A New Macromodeling Approach for Nonlinear Microwave Circuits Based on Recurrent Neural Networks

Yonghua Fang, *Student Member, IEEE*, Mustapha C. E. Yagoub, *Member, IEEE*, Fang Wang, *Student Member, IEEE*, and Qi-Jun Zhang, *Senior Member, IEEE*

***Abstract*—In this paper, a new macromodeling approach is developed in which a recurrent neural network (RNN) is trained to learn the dynamic responses of nonlinear microwave circuits. Input and output waveforms of the original circuit are used as training data. A training algorithm based on back propagation through time is developed. Once trained, the RNN macromodel provides fast prediction of the full analog behavior of the original circuit, which can be useful for high-level simulation and optimization. Three practical examples of macromodeling a power amplifier, mixer, and MOSFET are used to demonstrate the validity of the proposed macromodeling approach.**

***Index Terms*—Computer-aided design, macromodeling, neural networks, nonlinear circuits, optimization, simulation.**

I. Introduction

RECENTLY, a new computer-aided design (CAD) approach based on a neural-networks model has been introduced for microwave modeling, simulation, and optimization [1]–[4]. After being trained with microwave data, the neural model can be used in microwave design, providing fast answers to the task it has learned. Significant progress in applying the neural network to the modeling of passive electromagnetic (EM) structures [5], [6] and active devices [7], [8] has been made. This paper presents a further advance in this direction. For the first time, a recurrent neural network (RNN) methodology is presented here for macromodeling of general dynamic behavior of nonlinear microwave circuits.

A nonlinear macromodel aims to represent the input–output behavior of a nonlinear circuit in a form that is faster and simpler to evaluate than original nonlinear circuit simulation. It becomes very important due to the need for simulation and optimization of analog behavior at higher level design involving many subcircuits such as amplifiers and mixers [9]. Several approaches have been developed such as the behavioral model approaches [10], where frequency-domain information [11], [12] or time-domain information based on scattering function [13] is used to develop a nonlinear model, the equivalent-circuit-based approach [14], which involves the development of a simpler circuit topology and related nonlinear component expressions, and the model reduction approach [15], which uses original circuit equations as the starting point for deriving a reduced set of nonlinear differential equations. However, the task of developing more automatic and generic model structures with full analog behavior still remains very open. In this paper, we propose a novel alternative approach for macromodeling nonlinear microwave circuits based on RNNs.

An RNN is a special type of neural network having the capability of learning and then representing dynamic system behavior. It has been used in areas such as signal processing, speech recognition [16], system identification, and control [17]–[21]. The motivation for using RNNs in macromodeling nonlinear microwave circuits is threefold. Firstly, the neural-network learning capability can be used to learn the input–output behavior directly from measured or simulated input–output data of the original circuit, avoiding otherwise manual effort of developing equivalent-circuit topology. Secondly, the universal approximation property of full RNN confirms that the model has a theoretical base of representing the full analog behavior of the circuit with good accuracy. Thirdly, the evaluation of the RNN from input to output is very fast. With these motivations, we propose a new macromodeling approach using a buffered input–output RNN. The model is completely formulated in time domain since nonlinear dynamic behavior is best described in the time domain. Input and output waveforms of original circuits are used as training data. A training method based on back propagation through time (BPTT) is introduced for model development. Three practical examples of macromodeling a power amplifier, mixer, and MOSFET are used to confirm the validity of the proposed approach.

This paper is organized as follows. In Section II, we first state the dynamic representation of nonlinear circuits and the purpose of macromodeling. A new macromodeling method is proposed and the modeling structure with an RNN is presented. Model development procedure and a training method based on BPTT is described. In Section III, three examples are shown to demonstrate our approach. A conclusion is then presented in Section IV.

II. Macromodeling Nonlinear Circuits with RNN Structure and Training

A. Formulation of Circuit Dynamics

Let N_u and N_y be the total number of input and output signals in the nonlinear circuit, respectively, and K_p be the total number of circuit parameters. Let $\hat{\mathbf{y}} = [\hat{y}_1\ \hat{y}_2\ \dots\ \hat{y}_{N_y}]^T$, $\mathbf{u} =$

Manuscript received March 5, 2000; revised August 21, 2000. This work was supported in part by the Natural Science and Engineering Research Council of Canada, and in part by Micronet, a Canadian Network Center of Excellence in Microelectronic Devices, Circuits, and Systems.

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

Publisher Item Identifier S 0018-9480(00)10727-6.

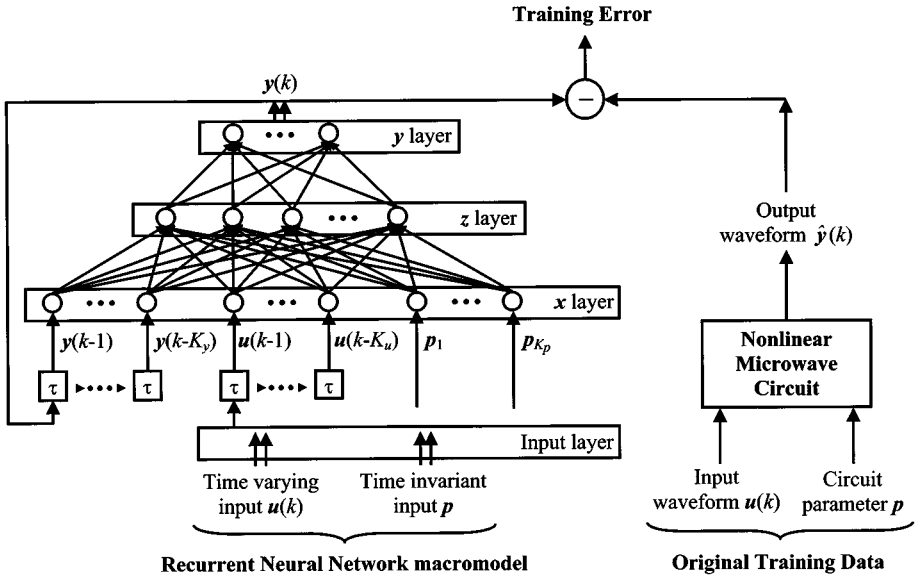


Fig. 1. Proposed RNN-based macromodel structure.

$[u_1\ u_2\ \ldots\ u_{N_u}]^T$, and $\mathbf{p} = [p_1\ p_2\ \ldots\ p_{K_p}]^T$ be vectors of the output signals, input signals, and circuit parameters of the nonlinear circuit, respectively, where T denotes transposition. The characteristics of the original nonlinear circuit can be generally described as a nonlinear system in state variable form as

$$\begin{aligned} \dot{\chi}(t) &= \varphi\big(\chi(t), \mathbf{u}(t), \mathbf{p}, t\big) \\ \hat{\mathbf{y}}(t) &= \psi\big(\chi(t), \mathbf{u}(t), t\big) \end{aligned} \tag{1}$$

where $\chi = [\chi_1\ \chi_2\ \ldots\ \chi_{N_s}]^T$ is the vector of state variables and N_s is the number of states. In a modified nodal formulation [22], the state vector $\chi(t)$ includes nodal voltages, currents of inductors, currents of voltage sources, and charge of capacitors. However, solving this original nonlinear differential equation is computationally intensive when the nonlinear circuit becomes large. For higher level design and optimization, where this circuit is used as a sub-module and repetitive evaluations for different circuit inputs are needed, a more simplified and convenient computational form should be used.

