

Multi-objective Optimization Tool for A Free Structure Analog Circuits Design Using Genetic Algorithms and Incorporating Parasitics

Yaser M. A. Khalifa

Electrical and Computer Department
State University of New York
New Paltz, NY 12561 USA
257-3764, 845

khalifay@newpaltz.edu

Badar K. Khan

Electrical and Computer Department
State University of New York
New Paltz, NY 12561 USA
338-4705, 845

Khan38@gmail.com

Faisal Taha

Electrical and Computer Department
State University of New York
New Paltz, NY 12561 USA
546-6856, 845

taha54@newpaltz.edu

ABSTRACT

This paper presents a novel approach for a free structure analog circuit design using Genetic Algorithms (GA). A major problem in a free structure circuit is its sensitivity calculations as a polynomial approximation for the design is not available. A further problem is the effect of parasitic elements on the resulting circuit's performance. In a single design stage, circuits are produced that satisfy a specific frequency response specifications using circuit structures that are unrestricted and with component values that are chosen from a set of preferred values including their parasitic effects. The sensitivity to component variations for the resulting designs is performed using a novel technique and is incorporated in the fitness evaluation function. The extra degrees of freedom resulting from unbounded circuit structures create a huge search space. The application chosen is a LC all pass ladder filter circuit design.

Categories and Subject Descriptors

B.8.2 [Performance Analysis and Design Aids]

General Terms

Design, Performance.

Keywords

Optimization, Analog Circuits, Genetic Algorithms.

1. INTRODUCTION

Computer-aided design tools for analog circuits consist almost invariably of circuit analysis or simulation packages; no general-purpose analogue circuit synthesis tools are available. Conventional numerical optimization techniques can be used to obtain combinations of component values that satisfy a given design specifications. In addition, conventional optimization techniques operate on circuits after having their desired responses

approximated by a suitable polynomial, using ideal components with unrestricted values, and only operating on pre-defined circuit structures. The restrictions introduced by pre-defining the circuit structure wastes extra degrees of freedom in the design process and can conceal many novel design structures that can satisfy the target response and at the same time enhance the design process.

In previous papers, [1-5] GA is applied to fixed structure circuit designs to satisfy certain specifications. As mentioned earlier, the predetermination of the circuit structure required in conventional design techniques is a limiting factor when considering a powerful search technique such as GA.

A few attempts have been made to include structure synthesis in genetically based optimization techniques such as GA [6,7] and Genetic Programming (GP) [8]. However these techniques suffered huge delays and memory usage and needed to be used in conjunction with circuit analysis packages such as PSPICE [9, 10].

There is still a need, however, for more efficient algorithms that are faster, self-sufficient and can function independently. An efficient measure of performance based on the sensitivity for component variations for the resulting circuits and the incorporation of the parasitic effects associated with components is still needed. This is particularly important in the development of new generations of Analog Programmable Analog Arrays (APAA) Programming Tools. In this paper an efficient GA technique for free structure analog circuit design is described. The technique incorporates a sensitivity measure to compare between potential solutions and incorporates parasitic effects.

2. Genetic Algorithms

The field of Genetic Algorithms [11-13] was founded by John Holland in the early 1970's. In [12] Holland emphasizes the ability of simple representations to encode complicated structures and the power of simple transformations to improve that structure. These representations are combined in what is called in biology a *chromosome*. A number of these chromosomes will constitute a *population*. Syntactic operations are then used to alter and improve these coded solutions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'07, July 7-11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-698-1/07/0007...\$5.00.

3. Implementation

3.1. Chromosome adaptation and structure

In [1-3] the predetermination of the circuit structures has helped in reducing the chromosome length. This was achieved by incorporating the order of the chromosome genes into the node connections information. However, in the case of free structure circuit synthesis this technique cannot be used.

