Chua's Circuit Topology Evolution Using Genetic Algorithm

Ádám Rák[a], Gaurav Gandhi[b], György Cserey[a,c]

[a]*Faculty of Information Technology, Pázmány Péter Catholic University, Budapest, Práter u. 50/a. 1083, Hungary, e-mail: rakad@itk.ppke.hu*
[b]*Cadence Design Systems India Pvt Ltd. Plot 57A, B & C, NSEZ Noida Special Economic Zone Noida, 201305 India, e-mail: ggandhi@cadence.com*
[c]*Infobionic and Neurobiological Plasticity Research Group, Hungarian Academy of Sciences, Pázmány Péter Catholic University, Semmelweis University, Práter u. 50/a, Budapest H-1083, Hungary, e-mail: cserey@itk.ppke.hu*

Abstract

In this paper, a novel approach of using genetic algorithm towards realization of Chua's circuit is presented. The paper demonstrates the use of genetic algorithm to evolve Chua's circuit from a set of given passive and active components. More than a dozen of the evolved designs are demonstrated to work satisfactorily in simulation. The evolved designs are found to be human competitive and are also demonstrated in to work in a laboratory set up.

Key words: chaotic circuit, circuit design automation, genetic algorithm, Chua's circuit

1. Introduction

Chaos theory describes the behavior of certain dynamical systems (systems whose state evolves in time) that may exhibit dynamics that are highly sensitive to initial conditions. Historically seen, Chua's circuit was the first successful physical implementation of a system designed to exhibit chaos [1]. It was the first system proved to be rigorously chaotic [2] and is also the simplest [3] circuit where chaos can be observed experimentally.

Since its inception 24 years ago, there are only a dozen different implementations of Chua's circuit [4, 5, 6, 7] that have been reported in the literature. A closer look at these designs yields that the novelty of these implementations lies in the new implementation of either Chua's diode or the oscillatory part of Chua's circuit. All these designs are derived from the human intuitive way of thinking and hence require a person skilled in the art to develop newer topologies.

To the best of authors' knowledge, software based approaches to evolve Chua's circuit topologies have not been presented in the literature. One reason for this can be that in order to evolve Chua's circuit (or any chaotic system), a rigorous analysis of its time series is required and the lack of necessary and sufficient condition for a system to exhibit chaos might have stopped researchers to attempt the same.

In the current brief, a SPICE based software is proposed, which uses genetic algorithm to evolve newer topologies for Chua's diodes. These evolved diodes can further be used to evolve several newer topologies for Chua's circuits. Note that to the best of the authors' knowledge, this is the first software based approach to evolve Chua's circuit topologies.

This paper is organized as follows. Section 2 presents a brief introduction of Chua's circuit. In Section 3, the details of the algorithm are presented which are used for automate the design of Chua's diodes. This section also discusses the approach to evolve newer Chua's circuit topologies. Some novel evolved circuitries and their simulation results are described in Section 4. Section 5 provides the laboratory measurements of one of the implemented new chaotic Chua's circuit. Discussion on several aspects of this work are summarized in Section 6.

2. Chua's Circuit : An Introduction

With the purpose of designing the simplest autonomous electronic circuit to generate chaotic signals, Chua's Circuit was first proposed in 1984 [1]. The presence of the chaotic attractor in this 3^{rd} order, extremely simple, autonomous circuit was first presented by T. Matsumoto [3], using computer simulations. From the circuit designer's perspective, Chua's Circuit consists of four linear elements (two grounded capacitors, one grounded inductor and one linear resistor) and one non-linear resistor. The dynamics of Chua's Circuit is described by the following set of equations:

$$\begin{array}{l} C_1\frac{dv_1}{dt} = G(v_2 - v_1) - f(v_1) \\ C_2\frac{dv_2}{dt} = G(v_1 - v_2) + i_3 \\ L\frac{di_3}{dt} = -v_2 \end{array} \tag{1}$$

Where v_1, v_2 and i_3 denotes the voltage across capacitor C_1, voltage across C_2 and current across inductor L, respectively, and *f(x)* is a nonlinear function defined by

$$i_R = f(v_R) = G_b v_R + 0.5(G_a - G_b) \cdot (|v_R + B_p| - |v_R - B_p|) \tag{2}$$

This equation denotes the 3-segment odd symmetric voltage current characteristics of a non-linear resistor, also called Chua's diode. Here G_a and G_b are the slopes of the segments and B_p denotes the breakpoint as shown in Figure 1b.

