Analysis of Some Special Functions for a Problem of Optimization of Analog Circuits

ALEXANDER ZEMLIAK
Department of Physics and Mathematics,
Autonomous University of Puebla,
Av. San Claudio y 18 Sur, CU, Puebla, 72570,
MEXICO

Abstract: - Further development of a generalized methodology for optimizing analog circuits is proposed. This methodology is based on the theory of optimal control. We have transformed the problem of minimizing the CPU time needed to optimize the circuit into the classical problem of minimizing the function in optimal control theory. In this case, we represent the process of optimizing the analog circuit as a controlled dynamic system. To analyze the properties of such a system, we propose to use the concept of the Lyapunov function of a dynamical system. The new special functions allow us to predict the CPU time for circuit optimization by analyzing the characteristics of the initial part of the process. It has been established that for any optimization strategy, there is a correlation between the behavior of these functions and the CPU time corresponding to these strategies.

Key-Words: - Circuit optimization, time-optimal strategy, control theory, Lyapunov function.

Received: January 5, 2023. Revised: August 16, 2023. Accepted: September 22, 2023. Published: October 11, 2023.

1 Introduction

The problem of reducing the processor time required to optimize electronic circuits is one of the most important problems associated with improving the quality of design. The design process begins with an initial guess, performed by analyzing the circuit at the starting point. The system parameters are then adjusted to obtain the performance characteristics included in the specification. The parameter tuning process may be based on an optimization procedure. Thus, we conduct design through analysis instead of solving a more complex problem - the synthesis of a complex system. Mathematically, we must minimize a special objective function that models the required properties of the designed circuit. Some methods reduce the time needed for circuit analysis. These include the well-known idea of using sparse matrix methods, [1], [2], and decomposition methods, [3], [4], [5]. Iterative methods, [6], employ decomposition at a nonlinear level. Optimization methods also have a very strong impact on the general properties of the circuit design process and CPU time. Methods for analog circuit optimization can be classified into two groups: deterministic optimization algorithms and stochastic search algorithms. Deterministic optimization methods have been developed for different applied problems. Advances in deterministic mathematical optimization methods, [7], [8], are creating promising directions for both unconstrained and constrained optimization. However, classical deterministic optimization algorithms may have several drawbacks: they may require that a good initial point be selected in the parameter space, and they may reach an unsatisfactory local minimum.

To overcome these problems, new methods have been developed recently. For example, there is a method that determines the initial point of the optimization process by centering, [9], or there are geometric programming methods, [10], that guarantee convergence to the global minimum. However, these methods require a special formulation of the calculation equations, which creates additional difficulties. In recent years, some alternative stochastic search algorithms have been developed (primarily evolutionary computation algorithms). A simulated annealing algorithm has been used successfully for global optimization, [11], [12], [13]. Methods based on evolutionary algorithms, genetic algorithms, differential evaluation, and genetic programming, [14], [15], [16], [17], [18], [19], have been developed for different applications. Genetic algorithms have been employed as optimization routines for analog circuits due to their ability to find satisfactory solutions. An evolutionary algorithm known as the particle swarm optimization technique competes well with genetic algorithms. This method has been successfully used to solve electromagnetic problems and optimize microwave systems, [20], [21]. The authors of stochastic circuit optimization methods state that their algorithms provide a considerable (by 1–2 orders of magnitude) gain in time compared to traditional deterministic approaches.

The deterministic and stochastic methods mentioned above are very different in their approach to the optimization procedure. However, all of these methods use Kirchhoff's laws to analyze circuits at each stage of the optimization procedure. They use the traditional approach, which is based on circuit analysis and parametric optimization, either in deterministic or stochastic form. Nevertheless, [22], [23], [24], propose another approach, which redefines the design problem by abandoning the idea of obeying Kirchhoff laws during optimization. This approach leads to a significant gain in processor time, [24]. The most general formulation of the circuit optimization problem was proposed in [25], [26]. There, the problem of analog circuit optimization is defined in terms of control theory. We believe that this approach allows us to significantly speed up deterministic optimization methods and compete with stochastic algorithms in terms of computational time. This approach provides us with a set of different optimization strategies, allowing us to search for one or more strategies with the shortest CPU time. It has been shown that the new approach allows us, in principle, to substantially reduce the CPU time for circuit optimization. This occurs due to the fact that the framework of the generalized methodology contains a practically unlimited number of strategies. This, in turn, allows us to control the optimization process by redistributing computer resources between circuit analysis and parametric optimization. The conventional optimization strategy (COS) performs circuit analysis at each step of the optimization procedure and is not optimal in terms of time. For the optimal strategy, the gain in CPU time (compared with the COS) rises when the size and complexity of the circuit increase, [25]. Developing an algorithm that will construct the best optimization strategy is the main task for the realization of the potential of this approach. In order to develop and obtain the best optimization strategies, we must identify their most significant properties. The study of qualitative and quantitative properties and characteristics of optimal (or quasi-optimal) design is the first step toward determining the necessary structure of an optimal algorithm.

2 Problem Formulation

In accordance with the conventional approach, the process of electronic circuit optimization is defined as the problem of minimizing an objective function, $C(X)$, $X \in R^N$, with constraints given by a system of the circuit´s equations based on Kirchhoff's laws. We assume that, by minimizing $C(X)$, we achieve all our design goals. A methodology that was proposed before, [25], generalizes the circuit optimization problem by introducing a special control vector $U = (u_1, u_2, ..., u_m)$ and a special generalized objective function $F(X,U)$.

The electronic circuit design process can be defined, in accordance with [26], as the problem of minimizing the generalized objective function $F(X,U)$ based on the vector equation (1) with the constraints (2). The mathematical model of the electronic circuit represents the main constraints of the optimization problem.

$$X^{s+1} = X^s + t_s \cdot H^s, \tag{1}$$

$$\left(1-u_j\right) g_j(X) = 0,\ j = 1,2,\ldots,M, \tag{2}$$

where $X = (X', X'')$, $X' \in R^K$, is the vector of independent variables, $X'' \in R^M$ is the vector of dependent variables, M is the number of the circuit's dependent variables, K is the number of independent variables, N is the total number of variables (N=K+M), and t_s is an iteration parameter. The equation (1) describes a two-step minimization procedure, and the function $H \equiv H(X,U)$ determines the direction in which the generalized objective function $F(X,U)$ decreases. The functions $g_j(X)$ for all j define the equations of the circuit model. The components of control vector U are the set of control functions: $U = (u_1, u_2, \ldots, u_m)$, where $u_j \in \Omega$, $\Omega = \{0;1\}$. The generalized objective function $F(X,U)$ can be defined, for example, as follows:

$$F(X,U) = C(X) + \psi(X,U), \tag{3}$$

where $C(X)$ is a non-negative objective function and $\psi(X,U)$ is a penalty function. The structure of the penalty function must potentially include all the equations of the system (2) and can be defined, for example, as follows:

$$\psi(X,U)=\frac{1}{\varepsilon}\sum_{j=1}^{M}u_j\cdot g_j^2(X), \qquad (4)$$

where ε is an additional coefficient used to adapt the penalty function. In our context, ε equals 1.

