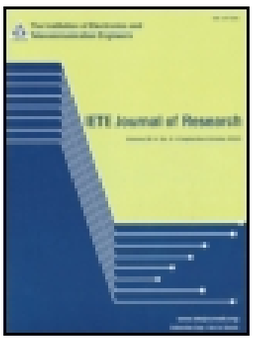


IETE Journal of Research

ISSN: 0377-2063 (Print) 0974-780X (Online) Journal homepage: http://www.tandfonline.com/loi/tijr20

Low Noise Amplifier Synthesis Using Multidimensional MLP Neural Network

S. E. Sorkhabi, M. R. Mosavi & M. Rafei

To cite this article: S. E. Sorkhabi, M. R. Mosavi & M. Rafei (2017): Low Noise Amplifier Synthesis Using Multidimensional MLP Neural Network, IETE Journal of Research, DOI: 10.1080/03772063.2017.1353927

To link to this article: http://dx.doi.org/10.1080/03772063.2017.1353927

Published online: 23 Aug 2017.

IETE JOURNAL OF RESEARCH, 2017
https://doi.org/10.1080/03772063.2017.1353927

Low Noise Amplifier Synthesis Using Multidimensional MLP Neural Network

S. E. Sorkhabi, M. R. Mosavi and M. Rafei

Iran University of Science and Technology, Narmak, Tehran 16846-13114, Iran

ABSTRACT

In this paper, a new method is proposed for determining the design parameters of radio frequency (RF) low noise amplifier (LNA) to obtain a set of desired design specifications. The proposed method is a simulation-based approach in which the multidimensional multilayer perceptron (M-MLP) neural network (NN) is utilized as the synthesis tool. The outputs of the M-MLP NN constitute the design variables, i.e., the numerical values of the passive components, transistors' sizing, and the biasing conditions. The algorithm is implemented in MATLAB and the HSPICE simulator is utilized to verify the synthesis results. Strictly speaking, in each feed-forward process of the NN, a set of design parameters are produced at the output of the M-MLP NN, corresponding to which a set of design specifications are derived using HSPICE. These specifications include the input and output return loss, power gain, reverse isolation, noise figure, K stability factor, 3-dB bandwidth, and centre frequency. Considering the differences between the derived specifications and the desired ones, the connection weights of the M-MLP NN are altered in a way that it improves the overall behaviour of the algorithm toward satisfying the competing design measures. In other words, having the desired specifications, our method yields a completely automated design procedure. The formulation of the approach is described in detail. The dynamic learning rate is utilized to further improve the performance of the algorithm to search the solution space efficiently. The TSMC model for the 0.18 μm CMOS process is used in the LNA design.

KEYWORDS

Design automation; Dynamic learning rate; LNA synthesis; Multidimensional MLP neural network; Simulation-based methods

1. INTRODUCTION

Low noise amplifier (LNA) is the first block in a radio frequency (RF) receiver which intends to amplify the weak noisy signal received by the antenna, while adding the minimum possible noise to it. The main design specifications that are considered in the performance evaluation of the LNAs are the impedance matching criteria in each of the input and output ports, power gain, noise figure, stability factor, bandwidth, and the linearity criterion [1,2]. It is usually a highly challenging task to make these specifications meet the desired requirements due to many diverse non-ideality issues occurring in the RF range of frequencies. Often, repetitive trial-and-error experiments are the integral part of the design hierarchy which demand a great deal of time to fulfil an acceptable design. In the recent decade, optimization methods are proposed as a remedy to come up with these challenges and led to partial automation of the design process. These methods can be mainly categorized as simulation-based and equation-based approaches [3].

Literature that have benefitted the former approach generally utilize the global search methods such as the genetic algorithm (GA) [4,5], simulated annealing (SA) [6,7], particle swarm optimization (PSO) [8,9], or combinatorial methods involving fuzzy algorithm [10] and numerical methods such as Levenberg–Marquardt [11]. These approaches usually provide multi-objective results, i.e., a set of non-dominated solutions to the synthesis problem, known as Pareto front. The solutions provided by the Pareto front create opportunity to evaluate the tradeoffs between the different specifications of the circuit. However, in the circuit applications, not all the solutions of the Pareto front are desirable. For instance, a solution that leads to a high gain and at the same time a low stability factor is not a case of interest. Consequently, in most of these methods, long computational times are consumed to generate the Pareto front in which some solutions are not even acceptable [12].

The equation-based methods are less accurate compared with former category due to the approximations and the simplifications involved in the equation extraction procedures. Meanwhile, since the successive circuit simulations are avoided, the latter features a very high speed, lack of which is considered as a downside for the former. Some of the recent studies in this field have employed genetic programming (GP) [13] in which the circuit design problem is devised as a convex optimization problem, where the relations between circuit parameters and specifications are assumed to be posynomial functions. Nevertheless, this is not the case in most of the RF

circuit synthesis problems where the solution space is usually non-convex. Hence, there is limitation on problems solvable with these methods. Also, one needs to redefine the optimization problem each time a new circuit topology is considered. An outstanding advantage of this approach is that the final solution is independent of the starting point which can be even of the infeasible type [14,15].

As opposed to multi-objective approaches, in single-objective methodologies, the objectives considered in the design problem are aggregated in a single cost function each multiplied in a weight proportional to the significance of the corresponding objective. The minimization of this cost function is then of concern [16,17]. Single-objective optimization does not benefit the diversity of the solutions provided by the multi-objective ones. Also, the lack of information about the tradeoffs between specifications makes it a difficult task to choose the appropriate weights related to each function. However, the final solution could be reachable at a very short time. Hershenson et al. in [13] has made use of GP as the LNA synthesis tool which leads to a single solution.

In the literature perused so far, which have used neural networks (NNs) in RF and analog circuit applications, the NNs are utilized as a modelling tool to obtain equivalent models for the devices or circuits. The resulting models along with the well-known optimization techniques such as GA and PSO are exploited for accelerating the synthesis procedure and to avoid the high computational cost, typical to the circuit simulators [18,19]. In these methods after spending a relatively long time on training the NN, accurate, precise, and fast models will be generated that are valid only in the training interval, due to the poor extrapolation known for the NNs [20]. In general, it is probable that better responses, i.e., global minimum, would be achievable for some parameter values outside the training set [21–24]. In [25], a parameter-free penalty function method has been offered in which the multilayer perceptron NN (MLP NN) is utilized to solve a single-objective optimization function. An online training method is used that does not need predetermined training samples. Only a theoretical example is given and no applied synthesis problem is presented.

In the current work, as opposed to methods that utilize NNs as modelling tools, the MLP NN is employed in a quiet different way, i.e., as a synthesis tool to design a cascoded LNA. Readers can refer to [26] for more details on the synthesis method. To the best of our knowledge, employment of the multidimensional MLP NN (M-MLP NN) directly to search the solution space in the RF circuits' synthesis problem is first presented in this work.