In the case of free structure design, the solution space is much larger and the optimization process is composite. It involves the optimization of two different but highly correlated problems; the structure as well as the sizing of a circuit. Hence, all specifications of the electrical circuit including the structure and the sizing of all the components should be included in the chromosome representation.

The specifications of the electrical circuit include the number of components in the circuit, the type of each component and a list of connections between the components. Circuit components can include a variety of different types of components, including resistors, capacitors, inductors, etc...

Given the above information, the chromosome for each circuit comprises a number M of groups of equal bit lengths, and an extra group of bits representing the number of nodes in the circuit, Figure 1.

Each gene in the chromosome specifies a component and contains four fields (i) Type of component (L, C, etc.), (ii) A pointer to a menu containing component value, (iii) and (iv) circuit nodes to which the component is connected, see Figure 1.

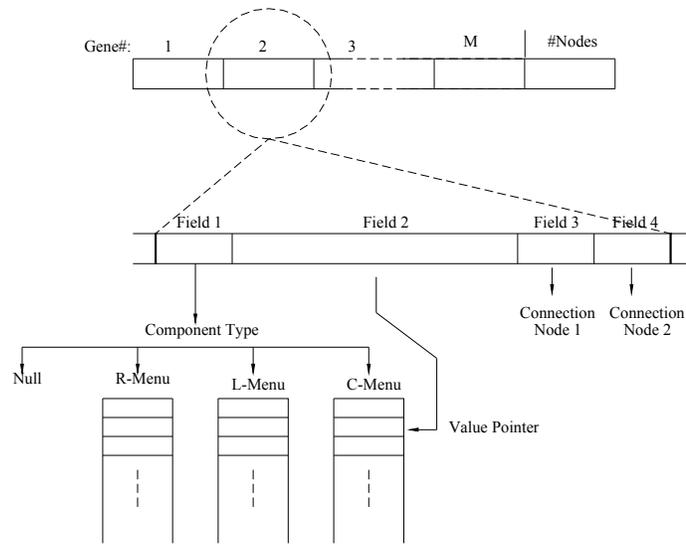


Fig. 1. Chromosome structure.

It is generally desirable for a low cost design to use the least number of components. For this reason a Null component, representing an open circuit, is introduced into the component type menu.

3.2. The design program

On entry the user is required to supply the following input data:

Type of filter (Low pass, High pass, Band pass)

- ω_1, α_{min}
- ω_2, α_{max}
- N, the maximum number of nodes. Optional, and can be used to limit the search to a certain number of designs.
- M, the maximum number of components (optional).

```

FOR population_member = 0 to population_size, DO
    Determine N, number of nodes
    FOR gene_location = 1 to M, DO
        Decode fields 1,2,3 and 4
        Descale fields 3 and 4 according to N
        Update NAM
    END FOR
    FOR fequency_test_point = 0 to maximum, DO
        LU decompose NAM
        Calculate response
        Compare with template and find error
    END FOR
END FOR
    
```

Fig. 2 Pseudo code for a single generation evaluation

The execution of the program starts by calculating N, maximum number of nodes. Then genes representing elements are then decoded one at a time, splitting them into four fields as explained in section 3.1.

Upon determining type, value and connection nodes, the component value is inserted in the corresponding Nodal Admittance Matrix (NAM) location [14]. When all elements are decoded, contents of the resulting entries in NAM are passed to the calculation function, in which Lower-Upper (LU) decomposition and solutions to the matrix equations are solved.

