Title
Knowledge-Based Evolutionary Linkage in MEMS Design Synthesis

Authors
Cobb, Corie L.
Zhang, Ying
Agogino, Alice M.
et al.

Publication Date
2008

Knowledge-Based Evolutionary Linkage in MEMS Design Synthesis

Corie L. Cobb[1], Ying Zhang[2], Alice M. Agogino[1], and Jennifer Mangold[1]

[1] Mechanical Engineering Department at the University of California, Berkeley, CA 94720, USA
ccobb@berkeley.edu, agogino@berkeley.edu, jam@me.berkeley.edu
[2] School of Electrical and Computer Engineering at the Georgia Institute of Technology, Savannah, GA 31407, USA
yzhang@gatech.edu

Abstract. Multi-objective Genetic Algorithms (MOGA) and Case-based Reasoning (CBR) have proven successful in the design of MEMS (Micro-electro-mechanical Systems) suspension systems. This work focuses on CBR, a knowledge-based algorithm, and MOGA to examine how biological analogs that exist between our evolutionary system and nature can be leveraged to produce new promising MEMS designs. Object-oriented data structures of primitive and complex genetic algorithm (GA) elements, using a component-based genotype representation, have been developed to restrict genetic operations to produce feasible design combinations as required by physical limitations or practical constraints. Through the utilization of this data structure, virtual linkage between genes and chromosomes are coded into the properties of predefined GA objects. The design challenge involves selecting the right primitive elements, associated data structures, and linkage information that promise to produce the best gene pool for new functional requirements. Our MEMS synthesis framework, with the integration of MOGA and CBR algorithms, deals with the linkage problem by integrating a component-based genotype representation with a CBR automated knowledge-base inspired by biomimetic ontology. Biomimetics is proposed as a means to examine and classify functional requirements so that case-based reasoning algorithms can be used to map design requirements to promising initial conceptual designs and appropriate GA primitives. CBR provides MOGA with good linkage information through past MEMS design cases while MOGA inherits that linkage information through our component-bsased genotype representation. A MEMS resonator test case is used to demonstrate this methodology.

1 Introduction

Microelectromechanical Systems (MEMS) are small micro-machines or micron-scale electro-mechanical devices that are fabricated with processes adapted from Integrated Circuits (ICs). Although still a relatively new research field, MEMS devices are being developed and deployed in a broad range of application areas, including consumer electronics, biotechnology, automotive systems and aerospace. Example MEMS devices include accelerometers in automotive airbags and micro-mirrors for optical switching in data communication networks. As MEMS devices grow in complexity, there is a greater need to reduce the amount of time MEMS designers spend in the

Y.-p. Chen, M.-H. Lim (Eds.): Linkage in Evolutionary Computation, SCI 157, pp. 461–483, 2008.
springerlink.com

initial conceptual stages of design by employing efficient computer-aided design (CAD) tools.

Working with a multidisciplinary research team at the Berkeley Sensor and Actuator Center (BSAC), our work with Evolutionary Computation (EC) is focused on the conceptual design of MEMS devices. Zhou et al. [1] were the first to demonstrate that a multi-objective genetic algorithm (MOGA) can synthesize MEMS resonators and produce new design structures. SUGAR [2], a MEMS simulation tool, was used to perform function evaluations on constraints and fitness values. Kamalian et al. [3] extended Zhou's work and explored interactive evolutionary computation to integrate human design expertise into the synthesis process. They also fabricated and tested the emergent designs in order to characterize their mechanical properties and identify deviations between simulated and fabricated features [4]. Zhang et al. [5, 6] implemented a hierarchical MEMS synthesis and optimization architecture, using a component-based genotype representation and two levels of optimization: global genetic algorithms (GA) and local gradient-based refinement. Cobb et al. [7] created a case-based reasoning (CBR) tool to serve as an automated knowledge base for the synthesis of MEMS resonant structures, integrating CBR with MOGA [8] to select promising initial designs for MOGA and to increase the number of optimal design concepts presented to MEMS designers.

In related research, Muhkerjee et al. [9] conducted work on MEMS synthesis for accelerometers using parametric optimization of a pre-defined MEMS topology. They expanded the design exploration within a multidimensional grid in order to find the global optimal solution. Wang's [10] approach to MEMS synthesis utilized bond graphs and genetic programming with a tree-like structure of building blocks to incorporate knowledge into the evolutionary process, similar to work by Zhang [6]. Li et al. [11] concentrated on developing automated fabrication process planning for surface micromachined MEMS devices that relieves designers from the tedious work of process planning so they can concentrate on the design itself. MEMS CAD has matured to the point that there are now commercial CAD programs, such as Comsol® and IntelliSuite®, that offer MEMS designers pre-made modules and cell libraries, but there is little automatic reasoning in place for the user on how and when these components should be used.

Our EC method employs a genetic algorithm as the evolutionary search and optimization method. GAs were introduced by Holland [12] to explain the adaptive processes of evolving natural systems and for creating new artificial systems in a similar way, and Goldberg [13] further demonstrated how to use them in search, optimization, and machine learning. Chen et al. [14] noted that traditional GAs require users to possess prior domain knowledge in order for genes on chromosomes to be correctly arranged with respect to the chosen operators. The performance of a GA is heavily dependent upon its encoding scheme. When prior domain knowledge is available, the design problem can be solved using traditional genetic algorithms. However, that is not always the case, and this is when methods such as linkage learning are needed. Chen [15] and Harik [16] both focused research efforts on the linkage learning genetic algorithm (LLGA) so that a GA, on its own, can detect associations among genes to form building blocks [15].

Linkage is an important part of GA performance. Tightly linked genes are synonymous with building blocks, but higher level linkage amongst building blocks is also necessary to ensure successful design solutions are reached. We propose an integrated MEMS design synthesis system which combines CBR with biologically inspired classifications and an evolutionary algorithm, MOGA, to help generate more varied conceptual MEMS design cases for a designer and her/his current design application.

In this chapter, we will explain our micro-resonator test case which will be highlighted throughout our work to explain our linkage concept. Next, we discuss MOGA and CBR and explain how linkage is achieved through our knowledge-based evolutionary algorithm. Lastly, we present a review of symmetry patterns observed in nature, as they pertain to resonant frequency-sensitive biological creatures, and explore the role that symmetry plays in our evolutionary synthesis process for the resonator example.

2 Evolutionary Computation for Resonant MEMS Design

2.1 MEMS Resonator Test Case

To date, our MEMS design synthesis program has focused on the design of resonant MEMS. A schematic of a MEMS resonator and its component decomposition are shown in Fig. 1. These designs have consisted of a fixed center mass (either with or without electrostatic comb drives) connected to four 'legs', each made up of multiple beam segments. We evaluated our MOGA synthesis program for several sets of performance objectives all calculated using the SUGAR simulation program.

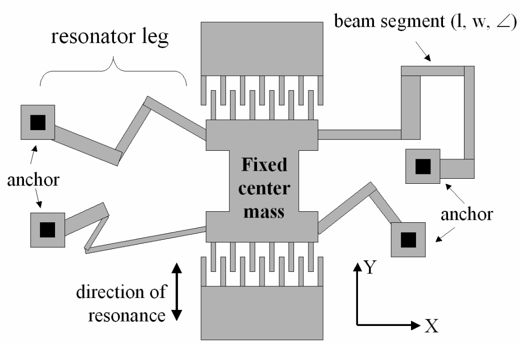


Fig. 1. Schematic of example resonator synthesis problem. The geometry of the center mass is fixed, while the number of beam segments per leg and the size and angle of each segment is variable [3].

As we are designing resonators, the most significant performance objective for all structures is the resonant frequency (f_0). Resonant frequency is the most critical requirement because if a resonator deviates too far from its frequency target it is essentially a useless design. Other performance objectives we have used for synthesis include the stiffness of the structure in the x or y-direction as well as the device area (defined by a bounding rectangle around the device).

