A Robust Evolvable System for the Synthesis of Analog Circuits
Un Sistema Evolutivo Robusto para la Síntesis de Circuitos Analógicos

Aurora Torres Soto[1], Eunice E. Ponce de León Sentí[1], Arturo Hernández Aguirre[2], María Dolores Torres Soto[3] and Elva Díaz Díaz[4]
[1]Universidad Autónoma de Aguascalientes. Departamento de Ciencias de la Computación
atorres@correo.uaa.mx, eponce@correo.uaa.mx
[2]Centro de Investigación en Matemáticas. Departamento de Ciencias de la Computación
artha@cimat.mx
[3]Universidad Autónoma de Aguascalientes. Departamento de Sistemas de Información
mdtorres@correo.uaa.mx
[4]Instituto Tecnológico y de Estudios Superiores de Monterrey. Campus Aguascalientes
elva.diaz@itesm.mx

Article received on July 12, 2009; accepted on November 11, 2009

Abstract
This paper presents a group of evolutionary mechanisms for the design of analog circuits, embedded on a genetic algorithm that performs the synthesis of an analog filter. The algorithm interacts with SPICE, to evaluate the fitness of evolved circuits. In order to model an analog circuit, a linear representation is introduced and its corresponding reproduction operators that preserve the valid topological analog circuit class closed. The novelty of this paper consists of the use of a linear representation in combination with the generation mechanism and closed operators that keep the non SPICE simulable circuits below one percent. Furthermore, the concept of preferred values is used into the generation mechanism and genetic operators in order to reduce the gap between the real circuits and the evolvable ones. The performance of the system at designing passive low pass filter is discussed and experiments performed show its efficiency.
Keywords: Analog Circuit Synthesis, Analog Filter Design, Genetic Algorithm, SPICE Simulation.

Resumen
Este artículo presenta un grupo de mecanismos evolutivos para el diseño de circuitos analógicos, integrados en un algoritmo genético que desarrolla la síntesis de un filtro analógico. El algoritmo interactúa con SPICE para evaluar la adaptabilidad de los circuitos evolucionados. Para modelar un circuito analógico, se emplea una representación lineal y operadores de reproducción que mantienen cerrada la clase de los circuitos topológicamente válidos. La novedad de este artículo consiste en el uso de la representación lineal en combinación con el mecanismo de generación y los operadores cerrados, de manera que se conserve el porcentaje de los circuitos no-simulables por SPICE, debajo del 1%. También se ha integrado el concepto de valores comerciales dentro de los mecanismos de generación y operadores genéticos, para reducir las discrepancias entre los circuitos implementados y los circuitos evolucionados. Este trabajo describe el desempeño del sistema mediante el diseño de un filtro pasa-bajas y su eficiencia.
Palabras clave: Síntesis de Circuitos Analógicos, Diseño de Filtros Analógicos, Algoritmo Genético, Simulación con SPICE.

1 Introduction

Although digital circuits have become very popular nowadays, almost all electronic circuits in today's era use analog circuits as essential building blocks [Das and Vemuri, 2007].

The difficulty of the design process of analog circuit is well known, since implies vast knowledge, experience and time, even for experienced designers. Another reason of its difficulty compared with digital design is related with the fact that analog design does not have regular structures [Tlelo et al, 2007]. Analog design is also described by Das (2008), as traditionally less systematic, and more heuristic and knowledge intensive than the digital one.

Circuit synthesis is the process of designing and constructing a network to provide a prescribed response to a specified excitation [Goh and Li, 2001]. This process is composed for two phases: the selection of a suitable topology and the sizing of all its components. While topology consists of the determination of the type of components and its connections; sizing refers to the selection of the components values.

Several successfully efforts have been made in the automated analog circuit synthesis for about a decade; especially those that use non-conventional search techniques like genetic programming (GP) [Hu et al., 2005; Koza et al., 1997] and genetic algorithms (GA) [Das and Vemuri, 2007; Go and Li, 2001, Torres et al., 2007] among others. The non-conventional search techniques have been successfully introduced to circuit design problems due to the creativity to derive completely new topologies and its adaptation through the adjustment of circuit parameters [Ando and Iba, 2000].