4. Least Squared Error Evaluation

The error measured is defined as the difference between the specified response constraint, $H_{Upp}(\omega)$ and $H_{Low}(\omega)$, denoted by the shading in Fig. 3, where $H_{Upp}(\omega)$ is the upper boundary of the frequency response, and $H_{Low}(\omega)$ is the lower boundary, and the actual response, $H_a(\omega)$, from the design represented by GA chromosomes. The actual frequency response, $H_a(\omega)$, will

hence be prescribed by its unique combination of component values. Therefore,

$$H(\omega, R_1, R_2, \dots, R_m, C_1, C_2, \dots, C_n, L_1, L_2, \dots, L_v) \quad (9)$$

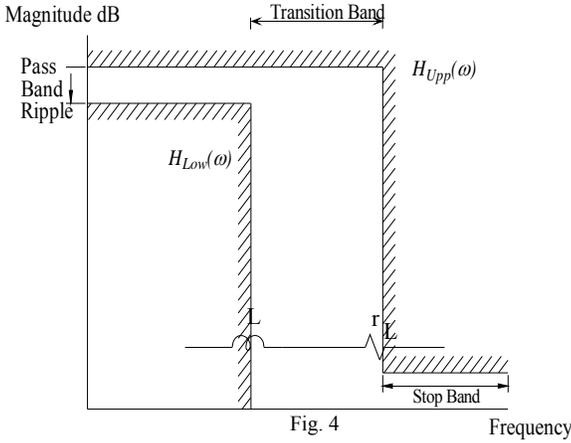


Fig. 3 Design Template

For simplicity all types of components will be declared as k . Thus H becomes

$$H(\omega, k_1, k_2, \dots, k_q) \quad (10)$$

where $q = m + n + v$

Hence the error function will depend on the region it is applied for. For the pass band the error function would be

$$err(\omega, k_1, k_2, \dots, k_q) = H_{Low}(\omega) - H_a(\omega, k_1, k_2, \dots, k_q) \quad (11)$$

while in the stop band the error function would be

$$err(\omega, k_1, k_2, \dots, k_q) = H_{Upp}(\omega) - H_a(\omega, k_1, k_2, \dots, k_q) \quad (12)$$

The Least-Squares error criteria E is used as a performance index because of mathematical simplicity and ease of programming. The error is defined as

$$E = \int_l^u err(\omega, k_1, k_2, \dots, k_q) \quad (13)$$

Substituting from Eq. (13) in Eq. (12), LSE would be

$$LSE = \int_{\omega_l}^{\omega_u} \{H(\omega) - H_a(\omega, k_1, k_2, \dots, k_q)\}^2 \quad (14)$$

where $H(\omega)$ is the boundary response $H_{Upp}(\omega)$ or $H_{Low}(\omega)$. Since the calculations must be performed numerically, the integration in Eq. (14) will be replaced by summation. Therefore,

$$LSE = \sum_{i=1}^m [H(\omega) - H_a(\omega, k_1, k_2, \dots, k_q)]^2 \quad (15)$$

where m is equally spaced frequency points assumed over the range of integration. This is chosen because of ease in programming.

4.1. Specified weighted errors

Eq. (15) gives equal weights for all frequency points within the range. This does not necessarily reflect the prescribed performance, for example, in the case of numerically small errors in the pass band. This numerically small error in the pass band will be more likely to affect the accuracy of a solution than a relatively larger error encountered in the stop band. Hence, a weight function can be inserted in eq. (15) to enable us to emphasize or de-emphasize the error in the various regions of the frequency spectrum^[19]. Therefore eq. (15) can be modified to be

$$LSE = \sum_{i=1}^m \{w(\omega) \cdot [H(\omega) - H_a(\omega, k_1, k_2, \dots, k_q)]\}^2$$

The weight function $w(\omega)$ has been arbitrarily chosen to be two in the pass band region and unity in the stop band region. Any other design weight functions can be easily applied instead.

Although the pass band will be characterized by a specific pass band ripple, the upper boundary of the response represents the maximum power transfer level, the claimed results of LC filter theory. Therefore, testing only the suitability of the lower limit of

results of LC filter theory. Therefore, testing only the suitability of the lower limit of the response would be sufficient, as the upper limit will never be exceeded in the pass band.

