

# AUTOMATIC SYNTHESIS USING GENETIC PROGRAMMING OF BOTH THE TOPOLOGY AND SIZING FOR FIVE POST-2000 PATENTED ANALOG AND MIXED ANALOG-DIGITAL CIRCUITS

*Matthew J. Streeter*

Genetic Programming Inc.  
Mountain View, California  
mjs@tmolp.com

*Martin A. Keane*

Econometrics Inc.  
Chicago, Illinois  
martinkeane@ameritech.net

*John R. Koza*

Stanford University  
Stanford, California  
koza@stanford.edu

## ABSTRACT

Recent work has demonstrated that genetic programming can automatically create both the topology (graphical structure) and sizing (numerical component values) for analog electrical circuits merely by specifying the circuit's high level behavior (e.g., its desired or observed output, given its input). This automatic synthesis of analog circuits is accomplished using only tools for the *analysis* of circuits (e.g., a circuit simulator) and without relying on any human know-how concerning the *synthesis* of circuits. This paper applies genetic programming to the automatic synthesis of five analog and mixed analog-digital circuits that duplicate the functionality of circuits patented after January 1, 2000. The five automatically created circuits read on some (but not all) of the elements of various claims of the patents involved (and therefore do not infringe). The described method can be used as an automated invention machine either to produce potentially patentable new circuits or to “engineer around” existing patents.

## I. INTRODUCTION

Patents represent current research and development efforts of the engineering and scientific communities.

Genetic programming can automatically create both the topology (graphical structure) and sizing (numerical component values) for analog electrical circuits (e.g., filters, amplifiers) composed of transistors, capacitors, resistors, and other components. Our method for automatically synthesizing analog circuits starts from a high-level statement of a circuit's desired behavior and characteristics (e.g., its desired or observed output given its input). It uses only *de minimus* knowledge about analog circuits. The method employs a circuit simulator (e.g., SPICE) for the *analysis* of candidate circuits, but does not use any deep knowledge or expertise about the *synthesis* of circuits.

This paper reports on a project in which we browsed the patent literature for patents on analog electrical circuits issued since January 1, 2000 to commercial enterprises or university research institutions. We then used genetic programming to automatically synthesize both the topology and sizing for circuits that duplicate the functionality of the patented inventions. The five inventions are shown in table 1.

**Table 1 Five post-2000 patented analog circuits**

Invention	Inventor(s) and date	Institution
Low-voltage balun circuit	Sang Gug Lee (2001)	Information and Communications University
Mixed analog-digital variable capacitance	Turgut Sefket Aytur (2002)	Lucent Technologies Inc.
Voltage-current converter	Akira Ikeuchi and Naoshi Tokuda (2000)	Mitsumi Electric Co., Ltd.
High-current load circuit for testing a voltage source	Timothy Daun-Lindberg and Michael Miller (2001)	International Business Machines Corporation
Low-Voltage cubic function generator	Stefano Cipriani and Anthony A. Takeshian (2000)	Conexant Systems, Inc.

The method described can be used as an invention machine either to produce potentially patentable new circuits or to “engineer around” existing patents.

The method has also automatically created both the topology and sizing for controllers, antennas, and networks of chemical reactions (Koza, Keane, Yu, Bennett, and Mydlowec 2000; Koza, Keane, Streeter, Mydlowec, Yu, and Lanza 2003).

Section 2 describes genetic programming. Section 3 explains the preparatory steps performed by the human user prior to launching a run of genetic programming. Section 4 presents the results. Section 5 covers novelty-driven evolution. Section 6 shows five 20<sup>th</sup> century

patented analog circuits that have been automatically synthesized by genetic programming.

## II. GENETIC PROGRAMMING

Genetic programming is an automatic method for solving problems. Specifically, genetic programming progressively breeds a population of computer programs over a series of generations. Genetic programming starts with a primordial ooze of thousands of randomly created computer programs and uses the Darwinian principle of natural selection, recombination (crossover), mutation, gene duplication, gene deletion, and certain mechanisms of developmental biology to breed an improved population over a series of many generations.

