Neural-Based Dynamic Modeling of Nonlinear Microwave Circuits

Jianjun Xu, *Student Member, IEEE*, Mustapha C. E. Yagoub, *Member, IEEE*, Runtao Ding, and Qi-Jun Zhang, *Senior Member, IEEE*

***Abstract*—A neural network formulation for modeling nonlinear microwave circuits is achieved in the most desirable format, i.e., continuous time-domain dynamic system format. The proposed dynamic neural network (DNN) model can be developed directly from input–output data without having to rely on internal details of the circuit. An algorithm is developed to train the model with time or frequency domain information. Efficient representations of the model are proposed for convenient incorporation of the DNN into high-level circuit simulation. Compared to existing neural-based methods, the DNN retains or enhances the neural modeling speed and accuracy capabilities, and provides additional flexibility in handling diverse needs of nonlinear microwave simulation, e.g., time- and frequency-domain applications, single-tone and multitone simulations. Examples of dynamic modeling of amplifiers, mixer, and their use in system simulation are presented.**

***Index Terms*—Modeling, neural networks, nonlinear circuits, optimization, simulation.**

I. Introduction

ARTIFICIAL neural networks (ANNs) have recently been recognized as a useful tool for modeling and design optimization problems in RF and microwave computer-aided design (CAD) [1]–[3]. They have been successfully used in a variety of applications such as modeling and optimization of high-speed very large scale integration (VLSI) interconnects [4], coplanar waveguide (CPW) circuits [5], spiral inductors [6], EM optimization [7], global modeling [8], yield optimization [9], and circuit synthesis [10], [11]. Knowledge-based approaches combining microwave empirical or equivalent circuit models together with neural network learning have also been studied [7], [12], [13] to further improve the training efficiency and model reliability.

This paper addresses an important application of ANNs, i.e., application to nonlinear circuit modeling and design. This could be a significant area because of the increasing need for efficient CAD algorithms in high-level and large-scale nonlinear microwave design. The brute-force way is to use original detailed circuit representation for system level simulation, leading to accurate but extremely slow computation. Conventional approaches addressing this issue are, e.g., behavioral modeling [14], [15], equivalent circuit [16], and model reduction [17] techniques, among which the behavioral modeling approach is presently the major method for system simulation in industrial applications. Recently, several ANN-based methods have been proposed for nonlinear modeling [8], [9], [18]–[23]. These works demonstrated neural networks as a useful alternative to the conventional approaches. The ANN approach has the potential to learn the nonlinear behavior from measured or simulated input–output data, avoiding otherwise manual effort of developing equivalent circuit topology. Similarly, ANN also avoids the need of availability of original circuit equations as required in model reduction techniques. The universal approximation property of ANN provides a theoretical basis of representing the full analog solutions of the circuit, overcoming the accuracy limitations in conventional behavioral models. The evaluation of the ANN from input to output is very fast [24].

Among the existing nonlinear neural modeling approaches, most of the earlier methods are developed under FET modeling motivations [8], [9], [18], [19]. The hybrid circuit/neural network approach in [8], [18] assumes the existence of a good equivalent circuit of the transistor and uses neural network to provide equivalent circuit parameters. The Volterra-kernel-based approach [19] is formulated with first-order Volterra kernel, which are represented by the ANN. The I-Q (current-charge) neural model approach [9] uses an ANN to provide FET intrinsic current and charges from intrinsic terminal voltages and physical/geometrical parameters. These are some of the pioneering steps in ANN-based nonlinear modeling. However, they are not sufficient to address the difficulties in circuit level modeling such as amplifiers and mixers. For example, the hybrid method is limited by the availability and quality of the equivalent circuit models; the first-order Volterra kernal assumption in [19] which was sufficient under FET device senarios may not be suitable for nonlinear circuits. The I–Q model in [9] can represent high nonlinearities and is suitable for harmonic balance (HB) simulation. However, its use of static neural networks makes it suitable only for FET intrinsic modeling and not suitable for modeling higher order dynamic effects in large nonlinear circuits with many internal nodes.

Recently, several ANN methods were introduced with emphasis on nonlinear circuit modeling, such as neural-network-based behavioral model [20], [21], which uses harmonic information of the nonlinear circuit behavior, and discrete recurrent neural network [22], [23] approaches, which use time-delayed signals to help producing the model input–output relationship. The neural network method in [21] is formulated to overcome

Manuscript received April 6, 2002; revised August 26, 2002. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, and in part by Micronet—A Canadian Network Center of Excellence in Microelectronic Devices, Circuits and Systems.

J. Xu and Q.-J. Zhang are with the Department of Electronics, Carleton University, Ottawa, ON, Canada K1S 5B6 (e-mail: qjz@doe.carleton.ca).

M. C. E. Yagoub is with the School of Information Technology Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N.

R. Ding is with the School of Electronics and Information Engineering, Tianjin University, Tianjin 300072, China.

Digital Object Identifier 10.1109/TMTT.2002.805192

the limitations in conventional behavioral models by providing bidirectional behavior allowing more accurate system simulation. Recurrent neural network approach [22] achieves a discrete time-domain model based on backpropagation-through-time training to learn the circuit input–output relationship. These works represent important recent milestons in the direction of ANN-based nonlinear modeling. However, because of the specific formats of these existing methods, there still exist limitations due to difficulties in their incorporations in standard nonlinear simulators, in establishing relations with large-signal measurement, limited flexibility for different simulations, or potential curse of dimensionality in multitone simulations.

The most ideal format to describe nonlinear dynamic models for the purpose of circuit simulation is the continuous time-domain format, e.g., the popularly accepted dynamic current-charge format in many harmonic balance simulators. This format, in theory, best describes the fundamental essence of nonlinear behavior, and, in practice, is most flexible to fit most or nearly all needs of nonlinear microwave simulation, a task not yet achieved by the existing ANN-based techniques. In the neural network community, such types of networks have been studied, e.g., Hopefield network [25], recurrent network [26], etc. However, they were mainly oriented for digital signal processing such as binary-based image processing [26], or system control with online correction signals from a physical system [27]. They are not directly suitable for microwave modeling. We must address continuous analog signals and our CAD method must be able to predict circuit behavior offline.

For the first time, an exactly continuous time-domain dynamic-modeling method is formulated using neural networks for large-signal modeling of nonlinear microwave circuits and systems in this paper. The model, called dynamic neural network (DNN) model, can be developed directly from input–output data without having to rely on internal details of the circuits. An algorithm is described to train the model with time- or frequency-domain information. Efficient representations of the DNN are proposed such that the model can be conveniently incorporated into circuit simulators for high-level and large-scale nonlinear microwave design. The model can be standardized even with diverse requirements of nonlinear modeling such as single-tone and multitone applications, and training with time- or frequency-domain data.

This paper is organized as follows. In Section II, we formulate the new dynamic neural network modeling technique and propose a training method for training the DNN. Two approaches of incorporating the DNN models into circuit simulators are proposed, one through a circuit representation and another through an efficient harmonic-balance-based representation. In Section III, examples of dynamic modeling of amplifiers, mixer, and their use in system simulation are presented. Accuracy and speed of using the DNN versus using conventional approaches are compared.

II. DNN Modeling of Nonlinear Circuits: Formulation and Development

A. Original Circuit Dynamics

Let $\boldsymbol{u} = [u_1\, u_2 \ldots u_{N_u}]^T$ and $\boldsymbol{y} = [y_1\, y_2 \ldots y_{N_y}]^T$ be vectors of the input and the output signals of the nonlinear circuit, respectively, where N_u and N_y are the number of inputs and outputs. The original nonlinear circuit can be generally described in state equation form as

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{\varphi}(\boldsymbol{x}(t), \boldsymbol{u}(t))\\ \boldsymbol{y}(t) &= \boldsymbol{\psi}(\boldsymbol{x}(t), \boldsymbol{u}(t))\end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is an N_S vector of state variables and N_S is the number of states, $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ represent nonlinear functions. In a modified nodal formulation [28], the state vector $\boldsymbol{x}(t)$ includes nodal voltages, currents of inductors, currents of voltage sources, and charge of nonlinear capacitors.

For a circuit with many components, (1) could be a large set of nonlinear differential equations. For system-level simulation including many circuits, such detailed state equations are too large, computationally expensive, and sometimes even unavailable at system level. Therefore, a simpler (reduced order) model approximating the same dynamic input–output relationships is needed.

