Knowledge-Based Neural Network Approach for Microwave Modeling and Design

by

Lei Zhang, B. Eng.,

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirement for the
degree of Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Electronics
Carleton University
Ottawa, Ontario K1S 5B6
Canada

©Copyright January 2003, Lei Zhang

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisisitons et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-83487-5

Our file Notre référence

ISBN: 0-612-83487-5

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

The undersigned recommend
the Faculty of Graduate Studies and Research
acceptance of the thesis

Knowledge-Based Neural Network Approach
for Microwave Modeling and Design

Submitted by

Lei Zhang, B. Eng.,
in partial fulfillment of the requirements for
the degree of Master of Applied Science

Q.J. Zhang, Thesis Supervisor

Michel Nakhlá, Chair, Department of Electronics

Abstract

This thesis presents the use of knowledge-based neural networks for microwave circuit modeling and design. A general method combining microwave empirical/equivalent model with artificial neural network is proposed. This method, called generalized knowledge-based neural network (GKBNN), unifies several existing methods and provides increased model accuracy and extrapolation capability, even if the training data is limited. The method also provides a systematic approach to efficiently handle a wider variety of modeling cases than the several existing knowledge based methods combined. It is applied to microwave device and transmission line modeling for high frequency/high speed circuit design.

The topic of knowledge based neural modeling for nonlinear microwave devices is pioneered for the first time in this thesis. Two methods, dynamic neural modeling utilizing difference method and neuro-space mapping applying space mapping concept, are proposed here. Instead of using conventional trial and error approach of adjusting equivalent circuit topology and equations, we use our proposed formulations allowing neural networks to learn the gap between existing device models and the device data. This work provides a new and efficient approach in automating the device modeling process to match device data.

Acknowledgements

First of all, I would like to express my sincere thanks to my thesis supervisor Professor Qi-Jun Zhang for his professional guidance, continued assistance, invaluable inspiration, motivation, suggestion, and patience throughout the research work and the preparation of this thesis. I am highly indebted to him for having trained me into a full-time researcher with technical, computation and presentation skills, and professionalism. And I also want to say thanks to my former supervisor, Professor Runtao Ding for giving me this chance studying here that made my life different. Professor Mustapha C.E. Yagoub from University of Ottawa is also thanked for his prompt answer of my questions and suggestion for practical applications.

My deep appreciation is given to my colleagues Xiaolei Ding and Jianjun Xu for their enthusiasm, promotional skills and helpful discussions. Yi Cao and other students in my group are thanked for the help, friendship, and enjoyable collective working.

Many thanks to Betty Zahalan, Peggy Piccolo, Lorena Duncan, Nagui Mikhail, Jacques Lemieux, Scott Bruce and all other staff and faculty for providing the excellent lab facilities and friendly environment for study and research.

And, as always, a heartfelt thank is given to my closest friend and inspiration, Zhen Xu, for her patience and sweet smiles always being there for me.

Finally, I want to dedicate this work to my parents for their endless love, support and encouragement for me to live my dream all the time.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Objective	4
1.3	Outline of the Thesis	6
2	Overview of Neural Based Microwave Modeling	8
2.1	Neural Network Application in Microwave/RF Circuit Design ..	8
2.2	Review of Neural Based Microwave Modeling Technique	10
2.3	Neural Network Based Active Device Modeling	12
2.4	Knowledge Based Neural Networks	13
2.4.1	Difference Method (DM)	14
2.4.2	Space Mapping (SM)	15
2.4.3	Prior Knowledge Input (PKI)	17
2.4.4	KBNN	18
2.5	Adjoint Neural Network.....	20
2.6	Dynamic Neural Network (DNN).....	21
2.7	Conclusions	23
3	Generalized Knowledge-Based Neural Network	24
3.1	Introduction	24
3.2	Problem Formulation	25
3.3	Proposed Generalized Knowledge-Based Neural Network (GKBNN) Method	26

3.4	Application Examples of GKBNN Approach	30
3.4.1	Method Comparison	30
3.4.2	FET Modeling by GKBNN	39
3.4.3	GKBNN Modeling of Transmission Lines	43
3.5	Conclusions	46
4	Knowledge Based Nonlinear Dynamic Modeling	47
4.1	Introduction.....	47
4.2	Difference Method (DM) for Nonlinear Modeling	48
4.3	Examples of Nonlinear GaAs MESFET Device Utilizing Difference Method	52
4.3.1	Example of Difference Method Based Current-Charge Mode.....	52
4.3.2	Example of Difference Method Based DNN Model	59
4.4	Conclusions	63
5	Neuro-Space Mapping (Neuro-SM) for Nonlinear Modeling	64
4.4	Introduction.....	64
4.5	Proposed Neuro-SM for Nonlinear Device Modeling	66
4.6	Examples	71
4.6.1	Neuro-SM Nonlinear Model Used for DC Case	71
4.6.2	Neuro-SM Nonlinear Model for GaAs MESFET Device	73
4.6.3	Use of Neuro-SM GaAs MESFET Model in Amplifier Design	77
5.4	Conclusions	82
6	Conclusions and Future Work	83
7	Bibliography	85

List of Figures

Figure 2-1: Illustration of the feedforward multilayer perceptrons (MLP) structure.	
Typically, the neural network consists of one input layer, one or more hidden layers, and one output layer	10
Figure 2-2: Structure of hybrid EM-ANN model utilizing difference method	15
Figure 2-3: Structure of space mapping model.....	16
Figure 2-4: Structure of PKI model	17
Figure 2-5: Illustration of the structure of knowledge-based neural networks (KBNN).	
The KBNN model includes six layers typically	19
Figure 3-1: Proposed GKBNN structure	29
Figure 3-2: Extrapolation capability comparison between MLP and different knowledge-based neural network structures for all the four cases	34
Figure 3-3: Test of extrapolation ability between different model structures for CASE 1	35
Figure 3-4: Test of extrapolation ability between different model structures for CASE 2	36
Figure 3-5: Test of extrapolation ability between different model structures for CASE 3	37
Figure 3-6: Test of extrapolation ability between different model structures for CASE 4	38
Figure 3-7: FET model structure applying GKBNN technique.....	40

Figure 3-8: S-parameter comparison between original data and GKBNN model for the FET modeling example. Solid lines (—) represent S-parameter of the original data. Dashed lines with circles (-o-) represent S-parameter of the model.	42
Figure 3-9: Transmission lines representing high-speed VLSI interconnects	43
Figure 3-10: Coupling inductance GKBNN model	44
Figure 4-1: Knowledge-based structure for nonlinear modeling utilizing DM concept.	48
Figure 4-2: Terminal current-charge model by neural network	50
Figure 4-3: Circuit representation of the DNN model	50
Figure 4-4: Current-charge neural model based on DM concept for nonlinear GaAs MESFET modeling	55
Figure 4-5: S-parameter comparison of nonlinear GaAs MESFET models before adding difference part of DIFF I. Solid lines (—) represent S-parameters of the original data. Dashed lines with circles (-o-) represent the S-parameters of the model.	56
Figure 4-6: S-parameter comparison of nonlinear GaAs MESFET models after adding difference part of DIFF I. Solid lines (—) represent S-parameters of the original data. Dashed lines with circles (-o-) represent S-parameters of the model.	57
Figure 4-7: S-parameter comparison of nonlinear GaAs MESFET models after adding difference parts DIFF I and II. Solid lines (—) represent S-parameters of the	

original data. Dashed lines with circles (-o-) represent the S-parameters of the model.	58
Figure 4-8: DNN model based on DM concept for nonlinear GaAs MESFET modeling	60
Figure 4-9: Y-parameter comparison of nonlinear GaAs MESFET models before adding DNN models. Solid lines (—) represent Y-parameters of the original data. Dashed lines with circles (-o-) represent Y-parameters of the model. The 1 st , 2 nd , 3 rd and 4 th rows are the real and imaginary parts of Y_{11} , Y_{12} , Y_{21} and Y_{22} , respectively.....	61
Figure 4-10: Y-parameter comparison of nonlinear GaAs MESFET models after adding DNN models. Solid lines (—) represent Y-parameters of the original data. Dashed lines with circles (-o-) represent Y-parameters of the model. The 1 st , 2 nd , 3 rd and 4 th rows are the real and imaginary parts of Y_{11} , Y_{12} , Y_{21} and Y_{22} , respectively.....	62
Figure 5-1: General 2-port Neuro-SM nonlinear model.....	67
Figure 5-2: (a) Original neural network $\mathbf{v}_c = \mathbf{f}_{ANN}(\mathbf{v}_f, \mathbf{w})$ and (b) adjoint neural network in Neuro-SM. (a) is used for DC and large-signal mapping and (b) is used for small-signal mapping. Vector $\mathbf{w}$ contains all weights $w^{(l)}_{ik}$ and $w^{(2)}_{kj}$, where i, k, j are indices of neurons in the input, hidden and output layers in (a). The adjoint neural network structure corresponds to a flip of the original neural network between inputs and outputs. The output of adjoint neural network is $\partial v_{ci} / \partial v_f$ where $i=1$ or 2 if the adjoint inputs are $[1, 0]$ or $[0, 1]$, respectively.	70

Figure 5-3: I-V comparison of SiGe HBT device with coarse model and Neuro-SM model	72
Figure 5-4: I-V comparison of physical MOSFET device with coarse model and Neuro-SM model	72
Figure 5-5: Structure of Neuro-SM GaAs MESFET nonlinear model	74
Figure 5-6: Comparison between original ADS GaAs MESFET model, coarse model and Neuro-SM model. (a) DC comparison and (b) S-parameter comparison at 4 different biases	76
Figure 5-7: Three stage power amplifier	78
Figure 5-8: Gain and VSWR comparison of three stage power amplifier	79
Figure 5-9: Time domain response (a), and frequency domain harmonic balance response (b) of the three stage amplifier using original ADS solution, Neuro-SM model and coarse mode. Input power is 7dBm, frequency is 6.5GHz	80
Figure 5-10: Time domain response (a), and frequency domain harmonic balance response (b) of the three stage amplifier using original ADS solution, Neuro-SM model and coarse mode. Input power is 7dBm, frequency is 5GHz	81

List of Tables

Table 3-I: Test error of various methods vs. size of training data for different cases. Best method of each case is highlighted in bold.....	33
Table 3-II: Number of training data needed for 1% accuracy regarding different neural network model structures and for all 4 cases	34
Table 3-III: FET model input/output parameters	41
Table 3-IV: Test error comparison between MLP and GKBNN FET models	41
Table 3-V: Geometrical parameters of transmission lines	45
Table 3-VI: Test error comparison between MLP and GKBNN transmission line models	45
Table 4-I: Inputs/outputs of GaAs MESFET model	54
Table 5-I: Test error as well as extrapolation capability comparison between pure neural model (DNN), coarse model and neuro-SM model	75
Table 5-II: Yield comparison of original ADS model, coarse model and Neuro-SM model	79

Chapter 1

Introduction

1.1 Motivations

Artificial Neural Networks (ANN) have been recently recognized as a useful tool for RF and microwave modeling and design [1] - [3]. Neural network models can be trained from physics/EM simulation or measurement data and subsequently used during circuit analysis and design. The trained models are fast and can represent physics/EM behaviors it learnt which otherwise are computationally expensive. The learning ability of neural networks is very useful when analytical model for a new device is not available, e.g., modeling of a new transistor. Neural network can also generalize meaning that the model can respond to new data that has not been used during training. Neural network can be more accurate than polynomial regression models, handle more dimensions than look-up tables, and allow more automation in model development than conventional circuit models. ANN has been successfully used in a variety of applications such as modeling and optimization of high-speed VLSI interconnects [4], CPW circuits [5], spiral inductors [6], microwave FETs and amplifiers [7][8], CMOS and HBTs [9] [10], EM-optimization [11], global modeling [12], yield optimization [7] and circuit synthesis [13] [14].