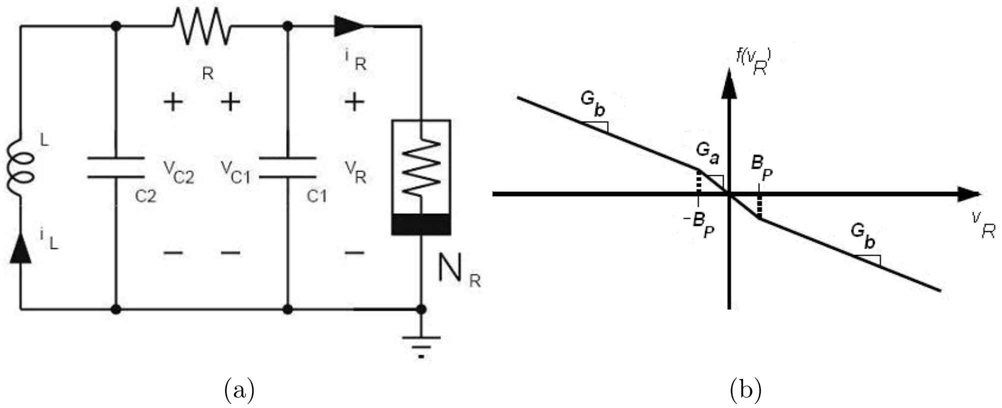


Figure 1: In Figure (a) Chua's circuit can be seen. The V-I characteristics of Chua's diode are depicted in Figure (b).

Figure 1a shows the circuit diagram of Chua's circuit, which is an autonomous dynamical system. For specific parameter values, this circuit [3] was shown to behave chaotically. It was later confirmed experimentally [8, 9] as well as mathematically [2] that indeed Chua's circuit is the simplest autonomous circuit which exhibits chaotic behavior. By adding a linear resistor in series with the inductor, the circuit can generate many more chaotic phenomena. This unfolded Chua's circuit is known as Chua's oscillator and the associated equations are called as canonical Chua's equation [10, 11].

3. Chua's Circuit Topology Evolution

The crucial part of the present work is to develop a genetic algorithm which is efficiently able to generate functioning circuits according to the given parameters. The chosen representation and the size of the population are of primary importance. The evolution time and computational resources required by the algorithm depends heavily on how to implement initialization, crossing, mutation and evaluation methods. Evaluation depends on a well defined fitness function. These details are usually problem-specific and following the recommendations of coursebooks mostly provides slow or non-convergent optimizations. Well chosen details, on the one hand prove deep understanding of the problem and on the other hand witness proficient practice of using genetic algorithm as a tool. Both are needed for a successful implementation in such a hard research area as automated chaotic nonlinear circuit design.

While overviewing the scientific literature, we could not find any appropriate metrics which would be able to decide fast and efficiently if a nonlinear behavior in general is chaotic or not. The approach to the problem as a whole makes it an even more complex problem, but for the special case of Chua's circuit we present a new method to define an appropriate metric.

The solution to this problem needs a different approach, described as follows: the Chua's circuit has three main parts. These are the oscillating circuit, the coupling part (usually a variable resistor) and the nonlinear analog resistor or diode (called Chua's diode) [12]. The nonlinear diode has been shown to be the reason for the diverse and complex behavior of chaotic dynamics in Chua's circuit. Thus evolving Chua's diode and connecting it to the rest of the circuit will help us in evolving novel implementations of Chua's circuit. Further, within an evolved Chua's circuit topology, replacing the oscillating part of the circuit with another circuit showing similar behavior evolves a newer topology for Chua's circuit.

3.1. Individuals

A simulation based approach to evolve Chua's circuit is used and SPICE is used to perform circuit simulations. The input to simulations is called individual, which contains the elements and their connections. This individual was modified in every generation. In order for the algorithm to be less intensive computationally, the algorithm was biased with an initial circuit typically containing a rough topology of the desired evolved design. Furthermore, in the individual, numbers are assigned to the circuitry nodes. The connections of circuit elements are given by these numbers. Thus the circuit is unequivocally described. In the course of the present method, the value of the resistors between nodes is iteratively tuned till the system has a specified behavior. At an early stage of the generation, there can be such nodes in the system to which no elements are connected. These nodes do not disturb the functionality of the electronic circuit rather they provide an opportunity for extension. These resistance values, which were generated in the course of the initializing process or added during generations, are stored in a matrix:

$$A_{i,j} = \begin{cases} 0, & \text{short} \\ -1, & \text{open} \\ a, & \text{resistance value} \end{cases} \tag{3}$$

The resistance matrix defined in this manner is practically the adjacency matrix of a graph, where the weight values between the nodes are equal to the resistance values between the nodes. The matrix is symmetric by definition, its size is $n \times n$, in the experiments n was equal to 20.

