A Neural Network Modeling Approach to Circuit Optimization and Statistical Design

A. Hafid Zaabab, *Student Member, IEEE*, Qi-Jun Zhang, *Member, IEEE*, and Michel Nakhla, *Senior Member, IEEE*

***Abstract*—The trend of using accurate models such as physics-based FET models, coupled with the demand for yield optimization results in a computationally challenging task. This paper presents a new approach to microwave circuit optimization and statistical design featuring neural network models at either device or circuit levels. At the device level, the neural network represents a physics-oriented FET model yet without the need to solve device physics equations repeatedly during optimization. At the circuit level, the neural network speeds up optimization by replacing repeated circuit simulations. This method is faster than direct optimization of original device and circuit models. Compared to existing polynomial or table look-up models used in analysis and optimization, the proposed approach has the capability to handle high-dimensional and highly nonlinear problems.**

I. Introduction

THE DEMAND in microwave industry for manufacturability and fast design cycles creates the need for statistical design techniques. Yield analysis and optimization, which take into account the manufacturing tolerances, model uncertainties, variations in the process parameters, etc., have become widely accepted as indispensable components of the circuit design methodology [1]–[5]. On the other hand, for microwave circuit design the effectiveness of modern CAD methods relies heavily on accurate models of active and passive elements. Models such as physics-based models (PBM) for active devices, e.g., [6], [7], become necessary. The use of such accurate models, however, is at the expense of much increased computational cost. Combining yield optimization and physics-based modeling results in a computationally very intensive task.

Standard optimization and statistical design approaches require repeated circuit simulations. Since each circuit simulation involves a CPU-intensive procedure to solve the physics-based equations, such existing optimization methods are more oriented towards off-line computations. They are not suitable for practical interactive design where designers may need to reoptimize the circuit after modifications in specifications, or even circuit topologies. To address this problem, two types of approximations have been previously used. 1) Multidimensional polynomial (or its variants such as splines or response surface) models, e.g., [3], [4], [8]–[10], to approximate and replace original simulations during optimization. However, this approach can handle only mild nonlinearity in high-dimensional space. It typically requires model building or updating during optimization, consuming valuable on-line CPU time. 2) The look-up table approach, e.g., [11]–[14], to approximate and to replace accurate device or circuit simulations. However, the size of the table grows exponentially with dimension and the table becomes too difficult to generate and manage when many parameters of a device or a circuit are involved.

On the other hand, neural networks have become a much important vehicle in the signal processing area for speech processing, vision, control systems, and more [15]–[18]. Recently, it has been applied to microwave impedance matching [19], to study the effects of design factors on printed circuit board (PCB) assembly yield [20], in modeling the properties of silicon dioxide films [21], and in manufacturing process modeling [22]. Neural networks enjoy some distinguished characteristics including the ability to learn from data, to generalize patterns in data, and to model nonlinear relationships. These appealing features make neural networks a good candidate for overcoming some of the difficulties in traditional device and circuit modeling and optimization. However, this potentially powerful modeling approach has not been seriously addressed in the literature and to bridge this gap is the objective of this paper.

Presented in the paper is a new approach to microwave circuit analysis, optimization, and statistical design featuring neural network models at either device or circuit levels. At the device level, the neural network represents a physics-oriented FET model yet without the need to solve device physics equations repeatedly during optimization [23]. At the circuit level, the neural network speeds up optimization by replacing repeated circuit simulations. The size of the proposed model does not grow as fast as exponentially with dimension and, in theory, can model any degree of nonlinearity. This proposed approach is much faster than traditional optimization.

In Section II, the structure of the neural network suitable for device and circuit approximation is presented. Section III describes the model training algorithm and parameters. In Section IV, the use of the model in two different cases, namely, simulation and optimization, is introduced. Section V describes the implementation of the neural network model into a CAD system. Finally, in Section VI, three examples with either device- or circuit-level modeling are presented illustrating the

Manuscript received September 2,1994; revised November 10, 1994. This work was supported by the Computer-Aided Engineering Chairs funded by Bell-Northen Research and the Natural Sciences and Engineering Research Council of Canada (NSERC), and by NSERC Research Grants.

The authors are with the Department of Electronics, Carleton University, Ottawa, Ont., Canada K1S 5B6.

IEEE Log Number 9410697.

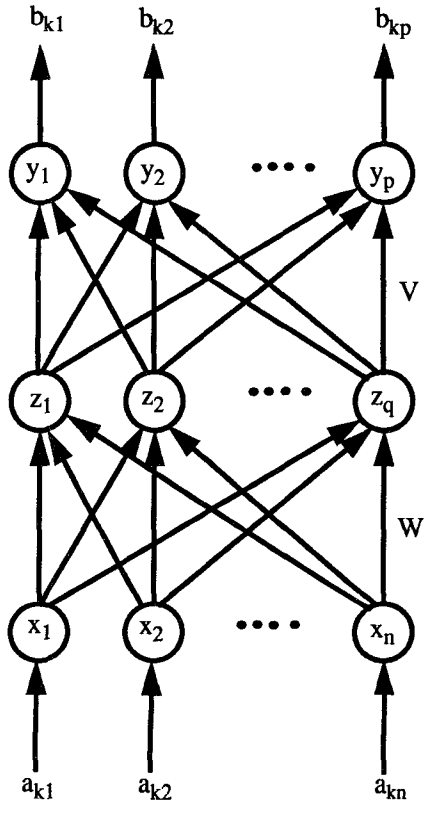

Fig. 1. A three-layer neural network.

advantages of the proposed neural network approach compared to standard analysis and optimization.

II. Structure of the Model

A neural network is a simplified mathematical model of a biological neural network. It consists of a collection of interconnected neurons. Let $\boldsymbol{x}$ be an n-vector containing parameters of a given device or a circuit, e.g., gate length and width of a FET or geometrical and physical parameters of high-speed VLSI interconnects [24], etc. Let $\boldsymbol{y}$ be a p-vector representing various responses of the device or the circuit under consideration, e.g., drain current of a FET. The relationship between $\boldsymbol{x}$ and response $\boldsymbol{y}$ is multidimensional and nonlinear.