Koza et al. (1997), developed a GP based approach, that let them to evolve both circuit topologies, as well as component sizes. Some other researchers have made similar approaches using GA. Lohn and Colombano (1998) for example, developed a linear representation for the automated circuit synthesis using passive elements, they used arrays of four bytes for representing a circuit. The first one was used for the opcode and the next three represented the component value. This implied the need of using a decode phase and limited the circuit topology. The papers [Zebulum et al., 1998] and [Lohn and Colombano, 1998] showed the use of GA on the passive filter synthesis while in [Zebulum et al., 1999], the goal was to evolve an active filter. In most recent works, researchers look for fixing the drawbacks detected in previous efforts. Go and Li (2001), incorporated some practical constraints into the development process, while Dastidar et al. (2005), presented a system representation scheme that employs a topological reuse-based approach. Khalifa et al. (2007), presented a technique to compare circuit sensitivities to changes in component values; Das and Vemuri (2007), worked on the assumption that two parents with comparable fitness function, produce better offspring. Hilder and Tyrrell (2007) proposed an evolutionary platform using an interconnected matrix of cells. Moreover Hu et al. [Hu et al., 2005], worked on the concept of robust design; they proposed a design paradigm for genetic programming. Among other aspects, the reliability of the circuits, the computational effort, the size of the circuit and the cost should be considered on the synthesis process in order to improve it.

In this paper, we present a robust group of mechanisms for the encoding, the generation and the evolution of analog circuits into a modified genetic algorithm; "Generator of Analog Circuit by a Genetic Algorithm" (GACGA System). Specifically, our goal is to eliminate the amount of topological incorrect circuits and reduce those that are non SPICE (Simulation Program with Integrated Circuits Emphasis) simulable in an evolvable process and, thus, reduce computational effort since to fix a non valid circuit implies a lot of computational resources and time. The results obtained by applying the GACGA system to the synthesis of passive low pass filter and the experiments performed are discussed.

The next sections have been organized in the following way; section 2 presents the general approach and some basic concepts. The core of section 3 is the description of the encoding and generation mechanisms working together with the genetic algorithm as well as evolvable operators. Section 4, describes the performed experiments and the results, conclusions and future work are discussed in section 5.

2 The approach

2.1 General Approach

The heuristic nature of analog circuit design [Das, 2008], makes this process much amenable for evolutionary techniques [Tlelo et al., 2007]; thus, we decided to use evolutionary computation techniques to create a system to represent, to generate and to combine circuits in order to explore the huge search space of this problem.

Early efforts on analog circuit synthesis were focused on the determination of the sizing (numerical values of all the circuit's components) on user supplied topologies. Later, the attention was put on the synthesis of both, topology and sizing [Koza et al. 1997].

Looking for an efficient topology representation, several data structures have been used; [Goh and Li, 2001 and Das and Vemuri, 2007] used arrays, [Hu et al., 2005 and Koza et al., 1997] chose trees and we selected linked lists. Linked lists let us create a very simple and flexible generation mechanism.

Since the most of computational resources and time in analog circuit synthesis is consumed by the fitness evaluation; and due to SPICE and some variations of it are the most used simulation tools in this area [Das and Vemury, 2007; Goh and Li, 2001, Lohn and Colombano, 1998, Koza et al., 1997, among others], we decided to create a group of evolvable mechanisms that reduce the amount of non SPICE simulable circuits generated in an evolvable process, preserving the valid topological analog circuit class closed.

The evolvable mechanisms are: 1) A linear representation technique, based on Ansi C linked links. 2) A robust generation mechanism and 3) A group of reproduction operators. These mechanisms were inserted into a modified genetic algorithm and the whole group was named GACGA system; that is discussed later in section 3.

To prove the effectiveness of the mechanisms; they were put to perform the synthesis of a high-order elliptic filter under different execution conditions. Three parameters of the system (genetic operator, growth index and maximum size) were tuned with three possible values each one and six replications were executed. Section 4 discusses these aspects in detail.

Next subsections include some basic concepts related with the mechanisms discussed later.

2.2 Preferred Values

Usual design approaches produce circuits in which component values are selected without any restriction. Then, this values are rounding to the nearest on a set of manufacturer's preferred ones; so it is not surprise that the performance of the real circuit differs from the evolvable one.

To ensure that the algorithm produces a circuit as close as possible to the actual circuit, we used manufacturer's preferred component values into the generation mechanism, according to E12 and E6 series, that were used due to their availability and low cost. The E-12 series is made up of 12 values per decade (10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68 and 82), while E-6 series has only 6 values per decade (10, 15, 22, 33, 47, 68). Also genetic operators have included this important aspect, so the offspring has only the manufacturer's preferred component values.

2.3 Low-pass Filter

Filters are circuits that block certain frequencies or bands of frequencies [Johnson, 2005]. A low pass filter is one that let pass low frequencies only and blocks high frequencies. Figure 1, illustrates the frequency response of an ideal low-pass filter. A real filter however, has a different behavior as shown in figure 2.