The outline of the paper is as follows. In Section 2, the proposed synthesis method is explained and the effective considerations in the performance of the algorithm are deliberated. Structure of the LNA and the related performance specifications are discussed in Section 3. Section 4 corroborates the proposed method using circuit synthesis results from the algorithm. The performance of the synthesized LNA is evaluated in comparison with other references. Also, the synthesized LNA simulations are provided. Section 5 concludes this paper and gives some guidelines for future research programmes.

2. PROPOSED METHOD FOR THE LNA SYNTHESIS

The outline of the proposed method is as follows: choosing a proper configuration and setting the initial weights for the M-MLP NN, an arbitrary vector is fed to the network as the input (the method is input-independent). After performing one feed-forward process, the outputs of the NN are generated, which constitute the design variables of the synthesis problem. Hence, the number of the neurons in the output layer of the network must be equal to the number of the design variables of the LNA. Now, having the initial values of the design variables, an HSPICE simulation is performed to evaluate the outputs regarding the desired objectives. The corresponding error of each of the circuit specifications is extracted and exploited in the weight updating process by the error back-propagation (EBP) algorithm. The feed-forward operation using the updated weights is performed again and the new design variables are generated. The netlist of the HSPICE is automatically modified, based on the new design variables and the specifications are again evaluated after a run of HSPICE. This procedure will continue until the design objectives are totally satisfied. In fact, during the execution of the algorithm and after each feed-forward process, the simulator automatically evaluates the performance of the algorithm in searching the solution space for the design parameters (we have linked MATLAB and HSPICE and the entire process is performed automatically). It is worth mentioning that the method utilized in this paper is circuit-topology-independent and hence implementable on different structures of RF circuits including LNAs. All that one needs to do is to define the HSPICE netlist, based on the structure under consideration, once at the beginning of each design process. Please note that, as previously mentioned, this method is input-independent.

This is because there are infinite sets of weights which can map an arbitrary input to a desired output. That is to say, by changing the input vector, the algorithm will seek for a new set of weights to reach to the same desired output. Although the method is flexible, the effect of the network structure must not be neglected. It is demonstrated in [26] that an input layer and one hidden layer prior to the last hidden and output layers, which generate the design parameters and specifications, respectively, is an appropriate structure to solve any highly nonlinear problem. Also, note that this method is a real-time application of the MLP NN. Therefore, there is no need for pre-collected training and test datasets. In fact, the MLP NN is not used as a modelling tool as usual, but it is utilized as an optimization/problem solver tool as introduced in detail in [26].

A flowchart of the proposed method is illustrated in Figure 1. The method results in just one solution at the end of the synthesis procedure, and, thus, saves a great deal of time compared to the methods that lead to a set of solutions. Meanwhile, despite single-objective methods, the design specifications are not aggregated in the form of a single cost function, but they distinctly affect the performance of the algorithm, as in multi-objective methodologies.

Figure 2 provides an overview of the proposed method which is basically discussed above. As is obvious, the connection weights in an MLP NN must be consecutively altered through a well-established algorithm. Herein, the algorithm exploited for the M-MLP NN training is an extended form of EBP algorithm [26]. In the prevalent applications of the MLP NN, the error

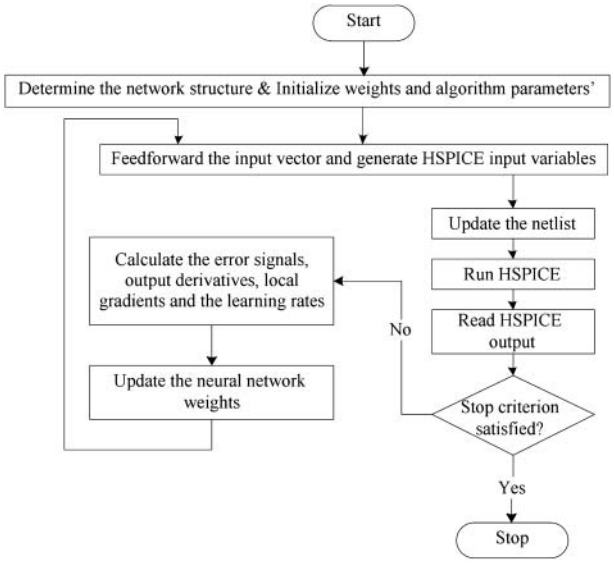


Figure 1: Flowchart of the proposed method

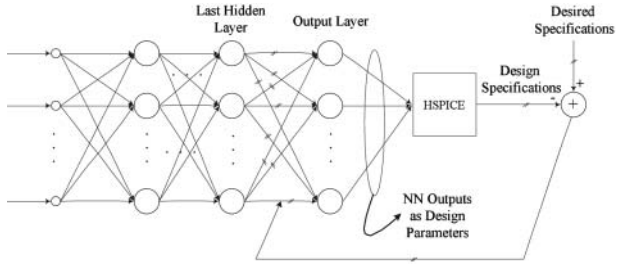


Figure 2: Utilizing the MLP NN as the circuit synthesis tool

signal at the output layer is calculated with regard to the desired target values to which the outputs of the MLP NN are assumed to reach. Since we have no idea what the desired target values for the design variables (M-MLP NN outputs) will be, a weight adjustment between the last hidden layer and the output layer is done using the error corresponding to each of the HSPICE outputs. This resembles the case in which each of the HSPICE-outputted specifications be considered as the output of an activation function in the output layer of the NN. This is to say, the complex analytical equations that the HSPICE uses to analyze the circuit play the role of the activation function of each of the neurons in a hypothetical additional layer after the output layer of the M-MLP NN in Figure 2. Accordingly, some modifications have been applied on the common structure of the MLP NN, and consequently on the learning algorithm that are succinctly described in the following subsections. The reader can refer to [26] for more details.

Bearing in mind the fact that each of the design parameters independently affects the design specifications at the output of the HSPICE block in Figure 2, the variations of each of these outputs due to the changes in the design parameters (M-MLP NN outputs), i.e., partial derivatives, must be involved in updating the weight matrices. In other words, this situation is similar to the case that multivariate activation functions are used instead of the prevalent single-variate ones. The use of multivariate activation functions in the MLP NNs was first proposed by Sollazi and Uncini in [27].

Regarding the above remarks, it is required that the connections between the last hidden layer and the output layer be of the multidimensional form; i.e., the number of the connections between each pair of the neurons must be equal to the number of the HSPICE outputs. This is different from what is known as the common structures of the MLP in which there is always a single connection between each pair of the neurons. The weights corresponding to each connection is then updated using one of the partial derivatives, i.e., the

derivative of each of the HSPICE outputs versus each of the M-MLP NN's outputs.

