Computer Aided EM Based Modeling and Optimization Using Neural Models

by

Biswarup Chattaraj, 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

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

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.

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 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.

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.

0-612-79697-3

Canada

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

Computer Aided EM Based Modeling and Optimization Using Neural Models

Submitted by

Biswarup Chattaraj, B. Eng.,

in partial fulfillment of the requirements for the degree of Master of Applied Science

Q.J. Zhang, Thesis Supervisor

M.C.E. Yagoub, Thesis Co-supervisor

M.S. Nakhla, Chair, Department of Electronics

Abstract

This work was motivated by the potential of neural networks in the design of electrical devices/circuits and components and the increasing need for fast, accurate and efficient modeling in the RF/microwave. The objective of this thesis is to develop algorithms/methodologies to combine some of the advanced concepts like Automatic Model Generation (AMG), Knowledge Based Neural Networks (KBNN), Prior Knowledge Inputs (PKI) networks, Space Mapping (SM) networks with the neural network structures. Following this a robust Knowledge based Automatic Model Generation (KAMG) technique is proposed that is aimed at generating microwave neural models of very high accuracy using fewer accurate data.

The other objective of this thesis is the EM based circuit optimization and statistical design based on neural models. The models generated by the concept mentioned above is applied to circuit design where EM-based neural models can represent original passive elements. EM based design of multilayer Printed Circuit Boards (PCB) including power/ground planes, drivers and other components is a challenging endeavor for the circuit designer. In this thesis we have described a new neural network formulation of the power plane, which allows automatic adjustment of component and connection geometry and then provide fast estimation of the EM effects with respect to the different connecting positions for circuit optimization and statistical design based on neural networks. By developing such models repetitive set of EM calculations for the power plane inductive mesh can be avoided thereby enhancing computational speed. The models then can be used efficiently for circuit simulation and design optimization.

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Q.J. Zhang and to my co-supervisor Prof. M.C.E. Yagoub, for their professional guidance, suggestion and support throughout the research work and preparation of this thesis. I am also grateful to Prof. Ram Achar for his invaluable suggestions during the course of my research work.

Vijay Devabhaktuni, Xiaolei Ding, Jianjun Xu, Dharmendra Saraswat and other students in my group are thanked for their help, friendship and enjoyable collective work. I am also grateful to Betty Zahalan, Peggy Piccollo, Scott Bruce and all other staff and faculty for providing the excellent lab facilities and friendly environment for study and research.

I would also like to thank Mr. Dennis Elias, Mrs. Najat Elias and Ms Binata for their endless love, support and encouragement.

Finally, I would like to thank my parents, Mr. Shyamal Baran Chattaraj and Mrs. Tripti Chattaraj, my brother Mr. Sreerup Chattaraj and other family members back in India for their love, support and encouragement. Without their years of encouragement and endless support, I would not have the opportunity to complete this work

Content

1	Introduction.....	1
1.1	Motivation of Research Work.....	1
1.2	Thesis Objective.....	4
1.3	Organization of the Thesis.....	5
2	Introduction to Neural based Modeling and Literature Review	7
2.1	Neural Model Development: Conventional Method.....	7
2.1.1	Neural Network Structure	9
2.1.2	Data Generation.....	11
2.1.3	Training the Neural Model	12
2.1.4	Testing the Neural Model.....	13
2.2	Knowledge Architecture.....	14
2.3	Circuit Application of Neural Model.....	15
3	Automatic Driving of Data Generators: A Concept	17
3.1	Automation of Data Generators.....	17

3.1.1 Automatic Driving of <i>Sonnet-Lite</i>	20
3.1.2 Automatic Driving of <i>Ansoft-HFSS</i>	23
4 A Knowledge Based Approach for Neural Model Generation ...	26
4.1 Coarse Model Generation.....	28
4.2 Sub-Model Training and Overall Model Initialization.....	28
4.2.1 Difference Method.....	29
4.2.2 Prior Knowledge Input.....	29
4.2.3 Space Mapping.....	30
4.2.4 Knowledge Based Neural Network.....	30
4.3 Final Training.....	35
4.4 Examples by KAMG Approach.....	35
5 Circuit Application and Statistical Design based on Neural	
Model	59
5.1 Introduction.....	60
5.2 Proposed representation of circuit components by neural model.....	62
5.2.1 Neural Models of Passive Components.....	62
5.2.2 Neural Models of Power Plane.....	66
5.2.3 Formulation of overall circuit including power plane.....	68
5.2.4 Statistical Design Concept	68
5.3 Examples.....	71