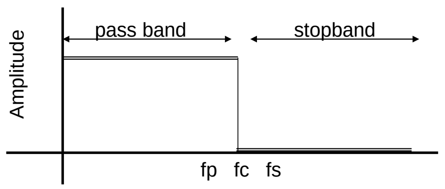

Fig. 1. The frequency response of an ideal low-pass filter

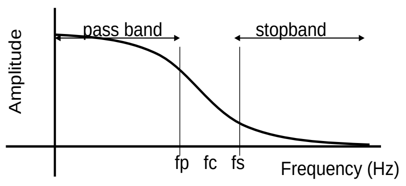

Fig. 2. The frequency response of a real low-pass filter

The low pass filter allows all signals below fc (cutoff frequency) to pass through and blocks those above it. Due to practical filter circuits, deploy variation of amplitude rejection with frequency (at the beginning of frequency, the signal is passed without effect; but as frequency grows up, the signal is effectively blocked), we left one don't care band located between the higher limit of the pass band (fp) and the lower limit of the stop one (fs). The attenuation of the filter is defined in terms of the maximum signal in its pass and stop bands. The low pass filter designed is described in section 4.

2.4 Encoding and Generation

One of the most important aspects in the success of any search algorithm is the representation of a solution. The encoding has to have several features that make it more or less suitable from problem to problem.

According to Mattiussi and Floreano (2007), there are two different encoding systems: Direct and developmental. Our approach uses the second one, since the genotype directs the construction of the electrical networks.

The generation mechanism is intimately related to the representation technique. Some desirable properties that were always in our minds are: Most circuits created are valid circuit graphs, the maximum number of elements in the circuit is a variable parameter, and the combination of the generation and encoding mechanisms of analog circuits lets us to perform quasi-closed genetic operators.

Avoiding non valid topologies, “GACGA” save worthy processing time and computational effort, considering that evaluation time is the longest.

3 Algorithm

The modified genetic algorithm developed, builds circuits located between initial and final nodes according to the template showed in figure 3, by means of an evolvable process. V_S is an ideal voltage source, R_S is the source resistance and R_L is the circuit load resistance.

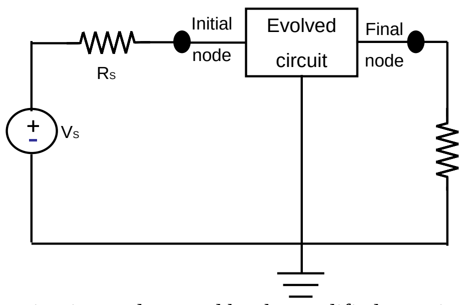

Fig. 3. The circuit template used by the modified genetic algorithm

3.1 Encoding Mechanism

As mentioned in section 2.4, a key for the success of a search algorithm, leans on the flexibility and robustness of the encoding system. In our experiments, we used the representation mechanism reported in [Torres et al., 2007] that is discussed later. In this encoding system, each candidate solution (chromosome) is represented by a dynamic data structure and thus its size adapts according to the number of elements on a circuit. A chromosome consists of several genes; each one represents an element of the circuit and its connections. An individual gene contains the information shown in figure 4. As can be seen from this figure, three parts of the gene are used to save topological information and the others keep the sizing circuit data. Even though in this work, three kinds of two terminals passive elements are used, this encoding may be extended to include any kind of two terminal elements.

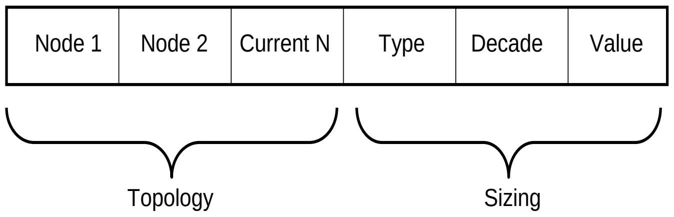

Fig. 4. The gene description

In figure 4, “Node 1” and “Node 2” refer to the nodes in each terminal of one element; while “Current N” is a node pointer used for the generation mechanism on the circuit creation process. Element “Type” could be 0(Capacitor), 1(Resistor) and 2(Inductor). “Decade” is the multiplier factor that forms components values together with “Value”. Along the present work, initial node is name “I”, final node is referred as “F” and ground as “T”. Table 1 shows how we encode the sizing data into each gene.