B. Statements of Macromodeling

The purpose of macromodeling is to develop a model that has a similar input–output relationship as the original complex circuit within an acceptance error range. At the same time, the evaluation of the macromodel should be much faster than that of the original circuit. Suppose the candidate macromodel is represented by $M : \mathbf{u}(t) \rightarrow \mathbf{y}(t)$, which is a functional on the space of input waveforms $\mathbf{u}(t)$. Suppose $[T_1, T_2]$ represents the time sampling range of interest for the input and output signals. Let $\mathbf{u}_{\max}$ and $\mathbf{u}_{\min}$ represent the upper and lower boundary of input signal $\mathbf{u}(t)$. For each input $\mathbf{u}(t), \mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max}, \mathbf{t} \in [T_1, T_2]$, the quality of the macromodel can be represented by the difference between the output of the macromodel and that of the original circuit in a l_2 norm

$$\int_{T_1}^{T_2} \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|^2\, dt. \tag{2}$$

C. Proposed Macromodel Structure Based on RNN

In order to derive a macromodel, we first reformulate the original problem in the discrete-time domain with a specific sampling rate into an input–output formulation as [19]

$$\begin{aligned} \hat{\mathbf{y}}(k) = \mathbf{g}\big(&\hat{\mathbf{y}}(k-1), \ldots, \mathbf{y}(k - M_{\hat{y}}), \mathbf{u}(k-1), \ldots, \\ &\mathbf{u}\big(k - M_u\big), \mathbf{p}\big) \end{aligned} \tag{3}$$

where k is the time index in the discrete time domain, M_y and M_u are the total number of delays of $\hat{\mathbf{y}}$ and $\mathbf{u}$, respectively, $M_u \leq M_y$ and $\mathbf{g}$ is a set of nonlinear functions. M_y and M_u also represent the order of the original nonlinear circuit dynamics. This model can be used as an alternative representation of the dynamics in the original circuit of (1).

In practice, it is often difficult to derive (3) analytically for a large-scale nonlinear circuit. Neural networks are well known to identify nonlinear relationships between input and output parameters, and have achieved success in solving practical modeling problems [16]. In this paper, we employ an RNN to learn the dynamic characteristics of nonlinear circuits and determine their macromodels. RNNs are neural networks with feedback from output to input. The proposed macromodel structure based on RNN is shown in Fig. 1. The input layer includes a time varying input part $\mathbf{u}$ and time-invariant part $\mathbf{p}$. The output layer is time varying signal $\mathbf{y}$. For example, if the macromodel is used to represent an amplifier, then $\mathbf{u}$ will represent the amplifier input signal, $\mathbf{p}$ will represent the circuit parameters related to the amplifier, and $\mathbf{y}$ will be the output signal from amplifier.

The first hidden layer of the RNN, labeled $\mathbf{x}$ layer, contains buffered history of circuit output signal $\mathbf{y}$, which is fed back from the output of the RNN, buffered history of the circuit input signal $\mathbf{u}$, and the circuit parameter $\mathbf{p}$. Let the time sample index be k. Let K_y and K_u be the number of buffers for output $\mathbf{y}$ and

input parameter $\mathbf{u}$, respectively. Then neurons in the $\mathbf{x}$ layer is defined as

$$x_i(k) = \begin{cases} y_j(k-n), & \text{for } i=(n-1)N_y+j; \\ & 1 \le j \le N_y; \quad 1 \le n \le K_y \\ u_j(k-n), & \text{for } i=K_yN_y+(n-1)N_u+j; \\ & 1 \le j \le N_u; \quad 1 \le n \le K_u \\ p_n, & \text{for } i=n+K_yN_y+K_uN_u; \\ & 1 \le n \le K_p. \end{cases} \tag{4}$$

Let N_x be the total number of neurons in the $\mathbf{x}$ layer; we have $N_x = K_p + N_yK_y + N_uK_u$.

The next hidden layer is called the $\mathbf{z}$ layer. The neurons in this layer contain sigmoid activation functions. Let N_z be the total number of hidden neurons in the $\mathbf{z}$ layer. The vector of weights between the $\mathbf{x}$ and $\mathbf{z}$ layers is denoted as $\mathbf{w} = [w_{11}\ w_{12}\ \dots\ w_{N_zN_x}]^T$, and $\theta = [\theta_1\ \theta_2\ \dots\ \theta_{N_z}]^T$ denotes the bias vector of neurons in the $\mathbf{z}$ layer. For a given set of values in the $\mathbf{x}$ layer, the responses of hidden neurons in layer $\mathbf{z}$ can be computed from delayed input and output signals as

$$\begin{aligned} \gamma_j(k) &= \sum_{i=1}^{K_y}\sum_{l=1}^{N_y} y_l(k-i)w_{j[(i-1)N_y+l]} \\ &\quad + \sum_{i=1}^{K_u}\sum_{m=1}^{N_u} u_m(k-i)w_{j[K_yN_y+(i-1)N_u+m]} \\ &\quad + \sum_{i=1}^{K_p} p_i w_{j[K_yN_y+K_uN_u+i]} + \theta_j, \\ z_j(k) &= \phi\big(\gamma_j(k)\big), \qquad j=1,\dots,N_z \end{aligned} \tag{5}$$

where $\phi(\gamma)$ is a sigmoid function, $\phi(\gamma) = 1/(1+e^{-\gamma})$, and w_{ij} is the weight between the ith neuron in the $\mathbf{z}$ layer and the jth neuron in the $\mathbf{x}$ layer.

The last layer is the output layer called the $\mathbf{y}$ layer. The $\mathbf{y}$ layer outputs are linear functions of the responses of hidden neurons in the $\mathbf{z}$-hidden layer. Let $\mathbf{v} = [v_{11}\ v_{12}\ \dots\ v_{N_yN_z}]^T$ be the vector of weights between the $\mathbf{z}$ layer and $\mathbf{y}$ layer and $\eta = [\eta_1\ \eta_2\ \dots\ \eta_{N_y}]^T$ be the bias vector of output neurons in the $\mathbf{y}$ layer. The output of the overall macromodel is computed as

$$y_i(k) = \sum_{j=1}^{N_z} z_j(k)v_{ij} + \eta_i, \qquad i=1,\dots,N_y \tag{6}$$

where v_{ij} is the weight between the ith neuron in the $\mathbf{y}$ layer and the jth neuron in the $\mathbf{z}$ layer.

Let the parameters of the RNN be denoted as $\mathbf{\Phi}, \mathbf{\Phi} = [\mathbf{w}^T\theta^T\mathbf{v}^T\eta^T]^T$. The part in the macromodel structure from the $\mathbf{x}$ layer, $\mathbf{z}$ layer, to the $\mathbf{y}$ layer is a feed forward neural network (FFNN) denoted as $\mathbf{y}(\mathbf{x},\mathbf{\Phi})$. The overall neural network realizes a nonlinear relationship

$$\begin{aligned} \mathbf{y}(k) = f(&\mathbf{y}(k-1),\dots,\mathbf{y}(k-K_\mathbf{y}),\mathbf{u}(k-1),\dots, \\ &\mathbf{u}\big(k-K_u\big),\mathbf{p},\mathbf{\Phi}\big). \end{aligned} \tag{7}$$

Here, the number of delay buffers K_y and K_u represent the order of dynamics in the RNN macromodel.

D. Model Development and RNN Training

The RNN macromodel will not represent the nonlinear microwave circuit behavior unless it is trained by training data. Following the requirement described in Section II-B, the macromodel should be trained to match the output of the original circuit under various input excitations in the interval $[T_1, T_2]$. In our approach, training data is a set of input and output waveforms of the original nonlinear circuit. They can be collected through simulation and/or measurements. A second set of waveform data, called "testing data," should also be generated from the original circuit for model verification. The excitations used for generating testing waveforms should be different from those for generating training waveforms.

Let $\mathbf{y}(t)$ represent the output response of the RNN, and $\hat{\mathbf{y}}(t)$ represent the output waveform of original nonlinear circuit, i.e., the simulation and/or measurements. Let training data be represented by input–output waveforms $(\mathbf{u}_q(t), \hat{\mathbf{y}}_q(t)), T_1 \le t \le T_2, q=1,\dots,N_w$, where $\mathbf{u}_q(t)$ and $\hat{\mathbf{y}}_q(t)$ are the qth input and output waveforms and N_w is the total number of sample waveforms.

