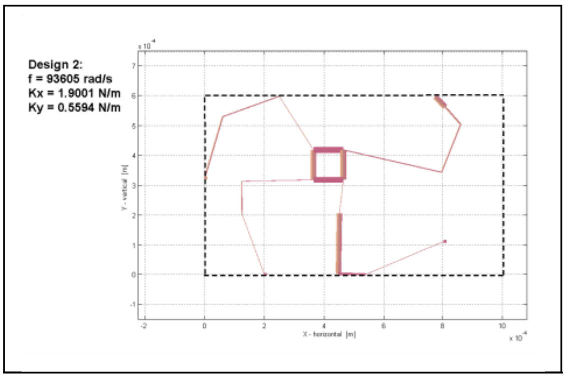


Figure 4.19: Design 2.

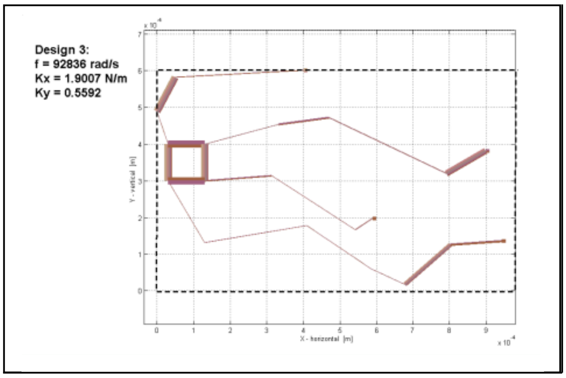


Figure 4.20: Design 3.

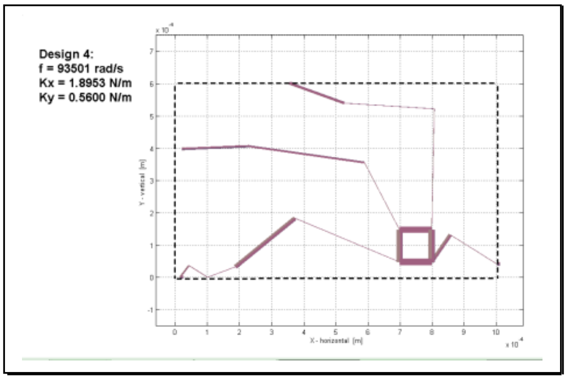


Figure 4.21: Design 4.

A Constrained Optimisation Task: Design Area Minimisation

The following experiment has been considered to examine the efficiency of the search method for optimisation tasks in addition to problems of constraint satisfaction already presented. The task consists of minimising the area of the device, defined as the bounding box around the resonator. This example is particularly relevant since in MEMS fabrication the area occupied by the device is directly proportional to its cost. The work developed at Berkeley by Kamalian [Kamalian, 2004] gives an example of such optimisation problem applied to the meandering resonator. Once again, this work has been used as a benchmark problem, and the case of the resonator with asymmetric legs has been considered (case 2 in [Kamalian, 2004]). Design objectives in this example are the minimum area and the minimum error from the target frequency of $f_0 = 10000\ rad/s$. The resonance is also constrained to be only in the *y*-direction. As in [Kamalian, 2004], the latter was obtained by highly penalising any design that had a lower stiffness in the x-direction than in the y-direction. Resonance in the *x*-direction would be impossible to excite with the comb drives aligned to the *y*-direction, as they can only perturb the centre mass in the y-direction [Kamalian, 2004].

The optimisation model for this design task is:

$$\min\{|\Delta f|, A\}, \qquad [4.5]$$

S.t. *Design Constraints (see Table IV.II, IV.III)*

where $\Delta f = (f_0 - f)$ is a "soft constraint" representing the error in natural resonant frequency and A is the area of the box surrounding the device. The hard constraint considered, which must be met throughout the search, are the ones listed in tables IV.II and IV.III.

The initial design used to start the search has a resonant frequency f_0 far from the target and a design area of $30\ \mu m^2$. Design variables considered are:

- the number of beams i
- the connectivity of the beams (device topology)
- length and width of each beam (w_i, l_i)
- the geometric position (x, y) of the nodes.

Figure 4.22 shows a representative Pareto-front of solutions obtained from a single run of *10,000* iterations of the design procedure. The coordinates of the plot represent the two objectives of the

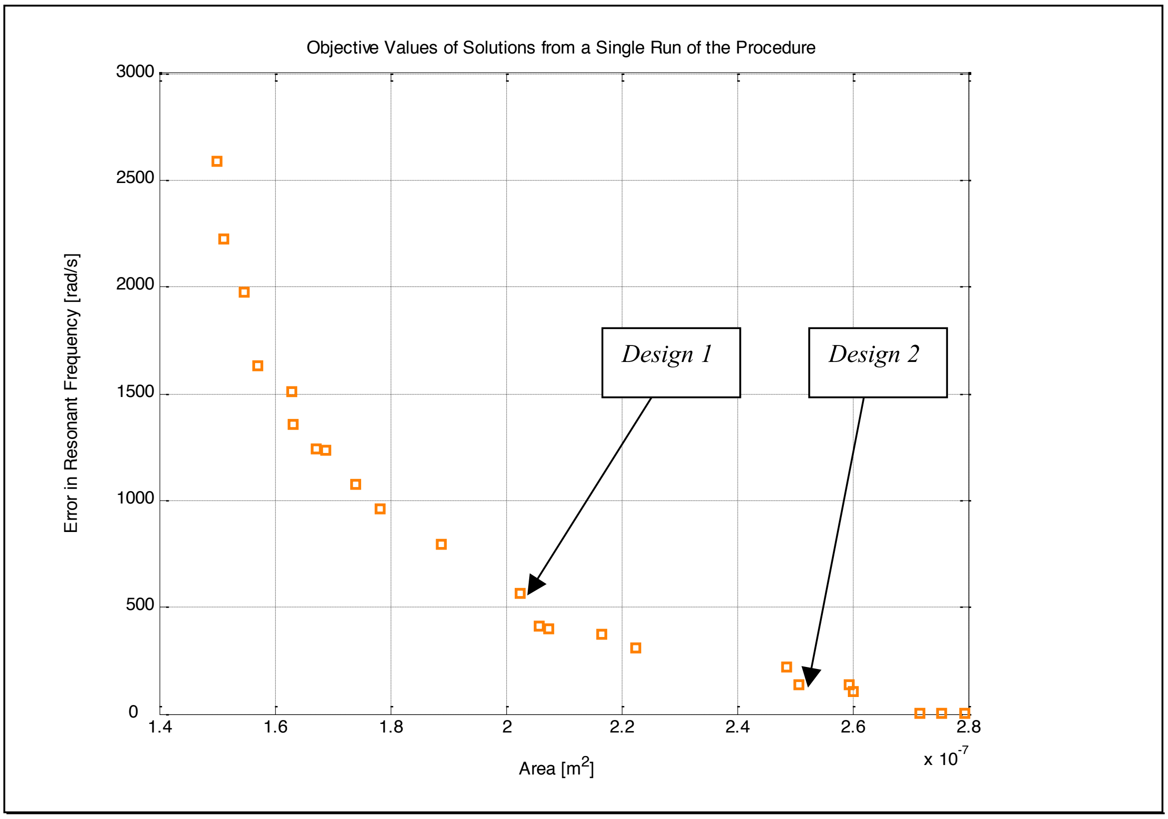
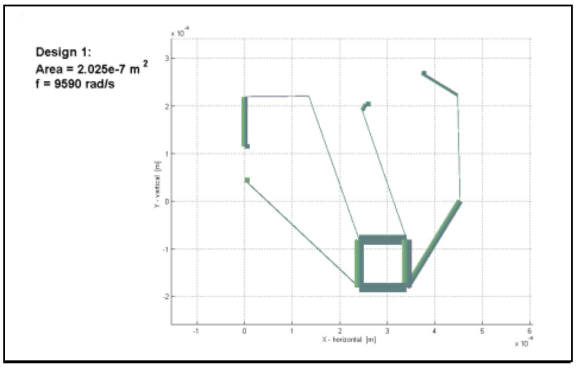


Figure 4.22: Area minimisation: Archive of solutions obtained from a single run of the search procedure at 10,000 iterations.



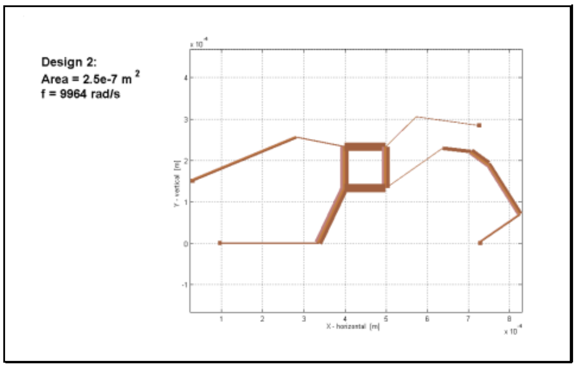


Figure 4.23: Design 1.

Figure 4.24: Design 2.

search to be minimised, the error in natural resonant frequency and the area of the device in the *x*- and *y*- direction respectively. Two of the designs obtained in this archive are also shown in Figure 4.23 and 4.24. Solution 1 presents a reduction of the area of the device of almost one-third of the total area covered by the initial design. Solution 2 presents a $\Delta f<1\%$. On average, the synthesised resonators represented in Figure 4.22 have design specifications that are much closer to the design objectives than the initial design. Results show that the method can effectively help designers in improving solutions and finding good compromises between target frequency and area minimisation.

4.3 Discussion of the Results and Conclusions

In this Chapter the capabilities of the proposed method have been compared with those of other published synthesis methods. Its efficiency in solving topology synthesis and constrained optimisation problems has been demonstrated, as well as its function of providing multiple and innovative solutions to designers.

In the first benchmark problem, the archive of designs shown presents a wide number of solutions for the task, also illustrating performance trade-offs, whereas deterministic methods such as homogenisation techniques produce only one solution. Not only do many of the results match the solution obtained through a continuum topology optimisation technique [Sigmund, 1997], but they also offer a variety of solutions to the synthesis task. More than *5%* of the designs obtained have performances that rival the optimal design synthesised by Sigmund. Moreover, even starting the search from a lumped design, some solutions resulted in a distributed configuration rather than a lumped one. This unexpected evolution of the synthesis process is also a successful result for the method, if compared to continuum topology optimisation. The main issue in synthesising compliant mechanisms using a continuous topology optimisation method is, in fact, the extensive post-processing of solutions. This post-processing procedure is eliminated using the present discrete method.

The second benchmark problem demonstrated the effectiveness of CNS-Burst in comparison with MOGAs. As seen in Figure 4.17, design solutions rival those obtained by Zhou, and the number of evaluations for each run used in the present work (*11,400*) is lower than the one used with MOGAs (population size of *400* for *30* generations, for a total *12,000* design evaluations). The research also showed that convergence to solutions included in Figure 4.17 occurred rather quickly and that the full *11,400* design evaluations were unnecessary. As for the case of area minimisation, a comparison with Kamalian's work [Kamalian, 2004] gives once again an idea of how competitive CNS-Burst is in solving problems of constrained optimisation. Even though for meandering resonators the area occupied by the comb drives is not taken into account as done by Kamalian, the comparison showed once again that better results have been obtained with a lower number of iterations. In general, the results at least matched and often improved results obtained by Zhou and Kamalian. These results illustrate the effectiveness of the method, especially considering its generality. The big variety of solutions obtained for this case study also gives an idea of how the method can help designers in choosing among different trade-offs, as well as fostering better understanding of the trade-off among objectives.

These experiments set up the basis for further applications and improvement of the method. Considering the encouraging results, a deeper investigation of the search technique in the field seems to be worthy. The subsequent work consisted in applying the method to complex and real MEMS design problems in order to prove its utility to the designers' community. The method has been modified and extended, without altering its generality and efficacy.

CHAPTER 5 – SYNTHESIS OF MICRORESONATORS

In the previous chapter benchmark problems have been used to compare CNS-Burst with other synthesis techniques. Chapter 1 highlighted desirable characteristics for generative tools, such as flexibility and scalability. This chapter presents examples that validate the efficacy of the method and test the above mentioned characteristics. The case studies have been specifically set to exemplify the innovative aspects introduced by the method in the field of MEMS automated synthesis.

As anticipated in Chapter 3, one of the objectives of this research is to prove the reusability and flexibility of the method through its application to different synthesis scenarios. A second objective is to explore how stochastic search techniques perform in generative tasks, testing them with three different types of optimisation problems: size, shape and topology optimisation. Although successful results in topology optimisation prove the real efficiency of the method in the synthesis process, the completeness of the method is demonstrated through its application to a variety of optimisation tasks. Size optimisation is introduced here to show how CNS-Burst is adaptable to different search requirements.

Another achievement of this work is the possibility to find accurate designs, in order to facilitate post-processing efforts. Considering the high number of failures at the fabrication stage, correct generation and simulation of solutions reduces considerably the amount of experimentation and prototypes required, and consequently the high fabrication expenses.

Finally, the experiments presented are aimed at exploring the use of the method in the MEMS design domain, as well as establishing its limits and advantages for the field.

As seen in Chapter 4, the case studies examined here are microresonators. Their applications and the interest behind these devices have been mentioned in 4.3.1. While the case studies shown in Chapter 4 were simplified examples, the ones considered here are real cases of applications in wireless and mobile communications. These design tasks are problems solved in everyday design practice, and demonstrate potential contributions of CNS-Burst to the MEMS design process.

Different design cases have been used for each optimisation task. Longitudinal mode resonators have been chosen to explore the size optimisation capabilities of the method. For testing the method in topology optimisation tasks, so-called 'sandwich resonators' have been used. For exploring the contribution of CNS-Burst in shape optimisation tasks, a methodology is presented, supported by an introduction to potential design cases. In each of these examples, comparison with hand analysis is used to benchmark the method.

5.1 A Size Optimisation Task

The application of CNS-Burst to size optimisation tasks serves the purpose of demonstrating the capabilities of the method in solving any design challenges. CNS-Burst's synthesis potentials go beyond the abilities of optimisation methods. The method is in fact able to perform multi-objective search of complex topologies with difficult design objectives that would not be possible with commercial optimisation toolboxes. The following design task also illustrates one of the main purposes of the method, to automatically generate spaces of alternative designs that tradeoff multiple design objectives.

5.1.1 The Free-free Beam Resonator Design

A large interest has been recently demonstrated by industry in low noise microelectromechanical-based reference oscillators. In particular, 'high-Q' microresonators with primary resonance in the *10MHz-3GHz* range raised the attention of wireless communication and mobile technologies manufacturers. For applications in these sectors, a typical microresonator considered by designers is the so-called free-free (FF) beam microresonator. The advantages of using this resonant structure will be clarified in the next section, where an explanation of the main design parameters of the resonator is given.

A FF-beam resonator includes a resonant structure (shaded red in Figure 5.1) and electrical and sense elements (shaded blue in Figure 5.1). The resonant structure is anchored in its centre (anchors shaded green in Figure 5.1), while its ends are free to resonate. At operational frequency, the resonant structure of the FF-beam resonator vibrates longitudinally. The two arms move anti-phase, changing the dimensions of the gaps between the resonant beam and electrodes and, as a consequence, altering the sensed capacitance between the electrodes. This longitudinal vibration mode is a subset of the class of bulk acoustic modes.

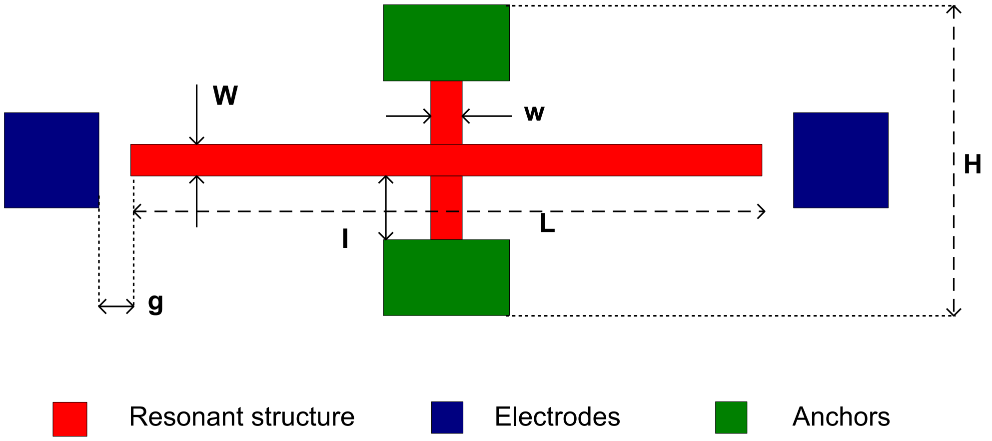


Figure 5.1: FF-beam resonator model (red: resonant structure; green: anchors; blue: electrodes).

For FF-beam resonators, a model derived from vibration theory has been developed and validated [Mattila, 2002] and will not be detailed here. According to the theory, the eigen mode of interest in FF-beam resonators is observed when the length of the resonator arm L equals a quarter of the bulk acoustic wavelength λ [Timoshenko, 1970]. The formula for the longitudinal mode frequency is expressed as follow:

$$f = \frac{v_s}{\lambda} = \frac{\sqrt{E/\rho}}{4L} \tag{5.1}$$

where v_s is the speed of sound, E is the Young's Modulus, ρ is the density of the material used (silicon in this case, with $E = 165\ GPa$ and $\rho = 2330\ kg/m^3$) and L is the half length of the resonant structure.

In order to explain the design parameters of the resonators, an equivalent mechanical model and its corresponding electrical model have been introduced [Mattila, 2002] and are reported in Figure 5.2. As mentioned in 4.3.1, a resonator can be thought as a second order mass-spring-damper mechanical system. The equivalence between a mechanical resonator and its equivalent electrical circuit is a common way to explain MEMS parameters and their behaviour.

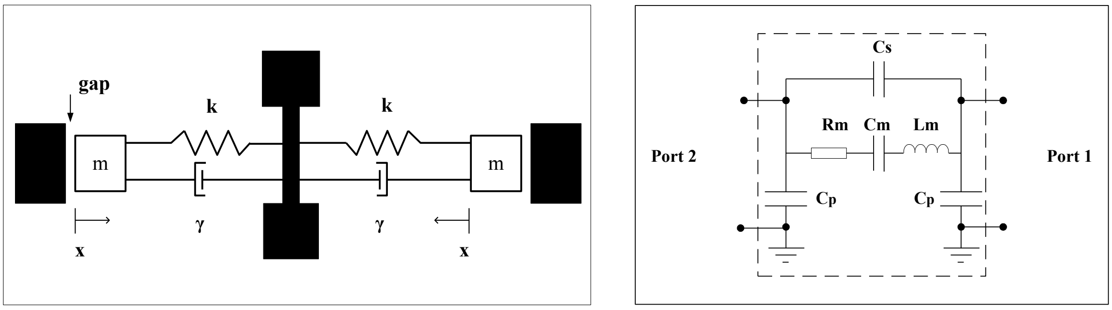


Figure 5.2: a) Equivalent mechanical model of the FF-beam resonator; b) Two-port equivalent electrical circuit of the resonator, including serial and parallel capacitances (C_s and C_p respectively) [Yan, 2006].

From the equivalence of the electrical and electromechanical circuits and the electrostatic, described by Mattila [Mattila, 2002], the following design characteristics can be defined for the resonator:

Motional Resistance:

$$R_m = \frac{\sqrt{k_{eff} m_{eff}}}{Q\eta^2} \tag{5.2}$$

Motional capacitance:

$$C_m = \frac{\eta^2}{k} = \frac{\eta^2}{m_{eff}\omega_0^2} \tag{5.3}$$

Motional inductance:

$$L_m = \frac{m_{eff}}{\eta^2} \tag{5.4}$$

Where

- η is the transduction coefficient, depending on direct current, bias voltage and the working capacitance/gap in vacuum. It can be defined as:

- $\eta = -V_{DC}\dfrac{\partial C}{\partial g} = 2V_{DC}\varepsilon_0 WT\dfrac{1}{(g+g')^2} \cong V_{DC}\varepsilon_0\dfrac{2WT}{g^2}$ (5.5)

  where

  $C = \dfrac{\varepsilon_0 WT}{(g+g')}$ is the capacitance sensed at one end of the vibrating arm;

  g' is the change of dimension of the gap at vibration ($g \ll g'$);

  ε_0 is the permittivity;

  V_{DC} is the voltage;

  T is the thickness of the resonator;

  $A = WT$ is the area of the capacitor (or section of the arm of the resonator).

- m_{eff} is the effective mass of the resonator (depending on the density of the material an geometry). This is the equivalent mass for the lumped system representation (Figure 5.2a) and is a function of the mode shape as well as the geometric parameters of the resonator. The effective mass differs from the mass of the FF-beam resonator by a constant factor and can be analytically calculated for the longitudinal mode as follows [Weaver, 1990]:

  $$m_{eff} = \frac{\rho LWT}{2} \tag{5.6}$$

- k_{eff} is the effective spring stiffness of the system (depending on the geometry of the resonating part and the Young modulus)
- Q is the quality factor
- $\omega_0 = 2\pi f$

R_m, C_m, L_m, Q, η and of course the resonance frequency f define the behaviour of the resonator. Only some of these design parameters will be considered here as objectives of the search. Their meaning in term of design will be described in the following sections.

5.1.1.1 Key Design Performance Criteria

Resonant Frequency: f

As the aim of the example presented is designing resonators for a desired resonant frequency, f is the most important design criteria to take into account. The longitudinal frequency for FF-beam resonators - the one of interest in this case study - is expressed in (*5.1*). Resonant frequency depends on used material and geometric characteristics. For example:

- High Young's modulus or low-density materials induce high resonant frequency
- Short resonant elements have higher resonant frequency.

Motional Resistance: R_m

The motional resistance, expressed in (*5.2*), represents electrical loss. A large R_m indicates an increase in the ratio between the energy dissipated and the energy stored in the resonator. Power dissipation and high electrical loss also result in higher noise phenomena. A small R_m is therefore desirable for microresonators.
Many factors influence the value of the motional resistance, such as the fabrication process and the material parameters. These include transducer gap, Young's modulus, density and polarisation voltage. However, different electrode drive modes and different vibration modes primarily determine the value of the motional resistance.

