A PSPICE OPTIMAL DESIGN TOOL UTILIZING GENETIC OPTIMIZATION

Mike Elmore
BAE SYSTEMS Controls
600 Main Street
Johnson City, NY
USA

Victor Skormin and Vladimir Nikulin
Electrical and Computer Engineering Department
Watson School, Binghamton University
Binghamton, NY
USA

Abstract

A design automation approach utilizing a simulation test-bed driven by a genetic optimization procedure, intended for electric circuit design applications, is developed. A Windows-based, off-the-shelf version of Pspice, capable of accurate representation of the most complex circuits and numerical assessment of their efficiency criteria performs the test-bed function. Application of genetic optimization results in attainable optimal or close-to-optimal solutions of the design problem subject to various criteria and constraints. The GenSpice package, implementing the approach, enables the designer to utilize the hardware resources to their full potential, as well as to minimize the response time to the market requirements. A power system design optimization case study provides an illustration of the application of the resultant software tool.

Keywords

Design, Genetic Algorithms, Simulation Optimization, Mathematical Modelling, Simulation Tools

1. Introduction

Designers of electric circuitry are facing ever-increasing and fast-changing demands on size, weight, efficiency, load conditions and performance characteristics of power supplies. In many instances, existing design procedures centered on experience and the intuition of designers, could be blamed for increasing the "response time" of the manufacturers to the market demands and cause difficulties in meeting the market requirements. Consequently, a novel power electronics component undergoes a lengthy period of experimental implementation and fine-tuning. Computer-based design automation/optimization approaches have the potential for the alleviation of these difficulties, replacing the experimentation by simulation and providing the designers the means for optimal decision-making. A number of publications illustrate various concepts in the development of such approaches [1], [2], [3]. However, these attempts have not yet significantly impacted the design practices. This could be explained by the fact that some authors rely on analytical models of the circuits to be designed that, typically, are too complex to develop and implement in software. This makes the approach inflexible and thus requires a significant effort when the model has to be modified. The optimal design problem, formulated on the basis of such a mathematical model, is often too formidable for most optimization techniques and its numerical solution is dependent on the "good initial conditions", specified by the designer.

The efficiency of genetic optimization in the analog circuit design problems was first demonstrated by [4], who applied it to optimize performance characteristics, such as gain and input offset voltage, of four different operational amplifiers. The resultant software tool requires the user to inspect PSpice net lists and enter encoding information into set-up menus.

Pspice-based models have been commonly used for design verification and fine-tuning. However, the potential of PSpice has never been used to the fullest due to the limited ability of a designer to fully comprehend the simulation data and establish and implement a strategy of simulation experiments leading to the optimal solution of the design problem. At the same time, genetic optimization procedures providing reliable means for solving the most complex, multivariable constrained optimization problems with multi-modal criteria have been available in various modifications. However, their application in conjunction with computer simulators is not feasible without establishing an automated data exchange between the two computer codes.

A user-friendly and flexible approach to genetic optimization with PSpice was first shown by Heimes and Elmore [5]. Fundamentally, the authors did not rely on overly complex analytical mathematical models, utilizing the PSpice simulation environment instead. They were successful in establishing a simulation/optimization scheme in which a sequence of PSpice-based simulation experiments is conducted "under the supervision" of a genetic optimization procedure. In this scheme, every simulation experiment returns the numerical value of a particular circuit performance criterion, and the genetic optimization procedure utilizes these values and implements the special strategy to modify the design parameters and initiate a new simulation experiment. This effort, resulting in a software tool known as GenSpice, was a part of a multi-disciplinary optimization that also used Ansoft Maxwell 3D to evaluate current density in thyristors and minimize the envelope volume of the thyristor assembly.

2. Optimal Circuit Design Problem

Design of an electrical circuit implies that specific numerical values of the circuit components, primarily resistors, inductors and capacitors be chosen in order to satisfy the design specifications and inherent constraints. The design specifications may reflect the electrical efficiency of the circuit, its frequency response, the required output voltage given the input voltage, etc. When the circuit configuration is defined, a mathematical model providing the definition of the circuit characteristics as a function of the numerical values of the circuit components could be established. In the conventional design procedure such a model is utilized by the designer to validate the design decisions. For example, a model of the circuit shown in Fig. 1 may