Such a definition of the circuit optimization problem allows us to redistribute the computation time between problems (1) and (2). A control function u_j has the following meaning: if $u_j=0$, the *j*th equation is present in the system (2) and the term $g_j^2(X)$ is removed from the equation (4); and, conversely, if $u_j=1$, the *j*th equation is removed from the system (2) and the term $g_j^2(X)$ is present in the equation (4). The vector *U* is this methodology's main tool: it controls the dynamic process of minimizing the objective function $C(X)$ in the minimum time possible. This definition allows us to express the problem of searching for the optimal strategy as the typical problem of minimizing a function, where the function is the CPU time. When defining the optimization process as a dynamical system (in terms of optimal control theory), a standard approach is to use differential equations, in continuous form. We can rewrite the main system of the optimization procedure (1) in continuous form as the following system of differential equations:

$$\frac{dx_i}{dt}=f_i(X,U), \quad i=1,...,N. \qquad (5)$$

Together with equations (2), (3), and (4), this system specifies the continuous form of the optimization process. The structure of the functions $f_i(X,U)$ is defined by a concrete optimization method. For example, for the gradient method, it takes the following form:

$$f_i(X,U)=-\frac{\delta}{\delta x_i}F(X,U),\ i=1,2,\dots,K, \qquad (6)$$

$$f_i(X,U)=-u_{i-K}\frac{\delta}{\delta x_i}F(X,U)+\frac{(1-u_{i-K})}{dt}\cdot\left[-x_i^{'}+\eta_i(X)\right], \qquad (6')$$
$$i=K+1,K+2,\dots,N,$$

where the operator $\delta/\delta x_i$ is defined as $\frac{\delta}{\delta x_i}\varphi(X)=\frac{\partial\varphi(X)}{\partial x_i}+\sum_{p=K+1}^{K+M}\frac{\partial\varphi(X)}{\partial x_p}\frac{\partial x_p}{\partial x_i}$ and determines the application of the gradient method for a complex function that has both independent and dependent variables, $x_i^{'}$ equals $x_i(t-dt)$; and $\eta_i(X)$ is the implicit function ($x_i=\eta_i(X)$) determined by the system (2). The components u_j of the control vector *U* play the role of control functions. In general, these functions depend on time. A control function u_j has the following meaning: the *j*th equation is present in the system (2), and the term $g_j^2(X)$ is removed from the equation (4) when u_j=0; and, conversely, the *j*th equation is removed from the system (2) and the term $g_j^2(X)$ is present in the equation (4) when u_j=1.

By using formulas (2) to (6), we formulate the circuit optimization process as a controllable process or a controllable dynamical system. The optimization process is now expressed as a typical problem for a controllable dynamical system. We must minimize the time of the optimization process, which means that we must minimize the transitional time of the dynamical system. The termination of the optimization process corresponds to a stationary state of the dynamical system after the transient process. Here, the control vector *U* is the main tool. If we formulate the problem in this way, the most complex task is the search for the behavior of the control functions u_j during the optimization process. Since the control functions u_j are piecewise continuous, the functions $f_i(X,U)$ are also piecewise continuous. To minimize the total CPU time, we must find the optimal behavior of the control functions u_j during the optimization process.

In the paper, [27], the effect of the additional acceleration of the process of optimization is investigated. This effect is connected to the possibility of the emergence of the sliding mode of a dynamic system, which is similar to the mode described in, [28]. This effect can significantly reduce the computing time for circuit optimization.

The problem for the system (5) with the non-continuous or non-smooth functions (6) and (6') can be solved most correctly using Pontryagin's maximum principle, [29]. Unfortunately, its application is limited to linear systems, and in the case of non-linear dynamical systems such as designing processes, its application is limited to low-dimensional problems. We must propose an alternative to using the maximum principle. To do this, we must obtain correlations between the CPU time and the characteristics of the circuit optimization process. Then we can estimate the CPU time of the optimization process by examining some of the special functions defined for this process.

We believe that the Lyapunov function of a dynamical system, which is one of the main elements of the theory of dynamical systems, can also be used to analyze the circuit optimization process. Therefore, the use of an approach based on the concept of the Lyapunov function of a dynamical system looks promising.

When choosing the Lyapunov function, there is a certain degree of freedom because the function does not have a unique form. Let us express the Lyapunov function of the circuit optimization process as follows:

$$V(X,U)=[F(X,U)]^r, \quad (7)$$

$$V(X,U)=\sum_i\left(\frac{\partial F(X,U)}{\partial x_i}\right)^2, \quad (8)$$

where r is a positive parameter. It is well known that a Lyapunov function can be defined in various forms, and the formulas (7) and (8) are two possible ways of expressing this type of function. Under specific additional conditions, both of these formulas define a Lyapunov function that has standard properties. Indeed, let us denote the vector $A=(a_1,a_2,...,a_N)$ as the final (stationary) point of the optimization procedure (i.e. the result of the circuit optimization process). The point A is the solution to the circuit optimization problem. Let us define another vector, Y, as follows: Y=X-A. The function V, as given by (7) or (8), is a piecewise continuous function whose first partial derivatives are also piecewise continuous. In addition, V satisfies the three main properties of the Lyapunov function: (1) $V(Y)$>0, (2) $V(0)$=0 and (3) $V(Y)\to\infty$ as $\|Y\|\to\infty$. This means that we can study the stability of the equilibrium position (the point Y=0) using Lyapunov's theorem. On the other hand, the solution to the problem (i.e. the point $A=(a_1,a_2,...,a_N)$) becomes known only at the end of the optimization process. Furthermore, it would be interesting to study the stability of the process during the optimization procedure. This is the reason why the formulas (7) and (8) do not explicitly depend on point A and can be conveniently used to analyze stability. Meanwhile, in these formulas, V also depends on the control vector U. Indeed, we can see that the value of the function $V(X,U)$ in (7) equals zero at the final point of the optimization process if the objective function of this process, $C(X)$, equals zero at that point as well. Since the function $C(X)$ is non-negative, the function given by equation (7) is a positive-definite function at all points distinct from the final point $A=(a_1,a_2,...,a_N)$. The function $V(X,U)$ increases when the point X moves away from the final point A. The equation (8) also defines a Lyapunov function if $\partial F/\partial x_i$=0 at the final point A and $V(A,U)$=0. On the other hand, $V(X,U)$>0 for all X. Finally, the Lyapunov function is a function of the vector U because all the coordinates x_i depend on U. We cannot prove only the third property of Lyapunov functions $\|X\|\to\infty$ because the behavior of $V(X,U)$ is unknown. However, practice has shown that $V(X,U)$ is an increasing function in a sufficiently large neighborhood of the endpoint A of the optimization.