To model such a nonlinear relationship, a multilayer neural network is employed. We use a three-layer neural network with n processing elements (PE) in the input layer, p PE's in the output layer, and q PE's in the hidden layer, as shown in Fig. 1. The input and output layers correspond to device or circuit parameters $\boldsymbol{x}$ and output responses $\boldsymbol{y}$, respectively. The hidden layer is represented by a q-vector $\boldsymbol{z}$. Let

$$\boldsymbol{a}_k = [a_{k1} \quad a_{k2} \cdots a_{kn}]^T \tag{1}$$

and

$$\boldsymbol{b}_k = [b_{k1} \quad b_{k2} \cdots b_{kp}]^T \tag{2}$$

be vectors representing the kth sample of the inputs and outputs, respectively, $k = 1, 2, \cdots, N$, where N is the total number of data samples. The weighting factors between the input and the hidden layers are w_{ih}, and between the hidden and the output layers are v_{hj}, where $i = 1, 2, \cdots, n$; $h = 1, 2, \cdots, q$, and $j = 1, 2, \cdots, p$. The output from the neural network can be computed as

$$y_j = \sum_{h=1}^{q} z_h v_{hj} \tag{3}$$

where z_h is a function defined as

$$z_h = f(\gamma_h) = \frac{1}{1 + e^{-\gamma_h}} \tag{4}$$

$$\gamma_h = \left(\sum_{i=1}^{n} a_{ki} w_{ih}\right) + \theta_h = \left(\sum_{i=1}^{n} x_i w_{ih}\right) + \theta_h \tag{5}$$

and where θ_h is a threshold value for the hth hidden PE. Theoretically, this model can approximate any nonlinear relationship [16]. The parameters of the model are the weighting factors w_{ih}, v_{hj}, and thresholds θ_h. The total number of these parameters is $n \times q + p \times q + q$. The size of the model, i.e., the number of model parameters, ultimately depends on the degree of nonlinearity of the problem. It does not grow exponentially when n, the number of inputs, is increased. Therefore, the model can work in high dimension.

III. Determination of Model Parameters

A. Neural Network Training

The neural network learns from samples of input–output data, i.e., $\boldsymbol{a}_k$ and $\boldsymbol{b}_k, k = 1, 2, \cdots, N$, where N is the total number of samples. The n-input parameters $\boldsymbol{a}_k$ could be physical/geometrical parameters of a FET device or circuit parameters. The p-outputs from the neural network $\boldsymbol{b}_k$ represent the electrical device parameters. The learning algorithm we used is based on multilayer error-correction learning, also called backpropagation [15]–[18]. During learning, the neural network automatically adjusts its weights and thresholds (i.e., w_{ih}, v_{hj}, and θ_h) so that the error E between neural network predicted y_j and sampled outputs b_{kj}

$$E = \sum_{k=1}^{N} E^k = \sum_{k=1}^{N} \left[\frac{1}{2} \sum_{j=1}^{p} (y_j - b_{kj})^2\right] \tag{6}$$

is minimized. This learning procedure is also called training.

There are two types of training algorithms: "on-line" training where neural network parameters are updated after each sample presentation, and "off-line" training where neural network parameters are updated after all samples are presented. In this work, we choose the "on-line" training approach since it is more efficient in most cases. The update equations are

$$v_{hj}^{k+1} = v_{hj}^{k} - \eta \frac{\partial E^k}{\partial v_{hj}} + \alpha (v_{hj}^{k} - v_{hj}^{k-1}) \tag{7}$$

$$w_{ih}^{k+1} = w_{ih}^{k} - \eta \frac{\partial E^k}{\partial w_{ih}} + \alpha (w_{ih}^{k} - w_{ih}^{k-1}) \tag{8}$$

and

$$\theta_{h}^{k+1} = \theta_{h}^{k} - \eta \frac{\partial E^k}{\partial \theta_{h}} + \alpha (\theta_{h}^{k} - \theta_{h}^{k-1}) \tag{9}$$

where η and α are positive-valued learning rate and momentum, respectively. The sensitivity through the neural network is computed as

$$\begin{aligned}\frac{\partial E^k}{\partial v_{hj}} &= \frac{\partial}{\partial v_{hj}}\left[\frac{1}{2}\sum_{j=1}^{p}(y_j - b_{kj})^2\right] \\ &= (y_j - b_{kj})z_h \\ &= \delta_j^{(3)} z_h \end{aligned} \tag{10}$$

$$\begin{aligned}\frac{\partial E^k}{\partial w_{ih}} &= \sum_{j=1}^{p}\frac{\partial E^k}{\partial y_j}\frac{\partial y_j}{\partial z_h}\frac{\partial z_h}{\partial \gamma_h}\frac{\partial \gamma_h}{\partial w_{ih}} \\ &= \sum_{j=1}^{p}(y_j - b_{kj})v_{hj}z_h(1-z_h)a_{ki} \\ &= z_h(1-z_h)\sum_{j=1}^{p}\delta_j^{(3)}v_{hj}a_{ki} \\ &= \delta_h^{(2)}a_{ki} \end{aligned} \tag{11}$$

and

$$\begin{aligned}\frac{\partial E^k}{\partial \theta_h} &= \sum_{j=1}^{p}\frac{\partial E^k}{\partial y_j}\frac{\partial y_j}{\partial z_h}\frac{\partial z_h}{\partial \gamma_h}\frac{\partial \gamma_h}{\partial \theta_h} \\ &= \sum_{j=1}^{p}(y_j - b_{kj})v_{hj}z_h(1-z_h) \\ &= z_h(1-z_h)\sum_{j=1}^{p}\delta_j^{(3)}v_{hj} \\ &= \delta_h^{(2)} \end{aligned} \tag{12}$$

where $\delta_h^{(2)}$ and $\delta_j^{(3)}$ represent "local gradients" at individual neuron in the second and third layers, respectively. The sample data $(\boldsymbol{a}_k, \boldsymbol{b}_k)$ can be obtained by device or circuit simulations done off-line, or obtained directly from measurement. The model parameters are then the final set of values w_{ih}, v_{hj}, and θ_h.

B. Training Algorithm

Our training alogorithm is based on the backpropagation technique [15]. Our modifications to the original backpropagation includes a learning rate and momentum adaptation in order to improve the speed of convergence:

Step 1: Choose the number of hidden neurons q and initialize the weights w_{ih}, v_{hj}, and thresholds θ_h with small random numbers. Choose initial values for η and α.

Step 2: Set $k = 1$.

Step 3: Supply training sample $(\boldsymbol{a}_k, \boldsymbol{b}_k)$, let $\boldsymbol{x} = \boldsymbol{a}_k$.

Step 4: Forward propagation:

- Compute the network's output $\boldsymbol{y}$ following (3)–(5).

Step 5: Back propagation of the error:

- Compute the error E^k given in (6), and the gradients $\partial E^k/\partial v_{hj}$, $\partial E^k/\partial w_{ih}$, and $\partial E^k/\partial \theta_h$ in (10)–(12);
- Adjust the parameters of the network using (7)–(9).

Step 6: $k = k + 1$ if $k \leq N$, where N is the total number of training samples, go to Step 3.

Step 7: Compute the cumulative error E.

Step 8: If the cumulative error E is less than a given training tolerance ϵ, stop the training process:

Step 9: If E is larger than its previous value then: decrease learning rate and momemtum, i.e., $\eta = \gamma \times \eta$ and $\alpha = \gamma \times \alpha$, go to Step 2.

Step 10: If E decreases then: increase learning rate and momentum, i.e., $\eta = 1/\gamma \times \eta$ and $\alpha = 1/\gamma \times \alpha$, go to Step 2.

