System-Level Synthesis of MEMS via Genetic Programming and Bond Graphs

Zhun Fan[+], **Kisung Seo**[+], **Jianjun Hu**[+],
Ronald C. Rosenberg[Ψ], **Erik D. Goodman**[+]

[+]Genetic Algorithms Research and Applications Group (GARAGe)
[Ψ]Department of Mechanical Engineering
Michigan State University, East Lansing, MI, 48824
{hujianju, ksseo, fanzhun, rosenber, goodman}@egr.msu.edu

Abstract. Initial results have been achieved for automatic synthesis of MEMS system-level lumped parameter models using genetic programming and bond graphs. This paper first discusses the necessity of narrowing the problem of MEMS synthesis into a certain specific application domain, e.g., RF MEM devices. Then the paper briefly introduces the flow of a structured MEMS design process and points out that system-level lumped-parameter model synthesis is the first step of the MEMS synthesis process. Bond graphs can be used to represent a system-level model of a MEM system. As an example, building blocks of RF MEM devices are selected carefully and their bond graph representations are obtained. After a proper and realizable function set to operate on that category of building blocks is defined, genetic programming can evolve both the topologies and parameters of corresponding RF MEM devices to meet predefined design specifications. Adaptive fitness definition is used to better direct the search process of genetic programming. Experimental results demonstrate the feasibility of the approach as a first step of an automated MEMS synthesis process. Some methods to extend the approach are also discussed.

1 Introduction

Mechanical systems are known to be much more difficult to address with either systematic design or clean separation of design and fabrication. Composed of parts involving multiple energy domains, lacking a small set of primitive building blocks such as the NOR and NAND gates in used VLSI, and lacking a clear separation of form and function, mechanical systems are so diverse in their design and manufacturing procedures that they present more challenges to a systematic approach and have basically defied an automated synthesis attempt.

Despite the numerous difficulties presented in automated synthesis of macro-mechanical systems, MEMS holds the promise of being amenable to structured automated design due to its similarities with VLSI, provided that the synthesis is carried out in a properly constrained design domain.

Due to their multi-domain and intrinsically three-dimensional nature of MEMS, their design and analysis is very complicated and requires access to simulation tools with finite element analysis capability. Computation cost is typically very high. A common representation that encompasses multiple energy domains is thus needed for modeling of the whole system. We need a system-level model that reduces the number of degrees of freedom from the hundreds and thousands of degrees of freedom characterizing the meshed 3-D model to as few as possible. The bond graph, based on power flow, provides a unified model representation across multiple energy domain system and is also compatible with 3-D numerical simulation and experimental results in describing the macro behavior of the system, so long as suitable lumping of components can be done to obtain lumped-parameter models. It can be used to represent the behavior of a subsystem within one energy domain, or the interaction of multiple domains. Therefore, the first important step in our method of MEMS synthesis is to develop a strategy to automatically generate bond graph models to meet particular design specifications on system level behaviors.

For system-level design, hand calculation is still the most popular method in current design practice. This is for two reasons: 1) The MEMS systems we are considering, or designing are relatively simple in dynamic behavior -- especially the mechanical parts -- largely due to limitation in fabrication capability. 2) There is no powerful and widely accepted synthesis approach to automated design of multi-domain systems.

The BG/GP approach, which combines the capability of genetic programming to search in an open-ended design space and the merits of bond graphs for representing and modeling multi-domain systems elegantly and effectively, proves to be a promising method to do system-level synthesis of multi-domain dynamical systems [1][2]. In the first or higher level of system synthesis of MEMS, the BG/GP approach can help to obtain a high-level description of a system that assembles the system from a library of existing components in an automated manner to meet a predefined design specification. Then in the second or lower level, other numerical optimization approaches [3], as well as evolutionary computation, may be used to synthesize custom components from a functionality specification. It is worthwhile to point out that for the system designer, the goal of synthesis is not necessarily to design the optimum device, but to take advantage of rapid prototyping and "design reuse" through component libraries; while for the custom component designer, the goal may be maximum performance. These two goals may lead to different synthesis pathways. Figure 1 shows a typical structured MEMS synthesis procedure, and the BG/GP approach aims to solve the problem of system-level synthesis in an automated manner in the first level.

