GA Automated Design and Synthesis of Analog Circuits with Practical Constraints.

C. Goh
University of Glasgow,
Department of Electronics Electrical Engineering,
Rankine Building, Glasgow G12 8LG
cindy@elec.gla.ac.uk

Y. Li
University of Glasgow,
Department of Electronics Electrical Engineering,
Rankine Building, Glasgow G12 8LG
Y.Li@elec.gla.ac.uk

Abstract- This paper develops a Genetic Algorithm (GA) based "growing" technique to design and synthesise analogue circuits with practical constraints such as the manufacturer's preferred component values. Most existing problems with evolutionary search techniques applied to circuit design are addressed. The GA technique developed is then applied both to synthesise the topology of a network as well to perform value optimisation on the components based on a set of commonly used component values (E-12 series). Passive filter networks synthesise this way are realisable, effective and of novel topologies. It is anticipated that this technique can be extended to active networks.

Keywords - CAD, Genetic Algorithm, Circuit Synthesis, Preferred Value Components, PSpice.

1 Introduction

The development of circuit theory and computer technology has made it possible for an engineer to analyse and simulate almost any circuits in a design process. However, design (in the context of circuit synthesis) and analysis is two different processes (Darlington 1999). Circuit synthesis is the process of designing and constructing a network to provide a prescribed response to a specified excitation. It involves both the *topology* and *sizing* of all components. The topology of a circuit consists of the type of each component and its connections. The sizing of a circuit consists of the component value(s) associated with each component in the circuit. This is the converse of the analysis problem where a response is to be calculated when a prescribed excitation is applied to a given network. In contrast, the synthesis problem may not have a unique solution and very often the search space increases exponentially with time. This makes circuit design a classical problem for evolutionary techniques to solve (Drechsler 1998).

Recently, considerable progress has been made in optimising the design process of analogue and mixed-analogue-digital circuits using non-conventional search techniques such as Simulated Annealing (SA) (Lam and Zwolinski 1997), Genetic Algorithm (Goldberg 1989, Goh and Li 2000, GrimbleBy 1995, GrimbleBy 1997), Genetic Programming (GP) (Koza *et. al.* 1996, Koza *et. al.* 1997). These methods have all found success in applications such as component placement and routing, circuit synthesis, component value optimisation, etc. To date, however, applications of such techniques to analogue circuit design often overlooked the following factors:

- *Reliability of circuits* – In the design process, component values are assumed to have ideal and unrestricted values. They are then converted to a practical circuit by rounding the exact component values to the nearest preferred values based on a set of readily available manufacturer's preferred values for cost-effective reasons. Such a conversion would result in degraded or unreliable circuits, dependent upon the 'fitness landscape' (Li 1996).
- *Computational effort* – Owing to the nature of such search techniques most of the circuits produced in the beginning of the evolutionary process are faulty circuits, which are not realisable, thus wasting precious computational time (Koza *et. al.* 1996, Koza *et. al.* 1997).
- *Size of the circuit* - In general, a more complex network will provide a better fit to any target response than a simple one. However, this would normally result in relatively large circuits being produced, which are undesirable especially in the case of system-on-a-chip design.
- *Cost* – This is a very important issue in circuit design as it encompasses all areas of the design process. For instance, err in any of the 3 factors mentioned above would result in an increase in the cost.

The objective of this paper is therefore to overcome these problems in the approach reported in this paper. In particular, the following measures have been taken:

- *Preferred component values* – Manufacturer's preferred component values (E-12 series) were used in value optimisation to ensure that the simulation produce a circuit that is as close to the actual circuit as possible.
- *Valid circuit graphs* – Our GA ensures that all circuits created are valid circuit graphs, which are realisable as practical circuits, thus eliminating the need for pruning unconnected branches. In this way, the amount of faulty circuits is minimised and the computational effort and time is greatly reduced.

0-7803-6657-3/01/$10.00 ©2001 IEEE

- *Growing technique* – Our system adopts a growing technique where the user can specify the boundary limits on the size of the circuits. This ensures that the final circuit is optimal not only in terms of its response, but also in terms of its size and the cost of building it.