Table 1. Sizing encoding system

Type	Decade	Value
C(0)	$10^{-6} – 10^{-9}$ (0-3)	E6 (0-5)
R(1)	$10^{+3} – 10^{+6}$ (0-3)	E12 (0-11)
L(2)	$10^{-3} – 10^{-6}$ (0-3)	E12 (0-11)

E6 and E12 series were selected because they are the most common and versatile. Also Goh & Li (2001), demonstrated the advantage of using preferred values compared with unrestricted values, since when values are rounded off, the error increases and the response no longer falls within the specifications.

With regard to the sizing encoding system, the extension to another type of elements, values and decades, is not difficulty, since there is enough to use at different ranges.

As an example of the encoding system, figure 5 shows an element and its corresponding gene.

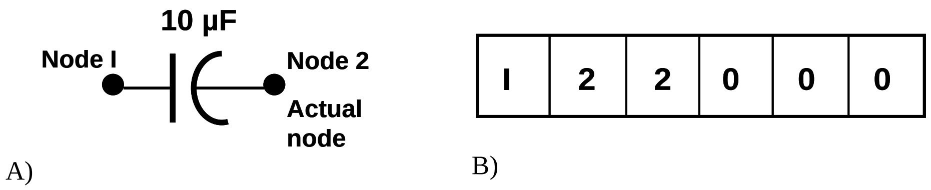

Fig. 5. A) An example of a circuit’s element and its connection nodes. B) The gene’s representation

3.2 Generation Mechanism and Genetic Algorithm

The generation mechanism works by means of a randomly generated operation code. In order to encourage the growth of circuits, we used a parameter named growth index. The GACGA system, also uses a maximum circuit size parameter that ends the generation circuit process if the circuit size reaches this parameter. Each instruction connects the actual node pointer to an existed or new node. The operation codes and their meaning are depicted in table 2.

Table 2. The operation code of the generation mechanism

Op code	Instruction
0	Connect to grown
1	Connect to final node
2	Connect to x node
3	Connect to new node

As seen in table 2, this circuit creation process is very flexible and easily implementable. Although it looks similar to the opcodes of Lohn and Colombano (1998), we have reduced the number of possibilities to four and we let the generation mechanism to finish the circuit building. Besides, the “Connect to x node” opcode, extends the search to very diverse topologies.

Once the operation code has been chosen, the type, the decade and the value of the new element are generated, and the corresponding instruction is executed. The circuit is built on linked lists to manage variable chromosome length.

The GACGA system is a modified genetic algorithm able to generate evolvable analog circuits. This system interacts with SPICE in the circuit evaluation and it has embedded the generation and encoding mechanisms discussed. The next pseudo code describes the general behavior of the GACGA system.

Algorithm 1. Pseudocode of the GACGA

```
1. Generate a Population of Pop_size individuals
2. Order Population by fitness
3. Select two progenitors
4. Select and execute the genetic operators producing
    two new members of the NewPopulation
5. Repeat from step 3 until NewPopulation has Pop_size
    members
6. Assign NewPopulation on Population
7. If the termination criterion is not satisfied then
    return to step 2
```

The Population generated is built by the generation mechanism previously discuss; after each circuit has been finished, its fitness and number of elements are calculated and saved for future use; once Population has Pop_size individuals, it is sorted by fitness and the building of NewPopulation begins.

The creation of NewPopulation consists on an iterative process; first the roulette method is started to select a pair of progenitors from Population and genetic operators are applied to them, as a result of this, two new members of NewPopulation are produced and their fitness and number of members are calculated and save and so forth until NewPopulation has Pop_size members. Then NewPopulation is assigned to Population and the NewPopulation building is repeated until termination criterion has been satisfied.

The fitness computation is made with the results of the simulation performed by SPICE on each circuit. This simulation is performed based on the circuit description (built by the generation mechanism) and leaded by the algorithm. Finally, the resulting frequency response is returned again to the genetic algorithm; see figure 6.

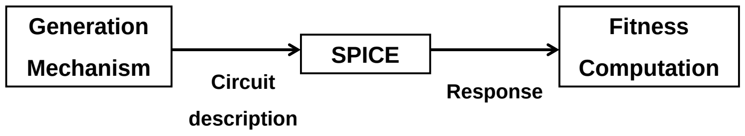

Fig. 6. The frequency response is obtained from SPICE

3.3 Fitness Function

The system created uses SPICE (Simulation Program with Integrated Circuit Emphasis) to evaluate the fitness of evolved circuits. The circuit description delivered to simulation, instructs SPICE to perform an AC sweep analysis.

The low-pass filter problem was extracted from [Koza et al., 1997] and is also discussed on [Hilder and Tyrrel, 2007 and Hu et al., 2005]; thus, the frequency response performance for a candidate filter is similar to those defined by Koza and other researchers. The fitness function is defined as a function of the sum of errors between the ideal frequency response and the actual candidate over N sampling points. The normalized version of the fitness function is shown in equation 3. This equation is a function of errors equations 1 y 2.