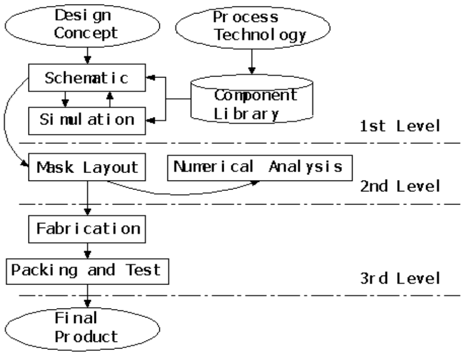

Fig.1. Structured MEMS design flow

However, in trying to establish an automated synthesis approach for MEMS, we should take cautious steps. Due to the limitations of fabrication technology, there are many constraints in design of MEMS. Unlike in VLSI, which can draw on extensive sets of design rules and programs that automatically test for design-rule violations, the MEMS field lacks design verification tools at this time. This means that no design automation tools are available at this stage capable of designing and verifying any kind of geometrical shapes of MEMS devices. Thus, automated MEMS synthesis tools must solve sub-problems of MEMS design in particular application domains for which a small set of predefined and widely used basic electromechanical elements are available, to cover a moderately large functional design space.

Automated synthesis of an RF MEM device, namely, a micro-mechanical band pass filter, is taken as an example in this paper. As designing and micromachining of more complex structures is a definite trend, and research into micro-assembly is already on its way, the BG/GP approach is believed to have many potential applications. More work to extend this approach to an integrated evolutionary synthesis environment for MEMS across a variety of design layers is also discussed at the end.

2. Design Methodology

2.1 Bond Graphs

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. For notation details and methods of system analysis related to the bond graph representation, see [4].

Bond graphs have four embedded strengths for design applications, namely, the wide scope of systems that can be created because of the multi- and inter-domain nature of bond graphs, the efficiency of evaluation of design alternatives, the natural combinatorial features of bond and node components for generation of design alternatives, and the ease of mapping to the engineering design process. Those attributes make bond graphs an excellent candidate for modeling and design of a multi-domain system.

2.2 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. In this research, a basic function set and a modular function set are presented and listed in Tables 1 and 2. Operators in the basic function set basically aim to construct primitive building blocks for the system, while operators in the modular function set purport to utilize relatively modular and predefined building blocks composed of primitive building blocks. Notice that numeric functions are included in both function sets, as they are needed in both cases. In other research, we hypothesize that usage of modular operators in genetic programming has some promise for improving its search efficiency. However, in this paper, we concentrate on another issue, proposing the concept of a realizable function set. By using only operators in a realizable function set, we seek to guarantee that the evolved design is physically realizable and has the potential to be manufactured. This concept of realizability may include stringent fabrication constraints to be fulfilled in some specific application domains. This idea is to be illustrated in the design example of an RF MEM device, namely, a micro-mechanical band pass filter.

Examples of modular operators, namely insert_BU and insert_CU operators, are illustrated in Figures 2 and 3. Examples of basic operators are available in our earlier work [6].

Table 1. Operators in Basic Function Set

Basic Function Set	
add_C	Add a C element to a junction
add_I	Add a I element to a junction
add_R	Add a R element to a junction
insert_J0	Insert a 0-junction in a bond
insert_J1	Insert a 1-junction in a bond
replace_C	Replace the current element
replace_I	Replace the current element
replace_R	Replace the current element
+	Sum two ERCs
-	Subtract two ERCs
enda	End terminal for add functions
endi	End terminal for insert func-
endr	End terminal for replace func-
erc	Ephemeral Random Constant

Figure 2 explains how the insert_BU function works. A Bridging Unit (BU) is a subsystem that is composed of three capacitors with the same parameters, attached together with a 0-junction in the center and 1-junctions at the left and right ends. After execution of the insert_BU function, an additional modifiable site (2) appears at the rightmost newly created bond.

As illustrated in Figure 3, a resonant 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 new 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. The reason these representations are chosen for the RU and BU components is discussed in the next, case study, section.