According to the Lyapunov method, information about the stability of the trajectory is contained in the time derivative of the Lyapunov function. We believe that the stability of any optimization trajectory correlates with the derivative of the Lyapunov function of the strategy corresponding to this trajectory. By computing the time derivative of the Lyapunov function, $\dot{V}=dV/dt$, we can estimate the stability of the dynamical system. The optimization process and its corresponding trajectory are steady if this derivative is negative. On the other hand, Lyapunov's direct method only gives sufficient stability conditions, not necessary ones. This implies that, if the derivative is positive, the process can lose stability or remain stable. If the derivative $\dot{V}$ is positive at separate points of a trajectory, it does not necessarily mean that the trajectory is unstable at those points. Only when $\dot{V}$ is positive on a positive measure can we be sure that the dynamical system is unstable. If this effect exists far from the final point, the optimization process is divergent and we cannot obtain the solution on this trajectory. If that is the case, we must change the strategy or the initial point of the optimization process. If by the end of the optimization process (i.e. near the endpoint), the derivative $\dot{V}$ becomes positive, we can say that the optimization process slows down significantly. This strategy goes in the cycle and cannot provide the required accuracy. As a result, the CPU time grows substantially. The effect is well-known in practical optimization. Here, if we cannot obtain an acceptable degree of accuracy, we must change the optimization strategy or the initial point.

In this paper, the direct computation of Lyapunov function *V* is based on the formula (7), unlike in, [30], where the formula (8) is used. This means that we must select the value of the parameter *r*. A preliminary analysis shows that this value must be less than 1. To enable the study of the behavior of the Lyapunov function and its derivative $\dot{V}$ in the best possible way, the dependencies of these functions must differ considerably for different optimization strategies. In our case, we obtain the best separation of the curves for the functions *V(X,U)* and $\dot{V}$ for different optimization strategies when *r* equals 0.5 (i.e. $V(X,U)=\sqrt{F(X,U)}$). Having carried out a detailed analysis of the behavior of the Lyapunov function and its derivative for different optimization strategies, we can choose promising strategies and discard unsuccessful ones. This kind of analysis also allows one to qualitatively determine how the processor time depends on the Lyapunov function and its derivative since both of these functions are important characteristics of the optimization process.

3 Numerical Results and Discussion

To demonstrate the strengths of the proposed approach, let's implement it in several examples. The circuit optimization process is implemented in the constant current mode. The static model of the Ebers-Moll transistor, [31], was used. The objective function is defined as the sum of the squared differences between the given and current voltages for some circuit nodes. Depending on the example, the final value of the objective function is defined as 10^{-8}–10^{-10}. As a test method for circuit optimization, we use the gradient method. However, as shown in [25], we can include any optimization method in the presented methodology.

The obtained numerical results depend on several factors: (1) the initial point of the optimization procedure in the parameter space, (2) the chosen “length” of the integration step (in the case where it is constant), and (3) the chosen method of the automatic adaptation of steps. Thus, not only can numerical results differ from optimization strategy to optimization strategy, but so can the ratio between them, [26]. For a given initial point of the optimization process, we can obtain one set of results for a collection of strategies; however, for a different initial point, we can obtain a different set of results for the same collection of strategies. We can draw the same conclusion with respect to changing the integration step. Nevertheless, among the many different optimization strategies, there are always strategies that carry out circuit optimization in significantly shorter CPU times than the traditional strategy. At the same time, there are strategies that are slower than the traditional ones. However, there is a certain invariant—the relation between the CPU time and the properties of the Lyapunov function—which can be used as a basis for the search for the structure of the best optimization algorithm for any initial point of the process and for any integration step.

Below, we analyze the properties of different optimization strategies by analyzing the behavior of the Lyapunov function’s derivative during the optimization process.

First of all, we wish to conceptually prove the relation between the CPU time and the properties of the Lyapunov function of the optimization process. In, [30], a hypothesis to the effect that there was a correlation between the CPU time and the properties of the Lyapunov function was proposed. We must demonstrate this link explicitly.

If we compute the time derivative of the Lyapunov function, $\dot{V}$, directly, we can see that this derivative is negative at the initial optimization stage for all trajectories (i.e. all possible strategies and their trajectories are stable at the beginning). At the same time, when the current point of a trajectory reaches somewhere in the neighborhood of the stationary point $A=(a_1,a_2,...,a_N)$, the derivative of the Lyapunov function becomes positive and the current optimization strategy loses its stability.

The analysis provided in, [30], allows us to conclude that the behavior of the Lyapunov function of the optimization process and its derivative is a rather informative source during the determination of the optimization strategy that minimizes the CPU time. However, we would also like to obtain some quantitative characteristics for the behavior of the Lyapunov function and its derivative.

The electronic circuits are optimized on the basis of the continuous form of the circuit optimization process (2)–(5). The iteration parameter t_s is constant but selected separately for each strategy. On the one hand, we must minimize the number of integration steps; on the other hand, we must obtain smooth dependencies for the Lyapunov function to adequately compute its derivative. This leads to a proportional increase in the number of integration steps and the CPU time for all the strategies. However, it allows us to obtain continuous and smooth dependences for the derivative of the Lyapunov function. We want to obtain an interrelation between relative CPU time and the behavior of the derivative of the corresponding Lyapunov function.

According to the theory of Lyapunov's direct method, the CPU time and the information on the stability of a trajectory are related to the time derivative of the Lyapunov function. In terms of control theory, the problem of constructing an optimization algorithm that minimizes the CPU time can be formulated as the problem of searching for a transient process of a dynamical system that minimizes the transitional time. In this search, the main tool is the control vector U, which allows us to change the structure of the functions $f_i(X,U)$ and, according to, [32], [33], to thereby modify the transitional time. To this end, we must ensure the maximum decrease rate of the Lyapunov function (i.e. the maximum absolute value of the derivative $\dot{V}$).

Let us define a more informative function, namely, the relative time derivative of the Lyapunov function $W = \dot{V}/V$. This function allows us to compare different strategies in terms of the behavior of the function *W(t)* and select the most promising ones from the point of view of the shortest CPU time.

The examples below show quantitative relationships that explain the correlation between processor time and the behavior of the *W* function. The optimization process presented below is implemented based on the continuous form given by equation (5). To present the behavior analysis of the functions *V(t)* and *W(t)*, we use test cases of passive and active nonlinear circuits. This allows us to explain the main features of the behavior of the function *W(t)*.