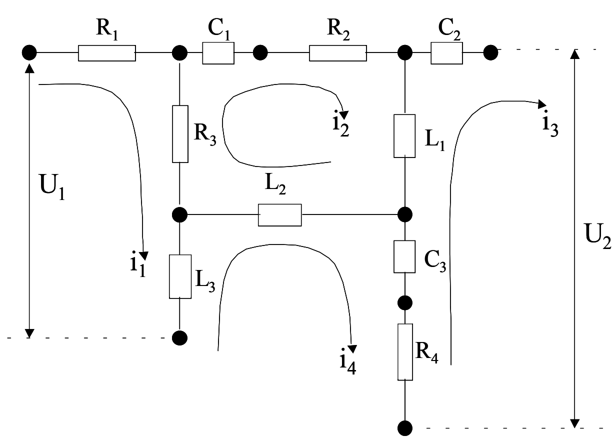

Figure 1 – Model of an Electrical Circuit

include equations providing an explicit definition of the power dissipation in the circuit, $P=P(U_1, R_i, L_j, C_k, i, j, k = 1, 2,\ldots)$, and the output voltage, $U_2=U_2\ (U_1, R_i, L_j, C_k, i, j, k = 1, 2,\ldots)$, for any combination of the numerical values of the circuit components and the input voltage U_1.

Since a typical circuit design problem has many acceptable solutions, the designer attempts to choose one favoring one of the resultant circuit characteristics, while complying with others. This leads to an optimal design problem that could be formalized on the basis of the mathematical model. For example, an attempt to maximize the electrical efficiency of the above circuit while achieving the required output voltage may be represented by the following nonlinear constrained optimization problem

$$\underset{R_i, L_j, C_k}{Min}\ P(U_1, R_i, L_j, C_k) \tag{1}$$

subject to $V_1 \leq U_2(U_1, R_i, L_j, C_k) \leq V_2$, $W_1 \leq U_1 \leq W_2$ and $R_i \geq 0$, $L_j \geq 0$, $C_k \geq 0$, for $i, j, k=1, 2, \ldots$ that most likely will be converted into a nonlinear, discontinuous, unconstrained optimization problem reflecting the method of penalty functions. One can realize that until recently optimal design of a practical power circuit would result in an unacceptable amount of analytical work, and the optimal solution would not be attainable due to the high dimension of the problem and multi-modal criterion (i.e. having many local minimum points). Consequently, the designed circuitry often does not utilize the hardware to its full potential, and the fine-tuning process results in lengthy delays.

The approach presented by the authors eliminates the need in the development of an analytical model of the circuit, utilizing common PSpice models. Such a model provides the necessary relationships between the design variables and the circuit characteristics and criteria numerically. In addition, the authors employed a genetic optimization procedure capable of finding the global optimum of a multi-modal criterion.

3. Genetic Optimization Algorithm

A mathematical model of an electrical circuit, reflecting the physical phenomena behind its operation, provides a description of the complex interrelation between various parameters of the circuit and characteristics of its performance. Such a model has a number of applications including accurate assessment of the circuit's performance, analysis of sensitivity of particular performance characteristics to design parameters, and provides a simulation test-bed for analysis of the entire electrical circuit. Moreover, the model presents a basis for the formalization of the optimal design problem of a power system, providing mathematical formulation for particular constraints and optimization criteria. Indeed, the design problem could be defined as the following nonlinear constrained optimization problem

$$Min\ C(X)/[X_1 \leq X \leq X_2,\ A \leq W(X) \leq B], \tag{2}$$

Where X is a vector of design parameters, which are customarily chosen by the power system designer, in order to assure certain performance characteristics; $X_1 \leq X \leq X_2$ is a set of conditions limiting numerical values of the design parameters reflecting the feasibility considerations; $W(X)$ is a vector-function representing operational characteristics of the system as functions of design parameters; $A \leq W(X) \leq B$ is a set of conditions presenting design specifications in terms of common figures of merit; and $C(X)$ is one of the operational characteristics of the system (or a linear combination of several operational characteristics taken with appropriate weights) designated as the design criterion.