Because of the increased use of neural networks in this area, an advanced requirement is addressed here, i.e., to improve model extrapolation capability and to obtain accurate model even if training data is limited. Such requirement cannot be met by conventional neural network structures, such as the popularly used MultiLayer Perceptrons (MLP) structure. In this thesis we utilize a concept called knowledge-based neural network [15]. The idea is to exploit the existing microwave empirical functions or equivalent circuit models together with the neural network to create an overall model. The empirical function part is computationally efficient, and can be used to simplify neural network learning. The neural network part will be trained from accurate microwave data to recover any characteristics, which may have been missed by the empirical function part. Knowledge-based models need less data to achieve the same accuracy as pure MLP. Extrapolation capability is also improved because of the embedded knowledge.

Four methods have been published recently along the direction of knowledge-based concept: Difference Method (DM) [2], Space Mapping (SM) method [11], Prior Knowledge Input (PKI) method [16], and KBNN method [15]. In the DM method, empirical function part gives an approximation of the output and neural network will be trained to learn the difference between the accurate data and the approximate model. In the SM method, the neural network module maps the original problem input space into a coarse (empirical) model input space. The coarse model then produces the overall output with improved accuracy. In the PKI method, the neural network learns the relationship between the outputs of the prior knowledge model and the original problem. In KBNN model, empirical function is used as neuron activation functions. Multiple empirical functions can be used and activated differently in different regions of the model input

space. The neural network part defines the different regions and regulates activation of different knowledge neurons accordingly.

These four existing methods are milestones in the area of neural based circuit modeling. However, the question of which method is better, and when to use which method has never been discussed in the literature. From circuit designer's point of view, either they have to try all the techniques or they may end up choosing a wrong one. To find a systematic and generalized solution to such a question is one of the motivations of this thesis.

The existing knowledge-based methods have been mainly formulated to address linear or passive component modeling [1]. In the case of nonlinear device modeling, the topic of knowledge-based approach remains unsolved. The most popular existing approach for device modeling is the equivalent circuit modeling approach such as [17] - [22]. Compared to detailed physical/EM models [23] - [25], equivalent circuit models are much faster but are less flexible, and their uses are often limited to the type of devices for which the model was developed. In order to develop a model, one has to firstly create a model structure and then do parameter extraction to determine parameters in the model. While parameter extraction can be done by computer-based optimization, model structure creation is often human based. The conventional process is to use human experience and skill to create an equivalent circuit topology and to create a nonlinear function for each of the nonlinear branches in the equivalent circuit. This process is trial and error based and is very human intensive. With the rapid development in microwave technology, new devices constantly evolve, and new models are constantly required. Manually modifying

the existing models to match new data is very inefficient and methods that can automatically solve the model structure problem are much desired.

Recently several methods of nonlinear modeling by neural networks have been proposed [26]. Neural models have been used to represent the DC characteristics of a physics-based MESFET [15], and small-signal behavior of an HBT [10]. It has been applied to large-signal transistor modeling [7] [27] and global modeling [12]. A recurrent neural network method using discrete time domain formulation was proposed in [28] to model nonlinear circuits and devices.

These methods represent important steps towards automating the device modeling process. However because the neural networks have to learn the device behavior from scratch without using existing device formula, more training data is needed, or the reliability of the model will be low. Since there already exists a vast body of device models, it would be more desirable to utilize the existing models and to use neural networks to compliment what is missing in the existing models. However, this important topic about nonlinear knowledge-based modeling technique has not been formally addressed in the literature, and to fill in this gap is another motivation of this thesis.

1.2 Thesis Objective

The main objective of this thesis is to develop novel knowledge-based neural network modeling techniques to enable fast and accurate model development, and at the same time, make the techniques also applicable to nonlinear device modeling and large signal circuit simulation and optimization. In this thesis, the following work regarding knowledge-based neural network modeling are presented:

- (1) The advantages and disadvantages of the existing knowledge-based techniques are examined together. A new method called Generalized Knowledge-Based Neural Network (GKBNN) [29], which unifies the existing methods and provides superior performance in various cases of microwave empirical information, is also proposed. Examples of utilizing this technique in microwave device and transmission line modeling for high frequency/high speed circuit design are demonstrated.
- (2) Knowledge-based dynamic neural modeling approach for nonlinear microwave devices is pioneered for the first time. Existing device models are combined with neural networks to form knowledge-based dynamic nonlinear device models. We propose a method to utilize the existing equivalent circuit topology as knowledge, and to train neural networks to automatically produce the nonlinear equations needed in the nonlinear branches of the equivalent circuit. We also proposed a new formulation based on a recently published dynamic neural network (DNN) [30] method to recover the difference between existing device models and training data, leading to a new method called knowledge-based dynamic neural networks.
- (3) A new approach, neuro-space mapping, is presented enabling the space mapping [11] concept to be applied to nonlinear device modeling and large signal circuit simulation. In this proposed approach, the voltage and current signals between the existing device model (namely, the coarse model) and the actual device behavior (namely, the fine model) are mapped by a neural network, such that the mapped neural model accurately matches the actual device behavior. New

training method for neuro-space mapping technique is also developed. Adjoint sensitivity technique [31] is applied in this gradient-based optimization process.

1.3 Outline of the Thesis

The thesis is organized as follows.

In chapter 2, neural network applications in microwave/RF circuit design are briefly reviewed. And a detailed review of different knowledge-based neural network structures and related techniques is also presented.

Chapter 3 introduces a new technique, Generalized Knowledge-Based Neural Network (GKBNN), which unifies all the existing knowledge-based modeling technologies. It retains all the advantages from all these existing methods and avoids their disadvantages. With this general structure, GKBNN can be applied to various microwave modeling cases and achieve faster and more accurate models. Examples are also demonstrated to validate the proposed GKBNN approach. For the first time, the comparative study of all the knowledge-based structures is done in this section. Examples for linear and nonlinear FET modeling as well as microstrip line coupling inductance modeling using the proposed GKBNN technique are also demonstrated here.

In chapter 4, knowledge-based neural models for nonlinear modeling are presented. Methods combining existing equivalent circuits with neural networks based on the concept of difference method are proposed. A knowledge-based dynamic neural network modeling approach is formulated and its training process is developed. Preliminary examples demonstrating these techniques are included.

Another novel technique for nonlinear microwave device modeling, neuro-space mapping (Neuro-SM) technique, is discussed detailedly in Chapter 5. A general structure for a nonlinear general 2-port network with input mapping neural network realized by controlled sources is proposed. A new training algorithm for DC, small-signal and large-signal simulation and optimization is developed. Examples of SiGe HBT, physics-based MOSFET and GaAs MESFET modeling and use of the models in harmonic balance simulation demonstrate that the neuro-space mapping is a systematic method to allow us to exceed the best performance from existing device models.

Finally, in chapter 6, conclusions and suggestions for future research are discussed.

Chapter 2

Overview of Neural Based Microwave modeling

As neural network techniques continue to gain recognition in microwave/RF circuit modeling and design, advanced requirement regarding model development cost, model speed and accuracy, as well as model extrapolation capability turns to be an important issue for neural network modeling. Knowledge-based neural networks are recent developments to meet such a requirement. Four existing knowledge-based neural network modeling approaches [2][11][15][16] will be discussed in this chapter. Adjoint neural network [31], which can provide sensitivity analysis for circuit simulation and optimization, and dynamic neural network [30] developed for nonlinear large-scale circuit modeling are also discussed.

2.1 Neural Network Applications in Microwave/RF Circuit Design

The drive in the microwave industry to meet the demands of high manufacturability and fast design cycles created a need for efficient statistical design techniques. Statistical analysis and yield optimization that take into account the manufacturing tolerance, model uncertainties, variation in the process parameters, etc., are widely accepted as indispensable components of the circuit design methodology [32] - [34]. Detailed

physical/ EM models of active/passive components can be an important step towards a design for the first-pass success, but the models are computationally intensive. Significant advances have been made in the exploitation of artificial neural networks as an unconventional alternative to modeling and design tasks in microwave/RF CAD. Neural network models are computationally efficient, and their ability to learn and generalize from data allows model development even when component formulas are unavailable. Compare to existing modeling techniques, neural network models are much faster than detailed physical/EM models [7] [15], more accurate than polynomial and empirical models [35], allow more dimensions than table lookup models [36], and are easier to develop when a new device/technology is introduced [37]. Once developed, these neural network models can be used in place of computationally intensive physical/EM models of active and passive components [6] [7] [15] to speed up microwave circuit design. Recent work by microwave researchers demonstrated the ability of neural networks to accurately model a variety of microwave components, such as microstrip interconnects [15], vias [2], spiral inductors [6], FET devices [9] [15], HBT devices [10], HEMT devices [38], filters [39], amplifiers [7] [28], coplanar waveguide (CPW) circuit components [5], mixers [28], antennas [40], embedded resistors [32] - [34], packaging and interconnects [4], etc. Neural networks have also been used in circuit simulation and optimization [7] [41], signal integrity analysis and optimization of VLSI interconnects [4] [42], microstrip circuit design [43], process design [44], synthesis [13] [14], and microwave impedance matching [45]. These pioneering works have established the framework of neural modeling technique in both device and circuit level of microwave applications.

2.2 Review of Neural Based Microwave Modeling Technique

A variety of neural network structures have been developed in the neural network community for microwave circuit, signal processing, control, etc. Feedforward neural network is a basic type of neural networks capable of approximating generic continuous and integrable functions. An important class of feedforward neural networks is multilayer perceptrons (MLP) [1]. Recently MLP neural models are widely used in microwave device modeling and circuit design. Typically, the MLP neural network consists of an input layer, one or more hidden layers and an output layer, as shown in Fig. 2-1.

Fig. 2-1: Illustration of the feedforward multilayer perceptrons (MLP) structure. Typically, the neural network consists of one input layer, one or more hidden layers, and one output layer.

Let $\mathbf{x}$ be an N_x -vector containing parameters of a given device or a circuit, e.g., gate length and gate width of a FET transistor; or geometrical and physical parameters of transmission lines. Let $\mathbf{y}$ be an N_y -vector containing the response of the device or the

circuit under consideration, e.g., drain current of a FET; or S-parameters of transmission lines. The relationship between $\mathbf{x}$ and $\mathbf{y}$ may be nonlinear and multidimensional. This kind of relationship is represented by

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad (2.1)$$

in original physical/EM problems.

A neural network (e.g., MLP) model can be used to represent such a relationship, by being trained through a set of $\mathbf{x}$ - $\mathbf{y}$ sample pairs, called training data, which is generated from original physical/EM simulations or measurement. Let the neural network model be represented by

$$\mathbf{y} = \mathbf{f}_{ANN}(\mathbf{x}, \mathbf{w}) \quad (2.2)$$

where $\mathbf{w}$ is a vector containing all neural network weights, which will be adjusted during neural network training process to make it best match training data. A basic description of the training objective is to determine $\mathbf{w}$ such that the difference between neural model outputs $\mathbf{y}$ and desired outputs $\mathbf{d}$ from simulation/measurement,

$$E(\mathbf{w}) = \frac{1}{2} \sum_{p=1}^{N_p} \sum_{k=1}^{N_y} (y_{pk}(\mathbf{x}_p, \mathbf{w}) - d_{pk})^2 \quad (2.3)$$

is minimized. Here d_{pk} is the k th element of vector $\mathbf{d}_p$, $y_{pk}(\mathbf{x}_p, \mathbf{w})$ is the k th output of the neural network model when the input presented to the network is $\mathbf{x}_p$, where p is the index of the training samples.

Once trained, the neural network model can be used to predict the output values given only the values of the input variables. Another stage called model test should also perform by using an independent set of input-output samples, called testing data, to test the accuracy of the neural network model. Normally, the test data should lie within the same input range as the training data but contains input-output samples which are never

seen in training stage. The ability of neural models to predict y with x value different from that of training data is called the generalization ability. A trained and tested neural model can then be used online during microwave design stage providing fast model evaluation replacing original slow physical/EM simulators.

2.3 Neural Network Based Active Device Modeling

Active device modeling is one of the most important areas of microwave CAD. Equivalent circuit models and physical models are two principal modeling techniques for modeling active devices in microwave/RF circuit design.