Quality Factor: Q

The quality factor can be expressed as:

$$Q = 2\pi \frac{W}{\Delta W} \tag{5.7}$$

where W is the maximum vibration energy stored by the system per cycle and ΔW is the energy dissipated per cycle of vibration [Hao, 2003]. In a microelectromechanical resonator the dissipation mechanisms are due to the following factors:

1) Fluidic damping Q_{air}: this component of the quality factor is the energy loss in vibrating structures due to fluid viscosity (air). Q_{air} does not influence Q when microresonators operate in moderate vacuum. Sometimes vibration amplitudes are so small that air damping does not represent an issue, even at atmospheric pressure.
2) Thermoelastic damping Q_{TED}: TED is a loss of energy in the bulk of material constituting the resonant structure, and is the most well-known and investigated component of Q [Liftshitz, 2000; Srikar, 2002]. It is the result of a phenomenon called thermoelastic friction, which takes place in any material subject to cyclical stress resulting in deformation. Any materials heat under compressive stress and cools under

tensile stress due to internal friction and elastic/non-elastic behaviour. This phenomenon appears even in perfect silicon crystals. Due to the heat flow from warmer to cooler regions, energy is lost as non-recoverable thermal energy. The amount of damping depends on the rate of this energy loss, that is related to vibration frequency and thermal relaxation time constant (time that the material requires to relax after a constant stress or strain is applied).

3) Surface damping $Q_{surface}$: this loss mechanism has been investigated only recently and mainly in relation to MEMS [Yasumura, 2000; Yang, 2002]. It can be explained with same mechanisms that explain Q_{TED}, except that it can be defined as bulk dissipation only in surface layers. These layers of oxide or other absorbed chemicals have much worse dissipation qualities than bulk material constituting the device. This phenomenon is particularly relevant in MEMS, as their surface/volume ratio is high. Surface treatments and dimensions choice can mitigate this phenomenon.
4) Support damping $Q_{support}$: this is also an energy loss phenomenon recently investigated in MEMS devices [Hao, 2003; Bindel, 2005]. The loss represents the portion of the vibration energy of a resonator dissipated by transmission through the supports. A resonant structure exerts a time-harmonic load on its support structure. Acting as an excitation source, the time-harmonic load further excites elastic waves propagating into the supports. Part of the vibration energy dissipated through elastic wave propagation in support media is commonly refereed to as anchor or support loss. Dimensions and mode of vibration have a major influence on the phenomenon, and clever design of the anchors can help mitigate it.

The measured Q is mainly the combination of these dissipation mechanisms and can be expressed as [Hao, 2005]:

$$\frac{1}{Q_{Tot}} = \frac{1}{Q_{air}} + \frac{1}{Q_{TED}} + \frac{1}{Q_{\sup port}} + \frac{1}{Q_{surface}} \tag{5.8}$$

A high Q is desirable as it translates into improved stability and a lower motional resistance. For FF-beam resonators Q is usually of the order of magnitude of *10000* in air and *11,5000* in vacuum. The design of the supports is of particular interest, as anchor loss is a factor that limits Q substantially.

Key Design Performance Criteria: Some Considerations

With the increasing demand of microresonators for high frequencies and with very high *Q*, bulk mode capacitive resonators have rapidly become an interesting choice, due to their high stiffness and consequently much higher resonant frequency (from *20MHz* to *400MHz* for RF applications). The bulk mode of interest is usually the lowest one among the longitudinal modes. Also, resonators that base their functioning on a flexural vibration mode, when fabricated in the scale to have eigenfrequency around *10MHz*, usually exhibit quality factors in the range of 10^3-10^6 [Lee, 2001]. It has been shown [Nguyen, 1999] that significantly higher *Q*-factors can be obtained in microresonators operating in bulk-acoustic mode. On the other hand, the disadvantage of bulk mode resonators is their small electrode surface, contributing to a smaller sensed capacitance and higher motional resistance.

As the parameters described in the previous sections show, microresonator design is complex, due to the many behavioural considerations. The interdependency of the design parameters creates several trade-offs to be taken into account. For example, increasing the electrode surface would decrease R_m, but this would also lead to a lower operational frequency of the system. It is indeed difficult to take into account all the relations between design parameters, especially if the design is carried out by hand. Automated synthesis techniques incorporating multiobjective optimisation can help designers analysing trade-offs or suggesting innovative solutions.

Two design objectives have been here considered for the design of the FF-beam resonator: motional resistance R_m and longitudinal resonant frequency *f*. The quality factor *Q* will be kept constant and will not be considered as an objective in the following example.

5.1.2 Optimisation Model with Two Design objectives

In this section results obtained using CNS-Burst to synthesise and optimise designs for FF-beam microresonators are reported. The design objectives of the search are:

- A target operational frequency f_0 of *20 MHz* (a constraint-satisfaction problem formulated as a soft constraint)
- A minimal motional resistance R_m (formulated as a minimisation problem).

The design variables are the length and width of the anchors (*w*, *l*) and the length and width of the resonant beam (*W*, *L)*, the position of the nodes but not the primitives connectivity. DC voltage, quality factor, capacitive gap and thickness of the resonators are design parameters in the optimisation process ($V_{DC} = 100\ V$, $Q = 10{,}000$, $g = 1\mu m$ *and* $T = 20\mu m$ respectively).

The optimisation model for this design task is:

$$\min\{|\Delta f|, R_m\}, \qquad (5.9)$$

$$S.t.\ H \leq 61\mu m$$

Design Constraints (see Table V.I)

where $\Delta f = (f_0 - f)$, *H* is the height of the resonator (*H* in Figure 5.1). The minimum width of the beam elements of $1\mu m$ is due to fabrication limitations (*w*, *W* in Figure 5.1).

Table V.I: Design constraints for the beam primitive.

Specific to primitives	Conflicts with other Primitives	Node connectivity
▪ Maximum length = 40 μm ▪ Maximum width = 220 μm ▪ Minimum length = 1 μm ▪ Minimum width = 180 μm ▪ Thickness = 20 μm ▪ Number of instances N=1	▪ No overlap allowed ▪ Minimum separations between any two of non-connected beams = 50μm ▪ Angle between the beams θ = 0°	▪ Can share normal (floating) node with maximum one other beam ▪ If connected to anchor type node, cannot share that node with any other primitive.

5.1.3 Implementation of the Design Objectives

The commercial package used for analysis and simulation of design solutions in this case study is the COMSOL-MEMS Module. Advantages of this package are:

- It incorporates the required MEMS multiphysics

- It handles geometric non-linearities
- It performs damped eigenfrequency analysis.

The main adaptations of CNS-Burst to the specific design task described in 5.1.2 have been the following:

- A new module was introduced to automatically post-process simulation results and find the bulk mode. The module acts as a mode/frequency selector, as explained in the following section.
- A new module was also introduced to automatic calculation of the motional resistance.

These two modules are described in the following sections.

Automated Search of Longitudinal Bulk Mode

The objective of the search for this case study is not only to find a resonator with the required resonant frequency, but also to find a structure operating in the longitudinal resonant mode. The vibration of the resonator must be restricted in space in order to avoid non-linear behaviours and overstepping of the geometric design boundaries. Limitations for the vibration mode must be both in the longitudinal direction and in the vertical direction, i.e. the vibration of the resonator must be in plane. Generating designs with the required frequency as well as the required vibration mode is a new and unexplored challenge in automated synthesis of MEMS, as the efforts of previous research primarily focused on generating designs for a target frequency [Kamalian, 2004; Zhou, 2002].

The primary effort of the search is to discover the bulk mode for each design created, and then verify that the resonant frequency corresponding to that mode is in the targeted range. It is desirable that the resonator functions at bulk mode frequency, as this mode guarantees higher quality factor than the first flexural natural mode. The bulk mode is usually higher than the first mode. Hence, the calculation of the first mode frequency would not correspond to the desired mode. The bulk mode number is not known a priori and the challenge in implementing this search rests with the fact that an extended spectrum of modes must be calculated and evaluated in order to find it. For this particular case, for example, about 28 modes must be considered before the bulk mode is found.

This search has been implemented through an algorithm constituted of two steps (Figure 5.3). The algorithm is applied to any new design generated. The first step of the search consists in extracting the values of the stress matrix for selected vertical sections of the arms of the resonator and examining their value. The value of the stress component in the longitudinal direction of the arms must be uniform across any vertical sections. This can be verified by randomly sampling points of the same section and comparing the longitudinal stress component for each point. As the two arms of the resonator move anti-phase, it must also be verified that the symmetric sections on the two arms have identical longitudinal stress component (in absolute value), but are opposite in sign. For this crosscheck, three stress sections are sampled on one arm of the resonator, as well as the three corresponding symmetrical sections on the other arm. The number of sections chosen (J_{MAX} in Figure 5.3) may be parametrised. A further constraint is required to confirm that the vibration is in plane, i.e. that the mode analysed is exactly the bulk one. This is easily verified by checking that a number of points selected on the surface of both arms of the resonator do not displace vertically. The bulk mode of interest in this case is the primary one. Once the bulk mode has been obtained, the second step of the algorithm consists in calculating the eigenfrequency corresponding to the bulk mode, in order to verify that it is in the desired range.

It is understandable how this verification process can be quite lengthy for each design. The number of modes examined can be high. Also, the simulation of each mode is expensive in terms of computational time. One way to automate this process and cut computational times is to avoid calculations of all modes starting from the first one. The search is started instead from a mode number (n in figure 5.3) that is an initial guess of the bulk mode number. The initial guess can be deduced from the hand analysis used to define the initial design, or from practical experience. The procedure involves simulating this n-mode behaviour and verifying whether it is the bulk mode or not. In case it is not the bulk mode, other modes are examined in the range ($n \pm i$), with i being a counter ranging from one to I_{MAX} (where $I_{MAX} = 5$ in this case). The motivation for searching the bulk mode in

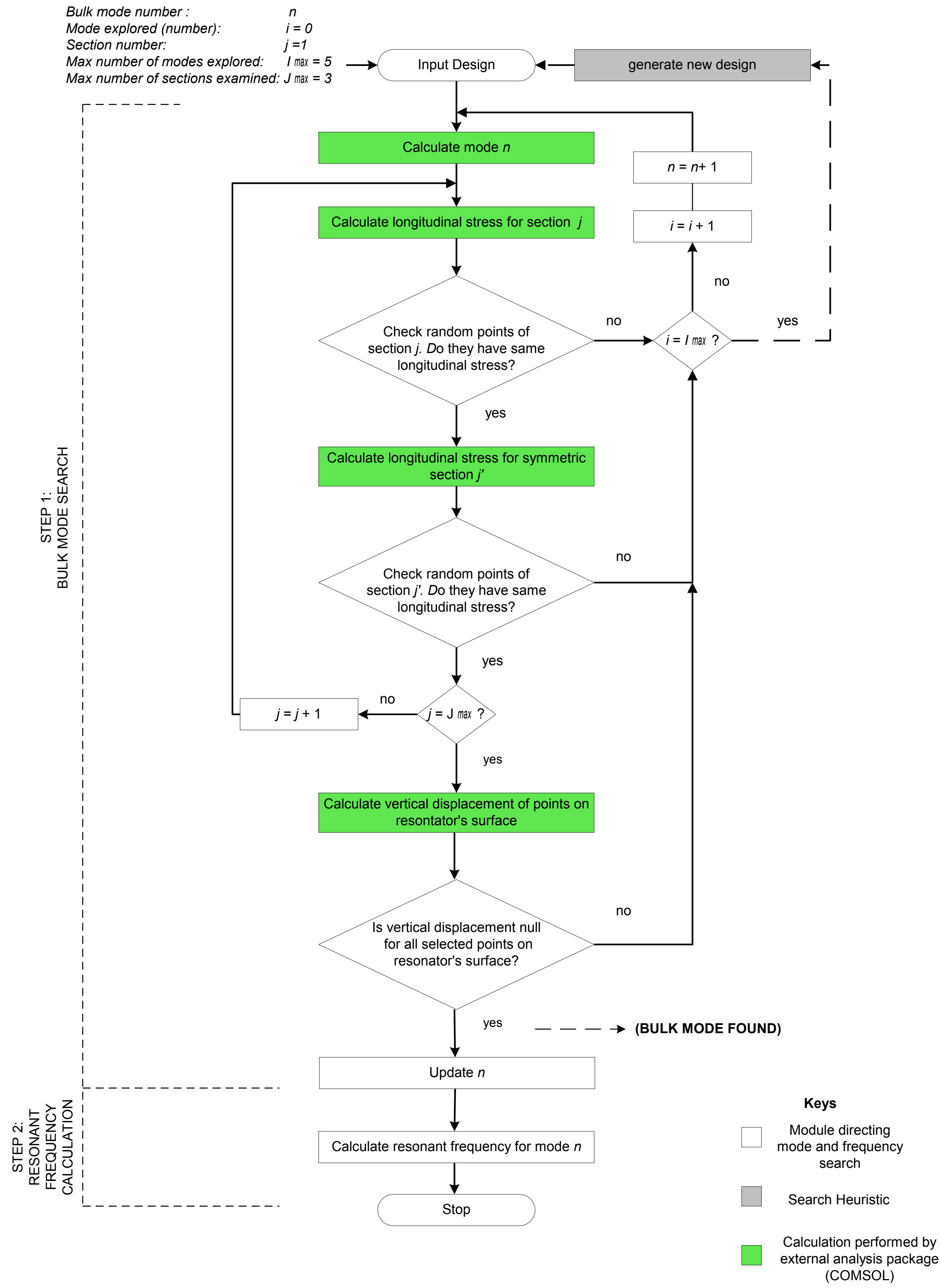


Figure 5.3: Longitudinal Bulk Mode Search Algorithm.

such a restricted range is that the modifications of each design during the search are not radical at each burst. As designs evolve slowly, it is possible that a newly generated design has bulk mode number close to the one of the previous design. Each time a new design has different bulk mode number from the design previously examined, *n* will be updated and used as a starting point for the next search of bulk mode. The *n ± 5* range has been chosen completing an unconstrained run of the algorithm. In this case the search of the bulk mode for each design involved the calculation of all its modes, starting from the first, until the bulk mode was found. This run highlighted how, fixing a *25%* maximum variation of primitive dimensions in each modification, the generated designs never presented bulk mode number higher than *n ± 5* (where *n* was the bulk mode number of the initial design analysed). Results from unconstrained searches run without limiting the variation of the primitive dimensions, showed that high changes in dimensions generated high changes in resonant frequency. Unlimited modification of the length of the resonator brought generation of resonators with resonant frequency out of the desired ranges. Hence, for this search the modification operator was constrained to apply a maximum of *25%* variation to the dimensions of the resonator.

Limitation of the process described is that some designs may be excluded from the search in case the bulk mode does not fall in the range *n ± 5*. This limitation is in fact necessary to avoid uncontrolled extension of computational times. Nevertheless, the automated search proved to be accurate as empirical investigation did not show the bulk mode number falling out of this range. The procedure allows designers to analyse resonant frequency and modes for many possible solutions in a short time. The automated search of both frequency and right mode is for designers a great advantage not introduced by any other synthesis tool for MEMS. This represents a necessary step forward in the implementation of simulation-based synthesis tools, allowing the solution of a common as well as complex MEMS problem for designers.

Automated Search of Motional Resistance

Motional resistance has been defined in (*5.2*) as:

$$R_m = \frac{\sqrt{k_{eff} m_{eff}}}{Q\eta^2} = \frac{m\omega_0}{Q\eta^2} \qquad (5.10)$$

where m_{eff} is the effective mass as defined in (*5.6*), η is the transduction coefficient as defined in (*5.5*), k_{eff} is the spring stiffness and the angular velocity ω_o is known form the calculation of the bulk frequency ($\omega_o = 2\pi f$). Q is the quality factor, a constant in this case study ($Q = 10{,}000$). The transduction factor η depends on direct current, bias voltage and the working capacitance in vacuum and it has been defined in (*5.5*).

The spring stiffness k_{eff} of the system can be calculated through simulation, applying a traction force to the resonator's ends and measuring their displacement. This calculation is easily automated using COMSOL simulation package. k_{eff} can also be easily defined by analytical calculation using the following [Mattila, 2002]:

$$k_{eff} = \pi^2 \frac{EWT}{8L} \tag{5.11}$$

Results from both types of calculation demonstrated to correspond.

The motional resistance as an objective function can then be defined as follow:

$$R_m = \frac{\sqrt{k_{eff} \frac{\rho LWT}{2} g^4}}{4QV_{DC}^2 \varepsilon_0^2 A^2} \tag{5.12}$$

or (in analytical form):

$$R_m = \frac{\pi \sqrt{\rho \varepsilon_0} g_0}{8QV_{DC}^2 \varepsilon_0^2 A} \tag{5.13}$$

The automated search of motional resistance in synthesis of MEMS is a novel objective introduced by this work, made possible by the automated calculation of the spring stiffness of the system. R_m depends on geometrical factors, varying inversely with the cross-section area of the resonator. Once again the applicability of CNS-Burst to size-optimisation problems could be used to explore the scaling of motional resistance for longitudinal resonator with different shapes of cross-section. An exact calculation of Q, as it will be shown in the following example, would allow for more accurate calculation of R_m.

5.1.4 Result of a Standard Search with Two Design objectives

In this section results obtained using CNS-Burst to synthesise and optimise designs for FF-beam microresonators are reported. Five searches at *5000* iterations each were run using an Athlon XP *2.8GHz*, *2GB* RAM machine, for a total time of *20mins/100* iterations circa. Designs were generated by half symmetry. The initial half design used for the search is shown in Figure 5.4. For the purpose of creating symmetric design alternatives, a new module was introduced. The initial design used to start the search presented a resonant frequency $f_0 = 20.396\ MHz$, and a motional resistance, $R_m = 229\ M\Omega$, and was obtained using hand calculations. The geometry of the initial design is described in Table V.II.

Figure 5.4 shows an archive of solutions obtained from the five optimisation runs. Each run, resulting in an archive of non-dominated solutions distributed on a concave Pareto front, is overlaid on the same axes. The two coordinates of the plot represent the objectives of the search (the error in resonant frequency and the motional resistance in the *x*- and *y*-direction respectively). The results show that *52%* of the solutions kept in the design archive have an error in resonant frequency, $\Delta f<1\%$, and that *93%* of the solutions have a motional resistance, R_m, less than the initial value, R_m. The design archive presents a variety of solutions to the task, also illustrating performance trade-offs. Figure 5.5 and 5.6 show two interesting design solutions obtained. Solution 1 shows a Δf of *0.0013%* from the target. Solution 2 has $R = 137\ M\Omega$ (reduction of *40%* from the initial value R_m). Solution 1 and 2 are compared with the initial design (comparison with hand analysis) in Table V.II. Figure 5.7 and 5.8 show a COMSOL simulation model and a manufactured example of FF-beam resonator respectively.

As for the examples shown in Chapter 4, convergence through the search has not been necessary for this simple example. The Pareto archive resulted unchanged immediately after the first iterations, meaning that the algorithm had converged rather quickly to the final solutions and before the maximum number of iterations set was reached. Moreover, the designs observed after few iterations demonstrated to have better performance than the initial solution, confirming the efficiency of the method.