C. Training Parameters

The efficiency of training depends on the following training parameters:

- Number of hidden layers: It has been theoretically proved that a multilayer neural network with at least one hidden layer can model arbitrarily complex nonlinear input/output relationship. In this work the total number of layers is fixed to three, i.e. only one hidden layer.
- Number of hidden neurons q: Once the number of hidden layers is fixed to one, the number of neurons in the hidden layer will determine the structure of our network. A large number of hidden neurons is required to model complicated relationships. But too many can result in an overtrained network. An overtrained network tends to memorize rather than to generalize from data.
- Learning rate η: This parameter determines the speed of convergence by regulating the step size.
- Momentum α: The momentum term is to prevent the training algorithm from settling in local minima. It also increases the speed of convergence. This parameter is usually set to a positive value less than 1.
- Training tolerance ϵ: This critical learning parameter determines the accuracy of the neural network outputs. A smaller training tolerance usually increases learning accuracy but can result in less generalization capability as well as longer training time.
- Learning rate adaptation γ: An adaptive learning rate decreases training time by keeping the learning rate reasonably high while insuring stability.

The optimal values of the parameters $q, \eta, \alpha, \epsilon$, and γ are problem-dependent and are obtained usually from experiment. Actual values of all these training parameters are given in Example 1 of Section VI.

IV. Use of the Model

A. Circuit Representation of the Neural Network Model

In order to connect the neural network model to a simulator, we need first a circuit representation of the model consistent with the device under consideration. In the case of a MESFET, output parameters are the gate, drain, and source currents I_{gc}, I_{dc}, and I_{sc} and the total charges Q_g, Q_d, and Q_s on the gate, drain, and source electrodes, respectively. The neural network model will have then an output layer with six

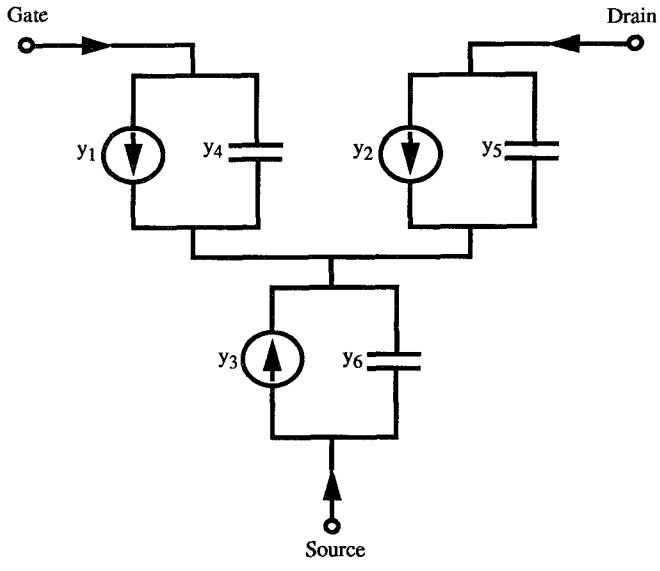

Fig. 2. Circuit representation of the six-output parameters at the neural network output layer, where $y = [I_{gc}, I_{dc}, I_{sc}, Q_g, Q_d, Q_s]^T$.

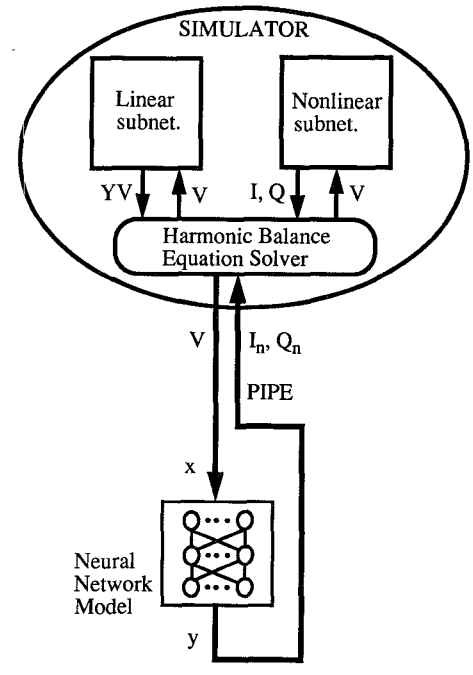

Fig. 3. Implementation of neural network models into circuit simulator.

parameters, i.e.

$$\begin{aligned} \boldsymbol{y} &= [y_1, y_2, y_3, y_4, y_5, y_6]^T \\ &= [I_{gc}, I_{dc}, I_{sc}, Q_g, Q_d, Q_s]^T. \end{aligned} \tag{13}$$

A circuit representation of this six-output neural network model is shown in Fig. 2.

Use of the model corresponds to the recalling mode, where the neural network will predict output responses from given input of device parameters. Two cases are considered in this paper, simulation and optimization.

B. Circuit Simulation

Our choice in this work is the use of harmonic balance method (HBM) for steady-state analysis of nonlinear periodic circuits. However, the technique can be applied to transient analysis as well. The HBM is an efficient tool for the simulation of nonlinear microwave circuits, e.g., [25]–[27]. In the HBM, the circuit is divided into linear and nonlinear subnetworks. This makes it simple to include the neural network model as an additional nonlinear subnetwork as shown in Fig. 3. In other words, when the model is used to model an active or passive element, it enters the overall circuit harmonic balance equation as

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{I}(\boldsymbol{V}) + j\boldsymbol{\Omega}\boldsymbol{Q}(\boldsymbol{V}) + \boldsymbol{I_n}(\boldsymbol{V}) + j\boldsymbol{\Omega}\boldsymbol{Q_n}(\boldsymbol{V}) \\ + \boldsymbol{YV} + \boldsymbol{I_{ss}} = 0 \end{aligned} \tag{14}$$

where $\boldsymbol{Y}$ is the nodal admittance matrix that describes the linear subnetwork, $\boldsymbol{V}, \boldsymbol{I}, \boldsymbol{I_{ss}}$, and $\boldsymbol{Q}(\boldsymbol{V})$ are the vectors that contain the Fourier coefficients of the respective time-domain waveforms at each node and all harmonics, as defined in [1], $\boldsymbol{V}$ representing voltages in the circuit, $\boldsymbol{I}$ and $\boldsymbol{Q}(\boldsymbol{V})$ representing, respectively, currents and charges of the nonlinear subnetwork, and $\boldsymbol{I_{ss}}$ representing the sources. The vectors $\boldsymbol{I_n}(\boldsymbol{V})$ and $\boldsymbol{Q_n}(\boldsymbol{V})$ represent the Fourier coefficients of the currents and charges entering the nodes from the neural network model. $\boldsymbol{\Omega}$ is the angular frequency matrix. For example, when the neural network models a FET, $\boldsymbol{I_n}(\boldsymbol{V})$ and $\boldsymbol{Q_n}(\boldsymbol{V})$ are computed, respectively, from the Fourier transform of the time-domain currents y_1, y_2, and y_3, and charges y_4, y_5, and y_6, which are provided by the neural network from the given gate and drain voltages [see (13)]. Notice that solving this harmonic balance problem does not require repeated solutions of the device physics equations as needed in the standard approach of [1], [6]. Another type of analysis is Monte Carlo analysis where the circuit is repeatedly simulated with randomly generated device parameters. Again in this case the neural network approach speeds up analysis by replacing repeated solutions of device physics equations.