Table 2. Operators in Modular Function Set

Modular Function Set	
insert_RU	Insert a Resonant Unit
insert_CU	Insert a Coupling Unit
insert_BU	Insert a Bridging Unit
add_RU	Add a Resonant Unit
insert_J01	Insert a 0-1-junction com-
insert_CIR	Insert a special CIR com-
insert_CR	Insert a special CR compound
Add_J	Add a junction compound
+	Sum two ERCs
-	Subtract two ERCs
endn	End terminal for add func-
endb	End terminal for insert func-
endr	End terminal for replace
erc	Ephemeral Random Constant

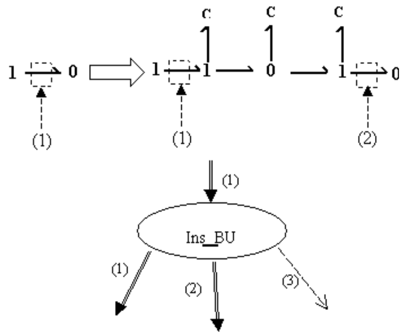

Fig. 2. Operator to Insert Bridging Unit

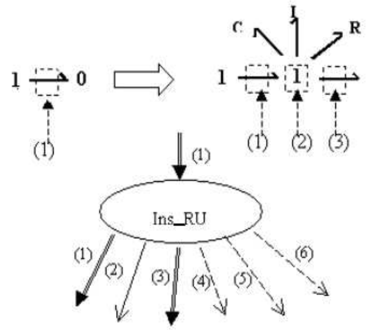

Fig. 3. Operator to Insert Resonant Unit

3. MEM Filter Design

3.1 Filter Topology

Automated synthesis of a RF MEM device, micro-mechanical band pass filters is used as an example in this paper [7]. Through analyzing two popular topologies used in surface micromachining of micro-mechanical filters, we found that they are topologically composed of a series of RUs and Bridging Units (BUs) or RUs and Coupling Units (CUs) concatenated together. Figure 4, 5, 6 illustrates the layouts and bond graph representations of filter topology I and II.

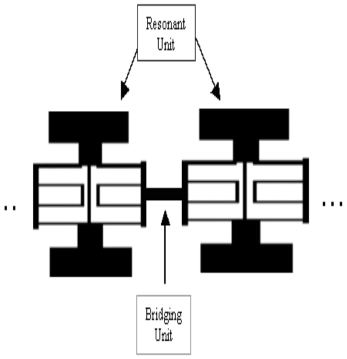

Fig. 4. Layout of Filter Topology I: Filter is composed of a series of Resonator Units (RUs) connected by Bridging Units (BUs).

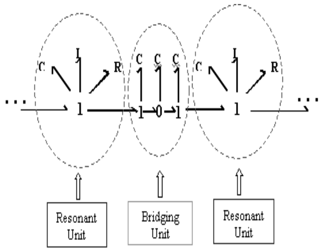

Fig. 5. Bond Graph Representation of Filter Topology I

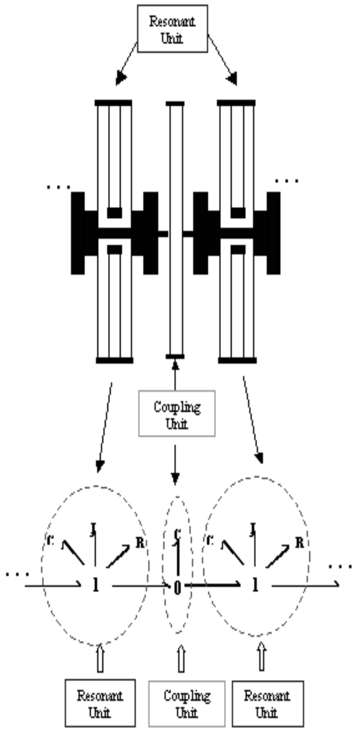

Fig. 6. Layout of Filter Topology II:
Filter is composed of a series of Resonator Units coupled by Coupling Units. Its corresponding bond graph representation is also shown.

3.2 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 for multi-port systems. In our filter design problems, we use the following bond graph as our embryo.

