Advances of Neural Network Modeling Methods for RF/Microwave Applications

Humayun Kabir, Yi Cao, Yazi Cao, and Qi-Jun Zhang
Department of Electronics
Carleton University, Ottawa, ON K2C3L5, Canada
hkabir@doe.carleton.ca, ycao@doe.carleton.ca, yacao@doe.carleton.ca, qjz@doe.carleton.ca

***Abstract*—** This paper provides an overview of recent advances of neural network modeling techniques which are very useful for RF/microwave modeling and design. First, we review neural network inverse modeling method for fast microwave design. Conventionally, design parameters are obtained using optimization techniques by multiple evaluations of EM-based models, which take a long time. To avoid this problem, neural network inverse models are developed in a special way, such that they provide design parameters quickly for a given specification. The method is used to design complex waveguide dual mode filters and design parameters are obtained faster than the conventional EM-based technique while retaining comparable accuracy. We also review recurrent neural network (RNN) and dynamic neural network (DNN) methods. Both RNN and DNN structures have the dynamic modeling capabilities and can be trained to learn the analog nonlinear behaviors of the original microwave circuits from input-output dynamic signals. The trained neural networks become fast and accurate behavioral models that can be subsequently used in system-level simulation and design replacing the CPU-intensive detailed representations. Examples of amplifier and mixer behavioral modeling using the neural-network-based approach are also presented.

***Index Terms*—** Behavioral modeling, computer aided design, neural network.

I. INTRODUCTION

Neural network is an information processing system, which can learn from observation and generalize any arbitrary input-output relationship similar to human brain function. It has been used in many modeling and design applications [1], [2] such as vias [3], transistor [4], amplifier [5], filters [6–9], etc. Neural network can capture multidimensional non-linear device behavior accurately. The evaluation of a neural network model is also fast. These unique qualities make neural network a useful alternative of EM-based modeling.

Models developed in the conventional approach are termed as forward model where the inputs are the physical or geometrical parameters such as dielectric, length, width etc. and the outputs are electrical parameters such as S-parameters. For design purpose, the EM simulator or the forward model is evaluated repetitively in order to find the optimal solutions of the geometrical parameters that can lead to a good match between modeled and specified electrical parameters. An example of such an approach is [10]. Conversely, an inverse model is defined as the opposite to the forward model such that the geometrical or physical parameters become the outputs and electrical parameters become the inputs of the inverse model. The inverse model provides the required geometrical solution for a given electrical specification. This avoids repetitive model evaluation.

Recently an inverse modeling methodology using neural network technique has been presented [8]. The training data for the inverse model is generated using a forward EM model or from device measurement. The training data is reorganized such that the geometrical parameters become outputs and the electrical parameters become inputs. A neural network model trained using this reorganized model becomes the inverse model of the original EM-model or device. However, this process of reorganizing data may lead to a non-uniqueness problem where multiple solutions may exist for a single input value. We call it a multi-valued problem. If two or more different input values lead to a single value in the forward model, then contradiction arises in the training data of the inverse model. As a result, neural network faces hard time to provide accurate

1054-4887 © 2010 ACES

solution at those points. To avoid these situations the data is first checked for contradictions. If contradiction exists, the inverse data is divided into sub-groups such that each sub-group does not contain any contradictory data. Multiple sub-models are then developed using the divided data. The sub-models are then combined using a special technique to form the overall inverse model. The description of various techniques is provided in the Section 2.

We also review the recent advances of neural network approaches for behavioral modeling of nonlinear microwave circuits. We focus on the specific artificial neural network (ANN) structures that are capable of learning and representing dynamic behaviors of nonlinear circuit blocks. Two ANN-based techniques, i.e., recurrent neural networks (RNN) [11–13] and dynamic neural networks (DNN) [14] techniques, are described from the perspective of nonlinear behavioral modeling. Numerical examples of modeling RF amplifiers and mixers are included.

II. INVERSE MODELING METHODS

A. Formulation of Inverse Model

Let $\boldsymbol{x}$ be an n-vector containing the inputs and $\boldsymbol{y}$ be an m-vector containing the outputs of the forward model. Then the forward modeling problem can be expressed as