Each dynamic response $\hat{\mathbf{y}}(t)$ from the RNN macromodel over the time interval $[T_1, T_2]$ is also called an RNN output trajectory. The process of training is to minimize the difference between the RNN trajectory $\mathbf{y}(t)$ and circuit waveform data $\hat{\mathbf{y}}(t)$ under various input signals $\mathbf{u}(t)$. Let N_t be the number of time samples in $[T_1, T_2]$, and $y_{iq}(k)$ be the ith output signal at kth time sample in the qth trajectory of the RNN and $\hat{\mathbf{y}}_{iq}(k)$ be the corresponding waveform sample of the original nonlinear circuit. The objective for training the macromodel is

$$\min_{\mathbf{\Phi}} \frac{1}{2}\sum_{q=1}^{N_w}\sum_{i=1}^{N_y}\sum_{k=1}^{N_t}\big(y_{iq}(k) - \hat{\mathbf{y}}_{iq}(k)\big)^2. \tag{8}$$

In order to employ efficient gradient-based optimization methods in the training of the macromodel, derivatives of the error function for training with respect to each parameter in the RNN are required to form the Jacobian matrix $\mathbf{J}$. As shown in (7), the outputs of the RNN macromodel are fed back to the input layer. Therefore, output $\mathbf{y}$ is not only a function of input $\mathbf{u}$, but also the previous output of itself. Thus, conventional error back propagation method is not applicable for neural-network training. An advanced macromodel training scheme based on BPTT [23] is developed here to derive the Jacobian matrix.

For the kth training sample in the qth waveform $(\mathbf{u}_q(k), \hat{\mathbf{y}}_q(k))$, the l_2 error of the macromodel is

$$E_q(k) = \sum_{i=1}^{N_y}\frac{1}{2}\Big\{y_{iq}(k) - \hat{\mathbf{y}}_{iq}(k)\Big\}^2. \tag{9}$$

To obtain derivatives of $E_q(k)$ with respect to weights $\mathbf{\Phi}$ for training, we need $dy_{iq}(k)/d\mathbf{\Phi}$, which is computed recursively from history of $dy_{iq}(k)/d\mathbf{\Phi}$ in the following way.

For $k=1$, initialize $(dy_{iq}(k)/d\mathbf{\Phi}) = (\partial y_{iq}(k)/\partial\mathbf{\Phi})$, where the left-hand side is the derivative of the RNN macromodel and

$\partial y_{iq}(k)/\partial \Phi$ in the right-hand side is the derivative of the FFNN part in the macromodel between $\mathbf{x}$ and $\mathbf{y}$, which can be computed by normal back propagation [16]. The subsequent derivative $dy_{iq}(k)/d\Phi$ for the RNN for $k > 1$ can be computed by using the history of $dy_{iq}(k)/d\Phi$ as

$$\frac{dy_{iq}(k)}{d\Phi} = \frac{\partial y_{iq}(k)}{\partial \Phi} + \sum_{m=1}^{L_y} \sum_{j=1}^{N_y} \left\{ \frac{\partial y_{jq}(k)}{\partial x_{[j+(m-1)N_y]}} \frac{dy_{jq}(k-m)}{d\Phi} \right\} \tag{10}$$

where

$$L_y = \begin{cases} K_y, & \text{if } k > K_y \\ k-1, & \text{otherwise.} \end{cases} \tag{11}$$

The recurrent back propagation includes two parts. First, normal back propagation [16] is done through the FFNN between the $\mathbf{x}$ and $\mathbf{y}$ layers to get the partial derivative $\partial y_{iq}(k)/\partial \Phi$. This represents the sensitivity of the circuit output waveform with respect to macromodel internal parameters. Secondly, $\partial y_{iq}(k)/\partial \mathbf{x}(k)$ is obtained from further back propagation to the FFNN input layer $\mathbf{x}$. This derivative represents the dependency of the circuit output waveform between adjacent time points and can be written as

$$\begin{aligned} \frac{\partial y_{iq}(k)}{\partial x_{[j+(m-1)N_y]}} &= \sum_{r=1}^{N_z} \frac{dy_{iq}(k)}{dz_r} \frac{dz_r}{dx_{[j+(m-1)N_y]}} \\ &= \sum_{r=1}^{N_z} v_{ir} z_r (1-z_r) w_{r[j+(m-1)N_y]}, \\ &\qquad \text{for } j = 1, \ldots, N_y; \quad m = 1, \ldots, K_y. \end{aligned} \tag{12}$$

The result of $dy_{iq}(k)/d\Phi$ is stored and used recursively as the history for computing the derivative at the $k+1$ step. After this procedure, the Jacobian matrix of each sample with respect to each parameter in the RNN macromodel can be formulated as

$$\mathbf{J} = \begin{bmatrix} \left[\frac{\partial E_1(1)}{\partial \Phi} \quad \frac{\partial E_1(2)}{\partial \Phi} \quad \cdots \quad \frac{\partial E_1(N_t)}{\partial \Phi}\right]^T \\ \left[\frac{\partial E_2(1)}{\partial \Phi} \quad \frac{\partial E_2(2)}{\partial \Phi} \quad \cdots \quad \frac{\partial E_2(N_t)}{\partial \Phi}\right]^T \\ \vdots \\ \left[\frac{\partial E_{N_w}(1)}{\partial \Phi} \quad \frac{\partial E_{N_w}(2)}{\partial \Phi} \quad \cdots \quad \frac{\partial E_{N_w}(N_t)}{\partial \Phi}\right]^T \end{bmatrix}. \tag{13}$$

Based on this gradient scheme, gradient-based optimization algorithms, such as the Levenberg–Marquardt and quasi-Newton methods, are used to train the macromodel.[1] Once an RNN macromodel is trained, it can then represent the parametric and dynamic input–output relationship of the original nonlinear microwave circuit.

[1]Q. J. Zhang, NeuroModeler version 1.02, Dept. Electron., Carleton Univ., Ottawa, ON, Canada, 1999.

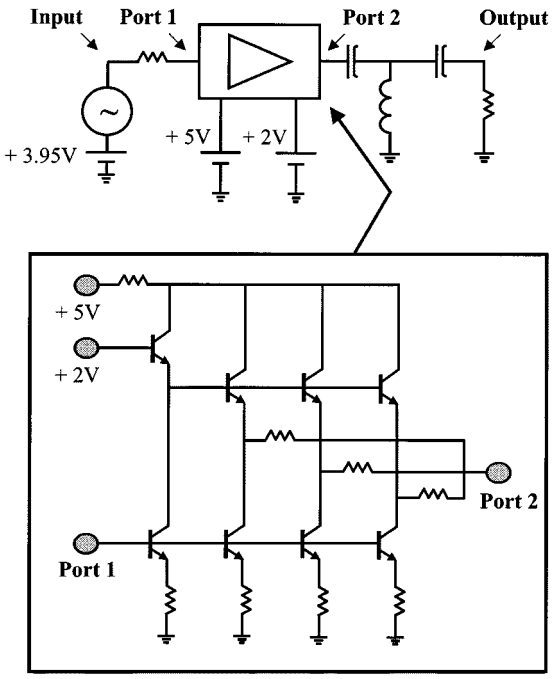


Fig. 2. Power-amplifier circuit to be represented by a macromodel based on an RNN.

E. Discussion

The number of delay buffers represents the order of the macromodel. Theoretically, to identify the complete dynamic characteristics, the number of delays should at least exceeds the number of states in the original circuit. For the purpose of model reduction and macromodeling, a system with much lower orders can predict the output of the original system successfully [14], [15]. By choosing K_y to be lower than the order of original system, our proposed RNN model automatically achieves model reduction effects. Another factor in the RNN model is the number of hidden neurons. Different number of hidden neurons represents the extent of nonlinearity between FFNN inputs and outputs (x and y). Too few hidden neurons cannot represent the nonlinearity sufficiently, while too many will lead to overlearning.