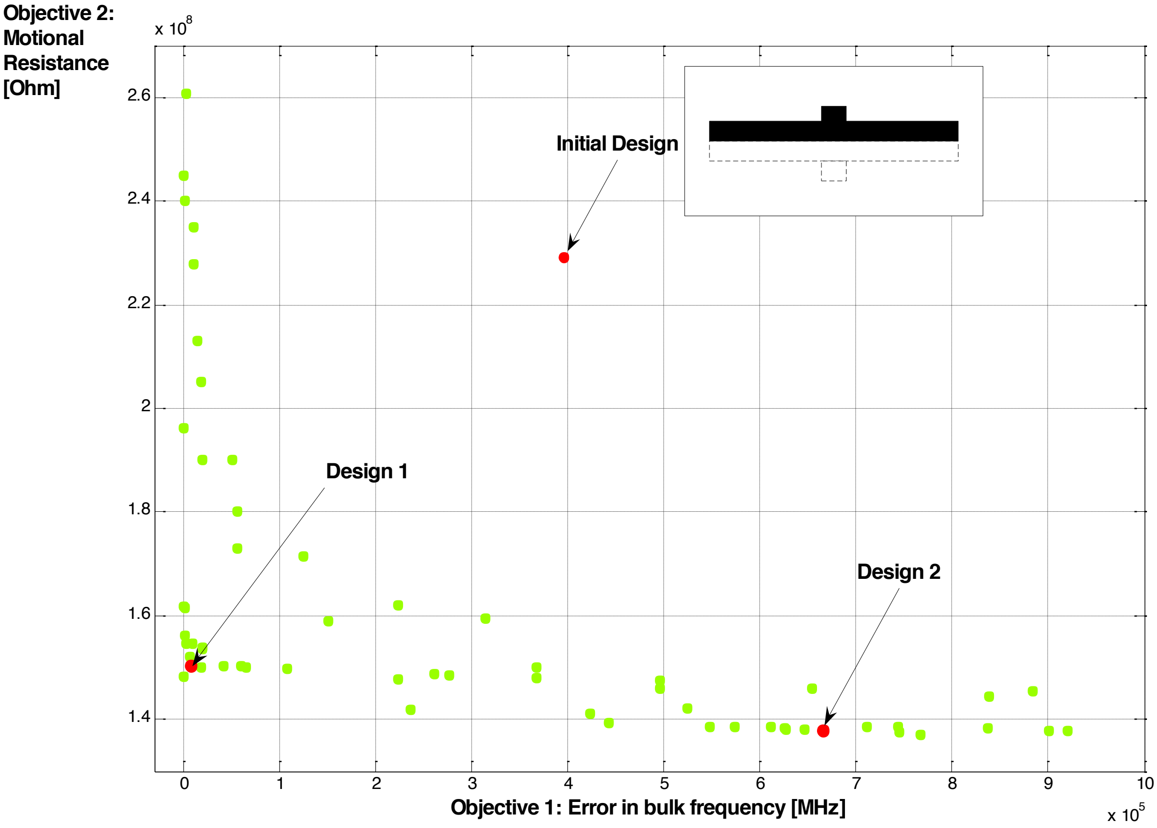
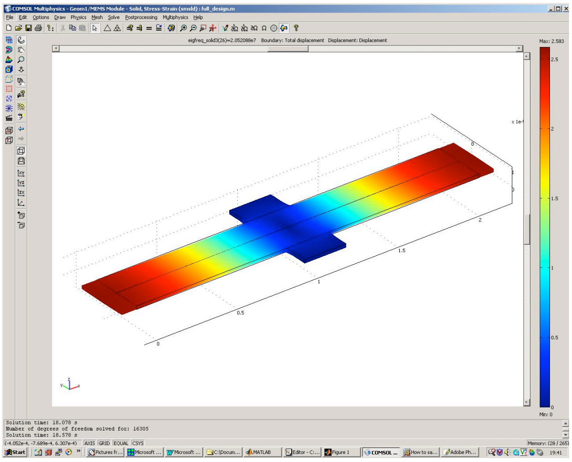


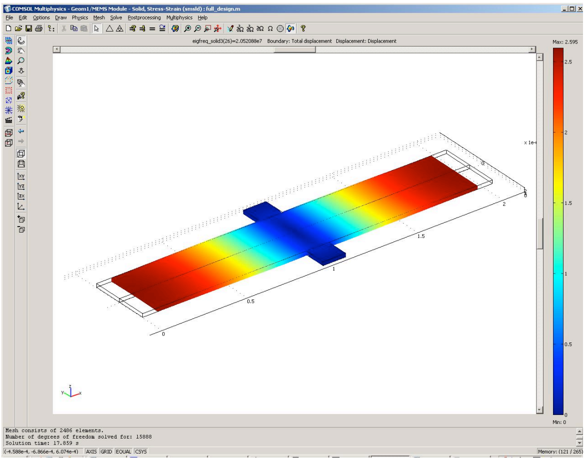
Figure 5.4: Five Archives of solutions resulted from 5 runs at 5000 iterations each and overlaid on the same axes.



```
Solution time: 18.078 s
Number of degrees of freedom solved for: 16305
Solution time: 18.578 s
```

```
(-4.052e-4, -7.689e-4, 6.307e-4)   AXIS  GRID  EQUAL  CSYS   Memory: (28 / 265)
```

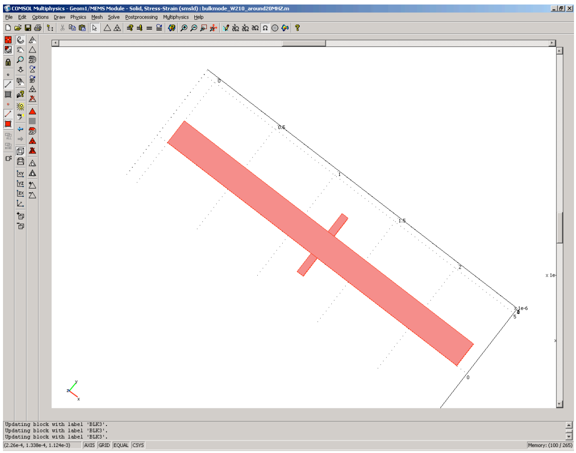
Figure 5.5: Design 1, Δf = 269, R_m = 150.



```
Mesh consists of 2486 elements.
Number of degrees of freedom solved for: 15888
Solution time: 17.859 s
```

```
(-4.588e-4, -6.866e-4, 6.074e-4)   AXIS  GRID  EQUAL  CSYS   Memory: (121 / 265)
```

Figure 5.6: Design 2, Δf = 665.859, R_m = 137.



```
Updating block with label 'BLK3'.
Updating block with label 'BLK3'.
Updating block with label 'BLK3'.
```

Figure 5.7: COMSOL simulation model.

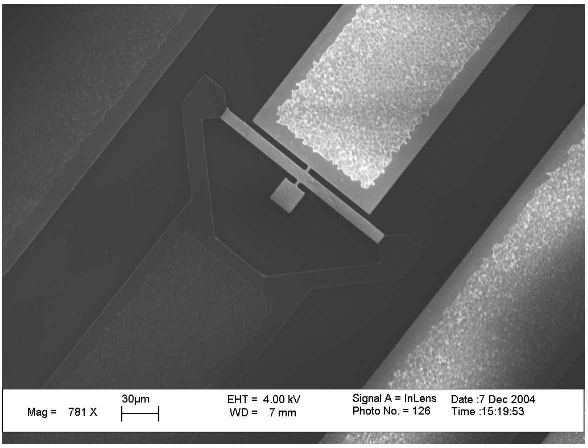


Figure 5.8: Manufactured solution [Yan, 2007].

Table V.II: Comparison of Solutions.

	W (µm)	**2L (µm)**	**w (µm)**	**l (µm)**	**H (µm)**	**Δf**	**R (MΩ)**
Solution1	*32.2*	*212.4*	*19.1*	*14.4*	*61*	*269*	*150*
Solution 2	*35*	*204.6*	*13*	*21.3*	*61*	*665859*	*137*
Hand Analysis	*21*	*210*	*10*	*20*	*61*	*396405*	*229*

5.1.5 Use and Capabilities of CNS-Burst in Size Optimisation Design Tasks

The example presented here has been a necessary step for automating the search of important design objectives, rather than a proof of the method's capabilities. Nevertheless, interesting results were shown, as well as a demonstration of the capacity of CNS-Burst to be adapted to different typology of design problems, including size optimisation ones.

Many practical design problems are size optimisation problems. For these tasks, and for more complex ones, designers can benefit from the use of a complete and adaptable generative tool like CNS-Burst. The possibility to use CNS-Burst as a size optimiser can be advantageous to solve groups of design tasks rather than for the optimisation of the size of a single structure. Much recent emphasis has been put into the study of new single shapes for silicon microresonators (squares, disks or rings rather than a single beam) and their correspondent resonant modes [Yan, 2007]. The scalability of design objectives with the main geometry variables of a resonant shape could be easily explored using size optimisation features of CNS-Burst instead of using lengthy manual approaches. Comparative studies on how design performance scales up with geometric variables for different resonant shapes would also be advantaged by the use of the CNS-Burst. The applications of this method would lead to a better insight into sensors and timing structures based on new MEMS resonators.

5.2 A Topology Optimisation Task

The goal of this section is to demonstrate the full potential of the method in generating novel design alternatives for complex topology optimisation tasks. The second goal is to prove that the innovation of solutions does not exclude accuracy of results. In the following example the efficiency of the method is demonstrated at its best, as well as its competitive advantage in respect to existing methods for MEMS synthesis. Some theoretical applications will also be presented here, in order to demonstrate CNS-Burst flexibility and its prospective use in a wide range of design applications.

5.2.1 The Sandwich Resonator Design

Sandwich resonators belong to a new generation of resonators used in reference oscillator applications that have raised interest in recent times [Yan, 2006]. The introduction of new typologies of silicon resonators comes from the necessity to address some technical challenges. One of these challenges is concerned with the equivalent motional resistance R_m. This key parameter determines the signal to noise ratio and power dissipation of a reference oscillator incorporating the MEMS resonator as a timing element [Yan, 2006]. The topology of the resonator and the coupling of mechanical and electrical domains have a strong influence on R_m. For this reason, sandwich resonators may better meet the scaling requirements associated with minimising motional resistance for reference oscillator applications. For a given resonant frequency the advantage of sandwich resonators, compared to FF-beam topology, is due to their large electrode surface, which can realise small motional resistance, large quality factor (small anchor loss) and low power consumption. Yan [Yan, 2006] shows that, in vacuum, for the same operating frequency, motional resistance of sandwich resonators is much smaller than those for of FF-beam resonators, although quality factor is slightly lower.

An example of sandwich resonator is shown in Figure 5.9. A sandwich resonator is so called because the resonant structure is sandwiched between two electrode beams. A typical structure is a regular one, where the sandwiched beams can be arrayed in parallel in the vertical direction (Figure 5.9). This type of structure is advantageous from the point of view of designers that perform design calculations by hand. The resonator is anchored at two edges of a central beam that runs through the length of the structure. The primary mode of interest is the bulk in-plane one. It involves in-phase longitudinal extension associated with the array beams. This mode can be driven using an electrostatic parallel-plate excitation mechanism where two electrodes are arranged parallel to the exterior beams (shaded blue in Figure 5.9). Figure 5.11 shows an optical micrograph of a sandwich resonator with outer electrodes electrically shorted together (resonator operated in a 1-port configuration).

The goal here is to synthesise a structure that resonates in the desired mode and for a desired range of frequencies. This task leaves sufficient flexibility to allow for the creation of any truss-like structures. Figure 5.10 shows the design area $A=H\text{x}L$ sandwiched between the electrode beams, where new topologies can be synthesised as an alternative to the typical arrayed ones.

A theoretical model for sandwich resonator behaviour has been developed and validated by Yan et al. [Yan, 2007] and will not be reported here. As the sandwich resonator is a periodic repeated structure of arrayed beams, in this model the sandwich resonator is considered as an array of subsystems constituted by single longitudinal FF-beam resonators, each of which can be simplified by a mass-mass-spring system. The analytical model was found to be a reasonable approximation towards estimating the resonant frequency [Yan, 2007]. For the sandwich resonator case study, the results obtained thought the application of CNS-Burst will be compared to hand analysis results based on this approximation. The equivalent electrical model for a resonator has been shown in Figure 5.2. As seen for the FF-beam resonator, motional resistance R_m, motional capacitance C_m, motional inductance L_m are defined by (*5.2*), (*5.3*), (*5.4*).

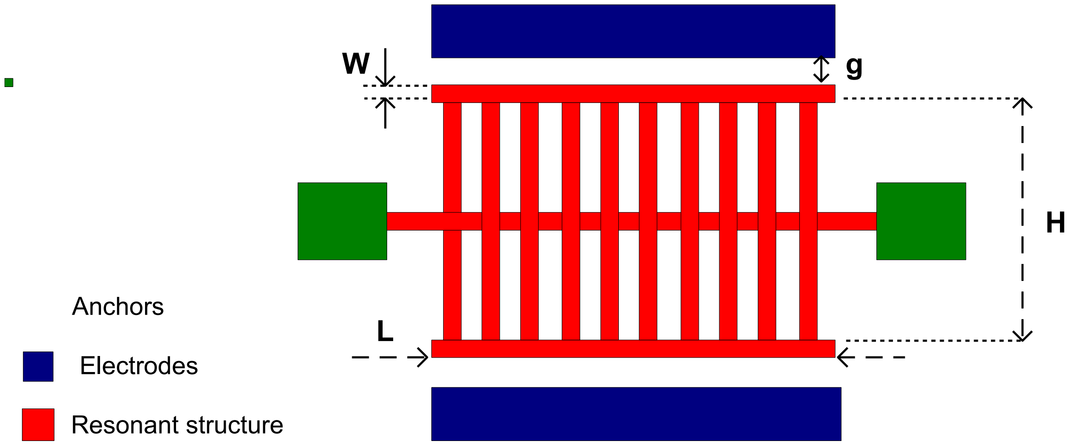


Figure 5.9. Sandwich resonator.

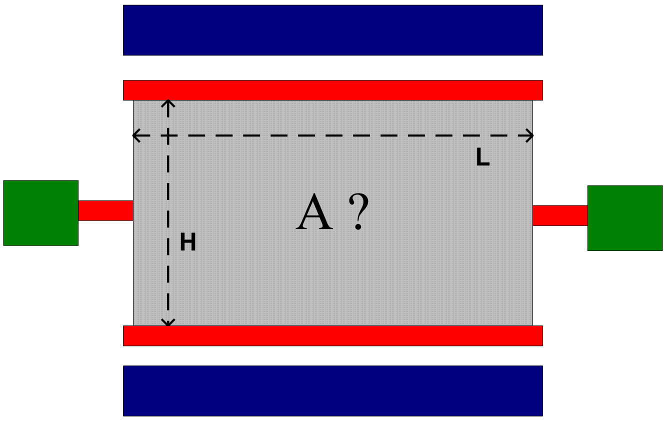


Figure 5.10. Sandwich resonator: the topology optimisation task.

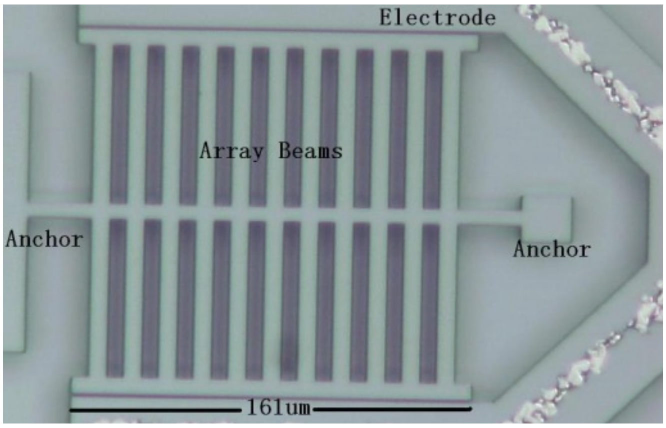


Figure 5.11: Optical micrograph of the resonator with outer electrodes electrically shorted together. The resonator is operated in a 1-poert configuration [Yan, 2007].

5.2.1.1 Key Design Performance Criteria

The key parameters for resonators design have been introduced in 5.1.1.1. Definitions, formulas and considerations expressed for FF-beam resonator are also valid for sandwich resonator.

The aim of this case study is again to design resonators vibrating in a desired frequency range. Resonant frequency f is the most important parameter to take into account in this search. The order of magnitude for frequencies considered here is again of *MHz*.

As mentioned before, the quality factor Q is a measure of the energy dissipated per cycle in the resonator. Here again it is assumed that, as for FF-beam resonators, anchor loss mechanisms are the dominant component of the quality factor. For sandwich resonators, Q can be as high as *13,000* in vacuum.

As for the motional resistance, it has been mentioned above that R_m is the parameter that justifies the recent interest in sandwich resonators. Motional resistance for sandwich resonators is lower than for other silicon microresonators. R_m is the equivalent electrical resistance presented at resonance by the resonator and can be deducted from the equivalent electrical circuit (Figure 5.15). Motional resistance of sandwich resonators can be deducted from the comparison with that of a single FF-beam resonator. R_m is equal to the square ratio of W_a and W_e by a factor n, where W_a and W_e are the width of the array beams and the width of the additional exterior layer (i.e. double of the half distance between axes of two consecutive beams), and n is the number off beam members of the sandwich:

$$\frac{R_{m-longitudinal}}{R_{m-sandwich}} = \frac{\dfrac{\pi\sqrt{E\rho}g^4}{8\varepsilon_0 Q V_{DC} T}\dfrac{1}{W_e}}{\dfrac{\pi\sqrt{E\rho}g^4}{8\varepsilon_0 Q V_{DC} T}\dfrac{W_e}{nW_a^2}} = \frac{nW_a^2}{W_e^2} \qquad (5.14)$$

where E is the Young Modulus, ρ is the density, ε_0 is the dielectric constant, Q is the quality factor, V_{DC} in the operational DC voltage, T the thickness of the structure, g the gap between resonator and electrode.

As a large R_m indicates an increase in the ratio between the energy dissipated and the energy stored in the resonator, a small R_m value is desirable for a resonant system. As mentioned in 5.1.1.1, the resonator topology and the coupling of the mechanical and electrical domains have a strong influence on the motional resistance. The advantage of the sandwich topology with respect to the FF beam is precisely the increased coupling between the mechanical and electrical domains due to the increased actuation area. For the same range of operational frequencies (order of *MHz*), a sandwich resonator's R_m value is usually in the order of magnitude of *KΩ*, while a FF beam resonator's R_m value is in the order of $10^2\ M\Omega$.

5.2.2 Optimisation Model with Three Design Objectives

This section reports the optimisation model used for the application of CNS-Burst to the sandwich resonator topology optimisation task. While the quality factor *Q* was considered a constant in the FF-beam size optimisation task, here *Q* represents one of the objectives of the search, bringing the number of design of objectives to three. The design objectives considered are:

- A target operational frequency *f* of *25 MHz* (a constraint-satisfaction problem formulated as a soft constraint)
- A minimal motional resistance R_m (formulated as a minimisation problem)
- A maximum quality factor *Q* (formulated as a minimisation).

The optimisation model for this design task is:

$$\min\left\{|\Delta f|, R_m, \frac{1}{Q}\right\}, \tag{5.15}$$

$$\text{S.t.} \quad 0 \le x_j \le 161\mu m, 0 \le y_j \le 155\mu m$$

Design Constraints (see Table V.III)

where $\Delta f = (f_0 - f)$, *i* is the number of beams, *j* is the number of nodes, (x_j, y_j) are the coordinates of a node. The minimum width (w_i) of the beam elements equal to 1µm is due to fabrication limitations. The boundary limits of the design area $A=H\text{x}L$ (Figure 5.10) is formulated as a constraint for the position of the nodes (x_j, y_j) to be in the design area.

The design variables are the length and width of the beam primitives (w_i, l_i) forming the design, the connectivity of the beam primitives, the position of the nodes, as well as the number of beams used (i). Length and width of the electrode beams, height of the resonator, DC voltage, capacitive gap and thickness of the resonators are considered parameters and kept constant in the calculations ($W = 14\mu m$, $L = 161\mu m$, $H = 155\mu m$, $V_{DC} = 80$ V, $g = 1\mu m$ and $T = 20\mu m$ respectively).

Table V.III: Design constraints for the beam primitive.

Specific to primitives	Conflicts with other Primitives	Node connectivity
▪ Maximum length = 161μm ▪ Maximum width = 40 μm ▪ Minimum length = 5 μm ▪ Minimum width = 1 μm ▪ Thickness = 20 μm ▪ Number of instances N≤10	▪ No overlap allowed ▪ Minimum separations between any two of non-connected beams = 1μm ▪ Angle between the beams θ ≥ 15°	▪ Can share normal (floating or irremovable) nodes with maximum 10 other beams ▪ If connected to anchor type node, cannot share that node with any other primitive.

5.2.3 Implementation of the Objectives and Details

Further complexity has been introduced with this case study, both in the higher number of design objectives considered as well as in the difficulty of the formulation of the objectives themselves. The goal of generating solutions as alternatives to the one used in practice represented a challenge. The quality factor has been formulated as a design objective, showing how the calculation of this complex design objective can be introduced in generative methods. Designs also required quarter symmetry in order for the device to vibrate according to requirements (as opposed to half symmetry of the FF-beam resonator), introducing further computational complexity in the generation of design alternatives. Generated designs included on average a high number of primitives and nodes that, in terms of CNS design representation, are translated in an expansion of dimensions of the representation matrixes and consequent increase in computational time. The large scale of design tasks also translated in an increase of memory usage connected to FEM calculations. Computational time and memory usage dimensions gave a measure of some difficulties in the development of a generative tool for MEMS and some directions on how these tools should evolve. Particular attention should be put in the choice of mathematical models used by the solver, to take care not only of the accuracy of the solutions but also of the potential difficulties above mentioned. The commercial package used for analysis and simulation of design solutions here is the COMSOL-MEMS Module. Five searches at *5000* iterations each were run using an Athlon XP *2.8GHz*, *2GB* RAM machine, for a total time of *1h/100* iterations circa. A new module was also introduced for the search of the sandwich-bulk mode. The module acts as a mode/frequency selector, as explained in the following section.

Automated search of sandwich-bulk mode

The objective of this case study is to find a resonator with the required resonant frequency and resonant mode. The resonant mode in question is the sandwich-bulk mode, as the structure vibrate in the direction traversal to the axis of the resonator (Figure 5.12). The vibration must also be in plane. Generating designs having the required frequency as well as the required vibration mode is again a new and unexplored challenge in automated synthesis of MEMS.

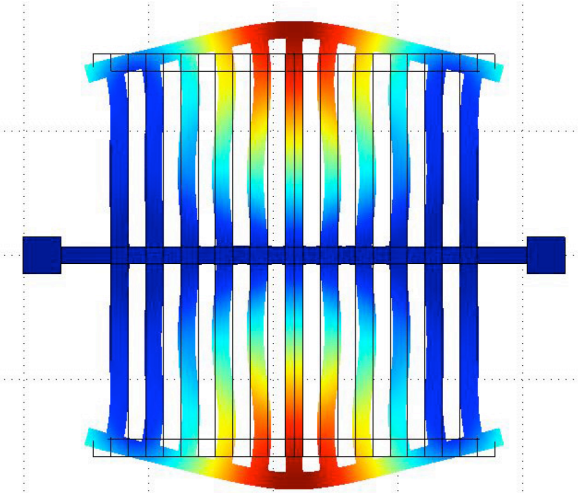


Figure 5.12: Sandwich-bulk mode.

The search of the bulk mode resonator has been automated through a new module that, exactly like the module introduced for the search of the longitudinal mode (5.1.3), first finds the right resonant mode for the generated design, then verifies that the resonant frequency correspondent to that mode is in the desired range. The algorithm performed by this module is reported in Figure 5.13.

