6

Automating the Hierarchical Synthesis of MEMS Using Evolutionary Approaches

Zhun Fan[1,2], Jiachuan Wang[3], Kisung Seo[1], Jianjun Hu[4], Ronald Rosenberg[5], Janis Terpenny[3], and Erik Goodman[1]

[1] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48823, USA (fanzhun | ksseo | goodman)@egr.msu.edu, http://www.egr.msu.edu
[2] Department of Mechanical Engineering, Technical University of Denmark, DK-2800 Kgs. Lynby, Denmark
[3] Department of Industrial Engineering and Operations Research University of Massachusetts Amherst, Amherst, MA 01003 USA (jiacwang | terpenny)@ecs.umass.edu, http://www-unix.ecs.umass.edu
[4] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48823, USA hujianju@egr.msu.edu, http://www.egr.msu.edu/ hujianju
[5] Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 48823, USA ron@egr.msu.edu, http://www.egr.msu.edu/ rosenber/

The complex device, component, and system design issues involved in integrated MEMS design call for a structured design methodology that borrows from VLSI design. In this chapter, we first discuss the hierarchy that is involved in a typical MEMS design. Then we move on to discuss how evolutionary approaches can be used to automate the hierarchical design and synthesis process for MEMS. At the system level, genetic programming, as a strong search tool, is used to generate and search in the topologically open-ended design space. Meanwhile, bond graphs are used to represent the lumped parameter models of MEMS that cut across mixed energy domains. The approach combining bond graphs and genetic programming can lead to satisfactory design candidates of system level models that meet the predefined behavioral specifications for designers to tradeoff. Then at the second level, namely the physical layout synthesis level, the selection of geometric parameters for component devices is formulated as a constrained optimization problem and addressed using a constrained GA approach. Considerations of feature size constraints can be incorporated into this approach very conveniently. A multiple-resonator microsystem design is taken as an example to illustrate the

integrated design automation idea using evolutionary approaches at multiple levels.

6.1 Introduction

MicroElectroMechanical Systems (MEMS) is a rapidly expanding technology that offers new ways of combining sensing, actuation, signal processing, computing and communication functions on a miniature scale. Although MEMS is a promising technology, it is very surprising that we have only seen a handful of successful commercial MEMS products which the market has demanded in large quantities, including automotive accelerometers and gyroscopes, pressure sensors, ink-jet print heads and a few others. Prevalence of design and fabrication of MEMS application-specific integrated circuits (ASICS) analogous to electronic ASICS is still not seen. Due to the complexity and intricacy involved in MEMS design, designing MEMS still remains an art in most applications, requiring a large amount of investment of human resources, time and money. Much of the investment is consumed in the iterative trial-and-error design process. Automated design synthesis helps engineers to develop rapid, optimal configurations for a given set of performance and constraint guidelines, and thus to shorten typical development cycles for MEMS (with a given fabrication technology) by a large factor and to enable design of far more complex MEMS than can be handled today. Electronic Design Automation (EDA) has achieved great success in both industry and academia. However, analogous research in design automation for MEMS seems to lag far behind, although considering the close affinity of MEMS and VLSI - MEMS actually evolved from microelectronics and inherited the fabrication techniques of VLSI - the potential successful applications of design automation of MEMS appear to be promising. It turns out that translating the key insights of silicon evolution success into MEMS technologies is a much more challenging task than most people have expected. Major research topics to be addressed include: 1) developing a broad base of building blocks in MEMS technologies so that huge networks of micro-devices could be assembled into arbitrary architectures with desirable functionalities, 2) abstracting design hierarchies to stratify and conquer design complexity, thus making the design more amenable to an automated process, 3) improving models of computation and extending current synthesis methodologies to facilitate generation of viable design candidates and smoother transitions from conceptual and embodied designs to process fabrication. 4) combining MEMS component layout extraction and lumped-parameter bond graph simulation and design synthesis to provide MEMS designers with VLSI-like environments enabling faster design cycles and improved design productivity.

This chapter seeks to partially address the above challenges, especially the first two. The proposed hierarchical and evolutionary design framework for MEMS aims to eliminate tedious and repetitive design tasks, facilitate

hierarchical problem decomposition, and combine the power of multiple evolutionary computation algorithms working simultaneously to identify better product designs and process solutions. In particular, we divide design representations of MEMS design into two levels, the system-level behavioral macromodel and the detailed-level physical geometric layout model. At the system level, we use a combination of genetic programming and bond graphs to automatically generate and search for viable design candidates represented by behavioral macromodels satisfying high-level design specifications. At the second detailed (layout) level, multiobjective constrained genetic algorithms are used to optimize the geometric parameters that relate the physical device model to the behavioral macromodel and meet more detailed design objectives.

6.2 Hierarchical MEMS Design Methodology

MEMS holds the promise of being amenable to structured automated design due to its similarities with VLSI. However, design and analysis of MEMS is much more complicated due to their multi-domain and intrinsically three-dimensional nature. In addition, because of limitations of fabrication technology, there are many constraints in design of MEMS. In MEMS, there are a number of levels of designs that need to be synthesized [1]. Usually the design process starts with basic capture of the schematic of the overall system, and then goes on through layout and construction of a 3-D solid model. So the first design level is the system level, which includes selection and configuration of a repertoire of planar devices or subsystems. The second level is 2-D layout of basic structures like beams to form the elementary planar devices. In some cases, if the MEMS is basically a result of a surface-micro machining process and no significant 3-D features are present, design of this level will end one cycle of design. More generally, modeling and analysis of a 3-D solid model for MEMS is necessary. However, even if we have obtained an optimized 3-D device shape, it is still very difficult to produce a proper mask layout and correct fabrication procedures. Automated mask layout and process synthesis tools would be very helpful to relieve designers from considering the fabrication details and focus on the functional design of the device and system [2]. After a "top-down" design path, a "bottom-up" verification process is usually followed to guarantee that at each design level the design specifications are met exactly as defined in Fig. 6.1. The ultimate goal is to develop tools for MEMS design to ensure first-pass success by having a well-defined "top-down" design path and "bottom-up" verification path.