Figure 1 presents a three-node nonlinear passive circuit.

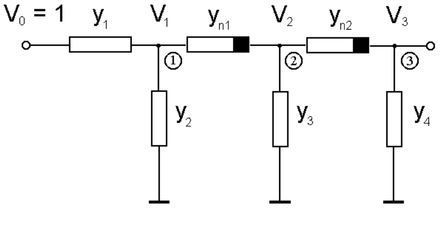

Fig. 1: Three-node nonlinear passive circuit

Here the circuit model (2) consists of three equations (*M*=3), and the control vector *U* consists of three components as well: $U = (u_1, u_2, u_3)$. The structural basis consists of eight different optimization strategies. The nonlinear elements are given as follows: $y_{n1} = a_{n1} + b_{n1} \cdot (V_1 - V_2)^2$ and $y_{n2} = a_{n2} + b_{n2} \cdot (V_2 - V_3)^2$. The vector X consists of seven components, which are set as follows: $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5 = V_1$, $x_6 = V_2$ and $x_7 = V_3$. Having determined the components using the above formulas, we automatically obtain positive conductivity values. This removes the issue of positive definiteness for each resistance and conductance and allows optimization over the entire space of values of these variables without any restrictions.

The circuit is a voltage divider, and the objective function can be defined by the formula $C(X) = (V_3 - V_{30})^2$, where V_{30} is the required value of the output voltage V_3, which must be obtained during the optimization process.

Table 1 presents the analysis of the results of the optimization process for the eight strategies that form the complete structural basis.

Table 1. Complete set of strategies of the structural basis for the three-node nonlinear circuit.

N	Control vector	Iterations number	Total processor time (sec)
1	(0 0 0)	518168	39.723
2	(0 0 1)	1250176	45.522
3	(0 1 0)	689354	22.855
4	(0 1 1)	220500	4.511
5	(1 0 0)	157426	5.720
6	(1 0 1)	401025	12.852
7	(1 1 0)	211908	6.091
8	(1 1 1)	444405	5.611

The first line of the table corresponds to the COS when $U = (0,0,0)$. For each strategy, we compute the CPU time that corresponds to the final point that minimizes the function *V*.

Figure 2 displays the behavior of the functions V and W, which are the normalized versions of the functions *V(t)* and *W(t)*. This normalization is carried out as follows: V=$V(t)/V_{max}$ and W=$W(t)/W_{max}$, where V_{max} and W_{max} are the maximum values of the functions *V(t)* and *W(t),* respectively, in the entire structural basis. We do similar normalization for all examples.

Our main goal is to define a main criterion that would allow us to compare different strategies and choose the fastest one when optimizing without directly calculating CPU time.

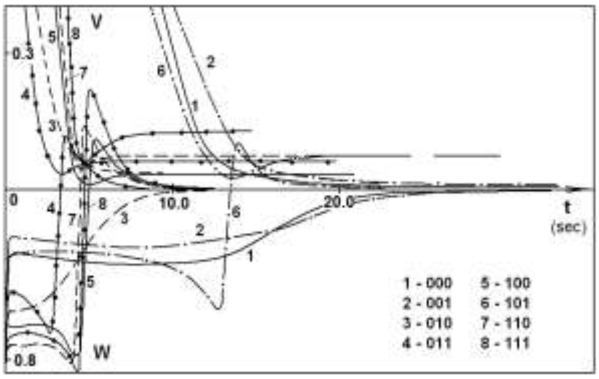

Fig. 2: Behaviour of the functions V and W for eight strategies during the optimization process, for a three-node nonlinear passive circuit

As can be seen from Figure 2, the functions V and W provide a comprehensive explanation of the characteristics of the optimization process. First of all, we can conclude that the Lyapunov function decreases in inverse proportion to processor time. The minimum value of the Lyapunov function corresponding to maximum accuracy is approximately the same for all strategies. Figure 2 shows that the Lyapunov function increases slightly after reaching its minimum value. This small increase corresponds to a small positive value of the derivative of the Lyapunov function. Later, this derivative approaches zero and the Lyapunov function reaches a constant value.

We can see the correlation between the total CPU time for a particular strategy and the behavior of the W function corresponding to that strategy. The greater the absolute value of the function W at the initial stage of the optimization process, the faster the Lyapunov function decreases. Let us recall that according to formulas (7) and (8), the Lyapunov function determines the distance to the endpoint of the optimization process. In this case, the total processor time will also be minimal.

Three groups of structural framework strategies can be distinguished. The first group includes strategies 4, 5, 7, and 8, which have the largest absolute value of the W function at the initial stage of the optimization process. At the same time, these strategies have the shortest CPU time. The second group includes strategies 1 and 2, which have the minimum absolute value of the function W. It is these strategies that have the most CPU time. The third group contains strategies 3 and 6, whose CPU time is intermediate. For these strategies, the behavior of the function W is also intermediate. Therefore, we can state that there is a correlation between the CPU time and the behavior of the function W.

The second example corresponds to the optimization of the one-stage transistor amplifier in Figure 3.

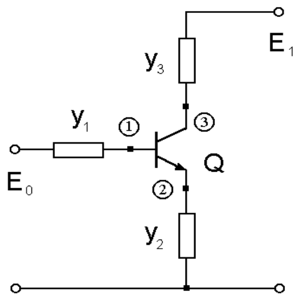

Fig. 3: One-stage transistor amplifier

The one-stage transistor amplifier has three independent variables, admittance y_1, y_2, y_3 (*K=3*), and three dependent variables, nodal voltages V_1, V_2, V_3 (*M=3*). The vector *X* is defined as $X = (x_1, x_2, x_3, x_4, x_5, x_6)$, where $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4 = V_1$, $x_5 = V_2$, $x_6 = V_3$. The objective function of the optimization procedure is determined by means of the formula $C(X) = (V_{EB} - k_1)^2 + (V_{CB} - k_2)^2$, where V_{EB} and V_{CB} are the current values of voltages in transistor junctions and k_1 and k_2 are the before-defined values of voltages on transistor junctions. The structural basis of optimization strategies has eight different strategies. The control vector consists of three control functions: $U = (u_1, u_2, u_3)$.

Let us define the voltages on transistor junctions as k_1= -0.35 V and k_2=5.9 V. The start point of the optimization process includes values for three admittances and three nodal voltages. The initial point of the vector *X* is defined as X^0=(0.05,0.1,0.1,1,1,2). The final point of the vector *X* is obtained after the process of optimization and it gives a real solution X^f=(0.0092, 0.0833, 0.0625, 1.26, 0.91, 7.16), that corresponds to the admittances (resistances): y_1=0.0847$_{10}$-3 (R_1=11.8$_{10}$3 Ω), y_2=6.94$_{10}$-3 (R_2=144 Ω), y_3=3.91$_{10}$-3 (R_3=256 Ω). This gives us an amplification coefficient of 60 or higher.