Building on the effectiveness of a GA in optimising component values of a circuit with a fixed topology, (Goh and Li 2000), a GA that is capable of designing practical and novel circuits by both synthesising the topology and optimising the component values simultaneously is presented in this paper. The next section focuses on the development of the methodology for circuit synthesis using the GA. Section 3 describes how optimal component values are selected from the set of manufacturer's preferred E-12 series component values. The applications and simulation results based on the proposed methodologies are presented in Section 4. Conclusions are drawn in Section 5.

2 GA Based Topology Synthesis

Figure 1 shows a circuit template used by the GA. Circuit being evolved is located between the input and output terminals. V_{source} is an ideal voltage source, R_{source} is the source resistance and R_{load} is the load resistance.

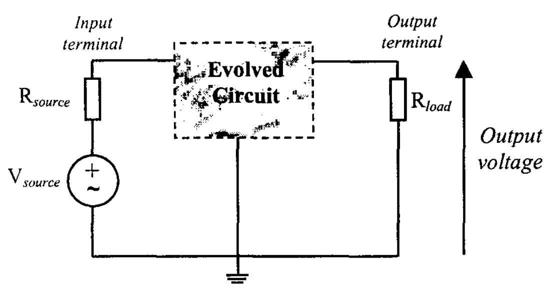

Figure 1. Circuit template used by the GA.

There are 2 ways in which circuit synthesis can take place:

- The user can allow the GA to freely evolve a circuit with all the given input specifications.
- If prior to the search, a suitable topology was known, the user can use this as a starting point for the GA. This approach requires the user to write the circuit netlist in the format given in Figure 2.

The GA utilises the main elements of a simple GA i.e. selection, reproduction, crossover and mutation. The main differences lie in the way these operations are performed and how real circuit elements are coded into genotypes.

Before starting the evolution, the user needs to input some specifications such as:

- A bound for the number of components in the circuit to be synthesised. The lower bound represents the initial starting number of components and the upper bound states the maximum allowable components in the circuit.
- Terminating conditions eg. Percentage of improvement before increasing number of components.
- Fitness function.
- Number of generations.
- Population size.
- The component types to be used.
- The range of values for each type of component.

```
* Schematic Netlist *

C_C2        2  0    0.012
L_L3        2  1    0.15
L_L4        1  3    0.047      Input node
R_Rload     3  0    1000
R_Rsource   2  4    1000       Output node
V_Vsource   4  0    DC 0V AC 2V
* comments begins with an asterisk.
```

Figure 2. Format of netlist.

The "growing" technique used by the GA is summarised in Figure 3.

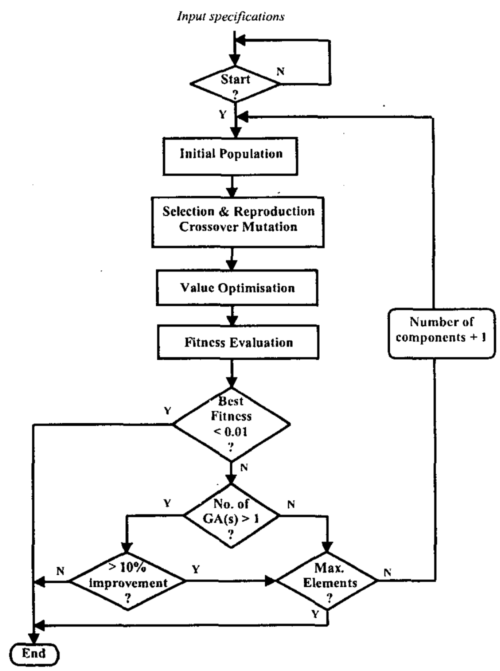

Figure 3. Overview of GA with "*growing*" technique.