Genetic programming (Koza 1992; Koza and Rice 1992; Koza, Bennett, Andre, and Keane 1999; Koza, Bennett, Andre, Keane, and Brave 1999; Koza, Keane, Streeter, Mydlowec, Yu, and Lanza 2003) breeds computer programs to solve problems by executing the following three steps:

- (1) Generate an initial population of compositions (typically random) of the functions and terminals of the problem.
- (2) Iteratively perform the following substeps (a generation) on the population of programs until the termination criterion has been satisfied:
  - (A) Execute each program in the population and assign it a fitness value using the problem's fitness measure.
  - (B) Create a new population of programs by applying the following operations to program(s) selected from the population with a probability based on fitness (with reselection allowed).
    - (i) Reproduction: Copy the selected program to the new population.
    - (ii) Crossover: Create a new offspring program for the new population by recombining randomly chosen parts of two selected programs.
    - (iii) Mutation: Create one new offspring program for the new population by randomly mutating a randomly chosen part of the selected program.
    - (iv) Architecture-altering operations: Select an architecture-altering operation from the available repertoire of such operations and create one new offspring program for the new population by applying the selected architecture-altering operation to the selected program.

- (3) Designate the individual program that is identified by result designation (e.g., the best-so-far individual) as the result of the run. This result may be a solution (or approximate solution) to the problem.

When genetic programming is used to automatically create computer programs, the programs are ordinarily represented as program trees (i.e., rooted, point-labeled trees with ordered branches). In contrast, electrical circuits are usually represented as labeled graphical structures in which each component is included in a cycle. Thus, there is a representational obstacle that must be overcome before genetic programming can be applied to the problem of automatically synthesizing circuits. This obstacle can be overcome by establishing a mapping between program trees and labeled cyclic graphs. The mapping from trees into circuits is accomplished by means of a developmental process. This process begins with a simple embryo. The embryo used herein consists of a single modifiable wire that is not initially connected to the inputs or outputs of the to-be-created circuit. An analog electrical circuit is developed by progressively applying the functions in a circuit-constructing program tree to the embryo's initial modifiable wire (and to succeeding modifiable wires and modifiable components).

The functions in the circuit-constructing program trees include

- (1) topology-modifying functions that alter the topology of a developing circuit (e.g., series division, parallel division, via between nodes, via to ground, via to a power supply, via to input, via to output),
- (2) component-creating functions that insert components (i.e., resistors, capacitors, and transistors) into a developing circuit, and
- (3) development-controlling functions that control the developmental process (e.g., cut, end).

## III. PREPARATORY STEPS

Genetic programming starts from a high-level statement of the problem's requirements couched in terms of human-supplied preparatory steps describing "what needs to be done." The five major preparatory steps for genetic programming entail determining

- (1) the set of functions,
- (2) the set of terminals,
- (3) the fitness measure for measuring the fitness of individuals in the population,
- (4) parameters for controlling the run, and
- (5) a termination criterion.

The main difference between the runs of genetic programming for the five problems is that we supplied

a different fitness measure for each problem (as described below). Construction of a fitness measure requires translating the problem's high-level requirements into a precise computation. We read the patent document to find the performance that the invention was supposed to achieve. We then created a fitness measure reflecting the invention's performance and characteristics. The fitness measure specifies what time-domain output value(s) are desired, given various time-domain input value(s). For each specific problem, a test fixture consisting of certain fixed components (such as a source resistor, a load resistor) is connected to the desired input port(s) and the desired output port(s). Circuits are simulated using SPICE (Quarles, Pederson, Newton, Sangiovanni-Vincentelli 1994).

The function and terminal sets for all five problems permit the construction of any circuit composed of transistors, resistors, and capacitors. We supplied models for transistors appropriate to the problem. We used the commercially common 2N3904 (*nnp*) and 2N3906 (*pnp*) transistor models unless the patent document called for a different model. We used 5-Volt power supplies unless the patent specified otherwise.

The control parameters and termination criterion were the same for all five problems, except that we used different population sizes to approximately equalize each run's elapsed time per generation.

For additional details, see Koza, Keane, Streeter, Mydlowec, Yu, and Lanza 2003.

We now describe the five fitness measures.

#### **A. Low-Voltage Balun Circuit**

The purpose of a balun (balance/unbalance) circuit is to produce two outputs from a single input, each output having half the amplitude of the input, one output being in phase with the input while the other is 180 degrees out of phase with the input, with both outputs having the same DC offset. The patented balun circuit uses a power supply of only 1 Volt. The fitness measure consisted of (1) a frequency sweep analysis designed to ensure the correct magnitude and phase at the two outputs of the circuit and (2) a Fourier analysis designed to penalize harmonic distortion.