All strategies of the structural basis give the same final point of the vector *X*. Table 2 presents the results of the analysis for all the optimization strategies of the structural basis for the one-stage amplifier.

Table 2. Complete set of strategies of the structural basis for one-stage transistor amplifier.

N	Control vector	Iterations number	Total design time (sec)
1	(0 0 0)	7683758	518.22
2	(0 0 1)	45900	2.42
3	(0 1 0)	1151505	60.14
4	(0 1 1)	47464	2.53
5	(1 0 0)	109784	5.87
6	(1 0 1)	4753	0.25
7	(1 1 0)	303579	14.83
8	(1 1 1)	4940	0.08

Figure 4 presents the behavior of the functions V and W for all the strategies of this basis.

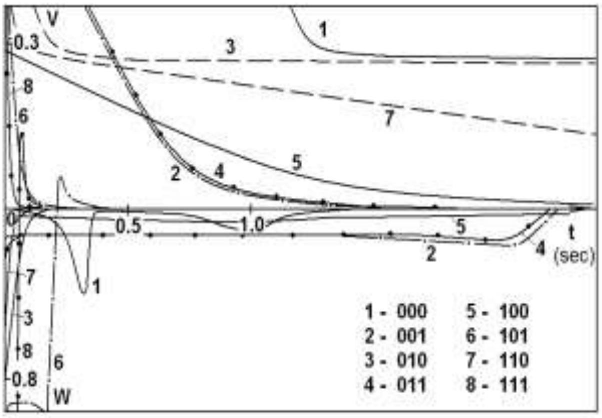

Fig. 4: Behaviour of the functions V and W for the complete structural basis during the optimization process, for a one-stage transistor amplifier

We can state that the two best strategies (8 and 6) minimize the CPU time (0.08 sec and 0.25 sec, respectively). At the same time, these strategies have the largest absolute value of the function W in the initial part of the optimization process. Conversely, strategies 1, 3, and 7 have the longest CPU time and small values of the function W in the initial part of the optimization process, while the function V has large values for these strategies. Therefore, we can state that there is a correlation between the behavior of the function W and the CPU time.

Another example corresponds to the optimization of the two-stage transistor amplifier in Figure 5.

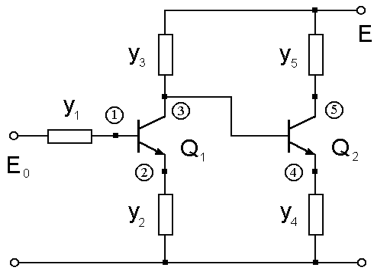

Fig. 5: Two-stage transistor amplifier

This circuit is defined by five independent variables, admittance y_1, y_2, y_3, y_4, y_5 (*K*=5), and five dependent variables, nodal voltages V_1, V_2, V_3, V_4, V_5 (*M*=5). The vector *X* is defined as $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$, where $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5^2 = y_5$, $x_6 = V_1$, $x_7 = V_2$, $x_8 = V_3$, $x_9 = V_4$, $x_{10} = V_5$. The control vector includes five control functions: $U=(u_1, u_2, u_3, u_4, u_5)$. The objective function of the optimization procedure is determined by means of the formula, $C(X) = \sum_{i=1}^{2} \left[(V_{EBi} - k_{1i})^2 + (V_{CBi} - k_{2i})^2 \right]$, where V_{EBi} and V_{CBi} are the current values of transistor junctions' voltages, k_{1i} and k_{2i} are the before-defined values of transistor junctions' voltages. These parameters are defined as: k_{11}= -0.3 V, k_{21}=5.5 V, k_{12}= -0.35 V, and k_{22}=6.2 V. The initial point of the vector *X* is defined as X^0=(0.05, 0.1, 0.1, 0.05, 0.1, 1, 1, 2, 1, 2). The final point of the vector *X* is obtained after optimization and it gives the solution: X^f=(0.0102, 0.0812, 0.0615, 0.094, 0.086, 1.2, 0.9, 6.7, 6.35, 12.9).

The structural basis for *M*=5 includes 32 different strategies of optimization. Table 3 and Figure 6 depict the results of the analysis of the optimization process for the two-stage transistor amplifier.

Table 3. Some strategies of structural basis for two-stage transistor amplifier.

N	Control vector	Iterations number	Total design time (sec)
1	(0 0 0 0 0)	165962	299.564
2	(0 0 0 0 1)	337487	737.551
3	(0 0 1 0 0)	44118	68.874
4	(0 0 1 0 1)	14941	19.061
5	(0 0 1 1 1)	21971	25.032
6	(0 1 1 0 1)	3106	3.572
7	(1 0 1 0 1)	5485	10.157
8	(1 0 1 1 1)	4544	4.560
9	(1 1 1 0 1)	2668	1.323
10	(1 1 1 1 1)	19330	1.669

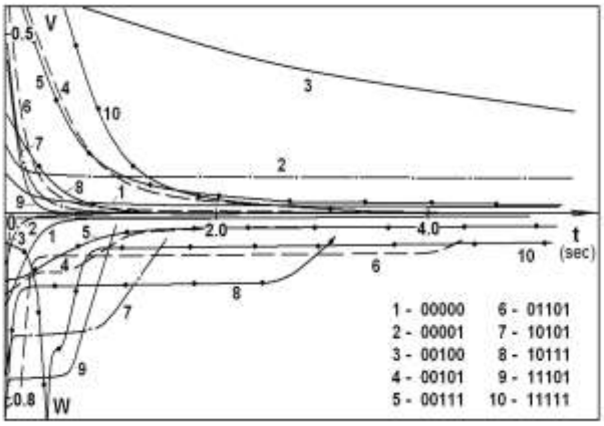

Fig. 6: Behaviour of the functions V and W for some strategies during the optimization process, for a two-stage transistor amplifier

Strategy 9 with a control vector (11101) is the best strategy among all of them. This strategy has a time gain of 227 times in comparison with the COS (strategy 1). Figure 6 shows the behavior of functions V and W for all optimization strategies from Table 3. As in the previous example, we observe a correlation between the CPU time and the behavior of the function *W(t)* at the initial part of the optimization process. We can identify three groups of strategies. The strategies in the first group have short CPU times. These are strategies 6, 7, 8 and 9. They have large absolute values of the function W during a long time interval.

Conversely, the strategies in the second group (strategies 1 and 2) have a high CPU time. On the other hand, these strategies have small absolute values of the function W in the initial part of the optimization process and over the full-time interval. The strategies in the third group (strategies 5 and 10) have intermediate values of the function W compared to the first two groups and have intermediate CPU times.