RNNs can be used to learn the dynamics both in the transient and steady-state stages, as will be shown in examples in Section III. When using an RNN macromodel, a good estimation of the initial state is necessary. For models trained using a transient waveform, the initial state can be considered to be zero or dc solution. If in the transient stage, the signal has sudden changes in the beginning: a separate RNN model with smaller sampling interval can be trained for initial state estimation. For models trained using steady-state waveform data, the initial states are not simply zeroes or constants. A time-delay neural network (TDNN) [24] can be trained using the information contained in training data and used to provide initial estimations.

III. Macromodeling Examples

A. RFIC Power-Amplifier Macromodeling Example

This example demonstrates the macromodeling of an RF integrated-circuit (RFIC) power amplifier, shown in Fig. 2 through

TABLE I
AMPLIFIER: RECURRENT TRAINING AND TESTING VERSUS DIFFERENT NUMBER OF HIDDEN NEURONS

Number of Hidden Neurons in z layer	Recurrent Training Error (3 buffers)	Recurrent Testing Error (3 buffers)
30	1.35e-2	1.43e-2
40	1.08e-2	1.11e-2
50	1.06e-2	1.04e-2
60	1.12e-2	1.19e-2

TABLE II
AMPLIFIER: COMPARISON OF RECURRENT MODEL AGAINST DIFFERENT NUMBERS OF BUFFERS

Number of buffers (K)	Recurrent Training Error	Recurrent Testing Error
1	3.11e-2	3.00e-2
2	1.81e-2	1.83e-2
3	1.06e-2	1.04e-2
4	9.10e-3	9.33e-3

the proposed RNN approach. The choice of a power amplifier is due to the fact that it is a key circuit in wireless-communication systems. The amplifier contains eight n-p-n bipolar junction transistors (BJTs) modeled by two internal HP-ADS nonlinear models Q34 and Q37.[2] Input parameters for the RNN are the voltage waveform of amplifier input and their sampling intervals. Output of the RNN is the voltage waveform of the amplifier output.

The first macromodel is constructed to represent the transient characteristics of the amplifier. The sampling intervals are changed with frequency so that 50 points per cycle are ensured. The training waveform is gathered by exciting the circuit with a set of signal frequencies ($f = 0.8, 1.0, 1.2$ GHz) and amplitudes ($A = 0.2, 0.3, \ldots, 1.3$ V). Testing data is generated using frequencies (0.9, 1.1 GHz) and amplitudes ($0.25, 0.35, \ldots, 1.25$ V), which are different from those used in training.

Training results and testing results are listed in Table I. Different numbers of delay buffers K ($K = K_u = K_y$) are tried in the RNN, as shown in Table II.

A set of sample test waveforms is shown in Fig. 3 for the RNN at frequencies of 0.9 and 1.1 GHz, and amplitudes 0.55 and 1.15 V. The RNN macromodel can reproduce amplifier output accurately even when amplifier input waveform is different from those used in training. Moreover, an RNN macromodel can generate similar output as original simulation with much faster speed. The evaluation of 900 different sets of input–output waveforms takes 10 s by the RNN macromodel and 177 s by the original simulation.

A second macromodel is trained to learn the large-signal behavior of the amplifier in the steady state. The sampling cycle is also proportional to frequency so that 50 points per cycle are ensured. The training waveform is generated from HP-ADS with the following set of excitation signal: $f = 0.8, 0.9, \ldots, 1.2$ GHz and $A = 0.2, 0.3, \ldots, 1.2$ volts. Testing data is gathered for two cycles using a set of samples different from those used in training ($f = 0.85, 0.95, \ldots, 1.15$ GHz and $A = 0.25, 0.35, \ldots, 1.15$ V). The overall accuracy (average test error of all test waveforms) is 0.728%. Fig. 4 shows the comparison of the RNN with a set of sample test waveforms (with three cycles) excited at sample frequencies of 0.95 and 1.05 GHz, and amplitudes of 0.45 and 1.15 V. As expected, the RNN macromodel can reproduce the amplifier output accurately even though these test waveforms were never used in training. Moreover, the training was done with one cycle of waveform data. In the testing of Fig. 4, the RNN can reliably predict the signal well beyond one cycle.

[2]HP-ADS version 1.3, Agilent Technologies, Santa Rosa, CA, 1999.

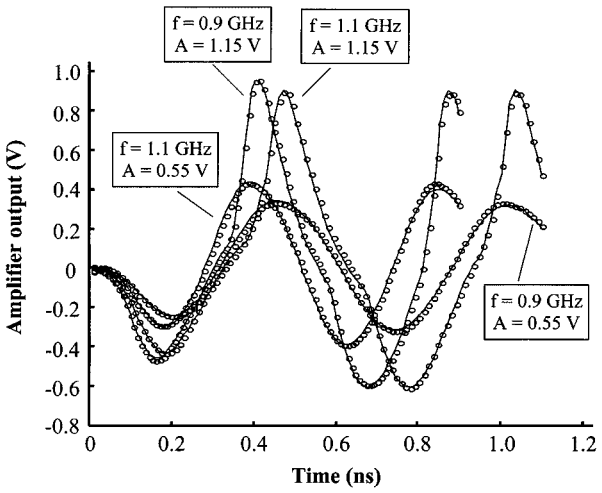


Fig. 3. Comparison between output waveforms from original amplifier (∘) and that from an RNN macromodel with three buffers (-) trained in transient state. Good agreement is achieved even though these waveforms have never been used in training.

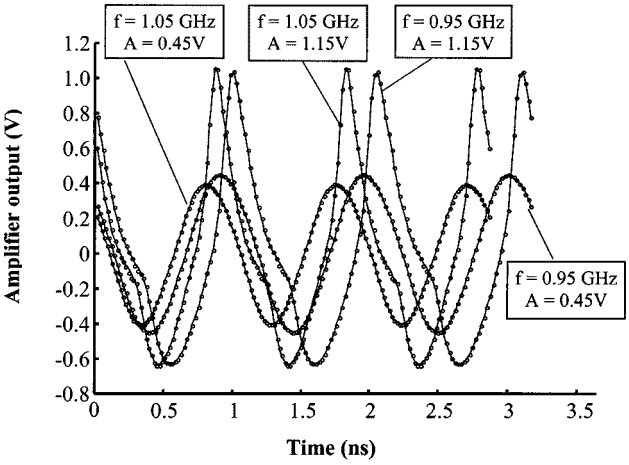


Fig. 4. Comparison between output waveforms from original amplifier (∘) and that from an RNN macromodel with three buffers (-) trained in steady state. Good agreement is achieved even though these waveforms have never been used in training.

B. RF Mixer Macromodeling Example

This example demonstrates the RNN method in macromodeling a mixer, which is a basic circuit in any transmission-reception process. The circuit is a Gilbert cell, shown in Fig. 5. The internal subcircuit contains 14 n-p-n BJT transistors (modeled by five different HP-ADS nonlinear models $M1$–$M5$). The RNN macromodel for the mixer has three inputs, namely, RF

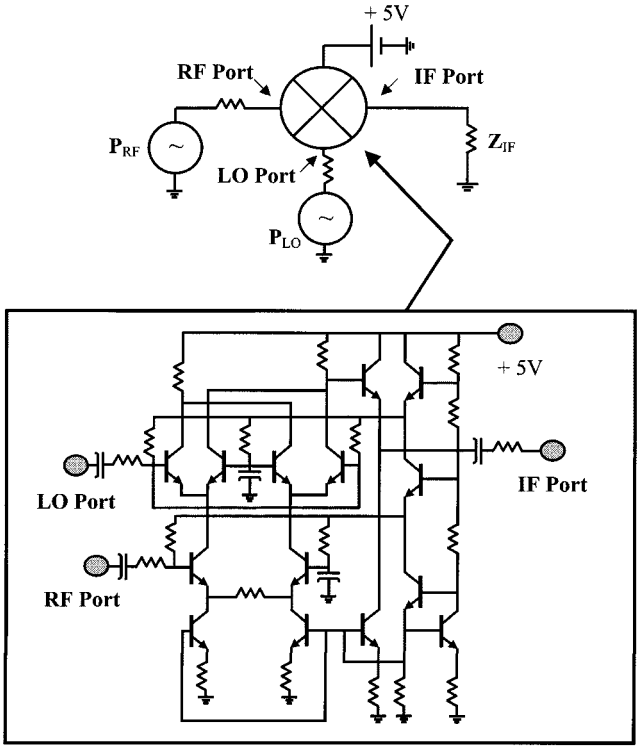


Fig. 5. Mixer circuit to be modeled by a macromodel based on an RNN.