2.2 SUGAR: MEMS Simulation with Modified Nodal Analysis

SUGAR [2] is an open-source MEMS simulation tool based on modified nodal analysis (MNA), allowing a designer to quickly prototype and simulate several complex MEMS structures for preliminary design applications.[1] Finite element analysis (FEA) calculations could take hours per simulation, making them infeasible for iterative design processes on complex systems. SUGAR and other similar lumped parameter nodal analysis simulation tools can perform these functional calculations with reasonable accuracy at a fraction of the time and can therefore allow the MEMS designer to explore larger design spaces. FEA and parametric optimization can then be used to refine the most promising of the design concepts produced by the MOGA evolutionary process.

2.3 Linkage with Component-Based Genotype Representation

Genetic linkage, in biological terms, refers to the relative position of two genes on a chromosome. Two genes are linked if they are on the same chromosome and are tightly linked if they are physically close to each other on the same chromosome. Genes that are closely linked are usually inherited together from parent to offspring [14]. Our MOGA data structure can be classified as "linkage adaptation" if we use the same terminology as Chen [14]. Linkage adaptation refers to specifically designed representations, operators, and mechanisms for adapting genetic linkage along with the evolutionary process. Chen states that linkage adaptation techniques are closer to biological metaphors of evolutionary computation because of their representations, operators, and mechanisms.

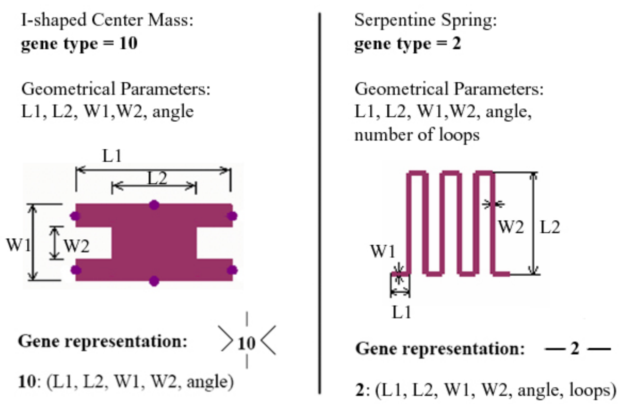
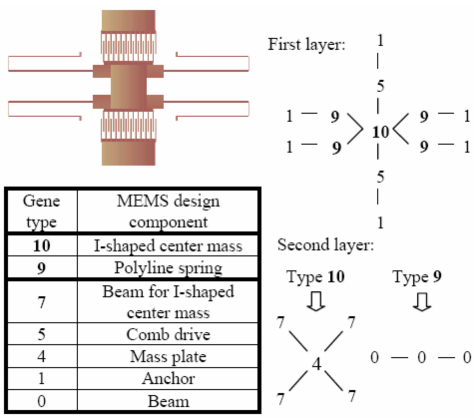


Fig. 2. Gene representation examples for MEMS building blocks [6]

---

[1] SUGAR can be accessed from:
http://sourceforge.net/projects/mems/

Our component-based genotype representation for MEMS design synthesis is supported by a hierarchical extendible design component library developed by Zhang et al. [6]. Each MEMS design component type is represented by a gene. This gene carries all salient information about the component: its geometric layout parameters, as well as constraints on how the component can be modified and what genetic operations can be applied to it (see Fig. 2). Each gene has external nodes through which components are connected and registered to one another. Two genes are on the same chromosome, which represents a design cluster or a simple MEMS design, if one of them can be reached from the other through any linkage path in the chromosome. Two genes are tightly linked if they share the same external node. For example, in Fig. 3 gene types 10 and 9 are tightly linked because they share the same external node and gene types 10, 9, 5, and 1 are on the same chromosome because each gene can be found by tracing the linkage path in the design.



Gene type	MEMS design component
10	I-shaped center mass
9	Polyline spring
7	Beam for I-shaped center mass
5	Comb drive
4	Mass plate
1	Anchor
0	Beam

Fig. 3. MEMS resonator gene representation [6]

A designer can predefine what gene types are allowed to be closely linked to a specific gene type and whether a position on the chromosome is a crossover point during the evolutionary process by associating special properties to certain linkage nodes in the chromosome. Based on predefined rules, the mutation operation can be applied at either the gene level or the chromosome level, providing a probability of changing linkage with the mutation operation during the evolutionary process.

3 Case-Based Reasoning and Biomimetic Inspired Ontology

Case-based Reasoning (CBR) is an artificial intelligence method that utilizes knowledge from a past situation to solve current problems. Shank's dynamic memory model [17] is regarded as the foundation for CBR. Kolodner used Shank's model to create the first CBR system called CYRUS which was a basic question and answer system [18]. CBR has been applied to a broad array of domains ranging from cooking recipes to the design of electro-mechanical devices. For example, Kritik [19], a

CBR system developed in the early 1980s, generated designs for physical systems such as electrical circuits. The first successful industry application of CBR was CLAVIER [20] which was used by Lockheed Martin for determining successful loads of composite material parts for curing in an autoclave. More recently, CAFixD [21] applied the principles of CBR to fixture design for various machining operations.

CBR is analogous to human cognition and thought processes; cases can be regarded as “memories,” while retrieval is similar to “reminding” one of a particular instance, and case representation is how one’s memories are organized. CBR involves indexing past knowledge, in the form of “cases” to enable effective retrieval of solutions for a current problem. Indexing and case representation are the two initial and most important stages of CBR, determining the ultimate performance of a CBR program.

In the context of our work, CBR takes advantage of previous human knowledge in the form of successful MEMS design cases to help guide humans and computational design tools towards more optimal design concepts. Previous work by Cobb et al. [8] has shown that the integration of a CBR knowledge base with a multi-objective genetic algorithm (MOGA) can increase the number of optimal solutions generated for a given MEMS design problem. CBR is used to help select the best candidates to be evolved in an evolutionary process such as MOGA. In the following sections, we will examine the biological analogs of case representation and indexing as well as how they can support linkage in MOGA.

3.1 Case Representation and Biological Taxonomy

Biological classification or taxonomy is a means by which biologists group and classify organisms. Taxonomy helps one identify evolutionary relationships and links between certain species and in the case of MEMS, certain design structures. Classifying organisms based on shared physical traits is how taxonomy began, but these classifications have been modified over the years to reflect Darwinian evolutionary relationships. Spiders are of interest to our work due to the parallels their physical appearance has to our MEMS resonator example. Biologists have classified over 40,000 species of spiders, but they believe there are still thousands of species which have not yet been identified and named. As more species are discovered the current biological classification system can expand and change.

The classification of animals and plants is inherently hierarchical; similar to the way our MEMS case library is hierarchical to demonstrate the relationships between different designs. The 40,000 species of classified spiders are further divided into three suborders with 38 families and 111 subfamilies. The groups described by taxonomy get more specific as one goes from the kingdom classification all the way down to the species group. Kingdom is the largest unit of classification (with approximately five kingdoms), phylum is the next unit of classification which further divides each kingdom, and this pattern continues down to the species level, forming a tree like hierarchy of organism representation. No two species of spiders, or any plant or animal, will have the same scientific name (defined by the genus and species). The scientific name is a unique identifier just as each unique MEMS design component has a distinct identification number and gene type to distinguish it from other designs and enable efficient case retrieval.

MEMS is still an exploratory field and new designs and pieces of the MEMS hierarchy are constantly being added, similar to the way newly discovered species of organisms are expanding the biological taxonomy system everyday. Varadan [22] noted that it is still premature today to create a robust categorization due to the fact that many MEMS devices are still in the research phases of development and have not matured for every application. MEMS categorization has often focused on fabrication methods and materials selection, geometry, or application areas [23]. There are a broad array of MEMS sensors and actuators available today. Bell et al. [24] categorized MEMS by considering work-producing actuators, force sensors and displacement sensors fabricated by surface or bulk micromachining in their work and did an in depth classification of these devices.

