2006-410: DEMONSTRATION OF CIRCUIT DESIGN USING RANDOMNESS, EVOLUTION AND NATURAL SELECTION

Glenn Kohne, Loyola College in Maryland

Glenn S. Kohne is currently associate professor of engineering science at Loyola College, Baltimore, MD. He received an M.E.S. from Loyola College in 1981 and a B.S.E.E. from the University of Maryland in 1970. His research interests include computer science, digital signal processing, and education.

Steven O'Donnell, Loyola College in Maryland

Mr. O’Donnell is a senior electrical engineering student at Loyola College in Maryland. He has studied abroad at Monash University in Melbourne Autralia. He has experience as a Hauber research grantee and as an intern at Middle River Aircraft Systems. His interests include skiing, waterskiing, fishing and traveling.

Demonstration of Circuit Design using Randomness, Evolution and Natural Selection

Key Words: circuit design, analog filter, Darwinian circuit, evolutionary circuit, genetic circuit design, transfer function, iterative solution, student research project

Prerequisite Knowledge: Basic Linear Circuit Analysis and Windows Application programming.

Objective(s):
1. To explore the design of circuits using randomness and evolutionary principles;
2. To develop a tool for demonstrating the principles and for future research;
3. To demonstrate that people can create tools to perform design projects requiring knowledge more advanced than that held by the designer of the tools.

Equipment and Materials (include sources if appropriate):
1. A high-end PC with Windows 2000 (or better)
2. A compiler for Visual Basic 6.0 (or better)
3. MatrixVB (MATLAB product)
4. Access to an engineering reference library

(To use the tool developed by this project, only the PC is necessary)

Introduction: Most analog circuit design is inherently evolutionary in that the final product is achieved through an analytical analysis to determine parameter values, build a prototype, and test the prototype for "fitness". Results of this testing frequently indicate changes in the original structure or parameter values, thereby causing the initial design to evolve into the final design. This loop generally begins with the selection of a circuit topography known to be a good candidate for the desired performance. The more experienced and skillful the designer, the more likely this first choice will resemble the final design. What follows becomes, essentially, a parameter optimization problem. Finding the optimal parameters manually, or even with the assistance of circuitry simulators is time and labor intensive. Tools available to help here include simulated annealing and gradient search methods. Shortcomings in these methods are the requirement of huge computational time and the difficulties of avoiding local maxima and minima.

Several broad categories of genetic evolution that have been applied to solve problems in computer programming, biology, chemistry, digital circuit design, and, to a lesser extent, analog circuit design. The tool developed here will work with both a variable structure and a variable number of parameters for components in each structure. The "brute-force" method of trying to optimize every circuit parameter against all the other circuit parameters for a fixed structure can require significant computation time. Trying to use brute-force on hundreds of different structures is simply impractical. The use of natural selection for fitness with genetic evolution of the fittest allows a relatively fast examination of a variety of structures as well as optimization of the element parameters. As with most engineering problem solving, this does not necessarily give the provably optimal circuit, but it does provide a means to design to a defined standard. It opens the door for circuit structures that might not have been considered by a human designer. It also allows for circuit designs that exceed human analytical skills and/or the accuracy of the traditional mathematical models used for various circuit components.

Procedure: The project consists of four segments. The first is the design of the user interface. Second is the design of the chromosome and the genetic manipulations. Third is design of the "fitness" determination metric. Conducting experiments is the final segment.

The user interface is used to establish the circuit "habitat" and "fauna". The habitat will be a selectable maximum number of nodes within a grid space. Whether or not a given node is connected to anything will be determined by the genetic rules selected for the particular experiment. The fauna consists of the circuit element models that can exist for this experiment. The existence of a given type of circuit element in an experiment does not necessarily mean that it will be used in any of the circuits in the experiment. That too will be determined by the genetic rules. Display of intermediate and final circuits and analyses data are also part of the user interface.

The chromosome contains all information necessary to describe a distinct individual circuit. The chromosome consists of a collection of genes. Each gene fully describes a single circuit element including its location within the grid space. In nature, the genetic information of the chromosome is encoded in very long strings of DNA. Each string is a sequence that looks like a ladder where each rung is one of four basic building blocks. In the computer we can use a higher level of encoding allowing for complex genetic information to be represented by relatively simple genes. Each gene consists of four items: the location in the grid space for attachment of each lead; the type of electrical component; and the numerical value of the component. A chromosome fragment might look like the following gene sequence.