The circuit is grown from the initial number of components and for each GA, the number of components is increased by one. However, if any of the terminating conditions are met before the maximum number of components is reached, the GA will be terminated. On the other hand, if the objective is not met when the maximum component number is reached, the GA will be terminated and the user then needs to refine the objective and terminating conditions. The advantages of growing a circuit are as follows:

- Offers the flexibility in the size of the resulting circuit. Unlike other methods discussed in the literature (Grimbleby 1995, Koza *et. al.* 1997), this method allows the user to control and restrict the number of components he desires in the final product.
- Minimise the number of components in the circuit by starting from the smallest possible.
- Minimise the waste of computer resources by using the smallest number of components if possible.

2.1 Circuit Representation

The representation of a solution is one of most important aspects in the successful working of any search algorithm. As the nature of solutions varies from problem to problem, a solution for a particular problem is also possible to be represented in a number of different ways. Thus, it becomes important to choose the representation that is most suitable for the chosen search algorithm. It is deemed that the representations should have the following desirable properties:

- It should allow almost any circuit within the design scopes to be represented .
- All circuits created are valid circuit graphs, thus reducing time and computer resources.
- If it is known prior to the search that certain topologies are suitable for the design task, the representation should permit the inclusion of these topologies in the search, thus reducing the search space.
- Time taken to transform the representation into a netlist for evaluation should be as short as possible.

Our GA was designed with all the above properties in mind. In this section, we describe how a circuit is represented in our GA. A gene consists of three elements namely nodes, values and type. Since the design task centers on the realisation of passive networks, only two-terminal components are used. For a circuit with n branches, there will be a maximum of $(n - 1)$ nodes. The value of the gene is stored in a lookup table and integer coding is used to work out its index accoding to a random number generated between 0 and 1. Each gene can take on four different types namely, null (0), resistor (1), inductor (2) and capacitor (3). Null genes are included to provide the flexibility to vary the size of the circuits created whilst keeping all the chromosomes at the same length. However, they will be removed when the chromosome is converted into a netlist for evaluation purpose.

A chromosome is made of the genes described above. It is also associated with a connection array showing how it is connected and a netlist used for evaluating its fitness. Figure 5 shows a chromosome of length 10 and the circuit created from it.

3, 4, 6, 0.4352	3, 6, 9, 0.9321	NULL	2, 9, 6, 0.3978	2, 9, 10, 0.2936	3, 10, 0, 0.2443	NULL	2, 10, 0, 0.0468	1, 10, 7, 0.4825	1, 7, 8, 0.6869

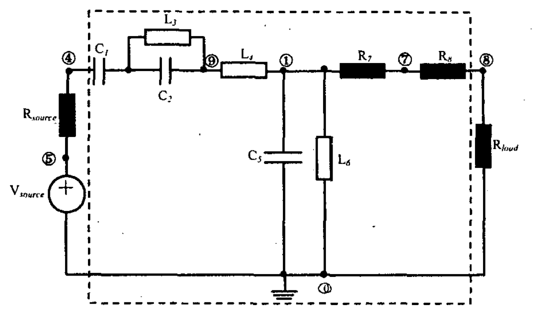

Figure 5. Chromosome and the circuit created from it.

2.2 Evolutionary Search

The GA works on a population of chromosomes which have the circuit representations as described in the above section. Random pairs of parents chosen by tournament selection were mated to produce a single offspring through single-point crossover. Crossover point was randomly chosen and the default crossover rate was set at 0.75. The mutation used in our GA is rather different from a mutation in the basic GA and will be illustrated in more detail. There are altogether 6 different types of mutation that can be performed and are described as follow:

- *Series mutation* – a new component is inserted in series with the mutated gene.
- *Parallel mutation* – a new component is inserted in parallel with the mutated gene.
- *Short-circuit mutation* – the mutated gene is replaced with a short-circuit.
- *Open-circuit mutation* – the mutated gene is replaced with an open-circuit.
- *Change element mutation* – the type of the mutated gene is changed.
- *Flip mutation* – the circuit is flipped by interchanging its input and output node.

Figure 5a and 5b illustrates how series and short-circuit mutation were performed respectively. It is evident that with these 6 types of mutations, any new trial solutions can be obtained from prior trial solutions without a premature loss of important information. The mutation rate used was between 0.1 to 0.3 depending on the size of the circuit.