C. Circuit Optimization

Our approach allows the neural network inputs $\boldsymbol{x}$ as optimization variables, e.g., physical/geometrical parameters of the device or circuits. The circuit responses can be obtained from a circuit simulator solving (14), or directly from a neural network output when it models the overall circuit. Let $\boldsymbol{\phi}$ be a vector of design variables and the set of error functions $e_j(\boldsymbol{\phi}), j = 1, 2, \cdots, m$, be the weighted difference between circuit responses and design specifications. The performance optimization problem can be posed as

$$\underset{\phi}{\text{Minimize}} \max\{e_1, e_2, \cdots, e_m\} \tag{15}$$

subject to electrical or physical/geometrical constraints on the circuit elements.

Let the nominal values of the circuit variables be $\boldsymbol{\phi}^0$. A number of random outcomes $\boldsymbol{\phi}^k, k = 1, 2, \cdots$, are generated around the nominal point $\boldsymbol{\phi}^0$ according to the statistical distributions of these parameters. Yield is defined as the ratio between the number of circuit outcomes passing design

TABLE I
RANGES OF NEURAL NETWORK INPUT PARAMETERS

Parameters	Notation	Range
Gate Length	L	0.9 - 1.1 μm
Gate Width	W	270 - 330 μm
Channel Thickness	a	0.27 - 0.33 μm
Doping Density	N_d	$8 \times 10^{22} - 12 \times 10^{22}$ $1/m^3$
Gate Voltage	V_G	-5.025 - 0 V
Drain Voltage	V_D	0 - 6 V

TABLE II
NEURAL NETWORK TRAINING PARAMETERS

Parameter	Notation	Value
No. of neurons in input layer	n	6
No. of neurons in output layer	p	4
No. of neurons in hidden layer	q	100
No. of samples	N	1000
Learning rate	η	0.2
Momentum	α	0.6
Training tolerance	ϵ	4.09e-02
Learning rate adaptation	γ	0.8

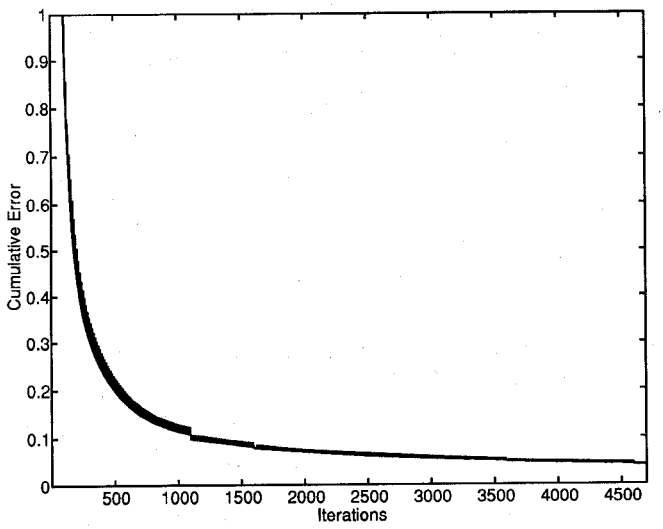

Fig. 4. Neural network learning curve.

specifications and the total number of outcomes. Numerical optimization is used to find the design center ϕ^0 such that the yield is maximized [3]–[5]. The present work is based on a generalized l_1 formulation of the problem [3], [4], i.e.

$$\underset{\phi^0}{\text{Minimize}}\, U(\phi^0) \equiv \sum_{k \in K} \alpha_k u(\phi^k) \tag{16}$$

where the index set K is

$$K = \{k | u(\phi^k) \geq 0\} \tag{17}$$

and $u(\phi)$ is a generalized l_p function of $e_i(\phi), i = 1, 2, \cdots, m$ and α_k is a properly chosen weighting factor.

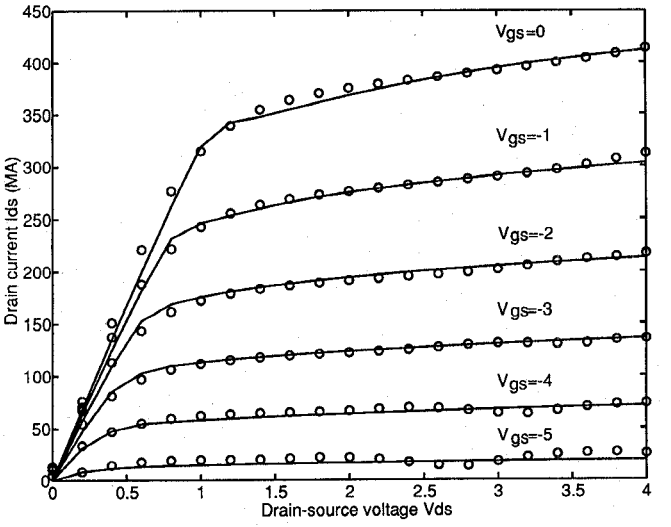

Fig. 5. Comparison of the dc characteristics using neural network model (o) with that of Khatibzadeh and Trew models (−).

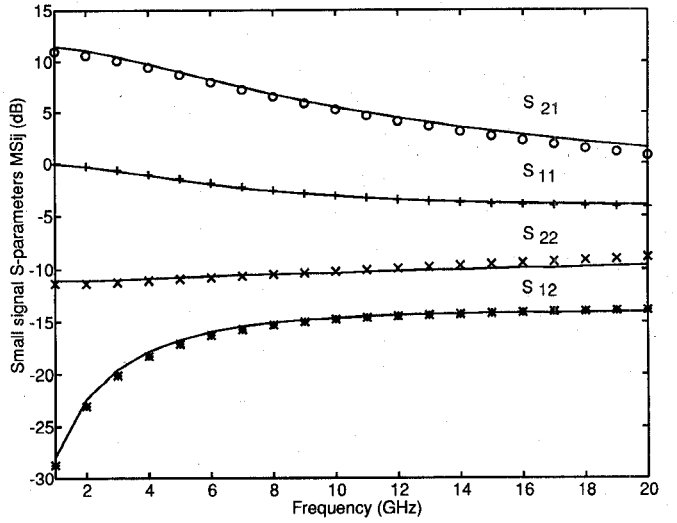

Fig. 6. Small-sigmal S-parameters (magnitude) comparison. (o, +, $*$, and $\times$) represent neural network results. (−) represents Khatibzadeh and Trew models.

V. IMPLEMENTATION

Circuit simulator solving the harmonic balance equations (14) with neural network models is implemented through the OSA90/Hope [28] CAD system, which provides combined dc/small/large signal analysis. The neural network was first trained off-line using sample data. The trained neural network model is then combined with the CAD system for analysis, optimization, Monte Carlo simulation and yield optimization of microwave circuits. The structure of various modules connected through UNIX pipe facilities is shown in Fig. 3. The pipe transfers input parameter values from the simulator to the neural network program and reads back the neural network calculated output parameters. The neural network does not have to be retrained during simulation or optimization, thus speeding up on-line analysis and optimization. According to (3)–(5) the evaluation of outputs from the neural network model is extremely fast.