3.2. Algorithm

The present algorithm consists of the following steps:

1. Generating the population
2. Selection
3. Crossover
4. Mutation
5. Evaluation

Generating the population: The initialization step of the algorithm is as follows: half of the population is set up with the initial individual, where the parameters of individuals are equal to the values of the initial individual. The other half of the population contains viable individuals generated by mutating the content of the initial individual. Generally at this phase, the number of the active nodes is small in the individuals. 100-200 individuals form the population characteristically.

Selection: In the course of sequencing, the fitness of each individual in the population is calculated and on the basis of this value, they are ordered in a row. The sorted sequence is divided into three equal groups based on fitness values. One member of the population from the third group is randomly selected and deleted.

Crossover: The crossover happens in this manner: from the two resistance matrices of the individuals, elements with $50-50\%$ equivalent chance are chosen into the offsprings. One of the offsprings is randomly deleted. The survivor is mutated with a given probability and then moved to the vacant position. No parents are deleted from the population. Due to the extreme sensitivity of the chaotic circuit's behavior, the usual GA crossover operation is not suitable to this problem. In contrast, the crossover operator we suggest, which introduces minimal changes in the current generation, allows us to preserve the good individuals throughout the evolution.

Mutation: All positive elements are taken in the matrix one at a time and their value are modified with a probability based on a Gauss bell-shaped curve with 0 expected value. If the new value of the resistance is smaller than 1Ω, then the resistance is substituted with a short. The given resistance is exchanged to an open with a probability between 0.01 and 0.2. The elements of the resistance matrix meaning an open are exchanged with a probability between 0.01 and 0.2 to $0 - 200K\Omega$.

Evaluation: The fitness value of the individual is calculated in the first step of the evaluation process. If this value is adequately small (the better an individual is, the higher the value of the fitness is), the algorithm stops, otherwise the algorithm iterates and continues to the **Selection** step. After the entire run, the evolved circuit is found to be the best evolved design for Chua's diode. Evolving a successful topology for Chua's circuit, the number of necessary generations generally is around the order of 10000.

3.3. Fitness function and its evaluation

The V-I characteristics of Chua's diode are as shown in Figure 1b. The aim was to realize electronic circuits which have similar V-I curve. For the evaluation of fitness, initially the V-I curve of the actual circuit is determined with SPICE. Subsequently, the following two approaches are followed to evolve the Chua's diode. In the first approach, the absolute value of the sum of the distances between the derived V-I curve and the ideal curve is taken to calculate the fitness error. In the second approach, this procedure is refined by adding an additional term to the fitness error. This term is derived from calculating the sum of the distances of the derivative function of the V-I curve of the derived circuit and the derivative function of the ideal V-I curve (Figure 2). Herein, the distances between the curve (or derivative curve) and the outstretched region (see Figure 2) of the ideal curve (or the derivative of the ideal curve) are calculated. With appropriate weighting, the system can tolerate a certain DC offset but then preserves the gradient of the curve.

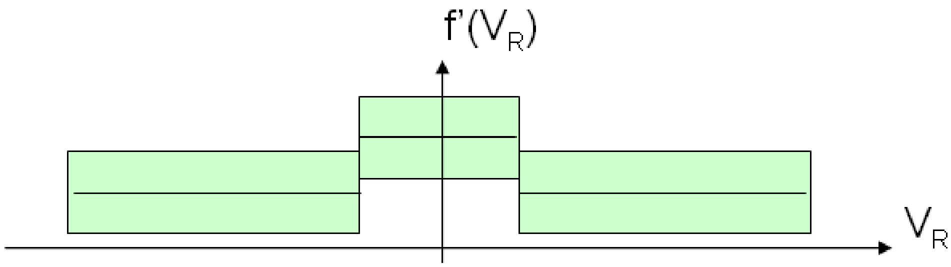


Figure 2: The derivative function of the V-I characteristic function of Chua's diode, where outstretched regions are also depicted.

Cost and fitness functions can be given by the next formulas:

$$Cost = \alpha \sum_{x} f(v(x), z(x)) + \beta \sum_{x} g(w(x), q(x)) \tag{4}$$

$$Fitness = \frac{1}{Cost + 1} \tag{5}$$

where the coordinates of the x-th point-element are $v(x)$ and $z(x)$ on the V-I figures, and $w(x)$ and $q(x)$ on the V-I derivative figures. α and β are weight parameters. The f function defines the distance from the ideal V-I curve bounded by parallel regions, and the g function defines the distance from the derivative function of the ideal V-I curve bounded by parallel regions (see Figure 2). These distances are Euclidean-distances from the borders of the regions. The distance is considered to be zero if a point was inside the bounded region. The curve was divided into 1000-2000 points. Whereas both approaches yielded to some newer topologies of Chua's circuit, the second approach was found to be faster and more accurate than the first one in evolving the circuit.