In the stop band region, only the upper limit of the specified attenuation level need to be considered, since any lower value of attenuation meets the specifications. The transition region is considered as a 'don't care' region. Thus it is sufficient to test just the pass band and the stop band for confirmations. The different frequency band boundaries are shown in Fig 3.

4.2 Performance improvements by controlling connection fields

The number of nodes comprising the circuit is determined at the beginning of the decoding process, to control the connection node numbers of the circuit components.

The fields determining the connection nodes are decoded in a way to produce node numbers within N. This is achieved by scaling down the whole range of values of allowed node numbers to only that of N. This helps to avoid circuit structures that would have components connected to nodes that do not exist in the defined structure and which are hence unacceptable, and would be a total waste of time to evaluate.

5. Sensitivity Comparison

In [5], the author has developed a rapid feasible region estimation technique using GA. The technique is modified to provide a measure of comparison among the resulting circuit designs using the free structure GA technique described in previous sections.

A hierarchical approach for circuit design using GA is implemented. Once convergence occurs and a family of solutions emerge, usually 4-6 in average, the structures of these circuits are fixed. For this stage a uniform closely spaced distribution of component values is used to help determine estimates for near and far limits of component values within which the required circuit specifications are met. At the end of this stage, the least minimum x_{\min}^i and largest maximum x_{\max}^i boundaries for each of the parameters are taken to be the estimate of the feasible region for that specific parameter. A measure for the transfer function sensitivity to variations in component x^i is:

$$\Delta x^i = \frac{x_{\max}^i - x_{\min}^i}{x^i}$$

And for m the total number of components in the design:

$$S^P = \sum_{i=1}^m \Delta x^i$$

The new fitness function will result from multiplying S^P by its current value.

6. Extension to Include Parasitic Effects

Parasitics can include wire and winding resistances and inductors in lumped circuit components as well inter-layer capacitances and node to ground voltages. It can also include resistances associated with polysilicon and doped semiconductor regions when used as conductors, and capacitances associated with any crossover, from any conductor to substrate, and with any depletion region in a reverse-biased pn junction, in integrated circuits. Careful layout design can eliminate some of these parasitics, but avoiding them all together is not feasible. Other factors which can also affect the amount of parasitics are the area of the circuit and the packaging density of the chip [15].

Parasitics can cause significant deviations in circuit performance and should be accounted for in early stages of the design, if their effects are to be minimized. In this chapter GAs, as described earlier, are modified to include some additional information the designer might want to consider for each circuit component.

Parasitic effects considered in this paper include inductor resistance as shown in Fig. 4, and series and parallel resistances and series inductance associated with a capacitor as shown in Fig. 4.

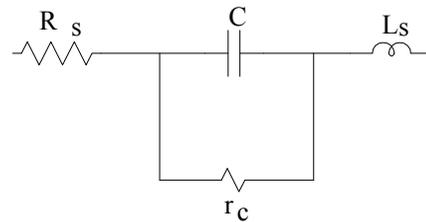


Fig. 4

To implement the parasitic effects into the circuit design optimization problem using the GA, an additional parameter is added into the component menu and this represents the parasitic effect of the associated component and the quality factor Q, Fig. 5 on the next page.

This Method will avoid adding to the complexity of the structure, and will make it possible to add or remove any effect from the component menu without having to alter the chromosome structure.

