A Bond Graph Representation Approach for Automated Analog Filter Design

Zhun Fan
Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824

Jianjun Hu
Computer Science and Engineering
Michigan State University
East Lansing, MI 48824

Kisung Seo
Genetic Algorithms Research and Applications Group
Michigan State University
2857 W. Jolly Rd., Okemos, MI 48823

Erik D. Goodman
Genetic Algorithms Research and Applications Group
Michigan State University
2857 W. Jolly Rd., Okemos, MI 48864

Ronald C. Rosenberg
Dept. of Mechanical Engineering
Michigan State University
East Lansing, MI 48824

Baihai Zhang
Dept. of Mechanical Engineering
Michigan State University
East Lansing, MI 48824

Abstract

We present a novel circuit representation scheme, namely bond graph, along with strong-typed genetic programming for the evolution of analog filter circuits. Bond graph is a concise and uniform language for the description of circuit systems and more general engineering systems. Many unique characteristics of bond graph makes it an attractive candidate for representing circuit in genetic programming design. The feasibility and efficiency of using bond graph as the representation technique of circuit systems are verified in our experiments with automated analogue filter design.

1 INTRODUCTION

Automatic synthesis of analog circuits is of great significance for electronic systems design, which involves the determination of the topology and sizing of the circuits. A variety of techniques have been applied in this area. Some methods incoorporated heuristics, some predefine the topology and then the automated procedure will optimize the sizing of the circuits. Some use divided stage of topology optimization with GA and parameter optimization with numerical optimization methods(Grimbleby, 1995). Some genetic algorithm approaches could also evolve both topologies and component parameters, however they allow only at most a limited amount of components to be evolved(Lohn, 1999). Using netlists as the representation technique for the circuit, and genetic programming as the evolutionary tool, Koza develops a set of very successful approaches to deal with circuit synthesis problems, evolving topologies and parameters simultaneuosly (Koza, 1999). However, their applications are confined in electrical domain currently and prohibitive computation cost also provides room for improvement.

In this paper, a different circuit representation scheme, namely bond graph, along with strong-typed genetic programming is used for the evolution of analog filter circuits. Bond graph is a concise and uniform language for the description of circuit systems and more general engineering systems. This makes bond graph an attractive candidate for representing circuit in genetic programming design. We test the feasibility and efficiency of using bond graph as the representation technique of circuit system in auotmated design of analogue filter design. Our results show that bond graph is a good candidate for dynamic system synthesis.

2 BOND GRAPH IN DESIGN

2.1 BOND GRAPH REPRESENTATION OF CIRCUITS

In the context of circuit system design, a bond graph consists of the following types of elements:

- C, I, and R elements, which are passive one-port elements that contain no sources of power, and represent capacitors, inductors, and resistors.

- Power source elements including S_e and $S_{f.,}$ which are active one-port elements representing sources of voltage or current, respectively. In addition, when the current of a current source is fixed as zero, it can serve as an ideal voltage gauge. Similarly, when the voltage of a voltage source is fixed as zero, it can serve as an ideal current gauge

- Transformer (TF) and gyrator (GY), which are two-port elements, and represent transformers and gyrators, respectively. Power is conserved in these elements.
- Junction 0 and 1, which are multi-port elements representing series and parallel relations among elements. They served to interconnect elements into subsystem or system models

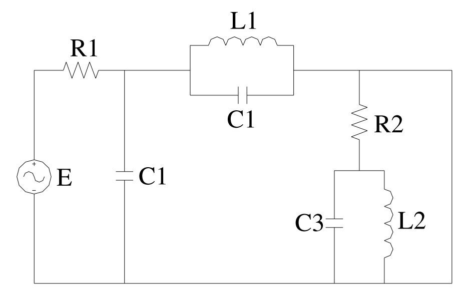
Bonds are used to connect any two elements in the bond graph.

2.2 ADVANTAGES OF BOND GRAPH REPRESENTATION IN GENETIC PROGRAMMING

It is clear that there is a correspondence relationship between a bond graph and its represented analog circuits. Actually, an automated procedure can be designed to transform a bond graph into a circuit. Based on this, we only considering the evolutionary design of a analog filter circuits represented by bond graph. Here the bond graph serves as an intermediate representation between the genotype, which is a genetic programming tree, and the phenotype, which represents an analogue circuit.

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, civil, hydraulic, and etc (Karnopp et al. 2000). Bond graph models can describe the dynamic behavior of physical systems by a directed graph consisting of idealized lumped elements based on the principle of conservation of power. These models provide very useful insights into the structures of dynamic systems.