TABLE III
MIXER: RECURRENT TRAINING AND TESTING VERSUS DIFFERENT NUMBER OF HIDDEN NEURONS

Number of Hidden Neurons in z layer	Recurrent Training Error (3 buffers)	Recurrent Testing Error (3 buffers)
30	7.77e-3	5.69e-3
40	5.53e-3	3.60e-3
50	6.03e-3	4.17e-3
60	6.40e-3	5.40e-3

TABLE IV
MIXER: COMPARISON OF RECURRENT MODEL AGAINST DIFFERENT NUMBERS OF BUFFERS

Number of buffers (K)	Recurrent Training Error	Recurrent Testing Error
1	8.81e-2	7.85e-2
2	2.59e-2	2.71e-2
3	6.03e-3	4.17e-3
4	8.21e-3	7.89e-3

input waveform, local oscillator (LO) input waveform, and IF load impedance. The sampling cycle is fixed proportionally to the highest frequency to be trained.

The first macromodel is constructed to model the dynamics of the mixer in the transient state. The training data is gathered in the following way: the RF frequency and power level changed from 1.8 to 3.0 GHz with a step size of 0.1 GHz, and from -60 to -30 dBm with a step size of 5 dBm, respectively. The LO signal is fixed at 1.75 GHz and -5 dBm. IF Load impedance is sampled at 30, 40, 50, 60, and 70 Ω. The transient time range for training $[T_1, T_2]$ is $[0, 1$ ns$]$. Test data is given using different set of samples from those used in training: RF frequencies (2.35, 2.55, 2.75 GHz), RF power levels (-38, -43, -47, -48, -53 dBm) and for a 50-Ω-load IF impedance with transient time up to 2 ns.

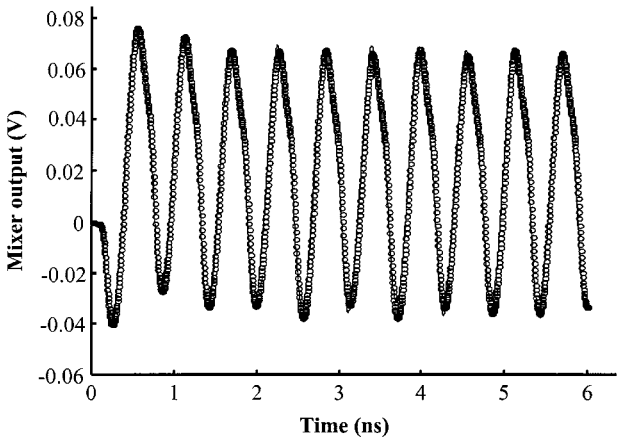


Fig. 6. Comparison between output waveforms from original mixer ($\circ$) and that from an RNN macromodel with three buffers (-) trained in transient state. Good agreement is achieved even though these waveforms have not been used in training ($f_{\mathrm{RF}} = 2.55$ GHz, $P_{\mathrm{RF}} = -48$ dBm, $Z_{\mathrm{IF}} = 50\ \Omega$).

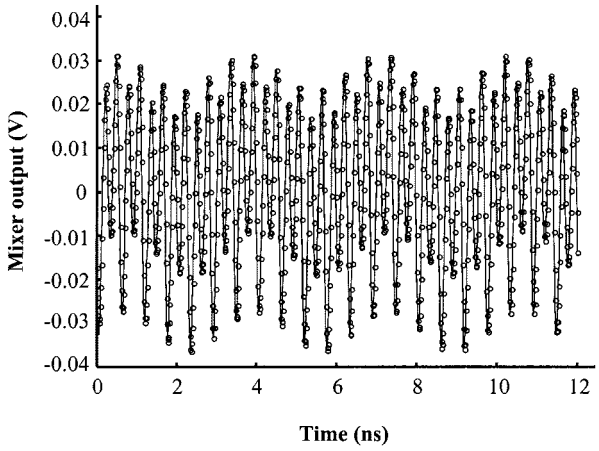


Fig. 7. Comparison between output waveforms from original mixer ($\circ$) and that from an RNN macromodel with three buffers ($-$) trained in steady state. Good agreement is achieved even though these waveforms have not been used in training. ($f_{\mathrm{RF}} = 2.05$ GHz, $P_{\mathrm{RF}} = -47$ dBm, $Z_{\mathrm{IF}} = 35\ \Omega$). This is one of the many test waveforms used.

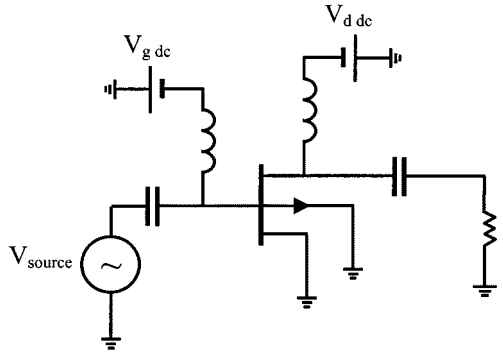


Fig. 8. Circuit used to generate training data for a transistor to be represented by an RNN macromodel. The training data is obtained using HSpice simulation with BSIM3—level 49 transistor model.

The training error and testing error are listed in Table III. Results of the RNN with a different number of buffers K ($K =$