A sample error function "ε" give us the absolute deviation between the actual output response and the target response over the "i" sampling point. In equation (1), $M(f_i)_{Target}$ denotes target magnitude at a f_i frequency, $M(f_i)_{Actual}$ is the magnitude of the actual evolved circuit at a f_i frequency and f_i is the sampling frequency.

$$\varepsilon_i = \left| M(f_i)_{Target} - M(f_i)_{Actual} \right| \quad (1)$$

The sampling points range from 1Hz to 100 KHz, logarithmically distributed. If the deviation from target magnitude is inacceptable according to the frequency band, then a penalty factor "λ" has to be assigned to the error function. Equation (2), expresses the total error function "ξ" over the N points.

$$\xi = \sum_{i=1}^{N} \lambda(\varepsilon_i) * \varepsilon_i \quad (2)$$

As the error (ξ) is smaller, the better is the filter. A ξ=0 means the generated filter is the ideal one. The fitness "F" we employed is showed in equation (3).

$$F = \frac{1}{1+\xi} \quad (3)$$

According to equation (3), F is given by a normalized expression and a fitness of unit represents an ideal circuit, since ξ=0 means that there is not difference between the target and the actual filter.

Equation (3) transforms the problem of minimizing deviation from target frequency response, into a maximization problem.

3.4 Genetic Operators

Starting from two parents randomly chosen by roulette, an offspring is produce through three possible operators: crossover, crossover and mutation; and mutation. The way an operator is selected from these three possibilities is random. Section 4 shows how the algorithm parameters were tuned.

Crossover operation, introduces new solutions into the genetic algorithm beginning from previous circuits; this operator is the responsible of changing some parts of a circuit by parts from another one. According to Dastidar et al (2005) and Das and Vemuri (2007), to avoid invalid circuits topologies, achieved through suitable connectivity rules, can reduce the unwanted search space for both, active and passive circuit synthesis. Once a structurally correct first population is generated by means of our generation mechanism, genetic operators maintain very low the amount of non SPICE simulable circuits.

The crossover operator generates topological modifications because it alters the connection order of the offspring. Crossover performed by this version of the GACGA system is single crossover point, but we have already designed two-point crossover. Figure 7 illustrates how crossover operator is performed. For simplicity, each gene representation includes only the information about Node1, Node2 and Type of the element. This operator is performed in a way that avoids the generation of open circuits as well as short circuits; thus, the topologically correct class is keep closed.

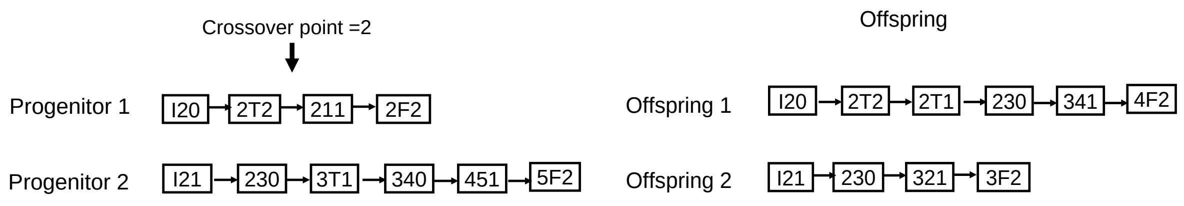

Fig. 7. An example illustrating the crossover operator

The mutation operator is executed at gene level; and it works by altering a randomly chosen gene with another one randomly generated. A mutated gene corresponds to a different type of element with different value, but connected to the same pair of nodes. Figure 8, shows an example of the use of this operator.

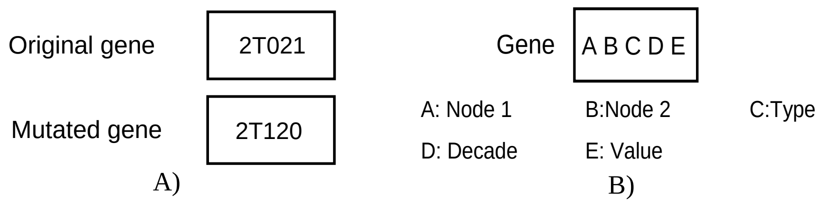

Fig. 8. A) An example of the mutation operator. B) The meaning of each allele

The crossover and mutation refers to the sequential application of both genetic operators. This operator has effects over both sizing and topology of the evolved circuits. This operator also has the objective to maintain closed the class of structurally correct circuits.