It can be stated that a large absolute value of the function *W(t)* at the initial stage of the optimization process leads to a reduction in computation time. On the other hand, the function *W(t)* is a normalized derivative. For this reason, it is very sensitive. There are some intersections between the curves corresponding to other strategies. To improve the quality of analysis, we propose to define an integral (9) of the function *W(t)* to obtain more pure correlations between the CPU time and the properties of the Lyapunov function.

$$S(t) = -\int_0^t W(t)dt = -\int_0^t \frac{dV}{dt}\cdot\frac{1}{V}dt = -\int_{V(0)}^{V(t)} \frac{dV}{V} = -\ln\left|\frac{V(t)}{V(0)}\right| \quad (9)$$

The behavior of the normalized function S for all strategies of Table 3 is presented in Figure 7. It is evident that all the curves are very well regulated as in the CPU time and the absolute value of the function S. There is a strong correlation between the function S and the computing time. A strategy with less computation time has a larger value of the S function at any given time. This means that we can predict the computing time for any optimization strategy through control of the function *V(t)*. We can analyze the functions *V(t)* for the initial time interval for the different strategies, and, on the basis of this analysis, we can predict the strategies that have a minimal total CPU time.

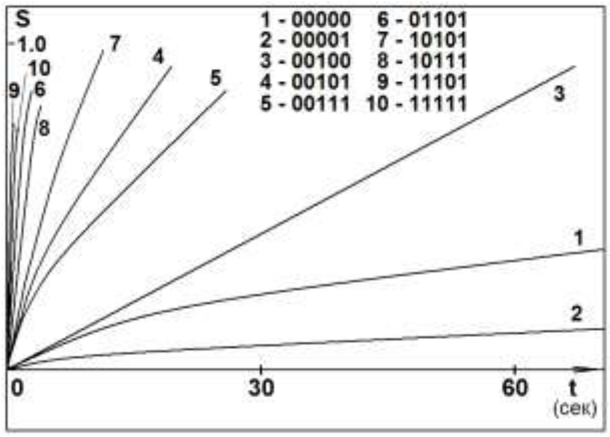

Fig. 7: Behavior of the function S for different optimization strategies, for a two-stage amplifier

The next example corresponds to the optimization of the three-stage transistor amplifier in Figure 8.

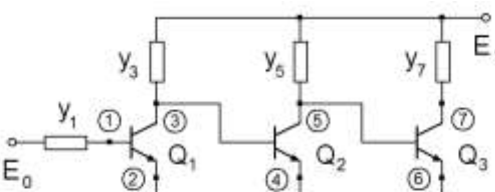

Fig. 8: Three-stage transistor amplifier

This circuit is defined by seven independent variables, admittances $y_1, y_2, y_3, y_4, y_5, y_6, y_7$ (*K=7*), and seven dependent variables, nodal voltages $V_1, V_2, V_3, V_4, V_5, V_6, V_7$ (*M=7*). The vector X is defined as $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14})$, where $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5^2 = y_5$, $x_6^2 = y_6$, $x_7^2 = y_7$, $x_8 = V_1$, $x_9 = V_2$, $x_{10} = V_3$, $x_{11} = V_4$,

$x_{12}=V_5$, $x_{13}=V_6$, $x_{14}=V_7$. The total structural basis contains 128 different design strategies. The control vector includes seven control functions: $U=(u_1,u_2,u_3,u_4,u_5,u_6,u_7)$. The objective function of the optimization procedure was determined by means of the formula, $C(X)=\sum_{i=1}^{3}\left[(V_{EBi}-k_{1i})^2+(V_{CBi}-k_{2i})^2\right]$, where V_{EBi} and V_{CBi} are the current values of transistor junctions' voltages and k_{1i} and k_{2i} are the before-defined values of transistor junctions' voltages. These values were defined as: k_{11}= -0.3V, k_{21}=5.4V, k_{12}= -0.3V, k_{22}=6.5V, k_{13}= -0.35V, k_{23}=6.6 V.

The initial point of the vector X is defined as X^0=(0.05, 0.1, 0.1, 0.05, 0.1, 0.05, 0.1, 1, 1, 2, 1, 2, 1, 2). The final point of the vector X is obtained after the process of optimization, and it gives the solution: X^f=(0.0102, 0.0812, 0.0615, 0.094, 0.086, 0.234, 0.206, 1.1, 0.8, 6.5, 6.2, 13.0, 12.65, 19.6). It is clear that all possible strategies give the same final point of the vector X. The structural basis for M=7 includes 128 different strategies of optimization. The results of the analysis of some strategies of the structural basis are given in Table 4.

Table 4. Some strategies of structural basis for three-stage transistor amplifier.

N	Control vector	Iterations number	Total design time (sec)
1	(0 0 0 0 0 0 0)	2354289	420.181
2	(0 0 1 0 1 0 1)	410889	217.150
3	(0 1 1 1 0 0 0)	375433	172.014
4	(1 0 1 0 1 0 1)	102510	43.211
5	(1 0 1 1 1 0 1)	147541	52.440
6	(1 0 1 1 1 1 1)	38751	12.753
7	(1 1 1 0 1 1 1)	43387	11.891
8	(1 1 1 1 1 0 0)	185085	140.624
9	(1 1 1 1 1 1 0)	147094	76.131
10	(1 1 1 1 1 1 1)	52651	4.782

All presented strategies have less computer time than the COS. Strategy 10 is the best one among all of them. This strategy has a time gain of 88 times in comparison with the COS.

The corresponding dependences of the function S during the optimization process are presented in Figure 9. This example, as well as all the previous ones, shows an unambiguous correlation between the behavior of the function S and the total CPU time required to optimize the circuit.

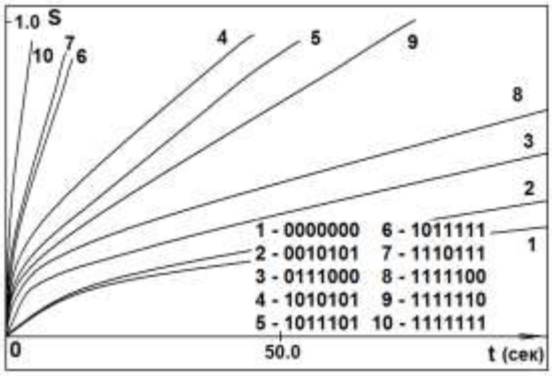

Fig. 9: Behavior of the function S for different optimization strategies, for a three-stage amplifier

The last example corresponds to the optimization of the amplifier with feedback which is shown in Figure 10.