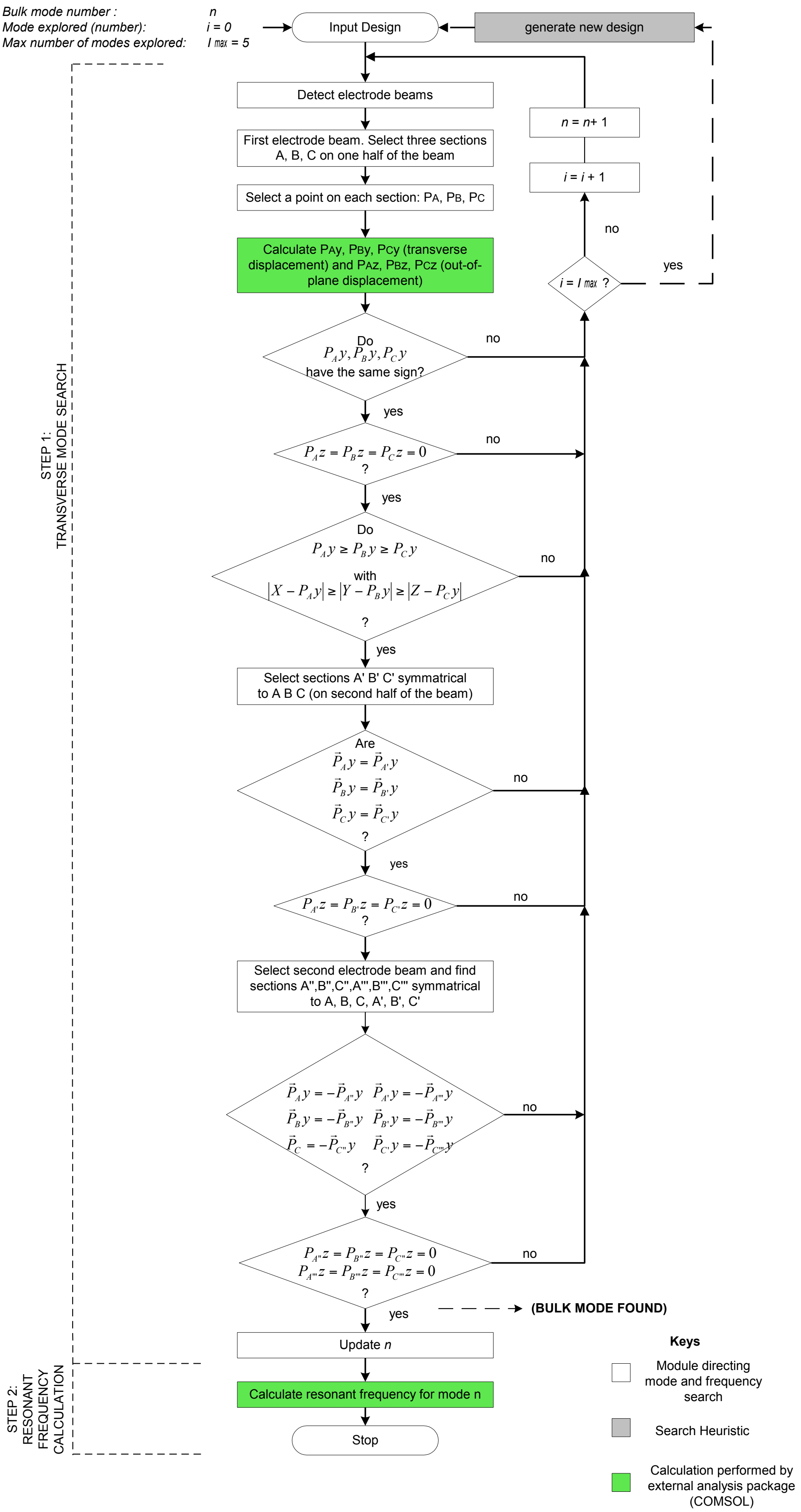


Figure 5.13: Sandwich-Bulk Mode Algorithm Search.

The search exploits the symmetry of the design and, in particular, the quarter-symmetry of electrode beams. For the bulk mode, symmetric points on the two electrode beams move in opposite phase but have equal absolute displacement. While symmetric points on the two halves of a single electrode beam displace in equal direction and value.

Given a generated design, the first step of the search consists in localising the electrode beams. Each design, according to the CNS Design representation, in represented by a set of primitives and nodes defining their extremities. Among the primitives, the electrode beams have the highest or lowest y-coordinate. This is due to the above-mentioned constraint for each node to fall inside the design area (*HxL* in Figure 5.10). The second step of the algorithm consists in examining one of the two electrode beams, and extracting the values of the displacements matrix for three sampling points (P_A, P_B, P_C) distributed on three cross-sections of one half of the electrode beam (*A*, *B*, *C*) (Figure 5.14). In order for the mode to be the bulk one, the transverse component of the displacement (*y*) for any point of the cross-sections must be degrading from the centre to the extremity of the beam
and must have the same sign. Also, there is no out-of-plane (*z*-displacement) for points belonging to the electrode beams. Finally, points $P_{A'}, P_{B'}, P_{C'}$, localised on the other half of the electrode beam and belonging to sections *A'*, *B'*, *C'* (symmetric to *A*, *B*, *C* with respect to the central points of the electrode beam $P(x,y)$), have y-displacements equal in absolute value to those of P_A, P_B, P_C.

A check on the second electrode beam is then executed. The six points considered on the first electrode beam ($P_A, P_B, P_C, P_{A'}, P_{B'}, P_{C'}$) and their symmetric points localised on the second electrode ($P_{A''}, P_{B''}, P_{C''}, P_{A'''}, P_{B'''}, P_{C'''}$) must have *y*-displacement equal in absolute value but opposite in sign. Points on the second electrode also do not have out-of-plane displacement. Once all these checks have been executed, if the resonant mode is confirmed to be the bulk mode, the algorithm will calculate the resonant frequency for that mode, in order to check that it is in the desired range.
As seen in 5.1.3, this verification process can be quite lengthy and computationally expensive for each design, especially if the modes are calculated starting from the first one, until the desired one is found. Again the process is automated here avoiding calculation of all modes and the search starts instead from a mode number (*n* in Figure 5.13) that is an initial guess for the bulk mode number. The procedure is identical to the one described in

5.1.3 for the bulk mode search as well as the motivations for exploring modes in the range ($n \pm i$), with i being a counter ranging from one to $I_{MAX} = 5$.

This procedure proved to be reliable as well as representing an advantageous tool for designers to analyse resonant frequencies and modes of many solutions in a short amount of time. Automated search for bulk modes also represents for designers a great advantage never introduced by any other synthesis tool for MEMS and a step forward in the implementation of useful design tools for MEMS synthesis.

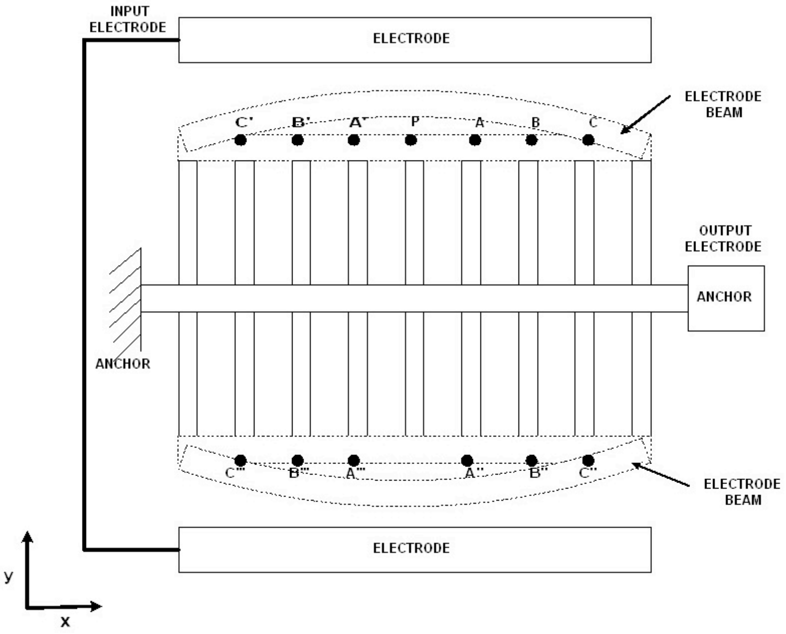


Figure 5.14: Points and cross-sections used for the detection of the bulk mode.

Automated Search of Quality Factor

As seen in (*5.7*) the quality factor can be expressed as the ratio between the maximum vibration energy stored by the system per cycle and the energy dissipated per cycle of vibration. In sandwich resonators the dissipation mechanisms are influenced by many factors (5.1.1), but anchor loss will be considered here as the dominant factor influencing Q.

The method used here to automate the calculation of Q is based on energetic consideration [Lee, 2008]. Starting from the assumption that the energy lost per cycle due to anchor loss is proportional to the total strain energy present in the anchor, Q can be expressed as follows:

$$Q = \alpha \frac{E_{total}}{E_{anchor}} \qquad (5.16)$$

where E_{total} represents the total strain energy of the entire structure (resonant parts plus anchors) for a certain mode shape and E_{anchor} is the strain energy present in the anchors for the same mode. Q is proportional to the ratio between the two strain energies by a factor α. Resonant modes with high ratio also present high Q. Strain energy in any parts of the structure can be easily calculated using COMSOL-MEMS analysis package. This method is not perfectly accurate, as it requires the estimate of the proportionality factor α. Nevertheless, the method gives a good measure of Q in terms of order of magnitude and is effective in the comparison and ranking of design alternatives. The parameter α is determined through measurements of Q obtained through experimental results [Yan, 2006] with values of the strain energies ratio obtained using COMSOL. Although a high Q is desirable for microresonators, the sandwich topology does not guarantee values of Q as high as for FF-beam resonators. Nevertheless, results presented in 5.2.2 show that Q's in the order of 10^4 are achievable.

The automated calculation of Q represents another advantage in using CNS-Burst for designers, especially considering that Q calculation is never straightforward and that no sort of automation has ever been implemented for this purpose. Many methodologies and tools have been in fact implemented to help designers calculating Q, but their development has been limited by the difficulty of dealing with all the factors influencing Q. Some of these recently released tools have demonstrated to be very accurate. Bindel [Bindel, 2005] uses the perfectly matched layer technique to implement the HiQLab tool, now freely available. Steeneken also makes use of a matched boundary layer [Steeneken, 2007] for the calculation of Q due to anchor losses. While Duwel et al. present a method to calculate Q as a result of TED using eigenfrequency analysis [Duwel, 2006].

The choice of integrating the strain energy method into CNS-Burst instead of other more sophisticated and accurate tools derives from practical reasons, as calculation of strain energy is particularly straightforward in COMSOL. Nevertheless, any other tools could have been integrated into CNS-Burst. HiQLab has a MATLAB command interface that would make it particularly suitable for CNS-Burst. While the matched boundary layer technique used by Steeneken, developed using COMSOL functionalities and analysis, could be easily integrated in the search. This example shows once again how external packages are not directly related to the capabilities of the CNS-Burst and how the great advantage of the method is in the possibility to be easily integrated with external tools according to the needs that direct the search.

Calculation of the Effective Spring Constant

As seen in (*5.2*), R_m depends on Q and on the effective spring constant k_{eff}. k_{eff} depends on the topology of the designs, and this makes it particularly difficult to calculate using formulas based on theory of vibration for beams, as designs may vary and assume non-standard topologies [Yan, 2006]. For each case, k_{eff} has been calculated instead using energetic considerations. Hooke's Law defines the spring constant as proportional to the quotient of the maximum elastic potential energy divided by the squared deformation at a given point and direction of the vibratory system:

$$\frac{1}{2}k_{eff}x^2 = E_{potential} \qquad (5.17)$$

The energy stored in the deformed structure at resonance (Figure 5.12) can be calculated as the integral of a traction force applied to the structure that causes the combined extension of the sandwich. Calculating k_{eff} through this method implies the application of a force in points of the electrode beams, in the way that the deformation obtained reproduces the mode at bulk resonant frequency. The calculation of the energy stored and the structure's displacement in any point is easily obtained with COMSOL.

Automated Search of Motional Resistance

From the definition given in (5.2), the formulation used for the search of motional resistance in sandwich resonators can be written as follows:

$$R_m = \frac{k_{eff}}{2\pi f Q \eta^2} \tag{5.18}$$

where k_{eff} is effective spring stiffness, and f is the bulk frequency. The transduction coefficient η has been defined in (*5.5*) and for a the sandwich resonator, can be formulated as follow:

$$\eta \cong 2V_{DC}\varepsilon_0 \frac{TL}{g^2} \tag{5.19}$$

The automated search of motional resistance in synthesis of MEMS is a novel objective introduced by this work, made possible by the automated calculation of the spring stiffness of the system. R_m depends on geometrical factor, varying inversely with the capacitive area sensed. For this reason R_m tends to be some orders of magnitude lower for sandwich resonators than for FF-beam resonators.

5.2.4 Research of Standard Search with Three Design Objectives

This section reports results of CNS-Burst application to the sandwich resonator topology optimisation task. The initial design (Figure 5.11) used to start the search is the one designed and experimentally tested by Yan [Yan, 2006] using traditional calculation methods. This design has a typical regular structure formed by eleven parallel beams, resonant frequency close to the objective target (f_0 = *24.38 MHz*) motional resistance around R_m = *800 KΩ*, and quality factor *Q=12,980* circa. The initial design reproduced with the CNS design representation resulted rather complex, being only a quarter of its topology formed by nineteen beam primitives and fifteen nodes. This design proved to be difficult to modify, due to its orthogonal structure and to the absence of nodes in the design area, but rather concentrated on the horizontal beams of the structure. This initial design was then modified dividing each of the vertical beams in two connected beams (Figure 5.15). The final structure resulted in twenty-one nodes and twenty-five primitives only for a quarter of the design (Figure 5.15). This structure could be easily modified during the search. The nodes of the frame of the structure (i.e. situated on the electrode beams) were defined as floating nodes, but with the additional feature of being irremovable. Also, their coordinates could not be changed. The Pareto archive resulted unchanged immediately after the first iterations, meaning that the algorithm had converged rather quickly to the final solutions and before the maximum number of iterations set was reached. Designs

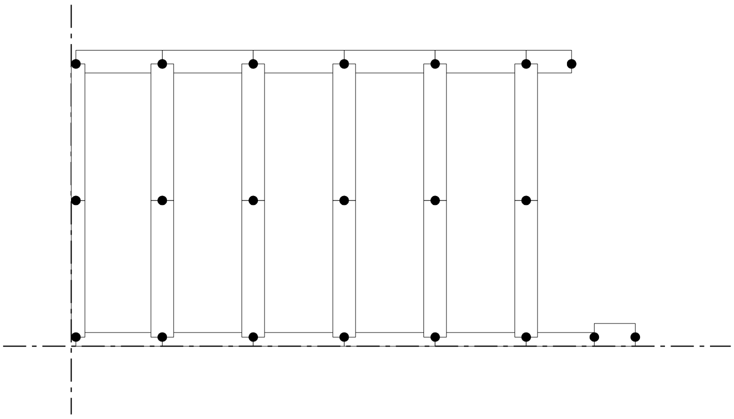


Figure 5.15: Initial Design (quarter).

observed after few iterations demonstrated to have better performance than the initial solution.

Figure 5.16 shows an archive of solutions obtained from five optimisation runs at *5000* iterations each. Each run, resulting in an archive of non-dominated solutions distributed on a Pareto front, is overlaid on the same axes. The three coordinates of the plot represent the objectives of the search (the error in resonant frequency, the motional resistance and the quality factor: in the x, y and z direction respectively). The design archive presents a variety of solutions to the task, also illustrating performance trade-offs. The results show that *27%* of the solutions kept in the design archives have an error in resonant frequency Δf < 1%, and that 69% of the solutions have smaller motional resistance R_m than the initial design. Design 1-3 in Figure 5.16 show some interesting solutions obtained. Design 3 shows a $\Delta f = 0.055\%$ of the target frequency. For design 2, $R_m=455K\Omega$ (decrement of *43%* from the initial value of R_m). A comparison of these designs with the initial design obtained with hand analysis is shown in Table V.IV. This preliminary search performed using simple a generate-and-test algorithm produce solutions comparable with the initial design, if not better for two of the design objectives. The solutions generated showed improved performance, as well as some interesting and innovative topologies. Figure 5.17 presents some of the original structures obtained in the search.

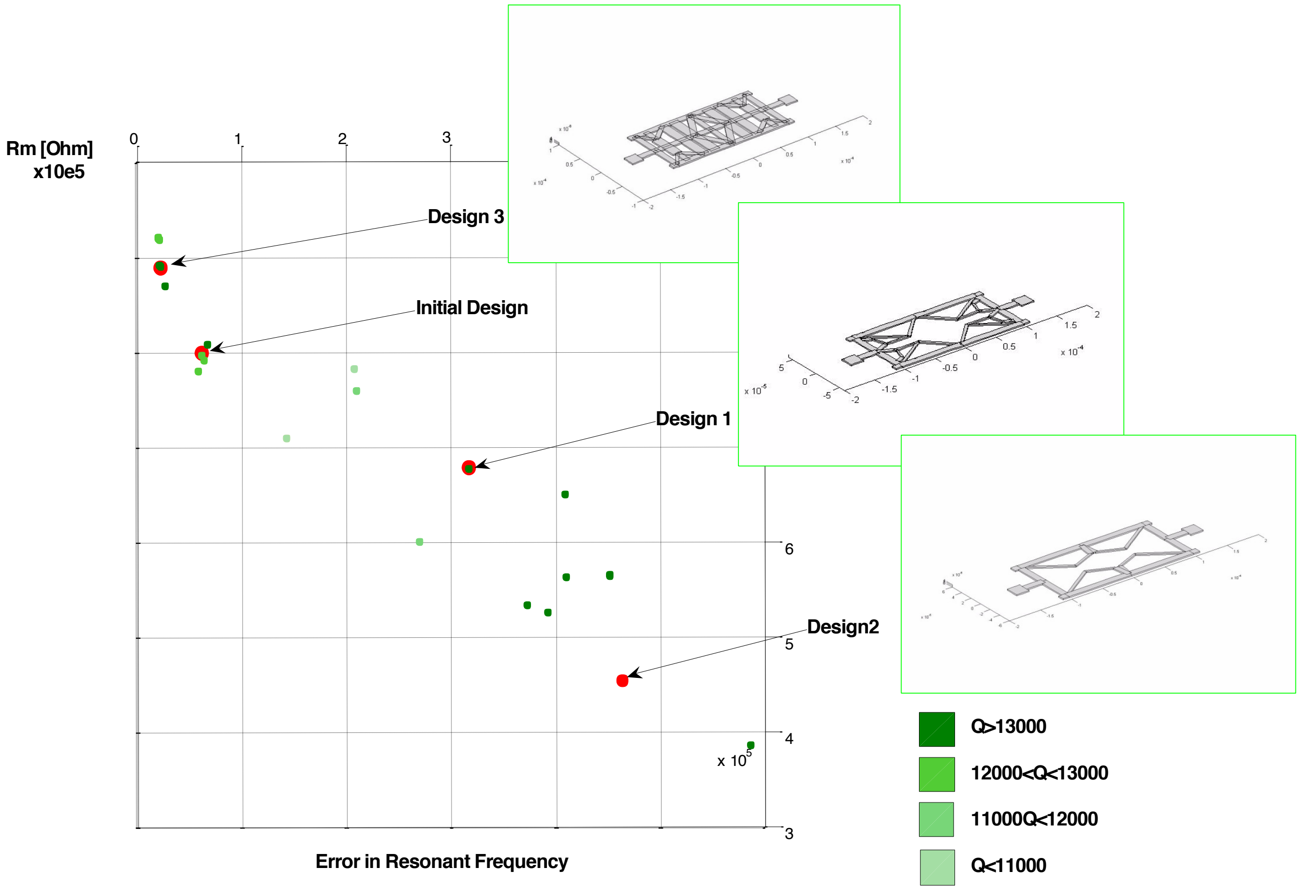


Figure 5.16: Archive of Solutions that resulted from five optimisation runs at 5000 iterations each (Objectives values for Designs 1-3 specified in Table V.IV).

Table V.IV. Comparison of Solutions.

	Δf	R (KΩ)	1/Q
Solution1	*3.178*	*679*	$7.24x10^{-5}$
Solution 2	*4.645*	*455*	$6.63x10^{-5}$
Solution 3	*0.22*	*889*	$6.63x10^{-5}$
Hand Analysis (Initial Design)	*0.62*	*800*	$7.68x10^{-5}$

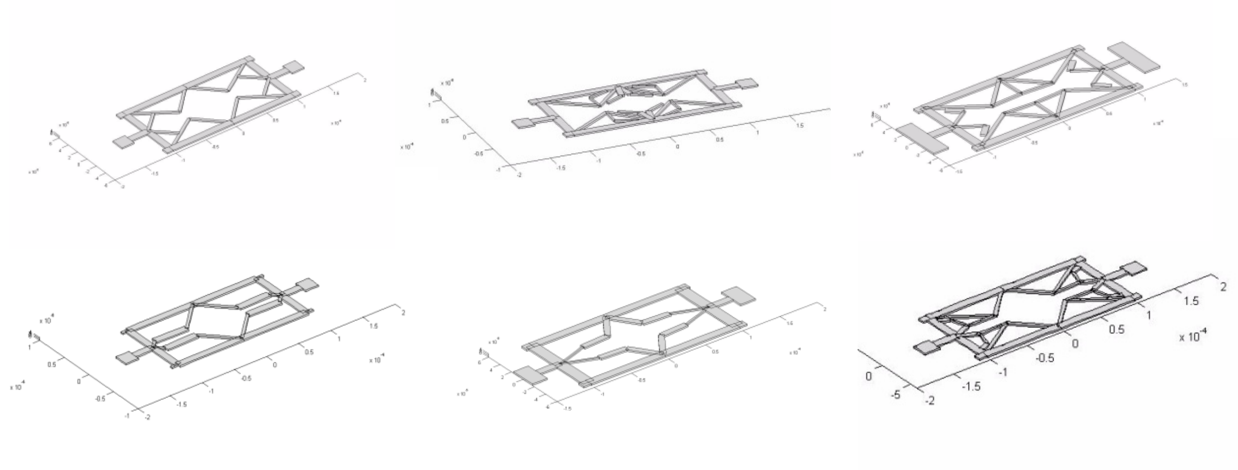


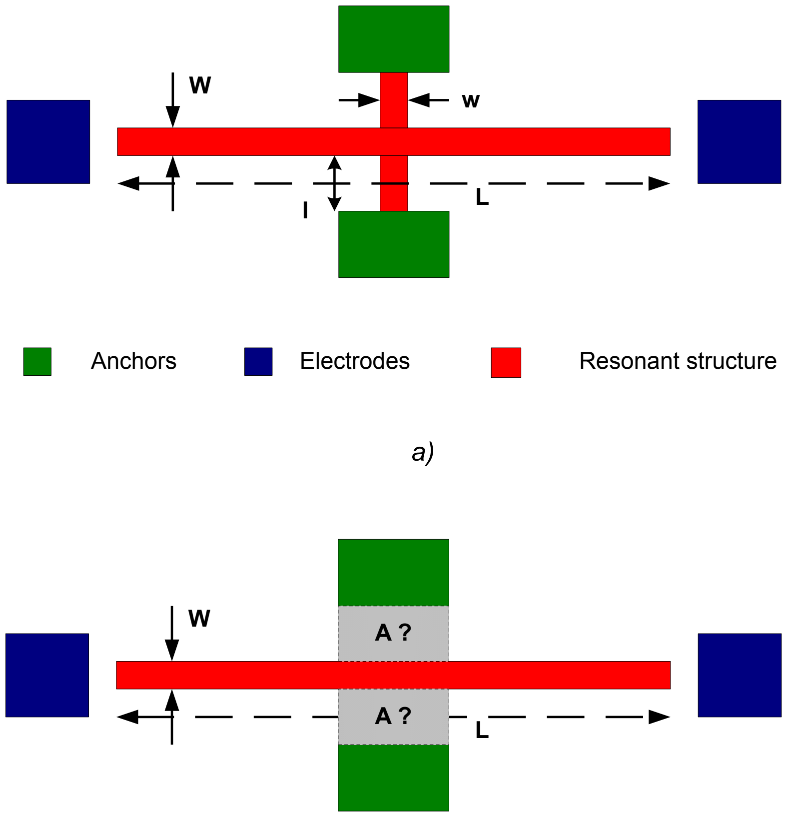
Figure 5.17: Variety of Sandwich resonator topologies obtained with CNS-Burst.