5.3.1	Amplifier.....	71
5.3.1.1	EM based amplifier circuit optimization using neural networks.....	71
5.3.1.2	EM based Monte-Carlo analysis of amplifier using neural networks...	74
5.3.2	Board-Level Interconnects.....	76
5.3.2.1	EM based circuit optimization of board-level interconnect using neural networks.....	76
5.3.2.2	EM based Monte-Carlo analysis of the board-level interconnect using neural networks.....	85
6	Conclusion and Future Research	86
6.1	Conclusion.....	86
6.2	Suggestion for Future Research	88
	Appendix A.....	89
	Appendix B.....	94
	Bibliography.....	97

List of Figures

2.1 Sequential flow-charts of steps involved in neural model development.	8
2.2 Multilayer Perceptron (MLP) structure. Typically, the network consists of an input layer, one or more hidden layers and an output layer.	10
3.1 Comparison between the conventional and the proposed method of data gen- eration. In the proposed method intensive PC work replaces the manual input.	19
3.2 Automatic Driving of Sonnet-Lite.	20
3.3. Automatic Driving of Ansoft-HFSS.	23
3.4. Steps for generation of Ansoft-HFSS macro.	24
4.1 Schematic Diagram of the proposed Difference Method.	31
4.2 Schematic Diagram of the proposed PKI Method.	32
4.3 Schematic Diagram of the proposed SM Method.	33
4.4 Schematic Diagram of the proposed KBNN Method.	34
4.5 3-D layout of the MOSFET modeled using proposed KAMG technique.....	37
4.6 Neural Model of 2-input MOSFET model.....	37
4.7 Accuracy comparison of 2-input MOSFET neural model generated by proposed KAMG and existing AMG techniques. KAMG achieved better accuracy with fewer fine data.....	38

4.8 Neural Model of the 5-input MOSFET model.....	41
4.9 Accuracy comparison of a 5-input MOSFET neural model generated by proposed KAMG and existing AMG techniques.....	41
4.10 3-D layout of Embedded Resistor modeled using proposed KAMG technique...	44
4.11 Neural Model of Embedded Resistor.....	45
4.12 Accuracy comparison of embedded resistor neural models generated by proposed KAMG and existing AMG.....	45
4.13 Comparison of real and imaginary parts of the S_{11} parameter of embedded resistor obtained from proposed KAMG (—) and from EM simulators (o)	46
4.14 3-D layout of Embedded Capacitor modeled using proposed KAMG technique.....	48
4.15 Neural Model of Embedded Capacitor.....	49
4.16 Accuracy comparison of embedded capacitor neural models generated by proposed KAMG and existing AMG techniques.	50
4.17 3-D layout of the 2-loop spiral inductor modeled using proposed KAMG technique.	52
4.18 Neural Model of 2-Loop Spiral Inductor.....	52
4.19 Accuracy comparison of 2-loop Spiral Inductor neural models generated by proposed KAMG and existing AMG techniques.....	53
4.20 Neural Model of 8-Loop Spiral Inductor.....	56
4.21 Accuracy comparison of 8-loop Spiral Inductor neural models generated by proposed KAMG and existing AMG techniques.....	57
5.1 Comparison of S-parameters of the lumped 10Ω resistor (—) and its EM	

based neural model (---).....	63
5.2 Physical illustration of a circuit including the power plane for EM based design optimization.....	67
5.3 Neural model input-output parameter of the power plane.	67
5.4 RC Amplifier.....	72
5.7 Amplifier circuit example where all the passive components and power plane are represented by neural models. N_P , N_R and N_C represent the neural models of the power planes, resistors and capacitors respectively.....	72
5.5 Comparison of the amplifier gain obtained by lumped elements (—) and by their neural models before (---) and after optimization (-o-o)... ..	74
5.6 Gain vs. Frequency for <i>ADS</i> lumped components (--o--) and Monte-Carlo analysis using Neural models (—).....	76
5.7 Physical representations of the 4-bits board-level interconnect.	77
5.8 4-Bit board-level interconnects circuit.....	78
5.9 The board-level interconnects example. Signal responses of the circuit before (—) and after (----) optimization. (a) Signal response V_a . (b) Crosstalk V_b (c) Signal response V_c . (d) Crosstalk V_d	81-82
5.10 Power Supply noise of the power plane for each of the 4 IC's used in the board-level interconnect before (—) and after (—) optimization. (a) Signal response at node V (ic_1); (b) Signal response at node V (ic_2); (c) Signal response at node V (ic_3); (d) Signal response at node V (ic_4);	83-84
5.11 Monte Carlo analyses of the board-level interconnect using Neural models....	85