Node A	Node B	Element type	Value
3	5	resistor (ohms)	2000
3	7	resistor (ohms)	5200
4	6	capacitor (uF)	.01
2	9	inductor (mH)	5

Using this genetic code, it is easy to generate a wide variety of individual circuits as the initial generation of fauna. After determining the fitness of each individual circuit (discussed later), ten percent of the top performers are selected to parent the next generation. In much of nature, differentiation among offspring is achieved by combining genetic information from two parents with a variety of occasional mutations. Unisexual reproduction is not rare and frequently gives rise to offspring that are morphologically indistinguishable from bisexually reproduced offspring. In the parasitic, Hymenoptera both modes may occur among closely related species yielding morphologically indistinguishable offspring. Unisexual reproduction is the method employed in this project. To form the chromosome of a child, various changes are imparted on some of the parental genes. This allows for each child to have a different structure than that of the parent. A resistor may become a capacitor. The attachment of one lead may change from one node to another. An individual gene might be deleted from the sequence. A new gene might be inserted. These types of changes in structure are considered radical and should occur relatively infrequently. It is necessary that they can occur because as in nature, sometimes a radical change produces a more robust offspring. Changes to the value of a given gene provide for more normal differentiation between the children. This mechanism provides for parameter optimization from generation to generation.

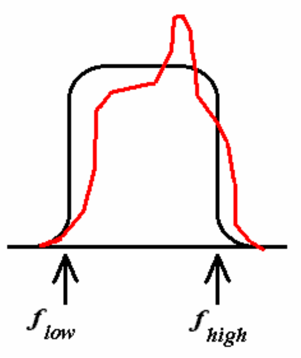
Design of the metric to determine fitness is the most challenging step in the experiment. Ultimately, fitness must be represented as a single number. Consider that a band pass filter is the goal. The fitness metric *f,* might be constructed as the sum of the error between the individual's response and the goal at a number of frequencies, N.

$$f = \sum_{i=1}^{N} \left(\left.\frac{V_O}{V_I}\right|_{\omega_{goal}} - \left.\frac{V_O}{V_I}\right|_{\omega_i} \right)^2$$

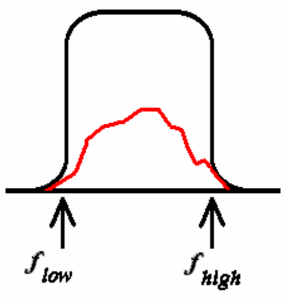
Another fitness metric might require that the response of the individuals be normalized in one of a number of ways before performing the squares computation. A third metric might combine the first two metrics giving them adjustable weights. In this case, the final fitness number for M metrics would be:

$$F = \sum_{m=1}^{M} Weight_m \cdot f_m$$

Consider the two cases below. Which actual response better conforms to the desired bandpass filter?



Circuit #1

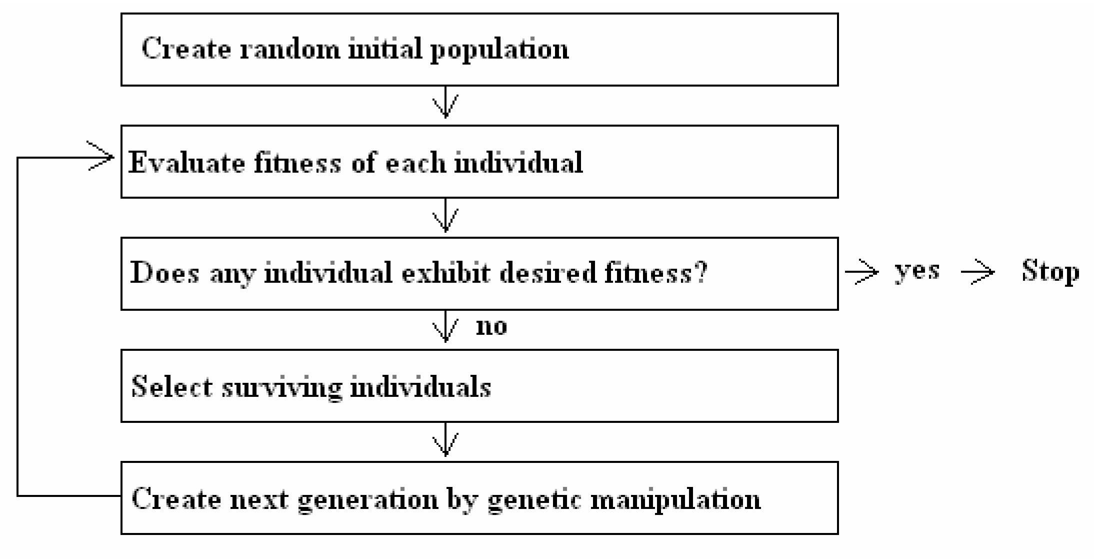


Circuit #2

If you give the highest number to the response that minimized the square of the error, you select circuit #1 as the more fit. If you normalize the responses and then give the highest number to the circuit with the minimum square of the error, you select circuit #2.