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{f}$ defines input-output relationship, $\boldsymbol{x} = [x_1\ x_2\ x_3\ \ldots\ x_n]^T$, and $\boldsymbol{y} = [y_1\ y_2\ y_3\ \ldots\ y_m]^T$. Then the inverse model can be defined as

$$\overline{\boldsymbol{y}} = \overline{\boldsymbol{f}}(\overline{\boldsymbol{x}}), \tag{2}$$

where $\overline{\boldsymbol{f}}$ defines the inverse input-output relationship, $\overline{\boldsymbol{y}}$ and $\overline{\boldsymbol{x}}$ contains outputs and inputs of the inverse model respectively. As an example, if a device contain four inputs and three outputs then $\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4]^T$ and $\boldsymbol{y} = [y_1\ y_2\ y_3]^T$. If input parameters x_3 and x_4 are design parameters, e.g., iris length and width of a waveguide filter and y_2 and y_3 are electrical parameters, e.g., couplings of the filter, then inputs of the inverse model become $\overline{\boldsymbol{x}} = [x_1\ x_2\ y_2\ y_3]^T$ and output vector becomes $\overline{\boldsymbol{y}} = [y_1\ x_3\ x_4]^T$. Figure 1 shows the diagrams of a neural network forward model and an inverse model. The inverse model of Fig. 1(b) is formulated by swapping partial inputs and outputs of the forward model of Fig. 1(a). Note that two input parameters and two output parameters are swapped.

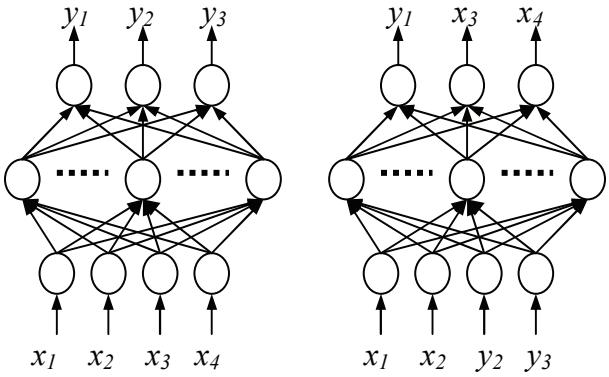


Fig. 1. Example illustrating neural network forward and inverse models, (a) forward model (b) inverse model. The inputs x_3 and x_4 (output y_2 and y_3) of the forward model are swapped to the outputs (inputs) of the inverse model respectively [8].

B. Non-Uniqueness in Inverse Training Data

If two different input values in forward model lead to the same value of output then a contradiction arises in the training data of the inverse model, because the single input value in the inverse model has two different output values (therefore contradictory data). Since we cannot train the neural network inverse model to match two contradictory data simultaneously, it is important to detect the existence of contradictions.

Detection of contradiction would have been straightforward if the training data were generated by deliberately choosing different geometrical dimensions such that they lead to the same electrical value. However in practice, the training data are not sampled at exactly those locations. Therefore, we develop numerical criteria to detect the existence of contradictions. We calculate the slope between samples within a specific neighborhood. A slope between two samples was calculated by dividing the normalized difference

of the *y*-values of the two samples with the normalized difference of the *x*-values of the two samples. If any of the slopes becomes larger than some user defined threshold value, then the data may contain contradictory samples. In that case we need to divide the data into groups such that the individual groups do not contain any contradiction. In this way we solve the problem of non-uniqueness of input-output relationship and thus contradictory sample in the inverse training data.

C. Method to Divide Inverse Training Data

If existence of contradictions is detected in training data, we perform data preprocessing. All the data samples, even though contradictory, are useful information and should not be deleted from the training data. In our method, we divide the data into groups so that contradictory samples are separated into different groups and the data in each group becomes free of contradiction. We divide the overall training data into groups based on derivatives of outputs vs. inputs of the forward model. Because variations in the output response changes directions with the change of input variables and multivalued problems occur when the response changes to a reverse direction. Thus derivative information is a logical criterion to detect such reverse phenomena. Let us define the derivatives of inputs and outputs that have been exchanged to formulate the inverse model, evaluated at each sample, as,

$$\left.\frac{\partial y_i}{\partial x_j}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(k)}},\ i \in I_y \text{ and } j \in I_x, \tag{3}$$

where, $k = 1,2,3,...,N_S$, N_S is the total number of training samples, I_x is an index set containing the indices of inputs of forward model that are moved to the output of inverse model, and I_y is the index set containing the indices of outputs of forward model that are moved to the input of inverse model. The entire training data should be divided based on the derivative criteria such that training samples satisfying

$$\left.\frac{\partial y_i}{\partial x_j}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(k)}} < \delta, \tag{4}$$

belong to one group and training samples satisfying

$$\left.\frac{\partial y_i}{\partial x_j}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(k)}} > -\delta, \tag{5}$$

belong to a different group, where δ is zero or a small positive number. This method exploits derivative information to divide the training data into groups. We compute the derivatives by exploiting adjoint neural network technique [15]. Multiple neural networks are then trained with the divided data. Each neural network represents a sub-model of the overall inverse model.