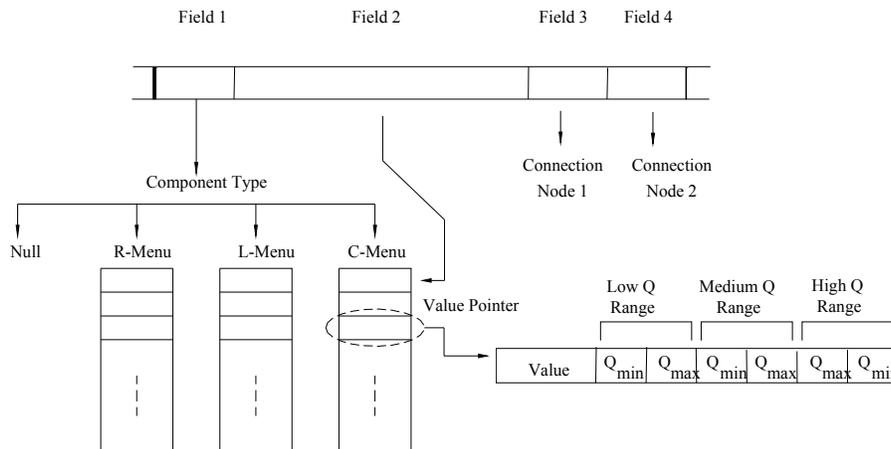


Fig. 5

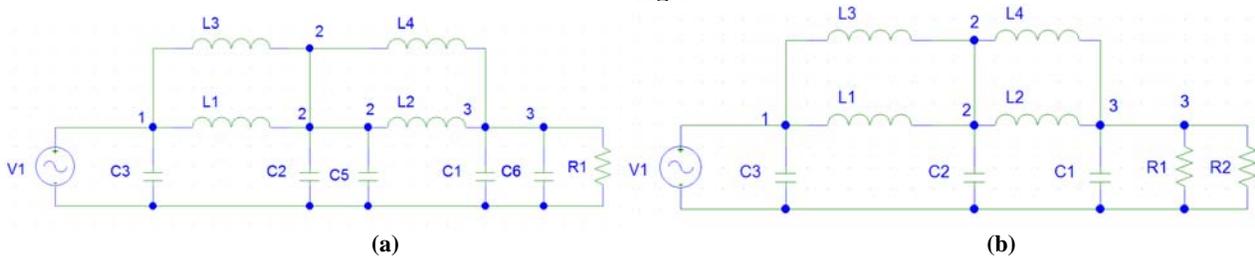


Fig. 6 (a) and (b) filter circuit with 1dB pass band ripple and attenuation of -100 dB at the stop band edge used to test GA program.

7. Results

In this implementation the desired design template is the starting point of the design process, whereas in conventional designs, polynomial approximation for the desired response is used [15]. This has the limitation of constraining the feasible solution to sets of design parameters that would satisfy the polynomial approximation. By designing directly from the template specifications, a designer would avoid that extra degree of approximation a polynomial fitting would introduce. At the same time it provides an extra degree of freedom which would enlarge the solution space considerably to include any design that would reside within the design template without necessarily being represented by a polynomial. Also the GA search starts from an arbitrary point in the solution space and need not to have any prior knowledge of the feasible region of the required design.

To make sure that the GA program works as planned, two circuits were designed in a fixed structure and used in the program. The criteria that the two circuits needed to meet was to achieve a maximum attenuation of 1 dB in the Pass Band and a minimum attenuation of -100 dB in the Stop Band. Fig 6 shows the two circuits used to test the GA program.

As mentioned previously, two filter circuits perform the same job and have a close topology with almost the same components' values. The GA program will then take these two topologies and search the large solution spaces for the best component ranges.

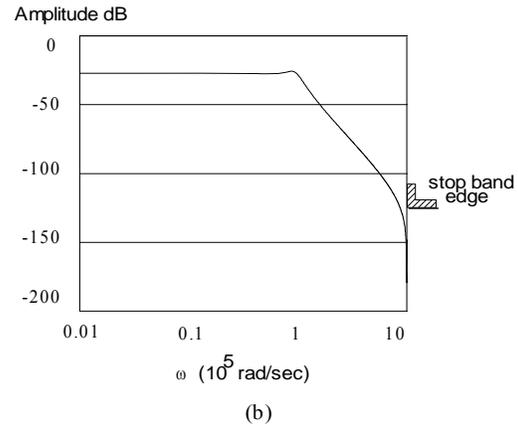
Table 1: (a) Table shows Components, Nodes, ranges, and actual values for each component in circuit (a), (b) Table shows Components, Nodes, ranges, and actual values for each component in circuit (b)