4 Experiments and results

As our design's benchmark, an analog low-pass filter was selected because it has been extensively studied by several researchers. The selected filter fulfills the next desired facts.

1. Is one of the most well defined passive analog circuits having a wide range of applications [Huelsman, 1993].
2. Its domain is established due to there are some well known design solutions and behavior specifications. This gives us an ample scope to verify our algorithm's performance.

The general filter specifications were extracted from [Koza et al., 1997], similar filters are also discussed in [Hilder and Tyrrel, 2007; Hu et al., 2005 and Torres et al., 2009]. This filter design was used in several approaches like genetic programming, genetic algorithms and estimation of the distribution algorithms among others. Figure 9 shows the general target behavior in our experiments.

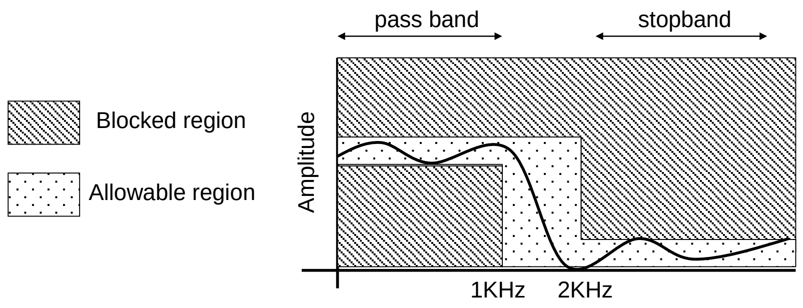

Fig. 9. Low-pass filter's specifications

The low pass filter shown in figure 9, allows all signals below 1 KHz to pass through (pass band) and blocks those above 2 KHz (stopband). We also established a "don't care" band, between 1KHz and 2KHz, in such a way that the fitness had not penalties independently of the output magnitude.

In the pass band, the allowable variation in the voltage magnitude variation is ±30mV (ripple voltage) whereas, in the stop band, this specification was fixed at ±10mV. The penalty factor "λ" used was 10, and the AC sweep analysis, performed by SPICE, measured the output voltage between 1 Hz and 100 KHz input, with twenty points per decade. The voltage values are then compared to the target voltages. Since the voltage source was set at 1V and the maximum acceptable voltage in the stopband is 10mV, the attenuation has to be at least 40dB (20xlog(0.010/1000)) between 1Kz and 2Khz. This goal can be satisfied by a high-order elliptic filter.

In order to demonstrate the effectiveness of the developed system, a set of experiments was performed. Table 3 shows the set of parameters that were altered as well as the used values to accomplish 6 replications of each parameters combination.

Table 3. Set of parameter values used on the experiments

Parameter	Tested Values
Population size	20
Number of generations	30
Genetic Operator	A, B, C
Growth Index	X, Y, Z
Maximum size	10, 15, 20

From table 3, it can be seen that each algorithm execution (from a total of 162), consist of 600 individuals (20x30). This is a very small amount when it is compared with Koza's (Population size about 320000) and other reported works (100000 circuit evaluations); thus, it is not surprising that the fitness values are low. However, the emphasis of this work was focused on the amount of non SPICE simulable circuits generated by means of the evolvable mechanisms, and the obtained results in this aspect are encouraging.

The genetic operator is a parameter that refers to the probability assigned to each one of the three genetic operators. The meaning of A, B and C values for this operator are shown on table 4.

The growth index is a parameter that our algorithm uses to establish different possibilities of node connections to the initial process of circuits generation. The later construction process has a uniform probability distribution. Table 5 shows the probability assigned to each possible node connection.

Table 4. The genetic operator values

Genetic Operator			
	Mutation	Crossover	Both
A	10%	60%	30%
B	30%	30%	40%
C	20%	50%	30%

Table 5. The growth index values

Growth Index			
	Ground	Final	New
X	30%	20%	50%
Y	20%	20%	60%
Z	10%	10%	80%

The maximum size refers to the maximum number of elements that a circuit could have. Although we maintained this parameter in low value, this restriction can be released. Again, the size of the generated circuits is adjusted, but the emphasis of this work was focused on the effectiveness of the mechanisms of generation and representation.

Table 6. Best parameters combinations and results

Parameters combination			Obtained results			
Genetic Operator	Growth Index	Maximum size	Fitness	Time (s)	Circuit size	Non SPICE simulations (%)
B	Z	15	0.003224	73.6	7	4.00%
A	Y	15	0.003052	33.4	3	0.17%
Average	executions		0.002889	52.2	5	5.97%

The variation parameters was introduced to demonstrate the effectiveness and robustness of our evolvable mechanisms in building simulable SPICE circuits, even though the GACGA's conditions are changing. We also use these experiments to calibrate our system for further work. Table 6 presents the best parameter combinations with regard to the fitness results and time; and the last line refers to the average results of the whole experiment.