B. Formulation of DNN Model

Let n be the order of the reduced model, $n < N_S$. Let $\boldsymbol{y}^{(i)}(t) = d^i\boldsymbol{y}(t)/dt^i$ and $\boldsymbol{u}^{(i)}(t) = d^i\boldsymbol{u}(t)/dt^i$ denote the ith-order derivatives of $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ with respect to t, respectively. In order to derive a dynamic model, the original problem (1) is reformulated into reduced order differential equations using the input–output variables as

$$\begin{aligned}\boldsymbol{y}^{(n)}(t) = \boldsymbol{f}\Big(&\boldsymbol{y}^{(n-1)}(t), \boldsymbol{y}^{(n-2)}(t), \ldots, \boldsymbol{y}(t),\\ &\boldsymbol{u}^{(n)}(t), \boldsymbol{u}^{(n-1)}(t), \ldots, \boldsymbol{u}(t)\Big)\end{aligned} \tag{2}$$

where $\boldsymbol{f}$ represents nonlinear functions. In this paper, we propose to employ an ANN to represent the nonlinear relationships between the dynamic information of inputs and outputs. Let $\boldsymbol{v}_i$ be a N_y vector, $i = 1, 2, \ldots, n$. Let $\boldsymbol{f}_{\mathrm{ANN}}$ represent a multilayer perceptron neural network [1] with input neurons representing $\boldsymbol{y}, \boldsymbol{u}$, their derivatives $d^i\boldsymbol{y}/dt^i$, $i = 1, 2, \ldots, n-1$, and $d^k\boldsymbol{u}/dt^k$, $k = 1, 2, \ldots, n$; and the output neuron representing $d^n\boldsymbol{y}/dt^n$. The proposed DNN model is derived from (2) as

$$\begin{aligned}\dot{\boldsymbol{v}}_1(t) &= \boldsymbol{v}_2(t)\\ &\ \vdots\\ \dot{\boldsymbol{v}}_{n-1}(t) &= \boldsymbol{v}_n(t)\\ \dot{\boldsymbol{v}}_n(t) &= \boldsymbol{f}_{\mathrm{ANN}}\big(\boldsymbol{v}_n(t), \boldsymbol{v}_{n-1}(t), \ldots, \boldsymbol{v}_1(t),\\ &\qquad \boldsymbol{u}^{(n)}(t), \boldsymbol{u}^{(n-1)}(t), \ldots, \boldsymbol{u}(t)\big)\end{aligned} \tag{3}$$

and the inputs and outputs of the model are $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t) = \boldsymbol{v}_1(t)$, respectively.

The overall DNN model (3) is in a standardized format for typical nonlinear circuit simulators. For example, the left-hand side of the equation provides the charge (Q) or the capacitor part, and the right-hand side provides the current (I) part, which are the standard representation of nonlinear components in many harmonic balance (HB) simulators. The proposed DNN overcomes the limitations of the previous static I–Q neural model of [9] which was only suitable for intrinsic FETs. The proposed DNN can provide dynamic current-charge parameters for general nonlinear circuits with any number of internal nodes in original circuit. The order n (or the number of hidden neurons in $\boldsymbol{f}_{\text{ANN}}$) represents the effective order (or the degree of nonlinearity) of the original circuit that is visible from the input–output data. Therefore, the size of the DNN reflects the internal property of the original circuit rather than external signals and, as such, the model does not suffer from the curse of dimensionality in multitone simulation.

C. Model Training

Our DNN model will represent a nonlinear microwave circuit only after we train it with data from the original circuit. We use training data in the form of input/output harmonic spectrums, which can be obtained through simulation or measurements. Let $\overline{\boldsymbol{U}}(\omega)$ and $\overline{\boldsymbol{Y}}(\omega)$ be such input and output spectrums, respectively, $\omega \in \Omega$, where Ω is the set of spectrum frequencies. The training data are generated using a variety of input samples, leading to a set of data $\overline{\boldsymbol{U}}_m(\omega)$ and $\overline{\boldsymbol{Y}}_m(\omega)$, where m is the sample index, $m = 1, 2, \ldots, M$, and M is the total number of samples.

A second set of data, called testing data should also be obtained similarly from the original circuit for model verification. The testing data should be generated using a set of input samples different from those used in training data.

Initial Training: We first train the $\boldsymbol{f}_{\text{ANN}}$ part of the DNN model in the time domain directly or indirectly using time-domain information. Suppose matrix $\boldsymbol{A}(\omega, t)$ represents the coefficients of Inverse Fourier Transform [29]. Let the derivative of $\boldsymbol{A}(\omega, t)$ w.r.t time t be represented as

$$\boldsymbol{A}^{(i)}(\omega, t) = \frac{\partial \boldsymbol{A}^i(\omega, t)}{\partial t^i}. \tag{4}$$

The training data for $\boldsymbol{f}_{\text{ANN}}$ can be derived from

$$\overline{\boldsymbol{y}}_m^{(i)}(t) = \sum_{\omega \in \Omega} \boldsymbol{A}^{(i)}(\omega, t) \cdot \overline{\boldsymbol{Y}}_m(\omega) \tag{5}$$

$$\overline{\boldsymbol{u}}_m^{(i)}(t) = \sum_{\omega \in \Omega} \boldsymbol{A}^{(i)}(\omega, t) \cdot \overline{\boldsymbol{U}}_m(\omega). \tag{6}$$

The initial training is illustrated in Fig. 1. The objective of the training is to adjust ANN internal parameters to minimize the error function, as shown by (7), at the bottom of this page, where T is the set of time points used by Fourier Transform [29].

This process is computationally efficient (without involving harmonic balance simulation) and can train the $\boldsymbol{f}_{\text{ANN}}$ from a random (unknown) start to an approximate solution. Because all input–output information in each sample of training data is at the same instance of time, this proposed technique is completely free from restrictions on sampling frequencies, representing a clear advantage over the previous discrete recurrent neural network method [22].

Final Training: The DNN model is further refined using results from initial training as a starting point. Final training is done in the frequency domain involving HB solutions of the DNN model. The error function for training is

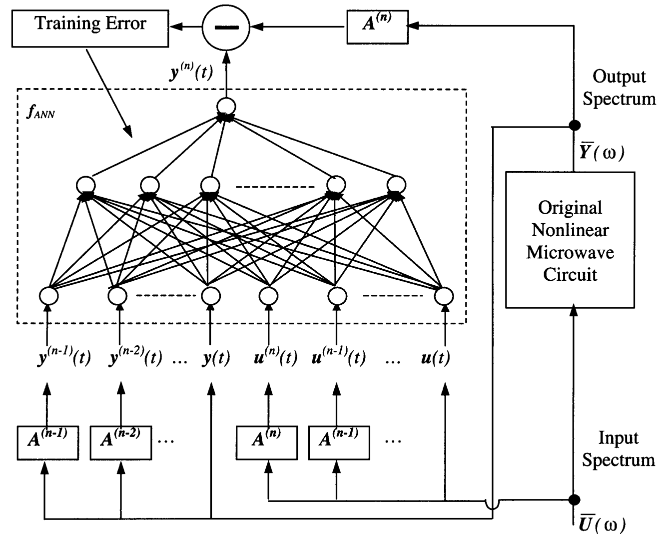


Fig. 1. *Initial training*: to train the $\boldsymbol{f}_{\text{ANN}}$ part in time-domain using spectrum data, where $\boldsymbol{A}^{(i)}$ is the time-derivative operator corresponding to (4).

$$E_2 = \frac{1}{2} \sum_{m=1}^{M} \sum_{\omega \in \Omega} \left\| \boldsymbol{Y}_m(\omega) - \overline{\boldsymbol{Y}}_m(\omega) \right\|^2 \tag{8}$$

where $\boldsymbol{Y}_m(\omega)$ and $\overline{\boldsymbol{Y}}_m(\omega)$ represent spectrum from model and mth sample of training data, respectively. In order to achieve the harmonic solutions $\boldsymbol{Y}_m(\omega)$ from the DNN model, we apply differentiation over the $\boldsymbol{f}_{\text{ANN}}$ using the adjoint neural network method [30]. The resulting derivatives fit the Jacobian matrix of harmonic balance equations.

The training technique presented here demonstrates that both time- and frequency-domain data can be used for DNN training. The compatibility of DNN training with large-signal harmonic data is an important advantage over the discrete recurrent neural network approach [22] whose training is limited to time-domain only.

D. Use of the Trained DNN Model in Circuit Simulation

1) Method 1—Circuit Representation of DNN: An exact circuit representation of our DNN model can be derived as shown in Fig. 2(a). The state variables are represented by voltages on unit capacitors with their currents controlled by other state variables, e.g., $C \cdot \dot{\boldsymbol{v}}_1(t) = \boldsymbol{v}_2(t)$, where $C = 1$. The dynamic model inputs are defined as voltages on unit inductors with their currents controlled by input dynamics of different orders, e.g., $\boldsymbol{u}^{(1)}(\boldsymbol{t}) = L \cdot \dot{\boldsymbol{u}}(t)$, where $L = 1$. In this way, the trained model can be conveniently incorporated into available simulation tools for high-level circuit and system design. This can be achieved in most existing simulators without doing computer programming.

$$E_1 = \frac{1}{2} \sum_{t \in T} \sum_{m=1}^{M} \left\| \boldsymbol{f}_{\text{ANN}} \left(\overline{\boldsymbol{y}}_m^{(n-1)}(t), \ldots, \overline{\boldsymbol{y}}_m(t), \overline{\boldsymbol{u}}_m^{(n)}(t), \ldots, \overline{\boldsymbol{u}}_m(t) \right) - \overline{\boldsymbol{y}}_m^{(n)}(t) \right\|^2 \tag{7}$$