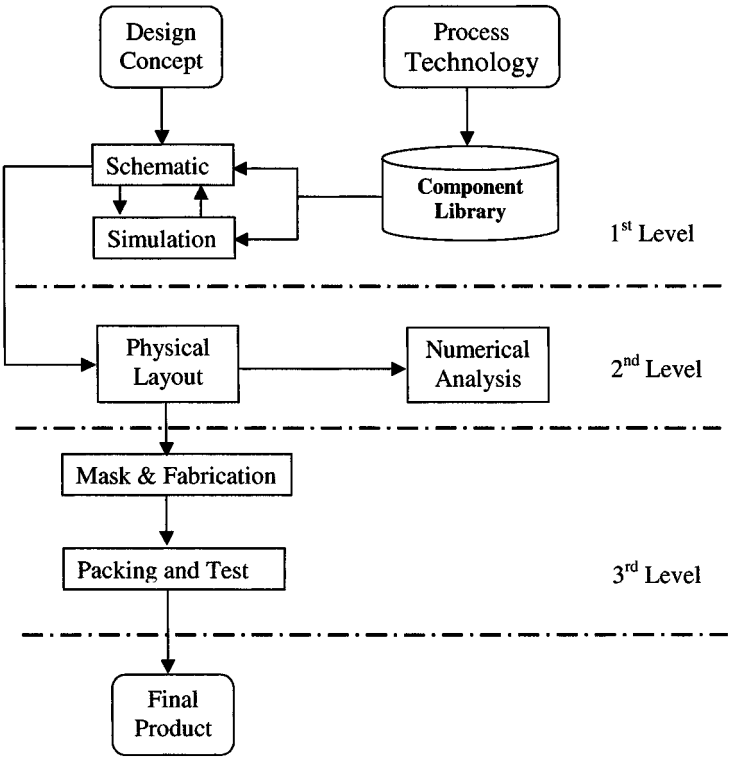

Fig. 6.1. Hierarchical design of MEMS

6.3 System-Level Synthesis of MEMS Using Genetic Programming and Bond Graphs

For system-level design, hand calculation is still the most popular method in current design practice. This is mainly because no powerful and widely accepted synthesis approach exists to automated design of multi-domain systems. In addition, most MEMS system-level design is accomplished by modeling entire microelectromechanical system as single behavioral entities having no lower hierarchical level in design. If there is any change in geometric parameters or topology, a whole new model must be created, and this substantially lengthens design cycles. Over the past two decades, computational design algorithms based on Darwin's principles of evolution have developed from academic curiosities into practical and effective tools for scientists and engineers. Gero, for example, investigates evolutionary systems as computational models of creative design and studies the relationships among genetic engineering, style emergence, and complex evolution [3]. Goodman et al. [4] studied evolution of engineering artifacts using heterogeneous parallel genetic algorithms. Koza has applied genetic programming to evolve analog filter cir-

cuits and can optimize the topology and sizing parameters of the evolved circuits simultaneously [5]. In this research, we use genetic programming as a strong search tool to explore the topologically open-ended design space for system-level behavioral models of MEMS. We also use bond graphs as a modeling tool to unify representations of mixed energy domains of MEMS. We call the overall approach the BG/GP approach.

6.3.1 Bond graphs

The reason we used bond graphs in research on MEMS synthesis is because MEMS are intrinsically multi-domain systems, unlike electronic systems. We need a uniform representation of MEMS so that designers can not only shift among different hierarchies of design abstractions but also can move around design partitions with different physical domains without difficulty. The bond graph is a modeling tool that provides a unified approach to the modeling and analysis of dynamic systems, especially hybrid multi-domain systems including mechanical, electrical, pneumatic, hydraulic components, etc. It is the explicit representation of model topology that makes the bond graphs a good candidate for use in open-ended design search. Fig. 6.2 shows an example of unique bond graphs representation of a resonator unit in three different application domains. It is also very natural to use bond graphs to represent a dynamic system, such as a mechatronic system, with cross-disciplinary physical domains and even controller subsystems (Fig.6.3). For notation details and methods of system analysis related to the bond graph representation, see [6]. Shah [7] identifies the importance of bond graphs for unifying multi-level design of multi-domain systems. Tay et al. [8] use bond graphs and GA to generate and analyze dynamic system designs automatically. This approach adopts a variational design method, which means they make a complete bond graph model first, and then change the bond graph topologically using a GA, yielding new design alternatives. However, the efficiency of this approach is hampered by the weak ability of GA to search in both topology and parameter spaces simultaneously. Terpenny and Jiachuan Wang have begun to explore combination of bond graphs and evolutionary computation [9]. Campell [10] also uses the idea of both bond graphs and genetic algorithms in his A-Design framework. In this research, we use an approach combining genetic programming and bond graphs to automate the process of design of dynamic systems to a significant degree.

6.3.2 Combining bond graphs and genetic programming

The most common form of genetic programming [5] uses trees to represent the entities to be evolved. Defining of a proper function set is one of the most significant steps in using genetic programming. It may affect both the search efficiency and validity of evolved results and is closely related to the selection of building blocks for the system being designed. By executing the genotype,

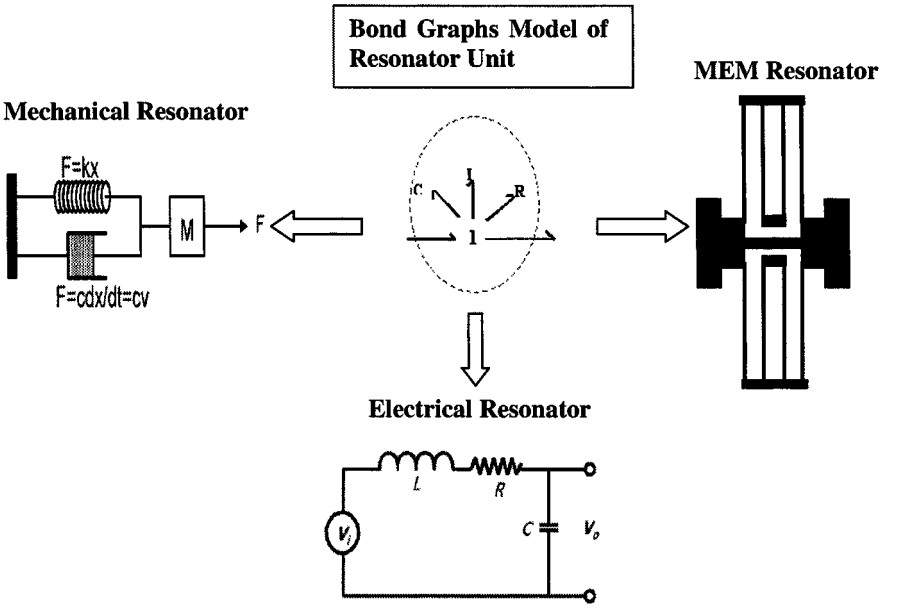

Fig. 6.2. Bond graphs representing a mechatronic system with mixed energy domains and a controller subsystem