Resistor R1 goes through series mutation

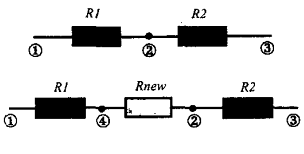

Figure 5a. Series mutation

Resistor R1 goes through short-circuit mutation

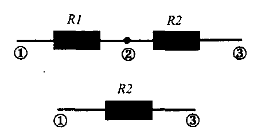

Figure 5b. Short-circuit mutation

3 Value Optimisation

A basic GA was used to optimise the component values based on a set of manufacturer's preferred values. The E-12 series which is made up of 12 values per decade (10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82) was used mainly because it is readily available and inexpensive, thus the overall cost to construct the evolved circuit would be low. The range of values which was used in our GA spans over 5 decades (61 values) and were determined by the type of components and the range specified by the user. For instance, capacitors have a lower bound of 0.1nF and an upper bound of 100uF. Inductors on the other hand have a lower bound of 0.1uH to 0.1H. The GA utilised uniform selection where each gene has a non-bias chance of being picked. Chromosomal information of each parent was integrated through single-point crossover with the crossover rate set at 0.75. Mutation rate was set at 0.1. The genes are represented as randomly generated real numbers between 0.0 –1.0 and integer coding was used to map these numbers to the index of a lookup table where the component values were stored. For example, 0.5 would mapped onto index 30 and 0.2463 would mapped onto index 14.

4 Applications and Results

This section applies the GA based design techniques developed in the previous two sections to two analogue filter design problems. Filters are chosen as the applications because it is a well-understood discipline within circuit design and has been extensively studied (Van Valkenburg 1982, Huelsman 1993). This allows us to compare and test the effectiveness of our evolved design to well-known designs. Lowpass filters are chosen because almost any other type of filters can be obtained from them eg. lowpass to highpass transformation. Figure 6 illustrates the frequency response of an ideal lowpass filter. However, an ideal lowpass filter is not realizable since a circuit can only be made up of a finite number of elements. In addition, the non-ideal properties of these elements also play a role in the resulting response. In order to circumvent this difficulty, the strict requirements must be modified towards tolerant specifications. The grey shaded areas in Figure 6 mark the unacceptable regions for the frequency response.

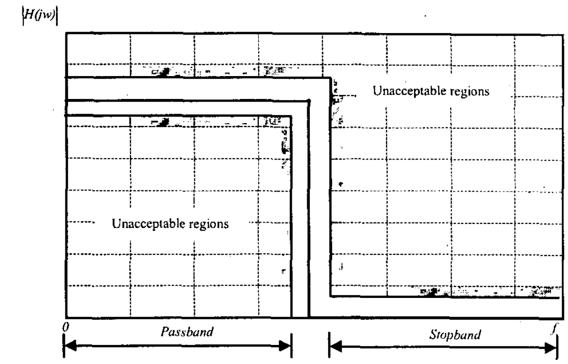

Figure 6. Lowpass filter – ideal shape and magnitude response tolerance.

The GA was asked to synthesise a suitable circuit that fits the target specifcations for two separate cases. In the first case, no starting point was provided and in the second case, a starting point was provided to direct the search.

4.1 Filter 1: Third Order Butterworth Filter

This design task centers around the design of a third order Butterworth filter. The reason for choosing this class of filter is because they are very common and circuits that implement them are readily found in filter design tables (Van Valkenburg 1982) and can be used as a comparison for our design. The Butterworth function takes the form of Eq. 1.

$$H(s) = \frac{1}{\sqrt{1+\left(\frac{f}{f_c}\right)^{2N}}} \qquad (1)$$

where N is the order of the filter.

The input specification for this design is summarised in Table 1. The error function is given in Eq. 2.

$$e = \left| H(j\omega) - \hat{H}(j\omega) \right|, \qquad e \in [0, \infty) \qquad (2)$$

The fitness is the sum of absolute deviation between the actual value of the voltage magnitude that is produced by the circuit when compared against the ideal. PSpice

(MicroSim Corp. 1997) was used to evaluate the circuits produced .