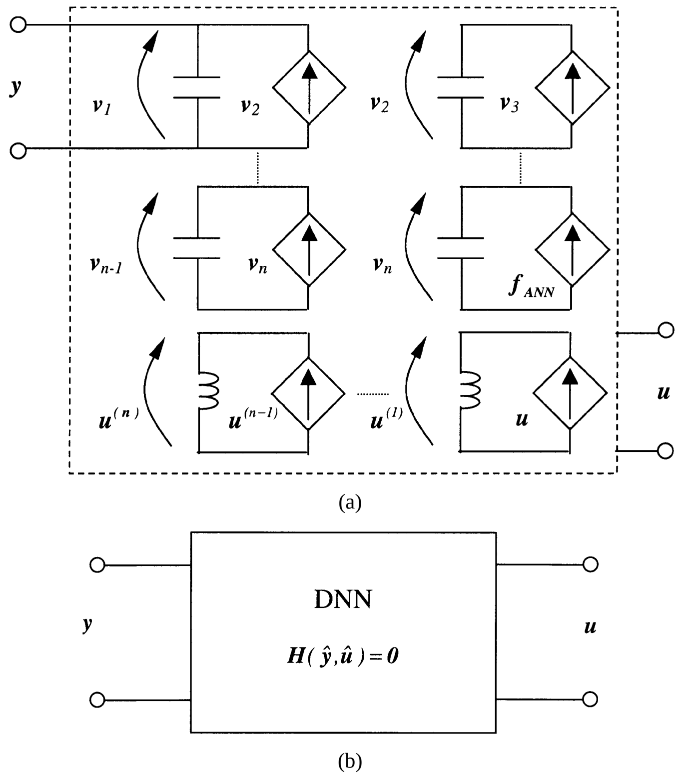


Fig. 2. Representations of DNN for incorporation into high-level simulation. (a) Circuit representation of the DNN model. (b) HB representation of the DNN model. The two representations are different only in implementation and they are numerically equivalent to each other.

2) Efficient HB Representation of DNN: Here we propose another method for incorporating the DNN model into circuit simulation. We use HB as the circuit simulation environment. Through the formulation described below, we are able to eliminate most of the state variables in DNN by Fourier Transform and use even fewer variables during HB simulation, further speeding up circuit simulation. The HB representation is shown in Fig. 2(b).

Let $\boldsymbol{U}(\omega)$ and $\boldsymbol{Y}(\omega)$ be the Fourier Transform of input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$, respectively. Let $\boldsymbol{B}(\omega, t)$ represent the Fourier Transform matrix, such that

$$\boldsymbol{Y}(\omega) = \sum_{t\in T} \boldsymbol{B}(\omega, t) \cdot \boldsymbol{y}(t) \tag{9}$$

$$\boldsymbol{U}(\omega) = \sum_{t\in T} \boldsymbol{B}(\omega, t) \cdot \boldsymbol{u}(t). \tag{10}$$

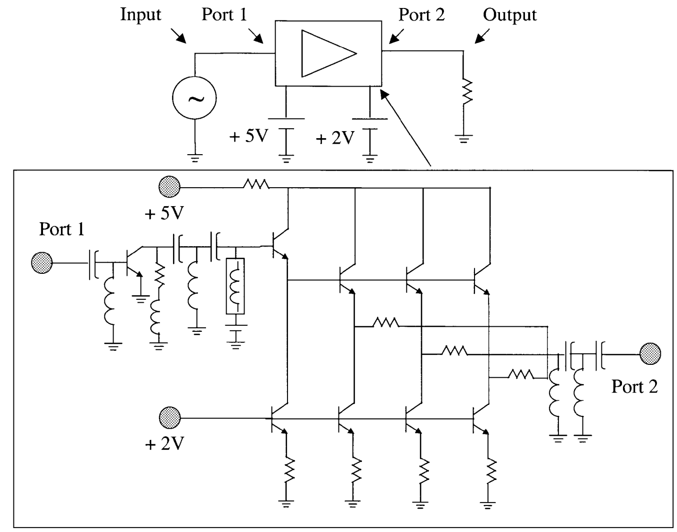


Fig. 3. Amplifier circuit to be represented by a DNN model.

Since

$$\boldsymbol{y}^{(i)}(t) = \sum_{\omega\in\Omega} \boldsymbol{A}^{(i)}(\omega, t) \cdot \boldsymbol{Y}(\omega) \tag{11}$$

$$\boldsymbol{u}^{(i)}(t) = \sum_{\omega\in\Omega} \boldsymbol{A}^{(i)}(\omega, t) \cdot \boldsymbol{U}(\omega) \tag{12}$$

premultiplying $\boldsymbol{B}(\omega, t)$ to the $\boldsymbol{f}_{\text{ANN}}$ equation in DNN model of (3), we have the HB equation for the DNN as (13), shown at the bottom of this page, where $\boldsymbol{Y}(\omega)$ is the Fourier Transform of the time-domain signal $\boldsymbol{y}(t)$ as defined earlier.

Substituting (11) and (12) into the $\boldsymbol{f}_{\text{ANN}}$ equation of the DNN in (3), we have an input–output waveform equation, as shown by (14) at the bottom of the following page.

Let $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{u}}$ be vectors containing $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ for all the time samples t, $t \in T$. Let $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{U}}$ be vectors containing $\boldsymbol{Y}(\omega)$ and $\boldsymbol{U}(\omega)$ at all the spectrum components ω, $\omega \in \Omega$. Since $\boldsymbol{A}(\omega, t)$, $\boldsymbol{B}(\omega, t)$ and $\boldsymbol{A}^{(i)}(\omega, t)$ contain Fourier base functions and their time derivatives, they are independent of any signals in the circuit and are constants during HB simulation. Therefore, the HB equations for DNN in (13) can be expressed as

$$\boldsymbol{F}(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{U}}) = \boldsymbol{0} \tag{15}$$

where $\boldsymbol{F}()$ means " nonlinear functions of ". Equation (14) can be expressed as

$$\boldsymbol{H}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}}) = \boldsymbol{0} \tag{16}$$

where $\boldsymbol{H}()$ also means " nonlinear functions of ".

$$\begin{aligned}
&\sum_{t\in T} \boldsymbol{B}(\omega, t) \cdot \sum_{\omega\in\Omega} \boldsymbol{A}^{(n)}(\omega, t) \cdot \boldsymbol{Y}(\omega) \\
&\quad - \sum_{t\in T} \boldsymbol{B}(\omega, t) \cdot \boldsymbol{f}_{\text{ANN}} \Bigg(\sum_{\omega\in\Omega} \boldsymbol{A}^{(n-1)}(\omega, t) \cdot \boldsymbol{Y}(\omega), \\
&\qquad \sum_{\omega\in\Omega} \boldsymbol{A}^{(n-2)}(\omega, t) \cdot \boldsymbol{Y}(\omega), \ldots, \sum_{\omega\in\Omega} \boldsymbol{A}(\omega, t) \cdot \boldsymbol{Y}(\omega), \sum_{\omega\in\Omega} \boldsymbol{A}^{(n)}(\omega, t) \cdot \boldsymbol{U}(\omega), \\
&\qquad \sum_{\omega\in\Omega} \boldsymbol{A}^{(n-1)}(\omega, t) \cdot \boldsymbol{U}(\omega), \ldots, \sum_{\omega\in\Omega} \boldsymbol{A}(\omega, t) \cdot \boldsymbol{U}(\omega) \Bigg) = \boldsymbol{0}
\end{aligned} \tag{13}$$