Consequently, the well-known equations of the EBP algorithm must be redefined in the following way [26]:

$$E = \frac{1}{2}\sum_{j=1}^{s} e_j^2(\mathbf{x}). \tag{1}$$

Equation (1) indicates the energy function of the M-MLP NN in which $e_j(\mathbf{x})$ is the error signal and s, in this application, indicates the number of the HSPICE outputs. Note that Equation (1) implies the cumulative error of the network only in the output layer, where the error due to each design specification contributes to the final error with a weight of 1. The objective of the M-MLP NN training is to minimize this energy function through the repetitive improvements in the connection weights. Referred to [26], the network training is defined as

$$w_{k,ji}^{(l)}(n+1) = w_{k,ji}^{(l)}(n) + \Delta w_{k,ji}^{(l)}(n). \tag{2}$$

In the above equation, $w_{k,ji}^{(l)}(n)$ implies the connection k between the neuron j in layer l and neuron i in layer $l-1$ in iteration n. Considering the above discussions, l is the output layer and $l-1$ is the last hidden layer. For all the other layers, the connections are unidimensional and the equations appear in their usual forms. The weighted sum of the outputs in the hidden layer is as in Equation (3):

$$v_{k,j}^{(l)} = \sum_{i=0}^{N(l-1)} w_{k,ji}^{(l)}(n) y_i^{(l-1)}(n). \tag{3}$$

The delta rule based on the utilized energy function yields

$$\begin{aligned}\Delta w_{k,ji}^{(l)} &= -\eta_{k,ji}^{(l)} \frac{\partial E}{\partial w_{k,ji}^{(l)}} \\ &= -\eta_{k,ji}^{(l)} \frac{\partial E}{\partial e_j^{(l)}} \cdot \frac{\partial e_j^{(l)}}{\partial y_j^{(l)}} \cdot \frac{\partial y_j^{(l)}}{\partial v_{k,j}^{(l)}} \cdot \frac{\partial v_{k,j}^{(l)}}{\partial w_{k,ji}^{(l)}} = \eta_{k,ji}^{(l)} \delta_{k,j}^{(l)} y_i^{(l-1)}.\end{aligned} \tag{4}$$

$\eta_{k,ji}^{(l)}$ is the learning rate corresponding to each connection and the local gradient is defined as

$$\delta_{k,j}^{(l)}(\mathbf{x}) = e_j^{(l)}(\mathbf{x}) . \frac{\partial \varphi_j^{(l)}(\mathbf{x})}{\partial x_k}\bigg|_{\left(\mathbf{x}=\left\{x_1=v_{1,j}^{(l)},\ldots,x_{M(l)}=v_{M(l),j}^{(l)}\right\}\right)}. \tag{5}$$

Note that in the above equation, regarding the last layer, $\varphi_j(\mathrm{x})$ indicates the HSPICE outputs and x_k are the outputs of the M-MLP NN. For the previous layers, $\varphi_j(\mathrm{x})$ is the activation function and x_k implies the input to the activation function. Therefore, the error signal is defined as Equation (6):

$$e_j^{(l)}(\mathbf{x}) = \sum_{k=1}^{M(l)} \left(\sum_{n=1}^{N^{(l+1)}} \delta_{k,n}^{(l+1)} w_{k,nj}^{(l+1)} \right). \tag{6}$$

Considering the last layer, the condition is somehow different. The error signal in the last layer is usually defined in the form of Equation (7).

$$e_j(\mathbf{x}) = y_j^* - f_j(\mathbf{x}) \tag{7}$$

in which y_j^* is the target or the desired value to which $f_j(\mathbf{x})$ ought to converge. Using Equation (7) would give rise to instability in the cases that the range of the considered function $f_j(\mathbf{x})$ (HSPICE output) grows unboundedly. Consequently, the value of $e_j(\mathbf{x})$ in Equation (7) is mapped to a limited interval using a strictly increasing squashing function, e.g., tanh(.) [26]. The synthesis problem in this paper is defined in the form of two kinds of constraint functions, i.e., the equality and inequality constraints. Considering each of these two types of constraints, the behaviour of the algorithm is interpreted in different ways. In the case of an equality constraint as it is defined in Equation (8), reaching one exact array of points, i.e., $\mathbf{x}$, is the target, so that this array could hold Equation (8). For instance, reaching an exact operating frequency of 5 GHz is of concern.

$$h_k(\mathbf{x}) = 0, \qquad k = 1, 2, \ldots K. \tag{8}$$

Therefore, the mapping function is defined as Equation (9).

$$e_j(\mathbf{x}) = \alpha_j . \tanh\Big(\Big(y_j^* - h_j(\mathbf{x}) \Big) / \sigma_j \Big). \tag{9}$$

In the case of the inequality constraint, which is defined in Equation (10), reaching the boundaries of the constraints and the internal points are of concern.

$$g_j(\mathbf{x}) \geq 0, \qquad j = 1, 2, \ldots, J. \tag{10}$$

That is to say, once the algorithm reaches the boundaries of the constraints, the conditions are satisfied and the error will be reduced to a user-defined level, i.e. λ. For example, if $|S_{21}| \geq 15$ dB is of concern, as soon as the algorithm reaches 15 dB, the error signal

gets a small predefined value close to zero and can be further diminished as the algorithm gets better values for the target inside the boundary (as $|S_{21}|$ gets larger values). Hence, the error signal is defined in the form of Equation (11).

$$e_j(\mathrm{x}) = \begin{cases} \alpha_j.\left(\tanh\left(\dfrac{-g_j(\mathrm{x})}{\sigma_j}+\theta_j\right)+1\right) & \text{for } g_j(\mathrm{x})<0 \\ 0 & \text{for } g_j(\mathrm{x})\geq 0 \end{cases} \quad \text{and} \quad j=1,\ldots,t. \tag{11}$$

Therefore, the error value at the boundary of the constraint where $g_i(\mathbf{x})$ gets to zero is defined to have a value of λ, and

$$\theta_i = \operatorname{arctanh}\left(\frac{\lambda_i}{\alpha_i}-1\right). \tag{12}$$

As the definitions in Equations (8) and (10) indicate, in both of Equations (9) and (11), the target value, i.e. y_i^*, is considered to be zero. As the error signal gets smaller and smaller, design variables would converge to the desired values. Note that in Figure 3(a) which is the case for equality constraint, as the algorithm gets further from the desired value, corresponding to the origin, an error is applied proportional to the distance from it; this is true for deviation from the origin in both sides. However, in Figure 3(b) which is the case for inequality constraint, as the algorithm gets closer to the boundary of the constraint, the error is reduced and finally gets to λ at the boundary. As stated above, for the solutions inside the boundary, this error (corresponding to the left plane in Figure 3(b)) can be further reduced.