In MEMS, designs are often classified based on their performance and functional characteristics. Sensors and actuators are the two most broad and commonly agreed upon categories of MEMS which can be divided further into families and classes. Similar to the work of Bell et al. [24], we will have two kingdoms in our classification system: sensors and actuators. Sensors and actuators can each be further divided into phylum or classes based upon their operating domains. For our purposes, we will assume six operating domains based upon input and output signals MEMS devices utilize: (1) Magnetic, (2) Thermal, (3) Electrical, (4) Mechanical, (5) Chemical, and (6) Optical.

Imagine the aforementioned domains placed in a 6 by 6 matrix (with all six categories each lined up on the rows and columns) to enable multiple input and output combinations. For example, a thermal-mechanical sensor might take a thermal input and have a mechanical deflection as its output. For a piezoelectric sensor, it will output a voltage in response to an applied mechanical stress, enabling a further categorization of the mechanical-electrical class. Because the user of our CBR program may be searching for designs based on input and output domains or application areas, it is important to index cases by both. Our MEMS hierarchy starts with sensors and actuators, and then branches out to the various input and output mechanisms, and under each of these are specific application areas (RF MEMS, Micro-fluidics, BioMEMS, Optical MEMS, etc.), and then divided further are whole MEMS devices, which are broken down into their various components and primitive elements.

Currently, our work focuses on resonant structures, such as resonators, accelerometers and micromechanical filters. Thus, in traversing the MEMS hierarchy, our work falls under the electrical (input and output domain) where electrostatics are primarily used. Fig. 4 is a condensed MEMS taxonomy graph, and is not inclusive of all MEMS devices. The portion shown demonstrates how the classification leads to accelerometers, filters, and resonators – the focus of our work. Resonators, the basic components of filters, can be further decomposed into masses, springs, comb drives, and anchors. Each one of the aforementioned components would have a unique identifier to distinguish them from others. Nguyen [25] classifies MEMS filters based on their ability to achieve a certain frequency range, an important part of being able to develop RF communication devices.

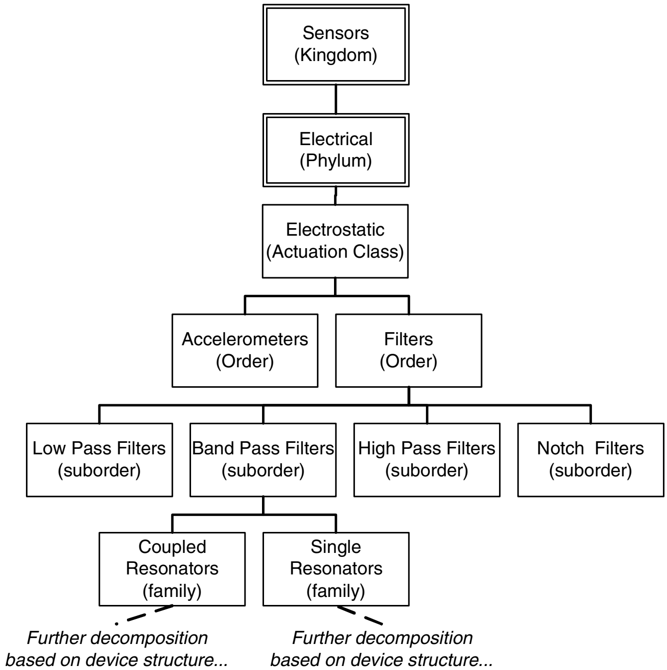


Fig. 4. MEMS hierarchy example with biological analogy

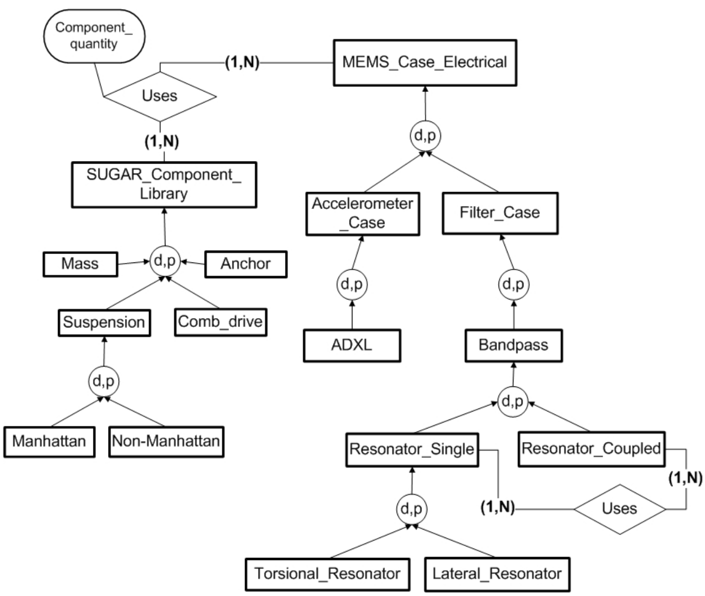


Fig. 5. MEMS database design

A hierarchy for biological organisms was created just as a hierarchy for CBR needs to be created in order to sort information and efficiently pull the most relevant primitives and designs for evolutionary computation. Ontology is a way to represent knowledge in a specific domain, helping an artificial intelligence (AI) program to define and retrieve objects. A general hierarchy or structure of ontology is the following [26]: objects, classes of objects, attributes of objects, and relations between objects. Shown in Fig. 5 is our current MEMS case library ontology. Using entity-relationship diagram notation, one can observe how objects such as MEMS resonators and filters are related together. In the diagram,'d,p' indicates a disjoint/distinct and partial relationship between classes, in order to account for designs that have not yet been created or added to the library. Attributes of each object include indices for quick retrieval and overall device performance. Our current CBR hierarchy classifies designs based on their shared functionality and performance.

3.2 Creating Evolutionary Linkage with Case-Based Reasoning

Linkage, as defined by Chen [14], refers to placing related genes close together on a chromosome. The GA programmer seeds the GA with initial designs with implicit linkages. The GA programmer may be adding her/his expertise to the codification in this process. This may be difficult to do, however, on new design problems in which the programmer has limited experience.

Applying the aforementioned definition to MEMS synthesis, we use the concept of linkage to refer to how closely MEMS building blocks should be linked in an evolutionary process. With the integration of CBR and MOGA (see Fig. 6), CBR defines the linkages for the user with an automated case-based library of previous MEMS

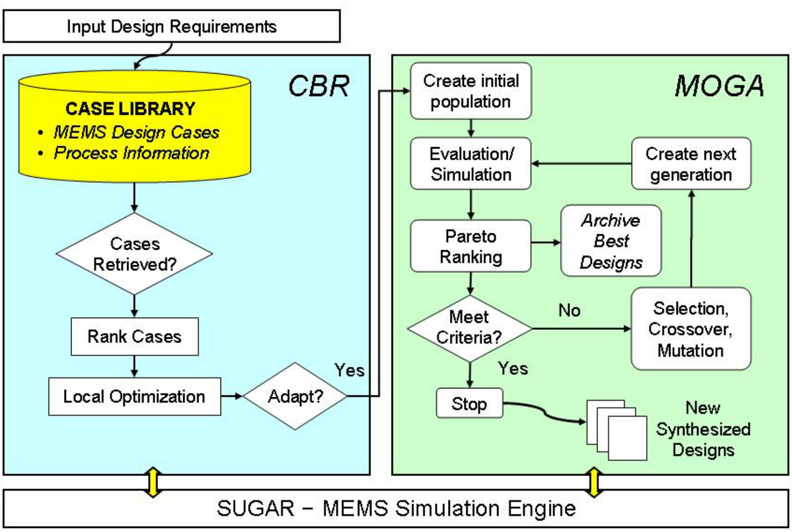


Fig. 6. MEMS design synthesis architecture

designs. CBR takes away from the user the burden of defining the problem by automatically selecting and optimizing design structures based on a few inputted design requirements. In the absence of CBR or a good seed design, MOGA may not converge to a design solution. Zhang et al. [5] noted that seeding MOGA with a good initial design is essential to helping MOGA converge to better design solutions in a practical number of evolutions. CBR can pull out the design cases close to local design optima for a given scenario. The designs are ranked according to the user's design requirements and are then encoded in the component-based genotype representation to enable the evolutionary process. Incorporating other powerful computational tools, such as CBR, with MOGA can help MOGA converge faster and more efficiently to optimal design concepts. The linkage problem is alleviated in our MOGA program because CBR inherently defines linkage for MOGA with its case examples.