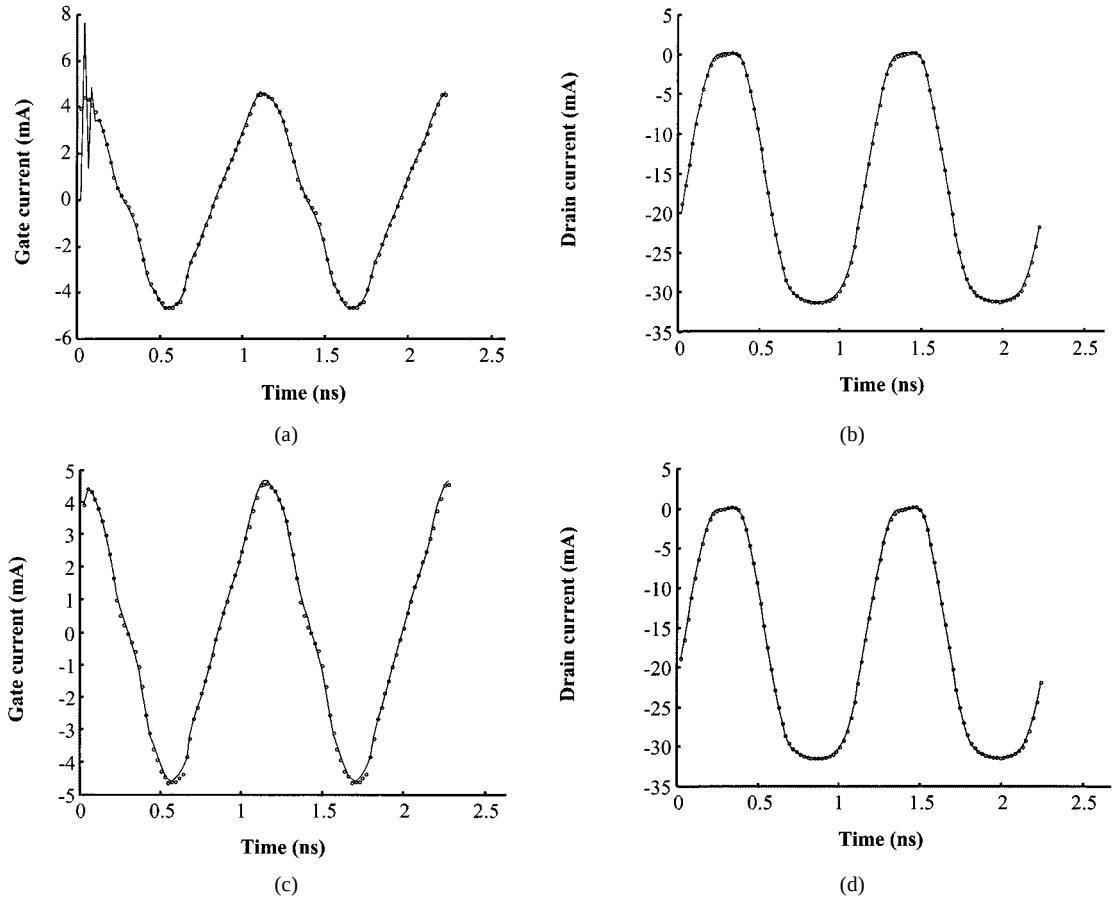


Fig. 9. Effect of initial state estimation and comparison between output waveforms from original transistor simulation ($\circ$) and that from an RNN macromodel with two buffers (-) for $f = 0.9$ GHz, $V_{\text{source}} = 1.55$ V, $V_{g\,\text{dc}} = -2.125$ V, and $V_{d\,\text{dc}} = -3.25$ V. (a) Gate current: without initial state estimation. (b) Drain current: without initial state estimation. (c) Gate current: with initial state estimation by an initial RNN. (d) Drain current: with initial state estimation by an initial RNN.

$K_u = K_y$) are shown in Table IV. Fig. 6 shows an example test result of an RNN at the RF frequency of 2.55 GHz, RF power level -48 dBm, IF load impedance of 50 Ω, and transient time up to 6 ns. Again, using the macromodel, the original analog dynamic behavior can be retained and the evaluation of such behavior is much faster than original circuit simulation.

A second macromodel is constructed to learn the behavior of the mixer in steady state. Training data is generated as follows: RF frequency and power level are changed from 1.8 to 3.0 GHz with a step size of 0.1 GHz and from -50 to -30 dBm with a step size of 5 dBm. LO signal frequency is 1.75 GHz and power level is -5 dBm. IF load impedance is sampled at 30, 40, 50, 60 and 70 Ω, and time range is up to 4 ns. Test data is given for a different set of RF frequencies (1.85, 1.95, ..., 2.95 GHz), RF power levels (-47, -42, -37, and -32 dBm) and for a load IF impedance sampled at 35, 45, 55, and 65 Ω with time range up to 12 ns. The overall test error for all the waveforms is 0.541%; the RNN macromodel can predict the output waveforms accurately even beyond the time range used in training. Fig. 7 shows an example test result of the RNN with RF frequency equal to 2.05 GHz, RF power level -47 dBm, and IF load impedance of 35 Ω. Again, using the macromodel, the original analog dynamic behavior can be retained and the evaluation of such behavior is much faster than original circuit simulation.

C. MOSFET Time-Domain Macromodeling Example

The last example is a transient device level example of a p-MOSFET transistor. The training data is collected by HSpice simulation using BSIM3 (level 49) model.[3] The physical parameters of this model are a length of $L = 0.4$ μm and a width of $W = 170$ μm.

The RNN macromodel has two inputs, namely, gate and drain voltage waveforms. The two outputs of the RNN are gate and drain current waveforms (I_g and I_d). To generate the training and testing waveform data, we used the circuit shown in Fig. 8. Sampling intervals are proportional to the frequency, thus, 50 points per cycle are ensured. The data is gathered by varying frequencies of excitation signal: $f = 0.8, 1.0, 1.2, 1.8, 2.0, 2.2$ GHz, source amplitudes: $V_{\text{source}} = 0.8, 1.0, 1.2, 1.4, 1.5, 1.6$ V, gate bias voltages: $V_{g\,\text{dc}} = -3.0, -2.75, \ldots, -2.0$ V, and drain bias voltages: $V_{d\,\text{dc}} = -5.0, -4.5, \ldots, -3.0$ V with transient up to two cycles.

Testing data are waveforms simulated in HSpice using a set of samples different from those in training (frequency $=0.9, 1.1, 1.9, 2.1$ GHz, source amplitudes: $V_{\text{source}} = 0.9, 1.1, 1.3, 1.45, 1.55$ V, gate bias: $V_{g\,\text{dc}} = -2.875, -2.625, \ldots, -2.125$ V, drain bias: $V_{d\,\text{dc}} =$

[3]HSPICE, Avant! Corporation, Fremont, CA 1996.