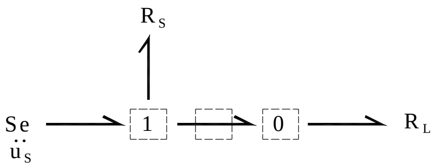

Fig. 7. Embryo of Design

3.3 Realizable Function Set

BG/GP is a quite general approach to automate synthesis of multidisciplinary systems. Using a basic set of building blocks, BG/GP can perform topologically open composition of an unconstrained design. However, engineering systems in the real world are often limited by various constraints. So if BG/GP is to be used to synthesize real-world engineering systems, it must enforce those constraints.

Unlike our previous designs with basic function sets, which impose fewer topological constraints on design, MEMS design features relatively few devices in the component library. These devices are typically more complex in structure than those primitive building blocks used in the basic function set. Only evolved designs represented by bond graphs matching the dynamic behavior of those devices belonging to the component library are expected to be manufacturable under current or anticipated technology. Thus, an important and special step in MEMS synthesis with the BG/GP approach is to define a *realizable* function set that, throughout execution, will produce only phenotypes that can be built using existing or expected technology.

Analyzing the system of MEM filters of [7] from a bond graph viewpoint, the filters are basically composed of Resonator Units (RUs) and Coupling Units (CUs). Another popular MEM filter topology includes Resonator Units and Bridging Units (BUs). A realizable function set for these design topologies often includes functions from both the basic set and modular set. In many cases, multiple realizable function sets, rather than only one, can be used to evolve realizable structures of MEMS. In this research, we used the following function set, along with traditional numeric functions and end operators, for creating filter topologies with coupling units and resonant units.

$$\begin{aligned}\Re 1 = \{ & f\_tree, f\_insert\_J1, f\_insert\_RU, \\ & f\_insert\_CU, f\_add\_C, f\_add\_R, f\_add\_I\}\end{aligned}$$

$$\begin{aligned}\Re 2 = \{ & f\_tree, f\_insert\_J1, f\_insert\_RU, \\ & f\_insert\_BU, f\_add\_C, f\_add\_R, f\_add\_I\}\end{aligned}$$

3.4 Adaptive Fitness Function

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

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

Compute their differences and get a sum of squared differences as raw fitness, defined as $Fitness_{raw}$.

If $Fitness_{raw} <$ Threshold, change f_{range} to $f_{range}^{*} = [f_{min}^{*}, f_{max}^{*}]$. Usually $f_{range}^{*} \subset f_{range}$. Repeat the above steps and obtain a new $Fitness_{raw}$.

Then normalized fitness is calculated according to:

$$Fitness_{norm} = 0.5 + Norm \big/ (Norm + Fitness_{raw})$$

The reason to use adaptive fitness evaluation is that after a GP population has reached a fairly high fitness value as a group, the differences of frequency responses of individuals need to be centered on a more constrained frequency range. In this circumstance, if there is not sufficient sampling within this much smaller frequency range, the GP may lack sufficient search pressure to push the search forward. The normalized fitness is calculated from the sampling differences between the frequency response magnitudes of the synthesized systems and the target responses. Therefore, we adaptively change and narrow the frequency range to be heavily sampled. The effect is analogous to narrowing the search window on a smaller yet most significant area, magnifying it, and continuing to search this area with closer scrutiny.

3.5 Experimental Setup

We used a strongly-typed version of lilgp to generate bond graph models. The major GP parameters were as shown below:

Population size: 500 in each of thirteen subpopulations
Initial population: half_and_half
Initial depth: 4-6
Max depth: 50 Max_nodes 5000
Selection: Tournament (size=7)
Crossover: 0.9 Mutation: 0.3

Three major code modules were created in this work. The algorithm kernel of HFC-GP was a modified version of an open software package developed in our research group -- lilgp. 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.

3.6 Experimental Results

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. Although significant fabrication difficulty is currently presented when fabricating a micro-mechanical filter with more than 3 resonators, it does not invalidate our research and the topological search capability of the BG/GP approach BG/BP shows potential for exploring more complicated topologies of future MEMS design and the ever-progressing technology frontiers of MEMS fabrication.

In Figure 8, K is the number of resonant units appearing in the best design of the generation on the horizontal axis. The use of hierarchical fair competition [8] is facilitating continual improvement of the fitness. As fitness improves, the number of resonant 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.