While performance characteristics of interest, such as bandwidth, can be considered linear with respect to frequency, overall description of an electrical circuit in terms of the physical parameters chosen by the designer is given by a set of nonlinear equations. One can realize that expressions for $W(X)$ and $C(X)$ reflect laws of physics, and in combination constitute the mathematical model of the system. Any set of particular numerical values of the vector of design parameters, $X=X^*$, constitutes a solution of the design problem. Any set of particular numerical values of the vector of design parameters, $X=X^{**}$, that satisfies the feasibility conditions and the design specifications, i.e. $X_1 \leq X^{**} \leq X_2$, $A \leq P(X^{**}) \leq B$, constitutes an acceptable solution of the design problem. The criterion $C(X)$ provides a numerical measure of goodness to each acceptable solution of the design problem, facilitating the selection of the optimal solution, $X=X^{OPT}$. While any designer is intended to obtain a design solution as close to the optimal solution as possible, the "goodness" of the design is based upon his/her experience and intuition and truly optimal design is still just a matter of good intentions.

There are two major factors preventing us from finding the optimal solution of a design problem. The first one is the complexity of solving a nonlinear constrained optimization problem. The second factor is that, typically, a nonlinear optimization problem has many "local" optimal solutions; among those the global minimum should be found. The first difficulty we address by converting the original constrained optimization problem into an unconstrained optimization problem via the method of penalty functions as follows

$$Min\ L(X), \tag{3}$$

where $L(X)=C(X)+P_1(X)+P_2(X)$ is a loss function and $P_1(X)$ and $P_2(X)$ are penalty functions.

The above penalty functions can are defined as follows

$$\begin{aligned} &If\ [X_1 \leq X \leq X_2], && P_1(X)=0 \\ &If\ [X_1 > X], && P_1(X) = (X_1-X)^T R1 (X_1-X) \\ &If\ [X > X_2], && P_1(X) = (X_2-X)^T R1 (X_2-X) \end{aligned} \tag{4}$$

and

$$\begin{aligned} &If\ [A \leq W(X) \leq B], && P_2(X)=0 \\ &If\ [A > W(X)], && P_2(X)= [A-W(X)]^T R2 [A-W(X)] \\ &If\ [W(X) > B], && P_2(X)= [B-W(X)]^T R2 [B-W(X)], \end{aligned} \tag{5}$$

where ^{T} is a transpose symbol and $R1>>1$ and $R2>>1$ are weight coefficients. One can realize that any successful minimization effort would result in the "enforcing" of the constraints on the vector of design parameters X.

Proliferation of genetic optimization algorithms, possessing the advantages of known random and direct search optimization procedures, combined with the availability of high performance computers alleviated the second obstacle in the way of the formal solution of design optimization problems.

The main advantages of genetic algorithms as optimization methods are [6]:

a) robustness, broad applicability;
b) reliability;
c) good performance in high-dimensional search spaces;
d) relatively easy to develop and implement;
e) no prior knowledge about the required search topology;
f) applicable to multiple criteria optimization;
g) easily combined with other solution methods;
h) efficient use of parallel hardware.

The main disadvantages are:

a) heuristic character that does not guarantee reaching the global optimum;
b) comparatively time consuming;
c) often ineffective in fine-tuning final solutions.

The strategy of genetic optimization is analogous to biological evolution. From this perspective, its DNA determines an organism's structure and its ability to survive. An offspring of two parents inherits some characteristics from both of them, as well as having others developed as a result of parenting. These additional characteristics may improve an offspring's "fitness," which increases its chances of surviving and passing those characteristics to future generations [7], [8].

When applied to our optimal design problem, the DNA of a population member represents a vector of design parameters X defined earlier. Therefore, each DNA component is a variable of interest that may take on a finite set of values. The "fitness" of a member of population is determined by a loss function (3). Members of a population are subjected to the following operators: reproduction, mutation, and selection. Reproduction is a genetic operator that combines parents to yield offspring. Mutation is another genetic operator that subjects a member of a population to a small change, thus generating a new offspring. Mutation is intended to "explore" the area around the best offspring and parents. Selection is a genetic operator, which chooses members with the highest "fitness" to form a successive generation.

The offspring is generated by the following *parenting* strategy:

a) select two parents (points in the n-dimensional space). Let parent 1 = $(x_{11}, x_{12}, ..., x_{1n})$, and parent 2 = $(x_{21}, x_{22}, ..., x_{2n})$;
b) generate a uniform variable γ on the interval [0,1];
c) let offspring = $(x_1, x_2, ..., x_n)$;
d) offspring coordinates are found as follows $x_1 = \gamma x_{11}+(1-\gamma) x_{21}$, $x_2 = \gamma x_{12}+(1-\gamma) x_{22}$, …, $x_n = \gamma x_{1n}+(1-\gamma) x_{2n}$;
e) repeat steps a) through d) until N_o offspring are generated;
f) repeat steps a) through e) for all parent combinations selected based on parenting probability.