The unique characteristic of bond graph is using junction 0 and 1 to represent the series and parallel relationships among components in normal complicated circuits. In fact, it is this concept that leads to the foundation of bond graph field (Paynter, 1991). Junctions transform common circuits into a very clean structure with few loops, which make the normal circuits very complicated. Figure 1 shows the comparison of a circuit and a corresponding bond graph. It is predicted that junction 0 and junction 1 representation mechanism can reduce circuit duality to logical duality and then has a clear computational advantage. The evaluation efficiency of the bond graph model is further improved due to the fact that analysis of causal relationships and power flow between elements and subsystems could reveal the system properties and inherent characteristics. This makes it possible for us to discard infeasible design candidates even before numerically evaluating them, thus reducing time of evaluation to a large extent. In addition, as virtually all of the circuit topologies created are valid, our system does not need to check validity conditions of individual circuit to avoid singular situations that could pull down the continuous run of program when evaluating them.



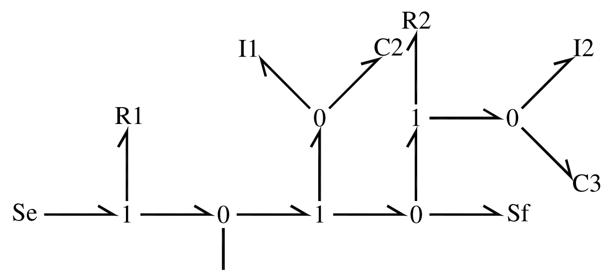


Figure1. Bond graph Representation of a Electrical Circuit

The other characteristic of bond graph is its resemblance to engineering design process (Xia, et al. 1991). As each component of the system can be represented correspondingly in bond graphs, junctions and elements could be added into or deleted from a model without causing too many changes. This emulates the engineering process of modifyng systems, refining simple designs discovered initially, adding size and complexity as needed to meet more complicated design demands step by step. As Genetic programming usually shows a weak causality of structure evolution (Rosca, J. P. 1995), this potential strong causality of bond graph modification process also makes bond graph representation an attractive technique to be used in genetic programming to explore the open electrical circuit system design space in evolutionary process.

Figure 3 gives a detailed illustration of a genetic programming tree that creates the bond graph of evolved high pass filter circuit that is shown in figure 5. With the help of genetic operators defined in (Seo et al. 2001), the genetic programming tree could create a large range of useful bond graph representations of electrical circuits flexibly.

3 GENETIC PROGRAMMING WITH A BOND GRAPH REPRESENTATION

3.1 FUCNTION SET AND TERMINAL SET

Design of function and terminal sets is of great importance in genetic programming design and is closely related to the representation of problem domain.

Corresponding to Koza's work, only the following set of bond graph elements: [C, I, R; 0, 1; Se, Sf] are used in this paper. This set is sufficient to explore meaningful design problems such as the filter design problem. The function set and terminal set used are listed at Table 1. All these operations operate on the embryo bond graphs and grow them into desired bond graphs. They ensure efficient yet complete explorations of the search space, with only a very small subspace missing from the whole possible search space. Among them Ins_J0, Ins_J1 is very flexible in manipulating the structure of bond graph. Details of these function and terminals are explained in (Seo et al. 2001).

Table 1 Function and terminal set for Bond graph evolution

Name	Description
add_C	Add a C element to junctions
add_I	Add an I element to junctions
add_R	Add an R element to junctions
insert_J0	Insert a 0-junction in bond
insert_J1	Insert a 1-junction in bond
replace_C	Replace current element with C element
replace_ I	Replace current element with I element
replace_ R	Replace current element with R element
+	Add two ERCs
-	Subtract two ERCs
endn	End terminal for add element operation
endb	End terminal for insert junction operation
endr	End terminal for replace element operation
erc	Ephemeral random constant (ERC)

4 EXPERIMENTS AND RESULTS

Filter design problem is used as a test for our approach of representing electrical circuit with bond graph. We use converted Matlab routines to evaluate frequency response of filters created. As Matlab provides many powerful toolbox for engineering computation and simulation, it facilitate development of source codes for our genetic programming evaluation dramatically. In addition, as all individual circuits to be evaluated are causally valid, the possibility of singular accurance is excluded, which enables the program to run continuously without interruption.