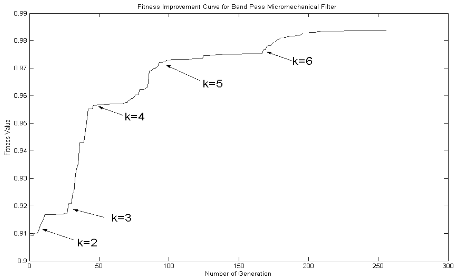

Fig. 8. Fitness Improvement Curve

The plot of corresponding system frequency responses at generations 27, 52, 117 and 183 are shown in Figure 9.

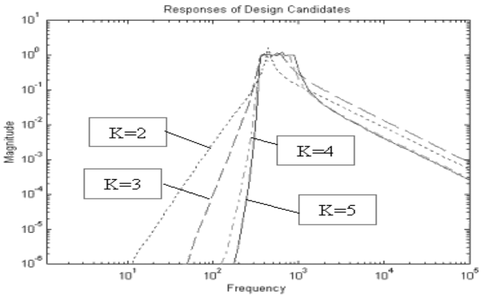

Fig. 9. Frequency responses of a sampling of design candidates, which evolved topologies with larger numbers, K, of resonators as the evolution progressed. All results are from one genetic programming run of the BG/GP approach.

A layout of a design candidate with three resonators and two bridging units as well as its bond graph representation is shown below in Figure 10. 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 necessary use a strictly realizable function set. Instead, a semi-realizable function set is used to relax the topological constraints with the purpose of finding new topologies not realized before but still realizable after careful design. Figure 11 gives an example of a novel topology for a MEM filter design. Attempts to fabricate topology of this sort are being carried out at the University of California, San Barbara.

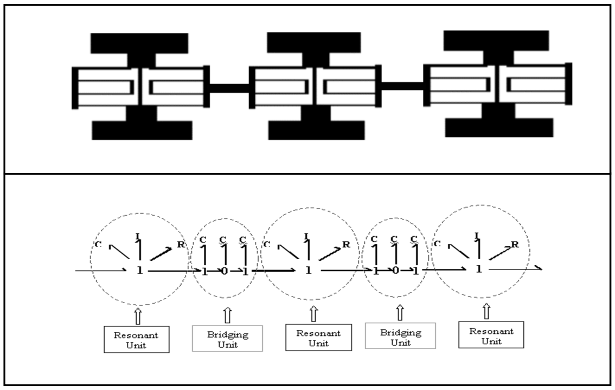

Fig. 10. Layout and bond graph representation of a design candidate from the experiment with three resonator units coupled with two bridging units.

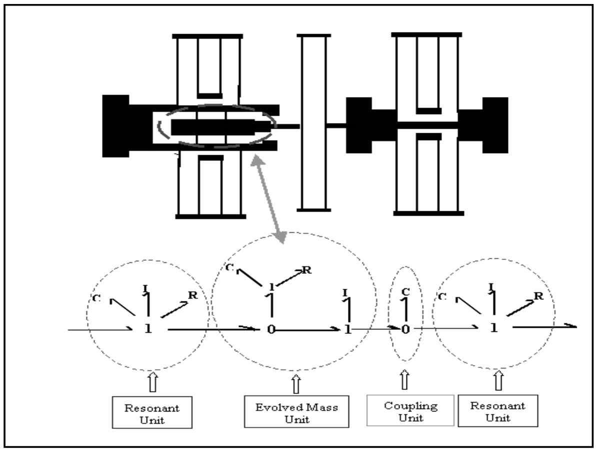

Fig. 11. A novel topology of MEM filter and its bond graph representation

4. Extensions

In MEMS, there are two or three levels of designs that need to be synthesized. 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.