Input Specifications	Values
Circuit Synthesis:	
Number of generations	100
Population size	200
Type of selection , size	Tournament, 2
Type of crossover, rate	Singlepoint, 0.8
Type of mutation, rate	See section 2.2, 0.2
Initial chromosome length	3
Value Optimisation:	
Number of generations	5
Population size	10
Type of selection , size	Uniform
Type of crossover, rate	Uniform, 0.75
Mutation rate	0.1
Initial chromosome length	Varies with circuit
Circuit Specifications:	
Maximum number of components	5
Minimum number of components	3
Component series	E-12
Capacitance in F (min, max)	1e-9, 100e-6
Inductance in H (min, max)	0.1e-6, 0.1
Passband frequency f_p (Hz)	925
Stopband frequency f_s (Hz)	3200
Passband gain K_p (dB)	3.01
Stopband gain K_s (dB)	22

Table 1. Input specifications for filter 1.

The best and average objectives for both cases were plotted as shown in Figure 7. We can see that when a starting point was given to the GA, the population tends to converge faster than in the first case. This implies that less time and computer resources needs to be spent on the search. It was estimated that the time taken for the design task was sped up by approximately 30% when a starting point was provided. Also, it can be seen that the results of the second case actually outperforms that of the first; both in terms of its best and average objectives.

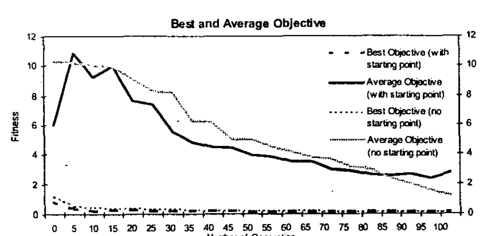

Figure 7. Best and Average Objective for filter 1.

Figure 8 shows the percentage of faulty circuits generated over 100 generations. The amount of faulty circuits was also reduced when a starting point was given. From this, we can conclude that when a suitable circuit is known prior to the search process, it should always be given to the GA as a starting search point.

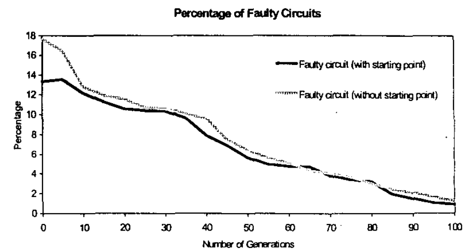

Figure 8. Percentage of faulty circuits generated.

Figure 9 shows the starting circuit given to the GA. In Figure 10, the frequency response of the evolved circuit was compared with the ideal response of the 3 rd order butterworth response filter. It can be seen that both the frequency responses are in good agreement with each other. The best circuit is given in Figure 11.

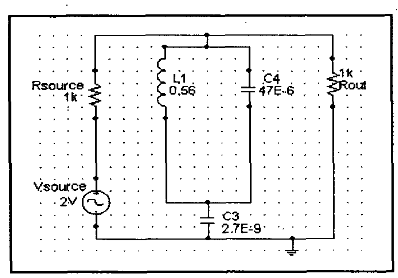

Figure 9. Starting circuit for filter 1.

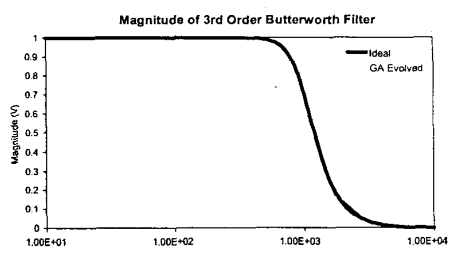

Figure 10. Frequency response of filter 1.