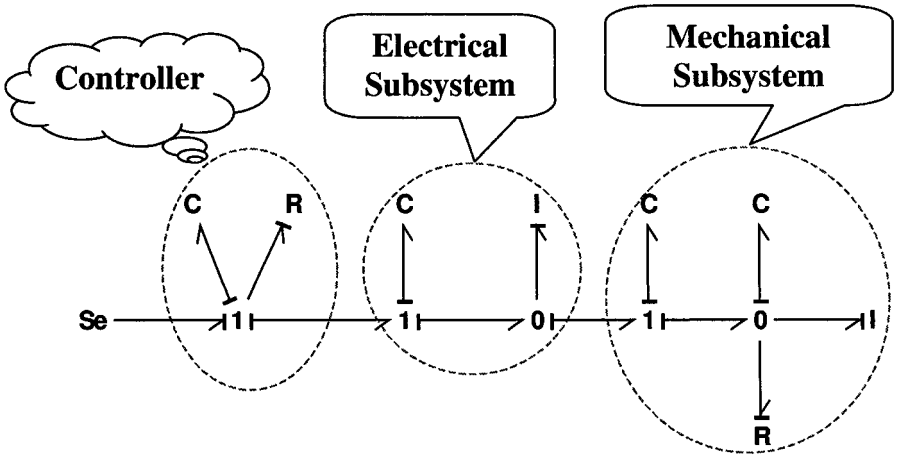

Fig. 6.3. One bond graph represents resonators in different application domains

a genetic programming tree that composes of functions in the function set as nodes of the tree, an arbitrary representative topology, or phenotype can be generated in a developmental manner. In this research, we have an additional dimension of flexibility in generating phenotypes, because bond graphs are used as modeling representations for multi-domain systems, serving as an intermediate representation between the mapping of genotype and phenotype, and can be interpreted as systems in different physical domains, chosen as appropriate to given circumstances. Fig. 6.4 illustrates the role of bond graphs in the mappings from genotypes to phenotypes. [11]

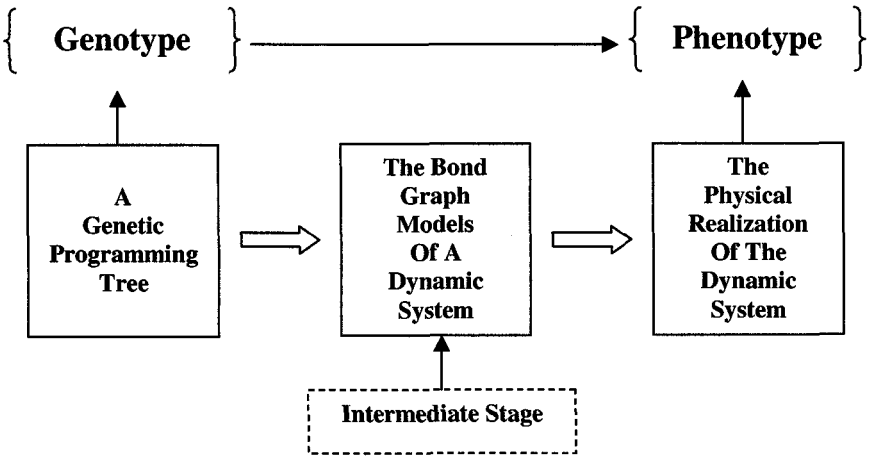

Fig. 6.4. Genotype-phenotype mapping

6.3.3 Filter topology

Automated synthesis of an RF MEM device, a micro-mechanical bandpass filter, is used as an example in this chapter [12]. Through analyzing two popular topologies used in surface micromachining of micro-mechanical filters, we found that they are topologically composed of a series of concatenated Resonator Units (RUs) and Bridging Units (BUs) or RUs and Coupling Units (CUs). Fig. 6.5 and Fig. 6.6 illustrates the layouts and bond graph representations of two widely accepted filter topologies I and II [12]. Their corresponding bond graph representations are also shown.

6.3.4 Function set

In this research, a GP function set is presented and listed in Table 6.1. Examples of operators, namely insert-CU and insert-RU, are illustrated in Fig.s 6.7 and 6.8. Fig. 6.7 explains how the insert-CU function works. A Coupling

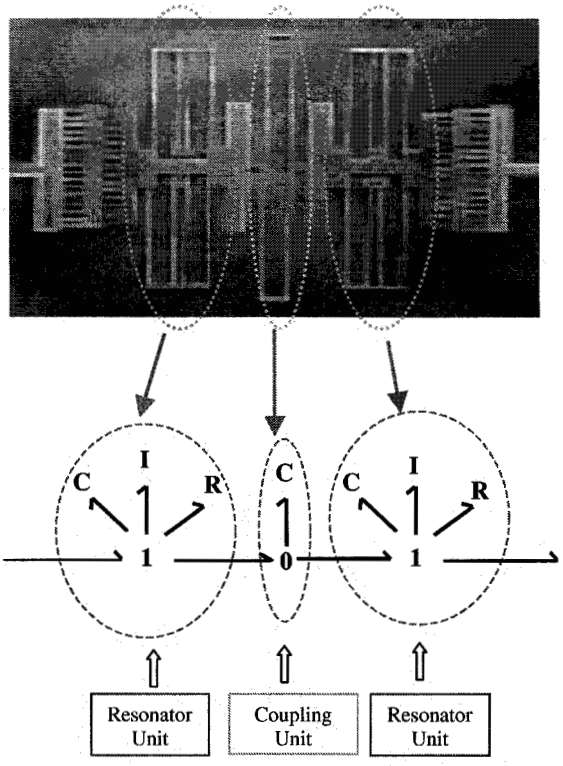

Fig. 6.5. MEM filter topology I

Unit (CU) is a subsystem that is composed of a capacitor attached with a 0-junction in the center and two bonds connecting 1-junctions at the left and right ends. After execution of the insert-CU function, an additional modifiable site (2) appears at the rightmost newly created bond. As illustrated in Fig. 6.8, a resonator unit (RU), composed of one I, R, and C component all attached to a 1-junction, is inserted in an original bond with a modifiable site through the insert-RU function. After the insert-RU function is executed, a new RU is created and one additional modifiable site, namely bond (3), appears in the resulting phenotype bond graph, along with the original modifiable site bond (1). The newly-added 1-junction also has an additional modifiable site (2). As components C, I, and R all have parameters to be evolved, the insert-RU function has three corresponding ERC-typed sites, (4), (5), and (6), for numerical evolution of parameters.

6.3.5 Design embryo

All individual genetic programming trees create bond graphs from an embryo. Selection of the embryo is also an important topic in system design, especially