List of Tables

4.1. NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 2-INPUT MOSFET MODELS AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.	38
4.2. COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECH- NIQUES TO ACHIEVE 2-INPUT MOSFET MODELS WITH 0.50% TEST ERROR....	39
4.3. ACCURACY COMPARISON BETWEEN 2-INPUT MOSFET NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 23 FINE DATA.....	39
4.4. NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 5-INPUT MOSFET MODELS AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.	42
4.5: COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECH- NIQUES TO ACHIEVE 5-INPUT MOSFET MODELS WITH 0.50% TEST ERROR....	42
4.6: ACCURACY COMPARISON BETWEEN 5-INPUT MOSFET NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 454 FINE DATA.	43
4.7: NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED RESISTOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.....	47
4.8: ACCURACY COMPARISON BETWEEN RESISTOR NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 18 FINE DATA.	47

4.9: NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED CAPACITOR	
MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.....	49
4.10: COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING	
TECHNIQUES TO ACHIEVE CAPACITOR MODELS WITH 1% TEST ERROR	50
4.11: NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 2-LOOP SPIRAL	
INDUCTOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.	53
4.12: ACCURACY COMPARISON BETWEEN 2-LOOP SPIRAL INDUCTOR NEURAL	
MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 386	
FINE DATA.....	54
4.13 COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING	
TECHNIQUES TO ACHIEVE 2-LOOP SPIRAL INDUCTOR MODELS WITH 1%	
TEST ERROR.....	54
4.14 NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 8-LOOP SPIRAL	
INDUCTOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER. ...	56
4.15: ACCURACY COMPARISON BETWEEN 8-LOOP SPIRAL INDUCTOR NEURAL	
MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 32	
FINE DATA.....	57
4.16: COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING	
TECHNIQUES TO ACHIEVE 8-LOOP SPIRAL INDUCTOR MODELS WITH 1%	
TEST ERROR.....	58

4.14 NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED RESISTOR

MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER..... 64

4.14 NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED CAPACITOR

MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.65

5.3 POSITION OF THE COMPONENT CONNECTIONS TO THE POWER PLANE FINE

BEFORE AND AFTER OPTIMIZATION..... 73

5.4 POSITION OF THE COMPONENT CONNECTIONS TO THE POWER PLANE FINE

BEFORE AND AFTER OPTIMIZATION..... 80

5.5 LENGTH AND WIDTH OF THE TRANSMISSION LINE BEFORE AND AFTER

OPTIMIZATION..... 80

Chapter 1

Introduction

1.1 Motivation of Research Work

The rapid development of commercial market for the electronics and the telecommunication industry over the past decades has led to many circuit design and analysis problem which are large and computationally expensive. This has resulted in an explosion of interest for improved circuit design approaches in the radio frequencies (RF) and microwave areas. RF and microwave circuit design has progressed considerably and has achieved a certain level of maturity in recent years. Electromagnetic (EM) simulation techniques for high frequency structures developed over the past decades have helped to bring the Computer Aided Design (CAD) for hybrid RF/microwave circuits to its current state of the art [1].

The efficiency of RF/microwave CAD tools relies heavily upon the speed and accuracy of component models [2-4]. It is for this reason several approaches to device modeling are being continuously researched upon. Detailed physics-based models (PBM) are very accurate [4] but are very slow. Equivalent circuit models and empirical models are faster than PBM's. However, they are problem-specific and lots of time and efforts are needed to develop such models. Table look-up models can be relatively fast,

but suffer from the disadvantages of large memory requirements and limitation on number of parameter [1]. For high-level optimization of circuit parameters, the component models are continuously varied both with frequency and geometrical and/or electrical parameters. So modeling techniques that can provide such continuous variations are essential.