D. Method to Combine Inverse Sub-Models

We need to combine the multiple inverse sub-models to reproduce the overall inverse model. For this purpose a mechanism is needed to select the right one among multiple inverse sub-models for a given input $\bar{\boldsymbol{x}}$. For convenience of explanation, suppose $\bar{\boldsymbol{x}}$ is a randomly selected sample of training data. Ideally if $\bar{\boldsymbol{x}}$ belongs to a particular inverse sub-model then the output from it should be the most accurate one among various inverse sub-models. Conversely the outputs from the other inverse sub-models should be less accurate if $\bar{\boldsymbol{x}}$ does not belong to them. However, when using the inverse sub-models with general input $\bar{\boldsymbol{x}}$ whose values are not necessarily equal to that of any training samples, the value from the sub-models is the unknown parameter to be solved. So we still do not know which inverse sub-model is the most accurate one. To address this dilemma, we use the forward model to help deciding which inverse sub-model should be selected. If we supply an output from the correct inverse sub-model to an accurate forward model we should be able to obtain the original data input to the inverse sub-model.

In our method input $\bar{\boldsymbol{x}}$ is supplied to each inverse sub-model and output from them is fed to the accurately trained forward model respectively, which generate different $\boldsymbol{y}$. These outputs are then compared with the input data $\bar{\boldsymbol{x}}$. The inverse sub-model that produces least error between $\boldsymbol{y}$ and $\bar{\boldsymbol{x}}$ is selected and the output from corresponding inverse sub-model is chosen as the final output of the overall inverse modeling problem.

We include another constraint to the inverse sub-model selection criteria. This constraint checks for the training range. If an inverse sub-model produces an output that is located outside its training range, then the corresponding output is

not selected. If the outputs of other inverse sub-models are also found outside their training range then we compare their magnitude of distances from the boundary of training range. An inverse sub-model producing the lowest distance is selected in this case.

III. ANN-BASED DYNAMIC BEHAVIORAL MODELING OF MICROWAVE CIRCUITS

A. Recurrent Neural Networks (RNN) for Time Domain Modeling

Conventional feed-forward neural networks (FFNN) [1] are well known for their learning and generalization capabilities. However, they are only suitable for mapping static input-output relationships. To model nonlinear circuit responses in time-domain, a neural network that can include temporal information is necessary. RNNs have been found to be a suitable candidate to accomplish this task. In the past, RNNs were successfully used in various engineering applications such as system control, speech recognition, etc [16]. For microwave dynamic modeling, the structure of a typical RNN is shown

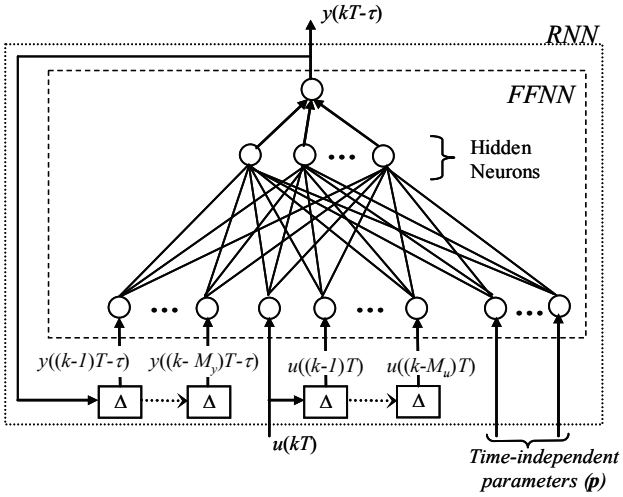


Fig. 2. RNN structure with output feedback (M_y). The RNN is a discrete time structure trained with sampled input-output data [12].

in Fig. 2 [12]. The RNN inputs include time-varying inputs $\boldsymbol{u}$ and time-independent inputs $\boldsymbol{p}$. The RNN outputs are the time varying signal $\boldsymbol{y}$. The input layer of the FFNN contains buffered (time-delayed) history of $\boldsymbol{y}$ fed back from the output layer, buffered history of $\boldsymbol{u}$, and $\boldsymbol{p}$. The hidden layer contains neurons with sigmoid activation functions. The FFNN outputs are linear functions of the responses of hidden neurons in the hidden layer. Let the trainable parameters of the RNN be denoted as $\boldsymbol{w}$. As can be observed from Fig. 2, the overall RNN structure, including both the FFNN part and feedback connections, realizes the following nonlinear dynamic relationship [12]

$$\begin{aligned}&\boldsymbol{y}(kT-\tau)\\&=\mathrm{f}_{\mathrm{FFNN}}(\boldsymbol{y}((k-1)T-\tau),...,\boldsymbol{y}((k-M_y)T-\tau),\\&\quad\boldsymbol{u}(kT),\boldsymbol{u}((k-1)T),...,\boldsymbol{u}((k-M_u)T),\boldsymbol{w},\boldsymbol{p})\end{aligned}\quad(6)$$

where k is the index for time step, T is the time step size, and τ is a delay element. The number of delayed time steps M_y and M_u, of $\boldsymbol{y}$ and $\boldsymbol{u}$ respectively, represent the effective order of original nonlinear circuit as seen from input-output data. $\mathrm{f}_{\mathrm{FFNN}}$ represents a static mapping function that could be any of the standard FFNN structures, e.g., a multilayer perceptron (MLP) neural network [1]. The formulation (6) is a generalization over the conventional RNN structure [13] by introducing the extra delay τ, which represents the delay between the input and output signals [12]. It has been found that this modified structure could help simplify the overall training of the model.