Compare to detailed physical models, equivalent circuit model is fast and accurate in some specific cases, so most frequently used approaches in today's circuit design are based on lumped equivalent circuits. A large variety of equivalent circuit models [17] [18] [19] have been developed in the past, because no single equivalent circuit can represent all kinds of device behaviors. The specific equivalent circuit structure in a model optimized for one type of device becomes a limitation of the model for other devices. With the rapid technology change, new types of semi-conductor devices are constantly evolving, and development of models to represent the new transistor behaviors is a continuous activity. Developing new equivalent circuit models requires human experience and judgement, and a time-consuming trial-and-error process is often needed in model development.

Recently neural networks are used for active device modeling to meet the requirement for fast and accurate model development [8] [9] [10]. Several modeling methods for modeling active devices have been published. Direct modeling approach, in which the

component external behaviors are directly modeled by neural networks, has been used in transistor modeling. It has been applied to model DC characteristics of a physics-based MESFET [15], small-signal HBT device [10] and large-signal MESFET device [7] [46]. Indirect modeling approach combines known equivalent circuit models together with neural network models to develop more efficient and flexible models and was used for modeling large-signal behaviors of an HEMT device [38] [47]. As the application of neural network modeling technique proved, trained neural models with measurement data can represent DC, small-signal and large-signal behaviors of a new device, even if the device theory/equations are still unavailable. And because neural network can learn the nonlinearity much more automatically and easily than manually formulating a nonlinear function, it is a very suitable and efficient alternative for such modeling activities.

2.4 Knowledge-based Neural Networks

A variety of neural network structures have been developed in the neural network community for microwave circuit, signal processing, control, etc., to meet various kinds of modeling requirement. But existing pure neural network (without using any approximation model) such as MLP is a kind of black-box model structurally embedding no-problem dependant information. A large amount of training data is usually needed to ensure model accuracy. However, generating large amounts of training data could be very expensive for microwave problems, e.g., physics based device simulation could be very expensive to generate many points in the model input parameter space. The conflict between the requirement of fast model development and the present status of expensive (time-consuming) data generation becomes a problem that pure neural network cannot solve.

The key to solve this problem is a concept called knowledge-based neural network [15]. The idea of knowledge-based neural network is to exploit the existing knowledge in the form of empirical function/equivalent circuit model together with the neural network model to develop faster and more accurate models. Existing microwave knowledge can provide additional information of the original problem that may not be adequately represented by the limited training data, and the neural network can help bridge the gap between empirical model and actual device model behaviors. Extrapolation capability is also enhanced because of the embedded knowledge in the model.

Recent publications address four different methods referring to this advanced technique: Difference Method (DM) [2], Space Mapping (SM) [11] method, Prior Knowledge Input (PKI) [16] method and Knowledge-Based Neural Networks (KBNN) [15] method.

2.4.1 Difference Method (DM)

Difference Method, also known as hybrid EM-ANN modeling method [2], is one of the earlier methods utilizing knowledge-based concept. Its structure is shown in Fig. 2-2. The hybrid EM-ANN model is formed by generating the difference between the existing approximate model (source model) and the EM simulation results (target model). The difference data is then used to train the neural network. This results in a smaller range of the output variables and a simpler input-output relationship. This method is expected to give good results when the difference has a simpler input-output relationship as a function of the inputs than the target data. This simpler input-output relationship requires less EM simulation points to capture important data trends. This simplification is very desirable since EM simulations consume a major portion of the time spent on developing

an EM-ANN model. The output of the approximation model together with the difference predicted by the trained neural network then becomes the overall output of the hybrid EM-ANN model.

As shown in Fig. 2-2, for each input sample x , the corresponding output $y' = f_{emp}(x)$ is computed from the approximate model, which could be empirical functions or equivalent circuit model. The difference between empirical approximation and training data Δy is represented by a neural network, say, a three layer MLP, and $\Delta y = f_{ANN}(x, w)$, w is the internal weights of the artificial neural network. The overall output of hybrid EM-ANN model is $y = y' + \Delta y$.

Fig. 2-2 : Structure of hybrid EM-ANN model utilizing difference method.

2.4.2 Space Mapping (SM)

Space mapping [11] is a novel concept for circuit design and optimization, which combines the computational efficiency of coarse models with the accuracy of fine models. The coarse models are typically empirical/equivalent circuit engineering models, which are fast but often have a limited validity range for their parameters, beyond which the simulation results may become inaccurate. On the other hand, physics/EM simulator

or measurement can provide detailed or fine models, which are accurate but CPU intensive. SM technique establishes a mathematical link between the coarse and the fine models, and directs the bulk of CPU intensive evaluations to the coarse model, while preserving the accuracy and confidence offered by the fine model.

Let the vectors $\mathbf{x}_c$ and $\mathbf{x}_f$ represent the design parameters of the coarse and fine models, respectively, and $\mathbf{R}_c(\mathbf{x}_c)$ and $\mathbf{R}_f(\mathbf{x}_f)$ be the corresponding model responses. $\mathbf{R}_c$ is much faster to calculate, but less accurate than $\mathbf{R}_f$. The aim of SM optimization is to find an approximate mapping $\mathbf{P}$ from the fine model parameter space $\mathbf{x}_f$ to the coarse model parameter space $\mathbf{x}_c$, i.e., $\mathbf{x}_c = \mathbf{P}(\mathbf{x}_f)$ such that $\mathbf{R}_c(\mathbf{P}(\mathbf{x}_f)) \approx \mathbf{R}_f(\mathbf{x}_f)$. As illustrated in Fig. 2-3, the mapping $\mathbf{P}$ is realized by a neural network $\mathbf{x}_c = f_{ANN}(\mathbf{x}_f, \mathbf{w})$. The coarse model then produces the overall output $\mathbf{y} = f_{emp}(\mathbf{x}_c)$, which should match the training data.

Fig. 2-3: Structure of space mapping model.

2.4.3 Prior Knowledge Input (PKI)

In PKI [16] method, the outputs of existing empirical model (source model) are used as inputs to the neural network model, in addition to the original problem (target model) inputs, shown in Fig. 2-4. In this case, the input-output mapping to be learned by the neural network is that between the output response of the existing approximate model and that of the target model.

For each $\mathbf{x}$ in the training data, a corresponding $\mathbf{y}' = f_{emp}(\mathbf{x})$ is computed using the empirical functions or equivalent circuit response. The neural network will then learn the mapping from target model inputs and source model outputs $\mathbf{x}' = (\mathbf{x}, \mathbf{y}')$, to the target data, resulting in a simpler input-output relationship as compared to the original problem, which requires less training data. After training, given input $\mathbf{x}$, first using empirical function to get an approximation $\mathbf{y}'$, then neural network will predict the final result $\mathbf{y} = f_{ANN}(\mathbf{x}', \mathbf{w})$.

Fig. 2-4 : Structure of PKI model.

2.4.4 KBNN

KBNN is a modeling approach combining microwave empirical experience with the power of learning of neural networks by incorporating microwave empirical or semi-analytical information into the internal structure of neural networks. The structure of knowledge-based neural networks (KBNN) [15] is illustrated in Fig. 2-5.

In KBNN neural network, the microwave knowledge is embedded as a part of the overall neural network internal section. There are six layers, which are not fully connected to each other, in the KBNN structure, namely input layer, knowledge layer, boundary layer, region layer, normalized region layer and output layer. The knowledge layer is the place where microwave knowledge resides, complementing the capability of learning and generalization of neural networks by providing additional information, which may not be adequately represented in a limited set of training data. The boundary layer can incorporate knowledge in the form of problem dependent boundary functions. The region layer contains neurons to construct regions from boundary neurons. The normalized region layer contains rational function-based neurons to normalize the outputs of region layer. The output layer contains second-order neuron combining knowledge neurons and normalized region neurons.

Compare to pure neural network structures, the prior knowledge in KBNN gives neural network more information about the original microwave problem, besides the information included in the training data. Consequently, KBNN models have better reliability when training data is limited or when the model is used beyond training range.

Fig. 2-5 : Illustration of the structure of Knowledge-based neural networks (KBNN). The KBNN model includes six layers typically.

2.5 Adjoint Neural Networks

Adjoint neural network can provide sensitivity information for circuit optimization and modeling [31]. Two neural networks work together to calculate sensitivity analysis, one called original neural network and another called adjoint neural network. The original neural network can be any kind of neural network structure such as MLP, radial-basis function network, KBNN, etc. Using second order derivative information, the neural network can be trained to learn not only device input/output relationship but also the derivative information, which is very useful in simultaneous DC/small-signal/large-signal device modeling.

Let $\mathbf{d}$ and $\mathbf{d}'$ represent the training data for the original output $\hat{\mathbf{y}}$ and its derivatives $d\hat{\mathbf{y}}/d\mathbf{x}$, respectively. Let I , K and S be the index sets of input and output neurons, and samples in training data $\mathbf{d}$, respectively. We formulate the error function for training as,

$$E = \frac{1}{2} \sum_{s \in S} \left[p_1 \sum_{k \in K} (\hat{y}_{ks} - d_{ks})^2 + p_2 \sum_{i \in I, k \in K} \left(\frac{d\hat{y}_{ks}}{dx_{is}} - (d')_{kis} \right)^2 \right] \quad (2.4)$$

where subscripts i , k and s (used for x , $\hat{\mathbf{y}}$, $\mathbf{d}$ and $\mathbf{d}'$) indicate input neuron i , output neuron k and sample s , respectively, and p_1 , p_2 are the weighting parameters. During training, both the original and the adjoint neural models share the same set of parameters $\mathbf{w}_i$, $i = 1, 2, \dots, N$. Therefore training one model will also result in the other model being updated. There are three types of training methods. (i) Train original neural model using input/output data $\mathbf{d}$, and after training, the outputs of adjoint model automatically become derivatives of original input/output. (ii) Train adjoint model only with derivative data $d\hat{\mathbf{y}}/d\mathbf{x}$. The original model will then give original input/output (i.e., $\mathbf{x}$ - $\hat{\mathbf{y}}$) relationship, which has the effect of providing integration solution over derivative training data. (iii)

Train both original and adjoint models together to learn $\mathbf{x}$ - $\hat{\mathbf{y}}$ and $d\hat{\mathbf{y}}/d\mathbf{x}$ data, which will help the neural model to be trained more accurately and robustly.

2.6 Dynamic Neural Networks (DNN)

Dynamic neural network (DNN) [30] modeling approach is a recent development regarding the issue of the increasing need for efficient modeling method in high-level and large-scale nonlinear microwave design. It is an exactly continuous time domain dynamic modeling method formulated using neural networks and can be developed directly from input-output data without having to rely on internal details of the circuits. DNN models can be trained with time or frequency domain information and then conveniently incorporated into circuit simulators for high-level and large-scale nonlinear microwave design.

Assume the original nonlinear circuit can be generally described in state equation form as,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \boldsymbol{\varphi}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= \boldsymbol{\psi}(\mathbf{x}(t), \mathbf{u}(t))\end{aligned}\tag{2.5}$$

where $\mathbf{u}$ and $\mathbf{y}$ are vectors of the input and the output signals of the nonlinear circuit, respectively, and $\mathbf{x}$ is a vector containing state variables, $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ represent nonlinear functions. In a modified nodal formulation [48], the state vector $\mathbf{x}(t)$ includes nodal voltages, currents of inductors, currents of voltage sources and charge of nonlinear capacitors.

For a circuit with many components, (2.5) 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. (2.5) is reformulated into reduced order differential equations using the input-output variables as,

$$\mathbf{y}^{(n)}(t) = \mathbf{f}(\mathbf{y}^{(n-1)}(t), \mathbf{y}^{(n-2)}(t), \dots, \mathbf{y}(t), \mathbf{u}^{(n)}(t), \mathbf{u}^{(n-1)}(t), \dots, \mathbf{u}(t)) \quad (2.6)$$

where $\mathbf{f}$ represents nonlinear functions.

The overall DNN model formulated in [30] is in a standardized format for typical nonlinear circuit simulators. DNN model overcomes the limitations of the previous static I-Q neural model of [7] which was only suitable for intrinsic FETs and 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 $\mathbf{f}_{ANN}$) 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 curse of dimensionality in multi-tone simulation.

DNN model will represent a nonlinear microwave circuit only after we train it with data from the original circuit. Both time and frequency domain data can be used for DNN training by the training algorithm developed in [30]. The compatibility of DNN training with large-signal harmonic data is an important advantage over the discrete recurrent neural network approach [28] whose training is limited to time-domain only.