The *mutation* operator is implemented according to the following scheme:

a) select point $(x_1, x_2, ..., x_n)$ from the population;
b) let the new point = $(x_1', x_2', ..., x_n')$;
c) generate a uniform variable μ on the interval [-1,1];
d) coordinate $x_i' = x_i+k^* \mu$, where k is a constant gain, and $i = 1, 2, …, n$;
e) repeat steps c) and d) until all coordinates of the new point are calculated;
f) repeat steps b) through e) until N_m mutations are generated;
g) repeat steps a) through f) for the points of the generation selected according to mutation probability.

The *selection* operator is employed as follows:

h) select the first N_g points from the expanded generation;
i) find the point x_{max} with the worst "fitness" criterion L_f^{max};
j) select the next point x with fitness criterion L_f from the generation;
k) if $L_f < L_f^{max}$, replace x_{max} with x, and go to step b), if $L_f \geq L_f^{max}$, discard point x, and go to step c);
l) continue until the last member of the generation is checked for "fitness."

Fig. 2 illustrates application of a genetic algorithm to the solution of a design optimization problem. The algorithm proceeds as follows. Combinations of design parameters represented by vector X form an N-dimensional space S. Since those parameters must have bounded values defined by the design constraints, an acceptable solution will be within an N-dimensional subspace $S1$, whose boundaries are defined by the following inequality

$$X_1 \leq X \leq X_2. \quad (6)$$

The algorithm forms an initial grid within this subspace by generating K uniformly distributed points X_i, $(i=1,2,...,K)$, that represent possible solutions of the constraint optimization problem.

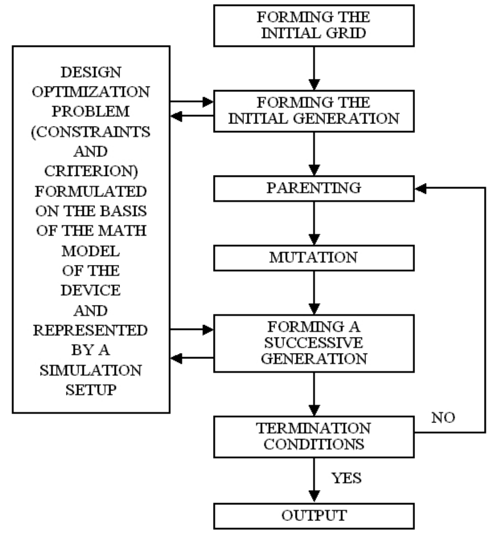

Figure 2 - Genetic Optimization-based Design

Each point X_i of the initial grid represents input parameters for the mathematical model, which is used to compute operational characteristics of the device $W(X_i)$. Based on the formulation of the optimization problem, the loss function $L(X_i)$ is calculated at each point of the initial grid. The first generation is formed by selecting $N_g<K$ points X_j, $(j=1,2,...,N_g)$ with the smallest values of $L(X)$.

The process of parenting involves producing N_o offspring per each pair of parents selected from the initial generation according to preset parenting probability. Each offspring is a new point in N-dimensional space located on the line connecting its parents X_i and X_j $(i\neq j)$, and selected in a random fashion, as described above.

Parenting is followed by the mutation stage. Each point selected from the expanded population, based on mutation probability, produces N_m mutations (points generated randomly in the immediate area). A successive generation is formed by computing the loss function for all the points and selecting N_g points with the smallest values of $L(X)$.

The processes of parenting, mutation and forming a successive generation are repeated until a termination condition is satisfied. The optimization routine produces the output in the form of a vector of design parameters X^{OPT}, which satisfies the constraints, posed by feasibility considerations and design specifications, and facilitates minimization/maximization of the design criterion $C(X)$.

4. Pspice Optimization Tool

4.1 Program Structure and Operation

The controlling program for optimization of a PSpice electrical circuit is written in Microsoft Visual BASIC 6.0. The student version of OrCAD PSpice Version 9.1 –

Web Update 1 was chosen for the PSpice simulator, because of its built-in COM interface. The Component Object Model (COM) interface consists of convenient, object-oriented programming commands that provide a "back door" into PSpice.