4.1 EMBRYO CIRCUIT AND BOND GRAPH

All individual genetic programming trees create circuit and bond graph from an embryo. Selection of embryo circuit is also an important topic in electrical circuit design, especially for multi-port systems. In our filter design problems, we use the following circuit and bond graph as our embryo, which is shown in figure 2.

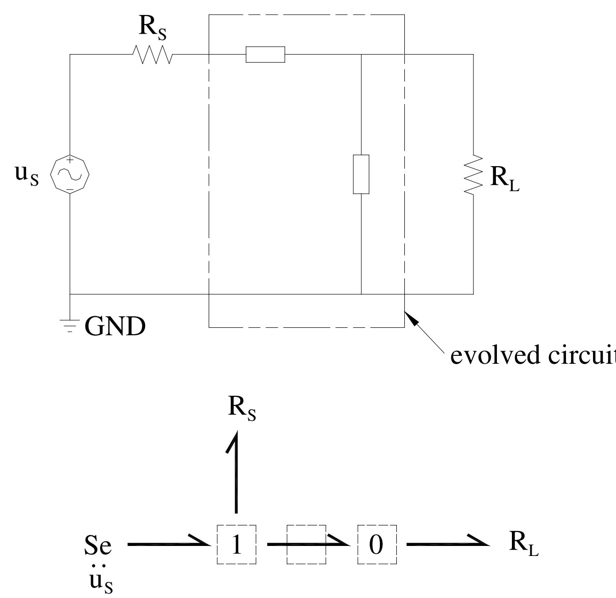


Figure 2. Embryo Circuit and its Bond graph Representation

4.2 DEFINITION FITNESS FUNCTION

Definition of fitness function is as the following:

Within interested range of frequency, uniformly sample 100 points. Compare the magnitudes of the sample points with target magnitudes, compute their difference and get a squared sum of difference as raw fitness, defined as $Fitness_{raw}$. Then normalized fitness is calculated according to formula

$$Fitness_{norm} = 0.5 + {}^{1}\!/(1+Fitness_{raw})$$

4.3 EXPERIMENTAL SETUP

We used a strongly-typed version [Luke, 1997] of lilgp [Zongker and Punch, 1996] to generate bond graph models. The GP parameters were as shown below

Number of generations: 500

Population size: 500

Initial population: half_and_half

Max depth: 16

Initial depth: 4-6

Max nodes: 1000

Selection: Tournament (size=7)

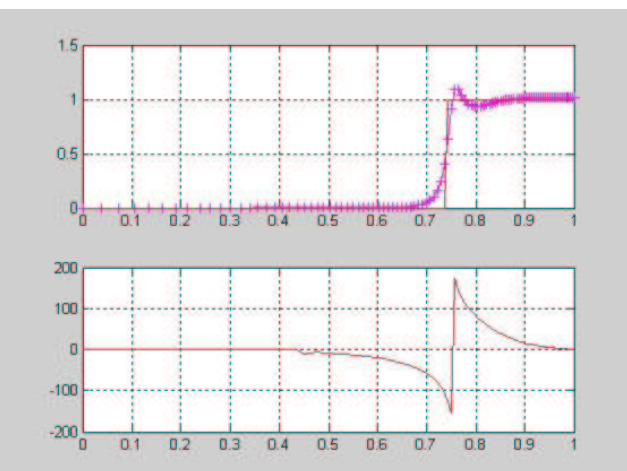
Crossover: 0.8

Mutation: 0.2

4.4 EXPERIMENTAL OBSERVATIONS

4.4.1 High pass filter design problem

The freuqency output of high pass filter evolved is show nat figure 4. The upper part is magnitude frequency response, while the lower part describes the phase frequency response. From the figure we could see that the result is quite satisfactory.

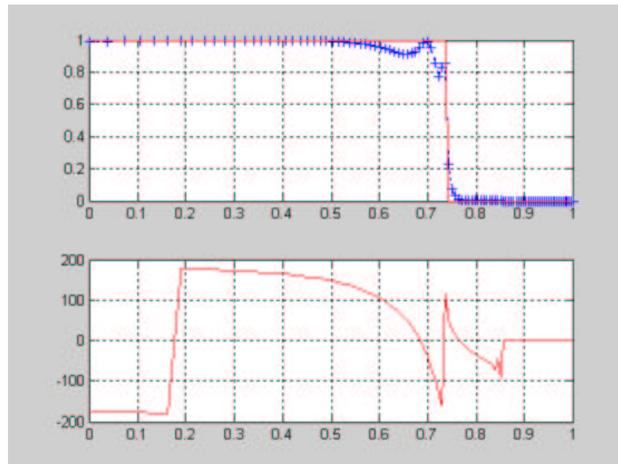


To get this result, our program run in a PIII 550 for 121.9 minutes. It took the genetic programming algorithm to evolve 272 generations.