2.7 Conclusions

In this chapter, the existing neural network based modeling techniques for microwave/RF circuit design, which are relevant to this thesis work, have been reviewed. Neural network based models can be used to achieve a significant speedup of microwave/RF simulation and optimization, by replacing electronic and microwave component models, which are represented by detailed physics/EM equations. These neural models can be trained with the corresponding physics/EM data. However, most of the existing neural network structures are of black box type without any problem dependent information embedded, and need a large amount of training data to get an accurate model, which results in high cost model development. Knowledge-based neural network is one solution to such problem. Recently knowledge-based neural networks have been widely used in microwave device modeling area, but it's still a challenge to model devices with highly nonlinearity characteristics.

Chapter 3

Generalized Knowledge-Based Neural Network

3.1 Introduction

As reviewed in previous chapter, neural networks have been recently recognized as an important alternative for microwave modeling and design [1]. They can be trained from microwave data. The trained neural network becomes a microwave model providing fast estimation of microwave component behavior for use in circuit simulation and optimization. It has been used in modeling EM structures [11] [13], semiconductor devices [9] [10], microstrip antennas [1], VLSI interconnects [15] and more [3], leading into improved efficiency in microwave CAD.

This chapter addresses an advanced requirement in this area, i.e., to improve model extrapolation capability and to obtain accurate model even if training data is limited. Such requirement cannot be met by conventional neural network structures, such as the popularly used MultiLayer Perceptrons (MLP) structure. A concept called knowledge-based neural network [4] is utilized to satisfy this requirement. The idea is to exploit the existing microwave empirical functions or equivalent circuit models together with the neural network to create an overall model. The empirical function part is computationally

efficient, and can be used to simplify neural network learning. The neural network part will be trained from accurate microwave data to recover any characteristics, which may have been missed by the empirical function part. Knowledge-based models need less data to achieve the same accuracy as pure MLP. With the knowledge embedded in neural network structure, extrapolation capability of the overall model is also improved.

Four knowledge-based methods will be discussed in this chapter. A new technique called Generalized Knowledge-Based Neural Network (GKBNN), which unifies the existing methods and provides superior performance in various cases of microwave empirical information, will be proposed. Comparisons of all these knowledge-based techniques are to be examined together for the first time and application examples demonstrated will also further prove the advantage and availability of the new GKBNN approach in microwave device and transmission line modeling for high frequency/high speed circuit design.

3.2 Problem Formulation

Let $\mathbf{x}$ and $\mathbf{y}$ be vectors of model inputs and outputs. Let the knowledge be represented by function $\mathbf{y} = f_{emp}(\mathbf{x})$, which can be either empirical functions or circuit formulas. Let $\mathbf{w}$ represent neural network internal weights. Let neural network input-output relationship be represented by $\mathbf{y} = f_{ANN}(\mathbf{x}, \mathbf{w})$, which can be realized by an MLP. Let Tr and Te be data sets representing training and testing data. Let E_T be the testing error calculated as the relative difference between the model output $\mathbf{y}$ and that in the testing data. The problem is to determine how the two parts $f_{emp}(\mathbf{x})$ and $f_{ANN}(\mathbf{x}, \mathbf{w})$ should interact such that the

testing error E_T of the combined model will be small even if data in training set Tr is limited.

The several recent publications [2] [11] [15] [16] are preliminary steps towards this direction. In the DM [2] method, empirical function part gives an approximation of the output and neural network will be trained to learn the difference between the accurate data and the approximate model. In the SM [11] method, the neural network module maps the original problem input-space $\mathbf{x}$ into a coarse (empirical) model input-space. The coarse model then produces the overall output with improved accuracy. In the PKI [16] method, the neural network part learns the relationship between the outputs of the prior knowledge model and the original problem. In KBNN [15] model, empirical function is used as neuron activation functions. Multiple empirical functions can be used and activated differently in different regions of the model input space. The neural network part defines the different regions and regulates activation of different knowledge neurons accordingly.

3.3 Proposed Generalized Knowledge-Based Neural Network (GKBNN) Method

The proposed GKBNN [29] employs the multiple empirical features of KBNN and expands it with several new neural network mappings between inputs and outputs. Fig. 3-1 shows the GKBNN structure.

The structure is composed of six parts, the knowledge part, the boundary/region neural network part $f_{ANN}(\mathbf{x}, \mathbf{w}_I)$, the gating part, the input mapping neural network part

$f_{ANN2}(\mathbf{x}, \mathbf{w}_2)$, the output mapping neural network part $f_{ANN3}(\mathbf{x}, \mathbf{w}_3)$, and the difference neural network part $f_{ANN4}(\mathbf{x}, \mathbf{w}_4)$.

The input mapping neural network provides a trainable mapping from original problem input $\mathbf{x}$ to an intermediate input space $\bar{\mathbf{x}}$, which is fed into the knowledge functions. The input mapping is

$$\bar{\mathbf{x}} = f_{ANN2}(\mathbf{x}, \mathbf{w}_2). \quad (3.1)$$

The knowledge part contains multiple empirical functions represented by multiple neurons. Let the empirical functions be represented by $\Psi(*)$. For neuron i in the knowledge part:

$$\bar{y}_i = \Psi_i(\bar{\mathbf{x}}), \quad i = 1, 2, \dots, N_z, \quad (3.2)$$

where N_z gives the number of knowledge functions. The output mapping neural network part provides a trainable mapping from the outputs of the empirical functions to the outputs of the overall model, i.e.,

$$\hat{\mathbf{y}} = f_{ANN3}(\bar{\mathbf{y}}, \mathbf{w}_3), \quad (3.3)$$

where $\bar{\mathbf{y}}$ is a vector containing the outputs from all the knowledge neurons. The boundary/region neural network part is a sub-neural network containing boundary layer B , region layer R , and normalized region layer R' . We define them based on the KBNN formulation [15]. Let N_b , N_r and N_y represent numbers of boundary neurons, region neurons, and number of model outputs in $\mathbf{y}$, respectively. Neuron i in the boundary layer is calculated by

$$b_i = B_i(\mathbf{x}, \mathbf{v}_i), \quad i = 1, 2, \dots, N_b \quad (3.4)$$

where v_i is a vector of parameters defining an open or closed boundary in the input space x . Let $\sigma(*)$ be a sigmoid function. The region layer R contains neurons to construct regions from boundary neurons,

$$r_i = \prod_{j=1}^{N_b} \sigma(\alpha_{ij} b_j + \theta_{ij}), \quad i = 1, 2, \dots, N_r \quad (3.5)$$

where α_{ij} and θ_{ij} are the scaling and bias parameters, respectively. The normalized region layer R' contains rational function based neurons to normalize the outputs of region layer,

$$r'_i = \frac{r_i}{\sum_{j=1}^{N_r} r_j}, \quad i = 1, 2, \dots, N_r, \quad N_{r'} = N_r. \quad (3.6)$$

The gating part allows different knowledge neurons to be activated according to the normalized region layer, such that different empirical functions can be used in different regions of the model input space, i.e.,

$$\tilde{y}_j = \sum_{i=1}^{N_z} \beta_{ji} \hat{y}_i r'_i + \beta_{j0}, \quad j = 1, 2, \dots, N_y. \quad (3.7)$$

The part from boundary layer to gating network, i.e., boundary layer, region layer and normalized region layer, combined together is represented by $f_{ANN1}(x, w_1)$. The difference neural network part recovers the remaining differences between input-output mapped and region-regulated empirical solutions and the accurate training data, i.e.,

$$\Delta y = f_{ANN4}(x, w_4). \quad (3.8)$$

Now the final outputs of the overall GKBNN model is

$$y_j = \tilde{y}_j + \Delta y_j, \quad j = 1, 2, \dots, N_y. \quad (3.9)$$

The training parameters of the overall GKBNN include w_1 , w_2 , w_3 and w_4 .

Fig. 3-1: Proposed GKBN model structure.

This GKBNN model achieves all the capabilities of existing methods and can efficiently handle a larger variety of modeling situations than all existing knowledge-based neural network methods. Alternatively, the effects of the existing technique can also be achieved by special cases of the GKBNN. For example, the DM method can be achieved from GKBNN by letting input-output mapping to be unity mapping and by defining only one region to cover the entire input space. The SM method can be achieved from GKBNN by letting the output mapping to be unity, the difference neural network to be zero, and using only one region in GKBNN. The PKI method can be achieved by making the input mapping to be unity mapping, and using only one region in GKBNN. The KBNN method can be achieved using simple shift-scaling in input and output mapping, and making the difference neural network to be zero.

3.4 Application examples of GKBNN Approach

3.4.1 Method Comparisons

The purpose of this example is to compare the various methods using simple cases to provide clear contrasts between their advantages and disadvantages.

The original problem is assumed to be: $f(x) = x \sin(x) + 0.2x + 3$. (3.10)

For modeling purpose, we only have x-y data from the original problem and we should not know the original formula. Suppose we have four empirical functions:

$$\text{CASE 1: } f(x) = x \sin(x) + 0.05x + 0.5 \quad (3.11)$$

$$\text{CASE 2: } f(x) = 1.1x \sin(1.1x) + 0.22x + 3 \quad (3.12)$$

$$\text{CASE 3: } f(x) = 0.9x \sin(x) + 0.18x + 2.7 \quad (3.13)$$

$$\text{CASE 4: } f(x) = 0.55x \sin(1.1x) + 0.11x + 1.5 \quad (3.14)$$

It is very clear that compared to (3.10), only a linear difference was added to form (3.11), input scaling factor was added to form (3.12), output scaling factor was added to form (3.13), and (3.14) combines all the above three cases.

For each case, we train models using MLP, DM, SM, PKI, KBNN and GKBNN with different amount of training data (0, 2, 5, 10, 25 and 40). Training with 0 data means the model is initialized such that the overall model is equal to the empirical model. Training was done by NeuroModeler [49]. The model accuracy using 40 testing data for various cases is compared in Table 3-I.

Table 3-II gives number of training data needed to reach 1% accuracy. 40+ means more than 40 training data is needed, but the accuracy cannot be guaranteed. This table shows that knowledge-based models need far less training data than MLP. Fig. 3-2 shows extrapolation ability of different models, which are trained by 40 and tested by 80 data.

It is observed from Tables 3-I and 3-II that MLP requires large amount of training data than all other methods. DM is effective if the difference of empirical model and training data is a simple pattern (CASE 1) and ineffective otherwise. SM is effective when only input of the empirical function needs to be realigned (CASE 2). PKI is effective when only output of the empirical function needs to be realigned (CASE 3). KBNN can handle simple alignment in both input and output (CASEs 2 and 3), and not accurate enough if additional differences exist in empirical functions. GKBNN provides best performance in all cases. It achieves better accuracy and extrapolation capability with limited training data than all other techniques.

Fig. 3-3 ~ 3-6 show the extrapolation comparison between MLP and various knowledge-based methods for each cases. The model was trained only within the range of

$0 \leq x \leq 15$. Here we plot the model beyond the training range. It is observed that the MLP always show poor extrapolation. The various existing knowledge methods sometimes show good extrapolation and sometimes not, depending upon which method is used in which case. Only the proposed GKBNN shows good extrapolation in all the cases.

Table 3-I Test error of various methods vs. size of training data for different cases. Best method of each case is highlighted in bold.