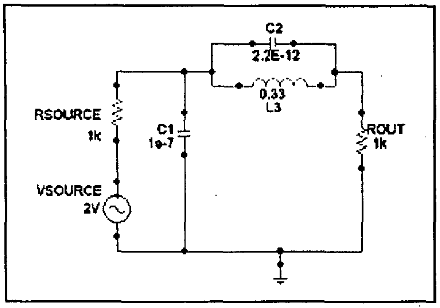

Figure 11. Best circuit for filter 1.

4.2 Filter 2: Elliptic Filter

This design task involves designing a filter that has similar specifications to a 5 th order elliptic filter presented in (Koza *et. al.* 1996). As compared with our earlier attempt, this proves to be a more challenging design task for the GA because of the stringent specifications imposed (Table 2).

Input Specifications	Values
Circuit synthesis:	
Number of generations	100
Population size	200
Type of selection , size	Tournament, 2
Type of crossover, rate	Singlepoint, 0.75
Type of mutation, rate	See section 2.2, 0.2
Initial chromosome length	12
Value Optimisation:	
Number of generations	10
Population size	20
Type of selection , size	Uniform
Type of crossover, rate	Uniform, 0.75
Mutation rate	0.1
Initial chromosome length	Varies with circuit
Circuit specifications:	
Maximum number of components	18
Minimum number of components	12
Component series	E-12
Capacitance in F (min, max)	1e-9, 100e-6
Inductance in H (min, max)	0.1e-6, 0.1
Passband frequency f_p (Hz)	1000
Stopband frequency f_s (Hz)	2000
Passband gain K_p (dB)	0.01
Stopband gain K_s (dB)	63.5

Table 2. Input specifications for filter 2.

PSpice was called to perform an AC small signal analysis on each circuit over 101 frequency points ranging from 1Hz to 100KHz on a logarithmic scale. A don't care band between 1KHz to 2KHz disregard any deivation from the target response. The fitness meaure shown in Eq. 3 does not penalise voltages that are within 1% of the target voltages; a penalty factor of 10 is applied to every unacceptable deivation from the target response.

$$e = \left| H(j\omega) - \hat{H}(j\omega) \right|, \qquad e \in [0, \infty)$$
$$f = \sum_{\omega} e \cdot w, \qquad w = \begin{cases} 1, & e \le 0.01 \\ 10, & e > 0.01 \end{cases} \qquad (3)$$

where w is the penalty factor.

The best and average objectives were plotted as shown in Figure 12.

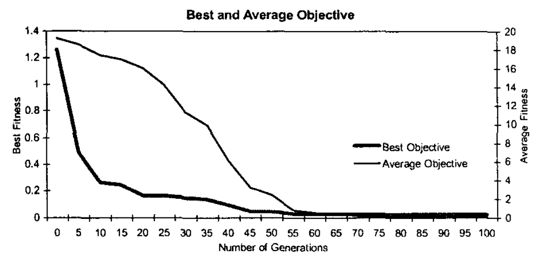

Figure 12. Best and average fitness of filter 2.

Figure 13 shows the percentage of faulty circuits for this design task. It can be seen that the percentage of faulty circuits produced decreases as the population converges. By approximately the last twenty generations, most of the circuits produced by the GA are relisable circuits, apart from a few trivial cases. This not only proves that our GA is effective, it also has time and cost implications.

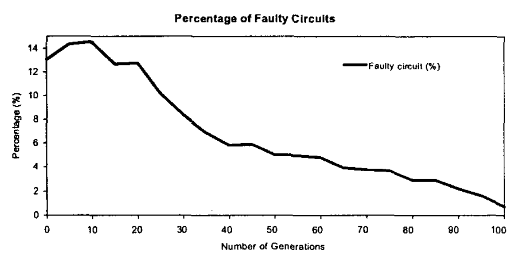

Figure 13. Percentage of faulty circuits for filter 2.

Further investigations were conducted to demonstrate the advantage of using the E-12 series component values as compared with unrestricted values in value optimisation. In the first experiment, unrestricted values were used for value optimisation, these values were then rounded off to the nearest value in the permitted set. In the second experiment, values from the E-12 series were used to perform value optimisation. A $\pm$5% component tolerance has been added to these values to make the design more accurate. The component values obtained from these experiments were tabulated in Table 3. A comparison of the results obtained from the 2 experiments were plotted in Figure 14a and Figure 14b.