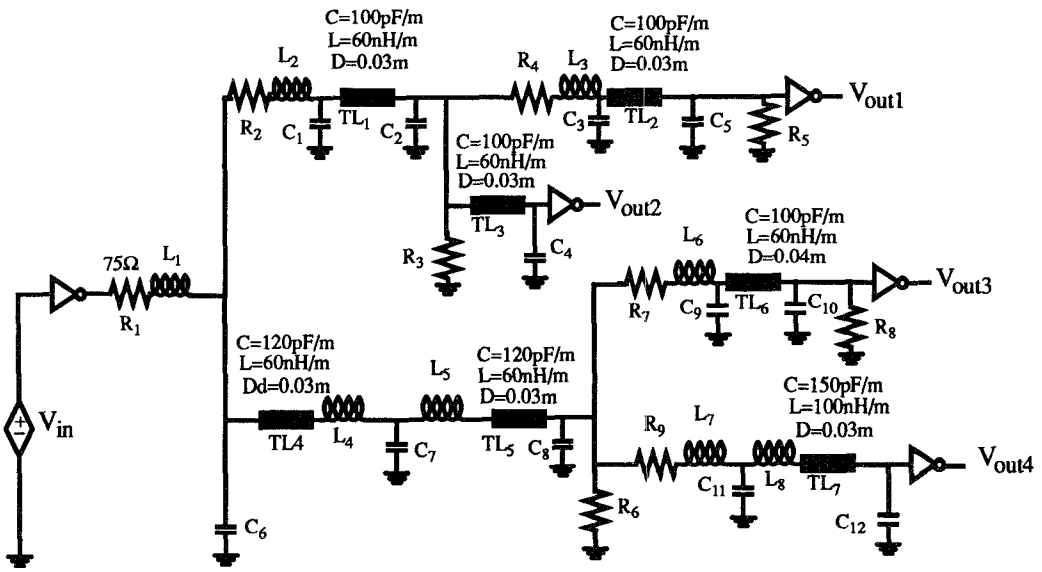

Fig. 7. A high-speed VLSI interconnect network represented by a seven transmission line circuit with nonlinear terminations.

TABLE III
VARIABLES FOR NOMINAL DESIGN

Design Variable	Before Optimization	After Optimization	Design Variable	Before Optimization	After Optimization
$S_{C1}(\mu m^2)$	353.1	326.8	n_{L4}	3.68	3.49
$S_{C2}(\mu m^2)$	2014.4	2001.5	n_{L5}	2.13	2.31
$S_{C3}(\mu m^2)$	212.3	224.6	n_{L6}	2.61	2.47
$S_{C4}(\mu m^2)$	354.2	343.8	n_{L7}	2.42	2.74
n_{L1}	3.06	3.50	n_{L8}	2.45	2.47
n_{L2}	3.56	3.76	n_{L9}	2.88	2.71
n_{L3}	2.84	2.91	n_{L10}	3.09	2.98

VI. EXAMPLES

A. Example 1—Physics-Oriented Neural Network Model of a MESFET

Physics-based device models are very CPU intensive specially when used for optimization or iterative simulations. A neural network model for this kind of devices will be very efficient in speeding up the simulation and optimization. The physical FET model chosen is the Khatibzadeh and Trew model [6]. Fig. 2 shows the circuit representation of the neural network model outputs, where I_{gc}, I_{dc}, and I_{sc} are the gate, drain, and source conduction currents, respectively. Q_g, Q_d, and Q_s stand for the total charges on the gate, drain, and source electrodes, respectively.

A three-layer neural network is used to model this FET. The input vector $\boldsymbol{x}$ for the neural network has six parameters including physical parameters: gate length L, gate width W, channel thickness a, and doping density N_d, and the gate–source and drain–source voltages V_{gs} and V_{ds}. To train the neural network each input parameter is allowed to vary over a certain range, as specified in Table I. Typical values of the neural network training parameters described in Section III are summarized in Table II. The learning curve, also called cost function [29], for this model is shown in Fig. 4.

We use new data different from the learning samples for verification of the neural network model. DC and small-signal S-parameter analysis predicted with our trained neural network model are compared to those simulated using the original Khatibzadeh and Trew Model, in Figs. 5 and 6, respectively.

B. Example 2—Transmission Line Circuit with Nonlinear Terminations

In this example, we demonstrate a different type of neural network model. Instead of modeling a device or a circuit element, we model the circuit responses of an entire circuit. Fig. 7 represents a high-speed VLSI interconnect network modeled by seven transmission lines and five nonlinear driver/receivers. Signal delay through such interconnect network is an important criterion in high-speed VLSI system design [24]. However, repeated signal delay analysis of this circuit is CPU intensive if done using conventional circuit simulators such as Spice. We choose six termination variables including capacitors, inductors, and resistors at the four terminations as input vector $\boldsymbol{x}$ for the neural network. The signal integrity responses $\boldsymbol{y}$

include the signal propagation delay of V_{out1} through V_{out4}. The number of hidden PE's in the neural network $q = 30$. A comparison of the four signal integrity responses predicted by the trained neural network with those from HSpice [30] was made for 100 sets of randomly generated samples of termination parameters which were not used for training. The result of such comparison is plotted in Fig. 8. The agreement of the neural network prediction with HSpice was generally within $\pm 0.2\%$.

This example illustrates the flexibility and generality of neural networks. Without changing equations and structures, neural networks are able to model not only devices but also circuits. Training enables neural networks to learn from different relationships.

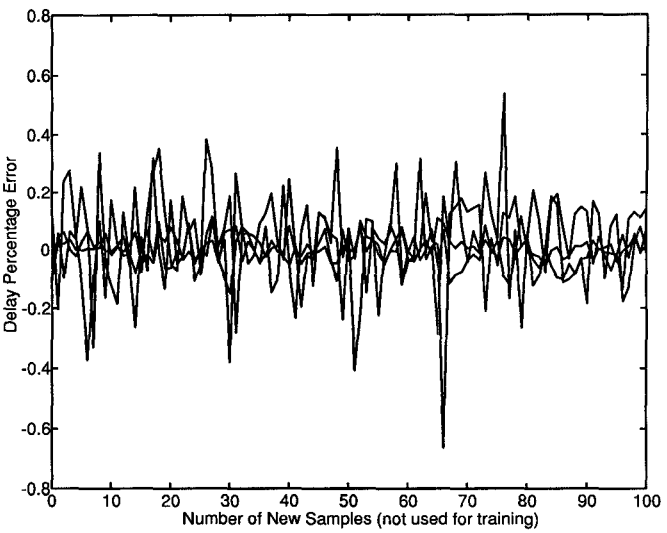

Fig. 8. The seven-transmission-line example. Percentage errors between signal delays predicted from the neural network model and that from exact simulation for 100 randomly generated sets of samples not used for training.