Fig. 3 illustrates how PSpice and GenSpice

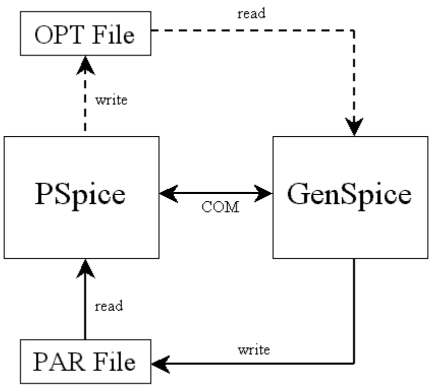

Figure 3 - PSpice and GenSpice Communication

communicate. COM acts as the "connecting 'glue' between software components, enabling unrelated software objects to connect and interact in meaningful ways [9]." COM objects are used to

a) start the PSpice simulation;

b) monitor the PSpice simulation status and detect when it has finished;

c) communicate PSpice simulation results.

At the start of the optimization GenSpice reads the names of design parameters *X* and their limiting conditions from a text file, which is indicated as the OPT file in Fig. 3. The OPT file is constructed at the same time the PSpice schematic is entered into Capture, the OrCAD schematic entry tool. After this initial operation, the optimization process shown in Fig. 2 commences. GenSpice creates the initial grid or a first generation parent population of parameters and writes the first parent to the PAR file, as shown in Fig. 3. The COM interface is used to start PSpice. PSpice reads the PAR file and simulates the circuit with the parameter values. When the simulation has concluded PSpice notifies GenSpice via COM. With COM GenSpice asks PSpice for the resultant operational characteristics *W(X)*, also known as, circuit measurements. GenSpice calculates and saves the penalty function *L(X)* for the run.

After all the original parent population has been evaluated a population of offspring are created from the parent population, using the *parenting* strategy and the *mutation* operator. GenSpice evaluates each offspring with PSpice in the same manner in which the original parent population was evaluated.

The *selection* operator is applied to the combined parent and offspring populations to create a second generation of parents. Selection is done in accordance to a *fitness* criterion.

The evaluation and selection process continues until all generations have been considered. At that time the optimum offspring X^{OPT} is declared, as the one, which produces an outcome, which is closest to the goal.

At the conclusion of the optimization the best offspring can be evaluated by writing the parameters that comprise the "genetic code" of the offspring into the PAR file and running the PSpice simulation. The result can be viewed in Probe, OrCAD's graphic display tool, and compared the GenSpice result.

4.2 User Input

Parent and offspring population sizes and the number of generations are chosen in Fig. 4, as well as, a *fitness*

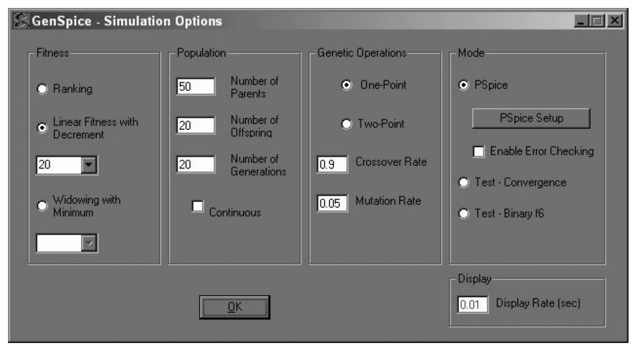

Figure 4 - Optimization Setup Menu

criterion. Several commonly accepted criteria are allowed. Genetic operations are selected. Probabilities for *parenting*, also know as *crossover*, and mutation are entered. One or two point crossover can be selected. In two-point crossover the "genetic material" in the middle of two parents are exchanged to form an offspring. The choices offered in this menu are guided by experience in optimization of circuits and guidelines found in the literature. [10]

The design criteria *C(X)* and operational characteristics *W(X)* are setup in Fig. 5. *C(X)* is also known as the Cost Function. Performance results are communicated to GenSpice through COM. The signal name is a voltage or current from the PSpice simulation. In Fig. 5 an output voltage is indicated as V(OUT). The phase margin of V(OUT) comes from a PSpice function called a Goal Function. The Goal Function, i.e., phase margin of V(OUT), is the measurement that is communicated to GenSpice via COM after each run of the simulation. Goal Functions are defined in Probe. Probe defines a standard set of commonly used Goal Functions, but the user can define new Goal Functions, as required. Only one operational characteristic *W(X)* is