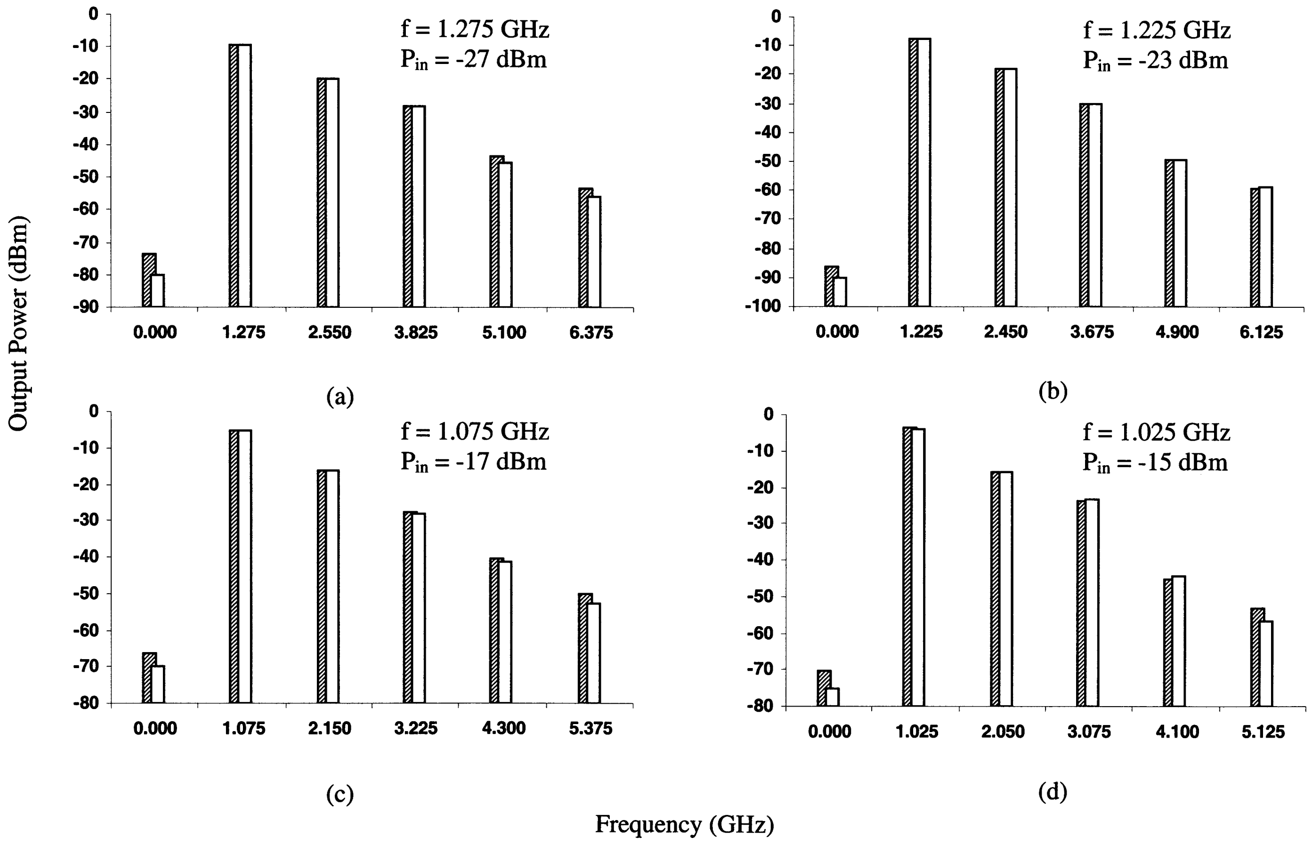


Fig. 4. Amplifier output: spectrum comparison between DNN (■), and ADS solution of original circuit (□) at load $= 50\ \Omega$. Excellent agreement is achieved even though such data were never used in training.

We call (15) or (16) as the HB representation of DNN. To implement DNN into HB circuit simulation, we program either (15) or (16) within the HB environment. In (15), given input harmonic values $\hat{\boldsymbol{U}}$, the DNN will produce output harmonic $\hat{\boldsymbol{Y}}$. In (16), given input waveforms $\hat{\boldsymbol{u}}$, the DNN will produce output waveforms $\hat{\boldsymbol{y}}$. Notice that (16) uses only $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{u}}$ (without explicit derivative variables) at all time points. The HB simulator will solve the overall HB equation including DNN during HB simulation.

In this way, the variables for HB simulation due to DNN are only $\hat{\boldsymbol{Y}}$, $\hat{\boldsymbol{U}}$. All higher order information of inputs and outputs will be implied by $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{U}}$ through Fourier transformations. Since the total number of nonlinear nodes from the DNN is n times less than that in the circuit representation of DNN, this HB simulation will have further computation speed up.

Notice that (15) or (16) is only used as a plug when DNN is plugged to the circuit simulator. The DNN model itself is the dynamic equation of (3). Since DNN is a continuous time-domain model, the model is independent of the choice of number of harmonics and number of time samples. Furthermore, DNN is independent of the number of tones in harmonic balance simulation. This flexibility of the DNN is clear progress over the existing behavioral neural models whose structure is dependent on the number of tones.

Although different in their implementations in circuit simulators, the two representations of the DNN, i.e., circuit and HB representations, are numerically equivalent. The former representation is more convenient to implement and the latter is computationally more efficient.

$$
\begin{aligned}
&\sum_{\omega\in\Omega} \boldsymbol{A}^{(n)}(\omega,t)\cdot\sum_{\tau\in T}\boldsymbol{B}(\omega,\tau)\cdot\boldsymbol{y}(\tau)\\
&\quad-\boldsymbol{f}_{\mathrm{ANN}}\Bigg(\sum_{\omega\in\Omega}\boldsymbol{A}^{(n-1)}(\omega,t)\cdot\sum_{\tau\in T}\boldsymbol{B}(\omega,\tau)\cdot\boldsymbol{y}(\tau),\\
&\qquad\sum_{\omega\in\Omega}\boldsymbol{A}^{(n-2)}(\omega,t)\cdot\sum_{\tau\in T}\boldsymbol{B}(\omega,\tau)\cdot\boldsymbol{y}(\tau),\ldots,\sum_{\omega\in\Omega}\boldsymbol{A}(\omega,t)\cdot\sum_{\tau\in T}\boldsymbol{B}(\omega,\tau)\cdot\boldsymbol{y}(\tau),\\
&\qquad\sum_{\omega\in\Omega}\boldsymbol{A}^{(n)}(\omega,t)\cdot\sum_{\tau\in T}\boldsymbol{B}(\omega,\tau)\cdot\boldsymbol{u}(\tau),\ldots,\sum_{\omega\in\Omega}\boldsymbol{A}(\omega,t)\cdot\sum_{\tau\in T}\boldsymbol{B}(\omega,\tau)\cdot\boldsymbol{u}(\tau)\Bigg)=\boldsymbol{0}
\end{aligned}
\tag{14}
$$

E. Discussions

The proposed DNN automatically achieves model reduction effect since the DNN order n can be chosen to be much less than the order of the original nonlinear circuit. By adjusting n, we can conveniently adjust the order of our model. Another factor in the DNN model is that the number of hidden neurons in $\boldsymbol{f}_{\rm ANN}$ represents the extent of nonlinearity between dynamic inputs and dynamic outputs. By adjusting the number of hidden neurons, we can conveniently adjust the degree of nonlinearity needed in the DNN model. Such convenient adjustments of order and nonlinearity in DNN make the model creation much easier than conventional equivalent circuit based approaches where manual trial and error may be needed to create/adjust the equivalent circuit topology and the nonlinear equation terms in it.

III. Dynamic-Modeling Examples

A. DNN Modeling of Amplifier

This example shows the modeling of nonlinear effects of an amplifier using the DNN technique. The amplifier internally has 9 n-p-n transistors modeled by Agilent-ADS nonlinear models Q34, Q37, and HP AT 41411 [31] shown in Fig. 3.

We train our DNN to learn the input–output dynamics of the amplifier. We choose a hybrid two-port formulation with $\boldsymbol{u} = [v_{\rm IN}, i_{\rm OUT}]^T$ as input, and $\boldsymbol{y} = [i_{\rm IN}, v_{\rm OUT}]^T$ as output. The DNN model includes

$$i_{\rm IN}^{(n)}(t) = f_{\rm ANN1}\left(i_{\rm IN}^{(n-1)}(t), i_{\rm IN}^{(n-2)}(t), \ldots, i_{\rm IN}(t), v_{\rm IN}^{(n)}(t), v_{\rm IN}^{(n-1)}(t), \ldots, v_{\rm IN}(t)\right) \quad (17)$$

$$v_{\rm OUT}^{(n)}(t) = f_{\rm ANN2}\left(v_{\rm OUT}^{(n-1)}(t), v_{\rm OUT}^{(n-2)}(t), \ldots, v_{\rm OUT}(t), v_{\rm IN}^{(n)}(t), v_{\rm IN}^{(n-1)}(t), \ldots, v_{\rm IN}(t), i_{\rm OUT}^{(n)}(t), i_{\rm OUT}^{(n-1)}(t), \ldots, i_{\rm OUT}(t)\right). \quad (18)$$

This input–output definition allows the model to be able to interact with external connections with other nonlinear circuits in a system level simulation.

The training data for the amplifier are gathered by exciting the circuit with a set of frequencies ($0.95 \sim 1.35$ GHz, step-size 0.05 GHz), powers ($-30 \sim -14$ dBm, step-size 2 dBm), and load impedances ($35 \sim 65$ Ω, step-size 10 Ω). In initial training, Fourier Transform sampling frequencies ranged from 47.5 to 67.5 GHz. Final training is done with optimization over harmonic balance such that modeled harmonics match original harmonics. We trained the model in multiple ways using different number of hidden neurons and orders (n) of the model as shown in Table I. Testing is performed by comparing our DNN model with the original amplifier in ADS, with different set of signals never used in training, i.e., different test frequencies ($0.975 \sim 1.325$ GHz, step-size 0.05 GHz), powers ($-29 \sim -15$ dBm, step-size 2 dBm) and loads (40, 50, and 60 Ω). The model is compared with original circuit in both time and frequency domains, and excellent agreement is achieved. Fig. 4 shows examples of spectrum comparisons. An additional comparison between our DNN model and the original amplifier is made using the 1-dB compression point. For example, at the excitation frequency 1.175 GHz, the 1-dB compression point is -35.6 dBm for the DNN model agreeing well with its original value of -35.0 dBm from the original amplifier. We also applied envelope transient analysis to the DNN amplifier model using the ADS envelope simulator. The model was driven with a 1.15-GHz carrier and modulated by a $\pi/4$ differential quadrature phase-shift keying (DQPSK) signal at 48.6 kb/s. The result of the simulation is illustrated in Fig. 5, showing two cases of power spectral regrowth at the DNN output, Fig. 5(a) when the amplifier model operates at 1-dB compression point, and Fig. 5(b) when the amplifier model operates at 10-dB compression point.