5.2.5 Another Topology Optimisation Case Study: The Anchors Loss Problem

The quality factor varies with the geometry of the resonator in a non-predictable way. Particularly interesting is the design of the resonator's supports, as anchor loss can limit Q substantially. The influence of support design on Q has been previously analysed in literature. Lee (Lee, 2008) analyses the effect of T-shaped supports on Q for disk and square resonators. For this purpose, Lee uses FEA simulation to calculate the relative strain energy of the anchors and bulk structure. Results of this work show a direct correlation of the supports geometry on extensional resonant modes, while no effect is detected for wineglass modes.

The following example demonstrates how CNS-Burst can be useful in this particular design problem, offering the possibility to explore how the design space for supports is related to Q, and also indicating directions for designing support topologies that positively influence Q.

This simple example uses the FF-beam resonator case seen in 5.1.2. Figure 5.18a shows the shape of a common resonator suspended by beam-supports, while Figure 5.18b highlights the design space (A) for different supports' topologies. The goal here is to find these topologies using CNS-Burst.



a)

b)

Figure 5.18: a) FF-beam resonator: a typical configuration; b) A: Design space for supports.

The optimisation problem is once again to design a resonator with maximum Q for a certain frequency and mode (f_0 = *20MHz* and longitudinal mode). The idea is find optimal designs modifying the supports' topology but without modifying the resonant structure. The optimisation model for this design task can be formulated as follow:

$$\min\{|\Delta f|, Q^{-1}\}, \qquad (5.20)$$

S.t. *Design Constraints (see Table V.V)*

where $\Delta f = (f_0 - f)$. The design variables are the number of beams used (i), the length and width of the beam primitives (w_i, l_i) forming the design, the connectivity of the beam primitives, as well as the position of the nodes. The minimum width (w_i) of the beam elements equal to 1μm is due to fabrication limitations. *W*, *L* and *T* (thickness of the resonator) are considered fixed parameters. Figure 5.19 shows the initial design used for the search (designs were generated by half symmetry).

Table V.V: Design constraints for the beam primitive.

Specific to primitives	Conflicts with other Primitives	Node connectivity
▪ Maximum length = 50 μm ▪ Maximum width = 30 μm ▪ Minimum length = 2 μm ▪ Minimum width = 1 μm ▪ Thickness = 20 μm ▪ Number of instances N≤10	▪ No overlap allowed ▪ Minimum separations between any two of non-connected beams = 1μm ▪ Angle between the beams θ ≥ 15°	▪ Can share normal floating nodes with maximum 10 other beams ▪ If connected to anchor type node, cannot share that node with any other primitive.

Q is calculated again using the following:

$$Q = \alpha \frac{E_{total}}{E_{anchor}} \tag{5.21}$$

where E_{total} and E_{anchor} represent the strain energy in the resonant structure and in the supports respectively. Figure 5.20 shows an archive of solutions obtained from a single optimisation run at *5000* iterations.

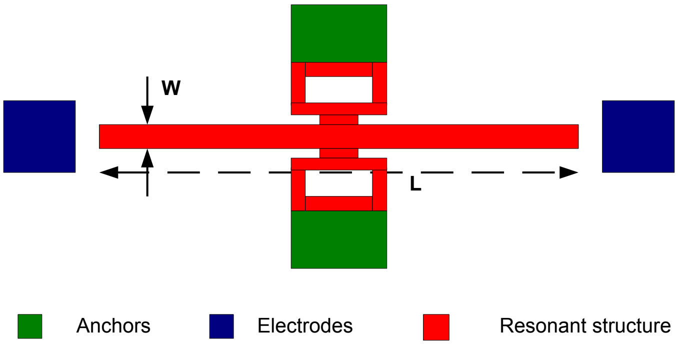


Figure 5.19: Initial Design.

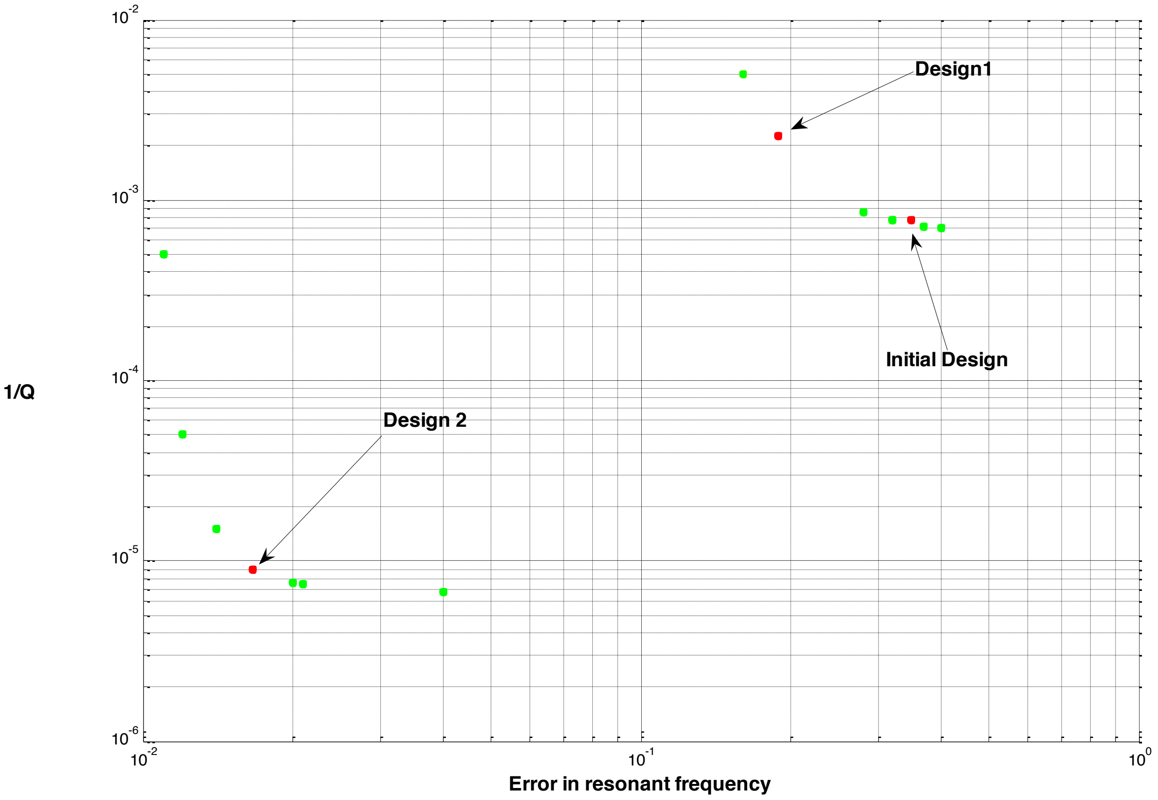
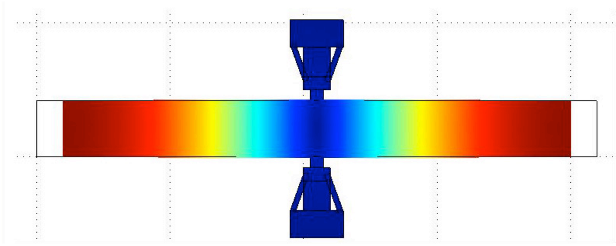
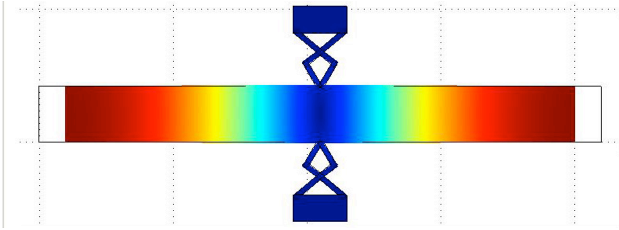


Figure 5.20: Archive of solutions that resulted from two runs of the search at 5000 iterations each (logarithmic scale).



a)



b)

Figure 5.21: a) Design solution 1; b) Design solution 2.

Results of this single run confirmed some interesting trends known to expert designers but useful to less knowledgeable designers looking for alternative supports that allow for higher *Q*. Supports topologies that present elements with higher elastic response (spring elements, as defined by Lee), also present a reduction of coupling between resonant structure and supports, and a reduction of support loss (i.e. higher *Q*). An example of these topologies is shown in Figure 5.21a. Design solutions that present supports' topologies with these characteristics are situated in the lower part of Figure 5.20). Topologies presenting higher stiffness are usually formed by wider elements, showing how these elements are connected to a lower *Q*. The second interesting trend is that the reduction of the contact surface between resonant structure and supports reduces *Q*. No consistent effects of support topology are noticed on the value of the resonant frequency *f*.

5.3 Conclusions

From the results presented in this chapter, the following considerations can be drawn:

- As anticipated in previous chapters, the flexibility of CNS-Burst has being demonstrated through the modelling and solution of many different MEMS design cases.
- CNS-Burst is a complete package for design and analysis of MEMS, being potentially applicable to any type of optimisation task. Its contribution to designers work can range from simple size optimisation to generation of innovative design solution.
- The scalability of CNS-Burst has been demonstrated: the method has been extended to handle design tasks presenting growing difficulties in the number of objectives considered, in the complexity of objectives or in the number of elements forming the design.
- Comparison with hand analysis results shows that complexity of design objectives does not necessarily imply lack of accuracy and precision in the results. Early stage designs of micro-devices can be subsequently improved through calculation of second order effects, in order to analyse fabrication issues and finally be manufactured. First stage designs obtained with CNS-Burst and obtained by hand showed same level of accuracy.

- In many cases designs generated by CNS-Burst rivalled those obtained by hands or were at least comparable in terms of one design objective. The method was able to compensate computational efforts with a drastic cut of the design times.
- In any of the presented cases, the method provided the possibility to choose preferred solutions in archives of Pareto optimal designs. Moreover, the advantage of having multiple solutions with similar performance trade-offs offers the possibility to test many designs at the fabrication stage, increasing the chance to overcome manufacturing difficulties.
- Novel objectives, never considered in other MEMS synthesis and optimisation tools before, were successfully implemented (quality factor and mode shapes among them). These objectives represent highly desirable performances to optimise for designers. The methods used to calculate these objectives are general and applicable to any design cases presenting the same objectives. Many MEMS generative tools found in literature pursued simplistic design objectives. When compared to these methods, CNS-Burst demonstrated its ability to tackle real, practical and very common design objectives, confirming its utility for designers.
- Topology design cases may require a considerable amount of computational time, which is a potential obstacle to the development and use of generative methods in complex design tasks. The results obtained show that a trade-off between computational time and successful outcome of the search is possible in automated synthesis.
- Finally, the method showed its utility not only in the generation of new designs, but also in the possibility to explore how design topologies affect design parameters. The relation between design layout and design parameters is often unknown, especially for novel designs. The role of CNS-Burst in behavioural analysis and in understanding important design criteria relationships is as important for designers as in proposing new design alternatives.

CHAPTER 6 – POTENTIAL USES OF CNS-BURST AND FUTURE DEVELOPMENTS

The results presented in the previous chapter confirmed the efficacy of CNS-Burst, its capabilities to scale-up in difficult design tasks, as well as its adaptability to different and complex applications. However, CNS-Burst has scope for improvement both in its capacity to be employed in a wide range of design tasks, and in the possibility to add on its 'high-level' characteristics, as listed in 1.2. This chapter is a discussion of potential uses and applications of the method and an anticipation of its future development.

Chapter 5 showed how the method can be successfully applied to topology and size optimisation tasks with good results. The capacity of the method to solve any search requirements is completed here with the introduction of some conceptual applications to shape optimisation tasks.

Finally, among the desired characteristics for generative tools listed in 1.2 is their ability to be adapted to different design domains. The last section introduces the possibility to include this feature in the method.

6.1 Use and Capabilities of CNS-Burst in Topology Optimisation Design Tasks

In the previous chapter two topology optimisation design tasks have been solved using CNS-Burst. The method has the potentials to be applied in a big variety of similar tasks. Also, CNS-Burst capabilities in solving topology optimisation problems represent the most innovative contribution to designers' activities. In the following sections further examples will be given on how CNS-Burst can be applied to practical MEMS design tasks involving topology optimisation. Although these tasks are only outlined and not solved, these cases are introduced to highlight the method's capacity to be adapted to different design problems, as well as its efficacy in exploring connections between design performance criteria and topology. The definition of a new design case requires setting up of new design objectives and a CNS representation for the initial design. While the definition of the initial design is straightforward, requiring the compilation of the geometrical matrices only (3.6.2), the definition of the design objectives may be complex and lengthy, as seen in the case of the bulk mode and transverse mode resonant frequencies.

6.1.1 Sandwich Resonator Case Study: Further Applications of CNS-Burst

Current work on sandwich resonators is focused on understanding the trade-off involving the number of beams in the structure, the mechanical coupling of the structure and the underlying mechanisms for mechanical losses in the structure [Yan, 2006]. The search of connections between number of beams and losses could be directed introducing some element of sophistication in the basic search performed by CNS-Burst. The number of parts constituting the structure could be, for example, evaluated and linked at the value of losses for each design solution. A parallel and automated search could link part numbers with degree of objective satisfaction (losses reduction in this case) and direct the search through those structures that have successful parts number. The number of beams could also be linked to the scaling of frequency with different topologies (as anticipated in the case of the FF-beam resonator). Finally, different topologies for sandwich resonators could also use shapes other than beams, for example plates or disks. Figure 6.1 shows some examples, although these structures are purely conceptual.

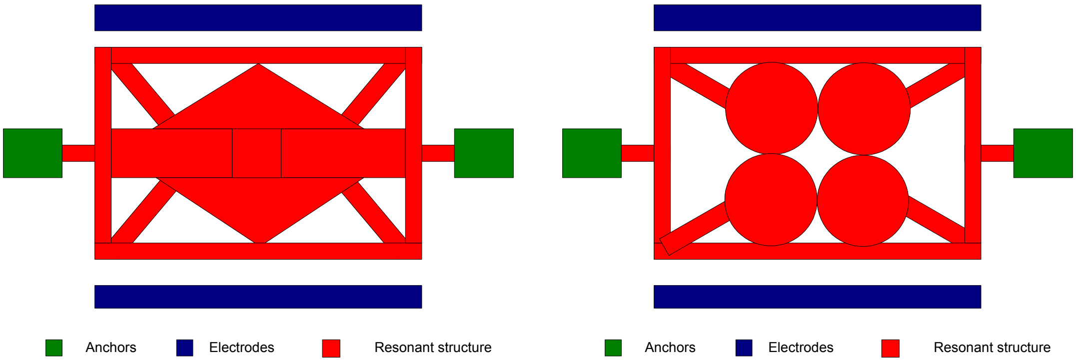


Figure 6.1: Conceptual design solutions for sandwich resonator topologies using shapes and beams.

6.1.2 Partial Topology Optimisation Case Studies: An Example

This example shows how CNS-Burst could be used in problems of partial topology optimisation, i.e. problems where only part of the resonant structure is modified in order to reach certain values of the design criteria. In this case a butterfly resonator (Figure 6.2a) is presented [Yan, 2004]. FF-beam resonators present very small electrodes surface, leading to high motional resistance. The butterfly structure resonates longitudinally like FF-beam resonators, but its extremities present a larger electrode surface, leading to higher capacitive area. In this way, the motional resistance can be decreased without largely affecting the resonant frequency. Again here the objective is to define the topology of the extremities of the structure for a certain frequency and mode, without modifying the length of the main resonating beam (design space A in Figure 6.2b). Variables are the width and length of the beams used to define the topology and their number. The only constraints of the problem are fabrication constrains.

Figure 6.3 shows some solutions to the task, illustrating how the method can be applied. Although these butterfly resonators are only conceptual solutions, they show the potential of the method in solving practical design tasks.

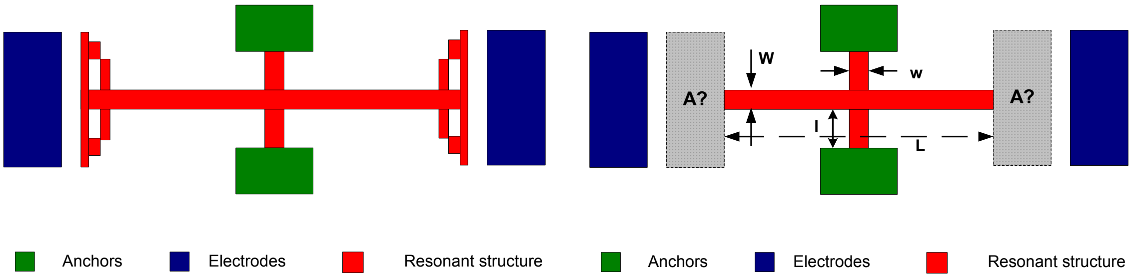


Figure 6.2: a) Butterfly resonator.

Figure 6.2: b) Design space.

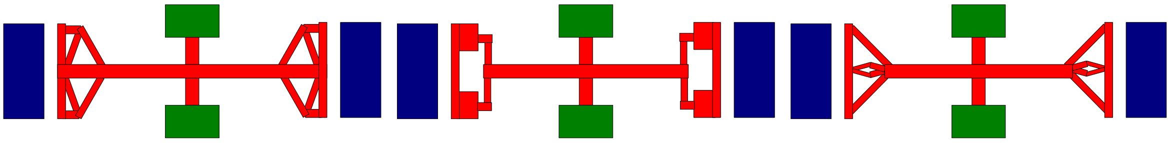


Figure 6.3: Examples of conceptual topologies.

6.2 Shape Optimisation for MEMS Design

This section demonstrates the extensibility of the method in handling design tasks that present increased complexity due to the presence of new primitive shapes in their design. Scaling the method for these tasks requires the extension of the CNS representation to include new building blocks. The novel resonators obtained in 5.2.4 were created making exclusive use of beams. The following examples show how the representation method and modelling capabilities of CNS-Burst can be easily extended to include new representation modules.

The introduction of new primitives allows the application of the method to shape optimisation tasks, completing the range of design optimisation problems that can be handled using CNS-Burst. The possibility of extending the method to shape optimisation is demonstrated through the use of theoretical examples.

Shape optimisation has an important role in MEMS design, considering the recent interest in investigating innovative shapes for resonators [Yan, 2007; Lee, 2008]. The search for resonators with a specific resonant frequency and mode has been lately extended to many new shapes that resulted advantageous when compared with traditional beam structures [Yan, 2007]. Figure 6.4 shows some examples of filters obtained arraying innovative resonators shapes, such as rings, disks and squares. CNS-Burst demonstrates to be a potentially useful tool in the definition of new micro-devices, as well as in the exploration of the effect of shape parameters variation on design.

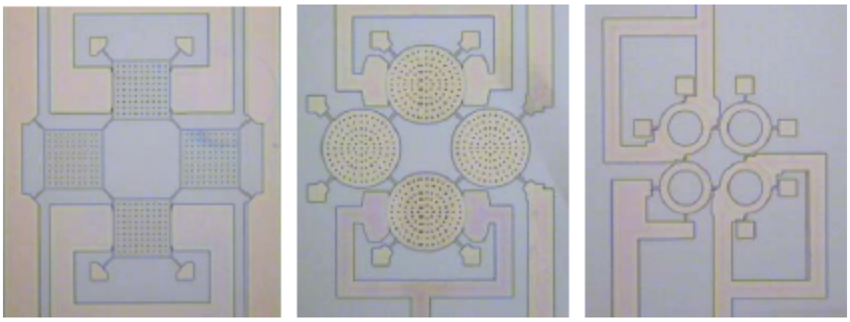


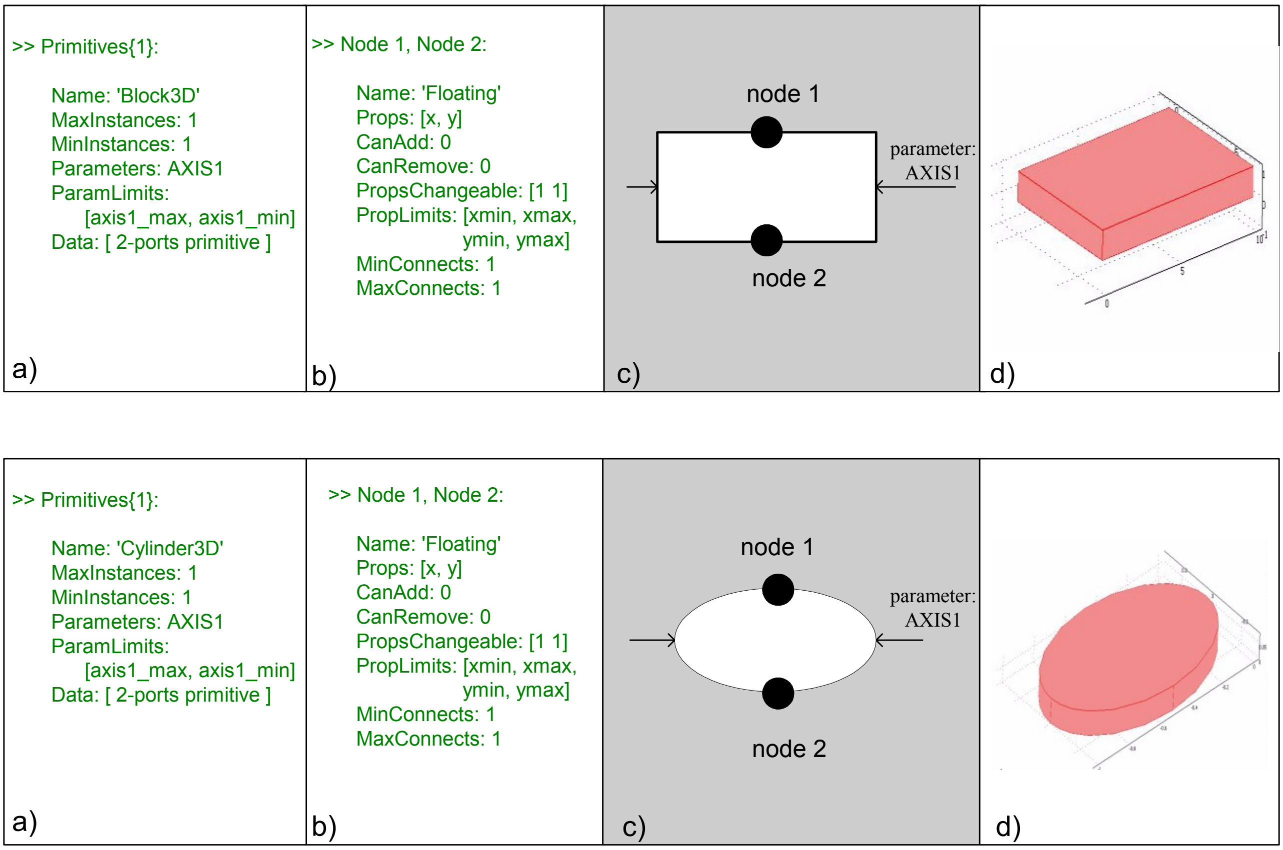
Figure 6.4: Coupled square, disk and ring microresonators filters [Yan, 2007].