The results were statistically analyzed and it was found that "Genetic Operator" is the parameter that most rebounds on execution time. The average time for each operator value is shown in table 7.

Table 7. Average of Time and Fitness according to Operator value

Genetic Operator	Time (s)	Fitness
A	40.083	0.002868
B	67.299	0.002870
C	49.246	0.002931

It can be seen (table 7) that operator B is the one who takes the longest time. To prove if there was a significant difference among group variances, homogeneity of variance in one-way model was used. The result was that there is not homogeneity of variances. Thus Shapiro-Wilk and Brown Forsythe tests were performed, obtaining that differences shown in the table 7 (respecting to time) are significant. In relation to Fitness, the average of the whole experiment was 0.002889, and using Kruskal-Wallis test, we found that there is not statistical evidence that parameters affect it.

In account of the amount of "Non SPICE Simulable Circuits", the average of non simulable circuits generated by each execution of the algorithm was less than 1%. Again, we used Kruskal-Wallis test and the results gave not significant differences among combination parameters of groups. Thus, the algorithm is stable.

In summary, these results allow us to conclude that the mechanisms that constitute the GACGA system described are very efficient to generate and reproduce SPICE simulable circuits; and that these mechanisms contribute to reduce the computational resources and the time in the artificial evolution of analog circuits.

5 Conclusions and future work

A robust group of evolutionary mechanisms was implemented into a genetic algorithm for the synthesis problem of a passive low pass filter. These mechanisms were tested under different algorithm operation conditions. Three solution parameters were taken into account to measure the system's performance: time, fitness and non SPICE simulable percentage. The GACGA system built topological correct circuits that were consistently SPICE-simulable (99%).

The best fitness function was obtained with BZ15 parameter combination while the best run time was reached with AY15. With respect to time, the genetic operator parameter was found to impact significantly, in such a way, that option "A"(10% mutation, 60% crossover and 30% of both) is the quickest combination. The parameters have not statistical effect on fitness nor the number of non SPICE simulable circuits produced. Due to this fact, we can postulate that independently of parameters values, the system avoids the generation of structurally wrong and non simulable circuits that normally consume a big deal of computational resources and time in the artificial evolution of analog filters. The evolvable process performed by the modified genetic algorithm and its operators, in combination with the proposed generation mechanism and the described circuit representation, do maintain the rate of non SPICE simulation circuits under a 1%. Consequently this set of mechanisms can be trustfully used for the synthesis problem of passive filters.

Currently we are working on the generation of new genetic operators and the problem models for its implementation in an EDA algorithm, looking to reduce the computational time in the whole evolvable process. Furthermore, we are analyzing the mechanisms of different metaheuristics to build robust hybrid design tools.