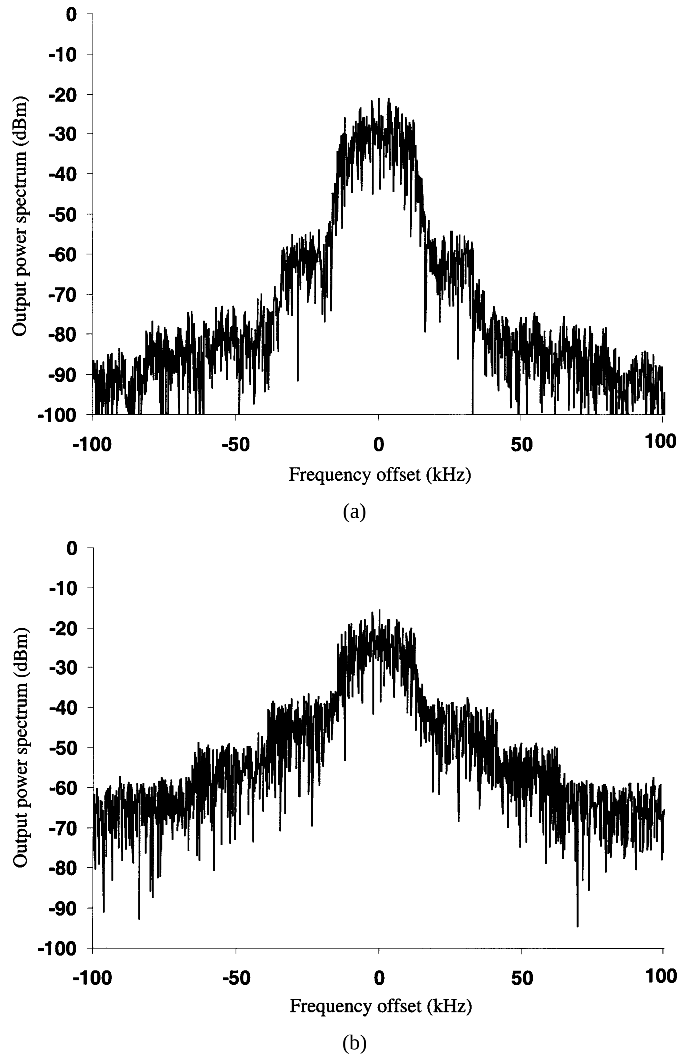


Fig. 5. Envelope transient analysis results for DNN amplifier model with $\pi/4$-DQPSK modulation. (a) DNN output power spectrum when the amplifier model operates at 1-dB compression point. (b) DNN output power spectrum when the amplifier model operates at 10-dB compression point.

To further demonstrate that the DNN model represents circuit internal behavior independent of external signals, we show a different use of the proposed technique for this amplifier. We use exactly the same formulation of the amplifier DNN model to handle two-tone harmonic balance effects. To further add to the challenge of this modeling task, we perform the training of the DNN using one-tone data and one-tone formulation of

TABLE I
AMPLIFIER: DNN ACCURACY FROM DIFFERENT TRAINING

No. of Hidden Neurons in Training (n=3)	Testing Error for Time Domain Data	Testing Error for Spectrum Data
40	4.2E-3	2.7E-3
50	2.9E-3	1.8E-3
60	3.6E-3	2.3E-3

Order n in Training	Testing Error for Time Domain Data	Testing Error for Spectrum Data
2	5.3E-3	4.3E-3
3	2.9E-3	1.8E-3
4	1.5E-2	9.9E-3

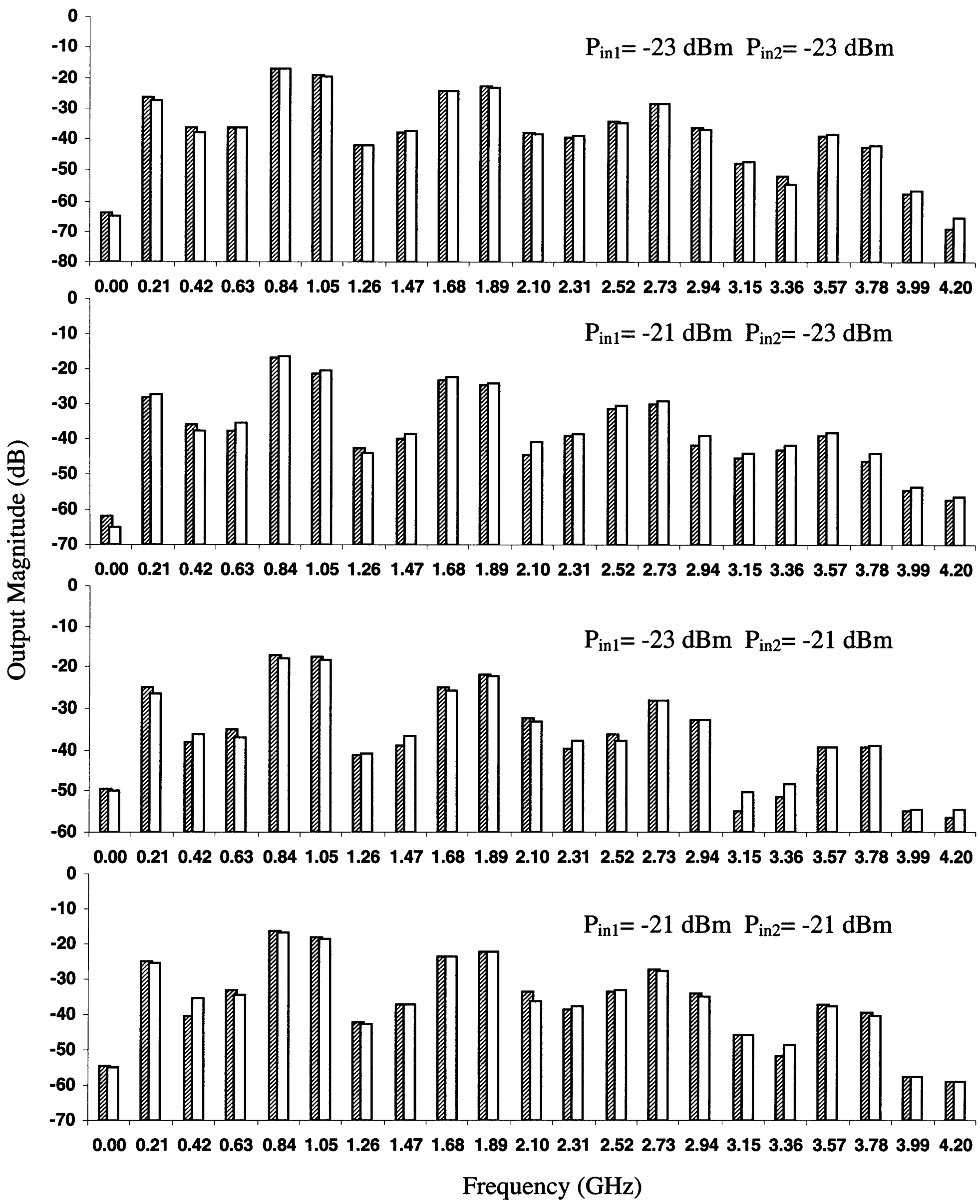


Fig. 6. Amplifier two-tone simulation result from DNN, which is trained under one-tone formulation: spectrum comparison between DNN (■) and ADS solution of original circuit (□). Excellent agreement is achieved even though such two-tone data were never used in training.

training (optimization). After training is finished, we will use the model for two-tone simulation. This ability of the DNN is progress over existing behavioral-based neural models where the model structure has to be different for different number of tones. The proposed DNN achieves uniform format regardless of the number of tones. For this demonstration, the training data

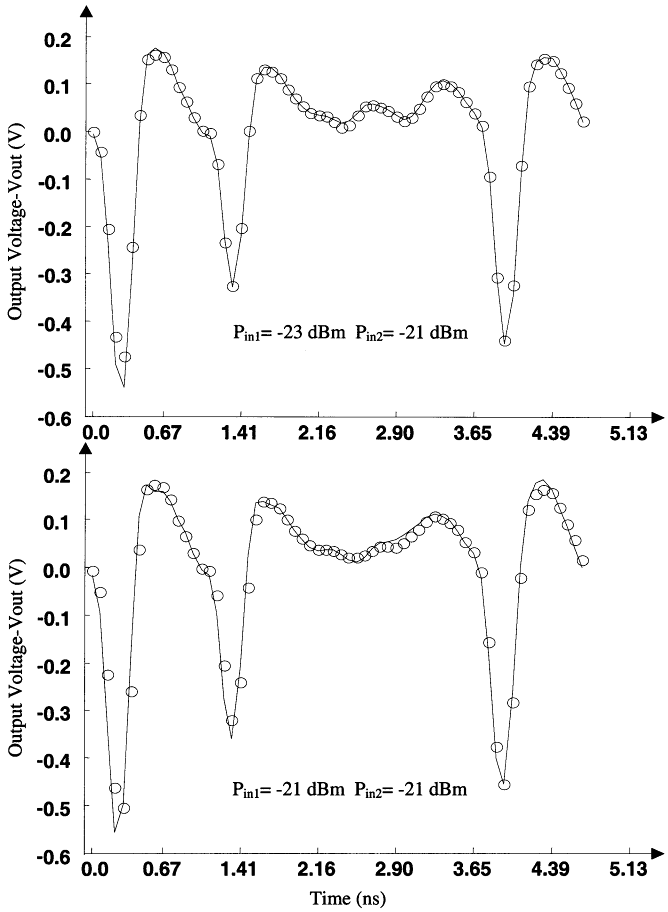


Fig. 7. Amplifier two-tone simulation result from DNN: time-domain comparison between DNN (—) and ADS solution of original circuit (o). Good agreement is achieved even though such data were never used in training.

for the amplifier are gathered by exciting the circuit with several patterns of input signal $v_{\mathrm{IN}}(t)$: fundamental frequencies (0.2 GHz, 0.22 GHz), powers at the fourth and the fifth harmonics ($-24 \sim -20$ dBm, step-size 2 dBm), and the total number of harmonics considered with harmonic balance simulation is 20. Testing is performed by comparing our model with original amplifier, with two-tone signal never used in training. For the first tone, fundamental frequency is 0.84 GHz, powers (-23 dBm, -21 dBm). For the second tone, fundamental frequency is 1.05 GHz, powers (-23 dBm, -21 dBm). The number of harmonics in the HB simulation for each tone is four, leading to a total number of 20 harmonics and intermodulated frequencies in the output signal. The two-tone solution from the DNN model is compared with the ADS solution of the original amplifier in both frequency and time domains, and excellent agreement is achieved as shown in Figs. 6 and 7, respectively. We also computed the third-order intercept point (IP3). For example, when the two-tone input powers are set to -23 dBm, the IP3 computed from our DNN model is 2.24 dBm, which is a good estimation of the original IP3 of 2.38 dBm from the original amplifier.