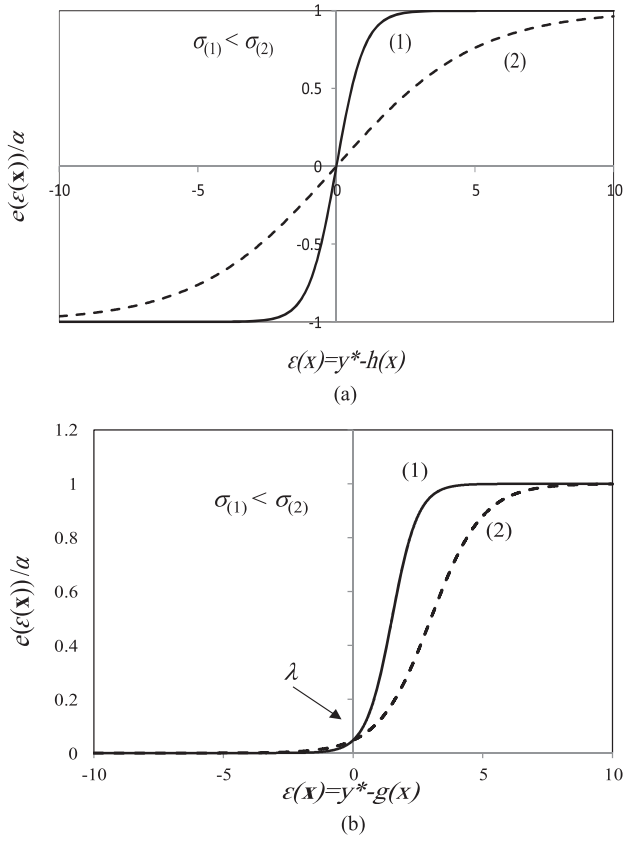


Figure 3: The error signal for (a) equality and (b) inequality constraints

In both cases, α_j designates the maximum value of the range of the error signal. In this work, it is considered to be between 0 and 1. Smoothness of the function, $e_j(\mathbf{x})$, alters by adjusting σ_j. The smaller the parameter σ_j is, the more are the variations of $e_j(\mathbf{x})$ around the origin and, therefore, the more sensitive the algorithm will be to the variations of $f_j(\mathrm{x})$ in the vicinity of the origin. As a rule of thumb, the value of σ_j is selected to be equal to the one-third of the maximum predictable value for $f_j(\mathrm{x})$. Both parameters, α_j and σ_j, are used to determine the priority of the constraints to be satisfied. If we generally regard the error signal as

$$e_j^{(L)}(\mathbf{x}) = \alpha_j.\left(\tanh\left(\frac{\varepsilon_j(\mathbf{x})}{\sigma_j}+\theta_j\right)+1\right), \tag{13}$$

then, Equation (5) should be altered to Equation (14):

$$\delta_{k,j}^{(L)}(\mathbf{x}) = \frac{\alpha_j}{\sigma_j}\left[1-\left(\frac{e_j^{(L)}(\mathbf{x})}{\alpha_j}-1\right)^2\right] e_j^{(L)}(\mathbf{x}) d_j^{(L)}(\mathbf{x}), \quad \text{for } \varepsilon_j(\mathbf{x}) = y_j * -f_j(\mathbf{x}) \tag{14}$$

in which

$$d_j^{(L)}(\mathbf{x}) = \left.\frac{\partial \varphi_j^{(L)}(\mathbf{x})}{\partial x_k}\right|_{\left(\mathbf{x}=\left\{x_1=v_{1,j}^{(L)},\ldots,x_m=v_{m,j}^{(L)}\right\}\right)}. \tag{15}$$

2.1 Learning Rate

Learning rate is one of the most important parameters of the algorithm to search the solution space effectively. It has been proved that the dynamic learning rate would improve the performance of the NN in both aspects of the algorithm speed and global search. In most cases, it is desirable to alter the learning rate in a decreasing manner, as the algorithm proceeds [28]. Learning rates dependent on the error or the derivative of the output function both have been already exploited [29] and adapted for the goal of global search.

Here, we benefit the dynamic learning rate proposed in [28]. The authors have proved that with this learning rate the global search of the space is guaranteed. The learning rate introduced in [28] is based upon the step size rule and, in iteration n, is defined as

$$\eta_n = \beta_n.\left(E(w(n)) - E_n^{\text{lev}}\right)/C^2 \tag{16}$$

$$0 < \underline{\beta} \le \beta_n \le \overline{\beta} < 2, \tag{17}$$

where C is a positive real constant and β_n is a positive random constant. If E^* is the smallest attainable value of the network energy, then E_n^{lev} is an estimation of E^* and it is modified in each iteration based on the observed value for $E(w(n))$. E_n^{lev} is defined as

$$E_n^{\text{lev}} = E_{\min} - \kappa_n. \tag{18}$$

where $E_{\min}$ is as (19), and κ_n is an adjustable parameter which should satisfy Equation (20).

$$E_{\min} = \min_{1 \le i \le n} E(w(i)). \tag{19}$$

$$n \to \infty \Rightarrow \kappa_n \to 0\ , \quad \sum_{n=0}^{\infty} \kappa_n^2 = \infty\,. \tag{20}$$

In this paper, κ_n is selected as

$$\kappa_n = \rho\ /\ \sqrt[\gamma]{n}. \tag{21}$$

In the above equation, ρ and γ are positive real constants, where $\gamma \ge 2$. κ_n will be changed in a diminishing way as the epochs increase, and the energy will gradually reach its minimum value. That is to say, the initial global search will eventually change to a local search. Note that the dynamic learning rate mentioned is applied just to the connections in the output layer. In general, a discriminate learning rate, η_{ij}, is considered for each of the connections between the output and the last hidden layer. For the rest of the layers, the learning rate is considered as a constant. The convergence of the algorithm using the learning rate presented in Equation (16) is comprehensively proved in [28].

3. CASCODED LNA FOR SYNTHESIS PURPOSE

Figure 4 depicts a cascoded LNA structure. M_1 operates in a common-source configuration and M_2 isolates the input and output of the amplifier. The degeneration inductor, L_{deg}, in the source of M_1 is devised to match the input impedance to a real impedance of 50 Ω without adding thermal noise. The characteristic impedance for

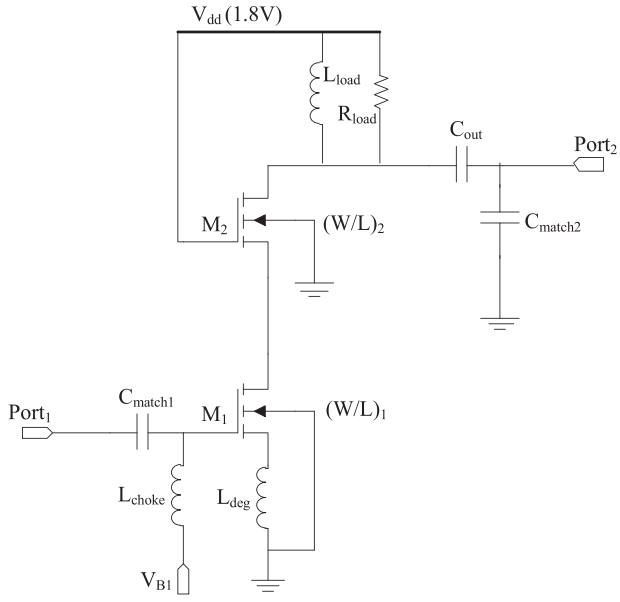


Figure 4: Cascoded LNA structure

the input and output ports is 50 Ω. Thus, impedance matching networks are considered to match both the input and output impedances to 50 Ω. It has been demonstrated that this structure has the best performance among others regarding the noise figure and high power-gain. A disadvantage often mentioned for the cascoded configuration is its weak linearity performance [30].

LNA performance is evaluated with specific performance metrics, some of which are succinctly described below.