CBR assists MOGA by propagating the linkage of effective building blocks and selecting designs near local optima. In a previous experiment [8], for each MOGA synthesis run, we used a population of 400 for 50 generations. Using constraint cases of (1) no symmetry, (2) y-axis symmetry, and (3) x- and y-axis symmetry, five runs of the MOGA process were conducted for each constraint case in order to see a good spread of design solutions. We found that when MOGA is seeded with good starting designs from CBR, in some instances, y-axis symmetry and x- and y-axis symmetry constraints generate more pareto optimal designs over 50 generations.

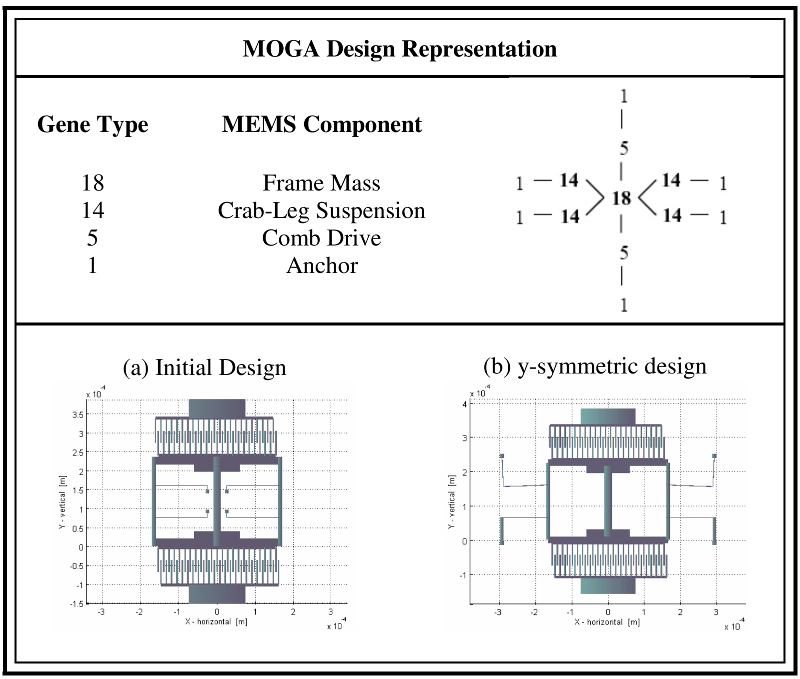


Fig. 7. Resonant frequency = 23.8 kHz for initial MOGA design (a); Resonant frequency = 24.8 kHz for a pareto optimal y-symmetric design generated by MOGA (b)

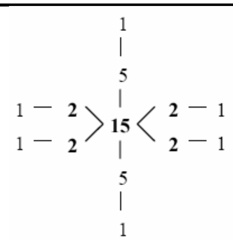
Shown in Fig. 7 is an example of tight linkage generated by our integrated CBR and MOGA program. The design requirements for this scenario were the following: $f_0 = 24.9$ kHz, $K_x/K_y \geq 8$, $Area \leq 2.1\text{e-}7\ \text{m}^2$. Eight designs were selected by CBR for a MOGA synthesis process for this given scenario. Because all eight CBR retrieved designs had similar linkage properties, we will highlight the best design here which was a resonator with an enclosed frame mass and crab-leg suspensions (two beams with a local 90 degree angle). For the best design shown in Fig. 7a, the mass and comb drives remained fixed while the crab-leg suspensions (which have the largest impact on the performance objectives) were allowed to change in width, length, and global orientation, but the crab-leg suspensions retained their local 90 degree angle.

As one can see in Fig. 7, the initial design in Fig. 7a generated an optimal design (Fig. 7b) which had the leg suspensions rotated outside of the frame mass. One would assume that if the objective is to minimize area, the suspensions would remain inside the mass, similar to the initial design in Fig. 7a. However, because frequency and stiffness were also part of the optimization problem, MOGA determined that a design with the suspensions outside of the mass could produce a better resonant frequency and stiffness ratio. The resonator design in Fig. 7b may have not been considered by a human MEMS designer, but due to the linkage knowledge CBR gave MOGA, the design is a good candidate for further analysis and fabrication.

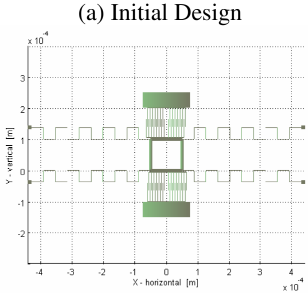
Fig. 8 shows another example of tight linkage in our MOGA process. The design requirements for this scenario are the following: $f_0 = 8.3$kHz, $K_x/K_y \geq 29$, $Area \leq$

MOGA Design Representation

Gene Type	MEMS Component
15	Hollow Ring Mass
2	Serpentine Suspension
5	Comb Drive
1	Anchor



(a) Initial Design



(b) y-symmetric design

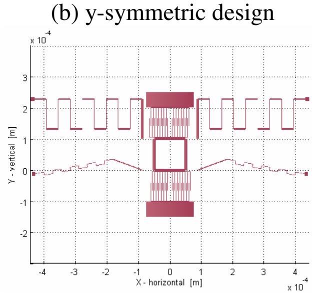


Fig. 8. Resonant frequency = 6969.3 Hz for initial MOGA design (a); Resonant frequency = 8299.9 Hz for a pareto optimal y-symmetric design generated by MOGA (b)

3.7e-7 m^2. In this particular case there was only one design selected by CBR which consisted of a hollow squared shaped mass with four serpentine springs. Again, the mass and comb drives remained fixed while the serpentine suspension blocks were free to mutate in length, width, number of loops, and their global angle orientation. This scenario also generated designs that had a similar appearance to spiders and insects (aside from the inherent manhattan geometry in the building blocks). Minimizing area is our main design objective for all of the designs in this experiment. The y-symmetry (symmetry around the vertical axis) constraint cases had the smallest design area average (2.608E-7 m^2) with a standard deviation of 7.031e-8 m^2.

4 Biomimetics: Role of Symmetry and Resonance

Applying manhattan geometries (90º angles) and symmetry constraints greatly reduces the search space and allows MOGA to optimize its search over a more manageable size. If convergence can be achieved, however, fewer constraints are preferred in an optimization problem as it broadens the search space to a wider selection of solutions. When MOGA runs unconstrained or with only symmetry constraints, the results produce designs that greatly differ from those designed by humans. Upon observation, these designs have an uncanny appearance to spiders, insects, and other organisms observed in nature. This prompted us to examine the biological analogies that exist between our EC generated resonators and biological organisms to help us understand which symmetry and geometric constraints might be an evolutionary advantage of natural life forms that use vibration or natural frequencies to survive.

4.1 Symmetry and Geometric Constraints

Symmetry is evident throughout the natural world – a butterfly's wings, a spider's web, and even physicists observe symmetry in distant galaxies. Symmetry has been used to try to understand the physical world since ancient times [27]. In the animal kingdom, bilateral symmetry is found in more complex species, where different parts of the animal's body perform different functions. Radial symmetry can be found in simpler life forms, such as starfish, where the entire body performs most of the life functions.