#### **B. Mixed Analog-Digital Register-Controlled Variable Capacitor**

This mixed analog-digital circuit has a capacitance that is controlled by the value stored in a digital register. The fitness measure employed 16 time-domain fitness cases. The 16 fitness cases ranged over all eight possible values of a 3-bit digital register for two different analog input signals.

#### **C. Voltage-Current Conversion Circuit**

The purpose of the voltage-current conversion circuit is to take two voltages as input and to produce a stable current whose magnitude is proportional to the

difference of the voltages. We employed four time-domain input signals (fitness cases) in the fitness measure. We included a time-varying voltage source beneath the output probe point to ensure that the output current produced by the circuit was stable with respect to any subsequent circuitry to which the output of the circuit might be attached.

#### **D. High-Current Load Circuit**

The patent covers a circuit designed to sink a time-varying amount of current in response to a control signal. The patented circuit employs a number of FET transistors arranged in parallel, each of which sinks a small amount of the desired current. The fitness measure consisted of two time-domain simulations, each representing a different control signal.

#### **E. Low-Voltage Cubic Signal Generator**

The patent covers an analog computational circuit that produces the cube of an input signal as its output. The circuit is "compact" in that it contains a voltage drop across no more than two transistors.

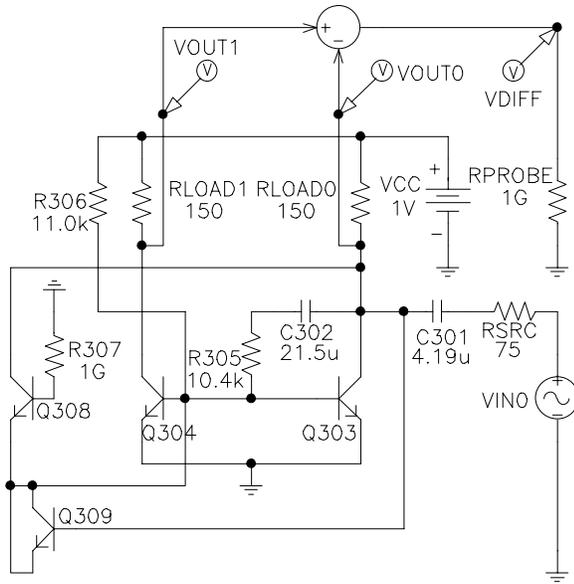
The fitness measure for this problem consisted of four time-domain fitness cases using various input signals and time scales. The compactness constraint was enforced by providing only a 2-Volt power supply.

## **IV. RESULTS**

#### **A. Low-Voltage Balun Circuit**

The best-of-run evolved circuit (figure 1) was produced in generation 97 and has a fitness of 0.429. The patented circuit has a fitness of 1.72. That is, the evolved circuit is roughly a fourfold improvement (less being better) over the patented circuit in terms of our fitness measure.

The evolved circuit is superior to the patented circuit both in terms of its frequency response and its harmonic distortion.



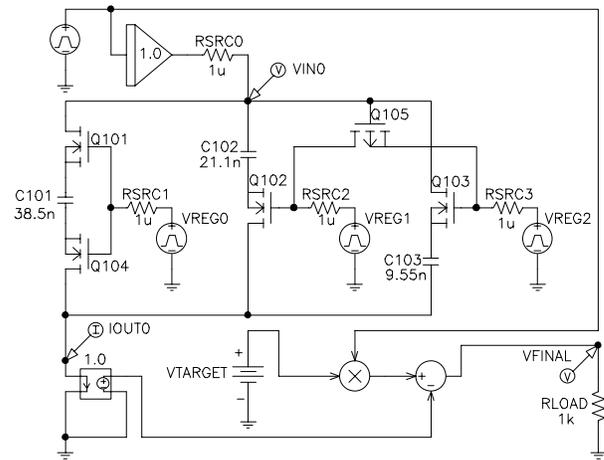
**Figure 1 Best-of-run balun circuit**

Lee (2001) identifies the essence of his invention in the patent documents. The difference between the prior art and Lee’s invention is a coupling capacitor located between the base and the collector of a certain transistor. This essential difference between the prior art and Lee’s invention is an integral part of claim 1 of Lee’s patent. The best-of-run genetically evolved balun circuit (figure 1) possesses the very capacitor that Lee identifies as the essence of his invention (called C302 in the figure). The genetically evolved circuit also reads on three additional elements of claim 1 of Lee’s patent. However, the genetically evolved circuit does not infringe Lee’s patent because it does not read on other elements enumerated in claim 1.