4. New Chua's circuits

The presented Algorithm has evolved more than a dozen new topologies for Chua's diodes and hence Chua's circuit in less than 24 hours on a single (Intel Xeon HyperThread, 3.4 GHz, 2 Gbyte RAM) machine. The following section presents simulation results of two derived Chua's circuits from the algorithm. Success rate (SR) of the algorithm was 0.63 for 22 independent runs. The average number of evaluations to solution (AES) index was 15236 for 12 independent runs.

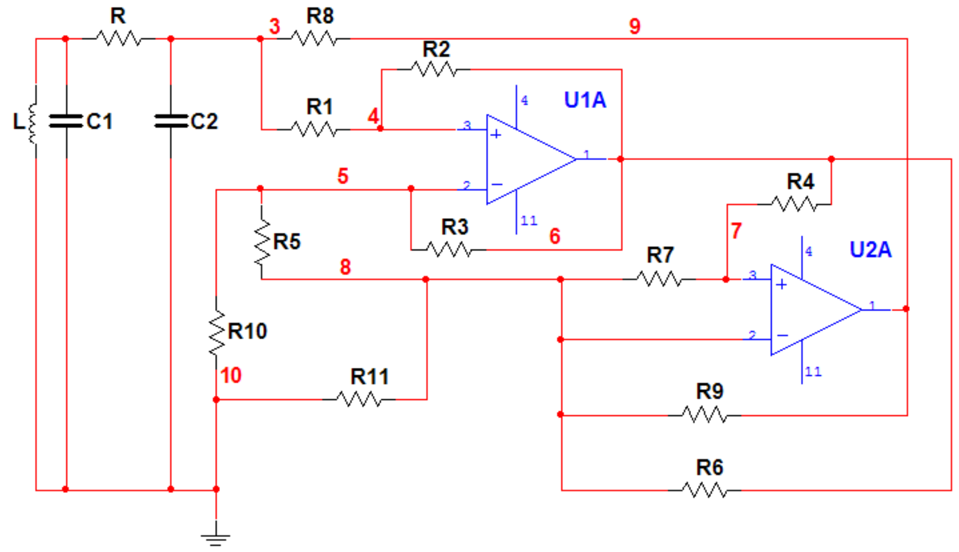


Figure 3: Schematic of the OP400 type operational amplifier based Chua's circuit. (for values see Table 1, third row)

4.1. Operational amplifier based Chua's circuit

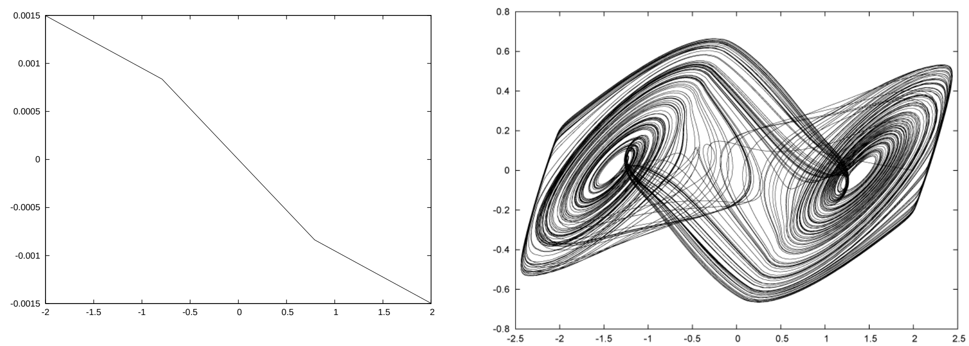
The circuit schematic is shown in Figure 3. The simulation results of the V-I curve is shown in Figure 4a. The chaotic behavior time series of the electronic circuit can be observed in Figure 4b. Kindly, refer to Table 1 for the component values used. The fitness evolution with respect to population is as shown in Figure 5.

4.2. Bipolar transistor based circuit

The circuit schematic of the bipolar transistor based Chua's circuit is shown in Figure 6. The V-I curve, resulted by simulation is presented in Figure 7a. The chaotic (double-scroll) behavior of the electronic circuit can be observed in Figure 7b.

The values of the components of the electronic circuit are shown in Table 2.

Note, that for both cases, the SPICE models of actual devices were used both for evolution as well as for simulation results presented here. Netlists of the evolved operational amplifier based Chua's diodes are presented in Table 1.



(a) (b)

Figure 4: (a) V-I curve of an OP400 type operational amplifier based Chua's diode (simulation results). (b) Double-scroll graph of an OP400 type of operational amplifier based Chua's circuit (simulation results).