Symmetry has typically been a sign of quality in nature, and symmetry perception has been demonstrated in humans, animals, and insects. Many studies have concluded that humans and other species find symmetrical patterns more favorable than asymmetrical ones. It has been suggested that preferences for symmetry adapted for reasons related to mate choice. For several species, females prefer a mate that has more symmetrical characteristics [28]; experiments performed with insects and birds found that females prefer to mate with males who have the most symmetrical ornaments [29]. Enquist and Arak [30] suggest that the preference for symmetry has evolved from the need to recognize objects no matter what their position or orientation may be. This preference for symmetry is prominent in the MEMS world where many designers highly favor symmetrical layouts and manhattan style geometry. In previous work, some of our nontraditional asymmetric MEMS designs were fabricated and characterized to help improve EC algorithms, and it was shown that the fabricated design behaved within reasonable agreement to simulation results [4].

Of the forms of symmetry in the animal world, bilateral symmetry is much more common than full symmetry. Even with bilateral symmetry, organisms often reflect a behavioral asymmetry with internal organs or a tendency for right or left-handedness as noted by Babcock [31]. Asymmetry is less prevalent in the natural world but can be observed in a select few organisms such as sponges (poriferans). In biology studies by Moller et al. [32], they found that growth rate and fluctuating asymmetry are negatively correlated, meaning asymmetric animals grow less rapidly than symmetric ones. Although organisms may exhibit bilateral and radial symmetry, most organisms have some type of observable asymmetry.

In the MEMS world, designers are tasked with developing physical forms that satisfy multiple functional requirements. It is tempting to think that simple designs with 90 degree angles are better than designs with irregular or nontraditional layouts. This can be the case in macroscale designs where non-perpendicular and parallel designs can be time-consuming and expensive from a manufacturing point of view. But in MEMS fabrication, lithography processes enable a designer to create almost any geometrical layout and all are equally easy to fabricate, with the only obstacle being the resolution capabilities of the lithography process, impacting the minimum size of features that can be fabricated.

Kamalian et al. [3] previously noted that optimal MEMS designs with multiple competing objectives need not have full symmetry or manhattan angles, but may benefit from symmetry about one axis – bilateral symmetry. Similar to our EC generated MEMS resonators, spiders have a large central mass and a similar number of legs on either side of their body. Spiders have evolved to have some degree of bilateral symmetry around the longitudinal axis, but none around the horizontal axis, similar to our y-symmetric resonator designs shown in Fig. 9. All species of spiders have a broad range of leg shapes, but none of them have manhattan geometries and most exhibit symmetry about only one axis.

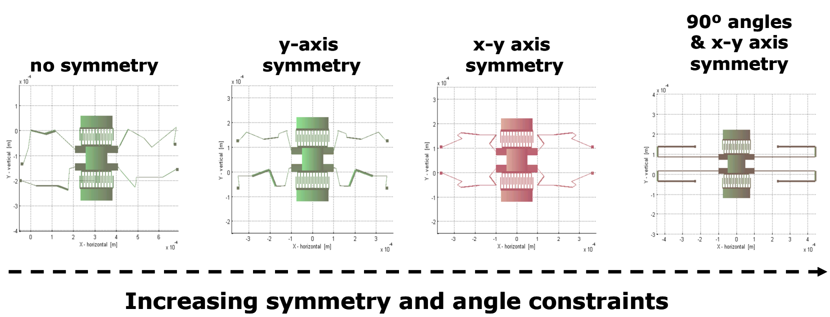


Fig. 9. Examples of MEMS resonator designs with increasing constraints

4.2 Purpose of Resonance and Vibration

We can further examine the spider as a biological analog to a resonator in its ability to detect prey by resonating with their vibrations. Vibration cues have been used by insects and spiders to locate and kill their prey. Without the use of vibration recognition, it may be difficult for insects to find their prey, because dense vegetation may limit their visual abilities. Vibration signals are also important, because many of the insect's or spider's prey produce vibrations through movement or feeding, which enables them to be located more easily [33].

Bola spiders catch their prey by mimicry, emitting the pheromones of the prey species. The wing-beat vibrations of the moths that fall victim to the bola spiders stimulate the spider to make a bolas in which to capture the moth [34]. Generally all web-spinning spiders detect and find prey in their webs through the vibrations generated by their prey. This is especially important because most species of web spiders do not have a strong sense of smell or good vision. Peters (1931) found that the spiders did not respond to a dead fly placed gently in its web. If, however, the fly arrived in the web with a jerk or if, once in the web, it was stimulated in some way, the spider responded [35]. There is a good deal of evidence that spiders discriminate between different types of signals. There have been several studies that demonstrated how spiders move towards vibrations of various frequencies, similar to the way MEMS resonators and bandpass filters attempt to hone in on certain frequencies for communication purposes. Resonators, which are basic building blocks of MEMS filters, are designed to reject certain frequencies from a wide range of signals and only allow a particular frequency band to pass through.

An important aspect of resonance in MEMS and nature is movement. Blickhan and Full [36] conducted a study of multi-legged locomotion in animals as diverse as cockroaches and kangaroos in order to develop a model of "legged terrestrial locomotion." They found that the dynamics of movement depend on the number of legs one has and the gait or movement pattern. Four- and six-legged creatures had greater whole body stiffness than two-legged creatures. The greater whole body stiffness in the four- and six-legged creatures resulted in higher natural frequencies, just as a higher overall stiffness results in a higher natural frequency in MEMS designs. Spiders generally have eight legs while insects have six legs. In MEMS, we mostly observe resonators with four main legs for stability. There are resonators with only two legs, but these tend to be slightly unstable with a tendency towards out of plane movement. In spiders, eight legs can enable them to move faster and give them the ability to travel in different directions easily. Some insects with six legs have a tendency to move forward more and not backwards and sideways as quickly as spiders. In our MEMS resonator design, we only want to move in one direction based on the comb drive actuation, hence four legs provides more balance and stability than two legs. Additional legs are not needed because in these MEMS resonator designs, motion in multiple directions is undesirable. However, if we look more broadly at other MEMS designs, such as micro-robots, more legs can be desirable to enable quick and easy movement in multiple directions.

After 3.8 billion years of "research and development," nature has discovered what works, what does not, and what is considered life sustaining, optimizing natural designs to meet the necessary functional needs. These "successful designs" are ever-changing to meet environmental requirements and are driven by an ultimate challenge: survival. Nature's solutions are sometimes not perfect; however they are solutions that are as good as they need to be to serve their intended purpose.

5 MEMS Case Study: An Analysis of Symmetry Constraints and Impact on Resonance

In the previous section, we looked at symmetry and resonance in nature. Since our synthesis system focuses on the structural design of MEMS, it is important to examine the different types of constraints we can embed in our MOGA linkage structure in order to produce the best performing MEMS designs. To better understand what role symmetry constraints we observe in nature have in our MOGA algorithm, an experiment with our resonator test case is performed to explore which combinations of symmetry and geometric constraints might produce the best performing micro-resonator designs.

5.1 Experiment Setup

In this experiment we enforce four different sets of constraints on our micro-resonator test case. Each mico-resonator is constructed of a 2μm thick layer of polysilicon material. The comb drives and center mass for the micro-resonator design are fixed while the springs are free to mutate, subjected to the following symmetry and angle constraints:

- ***C1:*** No symmetry or geometric constraints
- ***C2:*** Symmetry is enforced along the y-axis of the design (analogous to bilateral symmetry observed in organisms)
- ***C3:*** Symmetry is enforced about the x- and y-axis of the design
- ***C4:*** Symmetry is enforced about the x- and y-axis of the design and the suspensions (also known as 'legs') are restricted to 90º angles (analogous to how human designers traditionally create MEMS)

We place emphasis on symmetry constraints as these are most common types of structural constraints observed in nature. C4 includes a manhattan angle constraint and represents the typical constraints a human MEMS designer will impose upon the design of a resonant structure. Our goal is to better understand under what conditions symmetry that is found to be optimal in nature is also optimal in our MEMS resonator

Table 1. Polyline spring design parameters used for the MEMS resonator case study (*100μm only used for the 10kHz test case)

Mutation constraints for Polyline Spring	Parameter Value
Max. number of beams	7
Min. number of beams	1
Max. beam length	100μm/300μm*
Min. beam length	10μm
Max. beam width	10μm
Min. beam width	2μm

example using the MOGA algorithm. The resonator legs begin symmetrically from a center mass but are allowed to evolve with any number of joints in the legs (see Table 1 for leg design parameters). C1 has no symmetry constraints while C2 has the minimal bilateral constraints. C3 and C4 both have full symmetry, with C4 having the additional constraint of manhattan geometry. We wish to explore how these cases of minimal constraints (C1 and C2) compare to those with maximal constraints (C3 and C4).