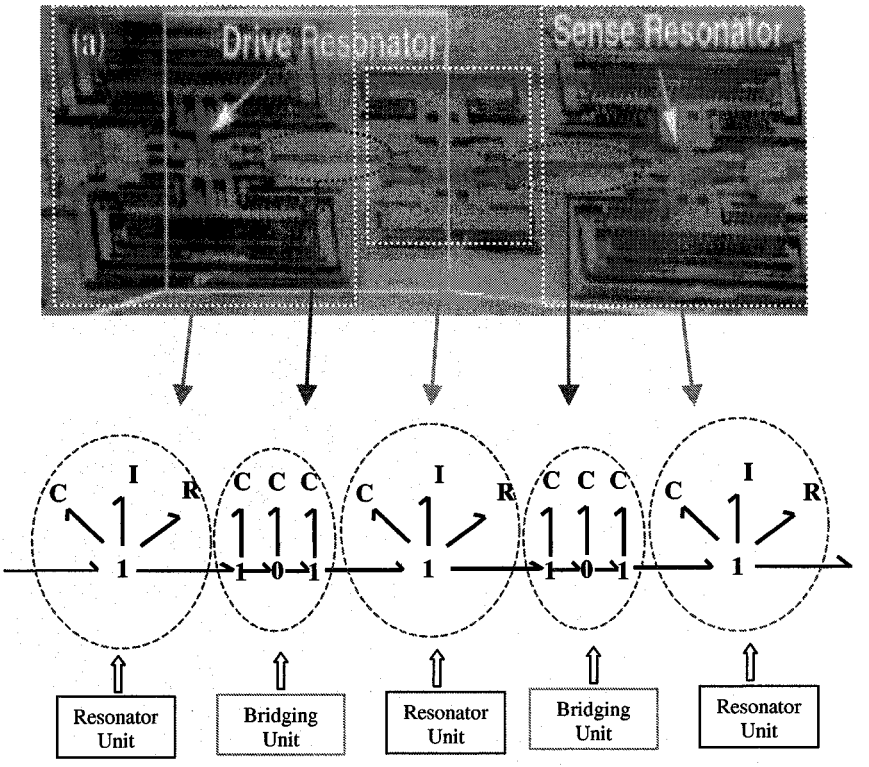

Fig. 6.6. MEM filter topology II

Table 6.1. Operators in modular function set

Operator Name	Functionality
Insert-RU	insert a resonator unit
Insert-CU	insert a coupling unit
Insert-BU	insert a bridging unit
Add-RU	add a resonator unit
Insert-J01	insert a 0-1-junction
Insert-CIR	insert a special CIR component
Insert-CR	insert a special CR component

for multi-port systems. In our filter design problems, we use the bond graph shown in Fig. 6.9 as our embryo.

6.3.6 Fitness function

Within the frequency range of interest, $f_{rang} = [f_{min}, f_{max}]$, uniformly sample 100 points. Here, f_{rang} = [0.1, 1000K] Hz. Compare the magnitudes of the

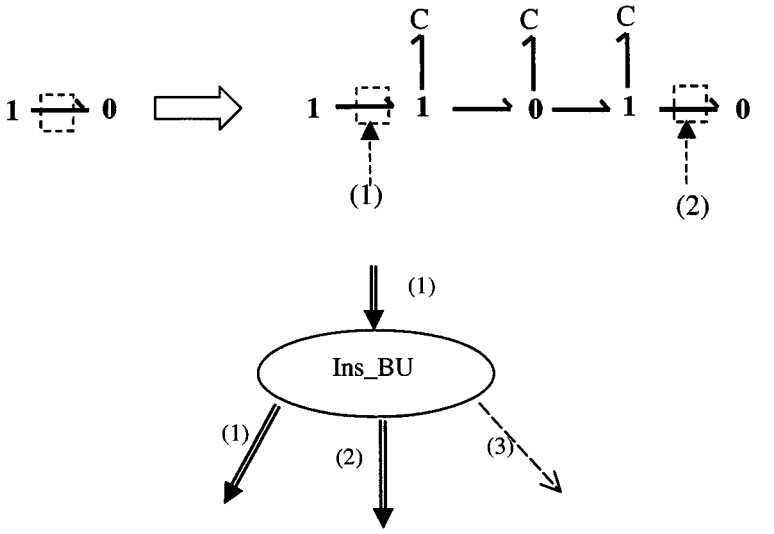

Fig. 6.7. Operator to insert Bridging Unit

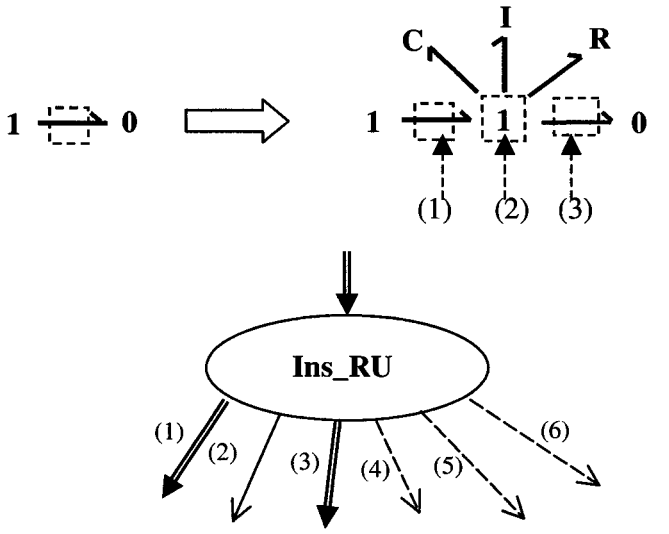

Fig. 6.8. Operator to insert Resonator Unit

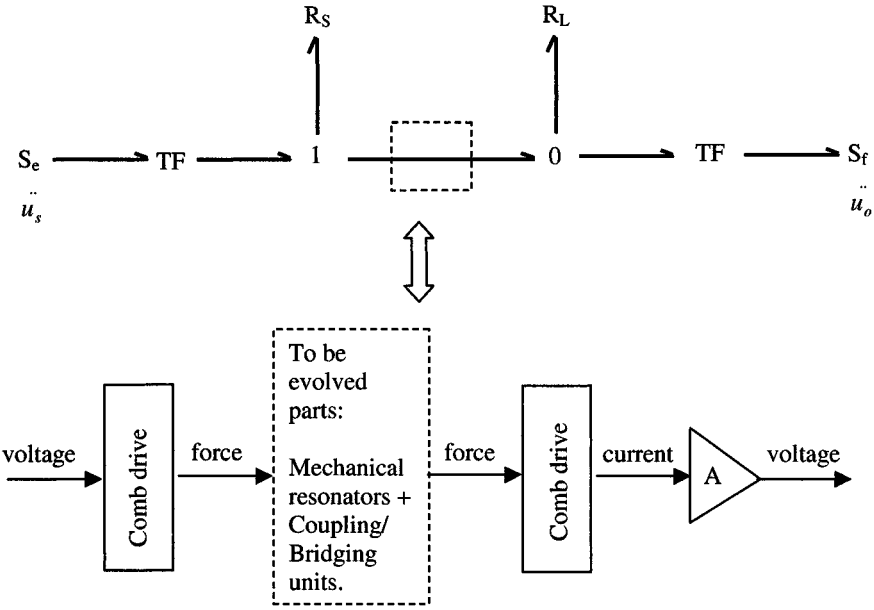

Fig. 6.9. Design Embryo of a Micro-Electro-Mechanical Filter

frequency response at target magnitudes, which are 1.0 within the pass frequency range of [316, 1000] Hz, and 0.0 otherwise, between 0.1 and 1000KHz.

6.3.7 Experimental setup

Three major code modules were created in this work. The algorithm kernel of HFC-GP was a strongly typed version [13] of an open software package developed in our research group – lilgp. Parameters for lilgp are shown in the tableau 6.2.