B. Dynamic Neural Networks (DNN) for Nonlinear Behavioral Modeling

For the purpose of circuit simulation, the most ideal format to describe nonlinear dynamics is the continuous-time domain formulation. In theory, this format best describes the fundamental essence of nonlinear behavior and in practice it is flexible to fit most or all the needs for nonlinear circuit simulation. On the other hand, for large-scale nonlinear microwave circuits, the detailed state equations [14] could be too complicated, computationally expensive, and sometimes even unavailable at system level. Therefore, a simplified (reduced order) model approximating the same dynamic input-output relationships is highly required. With these motivations, DNN technique [14] was presented for large-signal modeling of nonlinear microwave circuits and systems. Let N_d be the order of DNN representing the reduced order for original nonlinear circuit. Let

$\boldsymbol{v}_i$ be a N_y-vector, $i = 1,2,\ldots,N_d$. Let g_{ANN} represent an MLP neural network [1], where input neurons contain $\boldsymbol{y}$, $\boldsymbol{u}$, their derivatives $d^i\boldsymbol{y}/dt^i$, $i=1, 2, \ldots, N_d-1$, and $d^k\boldsymbol{u}/dt^k$, $k=1,2, \ldots, N_d$, and the output neuron represents $\mathrm{d}^{N_d}\boldsymbol{y}/\mathrm{dt}^{N_d}$. In [14], the DNN model was formulated as

$$\begin{aligned}
\dot{\boldsymbol{v}}_1(t) &= \boldsymbol{v}_2(t) \\
&\vdots \\
\dot{\boldsymbol{v}}_{N_d\text{-}1}(t) &= \boldsymbol{v}_{N_d}(t) \\
\dot{\boldsymbol{v}}_{N_d}(t) &= \mathrm{g}_{\mathrm{ANN}}(\boldsymbol{v}_{N_d}(t), \boldsymbol{v}_{N_d-1}(t), \cdots, \\
&\qquad \boldsymbol{v}_1(t), \boldsymbol{u}^{(N_d)}(t), \boldsymbol{u}^{(N_d-1)}(t), \cdots, \boldsymbol{u}(t))
\end{aligned} \tag{7}$$

where inputs and outputs of the DNN model are $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t) = \boldsymbol{v}_1(t)$, respectively. The overall DNN model is in a standardized format for typical nonlinear circuit simulators. For example, the left-hand-side of the equation provides the charge or the capacitor part, and the right-hand-side provides the current part. This format is the standard representation of nonlinear components in many harmonic balance (HB) simulators. In this way, DNN can provide dynamic current-charge parameters for general nonlinear circuits with any number of internal nodes in original microwave circuit [14]. The DNN technique has been successfully used in modeling of nonlinear microwave circuits such as mixers and amplifiers. The trained DNN behavioral models were also used to facilitate fast high-level HB simulations of circuits and systems. As a recent advance, the work in [17] introduced a mathematical way to determine the order of the DNN formulation from the training data. A further enhancement is made in [18] where a modified HB formulation incorporating constraint functions was presented to improve the robustness and efficiency of DNN-based HB simulation of high-level nonlinear microwave circuits.

IV. EXAMPLES

A. Development of Inverse Models for Waveguide Filter

Inverse models of a dual mode waveguide filter are developed. According to [8], the filter is decomposed into three different modules each representing a separate filter junction. Neural network inverse models of these junctions were developed separately using the proposed methodology.

The first neural network inverse model of the filter structure is developed for the internal coupling iris. The inverse model is formulated as

$$\bar{\boldsymbol{y}} = [x_3 \; x_4 \; y_3 \; y_4]^T = [L_v \; L_h \; P_v \; P_h]^T \tag{8}$$

$$\bar{\boldsymbol{x}} = [x_1 \; x_2 \; y_1 \; y_2]^T = [D \; fo \; M_{23} \; M_{14}]^T. \tag{9}$$

where D is the circular cavity diameter, fo is the center frequency, M_{23} and M_{14} are coupling values, L_v and L_h are the vertical and horizontal coupling slot lengths and P_v and P_h are the loading effect of the coupling iris on the two orthogonal modes, respectively. After formulating the inverse model training data were generated and the entire data was used to train the inverse model. Direct training produced good accuracy in terms of least square (L2) errors. However the worst-case error was large. Therefore in the next step the data was segmented into four sections. Models for these sections were trained separately, which reduced the worst-case error. The final model accuracy of the two methods is shown in Table 1. We can improve the accuracy further by splitting the data set into more sections and achieve as accurate result as required.