C. Example 3—Yield Optimization of a Three-Stage X-Band MMIC Amplifier

We consider a three-stage small-signal X-band cascadable MMIC (Monolithic Microwave Integrated Circuits) amplifier [1] shown in Fig. 9. The design is based on the circuit topology described in [31]. The amplifier contains three MESFET's. The matching circuits are composed of inductors and capacitors arranged in bandpass topology.

Physics-based models are used for both the MESFET's and passive elements of the amplifier. In this way, all the passive components, as well as active devices, can be simulated and optimized in terms of physical parameters. Since all devices are made from the same material and on the same wafer, they share common parameters. All three MESFET's have the same values for the critical electric field, saturation velocity, relative permittivity, built-in potential, low-field mobility, and high-field duffusion coefficient [1]. Thus the same neural network model, developed in Example 1, is used for all three MESFET's. All the MIM (metal–insulator–metal) capacitors have the same dielectric film, and all bulk resistors have the same sheet resistance. The geometrical parameters, on the other hand, can have different values for different devices, including the gate length, gate width, channel thickness, and doping density of the MESFET's, the metal-plate area of the MIM capacitors, and the number of turns of the spiral inductors. In other words, the neural network model is trained only once but is called three times, each time with a different set of input parameters $L^i, W^i, a^i, N_d^i, V_{gs}^i$, and V_{ds}^i, corresponding to MESFET$^i, i = 1, 2, 3$.

The specifications for the amplifier circuit include

- Passband (8–12 GHz): 12.4 dB $\leq$ gain $\leq$ 15.6 dB, input VSWR $\leq$ 2.8.
- Stopband (below 6 GHz or above 15 GHz): gain $\leq$ 2 dB.

There are 14 design variables, the area $S_{C1}, \cdots, S_{C4}$ of the metal plates of the MIM capacitors $C_1, \cdots, C_4$ and the number of turns $n_{L1}, \cdots, n_{L10}$ of the spiral inductors $L_1, \cdots, L_{10}$. As a first step, a nominal design optimization using neural network was carried out reducing the objective function of (15) from 6.7 to -0.15, all specifications being satisfied. Table III lists the 14 design variables before and after minimax optimization. In Fig. 10, the gain and input VSWR of the amplifier using neural network models before and after optimization are compared. To verify the optimization solution, the same parameters in Table III were used to simulate the X-band amplifier with the Khatibzadeh and Trew model, which is a much more complex model. We found all specifications being satisfied as illustrated in Fig. 11.

In the second step, yield optimization using l_1 centring algorithm as described in Section IV is performed with the minimax nominal design as a starting point. There are 37 statistical variables including the neural network inputs gate length, gate width, channel thickness, and doping density of the MESFET's, as well as the geometrical parameters of the passive elements, namely, the conductor width W_L and spacing S_L of the ten spiral inductors $L_1, L_2 \cdots, L_{10}$, the thickness d of the dielectric film for all MIM capacitors, and the area $S_{C1}, \cdots, S_{C4}$ of the metal plates of the MIM capacitors $C_1, \cdots, C_4$. The distributions for these 37 statistical variables are listed in Table IV. The correlation matrix between the three sets of MESFET parameters in [1] is used. The yield after minimax nominal design optimization was 26% with the neural network model and 32% with the Khatibzadeh and Trew model. The CPU time used for the Monte Carlo sweeps was 1 h and 30 min for the neural network approach and 40 h 34 min for the Khatibzadeh and Trew model, i.e., our approach is about 30 times faster. At the solution of yield optimization using neural network, the yield was improved to 58%. To verify this solution, we performed Monte Carlo analysis using Khatibzadeh and Trew model, the yield was 59%. Thus validity of the neural network approach was confirmed. The solution is given in Table V. The Monte Carlo sweeps before and after yield optimization are shown in Fig. 12. Yield optimization with 50 outcomes using neural network model took 50 min CPU time per iteration on a Sun SPARCstation 2. The corresponding CPU time using the Khatibzadeh and Trew model with quadratic model [10] is 4 h and 14 min. Table VI summarizes the CPU speedup achievement.

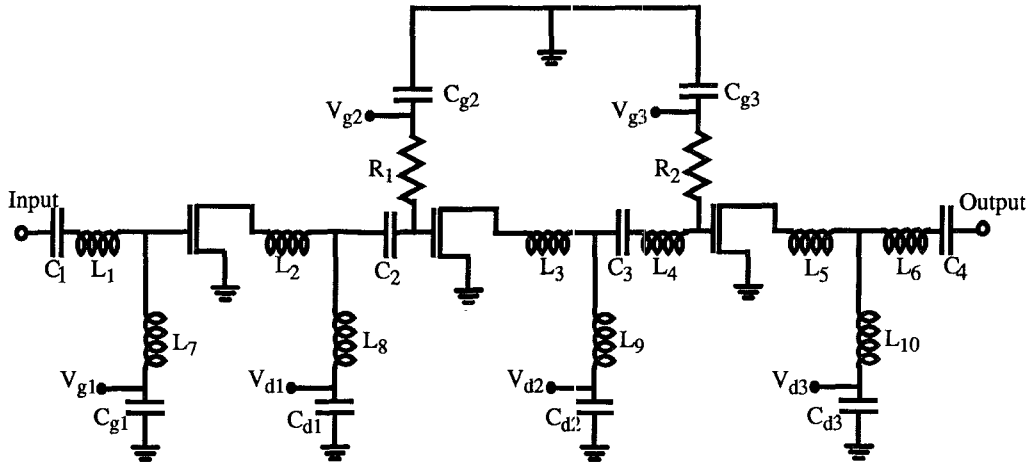

Fig. 9. Circuit diagram of an X-band amplifier.

TABLE IV
DISTRIBUTIONS FOR STATISTICAL VARIABLES, AFTER [1]

Variable	Mean	Deviation (%)	Variable	Mean	Deviation (%)
$N_d(1/m^3)$	1.0×10^{23}	7.0	d (μm)	0.1	4.0
$L(\mu m)$	1.0	3.5	$S_{C1}(\mu m^2)$	326.8	3.5
$a(\mu m)$	0.3	3.5	$S_{C2}(\mu m^2)$	2022.4	3.5
$W(\mu m)$	300	2.0	$S_{C3}(\mu m^2)$	218.2	3.5
$W_L(\mu m)$	20	3.0	$S_{C4}(\mu m^2)$	352.2	3.5
$S_L(\mu m)$	10	3.0			

TABLE V
DESIGN VARIABLES FOR YIELD OPTIMIZATION

Design Variable	Before Optimization	After Optimization	Design Variable	Before Optimization	After Optimization
$S_{C1}(\mu m^2)$	272.8	232.2	n_{L4}	3.49	3.58
$S_{C2}(\mu m^2)$	2001.5	2006.9	n_{L5}	2.31	2.38
$S_{C3}(\mu m^2)$	244.4	277.8	n_{L6}	2.47	2.49
$S_{C4}(\mu m^2)$	343.8	346.1	n_{L7}	2.74	2.72
n_{L1}	3.50	3.55	n_{L8}	2.47	2.49
n_{L2}	3.76	3.73	n_{L9}	2.71	2.73
n_{L3}	2.91	2.99	n_{L10}	2.98	3.00

TABLE VI
SUMMARY OF CPU COMPARISON

Application	Khatibzadeh & Trew Model	Neural Network Model	Speed-up ratio
Optimization	7 min 8 sec	1 min 6 sec	6
Monte Carlo	40 hours 34 min	1 hour 30 min	30
Yield Optimization	4 hours 14 min	50 min	5

VII. CONCLUSION

In this paper we have presented a nontraditional approach to microwave circuit analysis, optimization, and statistical design featuring neural network models. The results from our work have demonstrated the feasibility and the efficiency of using neural networks for physics-based device modeling. A systematic description of neural network and its integration with circuit simulations has been presented.