This example demonstrates that the same DNN structure can be used for single-tone or multitone harmonic balance simulations providing simplicity and flexibility in implementation, model development, and model usage over the existing neural network methods.

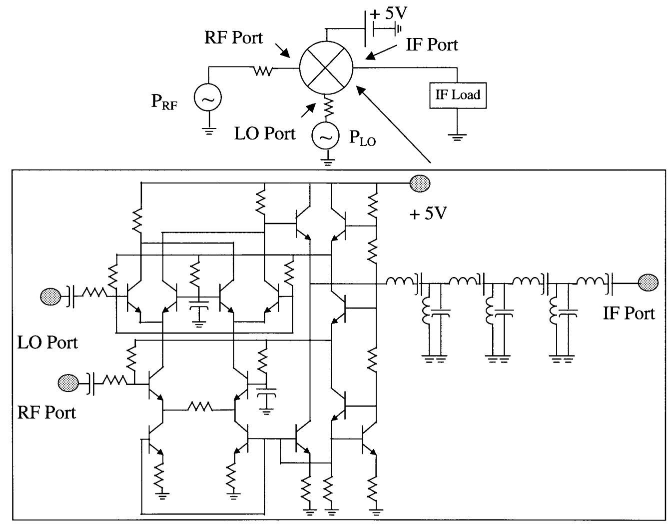


Fig. 8. Mixer equivalent circuit to be represented by a DNN model.

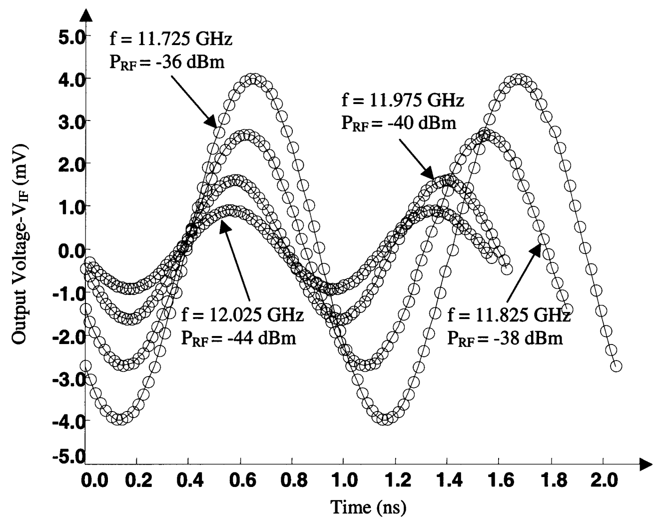


Fig. 9. Mixer V_{IF} output: time-domain comparison between DNN (—) and ADS solution of original circuit (o). Good agreement is achieved even though such data were never used in training.

B. Mixer DNN Modeling

This example illustrates DNN modeling of a mixer. The circuit internally is a Gilbert cell with 14 n-p-n transistors in ADS [31] shown in Fig. 8. The dynamic input and output of the model is defined in hybrid form as $\boldsymbol{u} = [v_{RF}, v_{LO}, i_{IF}]^T$ and $\boldsymbol{y} = [i_{RF}, v_{IF}]^T$. The DNN model includes

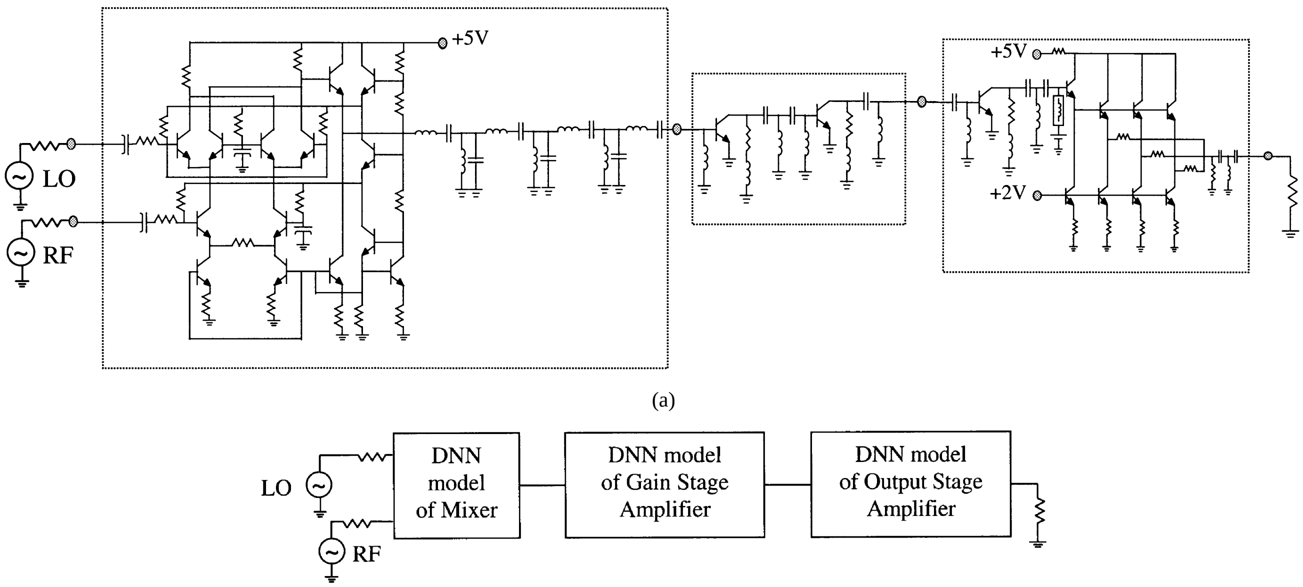
$$\begin{aligned} i_{RF}^{(n)}(t) =& f_{\mathrm{ANN1}}\Big(i_{RF}^{(n-1)}(t), i_{RF}^{(n-2)}(t), \ldots, i_{RF}(t), \\ & v_{RF}^{(n)}(t), v_{RF}^{(n-1)}(t), \ldots, v_{RF}(t)\Big) \quad (19) \end{aligned}$$

$$\begin{aligned} v_{IF}^{(n)}(t) =& f_{\mathrm{ANN2}}\Big(v_{IF}^{(n-1)}(t), v_{IF}^{(n-2)}(t), \ldots, v_{IF}(t), \\ & v_{RF}^{(n)}(t), v_{RF}^{(n-1)}(t), \ldots, v_{RF}(t), \\ & v_{LO}^{(n)}(t), v_{LO}^{(n-1)}(t), \ldots, v_{LO}(t), \\ & i_{IF}^{(n)}(t), i_{IF}^{(n-1)}(t), \ldots, i_{IF}(t)\Big). \quad (20) \end{aligned}$$

The training data are gathered in such way that RF input frequency and power level changed from 11.7 to 12.1 GHz with

TABLE II
MIXER: DNN ACCURACY FROM DIFFERENT TRAINING

No. of Hidden Neurons in Training (n=4)	Testing Error for Time Domain Data	Testing Error for Spectrum Data	Order n in Training	Testing Error for Time Domain Data	Testing Error for Spectrum Data
45	8.7E-4	6.7E-4	2	2.7E-3	1.9E-3
55	4.6E-4	2.0E-4	3	1.4E-3	8.6E-4
65	6.5E-4	4.6E-4	4	4.6E-4	2.0E-4



(b)

Fig. 10. DBS receiver subsystem. (a) Connected by original detailed equivalent circuit in ADS. (b) Connected by our DNNs.

step size 0.05 GHz and from −45 dBm to −35 dBm with step size 2 dBm, respectively. Local oscillator (LO) signal is fixed at 10.75 GHz and 10 dBm. Load is perturbed by 10% at every harmonic in order to let the model learn load effects. The DNN is trained with different numbers of hidden neurons and orders (n) as shown in Table II. Testing is done in ADS using input frequencies (11.725 $\sim$ 12.075 GHz, step size 0.05 GHz) and power levels (−44, −42, −40, −38, −36 dBm). The agreement between model and ADS is achieved in time and frequency domains even though those test information was never seen in training. Fig. 9 illustrates examples of test in the time domain.