The second inverse model of the filter is the IO iris model. The input parameters of IO iris inverse model are circular cavity diameter D, center frequency fo, and the coupling value R. The output parameters of the model are the iris length L, the loading effect of the coupling iris on the two orthogonal modes P_v and P_h, and the phase loading on the input rectangular waveguide P_{in}. The inverse model is defined as

$$\bar{\boldsymbol{y}} = [x_3 \; y_2 \; y_3 \; y_4]^T = [L \; P_v \; P_h \; P_{in}]^T \tag{10}$$

$$\bar{\boldsymbol{x}} = [x_1 \; x_2 \; y_1]^T = [D \; fo \; R]^T. \tag{11}$$

Four different sets of training data were generated according to the width of iris using mode-matching method. Each set was trained and tested separately using the direct inverse modeling method. The result of these direct inverse modeling is listed in Table 1. In the conventional direct method, L2 errors are acceptable but the

worst-case error is high. To reduce the worst-case error we split the first set of data into several segments and trained separately. These models produced acceptable accuracy. Same method was applied to the rest of the three models. The result is presented in Table 1, which shows that the proposed methodology produce more accurate result than the direct modeling method.

The last neural network inverse model of the filter is developed for tuning screw model. The input parameters of this model are circular cavity diameter D, center frequency fo, the coupling between the two orthogonal modes in one cavity M_{12}, and the difference between the phase shift of the vertical mode and that of the horizontal mode across the tuning screw P. The model outputs are the phase shift of the horizontal mode across the tuning screw P_h, coupling screw length L_c, and the horizontal tuning screw length L_h. The inverse model is formulated as

$$\bar{\boldsymbol{y}} = [y_3 \;\; x_3 \;\; x_4]^T = [P_h \;\; L_h \;\; L_C]^T \quad (12)$$

$$\bar{\boldsymbol{x}} = [x_1 \;\; x_2 \;\; y_1 \;\; y_2]^T = [D \;\; fo \;\; M_{12} \;\; P]^T . \quad (13)$$

All four techniques in Section II were applied to develop this model. The final model result is presented in Table 1, which shows that the accuracy of the tuning screw model is improved drastically using the proposed method. Minor improvement is realized for the coupling iris model in terms of L2 error (the improvement is mostly realized in terms of worst-case error), because the input-output relationship of this model is relatively simpler than that of the tuning screw model. The proposed method becomes more efficient and effective for complex devices.

Three-layer multilayer perceptron neural network structure was used for each neural network model and quasi-Newton training algorithm was used to train the neural network models. Testing data were used after training the model to verify the generalization ability of these models. Automatic model generation algorithm of NeuroModelerPlus [19] was used to develop these models, which automatically train the model until model training, and testing accuracy was satisfied. The training error and test errors were generally similar because sufficient training data was used in the examples.

Table 1: Comparison of error between conventional and proposed method for waveguide filter model [8].

Waveguide junctions	Inverse modeling methods	Model error (%)	
		L2	Worst case
Coupling iris	Conventional	0.46	14.2
	Proposed	0.32	7.20
IO iris	Conventional	1.30	54.0
	Proposed	0.45	18.4
Tuning screw	Conventional	7.51	94.25
	Proposed	0.59	8.10

B. Dual Mode 6-pole Waveguide Filter Design Using the Developed Neural Network Inverse Models

In this example we design a 6-pole waveguide filer using the proposed methodology [8]. The filter center frequency is 12.155 GHz, bandwidth is 64 MHz and cavity diameter is chosen to be 1.072". In addition to the three inverse models that were developed in Example A, we developed another inverse model for slot iris. The inputs of the slot iris model are cavity diameter D, center frequency fo and coupling M and the outputs are iris length L, vertical phase P_v and horizontal phase P_h. The normalized ideal coupling values are

$$R_1 = R_2 = 1.077$$

$$\boldsymbol{M} = \begin{bmatrix} 0 & 0.855 & 0 & -0.16 & 0 & 0 \\ 0.855 & 0 & 0.719 & 0 & 0 & 0 \\ 0 & 0.719 & 0 & 0.558 & 0 & 0 \\ -0.16 & 0 & 0.558 & 0 & 0.614 & 0 \\ 0 & 0 & 0 & 0.614 & 0 & 0.87 \\ 0 & 0 & 0 & 0 & 0.87 & 0 \end{bmatrix} . \quad (14)$$

Irises and tuning screw dimensions are calculated by the trained neural network inverse models developed in Example A. The filter is manufactured and tuned by adjusting irises and tuning screws to match the ideal response and the dimensions are listed in Table 2. Very good correlation can be seen between the initial dimensions provided by the neural network inverse models and the measured final dimensions of the fine tuned filter. Figure 3 presents the

response of the tuned filter and compares with the ideal one showing a perfect match between each other.