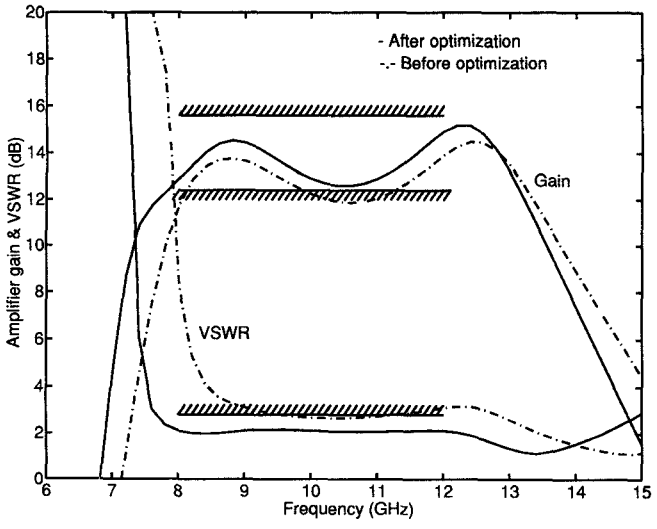

Fig. 10. Gain and input VSWR of the X-band amplifier with neural network models before (– · –) and after (—) nominal design optimization.

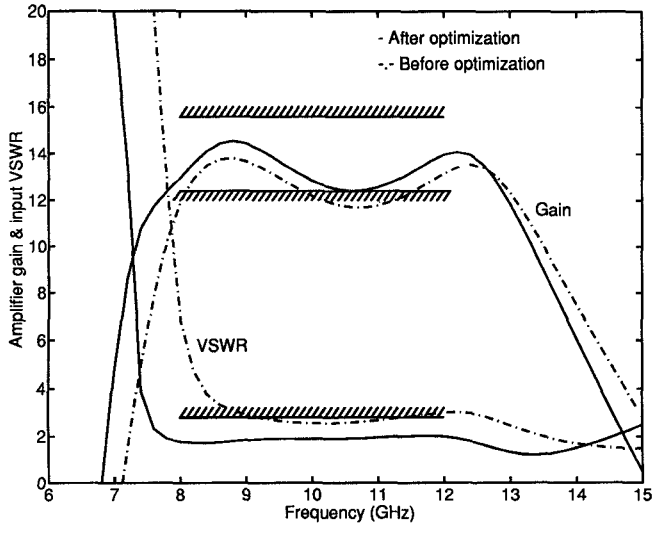

Fig. 11. Gain and input VSWR of the X-band amplifier with Khatibzadeh and Trew models before (– · –) and after (—) nominal design optimization.

By exploiting the flexibility and generality of the neural network model, we have demonstrated its use for device and circuit-level modeling as well. Even though a neural network model has no embeded electrical or physics equations, we have shown its capability to relate the circuit outputs to parameters at any level, e.g., electrical, physical, or both. In addition, its capability of learning from abstract data means it has the potential to model different types of devices without changing formulas.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. W. Bandler of McMaster University for facilitating the generation of training data for FET models. The examples were done under OSA90/Hope [28] environment.

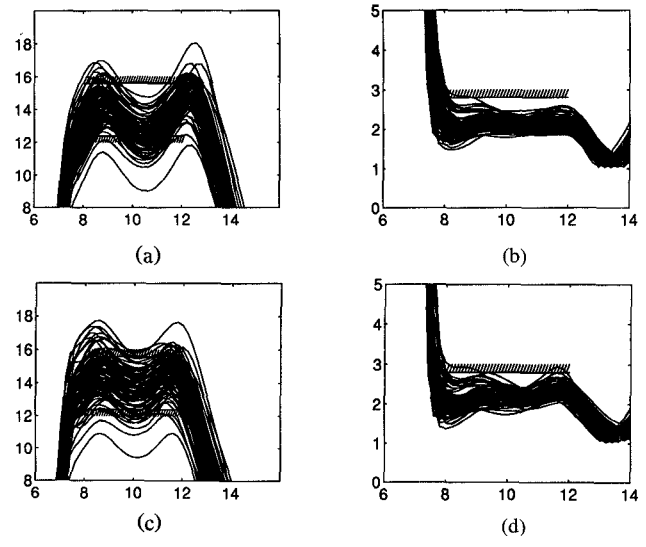

Fig. 12. Monte Carlo sweep of gain and input VSWR versus frequency (GHz) of the X-band amplifier using neural network models before optimization (a) gain (b) input VSWR, and after optimization (c) gain (d) input VSWR.

REFERENCES

[1] J. W. Bandler, R. M. Biernacki, Q. Cai, S. H. Chen, S. Ye, and Q. J. Zhang, "Integrated physics-oriented statistical modelling, simulation and optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1374–1400, 1992.

[2] D. E. Stoneking, G. L. Bilbro, P. A. Gilmore, R. J. Trew, and C. T. Kelley, "Yield optimization using a GaAs process simulator coupled to a physical device model," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1353–1363, 1992.

[3] J. W. Bandler and S. H. Chen, "Circuit optimization: The state-of-the-art," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 424–443, 1988.

[4] J. W. Bandler, Q. J. Zhang, J. Song, and R. M. Biernacki, "FAST gradient based yield optimization of nonlinear circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 38, pp. 1701–1710, 1990.

[5] J. Purviance and M. Meehan, "CAD for statistical analysis and design of microwave circuits," *Int. J. Microwave and Millimeter-Wave Computer-Aided Eng.*, vol. 1, pp. 59–76, 1991.

[6] M. A. Khatibzadeh and R. J. Trew, "A large-signal, analytical model for the GaAs MESFET," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 231–238, 1988.

[7] F. Filicori, G. Ghione, and C. U. Naldi, "Physics-based electron device modeling and computer-aided MMIC design," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1333–1352, 1992.

[8] J. A. Barby, J. Vlach, and K. Singhal, "Polynomial splines for MOSFET model approximation," *IEEE Trans. Computer-Aided Design*, vol. 7, no. 5, pp. 557–567, 1988.

[9] K. K. Low and S. W. Director, "A new methodology for the design centring of IC fabrication process," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 895–903, 1991.