The feasible design range for our initial resonator design is a resonant frequency (f_0) between 5-15 kHz and a stiffness ratio (K_x/K_y) between 1-10. The main design objective is the minimization of device area while achieving the required stiffness ratio and keeping the resonant frequency deviation to less than 5%. To explore the range of possible designs, four sets of design requirements for the micro-resonator were randomly generated, using the aforementioned bounds, and then used in a MOGA synthesis run (see Table 2). For comparison purposes, these results are included with the a previous design requirement test case used by Kamalian [3] and Zhang [5] where the resonant frequency target was 10 kHz and the stiffness in the x-direction only had to be greater than the stiffness in the y-direction.

Table 2. Randomly Generated Design Requirements

Name	Target Frequency(f_0)	Stiffness Ratio (K_x/K_y)
DR1	10.0 kHz	> 1 (x-axis stiffness greater than y-axis stiffness)
DR2	14.3 kHz	3
DR3	9.5 kHz	5
DR4	7.0 kHz	8
DR5	13.5 kHz	8

For each set of design requirements, we ran the MOGA process five times for each constraint case with a population of 400 designs for 50 generations. In our MOGA process, the inverse of the pareto rank is used as the fitness value of the design. Only the designs in the final pareto-optimal set which meet all of the initial design requirements are used in the analysis. The designs that are in the overall pareto set, have a frequency deviation within 5% of the target frequency and satisfy the stiffness ratio requirement are tallied after each MOGA synthesis process.

5.2 Analysis of Results

Table 3 shows the best designs in terms of best minimum and average area, as well as best minimum and average frequency error in the pareto sets for each of the design requirements (DR1-DR5). Note that C1 (no symmetry) appears to be favorable for achieving the best minimum area in the pareto set, whereas C4, the highly constrained full symmetry case with manhattan geometry, is favored for minimizing the average area across the entire pareto set of designs. In contrast, when considering frequency, the best minimum and average error results occur with the least constrained constraints cases, C1 and C2. To see if any of these competing trends are statistically significant we apply a Wilcoxon rank sum test to the data.

The Wilcoxon rank sum test [37], a non-parametric statistical test, is used to determine whether or not the constraint cases produce similar performing designs with respect to design area and resonant frequency deviation. To begin the rank test, we form the appropriate null hypothesis (H_0) and alternate hypothesis (H_a) using a significance level of 5% (or $\alpha = 0.05$):

- H_0: The distributions of the two compared constraint cases are identical
- H_a: The distributions of the two compared constraint cases are not identical and one distribution is shifted to the right or left of the other (implying one set of constraints generates better performing designs)

Table 3. Comparison of design area and frequency deviation

Design Requirements	Constraint Case	# of Pareto Optimal Solutions	Minimum Area [m^2]	Average Area [m^2]	Minimum Frequency Deviation [Hz]	Average Frequency Deviation [Hz]
DR1	C1	20	1.63E-07	1.92E-07	0.821	110.890
	C2	18	1.63E-07	1.79E-07	**0.147**	**96.972**
	C3	16	**1.46E-07**	1.96E-07	12.404	181.261
	C4	13	1.60E-07	**1.75E-07**	1.582	150.116
DR2	C1	23	**1.14E-07**	2.36E-07	0.143	**80.415**
	C2	31	1.36E-07	2.43E-07	**0.133**	119.509
	C3	17	1.30E-07	1.72E-07	1.604	121.910
	C4	14	1.28E-07	**1.53E-07**	1.680	156.698
DR3	C1	32	**1.29E-07**	4.31E-07	**0.041**	**45.237**
	C2	24	1.57E-07	2.12E-07	0.244	81.716
	C3	22	1.73E-07	2.79E-07	0.296	142.063
	C4	16	1.58E-07	**1.82E-07**	3.991	99.145
DR4	C1	27	**1.72E-07**	3.02E-07	**0.004**	45.618
	C2	21	2.05E-07	2.93E-07	0.165	**23.025**
	C3	15	2.05E-07	2.39E-07	2.869	69.650
	C4	23	1.90E-07	**2.22E-07**	0.812	116.348
DR5	C1	51	**1.20E-07**	3.50E-07	**0.003**	**27.185**
	C2	17	1.43E-07	1.95E-07	0.487	130.417
	C3	33	1.51E-07	1.86E-07	2.092	193.281
	C4	17	1.52E-07	**1.63E-07**	3.850	177.751

The p-values generated by the rank test for the micro-resonator case study are shown in Tables 4 and 5. In the instances where the p-value is less than the significance level, $\alpha = 0.05$, we can reject the null hypothesis (H_0) and accept the alternate hypothesis (H_a) indicating that the populations are significantly different. Conversely, when the p-value is greater than the significance level, we cannot reject the null hypothesis (H_0) within the context of this experiment.

Focusing on the frequency deviations (last column in Table 3), an analysis of the constraint cases demonstrates that MOGA produces statistically significant different pareto sets of designs between constraint cases C1&C3 and C1&C4 in four out of five instances, and C2&C3 in three out of five instances if we look at the entire pareto set. This trend implies that asymmetry and bilateral symmetry are preferred to full symmetry. The p-values for this scenario ranged from 1.88E-9≤p≤0.03355. If we focus on frequency deviation for the best designs from each MOGA synthesis run (see Table 5), constraint cases C1&C4 have statistically different distributions in all instances while C1&C3 have statistically different distributions in four out of five instances (0.0079≤ p≤0.0317).

Table 4. P-values for frequency deviation across the entire pareto set of designs

Design Requirements	C1&C2	C1&C3	C1&C4	C2&C3	C2&C4	C3&C4
DR1	0.8493	0.0772	0.4071	0.0942	0.5349	0.2635
DR2	0.1515	**0.0128**	**0.0107**	0.2532	0.1138	0.3934
DR3	0.3328	**0.0008**	**0.0295**	**0.0192**	0.2755	0.1433
DR4	0.9172	**0.0335**	**0.0042**	**0.0135**	**0.0009**	0.1888
DR5	0.0506	**1.88E-09**	**0.0001**	**0.0132**	0.1296	0.2777

Table 5. P-values for best minimum frequency deviations across each constraint case

Design Requirements	C1&C2	C1&C3	C1&C4	C2&C3	C2&C4	C3&C4
DR1	0.5476	**0.0079**	**0.0317**	0.0556	0.4206	0.0952
DR2	0.8413	0.0556	**0.0317**	0.2222	0.1508	0.6905
DR3	0.2222	**0.0317**	**0.0079**	0.0952	**0.0317**	1.0000
DR4	0.0952	**0.0079**	**0.0159**	0.0556	0.2222	0.0556
DR5	**0.0317**	**0.0079**	**0.0079**	0.2222	0.1508	0.5476

In addition to the frequency analysis, we also performed an analysis on design area. An analysis of the best performing designs from each synthesis run based on minimum area did not show a strong statistical difference. But, an analysis of area across the entire pareto set showed that C4 (full symmetry and manhattan angles) generates different pareto-optimal sets of designs for three out of five sets of design requirements for each possible constraint case combination (C1&C4, C2&C4, C3&C4). This supports results previously demonstrated by Kamalian [3]. Looking at Table 3, one can see that C4 had the best average design area overall for all of the design requirements.