Table 2: Comparison of dimensions obtained from EM model, neural network inverse models and measurement of the tuned 6-pole filter [8].

Filter Dimensions	EM Model (inch)	Neural Model (inch)	Measurement (inch)
IO irises	0.352	0.351	0.358
M23 iris	0.273	0.274	0.277
M14 iris	0.167	0.170	0.187
M45 iris	0.261	0.261	0.262
Cavity 1 length	1.690	1.691	1.690
Tuning screw	0.079	0.076	0.085
Coupling screw	0.097	0.097	0.104
Cavity 2 length	1.709	1.709	1.706
Tuning screw	0.055	0.045	0.109
Coupling screw	0.083	0.082	0.085
Cavity 3 length	1.692	1.692	1.692
Tuning screw	0.067	0.076	0.078
Coupling screw	0.098	0.097	0.120

The advantage of using the trained neural network inverse models is also realized in terms of CPU time compared to EM models. An EM simulator can be used for synthesis, which requires typically 10 to 15 iterations to generate inverse model dimensions. The time to obtain the dimensions using EM method is approximately 6.25 minutes compared to 1.5 milliseconds for the neural network inverse method.

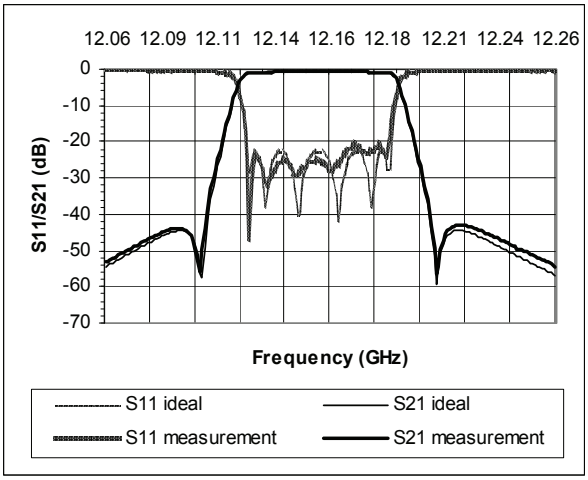
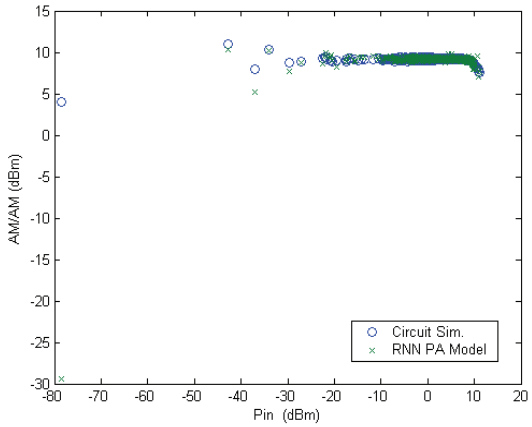


Fig. 3. Comparison of the 6-pole filter response with ideal filter response. The filter was designed, fabricated, tuned and then measured to obtain the dimensions [8].

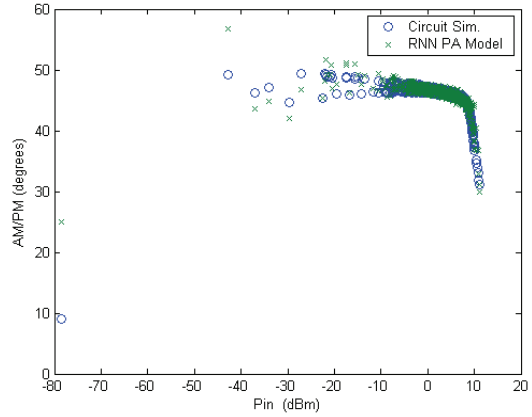
C. Development of Behavioral Models for a Power Amplifier Using RNN Technique