	No. of Training Data	0	2	5	10	40	
CASE 1	C	MLP	680.8	99.30	52.81	2.76	0.43
	A	DM	20.12	4.45	0.02	0.02	0.02
	S	SM	19.70	25.16	19.84	14.25	17.90
	E	PKI	19.26	19.24	9.25	6.77	1.59
		KBNN	20.12	36.58	23.08	3.16	3.19
	1	GKBNN	20.12	1.35	0.93	0.15	0.15
CASE 2	C	MLP	680.8	99.30	52.81	2.76	0.43
	A	DM	13.46	18.84	7.92	8.37	4.57
	S	SM	12.93	6.63	0.52	0.05	0.03
	E	PKI	12.98	15.62	12.40	13.72	11.28
		KBNN	12.95	7.18	1.4E-2	8.8E-5	5.5E-5
	2	GKBNN	12.95	0.45	5.7E-6	3.6E-6	3.2E-6
CASE 3	C	MLP	680.8	99.30	52.81	2.76	0.43
	A	DM	3.64	2.73	2.77	2.50	0.95
	S	SM	3.37	4.76	4.30	4.04	0.03
	E	PKI	3.61	1.11	2.06	0.32	0.02
		KBNN	3.64	0.53	8.2E-5	8.2E-5	5.5E-5
	3	GKBNN	3.64	0.02	2.5E-6	2.5E-6	3.2E-6
CASE 4	C	MLP	680.8	99.30	52.81	2.76	0.43
	A	DM	28.50	44.08	23.00	23.69	11.21
	S	SM	28.48	30.36	28.50	16.46	27.62
	E	PKI	28.39	25.18	23.69	54.50	20.77
		KBNN	28.50	106.3	34.45	8.99	9.9E-5
	4	GKBNN	28.50	4.40	1.38	0.43	3.2E-6

Table 3-II Number of training data needed for 1% accuracy for example 1.

	CASE 1	CASE 2	CASE 3	CASE 4
MLP	35	35	35	35
DIFF	4	40+	35	40+
SM	40+	4	35	40+
PKI	40+	40+	7	40+
KBNN	40+	3	1	35
GKBNN	3	1	1	7

Fig. 3-2: Extrapolation capability comparison between MLP and different knowledge-based neural network structures for all the four cases.

(a) MLP

(d) PKI

(b) DM

(e) KBNN

(c) SM

(f) GKBNN

Fig. 3-3: Test of extrapolation ability between different model structures for CASE 1.

(a) MLP

(d) PKI

(b) DM

(e) KBNN

(c) SM

(f) GKBN

Fig. 3-4: Test of extrapolation ability between different model structures for CASE 2.

Fig. 3-5: Test of extrapolation ability between different model structures for CASE 3.

Fig. 3-6: Test of extrapolation ability between different model structures for CASE 4.

3.4.2 FET Modeling by GKBNN

Neural network based FET model has recently been used in global modeling combining EM analysis with semiconductor physics [50]. Here we use our GKBNN approach to provide the model used in [50]. Such neural based FET model is much faster than physics model for efficient FDTD based global modeling. Curtice equivalent circuit model [17] is used as the knowledge part. Fig. 3-7 shows the model structure using GKBNN method. Model parameters are given in Table 3-III.

Using Agilent ADS [51] we generate 4 sets of data, i.e., Tr1, Tr2, Te1, and Te2, respectively. Tr1 contains limited training data (100 samples), Tr2 contains sufficient training data (275 samples), Te1 contains testing data within training range (605 samples), and Te2 contains testing data inside and outside training range (1100 samples). Fig. 3-8 shows the S-parameters comparison between original training data and GKBNN model. Table 3-IV shows the comparison of model accuracy and extrapolation capability between MLP and GKBNN. GKBNN provides better accuracy and extrapolation capability with less training data.

Fig. 3-7: FET model structure applying GKBNN technique.

Table 3-III FET model input/output parameters.

Overall Model Input x	Vg (gate voltage, $-4\sim 0\text{V}$) Vd (drain voltage, $0\sim 10\text{V}$) f (frequency, $1\sim 20\text{GHz}$)
Intermediate Inputs $\bar{x}$	Cgs, Cgd, Cds, G_{m1} , G_{m2}
Output y	S-parameters

Table 3-IV Test error comparison between MLP and GKBNN FET models.

Training Data Set	Model Type	Model Accuracy Test	Extrapolation Test
Tr1 (100)	MLP	6.73	17.15
	GKBNN	1.34	1.89
Tr2 (275)	MLP	0.74	1.77
	GKBNN	0.39	1.10

Figure 3-8: S-parameter comparison between original data and GKBNN model for the FET modeling example. Solid lines (—) represent S-parameter of the original data. Dashed lines with circles (o) represent S-parameter of the model.

3.4.3 GKBNN Modeling of Transmission Lines

High speed VLSI interconnect design [15] requires fast and accurate model of transmission lines. An example of this application is shown in Fig. 3-9. Fig. 3-10 shows the GKBNN structure in modeling coupling inductance of coupled transmission lines versus geometrical parameters defined in Table 3-V. Such neural based model will be much faster than direct EM model during VLSI interconnect optimization.

We can see from the model structure that in this example, both input and output mappings are unit mappings. Difference neural network is used to learn the difference between empirical formula and training data to get more accurate knowledge model. The empirical knowledge used is [9]:

$$l_{12} = \frac{\mu_r \mu_0}{4\pi} \ln \left[1 + \frac{(2x_4)^2}{(x_1 + x_3)^2} \right]. \quad (3.15)$$

Fig. 3-9: Transmission lines representing high-speed VLSI interconnects.

Fig. 3-10: Coupling inductance GKBNN model.

Two sets of training data (size 100 and 500) were generated by EM simulation [15]. A set of 500 testing samples was generated in the same range as training data to test model accuracy. Another set of 4096 data selected around and beyond boundary of the model training region is used to test extrapolation ability. Table 3-VI shows the test results. MLP (i.e., neural network without knowledge) requires large amount of training data, and it cannot generalize well outside its training region. But GKBNN can achieve very good model accuracy as well as extrapolation ability even with small amount of training data.

Table 3-V Geometrical parameters of transmission lines.

Parameters	Notation	Range
Conductor Width W	x_1	0.10 – 0.25 mm
Conductor Thickness T	x_2	17 – 71 μm
Conductor Separations S	x_3	0.10 – 0.76 mm
Substrate Height H	x_4	0.10 – 0.31 mm
Relative Dielectric Constant ϵ_r	x_5	3.7 – 4.8
Frequency f	x_6	0.5 – 2 GHz

Table 3-VI Test error comparison between MLP and GKBNN transmission line models.

Training Data Size	Model Type	Model Accuracy Test	Extrapolation Test
100	MLP	1.33	3.69
	GKBNN	0.32	0.44
500	MLP	0.56	0.91
	GKBNN	0.10	0.20

3.5 Conclusions

A unified approach of knowledge-based neural network, GKBNN, has been proposed in this chapter. With the combined microwave empirical/equivalent model information and unification of existing knowledge-based methods, this kind of neural network model can provide increased model accuracy and extrapolation capability, even if the training data is limited. Examples demonstrated show that GKBNN technique can be used to achieve fast and accurate models for active/passive devices and is useful for microwave circuit design.

Chapter 4

Knowledge-Based Nonlinear Dynamic Modeling

4.1 Introduction

Nonlinear device modeling is one of the most important areas of microwave CAD. Most popularly used approaches presently are based on equivalent circuit models, such as [17] – [22]. A large variety of such equivalent circuit models have been developed in the past, because no single equivalent model can represent all kinds of device behaviors. With the rapid technology change, new devices are constantly evolving. Existing models developed for existing devices are not always adequate to represent the behaviors in new devices. Developing new equivalent circuit models requires human experience and judgement. Often a time-consuming trial-and-error process is used for formulating new equivalent circuit topology and for creating formulas for nonlinear elements. Recently, several neural network based methods have been published to address the nonlinear modeling issues [7] [27]. These methods rely on device data for neural network training, leading to model development without the use of device equivalent circuits or device equations.

In this chapter, we propose a new method combining existing device equivalent circuit models with neural network learning. The motivation is that with existing device

models used as prior knowledge, we can develop more reliable models with less data, and at the same time improve model extrapolation capability. The use of neural network will help to makeup for what is missing in the existing equivalent circuit models by automatic learning process.

4.2 Difference Method (DM) for Nonlinear Modeling

Let the 2 port functional relationship $[i_{gequ}, i_{dequ}] = f_{equ}(v_g, v_d)$ represent an existing equivalent circuit model, where v_g and v_d are the terminal voltages, and i_g and i_d represent the terminal currents of the device. In typical FET modeling, v_g and v_d represent gate and drain voltages and i_g and i_d represent gate and drain currents. The relationship between these terminal voltages and currents are dynamic nonlinear relationships. We propose a knowledge structure based on the concept of difference method. The modeling structure combining existing models and neural network is proposed as shown in Fig. 4-1.

Fig. 4-1: Knowledge-based structure for nonlinear modeling utilizing DM concept.

We propose 2 methods of constructing the neural network part. Since the neural network model should also be nonlinear dynamic model, direct use of MLP or any of the methods in previous chapters is not suitable.

The first method we propose is to use a terminal current-charge model, shown in fig. 4-2. Here i_{NNg} , i_{NNd} , q_{NNg} and q_{NNd} are neural network outputs of a static neural network with v_g and v_d as input neurons. i_{NNg} and i_{NNd} represent the difference terminal currents at the gate and drain terminals, and q_{NNg} and q_{NNd} represent the difference terminal charges at the gate and the drain, respectively.

The second method of constructing the neural network part is based on a recent dynamic neural network (DNN) [30] method. The DNN can be developed without assuming any specific current-charge format, and has the potential to represent higher order dynamics. The format of the model is:

$$\begin{aligned} \dot{\mathbf{v}}_1(t) &= \mathbf{v}_2(t) \\ &\vdots \\ \dot{\mathbf{v}}_{n-1}(t) &= \mathbf{v}_n(t) \\ \dot{\mathbf{v}}_n(t) &= \mathbf{f}_{ANN}(\mathbf{v}_n(t), \mathbf{v}_{n-1}(t), \dots, \mathbf{v}_1(t), \mathbf{u}^{(n)}(t), \mathbf{u}^{(n-1)}(t), \dots, \mathbf{u}(t)) \end{aligned} \quad (4.1)$$

where the inputs and outputs of the model is $\mathbf{u}(t)$ and $\mathbf{y}(t) = \mathbf{v}_1(t)$, respectively. And a circuit representation of the model is shown in Fig. 4-3.

Fig. 4-2: Terminal current-charge model by neural network.

Fig. 4-3: Circuit representation of the DNN model.

Compared to the FET modeling example in chapter 3, where the overall knowledge model can be used for frequency domain small-signal purpose only, the knowledge model in this chapter is the complete nonlinear large signal model that can be used for time- and frequency-domain purposes.

The model development process includes 3 phases. Phase 1 is data preprocessing of DC and bias dependent S-parameter data. Phase 2 is neural network training, and Phase 3 is the reconstruction of the overall model for use in circuit design.

In Phase 1, we compute the difference between the device actual DC data (I_g and I_d) and those from the existing equivalent circuit model. Let the result be represented by ΔI_g and ΔI_d , respectively. In order to train the AC behavior, we also use the S-parameter data at various bias points for training. To facilitate computation, we convert the S-parameter into Y-parameter for both the device data and those computed from the existing model. We then compute the difference between the Y-parameter of the device and that of the existing model. Let ΔRY_{11} , ΔIY_{11} , ΔRY_{12} , ΔIY_{12} , ΔRY_{21} , ΔIY_{21} , ΔRY_{22} , ΔIY_{22} represent the result of such difference. Now we have V_g , V_d , frequency, ΔI_g , ΔI_d , ΔRY_{11} , ΔIY_{11} , ΔRY_{12} , ΔIY_{12} , ΔRY_{21} , ΔIY_{21} , ΔRY_{22} and ΔIY_{22} for each bias point. This can be used as training data for Phase 2.

In Phase 2, we train the neural model of Fig. 4-2 and Fig. 4-3 to learn the difference data processed in Phase 1. The objective of training is to minimize the least squares difference of $[\Delta I_g, \Delta I_d, \Delta RY_{11}, \Delta IY_{11}, \Delta RY_{12}, \Delta IY_{12}, \Delta RY_{21}, \Delta IY_{21}, \Delta RY_{22}, \Delta IY_{22}]$ between the current-charge neural model (or DNN model) and that of the data from Phase 1. The optimization methods used are conjugate gradient method and quasiNewton methods. For the current-charge based neural model, we use a recent adjoint neural network

method [31][52] to develop the difference model. For the DNN based neural model, we use a combined time and frequency domain training technique [30] to train the DNN to learn the difference data.