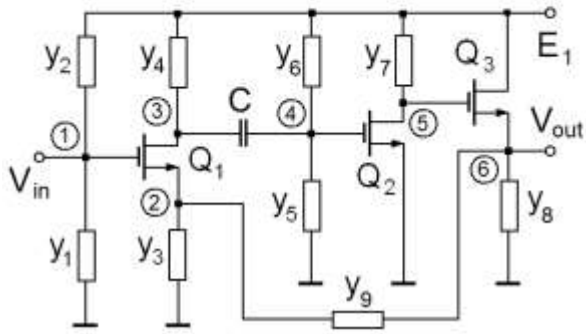

Fig. 10: Amplifier with feedback

The circuit contains six nodes. There are nine independent variables $y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8,y_9$ (K=9) and six dependent variables, V_1,V_2,V_3,V_4,V_5,V_6 (M=6). The control vector consists of eight components $U=(u_1,u_2,u_3,u_4,u_5,u_6)$, and the structural basis includes 64 strategies. The vector X includes 15 components $X=(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14},x_{15})$, where $x_1^2=y_1$, $x_2^2=y_2$, $x_3^2=y_3$, $x_4^2=y_4$, $x_5^2=y_5$, $x_6^2=y_6$, $x_7^2=y_7$, $x_8^2=y_8$, $x_9^2=y_9$, $x_{10}=V_1$, $x_{11}=V_2$, $x_{12}=V_3$, $x_{13}=V_4$, $x_{14}=V_5$, $x_{15}=V_6$. The objective function of the optimization procedure is determined by means of the formula $C(X)=(V_1-V_2-k_1)^2+(V_3-V_2-k_2)^2+(V_4-k_3)^2+(V_5-k_4)^2+(V_5-V_6-k_5)^2+(E_1-V_6-k_6)^2$, where k_1, k_2, k_3, k_4 , k_5, and k_6 are the before-defined values of GS and DS voltages for Q_1, Q_2, and Q_3. These parameters were defined as: k_1=-1.8 V, k_2=6.8 V, k_3=-2.0 V, k_4=6.8 V, k_5=-1.5 V, k_6=6.0 V.

The initial point of the vector X is defined as X^0=(0.01, 0.01, 0.01, 0.05, 0.01, 0.01, 0.01, 0.05, 0.01, 2, 1, 3, 2, 3, 2, 1). The final point of the vector X was obtained after the process of optimization X^f=(0.00816, 0.00447, 0.0224, 0.01, 0.0091, 0.00447, 0.01, 0.0224, 0.00557, 5.8, 3.8, 10.6, 1.8, 6.6, 5.1), which corresponds to the following values: $y_1=0.06658_{10}$-3 ($R_1=15.02_{10}3\,\Omega$), $y_2=0.02_{10}$-3 ($R_2=50_{10}3\,\Omega$), $y_3=0.502_{10}$-3 ($R_3=1.99_{10}3\,\Omega$), $y_4=0.1_{10}$-3, ($R_3=10.0_{10}3\,\Omega$), $y_5=0.083_{10}$-3 ($R_5=12.05_{10}3\,\Omega$), $y_6=0.02_{10}$-3 ($R_6=50_{10}3\,\Omega$), $y_7=0.1_{10}$-3 ($R_7=10_{10}3\,\Omega$), $y_8=0.5012_{10}$-3 ($R_8=1.995_{10}3\,\Omega$), $y_9=0.031_{10}$-3 ($R_9=32.26_{10}3\,\Omega$). All presented strategies reach the same final point of the vector X.

The results of the optimization process for some strategies of the structural basis are presented in Table 5.

Table 5. Some strategies of the structural basis for an amplifier with feedback.

N	Control vector	Iterations number	Total design time (sec)
1	(0 0 0 0 0 0)	6995	83.435
2	(0 0 0 0 1 1)	250	2.117
3	(0 0 0 1 1 1)	892	4.592
4	(0 0 1 0 1 1)	210	1.388
5	(0 0 1 1 1 1)	403	1.144
6	(0 1 1 1 1 1)	158	0.332
7	(1 0 1 1 1 1)	305	0.813
8	(1 1 1 1 1 1)	527	0.991

The best strategy 7 is 251 times faster than COS. The corresponding dependencies of the function S for these strategies are shown in Figure 11.

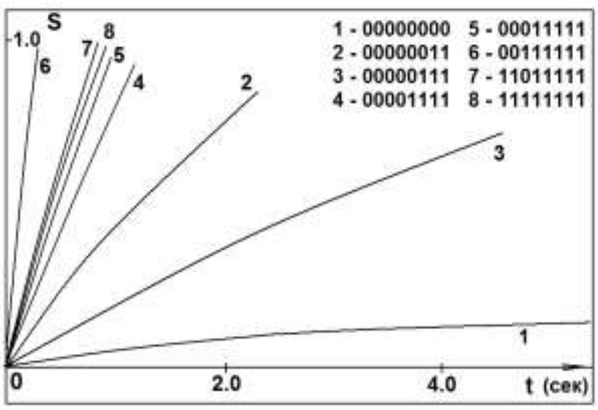

Fig. 11: Behavior of the function S for some strategies of structural basis during the optimization process, for operational amplifier

Like the preceding examples, this one demonstrates a strong correlation between the behavior of the function S and the processor time, which is necessary for circuit optimization. In addition, there is a good separation of the curves that correspond to the different functions S, and this fact significantly improves the verification.

In, [30], a hypothesis was put forward to the effect that there was a correlation between the CPU time and the properties of the Lyapunov function. We have proved the existence of this correlation. Using the generalized approach for circuit optimization proposed in this paper, we see a difference in CPU time for different optimization strategies. The detailed analysis presented in this section makes it possible to understand the root cause of this. For each optimization strategy, the CPU time is determined by the behavior of the derivative of the Lyapunov function of the optimization process. This function also estimates the comparative performance time for each optimization strategy.

Thus, it can be noted that there is a strong correlation between the processor time and the properties of the Lyapunov function. Summarizing the obtained results, we can conclude that by analyzing the behavior of the relative time derivative of the Lyapunov function of the optimization process $W = \dot{V}/V$, at the initial interval of the optimization process, it is possible to predict the total relative processor time for a given strategy. This means that we do not need to run the entire optimization process for each strategy in order to compare the total CPU optimization time for different strategies. To determine the strategy with the least processor time, it suffices to compare the behavior of the function *W(t)* or *S(t)* at the initial stage of the optimization process. Large absolute values of the *W* or *S* functions lead to a reduction in processor time. This property leads to the conclusion that the structure of the best circuit optimization algorithm should be based on the behavior of these functions.

The results obtained make it possible to reveal the main criterion for constructing an optimal or quasi-optimal circuit optimization algorithm. This criterion is the value of the derivative of the Lyapunov function. By comparing different strategies by this criterion, we can choose the best strategies at the beginning of the optimization process. In future work, this criterion should be used as a basis for recommendations on the possible structure of an optimal or quasi-optimal algorithm.