The fitness metric(s) must be tailored to the target goal. The fitness of filters, oscillators, attenuators, and others will be evaluated by measuring different parameters. Bandpass filters must have a frequency response that conforms to the shape of the target. Oscillators must oscillate at a given frequency with a minimum of distortion, etc. Providing different metrics is an ongoing activity for this project.

Conducting an experiment begins with the selection of the parameters defining the habitat and fauna. Also established at the beginning are the rules for mutations, the limits of the universe such as number of individuals in the initial population, maximum number of generations allowed, percentage of a generation that will survive, etc. The block diagram of the experimental algorithm is shown below:

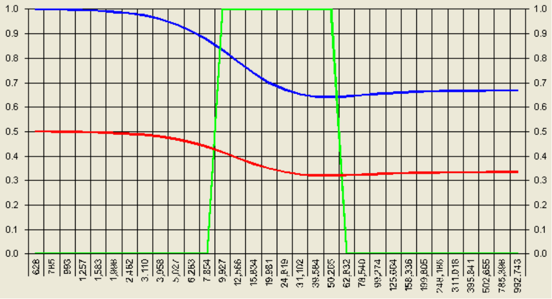

Create random initial population
Evaluate fitness of each individual
Does any individual exhibit desired fitness?
yes
Stop
no
Select surviving individuals
Create next generation by genetic manipulation

Results: The first experiment was designed to verify the software. The root generation was a single circuit with six resistors and three capacitors. From this root, thirty children were generated. The fittest 3 circuits were selected to survive. Each of the survivors parented ten children, yielding a second generation of 30. Each subsequent generation was produced by having three survive to parent ten. To ensure that fitness gains of the parents were not lost in the children, one child was made to be identical to its parent. The other nine children were mutated versions of the parent.

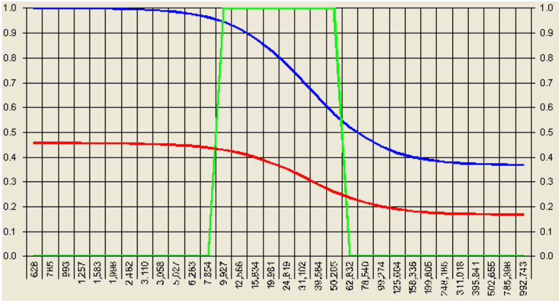
In the graphs below, the target circuit performance of a band-pass filter is shown by the green trapezoidal shape around the center of the x-axis. The pass-band centers at 21,991 (3,500 Hz) with a lower boundary of 9,927 (1,580 Hz) and an upper boundary of 50,266 (8,000 Hz).

The lower of the two curves (red) is the actual circuit response and the upper of the two curves (blue) is the normalized circuit response. For each graph, the x-axis is logarithmic from 628 (100 Hz) to 992,743 (15,800 Hz), and the y-axis is linear gain.

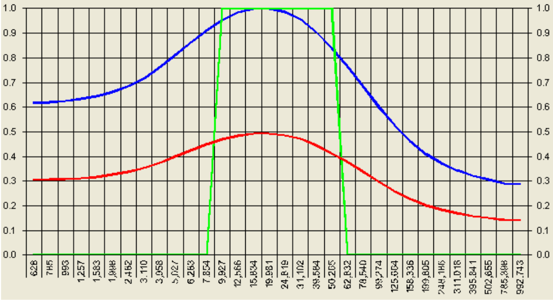
The root generation response:



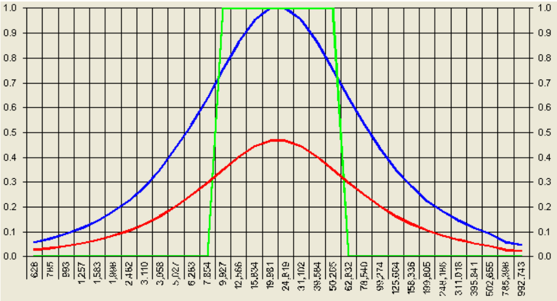
The best of the 11th generation:



The best of the 25th generation:



The best of the 60th generation:



The response of the root generation bears little resemblance to the target response. By the 11th generation, the response is obviously improved but is still very much different than the target. By the 25th generation, the response has become roughly the same shape as the target. In the 60th generation the response has appropriate shape as the target, is centered at the correct frequency, and has matching above-band and below-band performance.

Given the limited number of generations, the limited number of individuals in each generation and, the even more limited number of survivors in each generation, there was little opportunity for the circuits to evolve much beyond the original circuit's six resistors and three capacitors. The response at the 60th generation is pretty much the best that can be expected with so few components.