5. Physical realization

One evolved circuit was built in the lab to verify the correctness of the algorithm. Since the algorithm was instructed to generate characteristics to very high accuracy, these values to 6 decimal places are generated by the algorithm. Once the values are achieved, the decimal places can be truncated and still we get the required nonlinear resistor with almost negligible change compared to the exact values. The schematic of the realized Chua's circuit can be seen in Figure 8. Measurements of the breadboarded version of Chua's circuit with evolved Chua's diode are shown in Figure 9a and Figure 9b. The values used to implement Chua's circuit are shown in Table 3.

Note that all the simulations as well as the breadboarded designs have been using ($C_1 = 100nF$, $C_2 = 10nF$ and $L = 18mH$) and the bifurcation resistor have been varied between 0 - $2K\Omega$. GSpice circuit simulation software was used on a PC under Linux operational system. Usually the time requirement of a successful circuit evolution was between half an hour and two hours.

6. Discussion

Exploiting genetic algorithms to develop Chaotic Circuits is a novel approach towards obtaining their electronic implementation. Even in the present

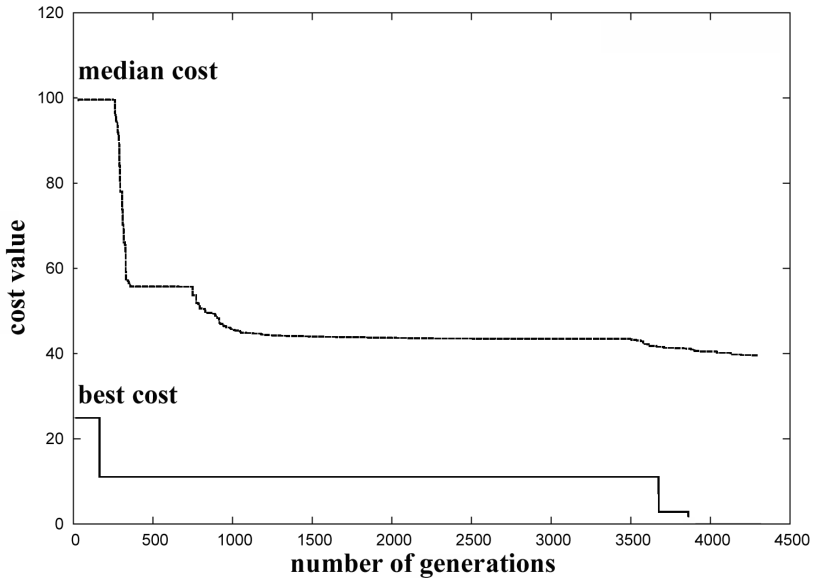


Figure 5: Cost evolution curve of an OP400 type operational amplifier based Chua's diode (horizontal axis is the number of generations, vertical axis is the cost value)

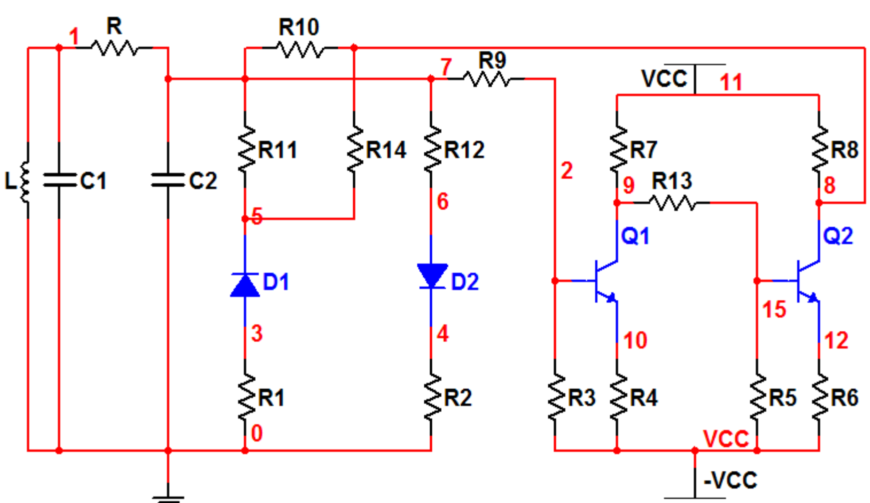
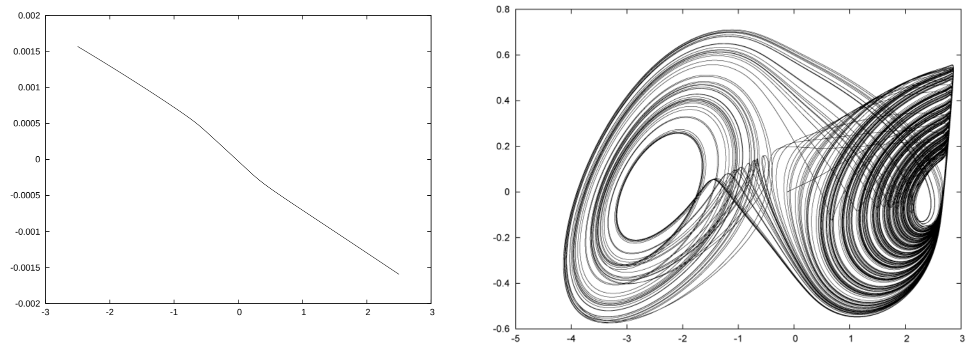