In Phase 3, we construct the overall model by parallelly connecting the existing equivalent circuit and the difference neural models, which can be either the current-charge model or the DNN model developed in Phase 2. When v_g and v_d signal are supplied to the model by user, the equivalent circuit model part will first produce a solution of i_{gequ} and i_{dequ} , and the current-charge neural model part will produce a correction, i.e., Δi_g and Δi_d . In the case of DNN based approach, the DNN will produce a higher order correction of Δi_g and Δi_d . With parallel connection, the total current for gate and drain terminals from the model will be $i_g = i_{gequ} + \Delta i_g$ and $i_d = i_{dequ} + \Delta i_d$, respectively. If the training in Phase 2 was done accurately, then this solution will be an accurate representation of the device behavior. This is achieved even though that the original equivalent circuits is not accurate.

4.3 Examples of Nonlinear GaAs MESFET Device Utilizing Difference Method

4.3.1 Example of Difference Method Based Current-Charge Model

This example shows knowledge-based method for nonlinear GaAs MESFET modeling. Training data for the device was obtained using simulated data from ADS simulator. Data was actually created using Statz model in ADS but we assume for this modeling example that we do not know the actual model behind the data. The purpose is

to develop a new model that accurately matches the data even though we are not using the original model.

The model used in this example is the knowledge-based method with the current-charge neural network representing the difference model. The overall model structure is shown in Fig. 4-4. Table 4-I shows model input parameters and their ranges.

Assume that the existing model that we are going to use as knowledge is the Curtice Model [17]. C_{gs} , C_{gd} , and i_{ds} are nonlinear voltage-controlled capacitances and drain current, respectively. Empirical formulas used are shown below [17].

$$i_{ds} = (a_0 + a_1 V_g + a_2 V_g^2 + a_3 V_g^3) \tanh(\gamma V_d) \quad (4.2)$$

$$Q_{gs} = \begin{cases} 2 \cdot V_{bi} \cdot C_{gs0} \cdot (1 - \sqrt{1 - \frac{V_{gs}}{V_{bi}}}), & V_{gs} < F_c \times V_{bi}; \\ 2 \cdot V_{bi} \cdot C_{gs0} \cdot (1 - \sqrt{1 - F_c}) + \frac{C_{gs0}}{(1 - F_c)^{3/2}} \times \left[(1 - \frac{3F_c}{2}) V_{gs} - F_c \cdot V_{bi} + \frac{V_{gs}^2 - (F_c \cdot V_{bi})^2}{4V_{bi}} \right], & V_{gs} \geq F_c \times V_{bi}. \end{cases} \quad (4.3)$$

The proposed nonlinear difference method is applied in this example to adjust the raw knowledge. Two kinds of differences are added to the empirical circuit. DIFF I is used to model the difference of nonlinear intrinsic part of the MESFET component, DIFF II is used to model the difference of the total model. Both of these two difference parts are realized by trained neural networks. A program called NeuroAdjoint is used to train the model to learn these differences. Fig. 4-5 ~ 4-7 show the modeling process of this structure. The number of hidden neurons for the i_{NNg} , i_{NNd} , q_{NNg} and q_{NNd} models is 12. The knowledge model (i.e., the Curtice model) was first optimized using parameter extraction to fit the original training data as much as possible. As expected, the model

only provides an approximate solution, and the match is not perfect as shown in Fig. 4-5. After training the current-charge neural network model to capture the difference, we then combine it with the Curtice knowledge model whose solution is compared with the original data in Fig. 4-6 and 4-7. This example shows that even though the knowledge of the equivalent circuit used is not perfectly correct, the overall knowledge-based model matches well with the original data.

Table 4-I Inputs/outputs of GaAs MESFET model.

Overall Model Input x	Vg (gate voltage, $-0.8 \sim -0.2\text{V}$) Vd (drain voltage, $2 \sim 5\text{V}$) f (frequency, $1 \sim 20\text{GHz}$)
Output y	S-parameters

Fig. 4-4: Current-charge neural model based on DM concept for nonlinear GaAs MESFET modeling.

Fig. 4-5 : S-parameter comparison of nonlinear GaAs MESFET models before adding difference part of DIFF I. Solid lines (—) represent S-parameters of the original data. Dashed lines with circles (o) represent the S-parameters of the model.

Fig. 4-6: S-parameter comparison of nonlinear GaAs MESFET models after adding difference part of DIFF I. Solid lines (—) represent S-parameters of the original data. Circles (o) represent S-parameters of the model.

Fig. 4-7: S-parameter comparison of nonlinear GaAs MESFET models after adding difference parts DIFF I and II. Solid lines (—) represent S-parameters of the original data. Dashed lines with circles (o) represent the S-parameters of the model.

4.3.2 Example of Difference Method Based DNN Model

This example shows DNN approach of knowledge-based method for nonlinear GaAs FET modeling. Training data for the device was the same as in 4.3.1, i.e., created using simulated data from ADS simulator. The knowledge used is also the same as that in 4.3.1, i.e., the Curtice model. In this example, we use the dynamic neural network based difference model to learn the differences between the original data and the existing knowledge model (Curtice model). The overall model structure is shown in Fig. 4-8. The model input parameters and their ranges are those shown in Table 4-II.

The difference data was obtained by the preprocessing phase as described earlier in the section. In order to train the DNN, we used a combination of a state-space based neural network training [33] and DNN training [30] to learn the difference between the Curtice model and the original data. The number of hidden neurons for the DNN models, i.e., i_g DNN model and i_d DNN model, is 72. The order of the DNN models is 3. Similar to the example in 4.3.1, the knowledge model (i.e., the Curtice model) was first optimized using parameter extraction to fit the original training data as much as possible. As expected, the model only provides an approximate solution, and the match is not perfect as shown in Fig. 4-9. We combined the Curtice knowledge model with the DNN difference model and compare the overall model behavior with the original data as shown in Fig. 4-10. This example also shows that even though the knowledge of the equivalent circuit used is not perfectly correct, the overall knowledge-based model matches well with the original data. Another interesting point is that DNN has the ability to represent higher order dynamics, therefore 1 set of such DNN model can produce solutions as good as 2 sets of difference models as in 4.3.1.

(a) ADS setup for original data generation

(b) Existing knowledge in form of Curtice equivalent circuit model

(c) Overall knowledge-based dynamic neural network including Curtice model and DNNs as the difference part

Fig. 4-8: DNN model based on DM concept for nonlinear GaAs MESFET modeling.

Fig. 4-9: Y-parameter comparison of nonlinear GaAs MESFET models before adding DNN models. Solid lines (—) represent Y-parameters of the original data. Dashed lines with circles (-o-) represent Y-parameters of the model. The 1st, 2nd, 3rd and 4th rows are the real and imaginary parts of Y_{11} , Y_{12} , Y_{21} and Y_{22} , respectively.

Fig. 4-10: Y-parameter comparison of nonlinear GaAs MESFET models after adding DNN models. Solid lines (—) represent Y-parameters of the original data. Dashed lines with circles (-o-) represent Y-parameters of the model. The 1st, 2nd, 3rd and 4th rows are the real and imaginary parts of Y_{11} , Y_{12} , Y_{21} and Y_{22} , respectively.

4.4 Conclusions

Nonlinear device modeling is one of the most important areas of microwave CAD. In this chapter, a novel dynamic neural modeling approach based on difference method for nonlinear microwave devices has been pioneered for the first time. Existing device models have been combined with dynamic neural networks (DNN) to form knowledge-based dynamic nonlinear device models. Equivalent circuit topology of the existing device model has been utilized as the knowledge part, and DNN models have been used to recover the difference between existing device models and training data. Preliminary examples have also demonstrated the advantage and availability of this new approach.

Chapter 5

Neuro-Space Mapping (Neuro-SM) for Nonlinear Modeling

5.1 Introduction

Artificial Neural Networks [1] – [4] and Space Mapping (SM) [11] are two recent developments in the microwave CAD area to address the growing challenges in today's microwave modeling, simulation and optimization. Neural networks can be trained to learn from microwave data, and the trained neural network can be used as microwave model providing fast solutions to the task it learned [1]. It has been applied to a variety of microwave modeling problems including both passive [53] and active [7] [30] modeling. Neural network computation is fast and it can generalize from data, allowing model development even when component formulas are unavailable. Space mapping, on the other hand, is an attractive concept for circuit design and optimization, combining the computational efficiency of coarse models with the accuracy of fine models. The coarse models are typically empirical equivalent circuit engineering models which are fast but has limited validity range for their parameters, beyond which the simulation results may become inaccurate. Detailed fine models can be provided by EM simulator or direct measurement. Space mapping establishes a mathematical link between coarse and fine models to achieve an optimization of fine accuracy without extensive use of fine model.

Recently space mapped neuromodeling technique was proposed combining neural networks with space mapping [11] [54]. A neural network is trained to map the coarse model towards fine model data. The results is a model with accuracy near fine model and the speed of the coarse model. The technique has been applied to passive component modeling such as bends, high temperature superconductor filters [11] [54] and embedded passives in multilayer printed circuits [55].

In this chapter, we present a new direction for space mapped neuromodeling, i.e., for large-signal dynamic nonlinear device modeling. Nonlinear device modeling is an important area of CAD, and many device models have been developed as reviewed in [56], and described in [57] – [59]. Thanks to the rapid technology development in semiconductor industry, new devices constantly evolve. Models that were developed to fit previous devices may not fit well the new devices. There is an ongoing and constant need of new models. The challenges facing CAD researchers are not only to develop better models, but also to develop new CAD methods to make the task of model creation more efficient and systematic. The latter aspect is addressed in this paper, i.e., CAD methods for efficient creation of device models.

We present a neuro-space mapping (Neuro-SM) technique, for automatic modeling of large-signal nonlinear devices. We define a given existing device model as coarse model. The proposed technique will automatically adjust and modify the coarse device model such that after space mapping, the mapped model will match the fine device data. To achieve this for large signal device modeling, the neural network mapping is formulated using voltage and current signals in the model. An efficient training technique is also proposed for training the neuro space mapping with DC and bias dependent S-parameter

data. The trained model is then used in large-signal harmonic balance simulation. The proposed technique can automatically improve a poor (i.e., coarse) device model into a good one (i.e., a fine model), avoiding otherwise tedious trial and error process in manual modification of the models. Compared with pure neural network modeling where the device is entirely modeled by neural networks, the proposed method has better extrapolation capability. Examples of DC, AC and large signal device models are presented.

5.2 Proposed Neuro-SM for Nonlinear Device Modeling

Coarse Model and Fine Model: Suppose that the existing/available models give only rough approximation of our device, and cannot accurately match the actual device data. Let the existing nonlinear device model be called the coarse model. The fine model in our case is only a fictitious model implied by actual device data from measurement or detailed/expensive device simulator.

Coarse Signal and Fine Signal: We use a 2-port device notation for our explanation. Let the terminal currents and voltage signals of the coarse device model be defined as $\mathbf{v}_c = [v_{c1}, v_{c2}]^T$ and $\mathbf{i}_c = [i_{c1}, i_{c2}]^T$, respectively. Let the terminal currents and voltages of the fine model be defined as $\mathbf{v}_f = [v_{f1}, v_{f2}]^T$, and $\mathbf{i}_f = [i_{f1}, i_{f2}]^T$, respectively. $\mathbf{v}_c$ and $\mathbf{i}_c$ are called coarse signals and $\mathbf{v}_f$ and $\mathbf{i}_f$ are called fine signals.

Neuro-SM: Fig. 5-1 shows the proposed Neuro-SM nonlinear 2-port network structure. The voltage signals for the overall model, i.e., v_{f1} and v_{f2} are not sent to the coarse model directly. Instead they are mapped (modified) into the voltages in the coarse model such that the modified coarse model response (say, $\mathbf{i}_c$) will match the fine signal.