For the second level -- two-dimensional layout designs of cell elements -- layout synthesis usually takes into consideration a large variety of design variables and design constraints. The most popular synthesis method seems to be based on conventional numerical optimization methods. The design problem is often first formulated as a nonlinear constrained optimization problem and then solved using an optimization software package [3]. Geometric programming, one special type of convex optimization method, is reported to synthesize a CMOS op-amp. The method is claimed to be both globally optimal and extremely fast. The only disadvantage and limitation is that the design problem has to be carefully formatted first to make it suitable for the treatment of the geometric programming algorithm. However, all the above approaches are based on the assumption that the structures of the cell elements are relatively fixed and subject to no radical topology changes [9]. A multi-objective evolutionary algorithm approach is reported for automatic synthesis of topology and sizing of a MEMS 2-D meandering spring structure with desired stiffnesses in certain directions [10].

The third level design calls for FEA (Finite Element Analysis). FEA is a computational method used for analyzing mechanical, thermal, electrical behavior of complex structures. The underlying idea of FEA is to split structures into small pieces and determine behaviors of each piece. It is used for verifying results of hand calculations for simple model, but more importantly, for predicting behavior of complex models where 1st order hand calculations are not available or insufficient. It is especially well suited for *iterative* design. As a result, it is quite possible that we can use an evolutionary computation approach to evolve a design using evaluation by means of FEA to assign fitness. Much work in this area has already been reported and it should also be an ideal analysis tool for use in the synthesis loop for final 3-D structures of MEMS. 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 fabricate procedures. Automated mask layout and process synthesis tools will be very helpful to relieve the designers from considering the fabrication details and focus on the functional design of the device and system instead [11]

Our long time task of research is to include computational synthesis for different design levels, and to provide support for design engineers in the whole MEMS design process.

5. Conclusions

This paper has suggested a design methodology for automatically synthesizing system-level designs for MEMS. For design of systems like the MEM filter problem, with strong topology constraints and fewer topology variations allowed, the challenge is to define a realizable function set that assures the evolved design is physically realizable and can be built using existing or anticipated technologies. Experiments show that a mixture of functions from both a modular function set and a basic function set form a realizable function set, and that the BG/GP algorithm evolves a variety of designs with different levels of topological complexity that satisfy design specifications.

Acknowledgment

The authors gratefully acknowledge the support of the National Science Foundation through grant DMII 0084934.

References

1. Fan Z., Hu J., Seo K., Goodman E., Rosenberg R., and Zhang B.: Bond Graph Representation and GP for Automated Analog Filter Design. Genetic and Evolutionary Computation Conference Late-Breaking Papers, San Francisco. (2001) 81-86
2. Fan Z., Seo K., Rosenberg R. C., Hu J., Goodman E. D.: Exploring Multiple Design Topologies using Genetic Programming and Bond Graphs. Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2002, New York. (2002) 1073-1080.
3. Zhou Y.: Layout Synthesis of Accelerometers. Thesis for Master of Science. Department of Electrical and Computer Engineering, Carnegie Mellon University. (1998)
4. Rosenberg R. C.: Reflections on Engineering Systems and Bond Graphs, Trans. ASME J. Dynamic Systems, Measurements and Control, (1993) 115: 242-251
5. Koza J. R.: Genetic Programming II: Automatic Discovery of Reusable Programs, The MIT Press (1994)
6. Seo K., Goodman E., and Rosenberg R.: First Steps toward Automated Design of Systems Using Bond Graphs and Genetic Programming, Proc. Genetic and Evolutionary Computation Conference, San Francisco (1-page abstract) and poster (2001) 189.
7. Wang K. and Nguyen C. T. C.: High-Order Medium Frequency Micromechanical Electronic Filters, Journal of Microelectromechanical Systems. (1999) 534-556
8. Hu J., Goodman E. D.: Hierarchical Fair Competition Model for Parallel Evolutionary Algorithms. CEC 2002, Honolulu, Hawaii, May, (2002)
9. Hershenson M. M, Boyd, S. P., and Lee T.H.: Optimal Design of a CMOS Op-Amp via Geometric Programming. Computer-Aided Design of Integrated Circuits and Systems (2001) 20(1): 1-21
10. Zhou N., Zhu B., Agogino A., Pister K.: Evolutionary Synthesis of MEMS design. ANNIE 2001, IEEE Neural Networks Council and Smart Engineering System Design conference, St. Louis, MO, Nov 4-7, (2001).
11. Ma L. and Antonsson E.K.: Automated Mask-Layout and Process Synthesis for MEMS. Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems (2000) 20-23