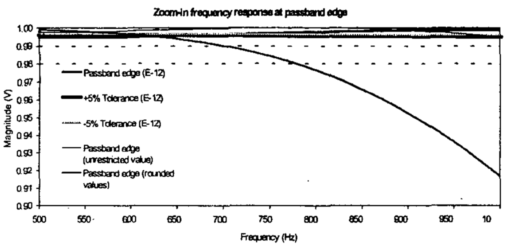

Figure 14a. Zoom in frequency response at passband edge.

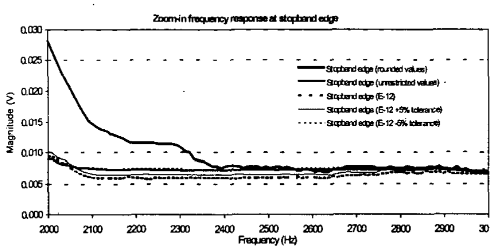

Figure 14b. Zoom in frequency response at stopband edge.

From Figure 14, it can be seen that when unrestricted values were used, the response seem to be within the target specifcations. However, when these values were rounded off to real values, there was a 121.36% increase in the error (0.04644 to 0.1028) and the response no longer falls within the specifications. On the other hand, when the E-12 series components were used for value optimisation, we acheieved a circuit that not only meets the specifications, but is also realisable as a practical circuit.

Component Values	Unrestricted Values	Rounded - Off Values	E-12 Series (+ 5%)
C1	4.7E-8	7.2E-9	6.8E-9
L2	15.0E-3	1.3E-4	1.2E-4
L3	68.0E-3	4.1E-5	3.9E-5
C4	1.0E-7	5.5E-8	5.6E-8
L5	82.0E-3	107E-3	10E-2
L6	5.6E-3	174E-3	180E-3
C8	1.0E-8	4.4E-8	4.7E-8
C9	1.0E-9	7.6E-8	8.2E-8
C10	1.8E-7	1.1E-7	1.2E-7
L11	82.0E-4	55.6E-3	5.6E-3
C13	5.6E-8	11.9E-8	1.2E-7
C15	1.2E-8	1.0E-9	1.0E-9

Table 3. Types of component values used for value optimisation.

A close examination on Figure 15 reveals that the circuit is of a novel topology. Although part of the circuit resembles the ladder structure, its overall structure is unique.

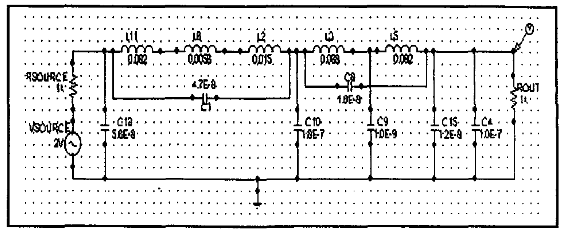

Figure 15. Best circuit for filter 2.

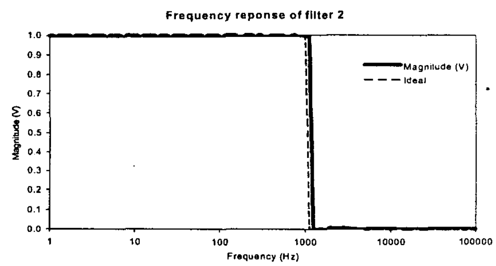

Figure 16. Frequency response of filter 2.

From figure 16, we can see that the frequency response falls within the target specifications. The time taken for the butterworth and the elliptic filter were 6 hours and 10 hours respectively. The designs were perform using a Pentium 400MHz PC. Figure 17 provides the user with an estimate of the average time taken for designing circuits containing a maximum of 10 and 20 nodes. It can be seen that the evaluation time of PSpice is almost 88% of the total design time for a 10-node circuit and approximately 92% when the circuit consists of 20 nodes. Since a large proportion of the total design task is used to evaluate circuits' performances, a parallel or distributed GA can be implemented to speed up the design process; where several circuits can be evaluated simultaneously.