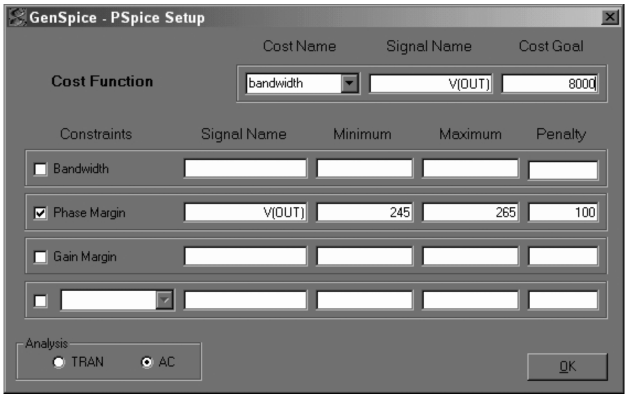

Figure 5 - Constraint and Cost Function Setup Menu

selected for this optimization in Fig. 5. It is desired to constrain the phase margin to within 245° and 265° (actually 65° and 85°, since the phase margin Goal Function does not account for the inversion in the amplifier for this circuit example). The Cost Function is simply the bandwidth of the circuit. A pull down menu offers a wide range of Cost Functions, which must be defined as Goal Functions in the PSpice simulation environment.

The input area for the parameters to be varied is entered in an Optimizer Parameters box on the Capture schematic. The name of the parameter and its range of allowable values is entered here. PSpice writes with specific instances of circuit components by placing curly brackets around the parameter value of the component. The bracketed parameters indicate to PSpice that the values of these components will be read from the PAR file.

It should be pointed out that additional electrical and non-electrical parameters can be assigned to components within PSpice, such as cost, size, power rating, weight and so on. Any of these parameters can be defined as parameters to vary on the PSpice schematic and written to the PAR file by GenSpice. GenSpice is not restricted to optimization of electrical performance only.

4.3 Additional GenSpice Features

GenSpice offers several features that assist in designing a successful optimization and the evaluation of results:

a) an HTML-based Help function that provides information on how to effectively use GenSpice and how the software works;
b) the ability to save and recall optimization setups;
c) a Pause button to halt a simulation and inspect intermediate results;
d) a plot function to display the best offspring of each generation;
e) a display to view all offspring and their fitness values after each generation;
f) a pedagogical tool that shows how two parents are selected and mutated to form an offspring;
g) and a text file that shows detailed information on each offspring of every generation.

In most cases these special features are not required to set up and run an optimization. However, they can be useful in guiding the user in the setup of an efficient simulation and providing insight into the results.

Fig. 6 shows the pedagogical feature of GenSpice

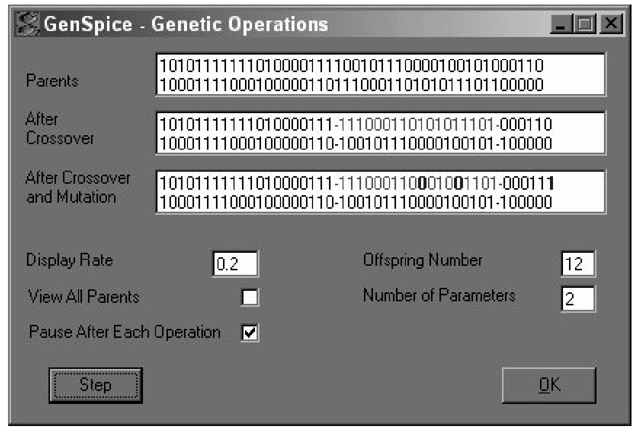

Figure 6 - Genetic Operations Display

previously mentioned. Two parents with the three binary encoded circuit parameters are shown before and after *crossover* and after both *crossover* and *mutation*. The two-point crossover segment is isolated with dashed lines and the mutated encoded circuit parameters are shown in bold type. This illustrates the "random creation" process, which is an essential part of genetic optimization.

5. Optimization Illustration Using GenSpice

A closed-loop buck converter with input filter was chosen to illustrate the use of GenSpice. The goal of the simulation is to optimize the bandwidth of the converter, while constraining the phase margin to an acceptable value. Compensation of switching converters is often done by trial-and-error either in the laboratory or on a simulation model, such as the one shown in Fig. 7. Analytical methods are available, but the typical practicing engineer often works under a severe schedule constraint and does not have the time to learn or apply the analysis, which can be complex.