6.2.1 Definition of New Primitives

Many generative design methods make use of basic shapes as building elements to create more complex geometries. Shape grammars, based on a set of shapes and rules, are an example of approach to the generation of designs. Part of the research conducted by Agarwal [Agarwal, 1999] consisted in developing a two-dimensional grammar able to create forms that meet the functionalities of MEMS. This parametric shape grammar is based on the use of basic elements to form actuators and springs as main constituents of comb-driven microresonators. The intent of this section is to introduce a set of new basic shapes for the generation of designs and theoretically demonstrate their potential utility in developing more complex and innovative resonators.

As seen in Chapter 3, the CNS-Burst method is based on the definition of primitives as building blocks and modification rules, which can be applied to any primitives. Primitives are elements provided with some geometrical properties as well as an additional set of physical properties. Mass primitives, for example, are defined by their dimensions and positions, as well as by their mass or density. The set of properties may also be null, like in the case of simple beams. In the example of size and topology optimisation presented in this chapter, beams have been the only primitive used (although mass primitives have been used in Chapter 4 for the synthesis of meandering resonators). It has been mentioned though, how new generation microresonators make use of shapes that may be more advantageous than typical beam-based structures. A generative method useful to MEMS designers should be able synthesis devices making use of these shapes.

The two new basic elements introduced in the library of primitives for CNS representation are the ‘block’ and the ‘cylinder’ primitive. Block primitives, exactly like beams, have rectangular section. While cylinder primitives have elliptical section. Figure 6.5a-c shows the CNS geometrical model of these two primitives, as well as their COMSOL simulation model (Figure 6.5d). The transformation of CNS geometrical objects into 3D COMSOL objects is executed using the COMSOL extrusion feature: a 2D simulation object corresponding to the CNS representation in 6.5c is extruded to created a 3D simulation object (Figure 6.5d). Disks and square plates are a particular case of block and cylinder primitives.



```
>> Primitives{1}:

     Name: 'Block3D'
     MaxInstances: 1
     MinInstances: 1
     Parameters: AXIS1
     ParamLimits:
          [axis1_max, axis1_min]
     Data: [ 2-ports primitive ]
```

```
>> Node 1, Node 2:

     Name: 'Floating'
     Props: [x, y]
     CanAdd: 0
     CanRemove: 0
     PropsChangeable: [1 1]
     PropLimits: [xmin, xmax,
                  ymin, ymax]
     MinConnects: 1
     MaxConnects: 1
```

```
>> Primitives{1}:

     Name: 'Cylinder3D'
     MaxInstances: 1
     MinInstances: 1
     Parameters: AXIS1
     ParamLimits:
          [axis1_max, axis1_min]
     Data: [ 2-ports primitive ]
```

```
>> Node 1, Node 2:

     Name: 'Floating'
     Props: [x, y]
     CanAdd: 0
     CanRemove: 0
     PropsChangeable: [1 1]
     PropLimits: [xmin, xmax,
                  ymin, ymax]
     MinConnects: 1
     MaxConnects: 1
```

Figure 6.5: Block and Cylinder primitives: Matlab and COMSOL objects.

The choice of transforming CNS primitive object into simulation object through extrusion comes from the fact that cylinders can be obtained from the extrusion of any closed surface. Ideally, the extruded surface can have any profile, allowing the creation of the most innovative building blocks. Extrusion is also a common representation feature in drawing and analysis computational tools, allowing CNS-Burst to be interfaced with any of these instruments.

As shown in Figure 6.5a, block and cylinder primitives are defined like any other CNS primitive introduced so far, but are different from the latter in the way they are transformed in COMSOL objects. Also, their specific characteristic is that they can be combined to form other primitives, as it will be shown in the next section.

6.2.2 Combination of Shapes

In many engineering fields such as MEMS, new design building blocks are always in demand in the search of ever more complex and innovative design solutions. It is understandable how a library of primitives would turn into a great advantage for designers. The two primitives introduced in the previous section can be used as single building blocks or can be employed to build an entire library of primitives. A set of Boolean operations, applicable only to block and cylinder primitives have been defined for this purpose. These operations are:

- Union (U)
- Intersection (∩)
- Difference (—).

Figure 6.6 graphically explains how these operations are applied to the block and cylinder primitives to form new set of primitives. Figure 6.6a gives an example of Boolean union between a block and a cylinder to form a concave shape. Figure 6.6b gives an example of Boolean intersection between the same to primitives. Figure 6.6c shows how these two primitives can be subtracted to form a half moon profile. This new profiles can be then extruded to form simulation objects as previously explained.

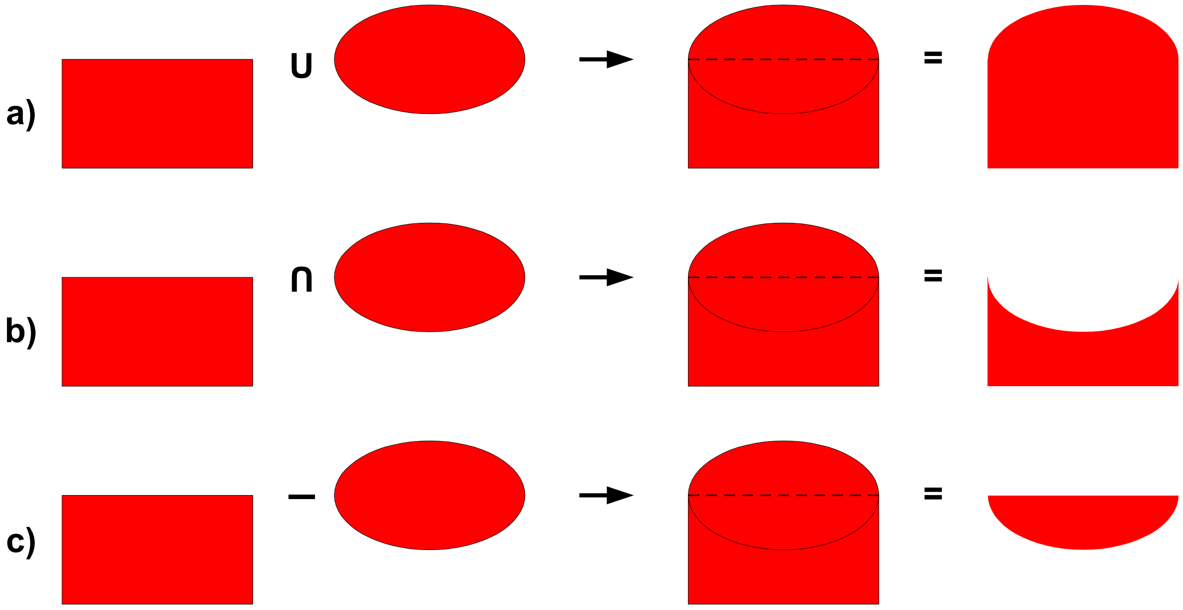


Figure 6.6: Boolean operations on basic primitives (section view): a) union; b) intersection; c) difference.

The new primitives generated by the application of the Boolean operations to blocks and cylinders are defined as 'combined primitives'. Rings are an important example of combined primitives (Figure 6.7), seen the interest in this shape for microresonators [Yan, 2007].

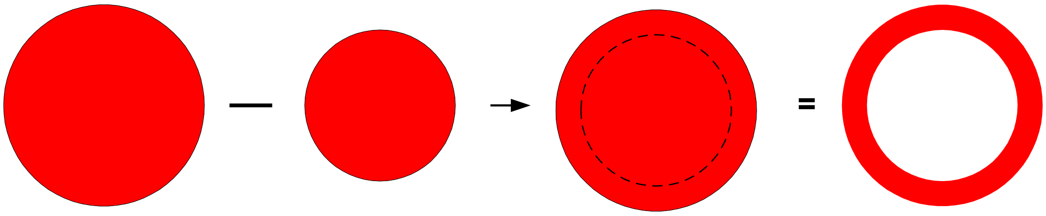


Figure 6.7: Ring: combined primitive obtained from Boolean difference applied to two basic disk primitives (section view).

Figure 6.8 shows some examples of combined primitives. These new shapes are not necessarily obtained applying a single Boolean operation to two basic primitives. The more complex shapes are obtained through a set of operation, or tree of modifications in Boolean terms. The most complex shapes are obtained by subsequent application of Boolean operations to blocks and cylinders, or just combining existing combined shapes. The only two basic primitives to be used with Boolean operations though are blocks and cylinders, due to their specific characteristics.

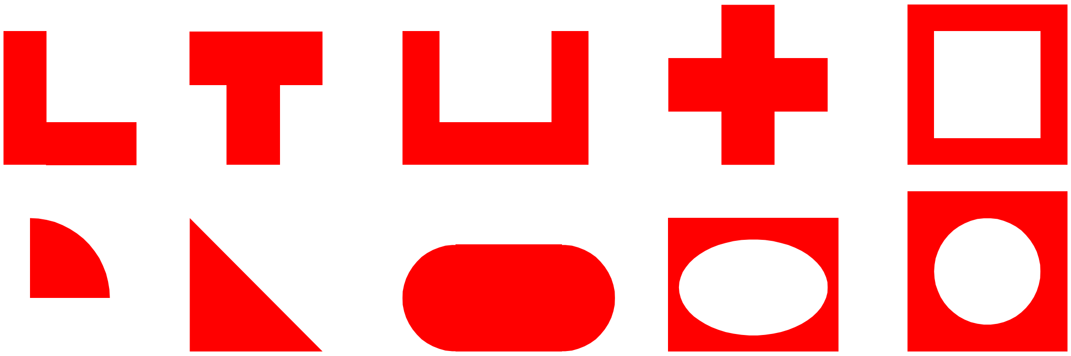
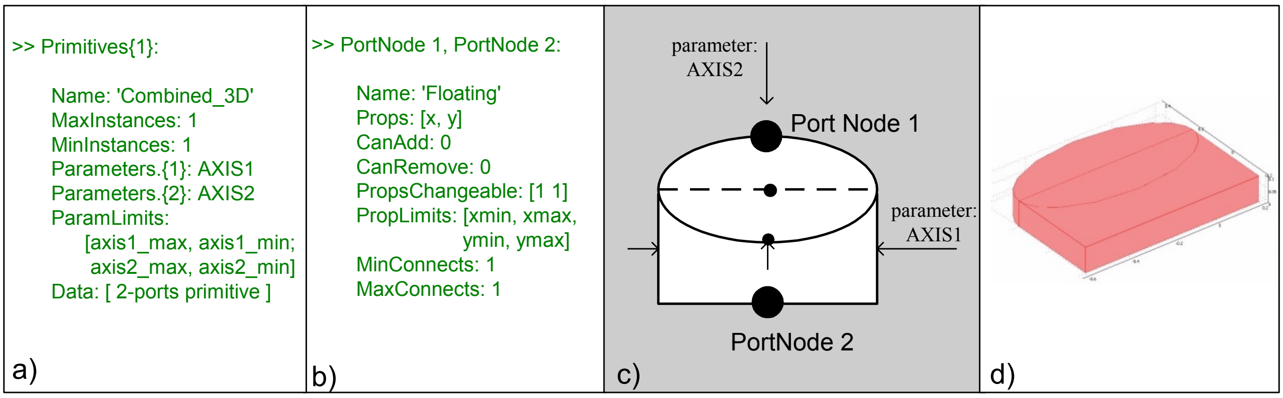


Figure 6.8: Combined primitives' library (section view).

CNS objects are easily transformed into COMSOL simulation objects, as COMSOL supports Boolean operations between shapes. An unlimited number of primitives can be added to the CNS library. Modification operators are applicable to any combined primitives and the search algorithm is free to use any primitives to compose the desired designs. Boolean operations and extrusion can deliver many shapes for simulation objects, opening the most varied possibilities for designers. Figure 6.9 shows a CNS-Burst representation of a combined primitive.



```
>> Primitives{1}:

      Name: 'Combined_3D'
      MaxInstances: 1
      MinInstances: 1
      Parameters.{1}: AXIS1
      Parameters.{2}: AXIS2
      ParamLimits:
           [axis1_max, axis1_min;
            axis2_max, axis2_min]
      Data: [ 2-ports primitive ]
```

a)

```
>> PortNode 1, PortNode 2:

      Name: 'Floating'
      Props: [x, y]
      CanAdd: 0
      CanRemove: 0
      PropsChangeable: [1 1]
      PropLimits: [xmin, xmax,
                   ymin, ymax]
      MinConnects: 1
      MaxConnects: 1
```

b)

Figure 6.9: Combined primitive: Matlab and COMSOL objects.

As defined in 3.2.3, combined primitives are in fact subsystems, being formed by more than one primitive and having only port nodes to interface with the rest of the system. The port-nodes are all is seen by the rest of the system in which they are included. The implementation of combined primitives allows the removal of design parameters of single primitives forming the subsystem. This encapsulation is at the base of the search algorithm's capability to manage the growing number of parameters when design complexity increases, hence favouring the method's scalability.

Finally, shapes could be combined a priori to form a library on which the search algorithm can draw or, as it is foreseeable for more advanced synthesis tools, the search algorithm could directly combine blocks and cylinders to automatically generate combined shapes while running. This last possibility offers an interesting point of investigation for future developments of the method.

6.2.3 Shape optimisation tasks: an Example of Application

Extending the method's use for shape optimisation tasks means finding the shape of the design building blocks that optimise the design objectives. This typology of design task requires the definition of primitives with different shapes other than the beams used so far to solve size and topology optimisation design tasks. The primitives introduced in the previous section and the possibility to have a library of building blocks allow solving this type of optimisation problems using CNS-Burst.

Here a concave clamped-clamped (CC) beam resonator [Yan, 2004] is presented as an example of application of combined primitives. A CC-beam resonator is usually a simple beam resonator clamped at its extremities that resonates transversally (Figure 6.10).

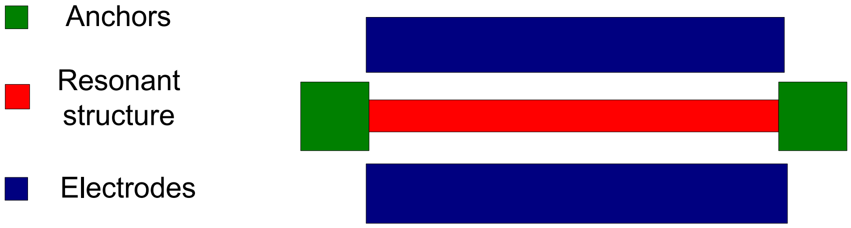


Figure 6.10: Clamped-clamped beam resonator.

As shown in Figure 6.11, a new version of this resonator is given using a particular shape (concave) for the resonant structure, obtained from the combination of basic primitives. In particular, one block primitive and two cylinder primitives are used, combined through the Boolean intersection operator. The advantage of this new design is a larger electrode surface compared to the straight beam structure. As seen in 5.1.3, the value of the motional resistance R_m is influenced by the surface of the electrodes. The more the surface of the electrodes is extended, the more R_m decreases. Adopting a concave structure rather than a beam one presents the advantage of decreasing the motional resistance without affecting the resonant frequency at same operational voltage.

In order to meet design objectives (a target frequency, for example), the shape of the resonant structure can be optimised applying variations to the value of the parameters defining the combined primitive. This simple example shows how intervention on a shape's parameters can solve design tasks. With the objective of maximising the electrodes surface for a given frequency, new resonators can be synthesised using a library of shapes.

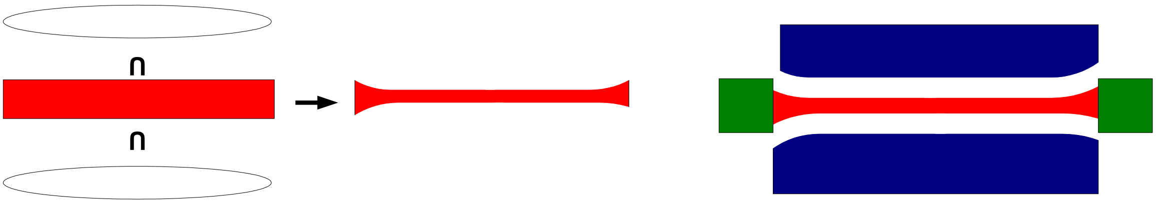
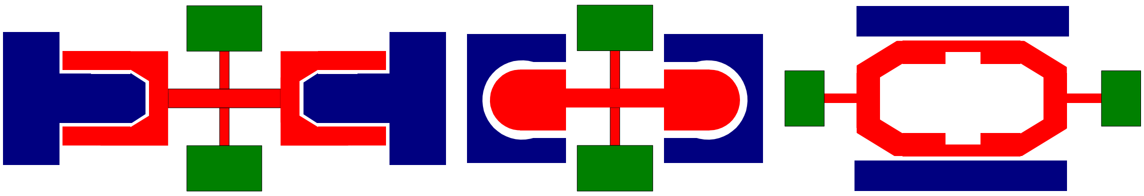


Figure 6.11: Concave resonator.

6.2.4 Capabilities of CNS-Burst: Examples of Complex Resonators

This section exclusively serves the purpose of showing with images how the complexity of designs can scale-up using combined shapes. Figure 6.12a shows a double fork resonator. This type of resonators is largely adopted in MEMS industry and the use of combined shapes could allow their investigation. Many new resonators designs have been explored recently [Yan, 2007], as the one ones shown in Figure 6.12b-c. These represents design alternatives for longitudinal and transverse mode respectively. The design synthesis and optimisation of these resonators would not be possible without the introduction of combined primitives.



a) *b)* *c)*

Figure 6.12: Scaling of design complexity: resonator designs.

Another interesting consideration is that shapes traditionally used for microresonators design have either straight or convex edges (beams or disks for example). Concave shapes are rarely introduced, although presenting the advantage of offering lower motional resistance, due to their extended electrode surface. With combined primitives it is possible to explore and analyse the benefits of these new concave shapes on design objectives. Figure 6.13 shows an example of microresonator featuring a resonant structure with convex ends.

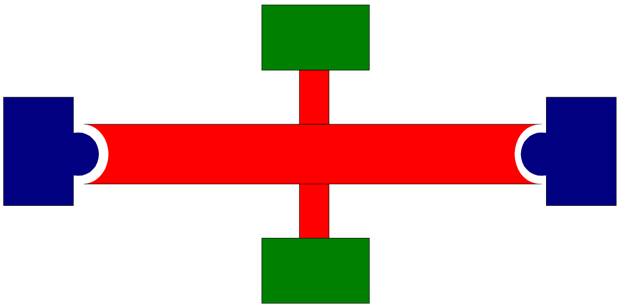


Figure 6.13: Concave combined primitives: application.

This section has presented another possible way of using the method for solving MEMS synthesis tasks that would benefit from the exploration of optimal shapes. Again, CNS-Burst could be particularly useful to determine the connection between unknown design shapes and design parameters, as well as allowing direct comparison with more traditional structures. Also, the scaling of design parameters with the shape optimisation can be explored.

6.3 CNS-Burst Multi-domain Potentials: the Electrode Primitive

As mentioned in previous chapters, one of the goals of this work is to develop a generative design tool to use in different design domains and in multiphysics design tasks. A tool with these properties would be advantageous to designers in solving tasks that require broad knowledge and experience in different fields of engineering.

This section discusses the possibility to extend the method to a second design domain, other than the structural mechanics. MEMS design is a cross-disciplinary activity involving both mechanical and electrical domains. The latter is the domain that can be added to the existing structural mechanics capabilities of CNS-Burst. The extensibility of the method is based on its modular structure and in particular on the possibility to add primitives to the CNS representation method. In this case, the new primitives introduced add to the geometrical properties of the existing primitives some physical properties.

The following considerations are based on the FF-beam resonator case study, but can be applied to the design of any micro-devices. Figure 6.14 shows a schematic diagram of the FF-beam resonator, highlighting how both the electrical and mechanical domain are involved in the sensing ability of the device. The possibility to include 'electrode primitives' in the generation of new designs is considered here. Voltage source building blocks can be used in a variety of design tasks, including those where voltage influences the optimal values of design parameters. In the case of resonators for example, voltage impacts motional resistance (5.1.1). The introduction of electrode primitives would allow exploring the scaling of motional resistance and frequency with voltage. This example shows how the method can be extended to include multi-domain capabilities.