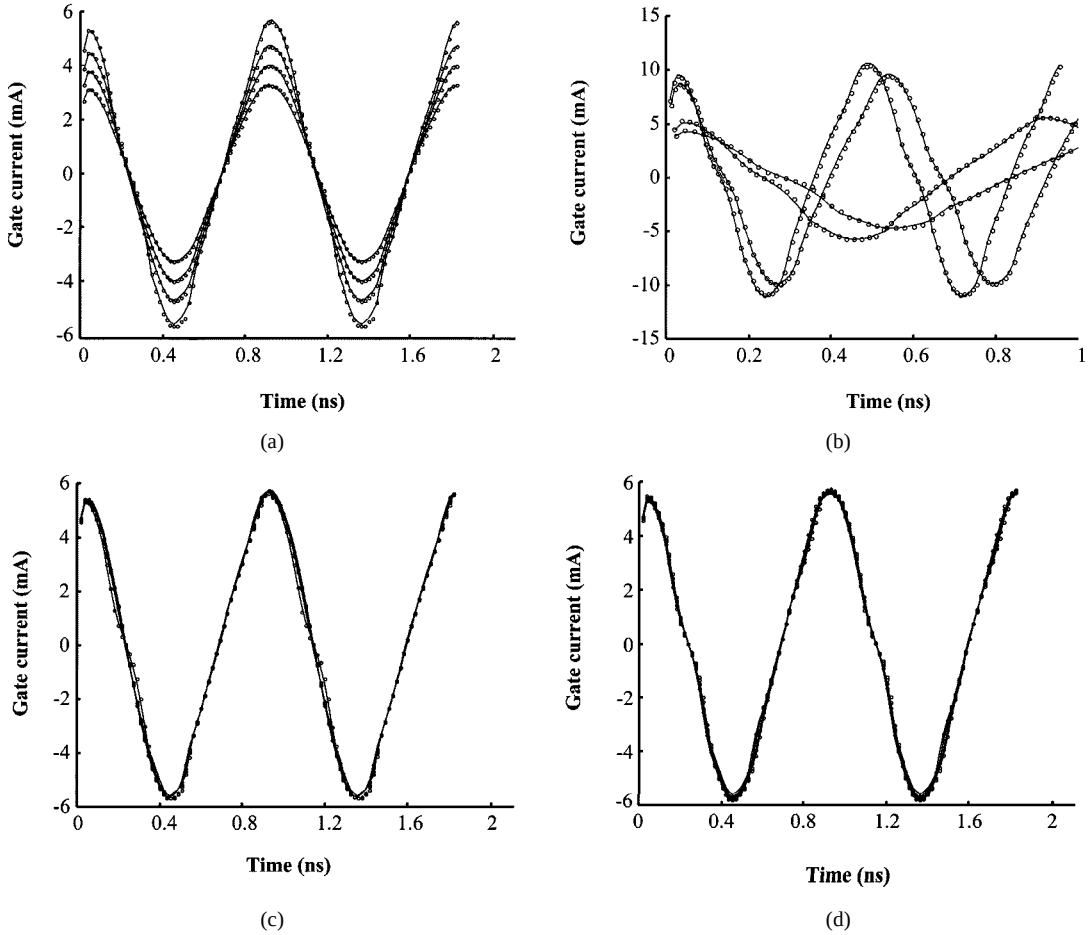


Fig. 10. Comparison between output waveforms of gate current from original transistor ($\circ$) and that from an RNN macromodel with two buffers (-) under various excitation with different parameters. Initial estimation by an initial RNN was used. The macromodel matches test data very well even though those various test waveforms have never been used in training. (a) Different $V_{\rm source}$ with $f = 1.1$ GHz, $V_{g\,\rm dc} = -2.125$ V, and $V_{d\,\rm dc} = -3.25$ V. (b) Different frequencies with $V_{g\,\rm dc} = -2.125$ V, $V_{d\,\rm dc} = -3.25$ V, and $V_{\rm source} = 1.55$ V. (c) Different $V_{g\,\rm dc}$ with $f = 1.1$ GHz, $V_{d\,\rm dc} = -3.25$ V, and $V_{\rm source} = 1.55$ V. (d) Different $V_{d\,\rm dc}$ with $f = 1.1$ GHz, $V_{g\,\rm dc} = -2.125$ V, and $V_{\rm source} = 1.55$ V.

$-4.75, -4.25, \ldots, -3.25$ V) with transient up to two cycles.

A separate RNN macromodel is trained to estimate the initial conditions for the above model. The training data for this initial RNN includes short segments of various training waveforms sampled at high sampling rate. The sampling interval of the training data for this initial macromodel is one-fifth of that in the main RNN macromodel. Three delay buffers are used. This initial RNN helps to accommodate the sudden rise and fall of the signal in the beginning. Fig. 9 shows the response of RNN under the same excitation with and without initial estimations.

The overall accuracy (test error) of the macromodel with initial estimation with respect to all test waveforms is 0.303%. The RNN macromodel can predict both gate and drain currents accurately even when the exciting gate and drain voltage waveforms are different from what is used in training. Figs. 10 and 11 show the examples of different test results with different set of variables (with varying source amplitude, frequency, and gate and drain bias voltages, respectively). The output trajectory from the proposed macromodel matches the test waveform from the original MOSFET very well under various excitations even though these waveforms have never been used in training.

D. Comparison Between Standard Neural Network and the Proposed RNN Methods for Nonlinear Macromodeling

In order to compare the proposed recurrent neural model with the conventional non-RNN model, the three examples in this section are also used to develop a conventional FFNN, and time-delay neural-network (TDNN) models. In an FFNN, the output is dependent only on the inputs at the same instantaneous time, while in a TDNN, only the history of input signals are used as the inputs of the macromodel. Both FFNNs and TDNNs are nonrecurrent models since they have no feedback from outputs to input. The test errors for FFNN, TDNN, and RNN models are listed in Table V.

As can be observed from the table, the proposed RNN method gives the best modeling results. The conventional FFNN model has poor accuracy since it can only represent a static input–output relationship and is not suitable for representing the dynamic behavior in the examples. The TDNN model is an improvement over the FFNN because a history of inputs is used in training, representing a partial dynamic information. However, the circuit output may be different even with the same input because of the differences in the history of the circuit responses. In this case, the nonrecurrent models (i.e., FFNN and TDNN) will try to learn the average between

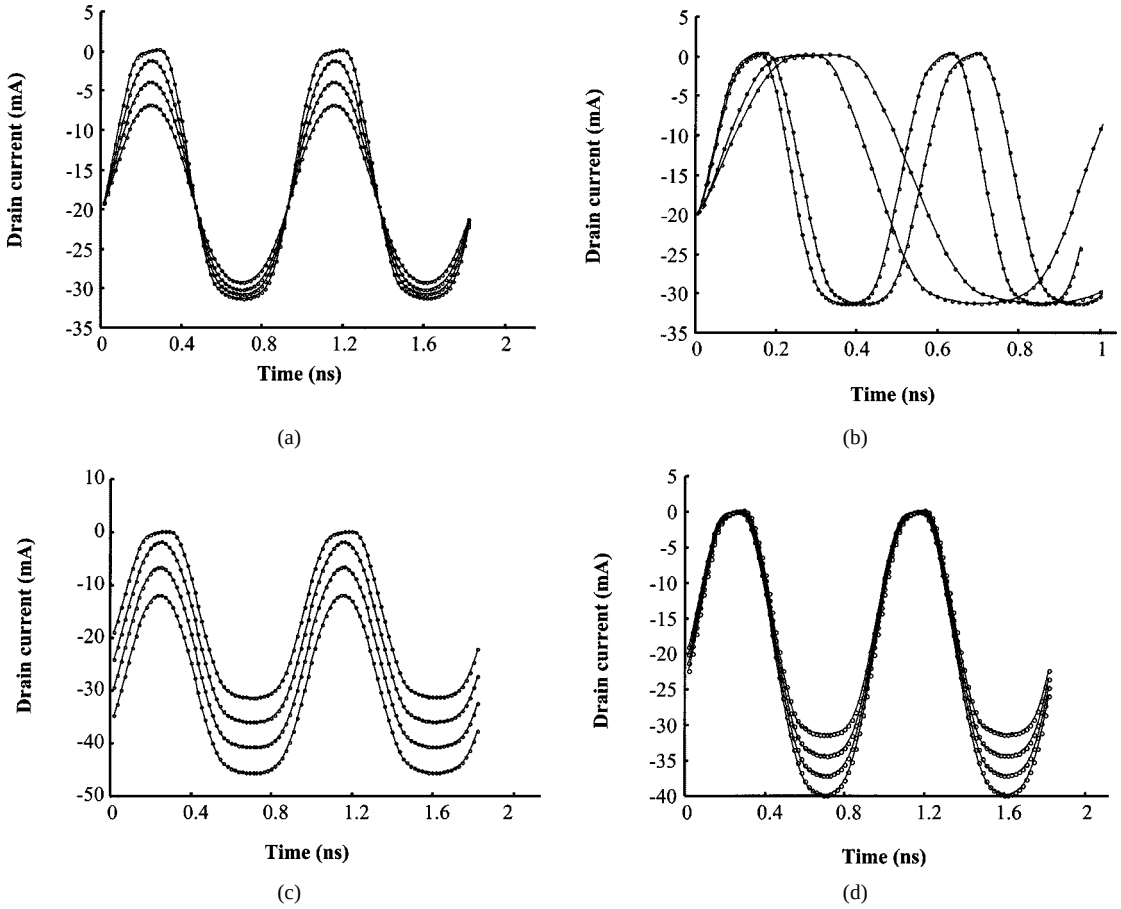


Fig. 11. Comparison between output waveforms of drain current from original transistor (∘) and that from an RNN macromodel with two buffers (-) under various excitation with different parameters. Initial estimation by an initial RNN was used. The macromodel matches test data very well even though those various test waveforms have never been used in training. (a) Different $V_{\rm source}$ with $f = 1.1$ GHz, $V_{g\,\rm dc} = -2.125$ V, and $V_{d\,\rm dc} = -3.25$ V. (b) Different frequencies with $V_{g\,\rm dc} = -2.125$ V, $V_{d\,\rm dc} = -3.25$ V, and $V_{\rm source} = 1.55$ V. (c) Different $V_{g\,\rm dc}$ with $f = 1.1$ GHz, $V_{d\,\rm dc} = -3.25$ V, and $V_{\rm source} = 1.55$ V. (d) Different $V_{d\,\rm dc}$ with $f = 1.1$ GHz, $V_{g\,\rm dc} = -2.125$ V, and $V_{\rm source} = 1.55$ V.