Circuit 1				Range	
Component	Actual Value	Nodes 1	Nodes 2	Low	High
C1	1.80	3	3	1.28	2.28
C2	2.20	2	2	2.2	2.72
C3	2.20	1	1	2.2	2.72
L1	2.20	1	2	1.68	2.65
L4	1.00	2	3	2.2	2.72
L3	1.00	2	1	0.5	1.09
C5	2.70	2	2	2.7	3.22
R1	1.00	3	3	0.48	1.4
L2	2.20	3	2	1	1.52
C6	1.00	3	3	0.48	1.48

(a)

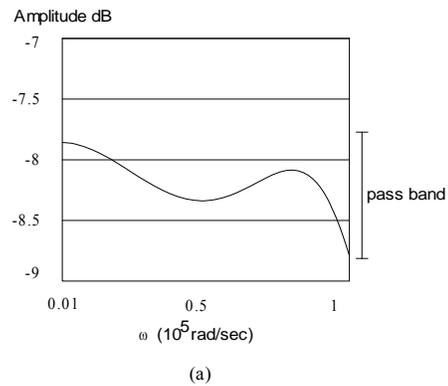
(b)

Circuit 2				Range	
Component	Actual Value	Nodes 1	Nodes 2	Low	High
L4	1.00	3	2	2.2	2.27
R1	0.10	3	3	0.1	0.62
C2	1.80	2	2	1.8	2.32
C1	1.20	3	3	0.77	1.57
C3	2.20	1	1	1.68	2.44
L2	2.20	2	3	0.57	1.36
L1	4.70	1	2	4.7	5.22
R2	0.15	3	3	0.15	0.67
L3	1.00	1	2	0.62	1.23



Even though the two circuits have a very close components values, the range generated by the GA program were different. The differences in ranges for each component in the circuits are shown in Table 2. We also note that C2 and C5 in circuit (a) are in parallel, which is equivalent to C2 in circuit (b).

As shown in Table 2, the two circuits varies with the difference at some components, but in general, Circuit (a) has a higher distance between its component values, meaning that the it allows more component values than Circuit(b). this means that Circuit (a) is more stable because the if any of its component changed by a specific factor for any reason, it will still meet the filter criteria which is 1 dB in the Pass Band and -100 dB in the Stop Band where as Circuit (b) could go out of these specifications because it has a smaller distance between the different components.



7.1 Low pass design

Different low pass design specifications were considered. One had a pass band ripple of 1 dB, minimum attenuation of

Fig. 7 Frequency response examples for low pass design template with high quality factor (a) pass-band and (b) stop band

Table. 2 Differences between circuits' components.

Circuit b				Circuit a			
Component	Range		Difference	Component	Range		Difference
L4	2.20	2.27	0.07	L4	2.20	2.72	0.52
R1 R2	0.06	0.32	0.26	R1	0.48	1.40	0.92
C2	1.8	2.32	0.52	C2 C5	1.20	1.47	0.27
C1	0.77	1.57	0.80	C1 C6	0.35	0.89	0.54
C3	1.68	2.44	0.76	C3	2.20	2.72	0.52
L2	0.57	1.36	0.79	L2	1.00	1.52	0.52
L1	4.70	5.22	0.52	L1	1.68	2.65	0.97
L3	0.62	1.23	0.61	L3	0.50	1.09	0.59
Average Difference			0.54	Average Difference			0.61

-100 dB at a transition ration of 10. An average group of five solutions emerged, the frequency response of one of these solutions is plotted in Fig 7. The circuit constructed by the GA is shown in Fig 8. It is noticed that well known Ω and T ladder structures have also been generated by the GA. A different design specification of 0.5 dB pass band ripple and -60 dB stop band attenuation was tested.

Figure 9 shows the resulting circuit design.