Here we present an example of ANN-based behavioral modeling where the RNN is used for modeling AM/AM and AM/PM distortions of a power amplifier [12]. The circuit to be modeled is an RFIC power amplifier in Agilent ADS [20], which is represented by a detailed transistor-level description. The training data are generated using a 3G WCDMA input signal with average power (P_{av}) of 1 dBm and center frequency of 980 MHz. The channel bandwidth (chip rate) is 3.84 MHz. Two RNN models, namely the In-phase RNN (K_1) and the Quadrature-phase RNN (K_2), are individually developed using 1025 input-output samples representing 256 symbols from ADS. The RNN models are trained in a step-wise manner using the automatic model generation algorithm [12], where the size (number of hidden neurons) and the order of the RNNs are automatically adjusted depending on the training status. After the RNN models achieve a good learning, for testing signals not considered in training, the AM/AM and AM/PM distortions can be faithfully modeled by the RNNs, as shown in Fig. 4(a) and Fig. 4(b), respectively. An additional benefit of using RNN behavioral models is the improved speed over conventional circuit simulators. For the RFIC amplifier example, ADS takes

approximately 100 seconds to run the entire envelope simulation for the 3G WCDMA input whereas each RNN only takes 0.16 seconds to reproduce accurately the output for the same 3G WCDMA input.



(a)



(b)

Fig. 4. (a) AM/AM distortion between ADS and RNN power amplifier (PA) behavioral model. (b) AM/PM distortion between ADS and RNN PA behavioral model [12].

D. Development of Behavioral Models for a Mixer Using DNN Technique

This example, based on [14], illustrates DNN modeling of a mixer. The circuit internally is a Gilbert cell with 14 NPN transistors in ADS [20]. The dynamic input and output of the model was defined in hybrid form as $\boldsymbol{u} = [v_{RF}, v_{LO}, i_{IF}]^T$ and $\boldsymbol{y} = [i_{RF}, v_{IF}]^T$, where v_{RF}, v_{LO}, and v_{IF} are the voltage values of radio frequency, local oscillator (LO), and inter-mediate frequency, i_{IF} and i_{RF} are the current values of intermediate frequency and radio frequency, respectively. The DNN model includes,

$$i_{RF}^{(n)}(t) = \boldsymbol{f}_{ANN1}[i_{RF}^{(n-1)}(t), i_{RF}^{(n-2)}(t), \cdots, i_{RF}(t), v_{RF}^{(n)}(t), v_{RF}^{(n-1)}(t), \cdots, v_{RF}(t)] \tag{15}$$

and

$$v_{IF}^{(n)}(t) = \boldsymbol{f}_{ANN2}[v_{IF}^{(n-1)}(t), v_{IF}^{(n-2)}(t), \cdots, v_{IF}(t), v_{RF}^{(n)}(t), v_{RF}^{(n-1)}(t), \cdots, v_{RF}(t), v_{LO}^{(n)}(t), v_{LO}^{(n-1)}(t), \cdots, v_{LO}(t), i_{IF}^{(n)}(t), i_{IF}^{(n-1)}(t), \cdots, i_{IF}(t)] \tag{16}$$

where n is the order of the DNN.

The training data were generated by varying the RF input frequency and power level from 11.7 GHz to 12.1 GHz with a step-size of 0.05 GHz and from -45 dBm to -35 dBm with a step-size of 2 dBm respectively. Local oscillator signal was fixed at 10.75 GHz and 10 dBm. Load was perturbed by 10% at every harmonic in order to allow the model learn the loading effects. The DNN was trained with different number of hidden neurons as shown in Table 3. Testing was done in ADS [20] using input frequencies from 11.725 GHz to 12.075 GHz with a step-size of 0.05 GHz and power levels at -44 dBm, -42 dBm, -40 dBm, -38 dBm, -36dBm. The agreement between model and ADS was achieved in time and frequency domains even though those test information was never seen in training. Figure 5 illustrates a comparison between the output of the DNN model and the ADS solution in time-domain.

Table 3: DNN accuracy from different training for the mixer example [14].

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

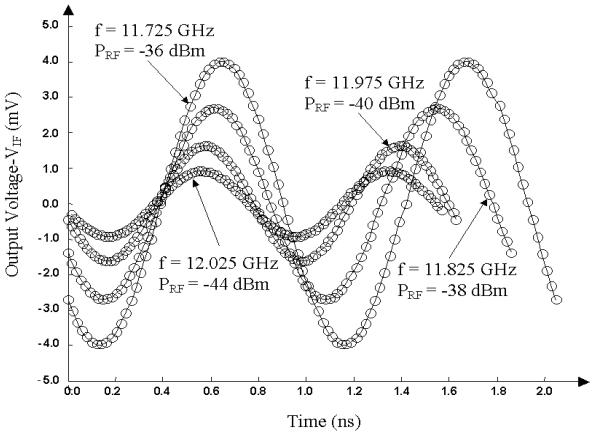


Fig. 5 Mixer v_{IF} output: time-domain comparison between DNN (—) and ADS solution of original circuit (o). Good agreement is achieved even though such data were never used in training [14].