Table 6.2. Parameter settings for genetic programming

Parameter	Setting
population size	500 in each of thirteen subpopulations
initial population	half and half
initial depth	4-6
maximum depth	50
maximum nodes	5000
selection method	tournament with size 7
crossover rate	0.9
mutation rate	0.3

A bond graph class was implemented in C++. The fitness evaluation package is C++ code converted from Matlab code, with hand-coded functions used to interface with the other modules of the project. The commercial software package 20Sim was used to verify the dynamic characteristics of the evolved design. The GP program obtains satisfactory results on a Pentium-IV 1GHz in 1000 1250 minutes.

6.3.8 Experimental result

Experimental results show the strong topological search capability of genetic programming and feasibility of our BG/GP approach for finding realizable designs for micro-mechanical filters [14]. In Fig. 6.11, K is the number of resonator units appearing in the best design of the generation on the horizontal axis. As fitness improves, the number of resonator units, K, grows - unsurprising because a higher-order system with more resonator units has the potential of better system performance than its low-order counterpart. The plot of corresponding system frequency responses at generations 27, 52, 117 and 183 are shown in Fig. 6.10. A layout of a design candidate with four resonators and three coupling units as well as its bond graph representation is shown below in Fig. 6.12. Notice that the geometry of resonators may not show the real sizes and shapes of a physical resonator and the layout figure only serves as a topological illustration. Using the BG/GP approach, it is also possible to explore novel topologies of MEM filter design. In this case, we may not necessarily use a strictly realizable function set. Instead, a semi-realizable function set may be used to relax the topological constraints, with the purpose of finding new topologies not realized before but still realizable after careful design. Fig. 6.13 gives an example of a novel topology for a MEM filter design. An attempt to fabricate this kind of topology is being carried out in a university research setting.

6.4 Second-Level Physical Layout Synthesis Formatting the Headings

Layout synthesis automatically generates valid or optimized geometric sizing parameters for cell components, which in most cases are commonly used micromechanical devices with fixed topologies, according to engineering design objectives. In this research, the cell component is a resonator device in MEMS domain. The design objectives come from either high-level specifications such as behavioral model parameters that need to be satisfied, or from layout-level objectives such as minimum areas occupied. Our approach is to model the design problem as a formal constrained optimization problem, and then solve it with powerful optimization techniques, resulting in a tool that automates the design synthesis of MEMS structures. Two categories of optimization techniques are used: one category includes stochastic algorithms

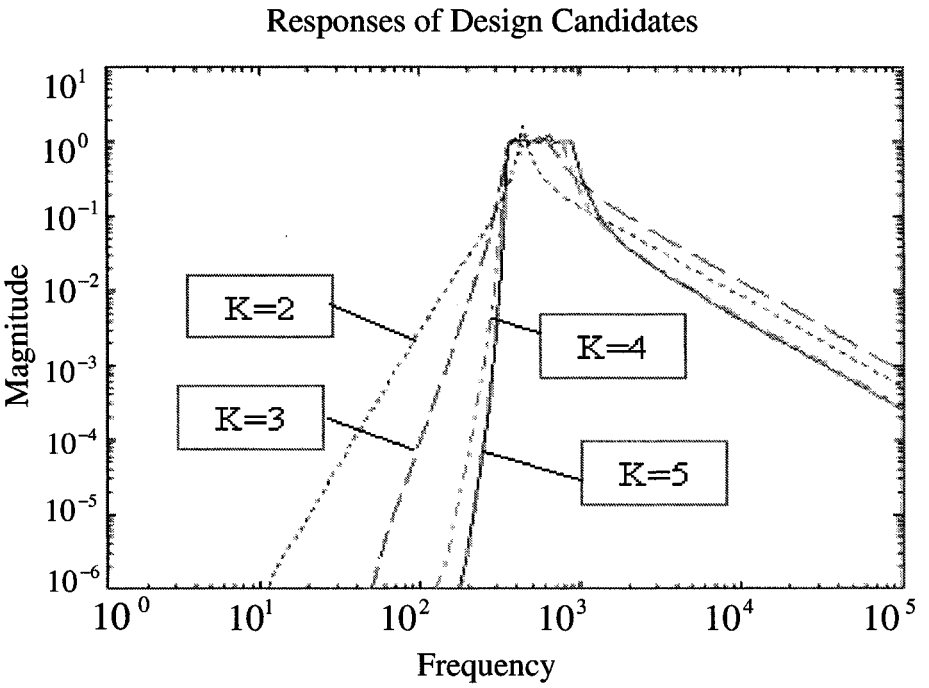

Fig. 6.10. Frequency responses of a sampling of design candidates, which evolved topologies with larger numbers, K, of resonators as the evolution progressed. All results are from one genetic programming run of the BG/GP approach

such as genetic algorithms, and the other category includes deterministic algorithms such as nonlinear programming. For both categories, the process of solving the optimization problem involves determining the design variables, the design constraints, and the design objective. We decided to use 14 design variables for an example cell component, a folded-flexure comb-drive microresonator fabricated in a polysilicon surface microstructural process (Fig. 6.14) in this research. Design variables and their constraints are listed as follows (Fig. 6.15) [15]:

It is noted that the first 13 design variables have units of μm. The fourteenth design variable has units of volts. In addition, we assume t = w_c = g = d. in our design for simplicity. Some design variables are predefined: they are $w_{ba} = 11$, $w_{ca} = 14$, $\delta = 4$, N = 10 . The constraints for the design variables are listed below.

$$2 \leq L_b \leq 400, 2 \leq w_b \leq 20, 2 \leq L_t \leq 400, 2 \leq w_t \leq 20 \tag{6.1}$$

$$2 \leq L_{sy} \leq 400, 10 \leq w_{sy} \leq 400, 10 \leq w_{sa} \leq 400, 10 \leq w_{cy} \leq 400 \tag{6.2}$$

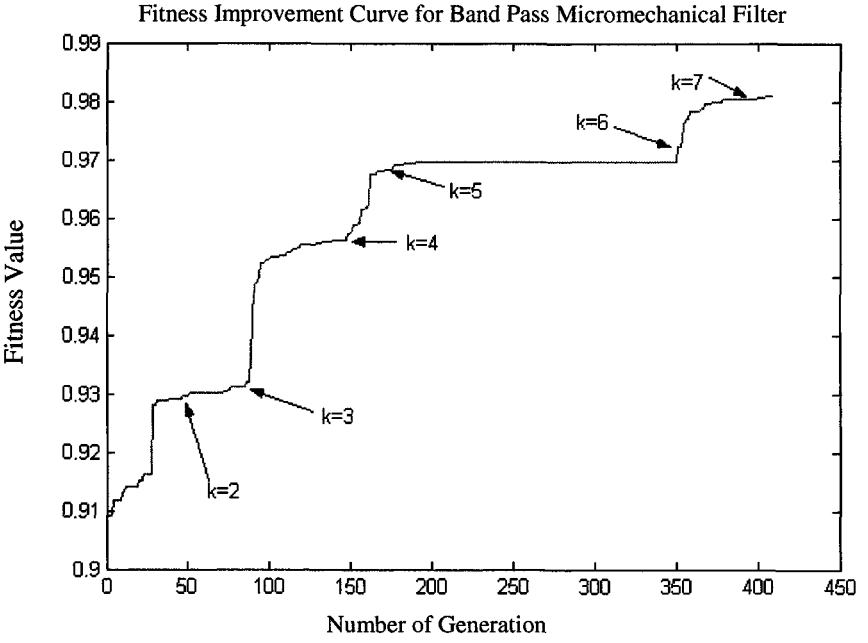

Fig. 6.11. Fitness improvement curve

$$10 \leq L_{cy} \leq 700, 8 \leq L_c \leq 400, 2 \leq w_c \leq 20, 2 \leq L_{sa} \leq 400 \tag{6.3}$$