References

1. **Ando, S. & Iba, H. (2000).** Analog circuit design with a variable length chromosome. *2000 Congress on Evolutionary Computation*, La Jolla, CA , USA, vol. 2, 994-1001.
2. **Das A. & Vemuri R. (2007).** An Automated Passive Analog Circuit Synthesis Framework using Genetic Algorithms. *IEEE Computer Society Annual Symposium on VLSI, ISVLSI '07*, Porto Alegre, Brazil, 145-152.
3. **Das, A. (2008).** *Algorithms for Topology Synthesis of Analog Circuits*. PhD Thesis, University of Cincinnati, Cincinnati, USA.
4. **Dastidar, T. R., Chakrabarti, P. P. & Partha R. (2005).** A Synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse. *IEEE Transactions on Evolutionary Computation*, 9(2), 211–224.
5. **Goh, C. & Li, Y. (2001).** GA Automated Design and Synthesis of Analog Circuits with Practical Constraints. *IEEE International Conference on Evolutionary Computation*, Seoul, Korea, vol. 1, 170-177.
6. **Hilder, J. & Tyrrell A. (2007).** An Evolutionary Platform for Developing Next Generation Electronic Circuits. In *Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation*, London, United Kingdom, 2483-2488.
7. **Hu, J., Zhong, X. & Goodman, E. D. (2005).** Open-ended Robust Design of Analog Filters Using Genetic Programming. *2005 Conference on Genetic and Evolutionary Computation*, Washington, DC, USA, 1619-1626.
8. **Huelsman, L. C. (1993).** *Active and Passive Analog Filter Design: An Introduction*. New York: McGraw-Hill, Inc.
9. **Johnson, C. D. (2005).** *Process Control Instrumentation Technology* (8th edition). Englewood Cliffs. New Jersey: Prentice Hall.
10. **Khalifa, Y., Khan, B. & Taha, F. (2007).** Multi-objective Optimization Tool for A Free Structure Analog Circuits Design Using Genetic Algorithms and Incorporating Parasitics. *2007 GECCO Conference Companion on Genetic and Evolutionary Computation*, London, United Kingdom, 2527-2534.
11. **Koza, J. R., Bennethh, F., Andre, D. & Keane, M. A. (1997).** Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming. *IEEE Transactions on Evolutionary Computation*, 1(2), 109-128.
12. **Lohn, J. & Colombano, S. (1998).** Automated Analog Circuit Synthesis using a Linear Representation. *Second International Conference on Evolvable Systems: From Biology to Hardware, Lecture Notes in Computer Science*, 1478, 125-133.
13. **Mattiussi, C. & Floreano, D. (2007).** Analog Genetic Encoding for the Evolution of Circuits and Networks. *IEEE Transactions on Evolutionary Computation*, 11(5), 596-607.
14. **Tlelo, E., Duarte, M. A., Reyes, C. A. & Reyes, G. (2007).** Automatic Synthesis of Electronics Circuits using Genetic Algorithms. *Computación y Sistemas*, 10(3), 217-229.
15. **Torres, A., Ponce, E., Torres, M. D. & Luna, F. (2007).** Algoritmo Genético aplicado al Diseño Evolutivo de Circuitos Analógicos. Memorias *3er Congreso de Computación Evolutiva*. Aguascalientes, México, 13-19.
16. **Torres, A., Ponce, E., Torres, M. D., Díaz, E. & Padilla, F. (2009).** Comparison of Two Evolvable Systems in the Automated Analog Circuit Synthesis. *8th International Conference on Artificial Intelligence (MICAI 2009)*. IEEE press. In press.
17. **Zebulum, R. S., Pacheco, M. A. & Vellasco, M. (1998).** Comparison of different evolutionary methodologies applied to electronic filter design. *1998 IEEE International Conference on Evolutionary Computation*, Piscataway, New Jersey, USA 434-439.
18. **Zebulum, R. S., Pacheco, M. A. & Vellasco, M. (1999).** Artificial Evolution of Active Filters: A Case of Study. *NASA/DoD Workshop on Evolvable Hardware, IEEE Computer Society*, Los Alamitos, CA 66-75.

***Aurora Torres Soto** received the B.S. degree in Electronic Engineering from the San Luis Potosí Institute of Technology in 1993 and the M.S. degree in Information Technologies from Universidad Autónoma de Aguascalientes in 2001. She is currently studying the Ph. D. degree in Artificial Intelligence in Universidad Autónoma de Aguascalientes. She is a full time professor at Computer Science Department of Autonomous University of Aguascalientes, Mexico. Her research interests include Evolutionary Computation and Analog Circuit Synthesis.*

***Eunice Esther Ponce de León Sentí** received her PhD degree in Mathematics Science from Cybernetic, Mathematics and Physics Institute from Ministry of Science, Technology and Environment, Cuba in 1998. She works as full time professor-researcher at Computer Science Department of Autonomous University of Aguascalientes, Mexico. Her current research interests include, Combinatorial Optimization, Computational Complexity and Metaheuristics.*

***Arturo Hernández Aguirre** is a researcher and professor in the Computer Science Department of the Center for Research in Mathematics (CIMAT). His research interests are Multiobjective Optimization, Evolvable Hardware, Evolutionary Computation, Evolutionary design of Neural Network Architectures and Computational Learning Theory using Neural Networks.*

***María Dolores Torres Soto** received the B.S. degree in Computing Sciences from the San Luis Potosí Institute of Technology in 1993 and the M.S. degree in Information Technologies from Universidad Autónoma de Aguascalientes in 2001. She is currently studying the Ph. D. degree in Artificial Intelligence in Universidad Autónoma de Aguascalientes. She is a full time professor at Information Systems Department of Autonomous University of Aguascalientes, Mexico. Her research interests include Evolutionary Computation, Supervised and Unsupervised Learning, Typical Testors and Data Mining.*

Elva Díaz Díaz *received Ph.D. degree from Autonomous University of Aguascalientes, Mexico in 2008. She is working as a Professor-researcher in Computer Science Department of Autonomous University of Aguascalientes and ITESM campus Aguascalientes. Her research interest are on Machine Learning, Computational Complexity and Metaheuristics.*