Neural network models exactly meet these requirements. They are continuous, multi-dimensional, and can easily handle nonlinearities in problem behaviors. The neural models are as fast as empirical models and as accurate as PBM's. One of the recent developments that may lead to an efficient usage of EM simulation for RF and microwave CAD is the use of artificial neural networks (ANN) models trained by full wave EM simulators. Fast, accurate and reliable neural network models can be developed from measured or simulated microwave data. Once developed, these neural models can be used in place of computationally intensive physics/EM models of active/passive devices to speed up microwave design.

Reliable CAD solutions need accurate neural models, which in turn, require lots of accurate training data. In cases where a library of 3D accurate models is required to compose certain specific problems, the training of such models requires lots of expensive data from a detailed 3D-EM simulator. If the time required for 3D accurate training data generation increase significantly, the model development process is stretched and it would not be available quickly to the designer. Lots of data could also mean high cost especially if the data has to come from highly CPU-expensive 3D-EM or any other detailed theory based simulator [5].

To improve the speed of generating such models has been a recent challenge in the EM-ANN area. Some of the recent techniques developed addressed these

challenges to some extent by reducing the need for expensive data for efficient model generation. Some of them are as follows. The Automatic Model Generation (AMG) [6] algorithm uses fewer data by avoiding unnecessary samples in smooth sub-regions of the input space and more samples in the non-smooth sub-regions. Knowledge networks such as Knowledge Based Neural Networks (KBNN) [1] and Prior Knowledge Input (PKI) network [7], utilize existing knowledge (e.g., empirical models), thereby reducing the need for expensive training data. Is it possible to combine these advanced concepts/techniques with neural model generation process, and to exploit new potentials in speeding up neural model generation? Addressing this technical gap is one of the motivations of this thesis. Our proposed algorithm Knowledge based Automatic Model Generation (KAMG) is the answer to this question. It not only improves the model generation speed considerably but also automates the entire process.

Recently the neural network approach has achieved a great success in circuit applications and statistical design. This is mainly attributed to the ability of neural network to compute circuit parameters nearly as accurate as those obtained from full-wave EM simulations. It also has demonstrated the capability to generalize and to predict accurate model parameters, for data outside the training set [8]. The neural network can be implemented to model the passive components used in the circuits to nearly the same degree of accuracy as can be afforded by EM simulations. These properties make it suitable for circuit applications and statistical design. Exploring the potential of neural network approaches for circuit applications and statistical design is the other motivation of this thesis work.

1.2 Thesis Objective

The objective of this thesis is to develop methodologies/algorithms to combine the advanced concepts of AMG, KBNN, PKI and SM networks with the neural network structure and exploit these knowledge architectures to generate neural models of highest possible accuracy [5,8]. Following this we proposed a robust Knowledge based Automatic Model Generation (KAMG) technique that is aimed at generating microwave neural models of highest possible accuracy using fewest accurate data. To validate the proposed algorithm, neural models of MOSFET's and embedded passive components were generated. Our proposed KAMG algorithm outperformed the conventional training method both in terms of model accuracy and model generation time. This validates our proposed approach as a better feasible alternative over the conventional methods.

The other objective of this thesis is the EM based optimization and statistical design by neural models and their application to power distribution and decoupling optimization. Neural models trained from EM data of microwave components act as a bridge connecting EM simulation with CAD tools. EM based design of multilayer PCB including power/ground planes, drivers and other components is a challenging endeavor for the circuit designer. In this thesis we have described a new neural network formulation of the power plane, which allows automatic adjustment of component and connection geometry and then provides fast estimation of the EM effects with respect to the different connecting positions for circuit optimization and statistical design based on neural networks. The neural network approach is demonstrated through CAD of an amplifier circuit and a board-level interconnects. In both examples, the power plane connection is placed efficiently with respect to the

voltage source and other components connected to it using EM based optimization to reduce the maximum amount of power source noise and solve various signal integrity problems.

1.3 Organization of the Thesis

In Chapter 2 we will present a brief introduction of neural based modeling and literature review. This includes a brief review of the conventional neural network model development techniques. A brief review of several knowledge architectures used in our formulation is also provided. The concept of circuit applications and statistical design based on neural networks with emphasis on dynamic geometric parameters based design and optimization is also reviewed.