Since the precise equation for this mapping is unknown, and the mapping in general can be nonlinear, a neural network becomes a logical choice as the realization of the mapping function. The input neurons for the neural network receives fine voltage signals v_{f1} , v_{f2} . The output neurons provides the mapped voltage signals for the coarse model, i.e., $v_c = f_{ANN}(v_f, \mathbf{w})$, where f_{ANN} represents the neural network and $\mathbf{w}$ is a vector containing all internal weights of the neural network. This neural network is then implemented as the functions in the voltage controlled voltage sources in our model shown in Fig. 5-1. We use current controlled current sources to pass i_c to i_f , in order to make the Neuro-SM model consistent with Kirchhoff's Laws as seen from the external terminals of the overall Neuro-SM model, which includes the coarse nonlinear model, all the controlled sources, and the neural network.

Fig. 5-1: General 2-port Neuro-SM nonlinear model.

Cases of Mapping: For the Neuro-SM model to perform accurately, the neural network should be trained. However, available transistor data may not be directly in the form of instantaneous voltages (coarse and fine) for the input-output neurons as required by regular neural network training algorithms. Here we establish the connections of the proposed mapping with typical types of transistor data, such as DC, bias-dependent S-parameters and large-signal harmonic data, in order to formulate a new neural network training approach.

The proposed Neuro-SM model is a full large-signal nonlinear model. The mapping for DC voltages $V_{c,DC}$ and $V_{f,DC}$ is directly achieved by the neural network as:

$$V_{c,DC} = f_{ANN}(V_{f,DC}, \mathbf{w}). \quad (5.1)$$

The small signal S-parameters are mapped via the mapping relationship of the Y matrices between the coarse model Y_c and fine model Y_f , as

$$Y_f = Y_c \cdot \left(\left. \frac{\partial f_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f} \right|_{\mathbf{v}_f = V_{f,Bias}} \right)^T. \quad (5.2)$$

where the derivative of f_{ANN} is obtained at bias point $V_{f,Bias}$ using the adjoint neural network in Fig. 5-2. As for large-signal case, the mapping of harmonic signals between the coarse model $V_c(k\omega)$ and fine model $V_f(l\omega)$ is:

$$V_c(k\omega) = \frac{1}{N_T} \sum_{n=0}^{N_T-1} f_{ANN} \left(\sum_{l=0}^{N_H} V_f(l\omega) \cdot W_N^{-(l\omega \chi n T)}, \mathbf{w} \right) \cdot W_N^{(k\omega \chi n T)} \quad k=0, 1, \dots, N_H. \quad (5.3)$$

where ω is fundamental frequency, N_H is number of harmonics, T is time interval and N_T is the number of time points. W_N is defined as $e^{-j2\pi}$.

Training of Neuro-SM Model: The overall training has two phases, initialization and formal training. In the initialization phase, we first initialize the neural network by a preliminary training to learn unit mapping, i.e., $v_c = v_f$. Training data can be obtained by assigning $[v_{c1}, v_{c2}]$ in a grid form across the entire operation range of the device. This initialization phase guarantees that the overall Neuro-SM model is equal to the coarse model, before actual device data is used in the formal training of the neural network. In the formal training phase, the neural network internal weights w are adjusted such that the Neuro-SM model matches device data better than the coarse model does. In this way, for a given device model, the proposed Neuro-SM model automatically exceeds or at least equals the performance of the given coarse model.

The overall training error can be the total difference between all available device data (such as DC and bias-dependent S-parameters) and the Neuro-SM model. The derivative of the training error versus neural network weights needed by neural network training algorithms is obtained by differentiating the mapping relationships of (5.1)-(5.3) with the derivative part in the coarse model done through circuit sensitivity techniques and the derivative part for f_{ANN} done through adjoint neural network sensitivity [31].

Use of Neuro-SM Model: After training, the Neuro-SM model can be used by user or circuit simulator. The neural network internal weights w are fixed. The voltage/current relationship of the model required by user or circuit simulator is that between v_f and i_f , which is obtained from Neuro-SM model through the mapping of coarse model signals as defined in Figure 5-1.

Fig. 5-2. (a) Original neural network $v_c = f_{ANN}(v_f, w)$ and (b) adjoint neural network in Neuro-SM. (a) is used for DC and large-signal mapping and (b) is used for small-signal mapping. Vector w contains all weights $w_{ik}^{(1)}$ and $w_{kj}^{(2)}$, where i, k, j are indices of neurons in the input, hidden and output layers in (a). The adjoint neural network structure corresponds to a flip of the original neural network between inputs and outputs. The output of adjoint neural network is $\partial v_{ci} / \partial v_f$ where $i=1$ or 2 if the adjoint inputs are $[1, 0]$ or $[0, 1]$, respectively.

5.3 Examples

5.3.1 Neuro-SM Nonlinear Models Used for DC Case

A. Neuro-SM Nonlinear Model for SiGe HBT Device

This example shows how Neuro-SM works in a simple DC case of a SiGe HBT device modeling, with data from measurement [60]. The coarse model used is a standard Curtice model [17]. The internal parameters of the coarse model are first optimized. However, the coarse model at its best provides only approximation of the device and lacks the complicated details seen in the device data. We applied the proposed Neuro-SM technique. The base current and collector voltage are first mapped onto the coarse model, and these mapped signals then excite the coarse model, resulting in an improved value of collector current. The input range is: $I_b = 0 \sim 1.05\text{mA}$ and $V_{ce} = 0 \sim 4\text{V}$. Simulation result comparison is shown in Fig 5-3. The neural network used 55 hidden neurons. In general, fewer (more) hidden neurons are needed if the coarse model is good (poor). This example demonstrates that using Neuro-SM, a basic Curtice model originally developed for GaAs MESFETs can now be automatically extended beyond its original limitation to match the highly irregular nonlinear behavior in the SiGe HBT example.

B. Neuro-SM Nonlinear Model for Physics MOSFET Device

We also apply our Neuro-SM technique to achieve a MOSFET device model using data from physics simulation by MINIMOS [61]. Gate voltage and drain voltage are used as inputs and drain-source current is output. The input range is: $V_g = 0 \sim 2\text{V}$ and $V_d = 0 \sim 2\text{V}$. Standard Curtice model again was used here as the coarse model, which was first optimized to achieve its best performance. For Neuro-SM model, the mapping part first mapped the input gate voltage and drain voltage onto the coarse model, and then the

coarse model will produce the drain-source current as output of the overall model. The comparison of I-V curves between the fine model, our Neuro-SM nonlinear model and coarse nonlinear model is shown in Fig. 5-4.

Fig. 5-3: I-V comparison of SiGe HBT device with coarse model and Neuro-SM model.

Fig. 5-4: I-V comparison of physical MOSFET device with coarse model and Neuro-SM model.

5.3.2 Neuro-SM Nonlinear Model for GaAs MESFET Device

This example illustrates a full large signal Neuro-SM model trained with both DC and bias dependent S-parameter data. The fine device data is generated from an ADS internal GaAs FET model [18]. The Curtice cubic model [17] is used as coarse model. There are clear differences between the coarse model and the fine device data, which cannot disappear even after the parameters in the coarse model are optimized as much as possible. We will use our proposed Neuro-SM to automatically adjust the coarse model to match the fine device data. Fig. 5-5 shows the structure of this Neuro-SM nonlinear model.

Training was done using both DC and S-parameter data at 150 bias points in the range (V_g : -1 to 0V, V_d : 0 to 5V, and frequency: 1 to 20 GHz). The number of hidden neurons used was 10. After training, we compare the Neuro-SM nonlinear model with the coarse model and the original ADS data. The result is plotted in Fig. 5-6, showing clear improvements using Neuro-SM over the coarse model. For comparison purpose, we also trained a pure neural network model (device entirely modeled by dynamic neural network (DNN) [30] without use of any equivalent circuit). Table 5-I gives the comparison of accuracy and extrapolation capability between the models. Two sets of data are used with 10% and 20% derivation beyond bias and frequency range for extrapolation test. The proposed Neuro-SM model outperforms the pure neural network model when the model is used outside its training range.

Fig. 5-5: Structure of Neuro-SM GaAs MESFET nonlinear model.

Table 5-I Test error as well as extrapolation capability comparison between pure neural model (DNN), coarse model and neuro-SM model.

Model Type	Within training range		Extrapolation beyond training range (10%/20%)
	Before training	After training	
Pure Neural Model	133.34	1.38	36.25/43.81
Coarse Model	10.57	10.57	10.36/10.44
Neuro-SM Model	10.57	1.46	1.87/2.03

Fig. 5-6: Comparison between original ADS GaAs MESFET model, coarse model and Neuro-SM model. (a) DC comparison and (b) S-parameter comparison at 4 different biases.

5.3.3 Use of Neuro-SM GaAs MESFET Model in Amplifier Design

In this example, we use the trained Neuro-SM GaAs FET nonlinear model of last section in three-stage power amplifier [31] simulation and optimization as shown in Fig. 5-7. All the passive components are represented by MMIC elements: spiral inductors, MIM capacitors, bulk resistors and transmission lines. Optimization is performed w.r.t. specifications on the Gain and VSWR of the amplifier as shown in Fig. 5-8. Eventhough the optimization was carried out using the proposed model, the amplifier solutions using the model match that of the amplifier using the original ADS model. We also performed large-signal harmonic balance simulation of the amplifier, and compared the solutions with that of the original ADS solutions, shown in Fig. 5-9 and 5-10. This verifies the validity of the large-signal behavior of the proposed Neuro-SM device model. To further demonstrate the use of the proposed model, we also performed yield optimization and Monte-Carlo analysis with 1000 statistical outcomes of the three stage amplifier using the Neuro-SM models while 53 geometrical parameters of all the resistors, capacitors, inductors and transmission lines were repetitively changed by optimization and statistical analysis. The results shown in Table 5-II further demonstrate that the Neuro-SM model can be used to provide reliable solutions for statistical analysis and optimization.

Fig. 5-7: Three stage power amplifier.

Fig. 5-8: Gain and VSWR comparison of three stage power amplifier.

Table 5-II Yield comparison of original ADS model, coarse nonlinear model and fine nonlinear model.

	Yield Before Optimization	Yield After Optimization
Original ADS model	46.2%	73.0%
Coarse model	17.4%	26.1%
Neuro-SM model	41.0%	69.6%

(a)

(b)

Fig. 5-9: Time domain response (a), and frequency domain harmonic balance response (b) of the three stage amplifier using original ADS solution, Neuro-SM model and coarse model. Input power is 7dBm, frequency is 6.5GHz.

(a)

(b)

Fig. 5-10: Time domain response (a), and frequency domain harmonic balance response (b) of the three stage amplifier using original ADS solution, Neuro-SM model and coarse model. Input power is 7dBm, frequency is 5GHz.

5.4 Conclusions

A new neuro-space mapping (Neuro-SM) approach has been presented enabling the space mapping (SM) concept to be applied to nonlinear device modeling and large signal circuit simulation. By modifying the voltage/current signals fed to the model using Neuro-SM technique, we can automatically improve an existing model of coarse accuracy into a new model of fine accuracy. New training methods for such mapping neural networks have been proposed. Examples of SiGe HBT, MOSFET and GaAs MESFET modeling and use of the models in harmonic balance simulation demonstrated that Neuro-SM is a systematic method to allow us to exceed the present capabilities of the existing device models.

Chapter 6

Conclusions and Future Work

This thesis addressed new approaches of improving neural network performance and reducing the cost of obtaining training data based on knowledge-based neural network. When training data are insufficient, knowledge-based structure has been shown much better performance than pure neural network structure such as MLP. Knowledge-based models can learn and predict behaviors beyond training ranges, which is one of the most significant characteristics of this advanced technique.

A unified approach of knowledge-based neural network, GKBNN, has been proposed for improved model accuracy and extrapolation capability with reduced need of training data. The proposed GKBNN retains the advantages of all the existing knowledge-based methods and is suitable in a much wider variety of cases than all the existing methods combined. The proposed method can be much faster than direct physical/EM models of microwave components and is useful for efficient microwave circuit design involving highly repetitive computations such as design optimization, statistical design and yield optimization.

We have also presented for the first time knowledge-based neural network approaches for nonlinear device modeling. Two kinds of knowledge-based neural network models are developed using difference method and space mapping concept. New problem formulations and training algorithms for both methods have also been

developed. Preliminary examples utilizing these proposed knowledge-based model structures have demonstrated that the proposed knowledge-based dynamic models can achieve good accuracy even though the knowledge of equivalent circuit models is only approximate. This proposed approach can lead to efficient model building, avoiding otherwise inefficient trial and error process in manual adjustment of equivalent circuit topology and nonlinear formulas.