### B. Mixed Analog-Digital Register-Controlled Variable Capacitor

Over our 16 fitness cases, the patented circuit had an average error of 0.803 millivolts. In generation 95, a circuit emerged with average error of 0.808 millivolts, or approximately 100.6% of the average error of the patented circuit. During the course of this run, we harvested the smallest individuals produced on each processing node with a certain maximum level of error. Examination of these harvested individuals revealed a circuit from generation 98 (figure 2) that approximately matches the topology of the patented circuit (without infringing).

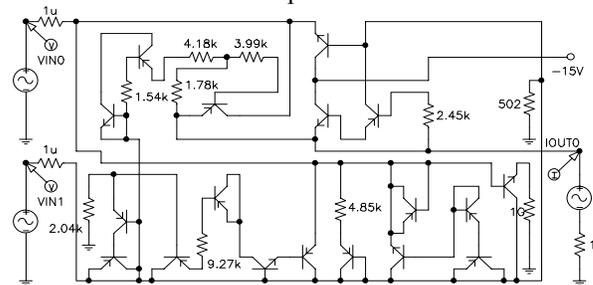
The genetically evolved circuit reads on all but one of the elements of claim 1 of the patented circuit (and hence does not infringe the patent).



**Figure 2 Best-of-run mixed analog-digital variable capacitor circuit**

### C. Voltage-Current Conversion Circuit

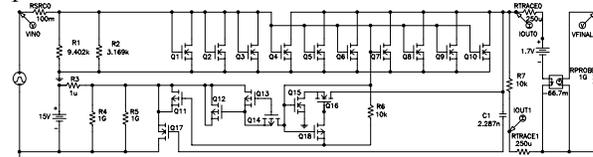
A circuit (figure 3) emerged on generation 109 of our run of this problem with a fitness of 0.619. That is, the evolved circuit has roughly 62% of the average (weighted) error of the patented circuit. The evolved circuit was subsequently tested on unseen fitness cases that were not part of the fitness measure and outperformed the patented circuit on these new fitness cases. The best-of-run circuit solves the problem in a different manner than the patented circuit.



**Figure 3 Best-of-run voltage current converter**

### D. High-Current Load Circuit

On generation 114, a circuit emerged that duplicated Daun-Lindberg and Miller’s parallel FET transistor structure. This circuit (figure 4) has a fitness (weighted error) of 1.82, or 182% of the weighted error for the patented circuit.



**Figure 4 Best-of-run high current load circuit**

The genetically evolved circuit shares the following features found in claim 1 of U.S. patent 6,211,726:

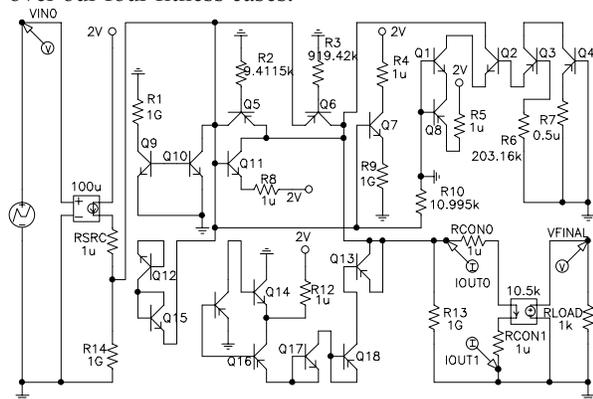
“A variable, high-current, low-voltage, load circuit for testing a voltage source, comprising:

“a plurality of high-current transistors having source-to-drain paths connected in parallel between a pair of terminals and a test load.”

However, the remaining elements of claim 1 in U.S. patent 6,211,726 are very specific and the genetically evolved circuit does not read on these remaining elements. The remaining elements of the genetically evolved circuit bear hardly any resemblance to the patented circuit. In this instance, genetic programming produced a circuit that duplicates the functionality of the patented circuit without infringing.