The evolved high pass filter circuit and bond graph are also show in figure 5. From the evolved bond graph, we could see that there still exists a lot of topology parts that could be simplified. We call it topology redundancy and this phenomena are observed again and again in our research. We think that this kind of redundancy in topology is useful for evolution. It helps the search process of genetic programming to bypass fitfall landscape of search space and avoid to be stuck in local minimum easily .

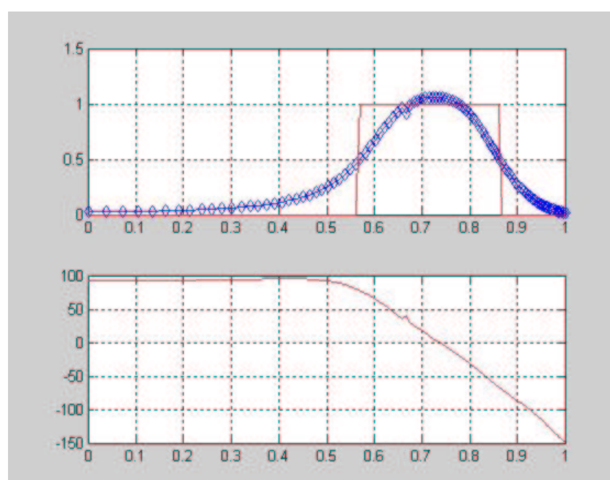
4.4.2 Low pass filter design problem

Figure 6 gives the frequency output result of the evolved low pass filter. The program run for 151.8 minutes and genetic programming algorithm evolved 157 generations to get this result.



4.4.3 Band pass filter design problem

Figure 6 gives the frequency output result of the evolved low pass filter. The program run for 193.8 minutes and genetic programming algorithm evolved 270 generations to get this result. Obviously it is the most difficult oen of the three filter design problems.



5 CONCLUSIONS

As a concise and uniform language for the description of circuit systems and more general engineering systems, bond graph proves to be a desirable representation of circuit in terms of automatic filter design with genetic programming. Bond graph has several unique characteristics: First, junction 0 and junction 1 representation mechanism can reduce circuit duality to logical duality. Second, the evaluation efficiency of the bond graph model is high as analysis of causal relationships and power flow between elements and subsystems could reveal the system properties and thus make it possible to discard infeasible design candidates even before numerically evaluating them. Third, bond graph representation of circuit permit almost all kinds of circuit systems to be created if we include loop manipulation in our function set. It can also be extended to represent diodes, transistors and other energy sources of circuit system. Last but not least, potentially strong causality of bond graph modification process makes bond graph representation very suitable in evolutionary process of genetic programming. Our experiments show that bond graph representation of circuit systems along with strong-typed genetic programming could yield satisfactory filter design in moderate time and computation expense.

To make the automated design of electrical circuits more applicable in practice, several research directions are taken by our group: First, design novel function sets of genetic programming to improve current design, compare their advantages and disadvantages over current function set, and meanwhile get more insights into the evolutionary process of genetic programming. Second, use parallel realization of genetic programming to enhance its ability of search and improve computation efficiency. Third, as practical designs usually have much more constraints and criteria in consideration, such as ensitivity criteria of the circuit, we are going to implement multiobjective genetic programming to tackle this problem. Last, successful extension to auotmated design of mechatronic systems will be a big breakthrough and is the target of our group.

Acknowledgments

This paper is supported by National Science Foundation under Grant DMI 0084934.