C. Nonlinear Simulation of DBS Receiver System

To further confirm the validity of the proposed DNN, we also trained a DNN model representing another amplifier (gain stage amplifier) using a way similat to that in Section III-A, and combined the three trained DNNs of mixer and amplifiers into a DBS receiver subsystem [32], where the amplifier trained in Section III-A is used as the output stage. The overall DBS system is shown in Fig. 10.

We have incorporated the DNN models of the amplifiers and mixer into harmonic balance simulation in two ways. The first way is to use the circuit representation of DNNs as described in Fig. 2(a) incorporated into ADS software. This is achieved by constructing the equivalent circuit in ADS using capacitors, controlled sources, and algebraic expressions representing f_{ANN} neural network function. The second way is to program the HB representation of DNN model of Fig. 2(b) for amplifiers and mixer according to (16). The overall DBS system output solved by the efficient HB representation of DNN's match completely with that solved using circuit representation of DNNs in ADS, confirming the consistency between the two representations of DNN as shown in Fig. 11(a). Next, we compare ADS harmonic balance simulation with the original DBS system in Fig. 10(a) with that using DNN models of amplifiers and mixer in Fig. 10(b). The overall DBS system solution using DNNs matches that of the original system as shown in Fig. 11(b), even though these obviously distorted signals were never used in training of any of the DNNs.

We also performed Monte Carlo analysis of the original and the DNN-based DBS systems under random sets of RF input frequencies and power levels. The statistics from the DNN-based system simulation, shown in Fig. 12, match those from the original system. The CPU for 1000 analyses of the DBS system

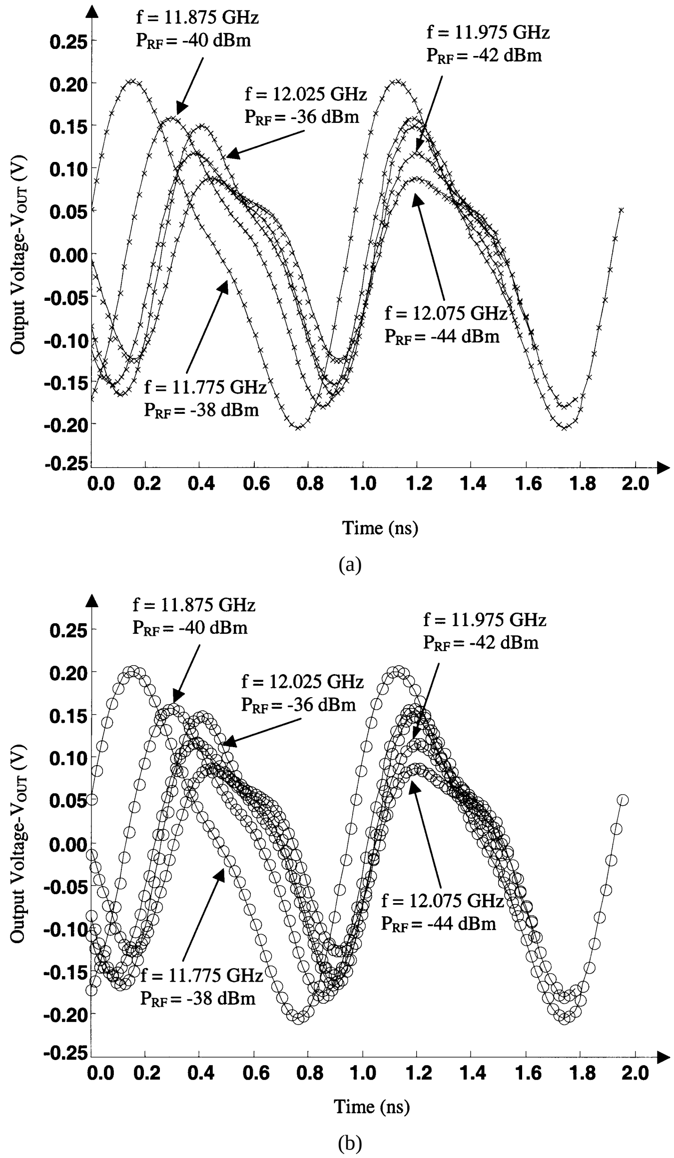


Fig. 11. (a) DBS system output. Comparison between system solutions using HB representation of DNN models (—), and circuit representation of DNN models (x). The solutions from the two representations of DNN are in complete agreement with each other. (b) DBS system output: comparison between system solutions using DNN models (—), and ADS simulation of original system (o). Excellent agreement is achieved even though these nonlinear solutions were never used in training.

using original circuits, using circuit representation of DNNs, and using HB representation of DNNs are 6.52, 3.94, and 0.81 h, respectively, showing efficiency of the DNN-based system simulation.

A further comparison is made between the proposed dynamic neural network, the conventional static neural network approach, and the conventional behavioral modeling approach (nonneural network approach). We trained three static neural networks using the static I–Q (current-charge) model of [9] to learn the two amplifiers and the mixer, and incorporated these models into ADS using NeuroADS [33]. The overall DBS system simulation using the static neural models was performed in ADS. As expected, such static models, while suitable for intrinsic FET modeling, are not accurate enough for amplifier and mixers even though the model incorporates

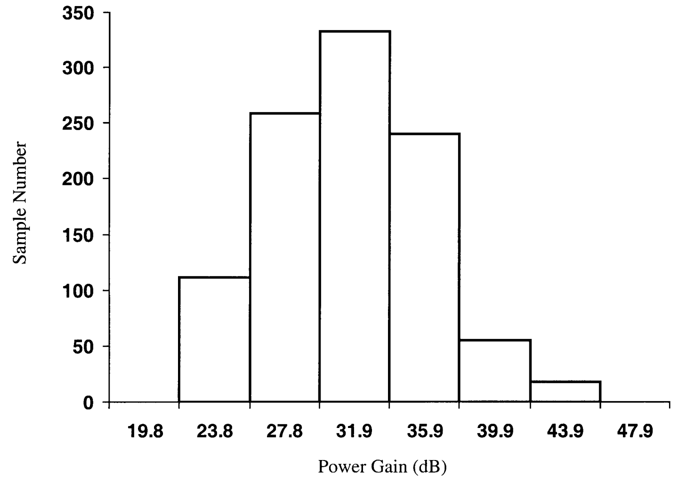


Fig. 12. Histogram of power gain of DBS system for 1000 Monte Carlo simulations with random input frequency and amplitude.

charge information. The overall error in the output signal of the DBS system is 6.1% relative to original detailed system simulation.

For the case of conventional behavioral modeling approach, we constructed three behavioral models to represent the two amplifiers and the mixer. The behavioral models were obtained in two ways, one way is to use the data-based behavioral model [31], and another way is to use optimization to optimize the behavioral model parameters in [31] to best match the behavior of the original amplifiers and the mixer. An overall DBS system simulation with the best behavioral models was used. As expected, the behavioral models run extremely fast, and provide only an approximate solution. Table III provides a summary of model test error for the two amplifiers and one mixer through different methods. Table IV provides comparisons of computation speed and accuracy with the different methods for the DBS system simulation. It is observed that the proposed DNN (i.e., dynamic neural network) approach provides the best overall performance being much faster than original system simulation and much more accurate than both the conventional behavioral modeling approach, and the static neural network approach.

IV. Conclusions

This paper has presented a neural network method for modeling nonlinear microwave circuits and its applications for high-level simulation. The model is derived in the most effective format, i.e., continuous time-domain dynamic format and can be developed from input–output data without having to rely on internal details of the circuits. A novel training scheme allows the training of DNN to learn from either time or frequency domain input–output information. After being trained, the proposed model can be conveniently incorporated into existing simulators. The proposed DNN retains or enhances the advantages of learning, speed, and accuracy as in existing neural network techniques; and provides additional advantages of being theoretically elegant and practically suitable for diverse needs of nonlinear microwave simulation, e.g., standardized implementation in simulators, suitability for both time- and frequency-domain applications, and multitone simulations. The technique allows

TABLE III
DBS SYSTEM COMPONENT MODELS: COMPARISON BETWEEN CONVENTIONAL BEHAVIORAL MODEL, STATIC NEURAL MODEL, AND DNNs

Techniques / Components	Conventional Behavioral Model	Static I-Q Neural Model	Proposed DNN Model
Mixer	3.4%	3.2%	0.02%
Gain stage Amplifier	1.2%	1.9%	0.09%
Output stage Amplifier	7.7%	2.9%	0.16%

TABLE IV
DBS-RECEIVER SUBSYSTEM: COMPARISON BETWEEN SYSTEM SIMULATION USING CONVENTIONAL BEHAVIORAL MODEL, STATIC NEURAL MODEL, DNNs, AND DETAILED ORIGINAL CIRCUIT

Techniques / Comparisons	DBS system simulation using Conventional Behavioral Model	DBS system simulation using Static I-Q Neural Model	DBS system simulation using HB Representation of DNNs	DBS system simulation using Circuit Representation of DNNs	DBS system simulation using Detailed Original Circuit
Test Error for Spectrum Data	10.3%	6.1%	0.21%	0.21 %	0.0% (reference for comparison)
CPU time for 1000 Monte-Carlo Analysis	0.18 hours	0.26 hours	0.81 hours	3.94 hours	6.52 hours

further realizing the flexibility of neural-based approaches in nonlinear microwave modeling, simulation, and optimization.