### E. Low-Voltage Cubic Signal Generator

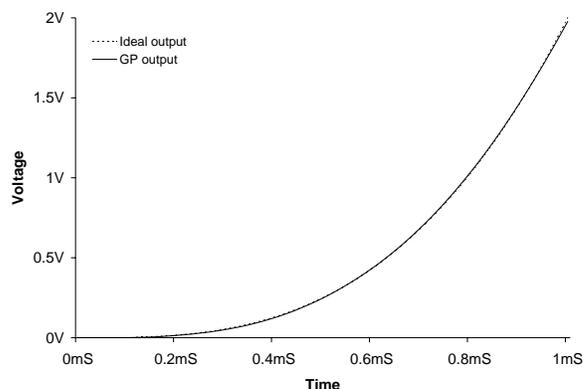
The best-of-run evolved circuit (figure 5) was produced in generation 182 and has an average error of 4.02 millivolts. The patented circuit had an average error of 6.76 millivolts. That is, the evolved circuit has approximately 59% of the error of the patented circuit over our four fitness cases.



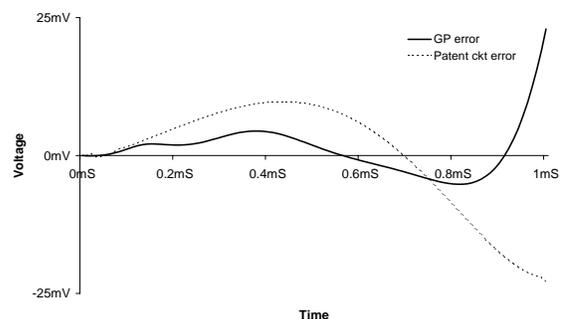
**Figure 5 Best-of-run cubic signal generator**

Averaged over the four fitness cases, the best-of-run individual from generation 182 has 4.20 millivolts average absolute error, and 26.7 millivolts maximum absolute error.

Figure 6 compares the output produced by the best-of-run cubic signal generation circuit from generation 182 (solid line) against the target cubic curve (dotted line). As can be seen, the two curves are almost indistinguishable.



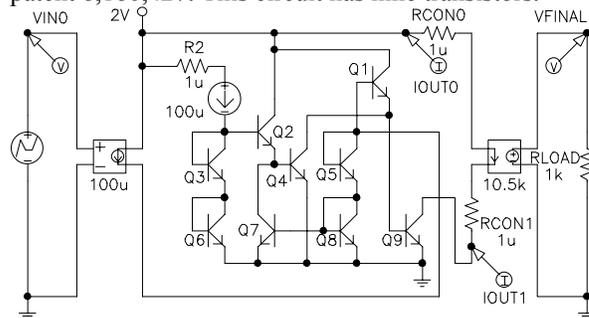
**Figure 6 Output produced by the best-of-run cubic signal generator**



**Figure 7 Comparison of the error of the best-of-run cubic signal generator and patented circuit**

Figure 7 compares the error of the best-of-run cubic signal generation circuit from generation 182 of the first run and the error of the circuit of U.S. patent 6,160,427. As can be seen from the figure, the error produced by the genetically evolved circuit is generally less than that produced by the patented circuit.

Figure 8 shows the cubic function generator of U.S. patent 6,160,427. This circuit has nine transistors.



**Figure 8 Patented cubic function generator**

Averaged over the four fitness cases, the patented circuit has 6.76 millivolts average absolute error, and 17.3 millivolts maximum absolute error.

The claims in U.S. patent 6,160,427 amount to a very specific description of the patented circuit. The genetically evolved circuit does not read on these claims and, in fact, bears hardly any resemblance to the patented circuit. In this instance, genetic programming produced a circuit that duplicates the functionality of the patented circuit with a very different structure.

#### V. NOVELTY-DRIVEN EVOLUTION

One may be interested in patenting a novel circuit for commercial advantage. Alternatively, one might simply have a scientific interest in producing novel solutions to challenging problems. Or, alternatively, one may want to avoid infringing an existing patent (either to avoid paying royalties or because the patent holder is unwilling to license a competitor).

In any of the above three situations, the fitness measure can incorporate the degree to which a candidate satisfies the problem's technical design requirements and the degree to which it avoids characteristics that read on prior art.

Because circuits can be conveniently represented by labeled graphs, a graph isomorphism algorithm can be applied to the candidate circuit and various template graphs representing key characteristics of the relevant prior art. For example, the templates for a balun would represent key characteristics of Lee's now-patented balun and all other prior art circuits (such as those cited by Lee himself in his patent). The measure of similarity can be based on the size of the maximal common subgraph between a candidate circuit and a template. For details, see Koza, Bennett, Andre, and Keane 1999.

#### VI. 20<sup>TH</sup> CENTURY PATENTED CIRCUITS

As an additional indicator of the ability of genetic programming to automatically synthesize both the topology and sizing of analog electrical circuits, table 2 shows five 20<sup>th</sup> century patented circuits that were previously automatically synthesized by means of genetic programming (Koza, Bennett, Andre, and Keane 1999).