These knowledge-based modeling approaches addressed in this thesis have the potential of automating the model creation and updating process, contributing to increased efficiency in computer-aided design of microwave circuits and systems. As a future direction, we will continue to expand knowledge-based nonlinear device modeling approach to cover more varieties of knowledge structures, such as PKI, KBNN and GKBBN, and apply the method to more varieties of device modeling examples and scenarios. Dynamic neural network will also be utilized in developing such knowledge-based techniques as SM, PKI, KBNN, etc., to achieve more efficient and flexible models. Adjoint sensitivity analysis using adjoint neural networks will contribute to new training method development. And we will also focus on developing knowledge-based structures that can be directly incorporated into circuit simulators to perform circuit simulation and optimization, which will make it possible for microwave engineers to efficiently apply the proposed techniques. This work makes computer-based automatic modification of existing large-signal device models become achievable, avoiding otherwise trial and error based manual modification of models, and it is also aimed at efficient and automatic updating of nonlinear device model libraries as new semiconductor technologies continue to evolve.

Bibliography

- [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, 1996.
- [3] P. Burrascano, S. Fiori, M. Mongiardo, "A review of artificial neural networks applications in microwave computer-aided design", *Int. J. RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, 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, 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, 1997.
- [6] G.L. Creech, B.J. Paul, C.D. Lesniak, T.J. Jenkins, and M.C. Calcaterra "Artificial neural networks for fast and accurate EM-CAD of microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 794-802, 1997.

- [7] 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, 1995.
- [8] Y. Harhouss, J. Rousset, H. Chéhadé, 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 and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, pp. 198-215, 1999.
- [9] V.B. Litovski, J.I. Radjenovic, Z.M. Mrcarica, and S.L. Milenkovic, "MOS transistor modeling using neural network," *Elect. Lett.*, vol. 28, pp. 1766-1768, 1992.
- [10] V.K. Devabhaktuni, C. Xi, and Q.J. Zhang, "A neural network approach to the modeling of heterojunction bipolar transistors from S-parameter data," *Proc. 28th European Microwave Conf.*, (Amsterdam, Netherlands), Oct. 1998, pp. 306-311.
- [11] 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, 1999.
- [12] S. Goasguen, S.M. Hammadi, and S.M. El-Ghazaly, "A global modeling approach using artificial neural network", *IEEE MTT-S Int. Microwave Symp. Digest*, (Anaheim, CA), pp.153-156, 1999.
- [13] 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 and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, pp. 175-186, 1999.
- [14] M. Vai and S. Prasad, “Neural networks in microwave circuit design – beyond black box models”, *Int. J. RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, pp. 187-197, 1999.
- [15] F. Wang and Q.J. Zhang, “Knowledge-based neural models for microwave design”, *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2333-2343, 1997.
- [16] P.M. Watson, K.C. Gupta and R.L. Mahajan, “Application of knowledge-based artificial neural network modeling to microwave components”, *Int. J. RF and Microwave CAE*, vol. 9, No. 3, pp. 254-260, 1999.
- [17] W.R. Curtice, “GaAs MESFET modeling and nonlinear CAD”, *IEEE Trans. Microwave Theory Tech*, vol. 36, pp. 220-230, 1988.
- [18] H. Statz, et al, “GaAs FET Device and Circuit Simulation in SPICE”, *IEEE Trans. Electron Devices*, vol. ED-34, No. 2, pp. 160-169, 1999.
- [19] Materka and T. Kacprzak, “Computer calculation of large-signal GaAs FET amplifier characteristics,” *IEEE Trans. Microwave Theory Tech.*, vol. 33, pp. 129-135, 1985.
- [20] D. E. Root, S. Fan, and J. Meyer, “Technology independent large-signal non quasistatic FET models by direct construction from automatically characterized device data”, *Proc. 21st Eur. Microwave Conf.*, pp. 927-932, 1991.
- [21] L. Angelov, H. Zirath, and N. Rorsman, “New empirical nonlinear model for HEMT and MESFET and devices”, *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 2258-2266, 1992.

- [22] C.M. Snowden, "Nonlinear modelling of power FET's and HBT's", *Int. J. Microwave and Millimeter-wave Computer-Aided Eng.*, vol. 6, pp. 219-233, 1996.
- [23] Q. Li and R. W. Dutton, "Numerical small-signal AC modelling of deep-level-trap related frequency dependent output conductance and capacitance for GaAs MESFET's on semi-insulating substrates", *IEEE Trans. Electron Devices*, vol. 38, pp. 1285-1288, 1991.
- [24] M. A. Khatibzadeh and R. J. Trew, "A large-signal analytical model for the GaAs MESFET", *IEEE Trans Microwave Theory Tech.*, vol. 36, pp. 231-238, 1988.
- [25] C.M. Snowden and R. R. Pantoja, "Quasi two-dimensional MESFET simulation for CAD", *IEEE Trans. Electron Devices*, vol. 36, pp. 1564-1574, 1989.
- [26] 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 and Microwave CAE*, vol. 11, pp. 4-21, 2001. (invited review paper).
- [27] J.W. Bandler, R.M. Biernacki, Q. Cai, S. H. Chen, S. Ye and Q. J. Zhang, "Integrated physics-oriented statistical modeling, simulation and optimization", *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1374-1400, 1992.
- [28] 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, 2000.
- [29] L. Zhang, R.T. Ding and Q.J. Zhang, "Generalized knowledge-based neural network for microwave modeling", *9th Int. Sym. On Antenna Technology and Applied Electromagnetics*, (Saint-Hubert, Quebec), pp. 558-561, 2002.

- [30] J.J. Xu, M.C.E. Yagoub, and Q.J. Zhang, "Neural based dynamic modeling of nonlinear microwave circuits" *IEEE MTT-S Int. Microwave Symp. Digest*, (Seattle, WA), pp. 1101-1104, 2002.
- [31] J.J. Xu, M.C.E. Yagoub, and Q.J. Zhang, "Exact adjoint sensitivity for neural based microwave modeling and design," *IEEE MTT-S Int. Microwave Symp. Digest*, (Phoenix, AZ), pp. 1015-1018, 2001.
- [32] X. Ding, B. Chattaraj, M.C.E. Yagoub, V.K. Devabhaktuni, Q.J. Zhang, "EM based statistical design of microwave circuits using neural models", *Int. Symp. on Microwave and Optical Technology (ISMOT 2001)*, (Montreal, Quebec), pp.421-426, 2001. (Invited Paper).
- [33] X. Ding, J.J. Xu, M.C.E. Yagoub, Q.J. Zhang, "A new modeling approach for embedded passives exploiting state space formulation", *32th European Microwave Conf. (EuMC 2002)*, (Milan, Italy), 2002.
- [34] Q.J. Zhang, M.C.E. Yagoub, X. Ding, D. Goulette, R. Sheffield, and H. Feyzbakhsh, "Fast and accurate modeling of embedded passives in multi-layer printed circuits using neural network approach", *Elect. Components & Tech. Conf.*, (San Diego, CA), pp. 700-703, 2002.
- [35] R. Biernacki, J.W. Bandler, J. Song, and Q.J. Zhang, "Efficient quadratic approximation for statistical design", *IEEE Trans. Circuit Syst.*, vol. 36, pp. 1449-1454, 1989.
- [36] P. Meijer, "Fast and smooth highly nonlinear multidimensional table models for device modeling", *IEEE Trans. Circuit Syst.*, vol. 37, pp. 335-346, 1990.

- [37] Q.J. Zhang, F. Wang, and M.S. Nakhla, "Optimization of high-speed VLSI interconnects: A review", *Int. J. of Microwave and Millimeter-Wave CAD*, vol. 7, pp. 83-107, 1997.
- [38] K. Shirakawa, M. Shimizu, N. Okubo and Y. Daido, "Structural determination of multilayered large signal neural network HEMT model", *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 1367-1375, 1998.
- [39] P. Burrascano, M. Mongiardo, C. Fancelli, and M. Mongiardo, "A neural network model for CAD and optimization of microwave filters", *IEEE Int. Microwave Symp.*, (Baltimore, MD), pp. 13-16, 1998.
- [40] C. Christodoulou, A.E. Zooghy, and M. Georgiopoulos, "Neural network processing for adaptive array antennas" *IEEE-APS Int. Symp.*, (Orlando, FL), pp.2584-2587, 1999.
- [41] G. Kothapali, "Artificial neural networks as aids in circuit design", *Microelectronics J.*, vol.26, pp. 569-678, 1995.
- [42] Q.J. Zhang and M.S. Nakhla, "Signal integrity analysis and optimization of VLSI interconnects using neural network models", *IEEE Int. Circuits Syst. Symp.*, (London, England), pp. 459-462, 1994.
- [43] T. Hong, C. Wang, and N.G. Alexopoulos, "Microstrip circuit design using neural networks", *IEEE Int. Microwave Symp.*, (Atlanta, Georgia), pp. 413-416, 1993.
- [44] M. D. Baker, C.D. Himmel, and G.S. May, "In-situ prediction of reactive ion etch endpoint using neural networks" *IEEE Trans. Components, Packaging, and manufacturing Tech. Part A*, vol.18, pp. 478-483, 1995.

- [45] M. Vai and S. Prasad, "Automatic impedance matching with a neural network", *IEEE Microwave Guided Wave Letter*, vol. 3, pp. 353-354, 1993.
- [46] 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, 1997.
- [47] K. Shirakawa, et al., "A large signal characterization of an HEMT using a multilayered neural network", *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1630-1633, 1997.
- [48] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, New York: Van Nostrand Reinhold, 1983.
- [49] *NeuroModeler*, Q.J. Zhang, Dept. of Electronics, Carleton University, Ottawa, Canada.
- [50] S. Goasguen, S. M. El-Ghazaly, "A coupled FDTD-artificial neural network technique for large-signal analysis of microwave circuits", *Int. J. RF and Microwave CAE*, vol. 12, No. 1, pp 25-36, 2002.
- [51] Advanced Design System (ADS), Agilent Technologies, 395 Page Mill Road, Palo Alto, CA, U.S.A.
- [52] *NeuroAdjoint*, Q.J. Zhang, Dept. of Electronics, Carleton University, Ottawa, Canada.
- [53] P.M. Watson, K.C. Gupta and R.L. Mahajan, "Application of knowledge-based artificial neural network modeling to microwave components", *Int. J. RF and Microwave CAE*, vol. 9, pp. 254-260, 1999.

- [54] J.W. Bandler, M.A. Ismail, J.E. Rayas-Sanchez, and Q.J. Zhang, "Neural inverse space mapping EM optimization," *IEEE MTT-S International Microwave Symp. Digest*, (Phoenix, AZ), vol. 2, pp. 1007-1010, 2001.
- [55] V.K. Devabhaktuni, B. Chattaraj, M. Yagoub and Q.J. Zhang, "Advanced microwave modeling framework exploiting automatic model generation, knowledge based neural networks and space mapping," *IEEE MTT-S International Microwave Symp. Digest*, (Seattle, WA), pp. 1097-1100, 2002.
- [56] M.B. Steer, J.W. Bandler and C.M. Snowden, "Computer-aided design of RF and microwave circuits and systems," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 996-1005, 2002.
- [57] J.M. Golio Ed., *The RF and Microwave Handbook*, Boca Raton, FL: CRC Press, 2001.
- [58] C.M. Snowden, *Semiconductor Device Modeling*, Stevenage, UK: Peregrinus, 1988.
- [59] J.W. Bandler, N. Georgieva, M.A. Ismail, J.E. Rayas-Sanchez and Q.J. Zhang, "A generalized space mapping tableau approach to device modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 49, pp. 67-79, 2001.
- [60] C.N. Rheinfelder, F.J. Beibwanger and W. Heinrich, "Nonlinear modeling of SiGe HBT's up to 50 GHz," *IEEE Trans. Microwave Theory and Tech.*, vol. 45, pp. 2503-2508, 1997.
- [61] MINIMOS v.6.0, Institute for Microelectronics, Technical University, Vienna, Austria.