REFERENCES

[1] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA: Artech House, 2000.

[2] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in multilayer circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2495–2503, Dec. 1996.

[3] P. Burrascano, S. Fiori, and M. Mongiardo, "A review of artificial neural networks applications in microwave computer-aided design," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 158–174, 1999.

[4] A. Veluswami, M. S. Nakhla, and Q. J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 712–723, May 1997.

[5] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2515–2523, Dec. 1997.

[6] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins, and M. C. Calcatera, "Artificial neural networks for fast and accurate EM-CAD of microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 794–802, May 1997.

[7] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sánchez, and Q. J. Zhang, "Neuromodeling of microwave circuits exploiting space mapping technology," *IEEE Trans. Microwave Theory Tech.*, vol. 47, pp. 2417–2427, Dec. 1999.

[8] S. Goasguen, S. M. Hammadi, and S. M. El-Ghazaly, "A global modeling approach using artificial neural network," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 153–156, 1999.

[9] H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349–1358, June 1995.

[10] P. M. Watson, C. Cho, and K. C. Gupta, "Electromagnetic-artificial neural network model for synthesis of physical dimensions for multilayer asymmetric coupled transmission structures," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 175–186, 1999.

[11] M. Vai and S. Prasad, "Neural networks in microwave circuit design – Beyond black box models," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 9, pp. 187–197, 1999.

[12] F. Wang and Q. J. Zhang, "Knowledge based neural models for microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2333–2343, Dec. 1997.

[13] P. M. Watson, K. C. Gupta, and R. L. Mahajan, "Application of knowledge-based artificial neural network modeling to microwave components," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 254–260, 1999.

[14] P. Viszmuller, *RF Design Guide, Systems, Circuits, and Equations*. Norwood, MA: Artech House, 1995.

[15] T. R. Turlington, *Behavioral Modeling of Nonlinear RF and Microwave Devices*. Boston, MA: Artech House, 2000.

[16] G. Casinovi and A. Sangiovanni-Vincentelli, "A macromodeling algorithm for analog circuits," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 150–160, Feb. 1991.

[17] P. K. Gunupudi and M. S. Nakhla, "Model-reduction of nonlinear circuits using Krylov-space techniques," in *Proc. IEEE Int. Design Automation Conf.*, June 1999, pp. 13–16.

[18] K. Shirakawa, M. Shimiz, N. Okubo, and Y. Daido, "A large signal characterization of an HEMT using a multilayered neural network," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1630–1633, Sept. 1997.

[19] Y. Harkouss, J. Rousset, H. Chehade, E. Ngoya, D. Barataud, and J. P. Teyssier, "The use of artificial neural networks in nonlinear microwave devices and circuits modeling: An application to telecommunication system design," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 198–215, 1999.

[20] J. Verspecht, F. Verbeyst, M. V. Bossche, and P. Van Esch, "System level simulation benefits from frequency domain behavioral models of mixers and amplifiers," in *Proc. European Microwave Conf.*, vol. 2, Munich, Germany, 1999, pp. 29–32.

[21] V. Rizzoli, A. Neri, D. Masotti, and A. Lipparin, "A new family of neural network-based bi-directional and dispersive behavioral models for nonlinear RF/microwave subsystems," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 12, pp. 51–70, 2002.

[22] Y. H. Fang, M. C. E. Yagoub, F. Wang, and Q. J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2335–2344, Dec. 2000.

[23] A. Neri, C. Cecchetti, and A. Lipparini, “Fast prediction of the performance of wireless links by sumulation-trained neural networks,” in *IEEE MTT-S Int. Microwave Symp. Dig.*, 2000, pp. 429–432.

[24] V. Devabhaktuni, M. C. E. Yagoub, Y. Fang, J. J. Xu, and Q. J. Zhang, “Neural networks for microwave modeling: Model development issues and nonlinear techniques,” *Int. J. RF Microwave Computer-Aided Eng.*, vol. 11, pp. 4–21, 2001.

[25] J. J. Hopfield and T. W. Tank, “’Neural’ computation of decisions in optimization problems,” *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[26] S. Haykin, *Neural Networks*. New York: IEEE Press, 1994.

[27] T. Hryces, *Neurocontrol, Toward An Industrial Control Methodology*. New York: Wiley- Interscience, 1997.

[28] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrand Reinhold, 1983.

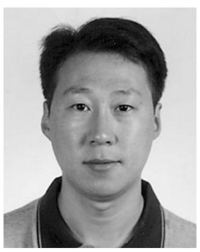
[29] K. S. Kundert, G. B. Sorkin, and A. Sangiovanni-vincentelli, “Applying harmonic balance to amost-periodic circuits,” *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 366–378, Feb. 1988.

[30] J. J. Xu, M. C. E. Yagoub, and Q. J. Zhang, “Exact adjoint sensitivity for neural based microwave modeling and design,” in *IEEE MTT-S Int. Microwave Symp. Dig.*, 2001, pp. 1015–1018.

[31] *ADS-Advanced Design System Version 2002*, Agilent Technologies, Santa Rosa, CA, 2002.

[32] M. C. E. Yagoub and H. Baudrand, “Optimum design of nonlinear microwave circuits,” *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 779–786, May 1994.

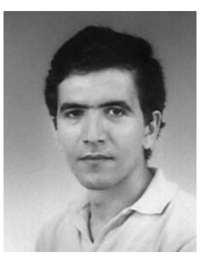
[33] Q. J. Zhang, “NeuroADS,” Dept. Electron., Carleton Univ., Ottawa, ON, Canada, 1999.



Jianjun Xu (S’00) was born in Liaoning, China, in 1975. He received the B.Eng. degree in electrical and electronics engineering from Tianjin University, Tianjin, China, in 1998. He is currently working toward the Ph.D degree in the Department of Electronics, Carleton University, Ottawa, ON, Canada.

His research interests include neural networks modeling and their applications in computer-aided design for electronics circuits.

Mr. Xu was the recipient of the Student Paper Award at IMS-2001, the 2002–2003 Ontario Graduate Scholarship in Science and Technology, and the 2002 Ontario Graduate Student Scholarship.



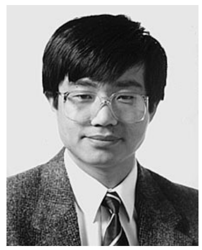
Mustapha C. E. Yagoub (M’96) received the Diplôme d’Ingénieur degree in electronics and the Magister degree in telecommunications from the Ecole Nationale Polytechnique, Algiers, Algeria, in 1979 and 1987, respectively, and the Ph.D. degree from the Institut National Polytechnique, Toulouse, France, in 1994.

He was with the Institute of Electronics , Université des Sciences et de la Technologie Houari Boumédienne, Algiers, Algeria, first, as an Assistant during 1983–1991, and then as an Assistant Professor during 1994–1999. From 1999 to 2001, he was with the Department of Electronics, Carleton University, Ottawa, ON, Canada, working on neural networks applications in microwave areas. He is currently an Assistant Professor in the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada. His research interests include neural networks for microwave applications, CAD of linear and nonlinear microwave devices and circuits, and applied electromagnetics. He has authored more than 70 publications in international journals and conferences. He is a coauthor of “*Conception de Circuits Linéaires et Non Linéaires Micro-Ondes*” (Toulouse, France: Cépadues Ed., 2000).

Runtao Ding was born in Shanghai, China, in 1938. He received the diploma from Tianjin University, Tianjin, China, in 1961.

Since 1961, he has been with the Department of Electronic Engineering and the School of Electronic Information Engineering, Tianjin University, where he is currently a Professor. From 1991 to 1996 and from 1996 to 1999, he was the Chairman of the Department and the Dean of the School, respectively. His research interests include nonlinear signal processing, image processing, neural networks, and circuit design.

Prof. Ding was a Co-Chair of the TPC of IEEE APCCAS’2000.



Qi-Jun Zhang (S’84–M’87–SM’95) received the B.Eng. degree from East China Engineering Institute, Nanjing, China, in 1982, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 1987.

He was with the System Engineering Institute, Tianjin University, Tianjin, China, in 1982 and 1983. During 1988–1990, he was with Optimization Systems Associates Inc.(OSA), Dundas, ON, Canada, developing advanced microwave optimization software. In 1990, he joined the Department of Electronics, Carleton University, Ottawa, ON, Canada, where he is presently a Professor. His research interests are neural network and optimization methods for high-speed/high-frequency circuit design, and has authored more than 150 papers on these topics. He is a coauthor of *Neural Networks for RF and Microwave Design* (Boston, MA: Artech House, 2000), a Co-Editor of *Modeling and Simulation of High-Speed VLSI Interconnects* (Boston, MA: Kluwer, 1994), and a contributor to *Analog Methods for Computer-Aided Analysis and Diagnosis* (New York: Marcel Dekker, 1988). He was a Guest Co-Editor for a Special Issue on High-Speed VLSI Interconnects of the *International Journal of Analog Integrated Circuits and Signal Processing* and twice a Guest Editor for the Special Issues on Applications of ANN to RF and Microwave Design for the *International Journal of Radio Frequency and Microwave Computer-Aided Engineering*.

Dr. Zhang is a member of the Professional Engineers of Ontario, Canada.