3.1 Design Specifications

Scattering parameters provide criteria for the performance evaluation of an electrical network. Most of the behavioural specifications of the RF circuits, such as the power gain, reflection coefficient, input and output impedance matching and the stability factor, are expressible by means of these parameters. In the case of a two-port network, as in this paper, they are expressed as follows:

- S_{11} or the input return loss is the ratio between the wave b_1 coming out of and a_1 going into the input port, without a wave traversing from the output port. S_{11} determines the impedance matching at the input port and its preferred value is less than −10 dB for a desirable performance.
- S_{12} or the reverse isolation describes a ratio of the wave a_2 travelling from the output toward the input

port being detected as b_1 with no incident wave a_1 at the input port.

- S_{21}, recognized as the power gain, describes a ratio of the wave a_1 travelling from the input toward the output port being detected as b_2 with no incident wave a_2 at the output port.
- S_{22} or the output return loss is the ratio between the wave b_2 coming out of and a_2 going into the output port, with no incident wave a_1 at the input port. S_{22} determines the impedance matching at the output port and it is desired to have a value less than -10 dB for acceptable performance of the network.

Another criterion often regarded is the noise figure which shows the degradation of the LNA performance in the presence of the noise. In other words, the noise figure is a criterion by which the degradation of the signal-to-noise ratio is determined.

$$\begin{aligned} NF &= 10\log(F) = 10\log\left(\frac{SNR_{in}}{SNR_{\text{out}}}\right) = 10\log\left(\frac{\frac{Signal_{in}}{Noise_{in}}}{\frac{Signal_{\text{out}}}{Noise_{\text{out}}}}\right) \\ &= 10\log\left(\frac{Gain \times Noise_{in} + Noise_{amp}}{Noise_{in}}\right) \\ &= 10\log\left(1 + \frac{Noise_{amp}}{Gain \times Noise_{in}}\right). \end{aligned} \tag{22}$$

SNR_{in} and SNR_{out} are the signal-to-noise ratio at the input and the output of the amplifier, respectively. In the ideal case, if the amplifier adds no extra noise, then the noise figure will be equal to 0 dB. The amplifier stability is evaluated with the Rollett stability factor as stated in Equation (23).

$$K = \left(1 + |\Delta|^2 - |S_{11}|^2 - |S_{12}|^2\right)/(2|S_{11}|.|S_{12}|) \tag{23}$$

in which

$$\Delta = S_{11} \times S_{22} - S_{21} \times S_{12}. \tag{24}$$

For the amplifier to be stable, it is necessary that $K > 1$ and $|\Delta| < 1$.

4. APPLICATION OF THE PROPOSED METHOD FOR LNA SYNTHESIS

In view of Figure 4, the design parameters for the synthesis problem are regarded as $[L_1, L_2, nr_1, nr_2, V_{B1}, R_{\text{load}}, L_{\text{load}}, L_{\text{choke}}, L_{\text{deg}}, C_{\text{match1}}, C_{\text{match2}}, C_{\text{out}}]$. These parameters represent the channel length of the transistors, finger numbers (W has been set to 2.5 μm in the utilized TSMC model), bias voltage for M_1, and the numerical values of the passive components shown in Figure 4. The synthesis objectives are listed in Table 1.

Table 1: The objectives of the synthesis problem

Specification	S_{11} (dB)	S_{12} (dB)	S_{21} (dB)	S_{22} (dB)	NF (dB)	K	BW_{3db} (MHz)	$Freq.$ (GHz)
Goal	<-10	<-25	>15	<-10	<2	>1	>500	5

Adjusting the parameters discussed in Section 2, the algorithm would be able to search the design parameters in the solution space so that the corresponding specifications would satisfy the goals in Table 1. These parameters are set as: $\rho = 1$, $\gamma = 0.5$, $C = 40$, $\lambda = [0.05, 0.05, 0.04, 0.05, 0.04, 0.05, 0.04]$, $\alpha = [0.37, 0.4, 0.4, 0.5, 0.4, 0.4, 0.4, 1]$, and $\sigma = [55, 20, 100, 100, 100, 100, 100, 4]$. The learning rate for the output layer is considered as dynamic and, for the rest of the layers, it is constant and equal to 0.2. The structure of the NN is [20, 15, 12], which indicates the 20 neurons in the first hidden layer, 15 neurons in the second hidden layer and 12 neurons in the output layer. The number of the neurons in the hidden layers is chosen empirically. Please note that, since there are infinite combinations of the input and the weights which result in the desired design parameters, the inputs to the NN is not of concern and all that the algorithm needs to do is to find the weights' matrix proportional to an arbitrary input vector that leads to the desired design variables. Also, since the problem of RF circuit design deals with high-order polynomial and exponential equations with inherent divergence characteristics, the algorithm is prone to instability. Thus, it is crucial to select small initial weights, and, therefore, all the initial weights are selected to be 0.01. The activation functions for the neurons in the hidden layers are all sigmoidal functions, i.e., $\varphi(x) = 1/(1 + \exp(-ax))$. Considering the output layer, and transistor sizing limitations imposed by the TSMC technology, considerations about the bias condition to maintain the transistors in the saturation region, and reasonable ranges for the passive components, the activation function for the output layer is not of the prevalent linear form. Instead, the linear activation function is bounded at the lower and upper ends for each output neuron consistent to the required ranges of the design parameters. As mentioned before and regarding Table 1, all the objectives in this paper are of the constraint type. The first seven goals are defined in the form of inequality

Table 2: The synthesis result of the LNA applying the proposed method

Design parameters	L_1 (μm)	L_2 (μm)	nr_1	nr_2	V_{B1} (v)	R_{load} (Ω)	L_{load} (nH)	L_c (nH)	L_d (pH)	C_{match1} (pF)	C_{match2} (pF)	C_{out} (pF)
Results	0.18	0.36	73	24	0.8084	404	3.511	1.6	1.053	0.192	0.116	0.25

Table 3: Constraints applied to the design parameters

Design parameters	L_1 (μm)	L_2 (μm)	nr_1	nr_2	V_{B1} (v)	R_{load} (Ω)	L_{load} (nH)	L_c (nH)	L_d (pH)	C_{match1} (pF)	C_{match2} (pF)	C_{out} (pF)
Min	0.18	0.18	4	4	0.55	0.1	0.1	0.1	0.1	0.01	0.01	0.01
Max	0.5	0.5	128	128	0.92	2	10	10	1000	2	2	2

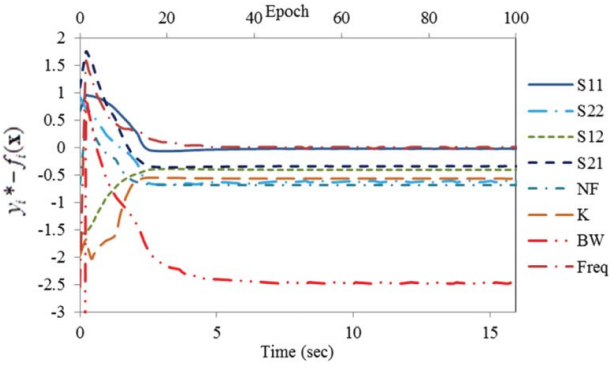


Figure 5: Behaviour of the algorithm behaviour during the synthesis process versus time and epochs

constraints and the frequency condition is defined as equality constraint. As a result, the error signals of Equations (9) and (11) are utilized, respectively. Starting from the initial values of $[0.18, 0.5, 115, 40, 1.5, 650, 5.5, 2.5, 1.5, 0.5, 0.2, 0.5]$, the algorithm will eventually converge to the design parameters in Table 2.