4 Conclusion

Based on the analysis presented in this paper, we can conclude that the properties of one or another circuit optimization strategy depend on the behavior of the Lyapunov function of the optimization process. A special function, the relative time derivative of the Lyapunov function, is a fairly informative source for finding strategies that minimize the processor time. We found a strong correlation between the properties of the Lyapunov function and the corresponding CPU time. The least processor time is also shown by those strategies that have the largest absolute value of the relative time derivative of the Lyapunov function in the initial section of the optimization trajectory. This property can become the basis for developing a better circuit optimization algorithm.

References:

[1] J.R. Bunch, and D.J. Rose, Eds., *Sparse Matrix Computations*, Acad. Press, N.Y., 1976.

[2] O. Osterby, and Z. Zlatev, *Direct Methods for Sparse Matrices*, Springer-Verlag, N.Y., 1983.

[3] F.F. Wu, Solution of Large-Scale Networks by Tearing, *IEEE Transactions on Circuits and Systems*, Vol.CAS-23, No.12, 1976, pp. 706-713.

[4] A. Sangiovanni-Vincentelli, L.K. Chen, and L.O. Chua, An Efficient Cluster Algorithm for Tearing Large-Scale Networks, *IEEE Trans. Circuits Syst.*, Vol.CAS-24, No.12, 1977, pp.709-717.

[5] N. Rabat, A.E. Ruehli, G.W. Mahoney, and J.J. Coleman, A Survey of Macromodeling, *IEEE International Symposium on Circuits and Systems*, 1985, pp. 139-143.

[6] A. Ruehli, A. Sangiovanni-Vincentelli, G. Rabbat, Time analysis of large-scale circuits containing one-way macromodels, *IEEE Trans. Circuits Syst*., Vol.29, 1982, pp. 185-191.

[7] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, N.Y.,1981.

[8] P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization*, Acad. Press, London,1981.

[9] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, and K. Antreich, Initial sizing of analog integrated circuits by centering within topology-given implicit specifications, *Proc. of the IEEE/ACM Int. Conf. CAD*, 2003, pp. 241-246.

[10] M. Hershenson, S. Boyd, and T. Lee, Optimal design of a CMOS op-amp via geometric programming, *IEEE Trans. CAD ICs*, Vol.20, No.1, 2001, pp.1-21.

[11] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, Optimization by simulated annealing, *Science*, Vol.220, 1983, pp.671-680.

[12] V. Delport, Parallel simulated annealing and evolutionary selection for combinatorial optimization, *Electronics Letters*, Vol.34, pp. 758-759.

[13] B. Hamma, S. Viitanen, and A. Torn, Parallel continuous simulated annealing for global optimization, *Optimization Methods and Software*, Vol.13, 2000, pp.95-116.

[14] D. Nam, Y. Seo, L. Park, C. Park, and B. Kim, Parameter optimization of an on-chip voltage reference circuit using evolutionary programming, *IEEE Trans. Evol. Comput.*, Vol.5, No.4, 2001, pp.414-421.

[15] N.F. Paulino, J. Goes, and A. Steiger-Garcao, Design methodology for optimization of analog building blocks using genetic algorithms, *Proc Symp.CAS*, 2001, pp.435-438.

[16] G. Alpaydin, S. Balkir, G. Dundar, An evolutionary approach to automatic synthesis of high performance analog integrated circuits, *IEEE Trans. Evol. Comput.*, Vol.7, No.3, 2003, pp.240-252.

[17] A. Srivastava, T. Kachru, and D. Sylvester, Low-Power-Design Space Exploration Considering Process Variation Using Robust Optimization, *IEEE Trans. CAD ICs*, Vol.26, No.1, 2007, pp.67-79.

[18] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F.V. Fernandez, Analog circuit optimization system based on hybrid evolutionary algorithms, *Integr., VLSI Jour.*, Vol.42, 2009, pp.137-148.

[19] M.L. Carneiro, P.H.P. de Carvalho, N. Deltimple, L. da C Brito, L.R.A.X. de Menezes, E. Kerherve, S.G. de Araujo, and A.S. Rocira, Doherty amplifier optimization using robust genetic algorithm and unscented transform, *Proc. Annual IEEE Northeast Workshop CAS*, 2011, pp.77-80.

[20] J. Robinson, and Y. Rahmat-Samii, Particle swarm optimization in electromagnetic, *IEEE Trans. Anten. Propag.*, Vol.52, No.2, 2004, pp.397-407.

[21] M.A. Zaman, M. Gaffar, M.M. Alam, S.A. Mamun, and M. Abdul Matin, Synthesis of antenna arrays using artificial bee colony

optimization algorithm, *Int. J Microw. Opt. Techn.*, Vol.6, No.8, 2011, pp.234-241.
[22] I.S. Kashirskiy, and Y.K. Trokhimenko, *Generalized Optimization of Electronic Circuits*. Tekhnika, Kiev, 1979.
[23] V. Rizzoli, A. Costanzo, and C. Cecchetti, Numerical optimization of broadband nonlinear microwave circuits, *IEEE MTT-S Int. Symp*., Vol.1, 1990, pp.335-338.
[24] E.S. Ochotta, R.A. Rutenbar, and L.R. Carley, Synthesis of high-performance analog circuits in ASTRX/OBLX, *IEEE Trans. on CAD*, Vol.15, No.3, 1996, pp.273-294.
[25] A.M. Zemliak, Analog system design problem formulation by optimum control theory, *IEICE Trans. on Fundam.*, Vol.E84-A, No.8, 2001, pp.2029-2041.
[26] A. Zemliak, Novel approach to the time-optimal system design methodology, *WSEAS Trans. Syst.*, Vol.1, No.2, 2002, pp. 177-184.
[27] A. Zemliak, and P. Miranda, Start point and trajectory analysis for the minimal time system design algorithm, *WSEAS Trans. Circuits Syst.* Vol.3, No.4, 2004, pp.765-770.
[28] R. Rojas, O. Camacho, R. Caceres, and A. Castellano, On sliding mode control for nonlinear electrical systems, *WSEAS Transactions Circuits Syst.* Vol.3, 2004, pp.783-788.
[29] L. S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, Inc., N.Y., 1962.
[30] A. M. Zemliak, Dynamic characteristics analysis of analogue networks design process. *IEICE Trans. Fundamentals*, Vol.E92-A, 2009, pp.652-657.
[31] G. Massobrio, P. Antognetti, *Semiconductor Device Modeling with SPICE*, Mc. Graw-Hill, Inc., N. Y., 1993.
[32] E. A. Barbashin, *Introduction to the Stability Theory*, Nauka, Moscow, 1967.
[33] N. Rouche, P. Habets, and M. Laloy, *Stability Theory by Liapunov's Direct Method*, Springer-Verlag, N.Y., 1977.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflict of interest to declare.