$$4 \leq x_o \leq 400, 0 \leq V \leq 100 \tag{6.4}$$

also a number of design constraints for the microresonator cell component, including both geometric constraints and functional constraints. In this chapter, without loss of generality, we consider the following constraints:

$$0 \leq L_{cy} + 2g + 2w_c \leq 700 \tag{6.5}$$

$$0 \leq L_{sy} + 2L_b + 2w_t \leq 700 \tag{6.6}$$

$$0 \leq 3L_t + w_{sy} + 4L_c - 2x_0 + 2w_{cy} + 2w_{ca} \leq 700 \tag{6.7}$$

$$4 \leq L_c - (x_0 + x_{disp}) \leq 200 \tag{6.8}$$

Among them, the first three are linear constraints, and the fourth is a nonlinear constraint because the term x_{disp} is highly nonlinear. $x_{disp} = QF_{e,x}/K_x$, where $F_{e,x} = 1.12\varepsilon_0 NV^2 t/g$,

Suppose that in the system-level synthesis, we get a set of behavioral parameters for the cell component of a microresonator as

$$K_x = 0.27N/m \tag{6.9}$$

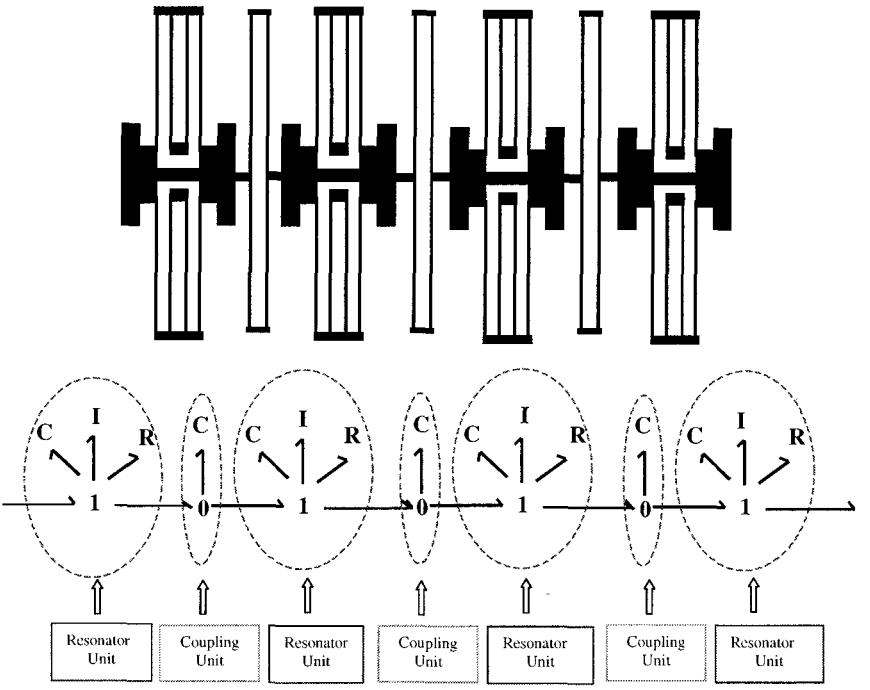

Fig. 6.12. Layout and bond graph representation of a design candidate from the experiment, with four resonator units coupled with three coupling units

$$B_x = 5.18 \times 10^{-6} kg \cdot m^2 \tag{6.10}$$

$$M_x = 4.0 \times 10^{-6} kg \tag{6.11}$$

Then we have three additional equation constraints.Equations to relate the design variables and the three behavioral model parameters are as follows:

$$K_x = \frac{2Etw_b^3}{L_b^3} \frac{L_t^2 + 14\alpha L_t L_b + 36\alpha^2 L_b^2}{4L_t^2 + 41\alpha L_t L_b + 36\alpha^2 L_b^2} \tag{6.12}$$

$$B_x = \mu[(A_s + 0.5A_t + 0.5A_b)(1/d + 1/\delta) + A_c/g] \tag{6.13}$$

$$M_x = M_s + \frac{1}{4}M_t + \frac{12}{35}M_b \tag{6.14}$$

where $\alpha = (W_t/W_b)^3$, $M_s = \rho A_s$, $M_t = \rho A_t$, $M_b = \rho A_b$, and