[10] R. M. Biernacki, J. W. Bandler, J. Song, and Q. J. Zhang, "Efficient quadratic approximation for statistical design," *IEEE Trans. Circuits Syst.*, vol. 36, no. 11, pp. 1449–1454, 1989.

[11] N. Jain, D. Agnew, and M. Nakhla, "Two-dimensional table lookup MOSFET model," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, CA, 1983, pp. 201–213.

[12] P. B. L. Meijer, "Fast and smooth highly nonlinear multidimensional table models for device modeling," *IEEE Trans. Circuits Syst.*, vol. 37, no. 3, pp. 335–346, 1990.

[13] D. M. Lewis, "Device model approximation using 2^n trees," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 1, pp. 30–38, 1990.

[14] M. G. Graham and J. J. Paulos, "Interpolation of MOSFET table data in width, length, and temperature," *IEEE Trans. Computer-Aided Design*, vol. 12, no. 12, pp. 1880–1884, 1993.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal

representations by error propagation" in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol.1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986.
[16] P. K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. Elmsford, NY: Pergamon, 1990.
[17] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoust., Speech, Signal Process. Mag.*, pp. 2–22, 1987.
[18] E. Sanchez-Sinencio and C. Lau, Eds., *Artificial Neural Networks*. New York: IEEE Press, 1992.
[19] M. Vai and S. Prasad, "Automatic impedance matching with a neural network," *IEEE Microwave Guided Wave Lett.*, vol. 3, pp. 353–354, 1993.
[20] Y. Li, R. L. Mahajan, and J. Tong, "Design factors and their effect on PCB assembly yield-statistical and neural network predictive models," *IEEE Trans. Comp., Pack., Manuf. Technol.*, pt. A, vol. 17, no. 2, pp. 183–191, 1994.
[21] S. S. Han, S. A. Bidstrup, P. Kohl, and G. May, "Modeling the properties of PEECVD silicon dioxide films using optimized back-propagation neural networks," *IEEE Trans. Comp., Pack., Manuf. Technol.*, pt. A, vol. 17, no. 2, pp. 174–182, 1994.
[22] S. H. Huang and H. C. Zhang, "Artificial neural networks in manufacturing: Concepts, applications, and perspectives," *IEEE Trans. Comp., Pack., Manuf. Technol.*, pt. A, vol. 17, no. 2, pp. 212–228, 1994.
[23] A. H. Zaabab, Q. J. Zhang, and M. Nakhla, "Analysis and optimization of microwave circuits and devices using neural network models," in *IEEE Int. Microwave Symp. Dig.*, San Diego, CA, 1994, pp. 393–396.
[24] M. Nakhla and Q. J. Zhang, Eds., *Modeling and Simulation of High-Speed VLSI Interconnects*. Boston, MA: Kluwer, 1994.
[25] M. S. Nakhla and J. Vlach, "A piecewise harmonic-balance technique for determination of periodic response of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 85–91, 1976.
[26] V. Rizzoli and A. Neri, "State of the art and present trends in nonlinear microwave CAD techniques," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 343–365, 1988.
[27] R. J. Gilmore and M. B. Steer, "Nonlinear circuit analysis using the method of harmonic balance—A review of the art. Part II. Advanced concepts," *Int. J. Microwave Millimeter-Wave Computer-Aided Eng.*, vol. 1, pp. 159–180, 1991.
[28] OSA90/HOPE V2.0, Optimization Systems Associates Inc., Dundas, Ont., Canada.
[29] O. Chen and B. Sheu, "Optimization schemes for neural network training," in *Proc. IEEE Int. Conf. on Neural Networks*, Orlando, FL, 1994, pp. 817–822.
[30] *HSpice User's Manual*, H9001, Meta-Software Inc., Campbell, CA, 1990.
[31] C. Kermarrec and C. Rumelhard, "Microwave monolithic integrated circuits," in *GaAs MESFET Circuit Design*, R. Soares, Ed. Boston: Artech House, 1988, ch. 9.

A. Hafid Zaabab was born in Ain Beida, O. E. Bouaghi, Algeria, on February 15, 1967. He received the B.Eng. degree from INELEC, Boumerdes, Algeria, in 1990, and the M.A.Sc. degree from the University of Ottawa, Ottawa, Ont., Canada, in 1993, both in electrical engineering. He was awarded a CIDA scholarship for the academic years 1990–1991 and 1991–1992. Currently he is working towards the Ph.D. degree in electrical engineering at Carleton University, Ottawa, Ont., Canada.

He is a teaching assistant at Carleton University. His research interests include optimization techniques, high-speed VLSI CAD, and neural network modeling for microwave and interconnect circuit design.

Qi-jun Zhang (S'84–M'87) received the B.Eng. degree from the East China Engineering Institute, Nanjing, China, in 1982, and the Ph.D. degree from McMaster University, Hamilton, Ont., Canada, in 1987, both in electrical engineering.

He was with the Institute of Systems Engineering, Tianjin University, Tianjin, China, from 1982 to 1983. He was a research engineer with Optimization Systems Associates Inc., Dundas, Ont.,Canada, from 1988 to 1990. During 1989 and 1990 he was also an assistant professor of electrical and computer engineering (part time) at McMaster University. He joined the Department of Electronics, Carleton University, Ottawa, Ont., Canada, in 1990 where he is presently an assistant professor. His professional interests include all aspects of circuit CAD with emphasis on large-scale simulation and optimization, statistical design and modeling, parameter extraction, sensitivity analysis, and optimization of microwave circuits and high-speed VLSI interconnections. He is a coeditor of *Modeling and Simulation of High-Speed VLSI Interconnects* (Kluwer, 1994) and a contributor to *Analog for Computer-Aided Analysis and Diagnosis* (Marcel Dekker, 1988).

Dr. Zhang is the holder of the Junior Industrial Chair in CAE established at Carleton University by Bell-Northern Research and the Natural Sciences and Engineering Research Council of Canada.

Michel S. Nakhla (S'73–M'76–SM'88) received the B.Sc. degree in electronics and communications from Cairo University, Cairo, Egypt, in 1967 and the M.A.Sc. and Ph.D. degrees in electrical engineering from Waterloo University, Waterloo, Ont., Canada, in 1973 and 1975, respectively.

In 1975, he was a postdoctoral fellow at University of Toronto, Toronto, Ont., Canada. In 1976 he joined Bell-Northern Research, Ottawa, as a member of the scientific staff, where he became manager of the simulation group in 1980 and senior manager of the computer-aided engineering group in 1983. In 1988, he joined Carleton University, Ottawa, where he is curently a professor in the Department of Electronics. His research interests include computer-aided design of VLSI and communication systems, high-frequency interconnects, and synthesis of analog circuits. He is coauthor of *Asymptotic Waveform Evaluation* (Kluwer, 1993) and coeditor of *Modeling and Simulation of High-Speed VLSI Interconnects* (Kluwer, 1994).

Dr. Nakhla is currently the holder of the Computer-Aided Engineering Industrial Chair established at Carleton University by Bell Northern Research and the Natural Sciences and Engineering Research Council of Canada.