The buck model is implemented with a PWM switch [11] and low-pass output filter. The input filter is included, because it has a significant effect on the small signal dynamics of the converter. The voltage feedback loop uses a PI controller to maintain regulation and meet the design specification for bandwidth and phase margin.

Three component values in the feedback amplifier are selected as constrained parameters that are allowed to vary. Each population member consists of binary-valued

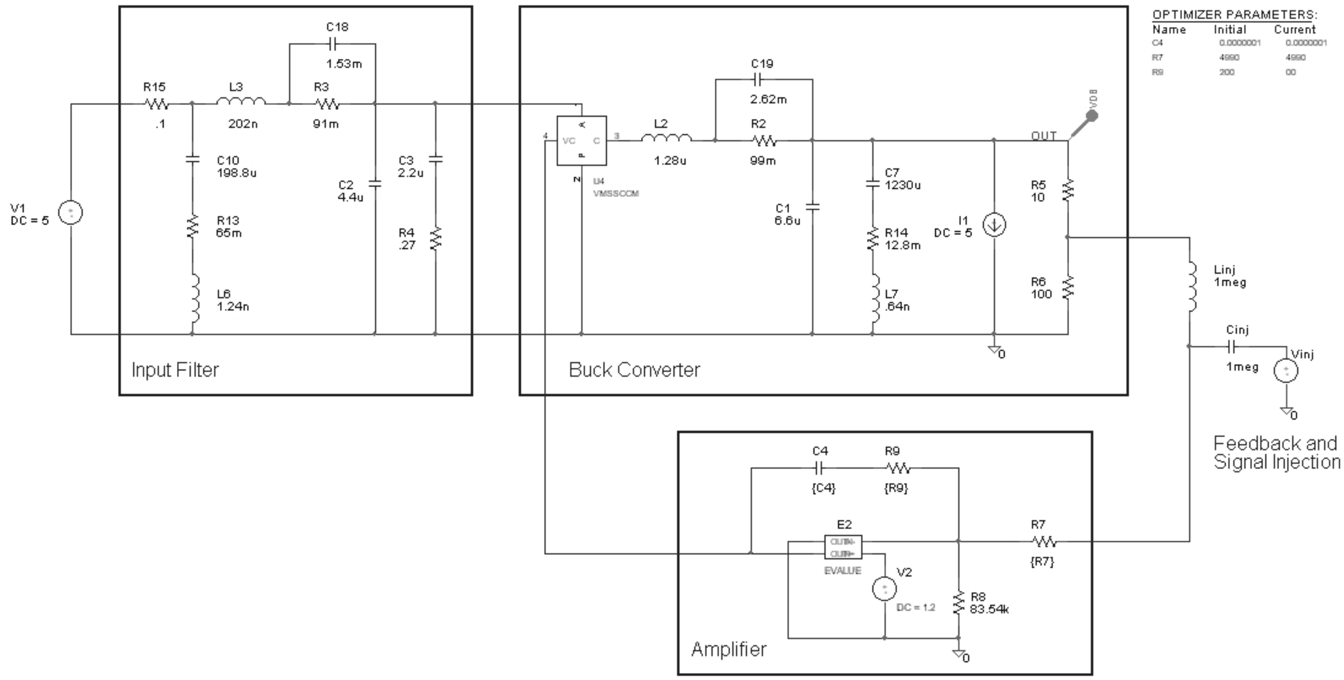

Figure 7 - PSpice Schematic of Closed-loop Buck Converter with Input Filter

concatenations of these three parameters. The design specification requires a bandwidth of 8kHz and a phase margin between 65° and 85°. The design is attempted with the trial-and-error method initially and the results are summarized in Table 1. The bandwidth specification

	Pre-optimized Result	Post-optimized Result
C4	100pF	907pF
R7	4.99kΩ	2.76kΩ
R9	200Ω	570Ω
Phase Margin	92°	68°
Bandwidth	2419 Hz	8008 Hz

Table 1 – Pre- and Post-Optimization Results

could not be met in a reasonable amount of time (< 1hour). The 2nd – order output filter of the buck and the effects of the input filter make a trail-and-error approach to compensation too difficult to achieve in a short time, unless luck intervenes. The result of the optimization using GenSpice is shown in Fig. 8. The bandwidth is within about 0.1% of 8kHz (Smallest Error box) and the phase margin has been successfully constrained to 247.84°–180° ≈ 68°. The three component values that produce this optimum level of performance are summarized in Table 1. These three component values can be written into the PAR file with a function available in the Edit pull-down menu and PSpice can be run to valid the results. The optimization took 7 minutes and 45 seconds to run.