**Table 2 Five 20<sup>th</sup> century patented circuits**

Invention	Inventor(s) and date	Institution
Darlington emitter-follower section (transistor circuit)	Sidney Darlington (1953)	Bell Telephone Laboratories
Ladder filter (LC circuit)	George Campbell (1917)	American Telephone and Telegraph
Crossover filter (LC circuit)	Otto Julius Zobel (1925)	American Telephone and Telegraph
"M-derived half section" filter (LC)	Otto Julius Zobel (1925)	American Telephone and

circuit)		Telegraph
Philbrick circuit (RC circuit)	George Philbrick (1956)	George A. Philbrick Researches

#### VII. CONCLUSIONS

We applied genetic programming to the problem of automatically synthesizing circuits that duplicate the functionality of five analog and mixed analog-digital circuits patented after January 1, 2000. The five automatically created circuits read on some (but not all) of the elements of various claims of the patents involved (and therefore do not infringe). Genetic programming can be used as an automated invention machine either to produce potentially patentable new circuits or to "engineer around" existing patents. As computer power continues to increase in accordance with Moore's Law, we anticipate that the use of genetic programming as an invention machine will become more and more common.

#### REFERENCES

- Aytur, Turgut Sefket. 2000. *Integrated Circuit with Variable Capacitor*. U. S. patent 6,013,958. Filed July 23, 1998. Issued January 11, 2000.
- Campbell, George A. 1917. *Electric Wave Filter*. Filed July 15, 1915. U.S. Patent 1,227,113. Issued May 22, 1917.
- Darlington, Sidney. 1953. *Semiconductor Signal Translating Device*. U.S. Patent 2,663,806. Filed May 9, 1952. Issued December 22, 1953.
- Cipriani, Stefano and Takeshian, Anthony A. 2000. *Compact cubic function generator*. U. S. patent 6,160,427. Filed September 4, 1998. Issued December 12, 2000.
- Daun-Lindberg, Timothy Charles and Miller; Michael Lee. 2000. *Low Voltage High-Current Electronic Load*. U. S. patent 6,211,726. Filed June 28, 1999. Issued April 3, 2001.
- Ikeuchi, Akira and Tokuda, Naoshi. 2000. *Voltage-Current Conversion Circuit*. U. S. patent 6,166,529. Filed February 24, 2000 in U. S.. Issued December 26, 2000 in U. S.. Filed March 10, 1999 in Japan.
- Koza, John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.
- Koza, John R., and Rice, James P. 1992. *Genetic Programming: The Movie*. Cambridge, MA: MIT Press.
- Koza, John R., Bennett III, Forrest H, Andre, David, and Keane, Martin A. 1999. *Genetic Programming III: Darwinian Invention and Problem Solving*. San Francisco, CA: Morgan Kaufmann.

- Koza, John R., Bennett III, Forrest H, Andre, David, Keane, Martin A., and Brave Scott. 1999. *Genetic Programming III Videotape: Human-Competitive Machine Intelligence*. San Francisco, CA: Morgan Kaufmann.
- Koza, John R., Keane, Martin A., Streeter, Matthew J., Mydlowec, William, Yu, Jessen, and Lanza, Guido. 2003. *Genetic Programming IV. Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers. In press.
- Koza, John R., Keane, Martin A., Yu, Jessen, Bennett, Forrest H III, and Mydlowec, William. 2000. Automatic creation of human-competitive programs and controllers by means of genetic programming. *Genetic Programming and Evolvable Machines*. (1) 121 - 164.
- Lee, Sang Gug. 2001. *Low Voltage Balun Circuit*. U. S. patent 6,265,908. Filed December 15, 1999. Issued July 24, 2001.
- Philbrick, George A. 1956. *Delayed Recovery Electric Filter Network*. Filed May 18, 1951. U.S. Patent 2,730,679. Issued January 10, 1956.
- Quarles, Thomas, Newton, A. R., Pederson, D. O., and Sangiovanni-Vincentelli, A. 1994. *SPICE 3 Version 3F5 User's Manual*. Department of Electrical Engineering and Computer Science, University of California. Berkeley. March 1994.
- Zobel, Otto Julius. 1925. *Wave Filter*. Filed January 15, 1921. U.S. Patent 1,538,964. Issued May 26, 1925.