When considering frequency deviation, it appears C1 is statistically better than C3 or C4 in almost all of the cases and C2 is significantly better than C3 and C4 in a majority of instances. But C1 is only statistically better than C2 for one instance – DR5 for the best minimum frequency deviation. The reverse is never the case – full symmetry with or without manhattan geometry shows no significant advantages for reducing frequency deviation. To highlight some of the design generated by our

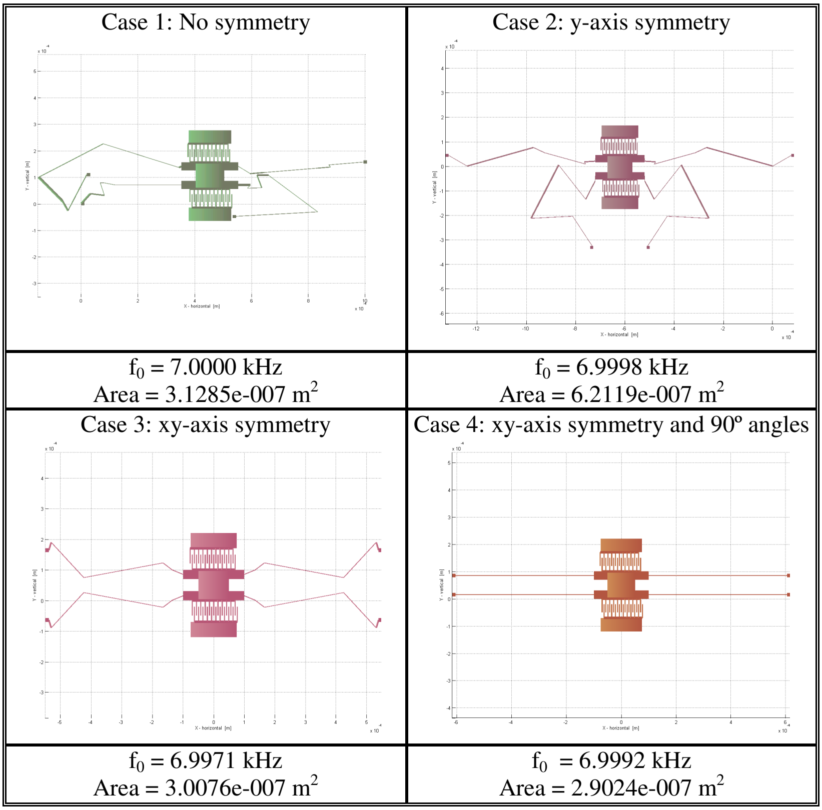


Fig. 10. Best designs based on resonant frequency for design requirement set DR4

MOGA constraint cases, Fig. 10 shows the best performing designs for the constraint cases for DR4 and Fig. 11 shows the best performing design based on frequency for the remaining design requirements (D1, D2, D3, and D5).

It is interesting to note that, in our previous discussion on symmetry and resonance observed in nature, bilateral symmetry is the preferred evolutionary design for spiders and similar insects based on their frequency needs for mating and catching prey. If we examine our results more closely, we must note that most of our design requirements favor asymmetrical or bilateral symmetry if frequency is the major consideration and full symmetry if average area minimization over the pareto set is the priority. However the difference between asymmetry and bilateral symmetry is not statistically significant. We can hypothesize that bilateral symmetry provides the balance between the competing objectives, but further investigation is required in order to validate this. Note that one of our design requirements involves a stiffness ratio, and this is a measure of resonator movement in the x- and y-direction. For our particular micro-resonator design, it is highly desirable to have a high stiffness ratio (rigidity in the x-direction and compliance in the y-direction) for the purposes of device stability. Thus, as we increase the stiffness ratio from a low value, such as $K_x/K_y = 1$ (DR1), to a high value such as $K_x/K_y = 8$ (DR4 and DR5), we are creating a bias against full symmetry in our optimization constraints. This bias in the stiffness ratio potentially forces the designs generated by MOGA to favor more asymmetrical layouts (C1 and C2) rather than fully symmetrical results (C3 and C4). This trend is shown in Table 4 where the bilateral symmetry C2 is statistically better than full symmetry C3 only for the higher stiffness cases DR3, DR4 and DR5.

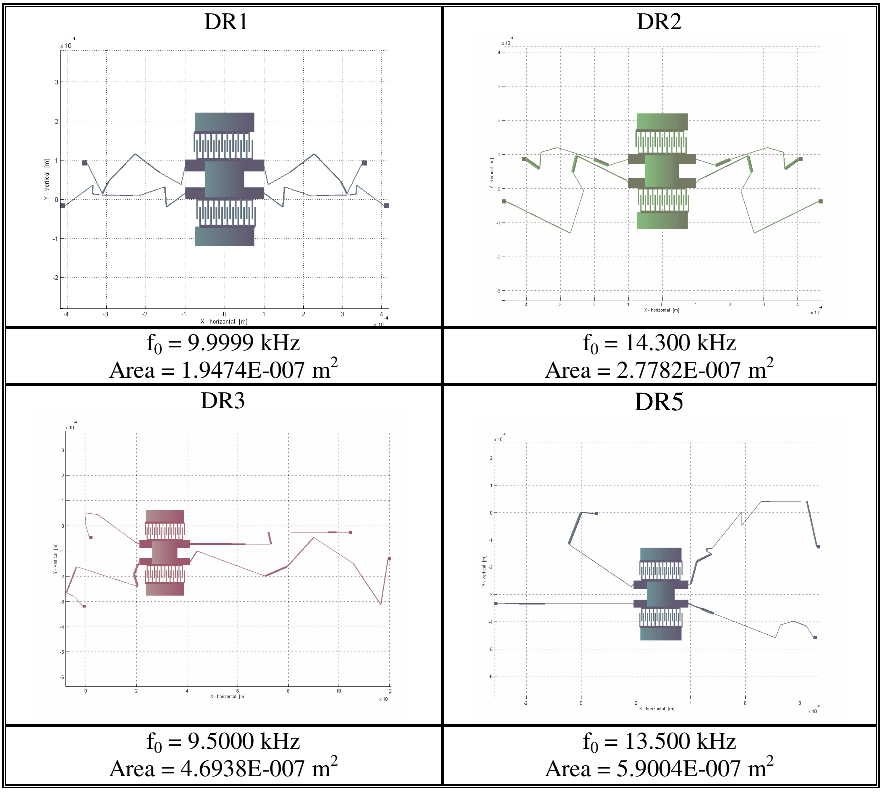


Fig. 11. Best performing designs based on frequency deviation for D1, D2, D3, and D5

Fig. 11 illustrates the best performing micro-resonator designs based on frequency deviation. Most of these designs have a very small deviation from the frequency goal if we look at the results in Table 3. The designs which have the smallest frequency deviation typically have one of the largest design areas in the pareto set. This is due to the conflicting objectives in our multi-objective optimization problem. There are trade-offs between the frequency, area, and stiffness objectives, and at this point, the human designer can decide which design in the pareto set is best suited for their MEMS design application. In this section, we have presented an analysis of the role symmetry constraints play in out MOGA linkage structure. Increasing the level of symmetry constraints can further restrict the search space to a more manageable size and enable our micro-resonator designs to achieve a smaller design area on average, but more asymmetrical designs are favored by MOGA for reducing frequency error and achieving the smallest design area. We hypothesize that the bilateral symmetry found in spiders and insects may be a compromise between frequency accuracy and compact size.

6 Summary and Conclusions

Our MEMS synthesis architecture, with the integration of MOGA and CBR, deals with the concept of linkage by using a component-based genotype representation and an automated design knowledge-base. CBR provides MOGA with good linkage information through past design knowledge while MOGA inherits linkage information through our component-based genotype representation. A MEMS micro-resonator test case was presented to show how symmetry constraints observed in nature can be embedded into our MOGA linkage structure to produce new promising MEMS design solutions. We found that when minimizing frequency error, asymmetry and bilateral symmetry are favored while conversely, when minimizing device area, the maximum constraints of full symmetry and enforced 90º angles are favored.