In Chapter 3 we will discuss about the automatic driving of simulation software used for our neural model development. Automatic generation of data by a simulator driver is one of the key aspects, which contributes the most towards model generation by KAMG algorithm and their application to circuit optimization and statistical design. In this chapter we have compared the manual method of data generation with our proposed automatic data generation procedure by a simulator driver. In our proposed method intensive PC work replaces the human labor. Thereafter, a brief discussion is carried out regarding the automatic driving of two-simulation software used in our examples.

In Chapter 4 we propose an efficient KAMG technique, aimed at generation of microwave neural models of highest possible accuracy using fewest accurate data. A completely new concept of neural model generation taking advantage of multiple data

generator is introduced. The proposed technique is validated through the demonstration of MOSFET, embedded passives (resistors, capacitors and spiral inductors) used in multi-layer PCB's. Several tables and graphs are included with the examples. These tables contain the notation of the input and output parameters of the model, range of the input parameters, and accuracy comparison of the neural models developed by several modeling techniques.

In Chapter 5, we will discuss about the EM based optimization and statistical design by neural models and their application to power distribution and decoupling optimization. For the first time we propose a new neural network technique that can model the EM behavior of the passive components and allow dynamic geometric parameter optimization. This chapter describes a new neural network formulation of the power plane, which allows automatic adjustment of component and connection geometry and then provides fast estimation of the EM effects with respect to the different connecting positions for circuit optimization and statistical design. The neural network approach is demonstrated through CAD of an amplifier circuit and a board-level interconnects.

Finally in Chapter 6, conclusions and future perspective of the thesis are described.

Chapter 2

Introduction to Neural based Modeling and Literature Review

2.1 Neural Model Development: Conventional Method

In the recent years a neural network based CAD approach developed by microwave researchers demonstrated the ability of neural networks to learn and model a variety of active and passive microwave components. Neural network possesses the unique advantage to compute circuit parameters very accurately and also has the ability to generalize. Once trained from desired data it can be plugged into any circuit simulation, which speeds up the circuit design optimization and statistical design. For these reasons the neural networks are extensively used to model a variety of microwave components/circuits such as FET [2-4,9-16], transmission lines [17-25], bends [26], vias [7,27], embedded resistors [28], CPW components [29], spiral inductors [8], HBT [30], HEMT [15,31], filters [32-33], amplifiers [34], antennas [35-41], etc.

The process of neural model development consists of various critical steps. The key issues for a neural model development are selection of model input and outputs, data generation, scaling, ANN structure selection, training, and testing as shown in figure 2.1. It often becomes very difficult for the microwave engineers to decide various aspects of model generation such as selection of range of the input parameters, selections of ANN structures, etc.

Figure 2.1: Sequential flow-charts of steps involved in neural model development.

2.1.1 Neural Network Structures

There are a variety of neural network structures, e.g., multi-layer perceptrons (MLP), radial basis function (RBF) networks, wavelet networks, self-organizing maps (SOM), knowledge-based neural networks (KBNN), etc. The model structure for ANN is selected based on the problem behaviors. For example, in the microwave area the MLP is most popularly used. Typically, the MLP neural network consists of an input layer, one or more hidden layer and an output layer, as shown in figure 2.2. The MLP uses switching functions such as sigmoid, arctangent, as hidden neuron activation functions [1][42].

RBF is a type of feedforward neural network structure, which uses radial basis activation functions in their single hidden layer. The RBF network in particular is suitable for modeling localized nonlinear behavior [15][43].

Wavelet neural networks have one input layer, one hidden layer and one output layer. It uses the wavelet functions as hidden neuron activation functions respectively. Since both RBF and wavelet networks have localized basis functions, the wavelet networks with radial wavelet functions can also be considered as an RBF network [44].

SOM is a special type of neural network based on the clustering concept. A SOM contains a set of neurons that are usually arranged into two-dimensional arrays. Each neuron represents a cluster in the sample distribution [43][45].

In KBNN, available microwave knowledge in the form of empirical functions and analytical approximations is embedded into the neural network structure. If empirical formulae are available, KBNN can be used to develop models from fewer amounts of data and speed-up the training process [18].

Figure 2.2: Multilayer Perceptron (MLP) structure. Typically, the network consists of an input layer, one or more hidden layers and an output layer.