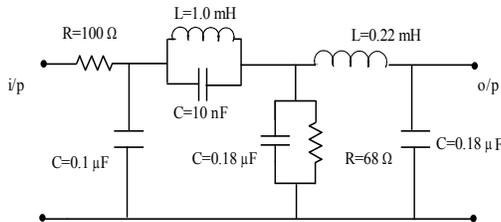


Fig. 8 Genetically designed circuit with a pass band ripple of 1 dB and stop band edge at -100 dB.

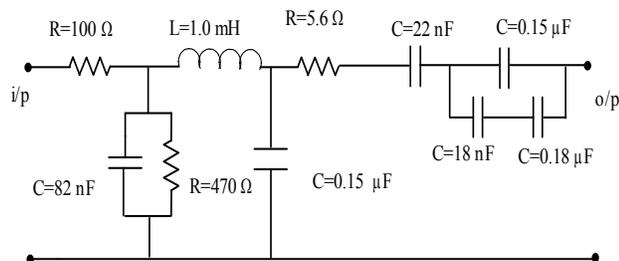
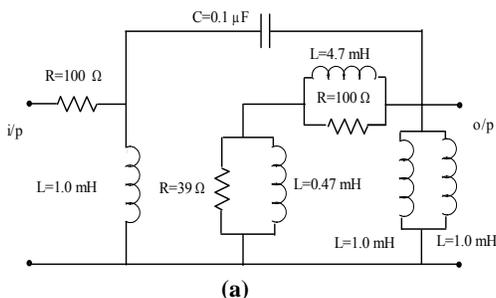


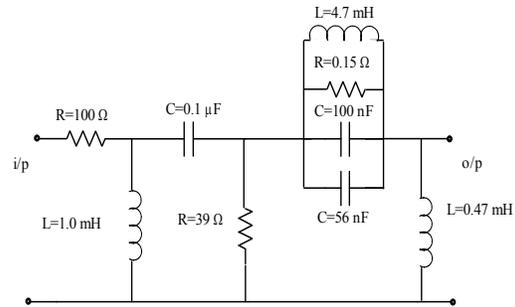
Fig. 9 Genetically designed filter circuit with 0.5 dB pass band ripple and -60 dB stop band attenuation

7.2 High pass design

The high pass filter design is done directly without having to use a low pass prototype and then perform frequency transformations. A design template of 1 dB pass band ripple and attenuation of -60 dB at the stop band edge with a transition ration of 10 is considered. A number of four solutions on average emerged two of these circuits are shown in Fig 10. The frequency responses of the resulting circuits are shown in Fig 11.

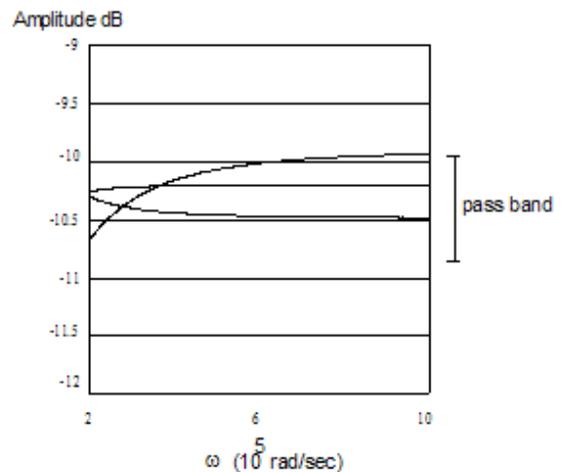


(a)