$$A_s = w_{sa}L_{sa} + 2w_{sy}L_{sy} \tag{6.15}$$

$$A_t = 2w_{ca}L_{cy} \tag{6.16}$$

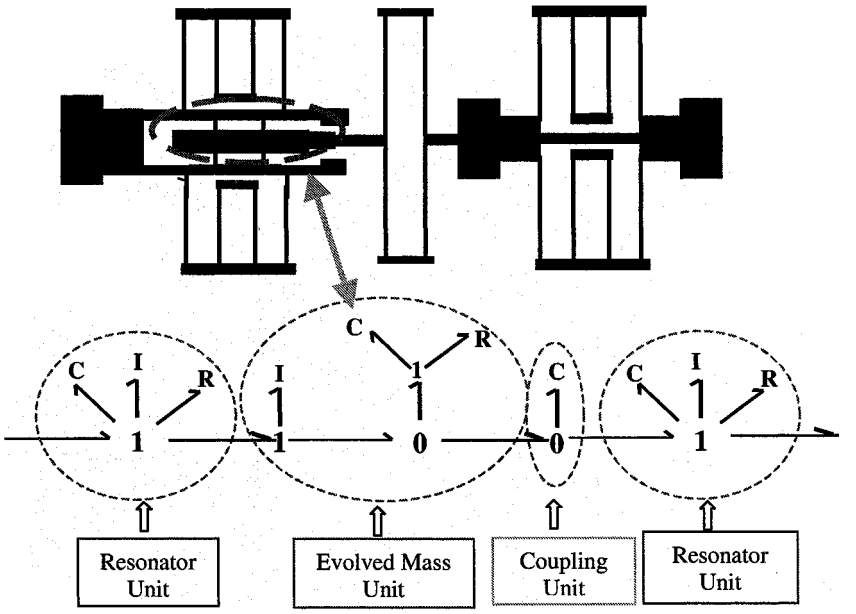

Fig. 6.13. A novel topology of MEM filter and its bond graph representation

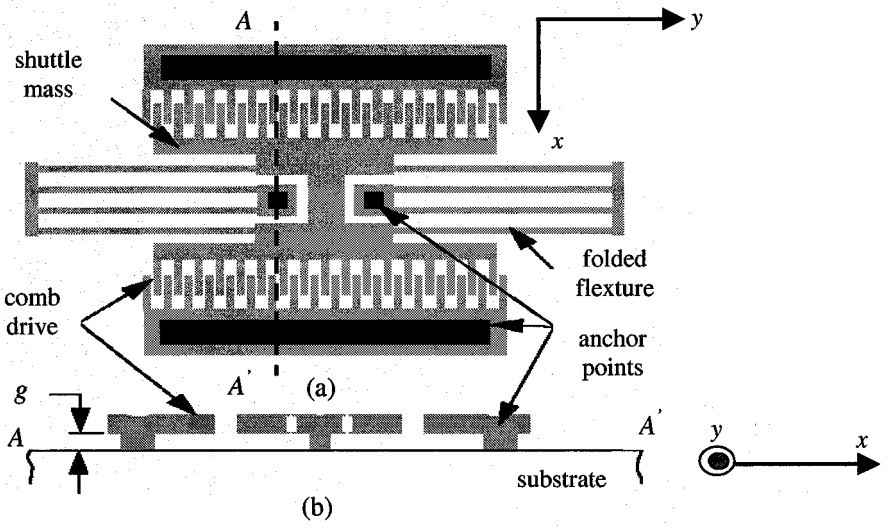

Fig. 6.14. A folded-flexure comb-drive microresonator fabricated in a polysilicon surface microstructural process a) Layout b) Cross-section A-A' [15]

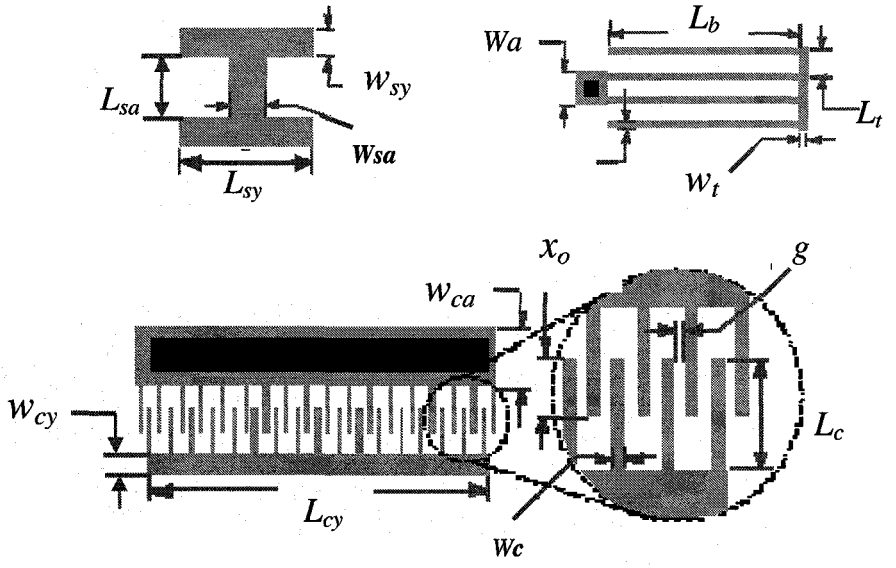

Fig. 6.15. Major design variables for microresonators

$$A_b = 8L_b w_b + 2w_t(2L_t + w_a + 2w_b) \tag{6.17}$$

As an alternative, we can also put reformulations of these three constraint equations into our design objectives, expressing them as differences to be minimized. In that case, we actually deal with a multi-objective constrained optimization problem.We take the objective function with the following normalized Sum of Squared Error (SSE) format:

$$\begin{aligned} f(\vec{x}) = & \frac{1}{0.27^2}(K_x - 0.27)^2 + \frac{1}{(5.18\times10^{-6})^2}(B_x - 5.18\times10^{-6})^2 \\ & + \frac{1}{(4.0\times10^{-6})^2}(M_x - 4.0\times10^{-6})^2 \end{aligned} \tag{6.18}$$

Finally, it is important to note the role of feature size in VLSI and MEMS design. Feature size, which is often represented as λ , means the minimum size a particular design can achieve, based on specific fabrication procedures. In addition, the actual sizes of geometric shapes should be integer multiples of the feature size λ, such as λ, 2λ , 5λ , 10λ etc. In this research, we set $\lambda = 0.09\mu$m .

While it is very difficult for many numerical optimization approaches (for example, gradient-based approaches) to include considerations of feature size constraints [15], it is quite convenient for genetic algorithms to do so. We need to modify the objective function only slightly, mapping real values of design variables to integer multiples of the feature size λ before using them in formulations of constraints and objectives. No modifications to the genetic algorithm are needed.

6.4.1 Solving the constrained optimization problem using GA

In trying to solve constrained optimization problems using genetic algorithms or classical deterministic optimization methods, penalty function methods have been the most popular approach, because of their simplicity and ease of implementation. In this chapter, we use a special constrained GA that exploits pair-wise comparisons in a tournament selection operator to devise a penalty function approach that does not require any penalty parameter. Careful comparisons among feasible and infeasible solutions are made so as to provide a search direction towards the feasible region. Once sufficient feasible solutions are found, a niching method (along with a controlled mutation operator) is used to maintain diversity among feasible solutions. This allows a real-parameter GA's crossover operator to continuously find better feasible solutions, gradually leading the search nearer to the true optimum solution [16]. The parameters for setting the constrained GA are listed in Table 6.3.

Table 6.3. The parameters for setting the constrained GA

Parameter	Setting
variable boundaries	rigid
population size	500
total number of generations	100
crossover probability	0.9
mutation probability	0.15
niching parameter	0.9
exponent(n for SBX)	2.0
exponent(n for mutation)	50.0

In nine runs of the genetic algorithm using different random seeds, we obtained the sizing parameters and values of the objective function NSSE (to be minimized) listed in Table 6.4.

It can be seen that during the nine runs using different seeds, the constrained GA performs very steadily. Almost all runs achieved NSSE within the range of 1.0E-06. The biggest NSSE is 1.4E-05. However, the normalized squared sum of errors of 1.4E-05 is still considered very good result. It also appears that there are many alternatives and rather different ways in which parameters can be set and still produce behavior rather close to that desired.