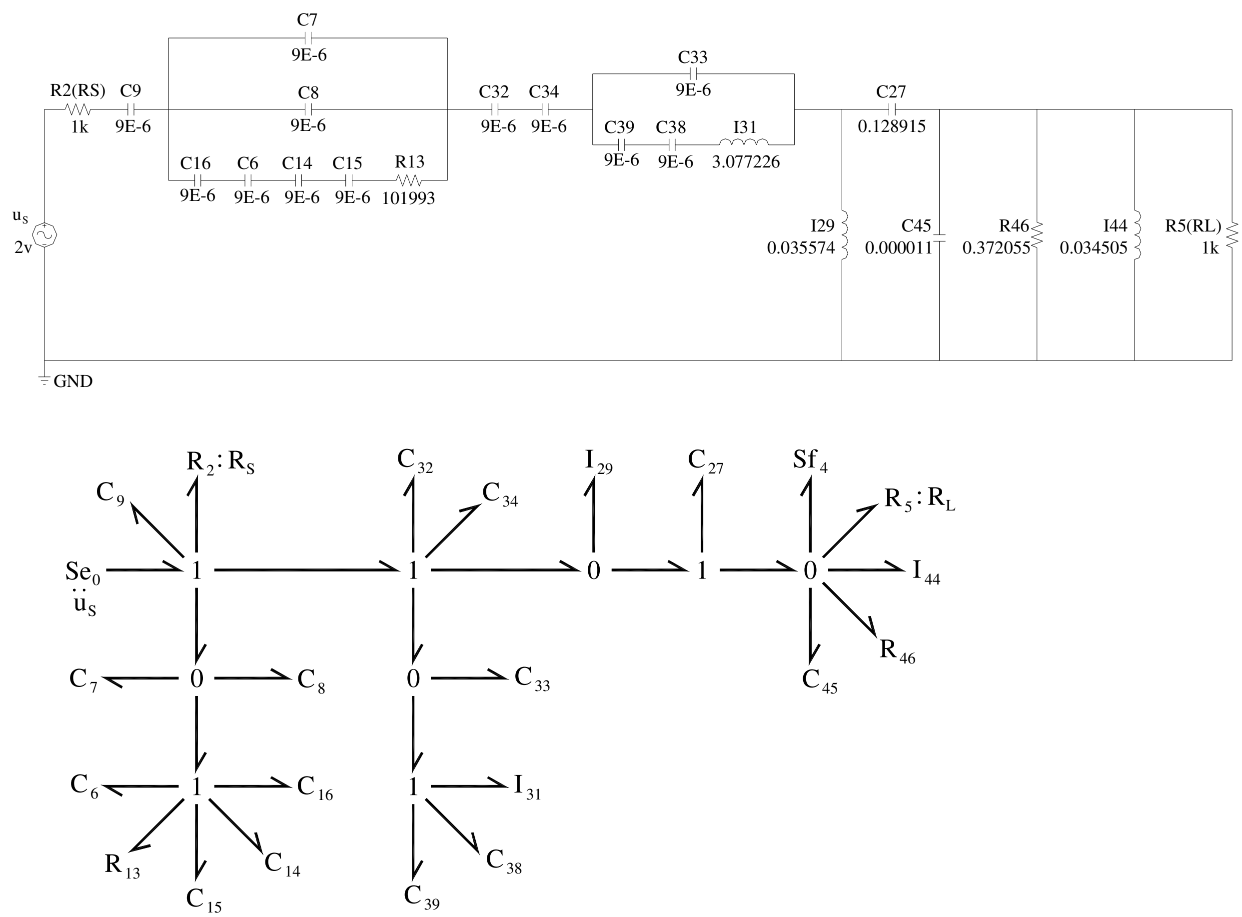


Fig 5. Evolved High Pass Filter Circuit and its Bond graph Representation

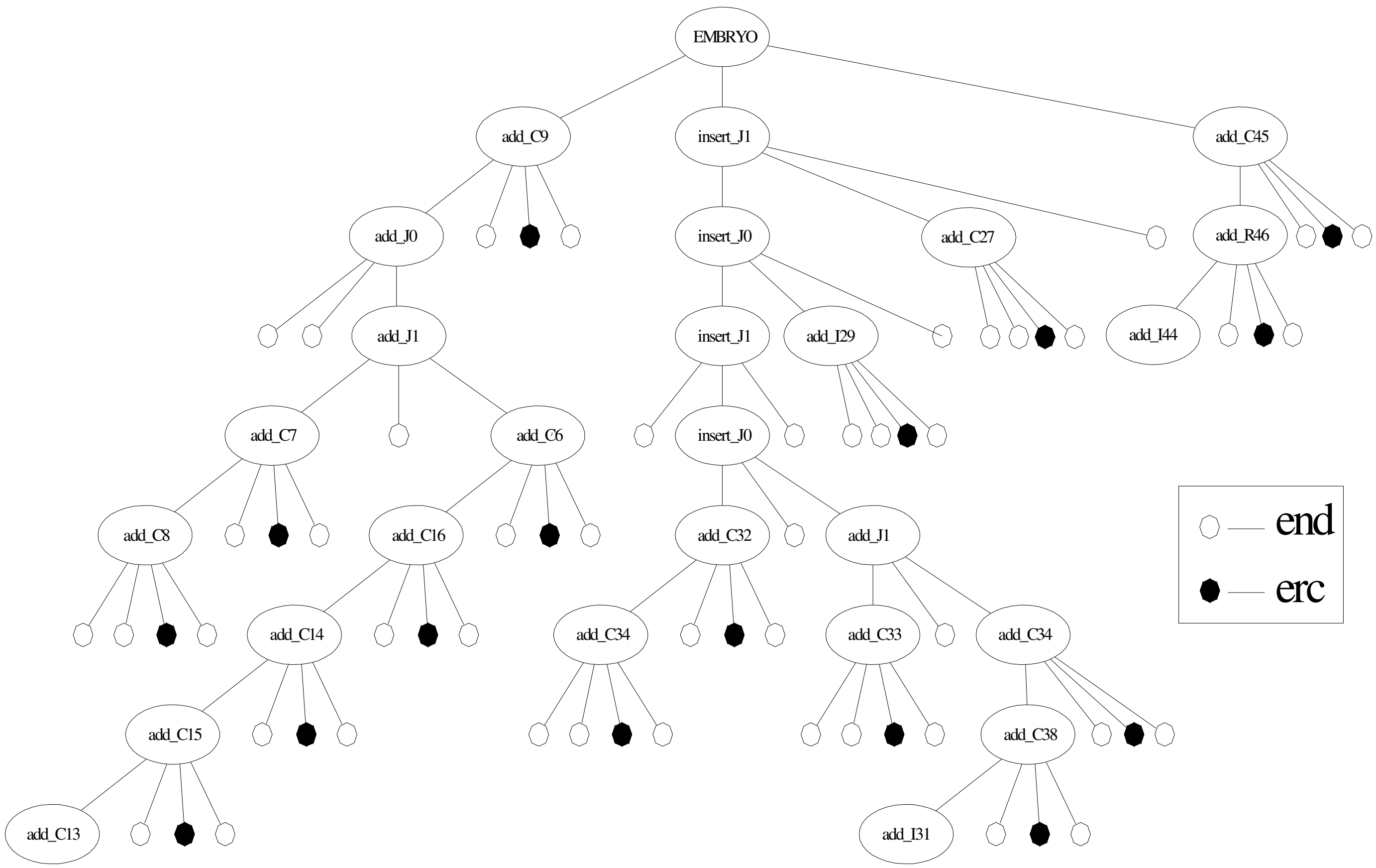


Figure 3. Genetic Progamming Tree that Creates Bond Graph for Electrical Circuit

References

J. B. Grimbleby (2000). Automatic analogue circuit synthesis using genetic algoriths. *IEE Proc. – Circuits Devices Syst.* : 319-323. #

D. C. Karnopp, D. L. Margolis and R. C. Rosenberg (2000). *System Dynamics: Modelling and Simulation of Mechatronic Systems. Third Edition*. New York: John Wiley & Sons, Inc.

J. R. Koza, F. H. Bennett III, D. Andre and M. A. Keane (1999b). The design of analogue circuits by means of genetic programming. In P. J. Bentley (ed.), *Evolutionary Design by Computers*, 365-385. London: John Wiley & Sons Ltd.

J. D. Lohn, S. P. Colombano (1999). A circuit representation techniques for automated circuit design. *IEEE Transactions on Evolutionary Computation*: 205-219. #

H. M. Paynter (1991). An epistemic prehistory of bond graphs. In P. C. Breedveld and G. Dauphin-Tanguy (ed.), *Bond Graphs for Engineers,* 3-17. Amsterdam, The Netherlands: Elsevier Science Publishers.

J. P. Rosca , D. H. Ballard (1995) Causality in genetic programming. In L. Eshelman (ed.), Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95), 256-263. San Francisco, CA: Morgan Kaufmann.

K. Seo, E. D. Goodman and R. C. Rosenberg (2001). First steps toward automated design of mechatronic systems using bond graphs and genetic programming. GECCO 2001

S. Xia, D. A. Linkens and S. Bennett (1991). Integration of qualitative reasoning and bond graphs: an engineering approach. In P. C. Breedveld and G. Dauphin-Tanguy(ed.), *Bond Graphs for Engineers,* 323-332. Amsterdam, The Netherlands: Elsevier Science Publishers.