Figure 6: Schematic of the bipolar transistor based Chua's circuit (for values see Table 2).



(a) (b)

Figure 7: (a) V-I curve of a bipolar transistor based Chua's diode (simulation result). (b) Double-scroll graph of a bipolar transistor based Chua's circuit (simulation result).

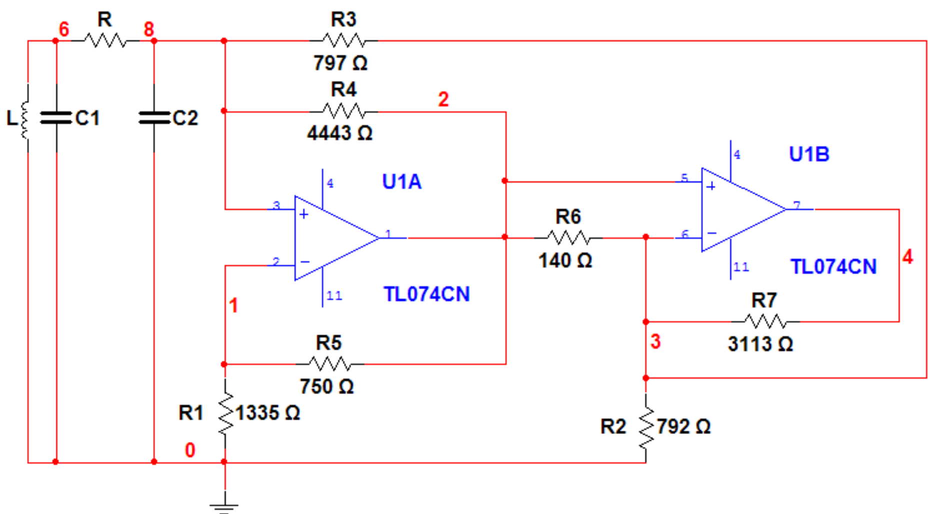
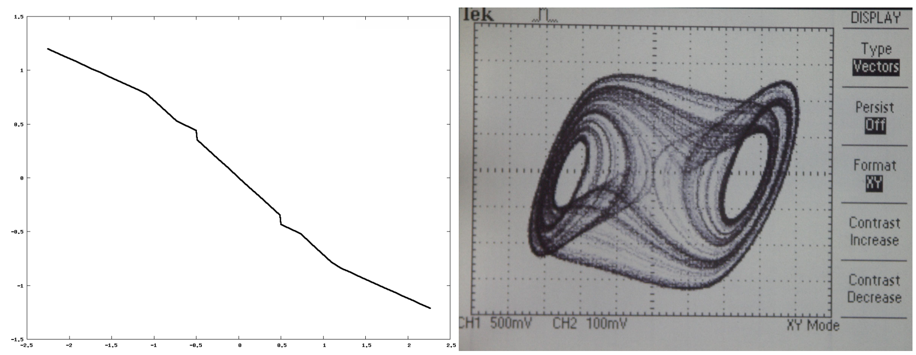


Figure 8: Schematic of a TL074CN type operational amplifier based Chua's circuit.

case of evolving Chua's circuits, previous knowledge of splitting the system is used. The knowledge of time series of chaotic systems is not used and hence this approach is limited to those chaotic circuits whose dynamics via



(a) (b)

Figure 9: (a) V-I curve of a TL074CN type operational amplifier based Chua's diode (oscilloscope measurement). (b) Double-scroll graph of a TL074CN type operational amplifier based Chua's circuit (oscilloscope measurement).

its electronic implementation is known. Thus, this case is not limited to Chua's circuit only but can be extended to evolve different case studies of nonlinear circuits.

Another direct application of the current approach can be in the topology evolution of traditional analog circuits. The shrinking technology and nonlinear behavior of MOS devices makes it attractive to explore the application of the current algorithm in evolving traditional analog circuits at nanoscale. However, this aspect needs further investigation [13, 14].