6.5 Summary

This chapter has suggested a design methodology for automatically synthesizing hierarchical designs for MEMS. While there has been much research using evolutionary computation techniques to synthesize MEMS [2][17], this is the

Table 6.4. Layout parameters obtained in nine GA runs(different random seeds)

Run No.	1	2	3	4	5	6	7	8	9
$L_b(\lambda m)$	261.6	261.4	261.1	262.4	262.3	260.8	261.7	261.9	262.6
$w_b(\lambda m)$	1.98	1.98	1.98	1.98	1.98	1.98	1.98	1.98	1.98
$L_t(\lambda m)$	3.87	4.32	3.87	3.60	8.46	2.43	2.52	5.13	6.84
$w_t(\lambda m)$	2.70	2.25	2.52	2.52	2.25	1.98	1.98	2.88	3.33
$L_{sy}(\lambda m)$	3.69	2.88	2.07	4.41	1.98	1.98	3.60	1.98	2.79
$w_{sy}(\lambda m)$	14.13	12.60	15.93	11.52	19.80	9.99	11.52	15.30	12.60
$w_{sa}(\lambda m)$	18.63	18.18	10.98	11.70	11.34	11.16	10.17	11.70	14.58
$w_{cy}(\lambda m)$	146.16	151.83	122.31	141.12	137.25	56.61	110.70	76.14	247.50
$L_{cy}(\lambda m)$	15.66	20.79	23.85	17.37	23.85	30.69	22.68	21.96	8.91
$L_c(\lambda m)$	199.3	187.3	174.1	202.4	181.9	154.7	188.2	162.1	161.9
$w_c(\lambda m)$	1.98	1.98	1.98	1.98	1.98	1.98	1.98	1.98	1.98
$L_{sa}(\lambda m)$	2.25	2.16	2.52	2.43	2.88	1.98	2.70	2.70	6.30
$x_o(\lambda m)$	10.26	96.12	24.66	34.92	10.35	14.94	30.87	20.34	25.83
$V(volt)$	66.06	70.29	75.51	64.98	72.27	85.14	69.93	81.09	81.27
NSSE	4.0E-6	3.0E-6	3.0E-6	1.0E-6	1.0E-6	1.4E-5	2.0E-6	2.0E-6	1.0E-6

first work reported to seek to automate the hierarchical MEMS synthesis process in an integrated framework. Our first step is to synthesize system-level behavioral models using a combination of genetic programming and bond graphs. Then as the second step, we use a constrained genetic algorithm to automatically optimize the geometric sizing parameters for the cell components. An example of MEM filter design with coupling of multiple microresonators is used to illustrate the approach. Extension of this work can lead to a composable design and synthesis environment for micromechatronic systems [18]. In addition, target cascading in optimal system design needs to be investigated in depth to propagate the desirable top-level design specifications to appropriate specifications for the various subsystems and components in a consistent and efficient manner [19][20]. More work is underway to improve the efficiency of genetic programming to explore topologically open-ended design spaces, and the robustness of the constrained genetic algorithm to solve real-world constrained optimization problems.

References

1. Fedder, G.K. and Q. Jing, A Hierarchical Circuit-Level Design Methodology for Microelectromechanical Systems, IEEE Transactions on Circuits and Systems II (TCAS), vol. 46, no. 10, pp. 1309-1315, Oct. 1999
2. Ma, L. and E. K. Antonsson, Automated Mask-Layout and Process Synthesis for MEMS, Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems, pp. 20-23, 2000
3. Gero J. S., Computers and Creative Design, in M. Tan and R. Teh (eds), The Global Design Studio, National University of Singarpo pp. 11-19, 1996

4. Eby, D., R. Averill, B. Gelfand, W. Punch, O. Matthews, E. Goodman, An Injection Island GA for Flywheel Design Optimization. 5th European Congress on Intelligent Techniques and Soft Computing, EUFIT'97. Vol. 1. pp. 687-691, 1997
5. Koza J. R., Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press, 1994
6. Rosenberg R. C., Reflections on Engineering Systems and Bond Graphs, Trans. ASME J. Dynamic Systems, Measurements and Control, 115, pp. 242-251, 1993
7. Vargas-Hernandez N., J. Shah, Z. Lacroix, Development of a Computer-Aided Conceptual Design Tool for Complex Electromechanical Systems, Computational Synthesis: From Basic Building Blocks to High Level Functionality, Papers from the 2003 AAAI Symposium Technical Report SS-03-02 pp. 255-261, 2003
8. Tay E. H., Flowers W. and Barrus J., Automated Generation and Analysis of Dynamic System Designs, Research in Engineering Design 10: pp. 15-29, 1998
9. Wang, J. and Terpenny, J., Interactive Evolutionary Solution Synthesis in Fuzzy Set-based Preliminary Engineering Design, Special Issue on Soft Computing in Manufacturing, Journal of Intelligent Manufacturing, Vol. 14. pp. 153-167, 2003
10. Campbell, M., Cagan J. and Kotovsky K., Agent-based Synthesis of Electro-Mechanical Design Configurations, Journal of Mechanical Design, Vol. 122. No. 1, pp. 61-69, 2000
11. Fan Z., Hu J., Seo K., Goodman E., Rosenberg R., and Zhang B., Bond Graph Representation and GP for Automated Analog Filter Design, GECCO-2001 Late-Breaking Papers, San Francisco, pp. 81-86, 2001
12. Wang K. and Nguyen C. T. C., High-Order Medium Frequency Micromechanical Electronic Filters, Journal of Microelectromechanical Systems, pp. 534-556, 1999
13. Luke S., Strongly-Typed, Multithreaded C Genetic Programming Kernel, http://www.cs.umd.edu/users/-seanl/gp/patched-gp/, 1997
14. Fan Z., Seo K., Rosenberg R., Hu J., Goodman E., System-Level Synthesis of MEMS via Genetic Programming and Bond Graphs, Proc. 2003 Genetic and Evolutionary Computing Conference, Chicago, Springer, Lecture Notes in Computer Science, July,2058-2071, 2003
15. Fedder G. and Mukherjee T., Physical Design for Surface-Micromachined MEMS, Proceedings of the Fifth ACM/SIGDA Physical Design Workshop, April, pp. 53-60, 1996
16. Deb K., An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg., Vol. 186, pp. 311-338, 2000
17. N. Zhou, B. Zhu, A.M. Agogino, K.S.J. Pister, Evolutionary Synthesis of MEMS (Microelectronic Mechanical Systems) Design, Proceedings of ANNIE 2001, Intelligent Engineering Systems through Artificial Neural Networks, Volume 11, ASME Press, pp. 197-202, 2001
18. C. J. J. Paredis, A. Diaz-Calderon, R. Sinha, and P. K. Khosla, Composable Models for Simulation-Based Design, Engineering with Computers 17, pp. 112-128, 2001
19. Kim, H.M., Michelena, N.F., Papalambros, P.Y., and Jiang, T., " Target Cascading in Optimal System Design, Proceedings of the 2000 ASME Design Automation Conference, DAC-14265, Baltimore, Maryland, USA, September 10-13, 2000

20. Kim, H.M., Target Cascading in Optimal System Design, Ph.D. Dissertation, Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA, 2001