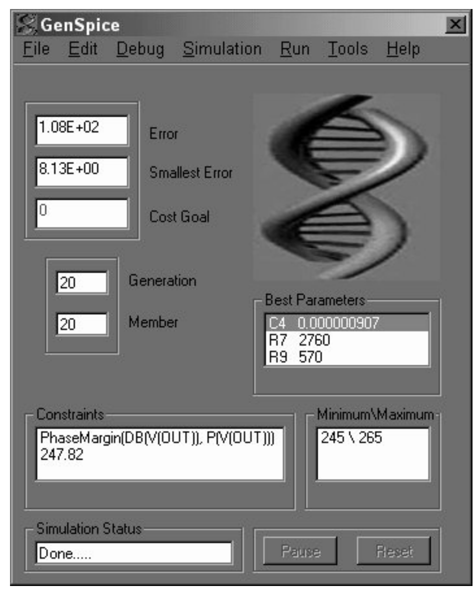

Figure 8 - GenSpice Main Menu

6. Conclusions

Performance optimization on an electrical circuit, using PSpice and a Visual BASIC program GenSpice, has been demonstrated. The simple optimization setup procedure consists of entering a schematic and design parameters into OrCAD Capture and defining operational characteristics, the design criteria and design specifications in two GenSpice menus. No custom programming is required. Newly defined Goal Functions can be reused in later optimizations, if desired. GenSpice utilizes functions built into PSpice to minimize the optimization setup.

GenSpice is a useful engineering and pedagogical tool for circuit optimization. The authors plan to offer GenSpice to interested parties at no cost. It is hoped that GenSpice will stimulate additional interest in developing low-cost, off-the-shelf optimization tools.

7. References

[1] S. Busquets-Monge, G. Soremekun, et al, Design optimization of a boost power factor correction converter using genetic algorithms, *IEEE Power Electronics Specialists Conference Record*, Dallas, USA, 2002, 1177 – 1182.

[2] J. Puhan, T. Tuma, and I. Fajfar, Optimisation methods in SPICE, a comparison, *Proc. IEEE European Conf. on Circuit Theory and Design*, Stresa, Italy, 1999, 1279 – 1282.

[3] J. Puhan, T. Tuma, Optimisation of analog circuits with SPICE 3F4, *Proc. IEEE European Conf. on Circuit Theory and Design*, Budapest, Hungary, 1997, 177 – 180.

[4] K. V. Noren and J.E.Ross, Analog circuit design using genetic algorithms," Online, Available: http://www.mrc.uidaho.edu/mrc/people/knoren/GAs/B-159_paper.PDF, Dec. 9, 2002.

[5] F. Heimes, M. Elmore, et al, Optimum design of a thyristor assembly for pulsed power applications, *18th Electric Launcher Association Meeting*, San Diego, USA, 2001.

[6] J. Biethahn and V. Nissen, Combinations of simulation and evolutionary algorithms in management science and economics, *Annals of Operations Research,* 52, 1994, 183 – 208.

[7] M.D. Vose, *The Simple Genetic Algorithm: Foundations and Theory* (Cambridge: MIT Press, 1999).

[8] R.L. Haupt and S. E. Haupt, *Practical Genetic Algorithms* (New York: John Wiley & Sons Inc., 1998).

[9] G. Eddon and H. Eddon, *Programming Components with Microsoft Visual Basic 6.0*, 2 (Redmond, Washington: Microsoft Press, 1998), pp. 15.

[10] L. Davis (Ed.), *Handbook of Genetic Algorithms* (New York: Van Nostrand Reinhold, 1991).

[11] V. Vorpérian, Simplified analysis of PWM converters using the model of the PWM switch part I: continuous conduction mode, *IEEE Transactions on Aerospace and Electronics Systems*, 26(3), 1990, 490-496.

The authors wish to thank the PSPICE team at Cadence Design Systems, PCB Systems Division for their help with understanding the programming interface between PSPICE and Visual Basic.