V. CONCLUSION

We have reviewed recent advances of neural network modeling techniques for fast modeling and design of RF/microwave circuits. Inverse modeling technique is formulated and non-uniqueness of input-output relationship has been addressed. A method to identify and divide contradictory data has been proposed. Inverse models are divided based on derivatives of forward model and then trained separately to produce more accurate inverse sub-models. A method to correctly combine the inverse sub-models is presented. The proposed methodology has been applied to waveguide filter modeling and design. Very good correlation was found between neural networks predicted dimensions and that of a perfectly tuned filter following the EM method. We have also reviewed recurrent neural network and dynamic neural network modeling techniques. These time-domain neural networks are trained to learn the nonlinear dynamic input-output relationship based on the external circuit signals. The resulting neural network models are fast and accurate to predict the behavior of RF and microwave circuits.

ACKNOWLEDGMENT

Authors would like to thank Dr. Ming Yu of COMDEV for his support and collaboration in the inverse modeling project. Also thanks to Dr. Ying Wang for her technical expertise on this work.

REFERENCES

[1] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microwave Theory and Tech.,* vol. 51, pp. 1339–1350, Apr. 2003.

[2] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design,* Artech House, Boston, 2000.

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

[4] B. Davis, C. White, M. A. Reece, M. E. Jr. Bayne, W. L. Thompson, II, N. L. Richardson, and L. Jr. Walker, "Dynamically configurable pHEMT model using neural networks for CAD," *IEEE MTT-S Int. Microwave Symp. Dig.,* vol. 1, pp. 177–180, June 2003.

[5] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microwave Theory and Tech.,* vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[6] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "Applications of artificial neural network techniques in microwave filter modeling, optimization and design," *Progress In Electromagnetic Research Symposium,* Beijing, China, pp. 1972–1976, March 2007.

[7] P. Burrascano, M. Dionigi, C. Fancelli, and M. Mongiardo, "A neural network model for CAD and optimization of microwave filters," *IEEE MTT-S Int. Microwave Symp. Dig.,* Baltimore, MD, vol. 1, pp. 13–16, June 1998.

[8] H. Kabir, Y. Wang, M. Yu and Q. J. Zhang, "Neural network inverse modeling and applications to microwave filter design" *IEEE Trans. Microwave Theory and Tech.,* vol. 56, no. 4, pp. 867–879, Apr. 2008.

[9] Y. Wang, M. Yu, H. Kabir, and Q. J. Zhang, "Effective design of cross-coupled filter using neural networks ad coupling matrix*," IEEE MTT-S Int. Microwave Symp. Dig.,* pp. 1431–1434, June 2006.

[10] M. M. Vai, S. Wu, B. Li, and S. Prasad, "Reverse modeling of microwave circuits with bidirectional neural network models," *IEEE Trans. Microwave Theory and Tech.,* vol. 46, pp. 1492–1494, Oct. 1998.

[11] Q. J. Zhang and Y. Cao, "Time-domain neural network approaches to EM modeling of microwave components," *Springer Proceedings in Physics: Time Domain Methods in Electrodynamics*, pp. 41–53, Oct. 2008.

[12] H. Sharma and Q. J. Zhang, "Automated time domain modeling of linear and nonlinear microwave circuits using recurrent neural networks," *Int. J. RF Microwave Comput.-Aided Eng.*, vol.18, no. 3, pp. 195–208, May 2008.

[13] 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 network," *IEEE Trans. Microwave Theory Tech.,* vol. 48, no. 12, pp. 2335–2344, Dec. 2000.

[14] J. J. Xu, M. Yagoub, R. T. Ding, and Q. J. Zhang, "Neural based dynamic modeling of nonlinear microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 2769–2780, Dec. 2002.

[15] J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, "Exact adjoint sensitivity analysis for neural-based microwave modeling and design," *IEEE Trans. Microwave Theory and Tech.*, vol. 51, pp. 226–237, Jan. 2003.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation,* 2nd ed., Prentice-Hall, pp. 751–756, 1999.

[17] J. Wood, D. E. Root, and N. B. Tufillaro, "A behavioral modeling approach to nonlinear model-order reduction for RF/microwave ICs and systems," *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 9, pp. 2274–2284, Sep. 2004.

[18] Y. Cao, L. Zhang, J. J. Xu and Q. J. Zhang, "Efficient harmonic balance simulation of nonlinear microwave Circuits with dynamic neural models," *IEEE MTT-S Int. Microwave Symp. Dig.*, , pp. 1423–1426, June 2006.

[19] Q. J. Zhang, *NeuroModelerPlus*, Department of Electronics, Carleton University, K1S 5B6, Canada.

[20] Advanced Design System (ADS) ver. 2006A, Agilent Technologies, Inc., 2006.