7. Conclusions

A scheme of evolving Chua's circuit using genetic algorithm was presented. It has been shown that evolution of nonlinear analog resistors using genetic algorithm is much faster than the human intuitive way of thinking. The evolved design of Chua's diode and hence Chua's circuit has been shown to work satisfactorily both on SPICE as well as on breadboarded components, thereby proving the utility of incorporating genetic algorithm to SPICE to evolve human competitive complex circuits reliably and faster.

Acknowledgment

The Hungarian Scientific Research Fund (OTKA) is gratefully acknowledged. The authors are also grateful to Professor Tamás Roska and Giovanni Pazienza for the discussions and their suggestions. The authors would also like to thank Varun Aggarwal for his constant support and help.

References

[1] L. O. Chua, "The Genesis of Chua's circuit," *AEU. Archiv für Elektronik und Übertragungstechnik*, vol. 46, no. 4, pp. 250–257, 1992.

[2] L. Chua, M. Komuro, and T. Matsumoto, "The double scroll family," *Circuits and Systems, IEEE Transactions on*, vol. 33, no. 11, pp. 1072–1118, 1986.

[3] T. Matsumoto, "A chaotic attractor from Chua's circuit," *Circuits and Systems, IEEE Transactions on*, vol. 31, no. 12, pp. 1055–1058, 1984.

[4] G. Gandhi, "An Improved Chua's Circuit and Its Use in Hyperchaotic Circuit," *Analog Integrated Circuits and Signal Processing*, vol. 46, no. 2, pp. 173–178, 2006.

[5] R. Senani and S. Gupta, "Implementation of Chua's chaotic circuit using current feedback op-amps," *Electronics Letters*, vol. 34, no. 9, pp. 829–830, 1998.

[6] R. Kilic, M. Alci, and H. Kuntman, "Improved realization of mixed-mode chaotic circuit," *International Journal of Bifurcation and Chaos*, vol. 12, no. 6, pp. 1429–1435, 2002.

[7] A. Radwan, A. Soliman, and A. El-Sedeek, "An Inductorless CMOS realization of Chua's circuit," *Chaos, Solitons and Fractals*, vol. 18, no. 1, pp. 149–158, 2003.

[8] G. Zhong and F. Ayrom, "Periodicity and Chaos in Chua's Circuit," *Circuits and Systems, IEEE Transactions on*, vol. 32, no. 5, pp. 501–503, 1985.

[9] L. Chua, C. Wu, and A. Huang, "A universal circuit for studying and generating chaos. I. Routes tochaos," *Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 10, pp. 732–744, 1993.

[10] R. Madan, “Special Issue on Chua's Circuit: A Paradigm for Chaos,” *Journal of Circuits, Systems and Computers*, 1990.

[11] C. Wu and L. Pivka, “From Chua's circuit to Chua's oscillator: A picture book of attractors,” *Nonlinear Dynamics of Electronic Systems, Proc. Workshop, eds. Davies, AC & Schwarz, W.(World Scientific, Singapore)*, 1994.

[12] A. Elwakil and M. Kennedy, “Chua's circuit decomposition: a systematic design approach for chaotic oscillators,” *Journal of the Franklin Institute*, vol. 337, no. 2-3, pp. 251–265, 2000.

[13] M. Terry, J. Marcus, M. Farrell, V. Aggarwal, and U. O Reilly, “GRACE: Generative Robust Analog Circuit Exploration,” *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3907, p. 332, 2006.

[14] V. Aggarwal and U. O'Reilly, “Design of Polynomial Models for Mosfets: Symbolic Regression Using Genetic Algorithms,” *Constraints*, vol. 51, no. L2, pp. 14–1.

Table 1: Netlists of the evolved Chua's diodes. These circuits contain two operational amplifiers and additional resistances. The positive input of one of the operational amplifiers is no. 4, the negative input is no. 5, positive power is no. 2, negative power is no. 1 and its output is no. 6. The other operational amplifier similarly connected to 7,8,2,1 and 9 nodes in order. In the Table, the first column contains the type of the operational amplifiers and the positive-negative power values. The second column defines the power-nodes. In the third column, the grounded nodes are given. The resistance cells contain two parameters. The upper one defines two nodes that the resistance connects. The lower one is the value of the resistance in Ω.