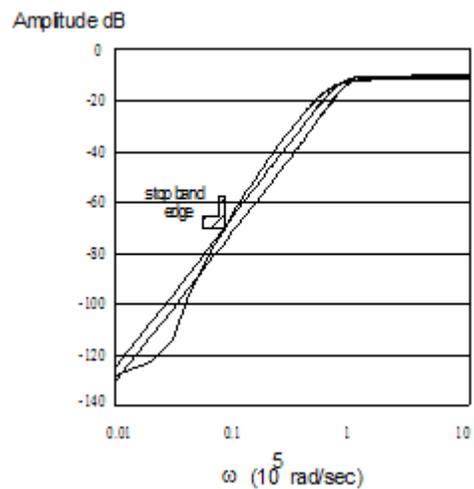


(b)

Fig. 10 (a) and (b) Genetically derived high pass filter circuit with 1dB pass band ripple and attenuation of -60 dB at the stop band edge



(a)



(b)

Fig.11. Frequency response examples for High pass design template with high quality factor (a) pass-band and (b) stop band

8. Conclusion

This paper has presented a novel circuit design tool by using the GA strong search abilities in both the structure and sizing domain of a circuit design problem. A successful design of the chromosome structure has been obtained to include the full circuit specifications. A novel technique to compare circuit sensitivities to change in component values among different circuit structures was developed.

It is noticed that well known Ω and T ladder structures have also been generated by the GA. That is due to the low sensitivity such structures have, to small changes in component values. This feature was detected by the GA as a result of the sensitivity comparison technique developed in this paper.

9. References

1. S. Zhao, L. Jiao, J. Zhao, and Y. Wang, "Evolutionary Design of Analog Circuits with a Uniform-design Based Multi-objective Adaptive Genetic Algorithm.", Proc. of NASA/DoD Conference on Evolvable Hardware, pp 26-29, Chicago, IL, 2005.
2. D.H. Horrocks and Y.M.A. Khalifa,, "Genetically Evolved FDNR and Leap-Frog Filters using Preferred Components Values", Proc. European Conference on Circuit Theory and Design, Istanbul, Turkey, pp 359-362, Aug. 1995.
3. C. Goh and Y. Li, "GA Automated Design and Synthesis of Analog Circuits with Practical Constrains", Proc. of the 2001 Congress on Evolutionary Computation CEC'01, pages 170-177, IEEE Press, 2001.
4. H. Muhlenbein, L. Zinchenko, V. Kureichik, and T. Mahnig, "Effective Mutation Rate for Probabilistic Models in Evolutionary Analog Circuit Design", Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02), Divnomorskoe, RUSSIA. 2002.
5. S. Ando, M. Ishizuka, and H. Iba, "Evolving analog circuits by variable length chromosomes", Proc. of the 2000 Congress on Evolutionary Computation CEC'00, pages 994-1001, IEEE Press, 2000.
6. Y.M.A. Khalifa and D.H. Horrocks, "Isomorphism Elimination for the Enhancement of Genetically Generated Analog Circuits", Proc. of The International Symposium on Circuits and Systems ISCAS'99, Orlando, Florida. USA, pp 314-317, June 1999.
7. J.B. Grimbleby, "Automatic Synthesis of Active Electronic Networks using Algorithms", IEE Genetic Algorithms in Engineering Systems: Innovations and Applications, Conference Publication No. 446, PP103-107, Sep. 1997.
8. J.R. Koza, D. Andre, F.H. Bennet III, and M.A. Keane, "Toward evolution of electronic animals using Genetic Programming", Proceeding of the first Annual Conference, Stanford University, Cambridge, July 28-31, 1995.
9. Fenical L., *PSPICE*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1992.
10. W. Blume, "Computer Circuit Simulation", Byte v 11 n 7, pp 165-166, 168-170, Jul 1986.
11. D.E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, Reading MA USA, 1989.
12. J. Holland, *Adaptation in natural and artificial systems*, University of Michigan, Ann Harbor, 1975.
13. L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, pp 36-38, 1991.
14. L. O. Chua, *Computer-Aided Analysis of Electronic Circuits*, Prentice-hall, 1975.
15. G.C. Times and J.W. LaPatra, *Circuit Synthesis and Design*, McGraw-Hill, New York, 1977.