Constraints applied to the design variables considering the utilized process and the conditions for the acceptable performance of the amplifier are tabulated in Table 3. Note that, both M_1 and M_2 should operate in the saturation region to provide acceptable performance. Figure 5 illustrates the behaviour of the algorithm during the time it searches the solution space. The difference between the attained specifications and the desired ones presented in Table 1 is displayed for each specification versus time and epoch.

With regard to Equations (9) and (11), the inequality constraints are satisfied as soon as the curves reach a negative value. In other words, because the boundary of the inequality constraint corresponds to the value of zero, the internal points of the constraint boundary get negative values. For the equality constraint, this difference will converge to a negligible value close to zero. As it is obvious, all the goals are satisfied in less than 100 epochs and in a time shorter than 15 s. The amplifier of Figure 4 is simulated by HSPICE for the parameters obtained by the algorithm and the results are provided in Table 4.

Figure 6 displays the behaviour of the circuit in the form of the input and output matching, reverse isolation, and power gain of the amplifier. All the computations are performed using a Pentium 4 system with a 2.6 GHz CPU and a 4GHz RAM. Table 5 compares the performance of the proposed method with some of the references. In order to make it easier to evaluate the performances in each case, three criteria are determined in the form of Equations (24)–(26). The definitions are adopted from [30] with some variations, according to the information available at the references in Table 5. For the reader's ease, the numerical values of the three types of the FOMs in Table 5 are scaled by a factor of 1E−5.

$$FOM_1 = (|S_{11}| \times |S_{22}| \times |S_{12}| \times |S_{21}| \times K \times Freq.)/NF, \tag{25}$$

$$FOM_2 = (|S_{11}| \times |S_{12}| \times |S_{21}| \times Freq.)/NF, \tag{26}$$

$$FOM_3 = (|S_{11}| \times |S_{22}| \times |S_{21}| \times K \times Freq.)/NF. \tag{27}$$

A larger FOM is always better. As it is tabulated in Table 5, the FOM pertained to this work, in each column of the table, has the highest value, which indicates the capability of the proposed method in the LNA synthesis problem.

Table 4: The HSPICE simulation results for the amplifier of Figure 4 using the parameters obtained by the algorithm

Specifications	S_{11} (dB)	S_{12} (dB)	S_{21} (dB)	S_{22} (dB)	NF (dB)	K	BW_{3db} (MHz)	$Freq.$ (GHz)
Results	−17.2	−39.3	18.2	−16.4	1.29	5.26	911.89	5

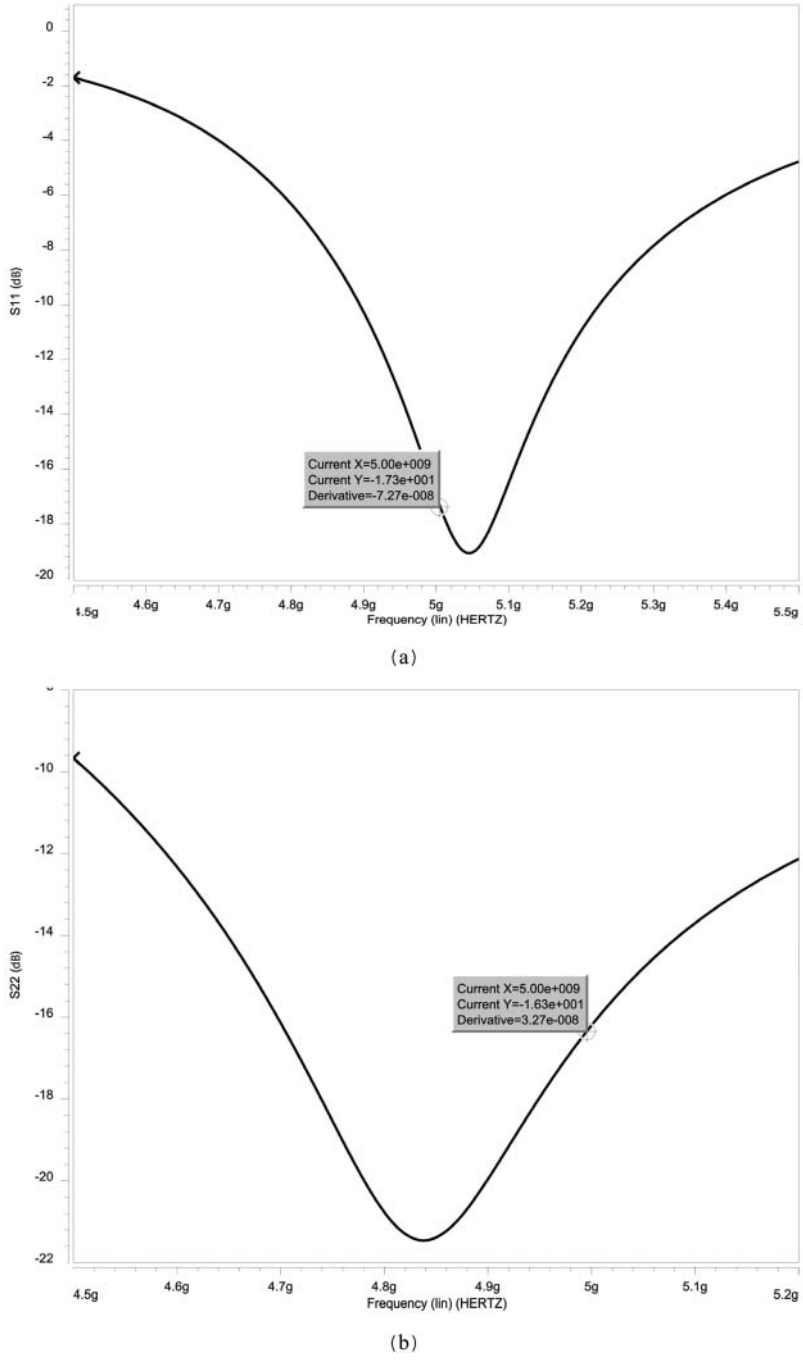
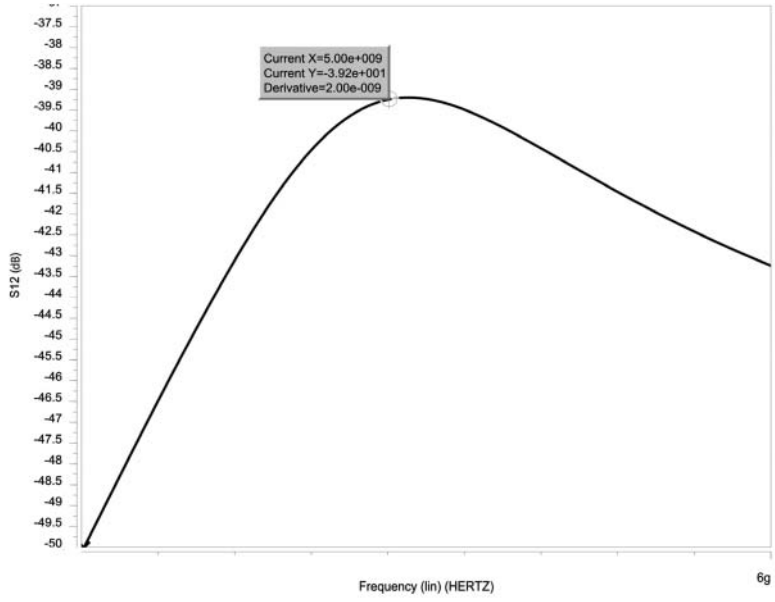
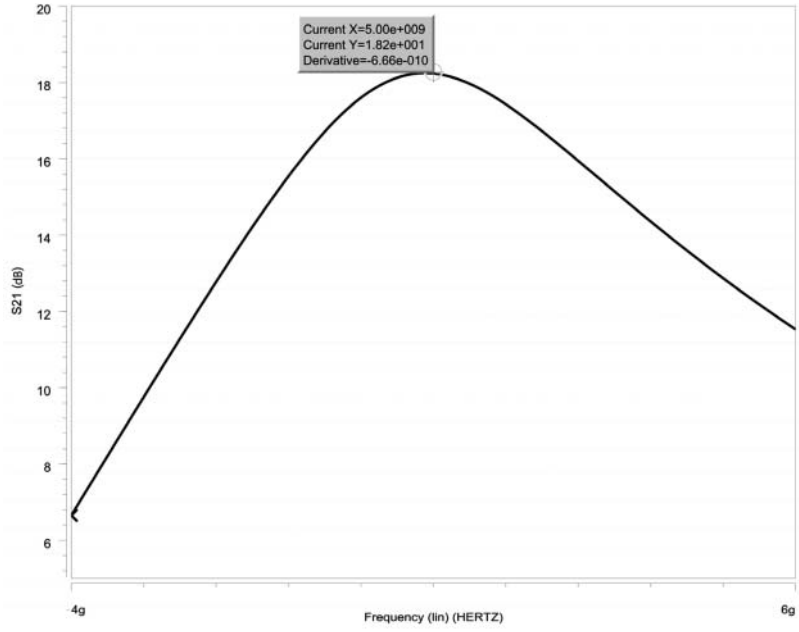