OPAmp	±V	0V	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19
SSM2275 ±12V	2-0 0-1	0,10	4-6 3161	5-6 1000	6-7 4427	3-8 2326	4-8 593	8-9 6772	5-10 1000												
AD8047 ±12V	2-0 0-1	0	0-1 1833	0-5 2485	0-6 748	0-8 1367	1-6 2550	1-9 369	2-9 623	3-4 436	3-5 13147	3-6 1140	3-8 1970	4-5 439	4-7 4703	5-6 2331	6-9 685	8-9 11660			
OP400 ±10.3V	2-0 0-1	0,10	3-4 141	4-6 4849	5-6 4144	6-7 9124	5-8 746	6-8 5551	7-8 1265	3-9 9842	8-9 7030	5-10 1000	8-10 4938								
AD645 ±12V	2-0 0-1	0	0-2 3143	0-5 1000	0-6 2021	0-8 1160	0-9 3069	1-2 1913	1-9 3478	3-4 368	3-7 662	3-8 1823	4-6 4259	5-6 1000	5-7 1868	6-7 1449	6-8 3586	6-9 670	7-8 3040	8-9 4705	
AD817 ±12V	2-0 0-1	0	0-1 3972	0-2 3847	0-5 1000	0-6 1719	0-9 12	1-2 3272	1-9 1796	2-6 5881	2-9 3168	3-4 95	3-7 1751	3-8 2356	4-6 7649	4-7 4553	5-6 1570	6-7 3013	6-8 591	7-8 1767	8-9 356
AD8024 ±12V	2-0 0-1	0,3	0-2 734	0-5 1263	0-6 6094	0-8 10052	0-9 1129	1-2 305	1-6 11050	1-9 421	2-6 4472	3-4 495	4-5 244	4-6 7052	4-8 2850	5-6 2334	5-7 7699	6-7 1915	6-9 13	7-8 465	8-9 9268
AD8079A ±12V	2-0 0-1	0,3	0-1 1073	0-2 2427	0-5 103	0-6 3850	0-8 1604	0-9 424	1-2 1191	1-9 495	2-9 1797	3-4 680	3-6 1629	3-8 2044	4-6 6093	5-6 1000	5-8 1002	6-7 1187	6-8 3667	6-9 11	7-9 1889
AD8011AN ±12V	2-0 0-1	0,3	0-1 3267	0-5 1469	0-6 2776	0-8 564	0-9 5250	1-2 1043	1-6 5417	1-9 981	2-9 8822	3-4 753	4-5 305	4-6 3613	4-7 4178	4-8 874	5-6 2573	5-8 1239	6-8 3168	7-8 671	8-9 4458
TL074 ±10.9V	2-0 0-1	0,4, 7,10	4-6 3691	5-6 1813	5-8 1332	8-9 9891	5-10 947														
OP113 ±12V	2-0 0-1	0,3	0-2 104	0-5 1000	0-6 348	0-9 1393	2-6 895	2-9 604	3-4 10	3-5 1445	3-9 1812	4-6 4775	4-7 656	4-9 1081	5-6 3284	5-7 1405	5-8 229	5-9 503	7-8 1278	8-9 2670	
SSM2275 ±12V	2-0 0-1	0,10	4-5 5955	4-6 6000	5-6 1000	6-7 1191	4-8 495	6-8 7196	4-9 8463	5-9 9313	6-9 493	5-10 1000									
AD645 ±11.4V	2-0 0-1	0,4,10	4-5 9239	4-6 6000	5-6 1000	4-7 4059	4-8 3318	5-8 2712	4-9 4642	6-9 509	8-9 8073	5-10 215									
AD746 ±12.8V	2-0 0-1	0,4,10	4-7 2795	7-8 6919	4-9 1089	5-9 3599	7-9 1821	8-9 1096	5-10 3538	8-10 9817											

component	connected nodes	type or value
VPSup	2 0	12.000000 V
VNSup	0 1	12.000000 V
Q1	4 5 6	Q2N2222A
Q2	7 8 9	Q2N2222A
D1	10 11	D1N4001
D2	12 13	D1N4001
R1	0 10	4063.419922 Ω
R2	0 13	2482.699951 Ω
R3	1 5	2145.459961 Ω
R4	1 6	350.000000 Ω
R5	1 8	100340.039062 Ω
R6	1 9	295.540009 Ω
R7	2 4	2118.000000 Ω
R8	2 7	1000.000000 Ω
R9	3 5	5380.000000 Ω
R10	3 7	352.000000 Ω
R11	3 11	4031.620117 Ω
R12	3 12	1041.199951 Ω
R13	4 8	48162.101562 Ω
R14	7 11	147.039993 Ω

Table 2: Netlist of the bipolar transistor based Chua's diode.

parameter	simulation value	real value
C_1	10 nF	10 nF
C_2	100 nF	100 nF
L_1	18 mH	18 mH
R_1	1334.760010 Ω	1334 Ω
R_2	792.000000 Ω	792 Ω
R_3	797.880005 Ω	797 Ω
R_4	4443.279785 Ω	4443 Ω
R_5	750.520020 Ω	750 Ω
R_6	140 Ω	140 Ω
R_7	3112.879883 Ω	3112 Ω

Table 3: Simulation and physical parameter values