If each generation consisted of 10,000 individual circuits where 1,000 survived to parent, the opportunity for an inductor to appear in parallel with one of the capacitors, and to survive through enough generations to obtain an appropriate value, would improve dramatically. Allowing the population to continue for 10,000 generations would enhance the refining of those circuit structures that are successful. It would provide opportunity for bizarre mutations to occur and, possibly survive. The problem with these numbers is the computer execution time required to evaluate all of the created circuits. One solution well suited to this problem is the use of parallel processing to reduce computation time. An effort to adapt this software to a parallel-processor system is currently underway by the co-author as a two semester Senior Design Project.

Instructor's Notes: This project is suitable for student research in a 10 week, 40 hours/week timeframe. Approximately three weeks should be planned for the design of the user interface that establishes the habitat and fauna, sets initial conditions and performance parameters, reports intermediate results and displays final circuitry and analyses. Three to four weeks should be planned for designing and testing fitness evaluation metrics. If providing more than one metric, it is sensible to begin with the one that is easiest to realize. Time permitting, more complex evaluation criteria can be programmed into an optional metric. The final three to four weeks should be reserved for running experiments to design several different circuits.

The co-author of this paper is a Senior engineering student. Much of the work reported here was performed by him under the aegis of the Hauber Summer Research grants program at our college. Assessment of the student's achievement involved both objective and subjective components. He developed a working knowledge of an object-oriented programming language and demonstrated the ability to design and implement a complex engineering application. The effective use of the MATLAB operators and functions demonstrated his ability to use modern engineering tools to solve matrix problems using complex numbers. The subject circuit for this work was a bandpass filter. Examination of the frequency response curves given above for the root, 11th, 25th, and 60th generations verifies two things: Firstly, that the method of using evolutionary concepts and natural selection is a viable way to achieve circuit design, and; Secondly, that the software designed and developed by the student performs the intended functions.

It would be instructive to actually build the final circuitry and evaluate with standard electronic testing equipment.

It would not be unreasonable to start with a generation of 1,000 individuals. Select the fittest 100 individuals to parent the next generation where each parent creates 10 children to maintain a second generation of 1,000. Obviously these numbers can be increased based on the speed of the computer used. The computational problem is well suited to solution by parallel processors. If you create a cluster of 10 computers, each computer would need to evaluate only 100 of the 1,000 individuals cutting processing time by a factor of 10. Creating a cluster of several dozen would also allow you to use much less sophisticated computers to achieve the same computational speed. As newer models of computers are introduced, many schools find themselves with plenty of excess computers that would fit nicely into a parallel processing cluster.

Preliminary results of this work were presented at the National Educators' Workshop: Update 2005, held in October 2005 at NIST in Gaithersburg, MD. At the conclusion of that conference, we were invited to submit this paper, documenting final results, to the NEW division at the ASEE annual conference.

Bibliography:

1. J.H. Holland, *Adaptation in Natural and Artificial Systems,* Univ. of Michigan Press, Ann Arbor, 1975

2. W. Druiskamp, D. Leenaerts, "Darwin: Analogue Circuit Synthesis Based on Genetic Algorithms," *International Journal of Circuit Theory and Applications,* Vol. 23, 1995, pp. 285-296

3. M.W. Kruiskamp, *Analog Design Automation using Genetic Algorithms and Polytopes,* Ph.D. Thesis, Dept. of Elect. Engr., Eindhoven University of Technology, Eindhoven, The Netherlands, 1996.

4. J.R. Koza, F.H. Bennett, D. Andre, M.A. Keane, F. Dunlap, "Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming," *IEEE Trans. On Evolutionary Computation,* vol. 1, no. 2, July, 1997, pp 109-128

5. R.S. Zebulum, M.A. Pacheco, M. Vellasco, "Comparison of Different Evolutionary Methodologies Applied to Electronic Filter Design," *1998 IEEE Int. Conf. On Evolutionary Computation,* Piscataway, NJ: IEEE Press, 1998, pp. 434-439.

6. K.V. Noren, J.E. Ross, "Analog Circuit Design Using Genetic Algorithms," *Second Online Symposium for Electronics Engineers,* Summer 2001.

7. J. Hu, K. Seo, S. Li, Z. Fan, R.C. Rosenberg, E.D. Goodman, "Structure Fitness Sharing (SFS) for Evolutionary Design by Genetic Programming," *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference,* NY, NY, Morgan Kaufman, San Francisco, July 2002, pp. 780-787.

8. G.S. Kohne, W.J. Karasz, "Design of Complex Technical Equipment by Darwinian Evolution," *Proceedings of National Educators' Workshop: Update 2003,* Langley Research Center, Hampton, Virginia, October 19-22, 2003.