Figure 6: Scattering parameters for the cascode LNA. (a) Input return loss, (b) output return loss, (c) reverse isolation and (d) power gain

Although it is demonstrated through seven popular benchmarks in [26] that the current method can perfectly optimize/solve highly nonlinear mathematical set of equations, almost independent from the starting point, it seems that in case of RF circuit synthesis, we encounter a much more complex problem as the algorithm needs to start from a feasible point to get to reasonable solutions. That is to say, in case of RF synthesis problem, starting from an infeasible initial point, the algorithm might lack efficiency and be trapped in a local minimum. It should be noted that this method cannot be considered as a replacement for intelligent search methods, but as a fast method which leads to reasonable and practical solutions for RF circuit design by starting from a feasible point. However, making the approach independent of the starting point is the highest priority of our research scope for future work.



(c)



(d)

Figure 6: (Continued)

Table 5: Performance comparison between the recently published LNAs

	$\lvert S_{11}\rvert$ (dB)	$\lvert S_{22}\rvert$ (dB)	$\lvert S_{12}\rvert$ (dB)	$\lvert S_{21}\rvert$ (dB)	*NF* (dB)	*K*	*Freq.* (GHz)	FOM_1	FOM_2	FOM_3
This work	17.2	16.4	39.3	18.2	1.29	5.26	5	41.134	0.476	1.046
[2][a]	14.4	–	28	13.3	2.9	–	5.5	–	0.102	–
[5]	11.22	–	32.2	28.5	2.26	–	5.25	–	0.239	–
[6]	15.4	18.37	37.19	18.86	2.92	3.992	0.9	2.441	0.033	0.065
[10]	14.1	22.6	39.3	12.7	0.979	10.7	2.14	37.199	0.154	0.946
[11]	27.89	20.01	–	20.22	1.45	4.976	2.5	–	–	0.968

[a]These are measurement results.

5. CONCLUSION

The fact that the design process of the RF circuits is a challenging one has encouraged the researchers to investigate alternatives to facilitate this procedure. Therefore, several attempts have been accomplished during the last decade to design these circuits optimally using the optimization algorithms, which, to some extent, led to the automation of the design procedure. In this paper, a new method was proposed for the synthesis of the RF circuits in which the M-MLP NN was benefited as the synthesis tool. The proposed method was utilized in the synthesis process of a cascoded LNA. This method is a simulation-based approach in which each time the feed-forward process is completed by the M-MLP NN to produce the design variables, HSPICE is used to evaluate the output specifications. To improve the performance of the algorithm, dynamic learning rate has been utilized in the training procedure. Also, the amplitude of the error signal has been mapped to a limited interval using a squashing function to avoid numerical divergence of the algorithm in the cases that the output domain grows unlimitedly. The results showed that the proposed algorithm was capable to achieve satisfying solutions in a very short time. The main advantage of the proposed algorithm is its generality and applicability to different circuit topologies with just modifying the HSPICE netlist based on the targeted circuit. Although this method benefits the high speed of convergence to final solutions, the necessity of calculating the derivatives of the outputs of HSPICE versus the design parameters imposes additional HSPICE runs in each iteration of the algorithm, i.e., the number of the simulation runs would be equal to the number of design parameters plus one, where the additional runs are performed to compute the output derivatives. It would be possible to achieve the final solutions in a much shorter time if these additional runs are eliminated. For the future researches in this field, employment of the derivative-free algorithms will be studied to train the M-MLP NN, through which there will be no need for derivative calculations. Therefore, synthesis process would be accelerated to a great deal. The other promising area for the future research is to devise an alternative to make the algorithm independent of the initial point. Due to this fact, the algorithm may lead to different local minimums each time the starting point is changed. However, the performance efficiency of the proposed method is verified with the final simulation results and the comparisons made in accordance with the other references.