As part of our future research plan, we will examine how linkage learning can be integrated with MOGA when CBR may not be able to select a good initial seed design. Further exploring biomimetic algorithms and biomimetic ties to MEMS synthesis algorithms is another area we plan to pursue, investigating how increasing the number of leg components on a MEMS design can create optimal solutions in other design areas such as micro-robots. We want to also further explore the role symmetry and angle constraints have on these types of new MEMS designs. Lastly, we are moving towards creating a broader MEMS classification scheme and building up a case library of MEMS filter designs and their accompanying components to further expand the range of designs covered by our program.

Acknowledgements

This work was supported in part by NSF grant CCR-DES/CC-0306557 and a Bell Labs Graduate Research Fellowship.

References

1. Zhou, N., Zhu, B., Agogino, A.M., Pister, K.S.J.: Evolutionary synthesis of MEMS (MicroElectronicMechanical Systems) design. In: Proc. of the Artificial Neural Networks in Engineering Conference, pp. 197–202 (2001)
2. Clark, J.V., Zhou, N., Bindel, D., Schenato, L., Wu, W., Demmel, J., Pister, K.S.J.: 3D MEMS simulation modeling using modified nodal analysis. In: Proceedings of the Microscale Systems: Mechanics and Measurements Symposium, pp. 68–75 (2000)
3. Kamalian, R.H., Agogino, A.M., Takagi, H.: The role of constraints and human interaction in evolving MEMS designs: microresonator case study. In: Proc. of 2004 ASME Design Engineering Technical Conferences and Design Automation Conference #DETC2004-57462 (2004)

4. Kamalian, R., Zhang, Y., Agogino, A.M.: Microfabrication and characterization of evolutionary MEMS resonators. In: Proc. of the IEEE Robotics & Automation Society Symposium of Micro- and Nano-Mechatronics for Information-based Society, pp. 109–114 (2005)
5. Zhang, Y., Kamalian, R., Agogino, A.M., Séquin, C.H.: Hierarchical MEMS synthesis and optimization. In: Proc. SPIE Smart Structures and Materials 2005, Smart Electronics, MEMS, BioMEMS, and Nanotechnology (# 5763-12), vol. 5763, pp. 96–106 (2005)
6. Zhang, Y., Kamalian, R., Agogino, A.M., Séquin, C.H.: Design synthesis of Microelectromechanical Systems using genetic algorithms with component-based genotype representation. In: Proc. of 2006 Genetic and Evolutionary Computation Conference, vol. 1, pp. 731–738 (2006)
7. Cobb, C.L., Agogino, A.M.: Case-based reasoning for the design of micro-electromechanical systems. In: Proc. of 2006 ASME International Design Engineering Technical Conferences & the Computers and Information in Engineering Conference, #DETC2006-99120 (2006)
8. Cobb, C.L., Zhang, Y., Agogino, A.M.: MEMS design synthesis: integrating case-based reasoning and multi-objective genetic algorithms. In: Proc. SPIE Int. Soc. Opt. Eng., Smart Structures, Devices, and Systems III. 6414: #641419 (invited paper, 2006)
9. Mukherjee, T., Zhou, Y., Fedder, G.: Automated optimal synthesis of microaccelerometers. In: Technical Digest of 12th IEEE International Conference on Micro Electro Mechanical Systems (MEMS 1999), pp. 326–331 (1999)
10. Wang, J., Fan, Z., Terpenny, J.P., Goodman, E.D.: Knowledge interaction with genetic programming in mechatronics systems design using bond graphs. IEEE Transactions on System, Man, and Cybernetics−Part C: Applications and Reviews, vol. 35(2) (2005)
11. Li, J., Gao, S., Liu, Y.: Solid-based CAPP for surface micromachined MEMS devices. Computer-Aided Design 39, 190–201 (2007)
12. Holland, J.H.: Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor (1975)
13. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Longman, Boston (1989)
14. Chen, Y.-P., Goldberg, D.E.: Convergence time for the linkage learning genetic algorithm. Evolutionary Computation 13, 279–302 (2005)
15. Chen, Y.-P., Goldberg, D.E.: Introducing subchromosome representations to the linkage learning genetic algorithm. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 971–982. Springer, Heidelberg (2004)
16. Harik, G.R.: Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. Ph.D. Thesis. University of Michigan, Ann Arbor (1997)
17. Shank, R.: Dynamic memory: a theory of reminding and learning in computers and people. Cambridge (1982)
18. Kolodner, J.: Case-based reasoning. Morgan Kaufmann Publishers, Inc., San Mateo (1993)
19. Goel, A.K., Bhatta, S., Stroulia, E.: Kritik: An Early Case-Based Design System. In: Maher, M., Pu, P. (eds.) Issues and Applications of Case-Based Reasoning in Design, pp. 87–132. Lawrence Erlbaum Associates, Inc, Hillsdale (1997)
20. Hennessy, D., Hinkle, D.: Applying case-based reasoning to autoclave loading. IEEE Expert 7, 21–26 (1992)
21. Boyle, I.M., Rong, K., Brown, D.C.: CAFixD: A Case-Based Reasoning Fixture Design Method – Framework and Indexing Mechanisms. In: Proc. of the 2004 ASME International Design Engineering Technical Conferences & the Computers and Information in Engineering Conference, #DETC2004-57689 (2004)

22. Varadan, V.K.: RF MEMS and their applications. John Wiley and Sons, Chichester (2002)
23. Walraven, J.A.: Introduction to applications and industries for microelectromechanical systems (MEMS). In: Proc. of 2003 International Test Conference (ITC), pp. 674–680 (2003)
24. Bell, D.J., Lu, T.J., Fleck, N.A.F., Spearing, S.M.: MEMS actuators and sensors: observations on their performance and selection for purpose. J. Micromech Microeng 15, 153–164 (2005)
25. Nguyen, C.T.-C.: RF MEMS in wireless architecture. In: Proc. of the 42nd Design Automation Conference, pp. 416–420 (2005)
26. Noy, N.F., McGuinness, D.L.: Ontology development 101: a guide to creating your first ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 (2001)
27. Stewart, I.: Why beauty is truth: A History of Symmetry. Joat Enterprises, Basic Books (2007)
28. Johnstone, R.A.: Female preferences for symmetrical males as a by product of selection for mate recognition. Nature 372, 172–175 (1994)
29. Swaddle, J.P., Cuthill, I.C.: Preference for symmetric males by female zebra finches. Nature 367, 165–166 (1994)
30. Enquist, M., Arak, A.: Symmetry, beauty, and evolution. Nature 372, 169–172 (1994)
31. Babcock, L.E.: Asymmetry in the fossil record. European Review 13, 135–143 (2005)
32. Moller, A.P., Manning, J.: Growth and developmental instability. The Veterinary Journal 166, 19–27 (2003)
33. Pfannenstiel, R.S., Hunt, R.E., Yeargan, K.V.: Orientation of a hemipteran predator vibrations produced by feeding caterpillars. J. of Insect Behavior 8, 1–9 (1995)
34. Haynes, K.F., Yeargan, K.V., Gemenol, C.: Detection of prey by a spider that aggressively mimics pheromone blends. J. of Insect Behavior 14, 535–544 (2001)
35. Parry, D.A.: The signal generated by an insect in a spider's web. J. Exp. Biol. 43, 185–192 (1965)
36. Blickhan, R., Full, R.J.: Similarity in multilegged locomotion: Bouncing like a monopode. J. Comp. Physiol A 173, 509–517 (1993)
37. Mendenhall, W., Sincich, T.: Statistics for engineering and the sciences, 4th edn. Prentice-Hall, Inc., Upper Saddle River (1995)