TABLE V
COMPARISON OF TEST ERRORS BETWEEN NONRECURRENT AND THE PROPOSED RECURRENT MODELS FOR NONLINEAR MODELING

	Standard FFNN Method	Standard TDNN Method	Proposed RNN Method
Amplifier Macromodel	1.56e-1	3.42e-2	1.04e-2
Mixer Macromodel	9.61e-2	8.12e-2	3.60e-3
MOSFET Macromodel	6.37e-3	4.32e-3	3.03e-3

different outputs when the inputs are the same and, therefore, cannot give accurate modeling solutions. The RNN method takes the history of outputs as additional inputs, thus feeding the RNN with rich information identifying differences between different dynamic states of the circuit, and giving the best modeling accuracy among all the methods.

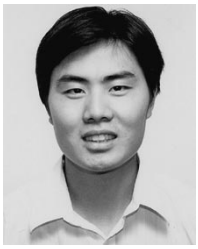
IV. CONCLUSION

For the first time, an RNN macromodeling approach for nonlinear microwave circuits has been presented. As a departure from the artificial neural-network methodology used for passive linear device modeling, the RNN macromodel provides fast and accurate representation of the time-domain dynamic behavior of the original circuit. The method is based on learning the input–output waveform and does not require conventional manual construction of an equivalent circuit. It opens a new potential for automatic model generation through neural-network learning.

REFERENCES

[1] *Int. J. RF Microwave CAE (Special Issue Applicat. ANN to RF and Microwave Design)*, vol. 9, no. 3, 1999.
[2] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349–1358, June 1995.
[3] P. Burrascano and M. Mongiardo, "A review of artificial neural networks applications in microwave CAD," *Int. J. RF Microwave CAE (Special Issue Applicat. ANN to RF and Microwave Design*, vol. 9, pp. 158–174, 1999.
[4] 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.
[5] 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.
[6] P. M. Watson, K. C. Gupta, and R. L. Mahajan, "Development of knowledge based artificial neural network models for microwave components," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Baltimore, MD, 1998, pp. 9–12.

[7] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "Device and circuit-level modeling using neural networks with faster training based on network sparsity," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1696–1704, Oct. 1997.
[8] S. Goasguen, S. M. Hammadi, and S. M. El-Ghazaly, "A global modeling approach using artificial neural network," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Anaheim, CA, 1999, pp. 153–156.
[9] P. Viszmuller, *RF Design Guide, Systems, Circuits, and Equations*. Norwood, MA: Artech House, 1995.
[10] T. R. Turlington, *Behavioral Modeling of Nonlinear RF and Microwave Devices*. Norwood, MA: Artech House, 2000.
[11] J. Verspecht, F. Verbeyst, M. Vanden Bossche, and P. Van Esch, "System level simulation benefits from frequency domain behavioral models of mixers and amplifiers," in *Proc. 29th European Microwave Conf.*, vol. 2, Munich, Germany, 1999, pp. 29–32.
[12] T. Wang and T. J. Brazil, "A Volterra mapping-based S-parameter behavioral model for nonlinear RF and microwave circuits and systems," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Anaheim, CA, 1999, pp. 783–786.
[13] M. I. Sobhy, E. A. Hosny, M. W. R. Ng, and E. A. Bakkar, "Nonlinear system and subsystem modeling in time domain," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2571–2579, Dec. 1996.
[14] G. Casinovi and A. Sangiovanni-Vincentelli, "A macromodeling algorithm for analog circuits," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 150–160, Feb. 1991.
[15] P. K. Gunupudi and M. S. Nakhla, "Model-reduction of nonlinear circuits using Krylov-space techniques," in *Proc. IEEE Int. Design Automation Conf.*, New Orleans, LA, June 1999, pp. 13–16.
[16] S. Haykin, *Neural Networks, A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[17] J. Aweya, Q. J. Zhang, and D. Y. Montuno, "A direct adaptive neural controller for flow control in computer networks," in *Proc. IEEE Int. Neural Networks Conf.*, Anchorage, AK, May 1998, pp. 140–145.
[18] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, Jan. 1990.
[19] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks—Part II: Observability, identification, and control," *IEEE Trans. Neural Networks*, vol. 7, pp. 30–42, Jan. 1996.
[20] T. Hrycej, *Neurocontrol: Toward an Industrial Control Methodology*. New York: Wiley, 1997.
[21] S. Omatu, M. Khalid, and R. Yusof, *Neuro-Control and Its Applications*. Berlin, Germany: Springer-Verlag, 1996.
[22] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrand, 1983.
[23] B. A. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks: A survey," *IEEE Trans. Neural Networks*, vol. 6, pp. 1212–1228, Sept. 1995.
[24] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time delay neural networks," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 37, pp. 328–339, Mar. 1989.



Yonghua Fang (S'99) was born in Zhejiang, China, in November 1976. He received the B.Eng. degree in computer science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1997, and is currently working toward the M.Eng. degree in electronics at Carleton University, Ottawa, ON, Canada.

His research interests include CAD of very large scale integration (VLSI) circuits and application of artificial intelligence in microwave device and circuit modeling.

Mustapha C. E. Yagoub (M'96) received the Diplôme d'Ingenieur 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.

From 1983 to 1991 and 1994 to 1999, he was with the Institute of Electronics, Université des Sciences et de la Technologie Houari Boumédienne, Algiers, Algeria, first as an Assistant and then as an Assistant Professor. Since 1999, he has been a Visiting Research Scholar in the Department of Electronics, Carleton University, Ottawa, ON, Canada. His research interests include neural networks, linear and nonlinear modeling, and optimization of microwave active circuits. He has authored or co-authored over 50 publications in theses area in international journals and conferences. He also co-authored *Conception de Circuits Linéaires et Non Linéaires Micro-ondes* (Toulouse, France: Cépadues, 2000).

Fang Wang (S'94) received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China, in 1993, and the M.Eng. and Ph.D. degrees from Carleton University, Ottawa, ON, Canada, in 1995 and 1999, respectively.

Her research interests include neural networks, modeling and their applications in CAD for electronics circuits.

She was a recipient of a 1997–1999 Postgraduate Scholarship presented by the Natural Science and Engineering Research Council of Canada (NSERC). She was also a two-time recipient of the Ontario Graduate Scholarship (1996 and 1997). She also received travel fellowship twice from the International Conference on Neural Networks (1997 and 1998).

Qi-Jun Zhang (S'84–M'85-SM'95) received the B.Eng. degree from the East China Engineering Institute, Nanjing, China, in 1982, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 1987.

From 1982 to 1983, he was with the System Engineering Institute, Tianjin University, Tianjin, China. From 1988 to 1990, he was with Optimization Systems Associates Inc. (OSA), Dundas, ON, Canada, where he developed advanced microwave optimization software. In 1990, he joined the Department of Electronics, Carleton University, Ottawa, ON, Canada, where he is currently a Professor. His research interests are neural network and optimization methods for high-speed/high-frequency circuit design. He has authored *Neural Networks for RF and Microwave Design* (Norwood, MA: Artech House, 2000), co-edited *Modeling and Simulation of High-Speed VLSI Interconnects* (Norwell, MA: Kluwer, 1994), contributed to *Analog Methods for Computer-Aided Analysis and Diagnosis*, (New York: Marcel Dekker, 1988), was Guest Co-Editor for a "Special Issue on High-Speed VLSI Interconnects" of the *International Journal of Analog Integrated Circuits and Signal Processing* (Norwood, MA: Kluwer, 1994), was a Guest Editor for the first "Special Issues on Applications of ANN to RF and Micorowave Design" for the *International Journal of RF and Microwave Computer-Aided Engineering* (New York: Wiley, 1999), and was a Guest Editor for the second "Special Issues on Applications of ANN to RF and Microwave Design" of the *International Journal of RF and Microwave Computer-Aided Engineering* (New York: Wiley, to be published 2001).

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