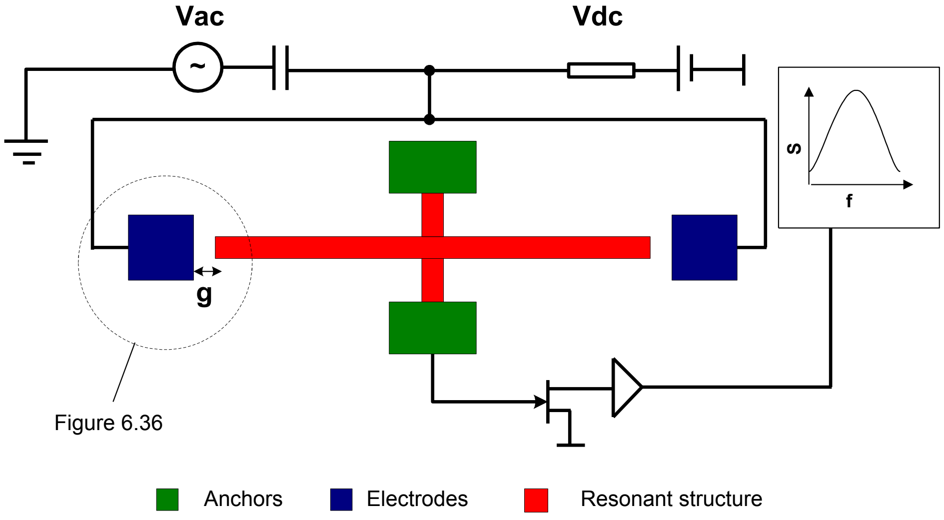


Figure 6.14: FF beam resonator: excitation detection of mechanical motion through capacitive sensing [Mattila, 2002].

Electrodes can be considered as primitives with voltage properties. The Electrodes can have the geometry of any basic or combined primitives, but they also include the physical property voltage (*V*). Other properties may be included in their definition, such as: the dielectric constant of the medium considered (ε), the value of the gap between electrode and the resonant structure (*g*), and the maximum value for voltage (V_{max}). V_{max} sets the maximum value for operation of capacitive structure before pull-in. The electrodes are anchored primitives, which means that the nodes defining their position are fixed. If *V* is considered a design variable, the search can direct values to assign to *V*. Also, the geometrical parameters defining the electrodes could be automatically updated to the geometry of the resonant structure. Figure 6.15 shows the electrode primitive that has been implemented but never used in a synthesis task so far.

```
>> Primitives{1}:

     Name: 'Electrode3D'
     MaxInstances: 1
     MinInstances: 1
     Parameters.{1}: WIDTH
     Parameters.{3}: VOLTAGE
     ParamLimits:
      [width_max, width_min]
      [0, Vmax]
     Data: [ε, g]
```

```
>> Node 1, Node 2:

     Name: 'Anchor'
     Props: [x, y]
     CanAdd: 0
     CanRemove: 0
     PropsChangeable: [0 0]
     PropLimits: [xmin, xmax,
                  ymin, ymax]
     MinConnects: 0
     MaxConnects: 0
```

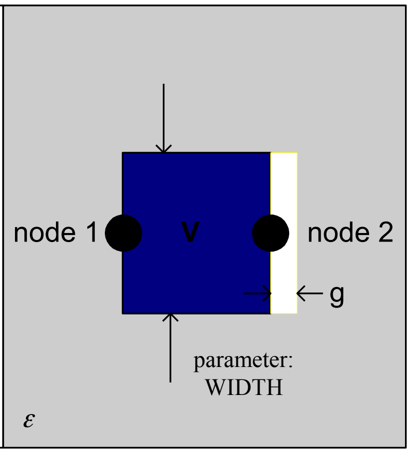


Figure 6.15: Electrode primitive: CNS-Burst implementation.

The transformation of the electrode primitive into a COMSOL simulation object is straightforward. The definition of boundary conditions for COMSOL objects include the possibility to set electric potentials. COMSOL also supports electrical domain analysis and simulation.

Many MEMS design applications would take advantage from the introduction of the electrical domain in the generation and simulation of designs. Electrode primitives could be used in any case study including design objectives that are affected by voltage variation, or in cases where the way in which design parameters scales-up with voltage is investigated.

Also, there are plenty of possible extensions to the introduction of the electrical domain. Electrical circuit simulation capabilities linked to design generation could be advantageous to designers, as well as links to synthesis tools applied to electronics design.

The electrode primitive has been defined specifically for MEMS design tasks, but could be used in the generation of any design (other than MEMS) that couples the electric domain with any other design domain.

6.4 Discussion and Conclusions

The flexibility of CNS-Burst has again being demonstrated through the modelling and solution of different MEMS design cases. CNS-Burst is a complete package for design and analysis of MEMS, potentially applicable to any type of optimisation task (size, shape and topology optimisation). The possibility to extend the method to more than a single design domain has also been shown.

Many of the future developments here introduced are still in at initial stages and open several questions. The introduction of new block, cylinder and combined primitives for example, leave room for thoughts on the way their connection should happen. In particular, the choice of where nodes have to be located on the surface of these new shapes to allow for connection may be of difficult definition. Primitives may need a variable number of nodes, to use according to which other primitives they are connected to. Also, not all the primitives can be connected to each other at node points. For example, many connections may bring difficulties in meshing in the node area at simulation stage. Some other connections may be impossible to realise at manufacturing stage. In any of these cases, there is the need for further modification rules to apply exclusively to the combination of these new primitives. Or further constraints may be added to the definition of primitives, restricting the way in which they can be combined.

Extending the applicability of the method to other design domains will also require the inclusion of further knowledge in order to determine how primitives can be added and how they can interact with each other. In any of these cases, many new elaborate and challenging developments can be forecasted for the method.

CHAPTER 7 - CONCLUSIONS

7.1 Summary of Results

This work has investigated the development of an automated design method aimed at facilitating synthesis through the generation of a range of feasible and optimally directed design alternatives. The method is aimed at assisting designers in the exploration of performance limits and trade-offs for synthesis tasks as well as reducing design time.

The method presented is a mutidomain, flexible, bottom-up, integrated simulation-based technique. A modular approach was used to address issues related to its generality and adaptability.

The method combines a multicriteria generate-and-test search algorithm, called Burst, with a Connected Node System (CNS) design representation and provides automatic links to multiphysics simulation for quantitative evaluation of design performance throughout the synthesis process. In any design task, the method provides designers with the possibility to choose preferred solutions in archives of Pareto optimal designs and have a better insight of design trade-offs.

In order to be initially tested and compared against existing techniques, the method was applied to two benchmark synthesis tasks in the domain of MEMS. In the first benchmark example the synthesis of a microcompliant mechanism with constrained input motion was considered. The second example examined consisted of synthesising a microresonator with specific spring stiffness in the x and y-direction and natural resonant frequency. The solutions obtained by the search procedure successfully met the design specifications, matched or improved solutions obtained through other synthesis techniques, and offered a variety of solutions to the synthesis task.

Subsequently, characteristics of flexibility and scalability of the method have been validated through more complex MEMS design tasks. The appraisal was provided by practical applications such as resonators developed for the mobile and satellite industry. The tasks examined ranged from size optimisation of longitudinal FF-beam resonators to more complex topology optimisation of sandwich resonators. The design objectives examined were longitudinal and transversal resonant frequencies, motional resistance and quality factor. This required significant method development for automated pre- and post-processing of simulation models and results. The results confirmed the effectiveness of the method in complex MEMS synthesis tasks as well as its potential utility for MEMS designers, opening new frontiers for the improvements and uses of synthesis methods.

7.2 Discussion of the Method: Contributions and Limits

Stemming from the analysis of weaknesses in the development and use of synthesis methods, this research has recognised the need for a multidomain generative tool, flexible enough to be adapted to different design tasks, able to scale up with the complexity of designs, and easy to be integrated with other design and evaluation tools to use in a variety of scenarios. The successful application of the method to the examples presented, showed the versatility of the approach, its capability to incorporate sufficiently accurate models and simulation software, and to be scalable in complex tasks. The results obtained confirmed that the straightforward adaptability of multi-domain synthesis methods creates the potential to provide customisable tools for automated design synthesis. This novel use of the computer in design practice will not only help designers to achieve the most beneficial design alternatives, but also has the potential to boost innovation in many fields of design.

The method was based on a new MO search algorithm (Burst), adopted to provide a simple and flexible stochastic approach to multicriteria search for synthesis tasks. Compared with other techniques, this choice was able to bring the following advantages:

- Improved exploration capability and efficiency of the search, not limiting in any way the offer of design alternatives

- No requirement of problem-specific tuning and capacity to be easily applied to a variety design tasks.

Another important result set by this work consists in the successful MEMS designs obtained. A step forward in the development of synthesis tools for MEMS design has been done with the application of the method to design tasks presenting complex objectives. Until this point, fundamental MEMS design parameters such as motional resistance or quality factor have been hard to integrate in a synthesis tool, due to their complexity and their multiphysics nature. The method was able to introduce new techniques to evaluate these objectives, demonstrating the potential of its use in everyday design practice. The examples examined are cases of real design practice, differently from many ideal and simplistic examples presented in previous literature on MEMS synthesis.

A common concern in the development of this method has been what Cagan defines as 'the appropriate balance between reasoning and brute force search in the synthesis process' [Cagan, 2005]. Increased reasoning capabilities can enable more sophisticated design representation and evaluation, while increased search abilities can enable more efficient exploration of larger design spaces. It was felt in this work, that time consuming activities oriented at more sophisticated approaches to evaluation and representation unbalanced the development of activities such as generation and search. Time-consuming activities of the like of creation of accurate simulation models, set-up of design objectives or, in general, the inclusion of design domain knowledge, dominated the development of the method. It was also argued in this work, that these activities had the main responsibility in limiting the success of automated synthesis methods, and were consequently the most important to be addressed. It is felt though, that there is a scope for improvement in results obtainable through refinement of search and optimisation techniques. This dichotomy in the development of the method confirmed the complex nature of synthesis tools, due to the difficulty that designers encounter in working on the different fronts that characterise the activities of a synthesis method. Further developments of the method may overcome these weaknesses, allowing a better distribution of computational capabilities among the different activities of the method.

7.3 Future Work and Conclusive Remarks

As request for synthesis tools is mounting and their development still lags behind, the future of synthesis methods in the next few decades will continue to be active. A long-term vision of the method's development as well as of synthesis methods in general is presented here:

- The development of new combined primitives, as seen in 6.4, brings new capabilities to the method, such as the possibility to use an extended library of shapes for the creation of new complex shape and topology optimisation design tasks. This latest feature is particularly important for MEMS design, for which innovative shapes are constantly under investigation. Together with a library of geometrical shapes, it would be also possible to implement a library of devices with specific roles, for example electrodes or anchors. The use of combined shapes will increase abstraction and encapsulation of functionalities, allowing the overall number of parameters to remain constant as higher levels of complexity are reached.
- MEMS offer many more applications and challenges for improving and extending the method's capabilities. One example could be the implementation of new complex design parameters as search objectives. The calculation of the quality factor seen so far is limited to the consideration of support losses, while any other of its components have been neglected. There are many types of dissipation to consider that are important for MEMS design, and future research could focus on discovering the optimal geometrical designs that also include external dominant dissipation mechanisms, such as surface loss and material damping. The calculation of Q could be perfected including external packages for the evaluation of its components. Also, other external packages could be integrated with the method to improve its capabilities, for example CAD tools developed for mask synthesis, in order to take into account advanced manufacturing issues.
- The structure and the architecture of the method, as well as its integration with external simulation and analysis have been of central interest in this work, while less attention was given to other aspects. For example, it is felt that the integrated multicriteria optimisation could be investigated more deeply. The flexibility of Burst, which was ideal for this investigation, is however offset by its naivety. A significant area of future investigation rests with the potential of the technique's multicriteria search capabilities

and how the method can be refined to guide the search more effectively and efficiently. Although optimisation methods have been the basis for the current generation and search for synthesis systems, future methods will focus on learning and cognition of models and techniques for automated design. Learning methods from AI will be increasingly integrated and will improve the automated synthesis process [Cagan, 2005]. For example, the search could be coupled with machine learning techniques (ML), as previously done by Vale [Vale, 2003], to allow reuse of learned concepts from previous searches. In his work, Vale advocates a more direct application of ML in synthesis techniques than is otherwise presented. However, Vale states that flexible computational design tools are necessary prerequisite for the inclusion these techniques [Vale, 2003]. Continued generation of new schemes for the incorporation of ML in synthesis in the future will create more sophisticated synthesis tools to deal with growing demand for intelligence in design for more complex scenarios [Vale 2003].

- A better grasp of the method can be reached using additional modules to facilitate the understanding of the search. The method can be thought of as a toolbox for designers, i.e. containing any tool that could facilitate and shorten the synthesis task. For example, a computational evaluation of the convergence of the archive could always be performed for those cases where the convergence is not immediate and intuitive, in order to have a precise measure of the research efficiency.
- An important step in the investigation of the method is to see how well results meet fabrication requirements. MEMS fabrication is not straightforward and sometimes designs do not correspond to what it is possible to manufacture. An interesting development would be to see how generated design can be successful at the manufacturing stage and how they compare with designs obtained by hand.
- Another interesting investigation of the multidomain nature of the method would be the adaptation to other domains and scenarios.
- The study of the general architecture of synthesis tools put interesting basis for future development of this method as well as of other methods. Looking at each of the main four activities of the method (representation, generation, simulation and search), the emphasis will be on how to make these aspects more effective, more efficient, more intelligent, more complete, and better integrated [Cagan, 2005]. In particular, the integration of synthesis methods not only with analysis tools, but also with CAD/CAM tools for detailed representation and manufacturing, could represent interesting applications for industry.

- Finally, the biggest challenge will be how to transfer this and any other research on generative design methods to the industrial sector. This will require time and adjustment to practical problems for designers and much more collaboration between synthesis methods implementers and designers.

Closing Note

This research contributes to the advancement of computational synthesis methods for complex engineering tasks. At this stage, few implementation obstacles can still slow down progresses in the field. However, in the light of the results obtained, future development and applications of generative design techniques offer an exiting and promising research prospect.

References

ACHICHE, A., BOLOGNINI, F. & FAN, Z. (2007) Review of Automated Design and Optimization of MEMS. *2007 IEEE International Symposium on Industrial Electronics.* Vigo, Spain.

ACHICHE, S., FAN, Z. & BOLOGNINI, F. (2007) Review of Automated Design and Optimisation of MEMS. *2007 IEEE International Symposium on Industrial Electronics.* Vigo, Spain.

AGARWAL, M. (1999) Supporting Automated Design Generation: Function Based Shape Grammars and Insightful Optimisation. Pittsburgh, USA, Carnegie Mellon University.

AGARWAL, M., CAGAN, J. & STINY, G. (2000) A Micro Language: Generating MEMS Resonators using a Coupled Form-Function Shape Grammar. *Environment and Planning B: Planning and Design,* 27**,** 615-626.

ANANTHASURESH, G. K., KOTA, S. & GIANCHANDANI, Y. (1994) A Methodical Approach to the Design of Compliant Mechanisms. *Solid-State Sensor and Actuator Workshop.*

ANANTHASURESH, G. K. & KOTA, S. (1995) Designing Compliant Mechanisms. *Mechanical Engineering***,** 93-96.

ANANTHAKRISHNAN, V., SARMA, R. & ANANTHASURESH, G. K. (2003) Systematic Mask Synthesis for Surface Micromachined Microelectromechanical Systems. *Journal of Micromechanics and Microengineering,* 13**,** 927-941.

ANANTHASURESH, S. G. K. (2003) *Optimal Synthesis Methods for MEMS*, Springer.

ANDERSON, R. A. (1967) *Fundamentals of Vibrations,* London, Collier-MacMillan Limited.

ANTONSSON, E. K. (1995) Structured Methods for MEMS Design. *NSF.*

ANTONSSON, E. K. & CAGAN, J. (2001) *Formal Engineering Design Synthesis,* Cambridge, Cambridge University Press.

ANTOULAS, A. C. (1999) *Approximation of Linear Dynamical Systems*, J.G. Webster.

BALDOCK, R. (2007) Structural Optimisation in Building Design Practice: Case-Studies in Topology Optimisation of Bracing Systems. *Department of Engineering.* Unversity of Cambridge, UK.

BECHTOLD, T., RUDNYI, E. B. & KORVINK, J. G. (2003) Automatic Generation of Compact Electro-Thermal Models for Semiconductor Devices. *IEICE Transactions on Electronics,* 86**,** 459-465.

BENSDØE, M. P. & KIKUCHI, N. (1988) Generating Optimal Topologiesin Structural Design Using a Homogenization Method. *Computer Methods in Applied Mechanics and Engineering***,** 197-224.

BENSDØE, M. P. & O., S. (2003) *Topology Optimisation: Theory, Methods and Applications,* NY, Springer, NY.

BILÍC, D. (2001) Micromachined Resonators. *Engineering, Electrical Engineering and Computer Science.* Berkeley, University of California at Berkeley.

BINDEL, D. S. & GOVINDJEE, S. (2005) Elastic PMLs for Resonator Anchor Loss Simulation. Department of Civil and Environmental Engineering University of California Berkeley.

BINDEL, D. S., QUÉVY , E., KOYAMA, T., GOVINDJEE, S., DEMMEL, J. W. & HOWE, R. T. (2005) Anchor Loss SImulation in Resonators. *IEEE Micro Electro Mechanical Systems Conference (MEMS'05).* Miami, Florida, US.

BOLOGNINI, F. (2003) The Performance Aspects of Mechanical Computational Design. Cambridge, University of Cambridge Engineering Department.

BOLOGNINI, F. (2004) Computational Synthesis of Microsystems. *Enginnering Department.* Cambridge, UK, University of Cambridge.

BOLOGNINI, F., SHEA, K., VALE, C. W. & SESHIA, A. A. (2006) A Multi-criteria System-based Method for Simulation-driven Design Synthesis. *ASME 2006 International Design Engineering Technical Conferences & Computer and Information in Engineering Conference (IDETC/CIE 2006).* Philadelphia, Pennsylvania, USA.

BOLOGNINI, F., SESHIA, A. A. & SHEA, K. (2007) A Computational Design Synthesis Method for MEMS Using COMSOL. *European COMSOL Conference 2007.* Grenoble, France.

BOLOGNINI, F., SESHIA, A. A. & SHEA, K. (2007) Exploring the Application of a Multi-domain Simulation-based Computational Method in MEMS Design. *International COnference on Engineering Design (ICED '07).* Paris, France.

BURGESS, S. C., MOORE, D. F., NEWLAND, D. E. & KLAUBERT, H. (1997) A Study of Mechanical Configuration Optimisation in Microsystems. *Research in Engineering Design,* 9**,** 46-60.

CAGAN, J., CAMPBELL, M. I., FINGER, S. & TOMIYAMA, T. (2005) A Framework for Computational Design Synthesis: Model and Applications. *Journal of Computing and Information Science in Engineering,* 5**,** 171-181.

CAMPBELL, M. I. (2000) The A-Design Invention Machine: a Means of Automating and Investigating Conceptual Design. PA, Carnegie Mellon Univesity.

CAMPBELL, M. I. (2001) An Automated Approach to Generating Novel MEMS Accelerometer Configurations. *Texas-Area MEMS Workshop.* Austin, Texas, USA.

CAMPBELL, M. I. & RAI, R. (2003) A Generalisation of Computational Synthesis Methods in Engineering Design. *American Association for Artificial Intelligence Spring Symposium.* Stanford, California, USA.

CAMPBELL, M. I., NAIR, S. & PATEL, J. (2007) A Unified Approach to Solving Graph Based Design problems. *ASME 2007 IDETC/CIE Conferences.* Las Vegas, NV, USA.

CARR, D. W., EVOY, S., SEKARIC, L., CRAIGHEAD, J. M. & PARPIA, J. M. (1999) Measurement of Mechanical Resonance and Losses in Nnometer Scale SIlicon Wires. *Applied Physics Lett.,* 75.

CHAKRABARTI, A. (2002) *Engineering Design Synthesis: Understanding Approaches and Tools*, Springer.

CHEN, Y. & WHITE, J. (2000) A Quadratic Model for Non-linear Model Order Reduction. *Modeling and Simulation of Microsystems.* San Diego, California, USA.

CHO, S., LEE, K. & KIM, T. (2002) Development of a Geometry-Based Process Planning for Surface Micromachining. *International Journal of Production Research,* 40**,** 1275-1293.

COMSOL (2007) COMSOL Multyphysics. 3.4 ed., www.comsol.com

DAVIES, A. G. (1967) *Solid Geometry in 3-D,* London, Chatto & Windus.

DEEPAK, S. R., DINESH, M. & ANANTHASURESH, G. K. (2008) A Comparative Study of the Formulations for Topology Optimization of Compliant Mechanisms. *ASME 2008 International Design Engineering Technical Conferences & Computers & Information in Engineering Conference.* New York, NY, US.

DENG, Y. M. & LU, W. F. (2004) A Conceptual Design Synthesis Framework for Micro-Electro-Mechanical Systems (MEMS). *ASME Design Engineering Technical Conference.* Salt Lake City, USA.

DUWEL, A. (2006) Engineering MEMS Resonators with Low Thermoelastic Damping. *Journal of Microelectromechanical Systems,* 15**,** 1437-1445.

FAN, Z., SEO, K., HU, J., ROSENBERG, R. C. & GOODMAN, E. D. (2003) System Level Synthesis of MEMS via Genetic Programming and Bond Graphs. *Genetic and Evolutionary Computation Conference.* Chicago, Illinois, USA.

FAN, Z., SEO, K., ROSENBERG, R. C., HU, J. & GOODMAN, E. D. (2003) Computational Synthesis of Multi-domain Systems. *AAAI Spring Symposium on Computational Synthesis.* Standford, California, US.

FAN, Z., WANG, J., WEN, M., ACHICHE, S., GOODMAN, E. & ROSENBERG, R. (2007) An Evolutionary Approach For Robust Layout Synthesis of MEMS. *Evolutionary Computation in Dynamic and Uncertain Environment,* . Springer.

FAROUKI, R. T. (1999) Closing the Gap between CAD Model and Downstream Application. *SIAM News,* 32**,** 303-319.

FEDDER, G. K. (1994) Simulations of Microelectromechanical Systems. Berkeley, California, USA, UC Berkeley.

FEDDER, G. K. & CLARK, K. H. (1995) Modelling and Simulation of Meander Suspensions. *International Conference on Simulation an Design of Microsystems and Microstructures.* Southampton, UK, Computational Mechanics Publications.