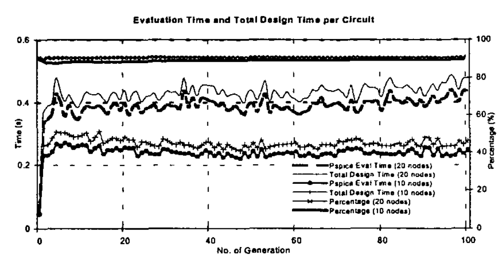

Figure 17. Evaluation and total design time taken.

5. Conclusion

In this paper, we have developed a linear circuit representation and a GA-based technique to automatically design circuits and, in particular, analogue filters. The results obtained have shown that our GA is both effective and robust especially when reliability of circuits and cost implications are of concern. We have also demonstrated the fact that when *a-prior* knowledge of a suitable circuit is known, it should be used as a starting point for the search. This not only helps to produce circuits with better objectives, but also reduced the number of faulty circuits produced as well as the time taken for the whole design process. In terms of component value optimisation, we have shown that circuits produced using values from a permitted set are not only reliable, but are also less time consuming because the ranges specified for the search are often pre-defined. In addition, the cost of building the circuit is lower since components are readily available. One of the future improvements to this work is to take into considerations the numerous specifications involved when designing an analogue circuit, e.g. sensitivity, signal-to-noise ratio, size, parasitic effects, etc. In order to satisfy these objectives, multi-objective optimisation techniques will be used. With the results obtained from our GA, it should be possible to extend this work to design and discover circuits whose performance will be on par with or better than those manually design by experience engineers, relieving them of this tedious task.

Acknowledgements

The authors would like to acknowledge and thank Mr Lipton Chan for the use of his Java-based GA library and for all the enthusiastic discussions held.

References

- Darlington S. (1999) "A History of Network Synthesis and Filter Theory for Circuits Composed of Resistors, Inductors, and Capacitors," IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications vol. 46 no. 1 pp.4 – 13
- Drechsler R. (1998) "Evolutionary Algorithms for VLSI CAD," Kluwer Academic Publishers
- Goh C. and Li Y. (1999) "Advanced Graph Theory and Genetic Algorithm Based Symbolic Circuit Design Automation," Proceedings of the Second Asia-Pacific Conference on Genetic Algorithms and Applications pp. 7 – 15
- Goldberg D.E. (1989) "Genetic Algorithms in Search, Optimisation and Machine Learning," Addison Wesley Publishing Company, Inc.
- GrimbleBy J.B. (1995) "Automatic Analogue Networks Synthesis Using Genetic Algorithms," Proceedings of First IEE/IEEE International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '95) pp. 53 - 58
- GrimbleBy J.B. (1997) "Automatic Synthesis of Active Electronic Networks Using Genetic Algorithms," Proceedings of Second IEE/IEEE International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '97) pp. 103 - 107
- Huelsman L.P. (1993) "Active and Passive Analogue Filter Design," New Work: McGraw Hill
- Koza J.R., Bennett III F.H., Andre D., Keane A.M. (1996) "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming," Artificial Intelligence in Design pp. 151 – 170, Kluwer Academic Publishers.
- Koza J.R., Bennett III F.H., Andre D., Keane A.M. (1996) "Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming," IEEE Transactions on Evolutionary Computation vol. 1 no.2 pp. 109 - 128
- Lam Y. and Zwolinski M. (1997) "Analogue Circuit Synthesis From Performance Specifications," IEE Colloquium (Digest) no. 331 pp. 4/1 - 4/6
- Li Y. (1996) "Neural and Evolutionary Computing IV." Lecture notes – Department of Electronics Electrical Engineering, University of Glasgow
- MicroSim Corporation (1997) "MicroSim PSpice & Basics – Circuit Analysis Software," version 8.0
- Valkenburg Van M.E. (1982) "Analogue Filter Design." Holt-Saunders International Editions