REFERENCES

1. T. K. Nguyen, C. H. Kim, G. J. Ihm, M. S. Yang, and S. G. Lee, "CMOS low-noise amplifier design optimization techniques," *IEEE Trans. Microwave Theory Tech.*, Vol. 52, no. 5, pp. 1433–42, 2004.
2. D. Linten, S. Thijs, M.I. Natarajan, P. Wambacq, W. Jeamsaksiri, J. Ramos, A. Mercha, S. Jenei, S. Donnay, and S. Decoutere, "A 5 GHz fully integrated ESD-protected low-noise amplifier in 90-nm RF CMOS," *IEEE J. Solid-State Circuits*, Vol. 40, no. 7, pp. 1434–42, 2005.
3. S. E. Sorkhabi and L. Zhang, "Automated topology synthesis of analog and RF integrated circuits: A survey," *Integr. VLSI J.*, Vol. 56, pp. 128–38, 2017.
4. Ö. S. Sönmez and G. Dündar, "Simulation-based analog and RF circuit synthesis using a modified evolutionary strategies algorithm," *Integr. VLSI J.*, Vol. 44, no. 2, pp. 144–54, 2011.
5. A. Somani, P. P. Chakarabarti, and A. Patra, "An evolutionary algorithm-based approach to automated design of analog and RF circuits using adaptive normalized cost functions," *IEEE Trans. Evol. Comput.*, Vol. 11, no. 3, pp. 337–53, 2007.
6. F. Kalantari, N. Masoumi, and A. R. Hoseini, "High gain LNA design for WMAN receiver & optimization with simulated annealing algorithm," in *Proceedings of the International Conference*, Gdynia, Poland, 2006, pp. 299–303.
7. G. Tulunay and S. Balkır, "A synthesis tool for CMOS RF low-noise amplifiers," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 27, no. 5, pp. 977–82, 2008.
8. M. Kotti, A. Sallem, M. Bougharriou, M. Fakhfakh, and M. Loulou, "Optimizing CMOS LNA circuits through multi-objective meta heuristics," in *XIth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design*, Gammarth, Tunisia, 2010, pp. 1–6.
9. V. Ceperic, Z. Butkovic, and A. Baric, "Design and optimization of self-biased complementary folded cascoded," in *IEEE Mediterranean Electrotechnical Conference,* Malaga, Spain, 2006, pp. 145–48.
10. X. Wang, C. Zhou, Z. Zhang, T. Ren, and L. Liu, "Optimal RF IC design based on fuzzy genetic algorithm," in *Asia Pacific Conference on Postgraduate Research in Microelectronics & Electronics*, Shanghai, China, 2009, pp. 229–32.
11. Y. Li, "A simulation-based evolutionary approach to LNA circuit design optimization," *Appl. Math. Comput.*, Vol. 209, no. 1, pp. 57–67, 2009.
12. C. Sanchez-Lopez, R. Castro-Lopez, E. Roca, F. V. Fernandez, R. Gonzalez-Echevarria, J. Esteban-Muller, J. M. Lopez-Villegas, J. Sieiro, and N. Vidal, "A bottom-up approach to the systematic design of LNAs using evolutionary optimization," in *XIth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design*, Gammarth, Tunisia, 2010, pp. 1–5.
13. M. M. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a CMOS Op-Amp via geometric programming,"

IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., Vol. 20, no. 1, pp. 1–21, 2001.

14. W. K. So, W. T. Cheung, Y. Liu, H. K. Kwan, and N. Wong, "Design and optimization of highly linear CMOS low noise amplifier via geometric programming," in *7th International Conference on ASIC*, Guilin, China, 2007, pp. 423–6.

15. W. T. Cheung and N. Wong, "Optimized RF CMOS low noise amplifier design via geometric programming," in *International Symposium on Intelligent Signal Processing and Communications*, Tottori, Japan, 2006, pp. 423–6.

16. I. Y. Kim and O. L. de Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Struct. Multidisc. Optim.*, Vol. 29, no. 2, pp. 149–58, 2005.

17. K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *Int. J. Comput. Intell. Res.*, Vol. 2, no. 3, pp. 273–86, 2006.

18. S. K. Mandal, S. Sural, and A. Patra, "ANN- and PSO-based synthesis of on-chip spiral inductors for RF ICs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 27, no. 1, pp. 188–92, 2008.

19. G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 22, no. 2, pp. 198–212, 2003.

20. Th. Ebner, Ch. Magele, B. R. Brandstatter, and K. R. Richter, "Utilizing feed-forward neural networks for acceleration of global optimization procedures," *IEEE Trans. Magnet.*, Vol. 34, no. 5, pp. 2928–31, 1998.

21. Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microwave Theory Tech.*, Vol. 51, no. 4, pp. 1339–50, 2003.

22. Q. J. Zhang, J. Bandler, S. Koziel, H. Kabir, and L. Zhang, "ANN and space mapping for microwave modeling and optimization," *IEEE MTT-S Int. Microwave SympDigest*, 2010, pp. 980–3.

23. J. E. R. Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microwave Theory Tech.*, Vol. 52, no. 1, pp. 420–35, 2004.

24. D. Gorissen, L. Zhang, Q. J. Zhang, and T. Dhaene, "Evolutionary neuro-space mapping technique for modeling of nonlinear microwave devices," *IEEE Trans. Microwave Theory Tech.*, Vol. 59, no. 2, pp. 213–29, 2011.

25. J. Reifman and E. E. Feldman, "Multilayer perceptron for nonlinear programming," *J. Comput. Oper. Res.*, Vol. 29, no. 9, pp. 1237–50, 2002.

26. M. Rafei, S. Ebrahim Sorkhabi, and M. R. Mosavi, "Multi-objective optimization by means of multidimensional MLP neural networks," *Neural Network World*, Vol. 14, no. 1, pp. 31–56, 2014.

27. M. Solazzi and A. Uncini, "Regularizing neural networks using flexible multivariate activation function," *J. Neural Networks*, Vol. 17, no. 2, pp. 247–60, 2002.

28. R. Zhang, Z. Xu, G. Huang, and D. Wang, "Global convergence of online BP training with dynamic learning rate," *IEEE Trans. Neural Netw. Learn. Syst.*, Vol. 23, no. 2, pp. 330–41, 2012.

29. L. Behera, S. Kumar, and A. Patnaik, "On adaptive learning rate that guarantees convergence in feed-forward networks," *IEEE Trans. Neural Netw.*, Vol. 17, no. 5, pp. 1116–25, 2006.

30. A. Mandan, M. J. McPartlin, C. Masse, W. Vaillancourt, and J. D. Cressler, "A 5 GHz 0.95 dB NF highly linear cascoded floating-body LNA in 180 nm SOI CMOS technology," *IEEE Microw. Wireless Components Lett.*, Vol. 22, no. 4, pp. 200–2, 2012.

Authors

S. E. Sorkhabi received BSc degree in electrical engineering from University of Tabriz, Tabriz, Iran and MSc. degree from Iran University of Science and Technology, Tehran, Iran in 2008 and 2013, respectively. Her research interest includes RF and microwave integrated circuit design, RF and microwave front-ends for wireless applications, and RF and microwave circuit design for biomedical applications.

E-mail: s_ebrahimi@elec.iust.ac.ir

M. R. Mosavi received his BS, MS and PhD degrees in Electronic Engineering from Iran University of Science and Technology (IUST), Tehran, Iran in 1997, 1998 and 2004, respectively. He is currently faculty member of Department of Electrical Engineering of IUST as full professor. He is the author of about 300 scientific publications on journals and international conferences. His research interests include circuits and systems design.

E-mail: m_mosavi@iust.ac.ir

M. Rafei received MSc degree from Iran University of Science and Technology, Tehran, Iran in 2013. His research interest includes RF analog circuit design, circuit design automation, optimization using neural networks and evolutionary algorithms.

E-mail: rafei@elec.iust.ac.ir