FEDDER, G. K., IYER, S. & MUKHERJEE, T. (1997) Automated Optimal Synthesis of Microresonators. *9th International Conference on Solid State Sensors and Actuators (Transducres '97).* New York, IEEE.

FONSECA, C. M. & FLEMING, P. J. (1993) An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evol. Comp.,* 3**,** 1-16.

FRECKER, M. I., ANANTHASURESH, G. K., NISHIWAKI, S. & KOTA, S. (1997) Topological Synthesis of Compliant Mechanisms using Multi-Criteria Optimisation. *Transaction of the ASME 1997,* 119**,** 238-245.

FRECKER, M., KIKUCHI, N. & KOTA, S. (1998) Topology optimisation of Compliant Mechanisms with Multiple Outputs. *Structural and Multidisciplinary Optimization,* 17**,** 269-278.

GERO, J. S. (1994) *AI and Creativity,* Dordrecht, Kluwer.

GIBSON, D., PURDY, C., HARE, A. & JR., B. F. (1999) Design Automation of MEMS Systems Using Behavioral Modeling. *IEEE Great Lakes Symposium on VLSI.* Great Lakes, USA.

GLOVER, F. & LAGUNA, M. (1997) *Tabu Search,* Boston, MA, USA, Kluwer Academic.

GRIERSON, D. E. & KHAJEHPUR, S. (2002) *Conceptual Design Optimization of Engeneering Structures*.

HAO, Z., ERBIL, A. & AYAZI, F. (2003) An Analytical Model for Support Loss in Micromachined Beam Resonators with In-plane Flexural Vibrations. *Sensors and Actuators A,* 109**,** 155-164.

HAO, Z. & AYAZI, F. (2005) Support Loss in Micromechanical Disk Resonators. *Micro Electro Mechanical Systems, 2005. MEMS 2005. 18th IEEE International Conference on.*

HEDGE, S. & ANANTHASURESH, G. K. (2008) Topology Optimisation of Micromachined Structures with the Manufacturing Constraints of KOH Etching of (110) Silicon. *ASME 2008 International Design Engineering Technical Conferences & Computers & Information in Engineering Conference.*

HERBLEB, J. D. & MOITRA, D. (2001) Global Software Development. *IEEE Software***,** 16-20.

HOLLAND, J. H. (1975) Adaptation in Natural and Artificial Systems. MI, University of Michigan, Ann Arbor.

HOWELL, L. H. (2001) *Compliant Mechanisms,* NY, John Wiley & Sons, Inc, NY.

HSU, W. T., CLARCK, J. R. & NGUYEN, C. T. C. (2001) Q-Optimized Lateral Free-Free Beam Micromechanial Resonators. *11th Int. COnference on Solid-State Sensors & Actuatos (Transducers '01).* Munich, Germany.

HUBBARD, T. J. & ANTONSSON, E. K. (1996) Design of MEMS via Efficient Simulation of Fabrication. *ASME Design Engineering Technical Conference DETC/DFM.* Irvine, California, US.

HUNG, E., YANG, Y.-J. & SENTURIA, S. D. (1997) Low-Order Models for Fast Dynamical Simulation of MEMS Microstructures. *Transducers'97.* Chicago, Illinois, USA.

HUNG, E. & SENTURIA, S. D. (1999) Generating Efficient Dynamical Models for Microelectromechanical Systems from a Few Finite-Element Simulation Runs. *Journal of Microelectromechanical Systems,* 8**,** 280-289.

IYER, S. V. (1998) Layout Synthesis of Microresonators. Pittsburgh, PA, Carnegie Mellon University.

JUNEIDI, Z., TORKI, K., CHARLOT, B. & COURTOIS, B. (2001) *MEMS Synthesis and Optimization*.

KAAJAKARI, V., MATTILA, T., OJA, A. & SEPPÄ, H. (2004) Nonlinear Limits for Single-Crystal Silicon Microresonatos. *Journal of Microelectromechanical Systems,* 13.

KAMALIAN, R., ZHOU, N. & AGOGINO, A. M. (2002) A Comparison of MEMS Synthesis Techniques. *Pacific Rim Workshop on Transducers and Micro/NanoTechnologies.* Xiamen, China.

KAMALIAN, R. (2004) Evolutionary Synthesis of MEMS Devices. *Department of Mechanical Engineering.* Berkeley, California, USA, University of California at Berkeley.

KAMALIAN, R., AGOGINO, A. M. & TAKAGI, H. (2004) The Role of Constraints and Human Interaction in Evolving MEMS Designs: Microresonator Case Study. *ASME DETC/DAC, 2004 Design Engineering Technical Conference-Design Automation Conference.* Salt Lake City, Utah, USA.

KENNEDY, J. & EBERHART, R. C. (1995) Particle Swarm Optimization. *IEEE International Conference on Neural Networks.* Perth, Australia.

KIRKPATRICK, S., GELATT, C. & VECCHI, M. (1983) Optimization by Simulated Annealing. *Science,* 220**,** 671-680.

KOTA, S., HETRICK, J., LI, Z. & SAGGERE, L. (1999) Tailoring Unconventional Actuators using Compliant Transmissions: Design Methods and Applications. *IEEE/ASME Transactions Mechatronics,* 4.

KOTA, S., HETRICK, J., LI, Z., RODGERS, S. & KRYGOWSKI, T. (2000) Synthesising High-Performance Compliant Stroke Amplification System for MEMS. *IEEE Annual International Workshop on MEMS.*

KOTA, S., JOO, J. & KIKUCHI, N. (2000) Topological Synthesis of Compliant Mechanisms using Linear Beam Elements. *MECH.STRUCT. & MACH.,* 28**,** 245-280.

KOTA, S. (2003) Computational Synthesis of Mechanical Systems. *2003 AAAI Spring Symposium.* Stanford, California, US, American Association for Artificial Intelligence.

KOTLARSKY, J. (2007) Re-engineering at LeCroy Corporation: the Move to Component-based Systems. *Journal of Information Technology***,** 465-478.

KOZA, J. R., KEANE, M. A. & STREETER, M. J. (2003) What's AI Done for Me Lately? Genetic Programming's Human Competitive Results. *IEEE Intelligent Systems,* 18**,** 25-31.

LEE, J. E.-Y., YAN, J. & SESHIA, A. A. (2008) Quality Factor Enhancement of Bulk Acousitc Resonators Through Anchor Geometry Design. *Eurosensors XXII.* Dresden, Germany.

LEE, J. E.-Y., ZHU, Y. & SESHIA, A. A. (2008) A Bulk Acoustic Mode Single-Crystal Silicon Microresonator wit High-quality Factor. *Journal of Micromechanics and Microengineering,* 18.

LEE, S., DEMIRCI, M. U. & NGUYEN, C. T. C. (2001) A 10MHz Micromechanical Resonator Pierce Reference Oscillator for Communications. *Transducers '01.* Munich, Germany.

LEE, S. & NGUYEN, C. T. C. (2004) Mechanically-Coupled Micromechanical Resonator Arrays for Improved Phase Noise. *IEEE Int. Ultrasonics, Ferroelectrics and Frequency Control 50th Anniv. Joint Conference.* Montreal, Canada.

LI, H. & ANTONSSON, E. K. Genetic Algorithm in MEMS Synthesis. *ASME International Mechanical Engineering Congress and Exposition (MEMS).* Ahaheim, California, USA.

LI, H. & ANTONSSON, E. K. (1998) Evolutionary Techniques in MEMS Synthesis. *ASME Design Engineering Technical Conferences.* Atlanta, Georgia, USA.

LIFTSHITZ, R. & ROUKES, M. L. (2002) Thermoelastic Damping in Micro and Nanomechanical Systems. *Phys. Rev.,* B 61**,** 5600-5609.

LIPSON, H. & POLLACK, J. B. (2000) Automatic Design and Manufacture of RObotic Lifeforms. *Nature,* 406**,** 974-978.

MA, L. & ANTONSSON, E. K. (2000) Applying Genetic Algorithms to MEMS Synthesis. *ASME International Mechanical Engineering Congress and Exposition.* New York, NY, USA.

MA, L. & ANTONSSON, E. K. (2000) Mask-Layout and Process Synthesis for MEMS. *MSM2000, Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, IEEE.* San Diego, California, USA.

MA, L. & ANTONSSON, E. K. (2001) Robust Mask-Layout Synthesis for MEMS. *International Conference on Modeling and Simulation of Microsystems.*

MALOUF, N. (2000) *An Introduction to Microelectromechanical Systems Engineering,* MA, Artech House.

MATTILA, T., KIIHAMÄKI, J., LAMMINMÄKI, T., JAAKKOLA, O., RANTAKARI, P., OJA, A., SEPPÄ, H., KATTELUS, H. & TITTONEN, I. (2002) A 12MHz Micormechanical Bulk Acoustic Mode Oscillator. *Sensors and Actuators A*, 1-9.

METROPOLIS, N. & ULAM, S. (1949) The Monte Carlo Method. *Journal of the American Statistical Association,* 44**,** 335–341.

MIETTINEN, K. M. (1999) Dordrecht, Kluwer Academic.

MUKHERJEE, T., IYER, S. & FEDDER, G. (1998) Optimisation-based Synthesis of Microresonators. *Sensors and Actuators, A* 70**,** 118-127.

MUKHERJEE, T., ZHOU, Y. & FEDDER, G. K. (1999) Automated Optimal Synthesis of Microaccelerometers. *12th IEEE International Conference on Microelectromechanical Systems (MEMS '99).*

MUKHERJEE, T., ZHOU, Y. & FEDDER, G. K. (1999) Automated Optimal Synthesis of Microaccelerometers. *12th IEEE International Conference on Micro Electro Mechanical Systems (MEMS '99).* Orlando, FL, USA.

NGUYEN, C. T. C. & HOWE, R. T. (1999) An Integrated CMOS Micromechanical Resonator High-Q Oscillator. *IEEE J. Solid-State Cir.,* 34**,** 440-455.

NOCEDAL, J. & WRIGHT, J. S. (2000) *Numerical Optimization,* NY, Springer Verlag

ONGKODJODJO, A. & TAY, F. E. H. (2002) Global Optimization and Design for Microelectromechanical Systems Devices Based on Simulated Annealing. *Journal of Micromechanics and Microengineering***,** 878-897.

OSTERGAARD, D. & GYIMESI, M. (2000) Finite Element Based Reduced Order Modeling of Micro Electro Mechanical Systems (MEMS). *3rd International Conference on Modeling and Simulation of Microsystems.* San Diego, California, USA.

OTTO, K. & WOOD, K. (2001) *Product Engineering: Techniques in Reverse Engineering and New Product Developement,* New Jersey, Prentice Hall.

OUSTERHOUT, J. K. (1998) Scripting: Higher Level Programming for the 21st Century. *IEEE Computer,* 31**,** 23-30.

PAHL, G. & BEITZ, W. (1996) *Engineering Design*, Springer.

PAPALAMBROS, P. Y. & WILDE, D. J. (2002) *Principles of Optimal Design: Modelling and Computation,* Cambridge, Cambridge University Press.

PARETO, V. (1896) *Manual of Political Economy,* New York, A.M.Kelly.

PARK, H. M. & LEE, M. W. (1998) An Efficient Method Of Solving The Navier–Stokes Equations For Flow Control. *International Journal for Numerical Methods in Engineering,* 41**,** 1133-1151.

PARK, K.-W. & GRIERSON, D. E. (1999) Pareto-optimal Conceptual Design of the Structural Design of Buildings using a Multicriteria Genetic Algorithm. *Computer - aided Civil and Infrasctucture Engineering,* 14.

PEDERSEN, C. B. W., BUHL, T. & SIGMUND, O. (2001) Topology Synthesis of Large-displacement Compliant Mechanisms. *International Journal for Numerical Methods in Engineering,* 50**,** 2683-2705.

PETERSEN, K. E. (2000) Bringing MEMS to Market. *Solid-State Sensor and Actuator Workshop.* Hilton Head Island, South Carolina.

REDDY, J. N. (1993) *An Introduction to the Finite Element Method*, McGraw Hill.

REEVES, C. R., RAYWARD, V. J., OSMAN, I. H. & SMITH, G. D. (1996) *Modern Heuristic Techniques,* Chicester, John Wiley & Sons.

REWIENSKI, M. & WHITE, J. (2001) A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuit and Micromachined Devices. *Computer-Aided Design.* San Jose, California, USA.

REWIENSKI, M. & WHITE, J. (2002) Improving Trajectory Piecewise-Linear Approach to Nonlinear Model Order Reduction for Micromachined Devices Using an Aggregated Projection Basis. *Modeling and Simulation of Microsystems.* San Juan, Puerto Rico.

RUDOLPH, S. & KORMEIER, T. (2005) On Self-Similarity as a Design Paradigm. *ASME 2005 IDETC/CIE, Long Beach, CA, USA.*

SAXENA, A., WANG, X. & ANANTHASURESH, G. K. (2000) PennSyn: A Topology Synthesis Software for Compliant Mechanisms. *Proceedings of the DETC'00 ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Baltimore, Maryland, USA.

SCALES, L. E. (1985) *Introduction to non-linear Optimisation,* London, MacMillan Publishers Ltd.

SCHIEK, R. L. & SCHMIDT, R. C. (2005) Automated and Integrated Mask Generation from a CAD Constructed 3D Model. *Nanotechnology Conference and Trade Show - NSTI Nanotech.* Anaheim,California, USA.

SCHIEK, R. L. & SCHMIDT, R. C. (2006) Automated Surface Micro-Machining Mask Creation from a 3D Model. *Microsystem Technologies,* 12**,** 204-207.

SENTURIA, S. D. (1998) CAD Challenges for Microsensors, Microactuators and Microsystems. *Proceedings og IEEE,* 86**,** 1611-1626.

SENTURIA, S. D. (2001) *Microsystem Design*, Kluwer Academic Publisher.

SHEA, K. (1997) Essays of Discrete Structures: Purposeful Design of Grammatical Structures by Directed Stochastic Search. *Engineering Department.* Carnegie Mellon University.

SIARRY, P. & COLLETTE, Y. (2003) *Multiobjective Optimization*, Springer.

SIGMUND, O. & ', B. D. (1996) Some Inverse Problems in Topology Design of Materials and Mechanisms. IN KLUWER (Ed.) *Simposium on Optimisatrion of Mechanical Systems.* Schielen, Netherlands.

SIGMUND, O. (1997) On the Design of Compliant Mechanisms using Topology Optimisation. *Mechanism of Structures and Machine,* 25**,** 495-526.

SILVEIRA, L. M., KAMON, M., ELFADEL, I. & WHITE, J. (1999) A Coordinate-transformed Arnoldi Algorithm for Generating Guaranteed Stable Reduced-Order Models of RLC Circuits. *Conputer Methods in Applied Mechanics and Engineering,* 169**,** 377-389.

SIMON, H. A. (1969) *The Science of Artificial,* Cambridge, MA, MIT Press.

SOMMERVILLE, I. (1995) *Software Engineering,* UK, Addison Wesley.

SPALL, J. C. (2003) *Introduction to Stochastic Search and Optimisation*, Wiley.

SRIKAR, V. T. & SENTURIA, S. D. (2002) Thermoelastic Damping in Fine-grained Polysilicon Flexural Beam Resonators. *Journal of Microelectromechanical Systems,* 11**,** 499-504.

STARLING, A. & SHEA, K. (2003) From Discrete Structures to Mechanical Systems: A Framework for Creating Performance-based Parametric Synthesi Tools. *American Association for Artificial Intelligence Spring Symposium.* Stanford, California, USA.

STARLING, A. C. (2004) Performance-based Computational Synthesis of Parametric Mechanical Systems. *Enginnering Department.* Cambridge, University of Cambridge.

STEENEKEN, P. G., RUIGROCK, J. J. M., KANG, S., VAN BEEK, J. T. M., BONTEMPS, J. & KONING, J. J. (2007) Parameter Extraction and Support-loss in MEMS Resonators. *European COMSOL Conference 2007.* Grenoble, France.

STEIDEL, R. F. J. (1979) *An Introduction to Mechanical Vibrations,* New York, John Wiley & Sons.

SUGAR (1998) SUGAR MEMS SIMULATOR. Berkeley, California, USA, http://www-bsac.eecs.berkeley.edu/cadtools/sugar/sugar

SUPPAPITNARM, A., SEFFEN, K. A., PARKS, G. T. & CLARKSON, P. J. (1999) Design by Multiobjective Optimisation using Simulated Annealing. *ICED'99 International Conference On Engineering Design.* Munich, Germany.

SWART, N. R., BART, S. F., ZAMAN, M. H., MARIAPPAN, M., GILBERT, J. R. & MURPHY, D. (1998) AutoMM: Automatic Generation of Dynamic Macromodels for MEMS Devices. *IEEE Elenventh Annual International Workshop on Microelecreomechanical Sytems.*

TALBI, E. G. (1998) A Taxonomy of Hybrid Metaheuristic. Lille, France, University Of Lille.

THOMSON, W. T. (1993) *Theory of Vibration with Applications,* London, Chapman & Hall.

TIMOSHENKO, S. P. & GOODIER, J. N. (1970) *Theory of Elasticity,* Singapore, McGraw-Hill.

TUSHAR, U. (2005) Accurate and Robust Geometric Modeling for Simulation Of IC and MEMS Fabrication Processes. *2005 ACM Symposium on Solid and Physical Modeling.* Cambridge, Massachussets, USA.

ULRICH, K. T. & EPPINGER, S. D. (1995) *Product Design and Development*, McGraw Hill.

VALE, C. A. W. (2002) Multiobjective Dynamic Synthesis via Machine Learning. Cambridge, UK, University of Cambridge.

VALE, C. A. W. (2003a) The AISearcher Algorithm – general Principles and Implementation. Cambridge, UK, University of Cambridge.

VALE, C. A. W. (2003b) Connected-Node System Design Representation: Technical Implementation. Cambridge, UK, University of Cambridge.

VALE, C. A. W. & SHEA, K. (2003c) MEMS Design Using a Connected- Node Design Representation: Preliminary Report. Cambridge, UK, University of Cambridge.

VALE, C. W. A. & SHEA, K. (2003d) Learning Intelligent Modification Strategies in Design Synthesis. *AAAI Symposium.* Standford, California, USA, AAAI Press.

VALE, C. W. A. & SHEA, K. (2003e) A Machine Learning-based Approach to Accelerating Computational Design Synthesis. *ICED 2003, International Conference of Engineering Design.* Stockholm, Sweden.

VAN KUIJK, J., SCHROPER, S. D. & DASILVA, M. (2005) Design Automaton for MEMS/MST. *Design, Automation and Test in Europe COnference & Exhibition.* Munich, Germany.

WANG, F. & WHITE, J. (1998) Automatic Model Order Reduction of a Microdevice using the Arnoldi Approach. *ASME IMECE '98 DSC.* Anaheim, California, USA.

WANG, K., WONG, A. C. & NGUYEN, C. T. C. (2000) VHF Free-Free Beam High-Q Micromechanical Resonators. *Journal of Microelectromechanical Systems,* 9**,** 347-360.

WEAVER, W. J., TOMOSHENKO, S. P. & YOUNG, D. H. (1990) *Vibration Problems in Engineering,* New York, John WIley & Sons.

WILSON, L. The Mathematical Description of Shapes and Form.

WOOD, W. H. & DYM, C. L. (2003) Integrated Computational Synthesis: a Design Perspective. *AAAI Spring Symposium on Computational Synthesis.* Standford, California, US.

YAN, J. (2004) MEMS Resonator Design. Cambridge, University of Cambridge Engineering Department.

YAN, J., SESHIA, A. A., STEENEKEN, P. G. & VAN BEEK, J. T. M. (2006) A Silicon MEMS Bulk Mode 'Sandwich' Resonator. *17th European Workshop on Micromachining, Micromechanics and Microsystems (MME 2006).* Southampton, UK.

YANG, J. & ONO, T. (2002) Energy Dissipation in Submicrometer Thick Single-crystal Silicon Cantilevers. *Journal of Microelectromechanical Systems,* 11**,** 775-783.

YAN, J. (2007) Micro/Nano-Electromechanical Resonatos for Signal Processing. *Engineering Department.* Cambridge, University of Cambridge.

YANG, Y. J., KAMON, M., RABINOVICH, V. L., GHADDAR, C., DESHPANDE, M., GREINER, K. & GILBERT, J. R. (2001) Modeling Gas Damping and Spring

Phenomena in MEMS with Frequency Dependent Macro-Models. *14th IEEE International Conference on MEMS.* Interlaken, Switzerland.

YANG, Y. J., CHENG, S. Y. & SHEN, K. Y. (2004) Macromodeling of Coupled-Domain MEMS Devices with Electrostatic and Electro Thermal Effects. *Journal of Micromechanics and Microengineering,* 14**,** 1190–1196.

YANG, Y. J. & YU, C. C. (2004) Extraction of Heat-transfer Macromodels for MEMS Devices. *Journal of Micromechanics and Micorengineering,* 14**,** 587-596.

YASUMURA, K. Y., STOWE, T. D., CHOW, E. M., PFAFMAN, T., KENNY, T. W., STIPE, B. C. & RUGAR, D. (2000) Quality Factors in Micron and Submicron-thick Cantilevers. *Journal of Microelectromechanical Systems,* 9**,** 117-125.

YIN, L., ANANTHASURESH, G. K. & BESTLE, D. (2001) Topology Optimisation of Compliant Mechanisms with Multiple Materials using a Peak Function Material Interpolation Scheme. *Structural and Multidisciplinary Optimization,* 23**,** 49-62.

ZHOU, N., ZHU, B., AGOGINO, A. M. & PISTER, K. S. J. (2001) Evolutionary Synthesis of MEMS MicroElectronicMechanical Systems Design. *Proceedings of the Artificial Neural Networks in Engineering (ANNIE2001).* St Louis, USA, ASME Press.

ZHOU, N. (2002) Simulation and Synthesis of MicroElectronicMechanical Systems. *Department of Engineering.* Berkeley, California, USA, UC Berkeley.

ZHOU, N., AGOGINO, A. M. & PISTER, K. S. (2002) Automated Design Synthesis for Micro-Electro-Mechanical Systems (MEMS). *ASME Design Automation Conference.* Montreal, Canada.