2.1.2 Data Generation

Every modeling approach requires data and so does the neural modeling approach. The data can come either from detailed simulation software (simulator) or from a measurement setup. Typical examples include simulators such as *MINIMOS* [46], *Sonnet-Lite* [47], *Ansoft-HFSS* [48], *HSIPCE* [49], *Agilent-ADS* [50], *Ansoft-Ensemble* [51] and measurement equipment such as vector or scalar network analyzers.

A complete process of generation of an accurate neural model requires three sets of data: training data, validation data and test data.

Training Data – These sets of data generally come from EM simulations, as measurement results from actual components can be very costly. As the name suggests this set of data is used to train the neural models by updating the neural network training weights.

Validation Data – These sets of data are generally required to monitor the training quality of the neural network model. This will give an indication to terminate the training.

Test Data – The models generated with the training data are tested with this set of data. These data actually determine the final quality of the models generated.

If the range of the input parameters space $\mathbf{x}$ over which the neural model would be used during microwave design is $\mathbf{x}_{min}$, $\mathbf{x}_{max}$ the validation data and the test data should be in that range but the training data is generally suggested to sample slightly beyond the model utilization range. This generally ensures better performance of the neural model at the boundaries of input parameter space [1][42].

2.1.3 Training the Neural Model

Neural network training is an optimization process in the weight space using optimization-based training algorithms such as Back-propagation [43][52], Conjugate gradient [53], Levenberg Marquardt [54], quasi-Newton [55-56], Genetic algorithm [31], Huber-quasi-Newton [8], Sparse Training algorithm [8] etc.

Let $\mathbf{x}$ be an n -vector containing the external inputs (stimuli) to a microwave model and $\mathbf{y}$ be an m -vector containing the outputs of the model. The detailed or theoretical based relationship between $\mathbf{x}$ and $\mathbf{y}$ can be represented as $\mathbf{y} = \mathbf{g}(\mathbf{x})$ where $\mathbf{g}$ can represent detailed semi-conductor physics relationship in the case of active components and EM relationship in the case of passive components. The neural network structure to be trained will have n input neurons and m output neurons. Let $\mathbf{w}$ be the vector containing all the weight parameters inside the neural network. The neural network then can be denoted mathematically by $\tilde{\mathbf{y}} = \mathbf{h}(\mathbf{x}, \mathbf{w})$, where $\mathbf{h}$ is a neural network model, and $\mathbf{w}$ is the neural model weight vector. During neural network training these weights are adjusted such that the error between the original output and the neural network output is minimized. One of the major concern during training is overlearning or underlearning of the neural models. Whenever the neural network can memorize the training data but cannot generalize well then it is called overlearning [57]. The main reasons attributed to this are either many hidden neurons or insufficient training data. Too many hidden neurons add too much freedom in $\mathbf{x}$ - $\mathbf{y}$ relationship. This situation can be avoided either by reducing certain number of hidden neurons or by adding more samples to the training data. Underlearning on the other hand is such a situation in which the neural network has difficulties to learn the training data itself. The main reasons attributed to this cause are inadequate number of

hidden neurons, deficient in training or training procedure stuck at a local minimum. The solutions to these problems are adding more hidden neurons or continuing training process or perturbing the current training process to escape from the local minimum.

2.1.4 Testing the Neural Model

This is the final step of the model generation. Actually this step is performed after the models are generated. This step ensures the quality of the model produced. Depending upon the quality, the model needs to be re-trained if the testing error is too high. An independent set of data known as the test data is required for this purpose. A quantity δ_{pk} for the error measurement is defined as

$$\delta_{pk} = \frac{\tilde{y}_k(x_p, w) - d_{pk}}{d_{k, \max} - d_{k, \min}}, \quad k=1, \dots, N, \quad p \in T_E \quad (2.1)$$

where d_{pk} is the output from the simulator at instant p and $d_{k, \max}$ and $d_{k, \min}$ represents the minimum and maximum value of the simulation.

A quality criterion based on the q^{th} norm is then defined as

$$Mq = \left[\sum_{p \in T_E} \sum_{k=1}^{N_y} |\delta_{pk}|^q \right]^{\frac{1}{q}} \quad (2.2)$$

When $q = 1$, the average test error can be calculated directly from M_1 as

$$\text{Average Test Error} = \frac{M_1}{N_{T_E} N_y} \quad (2.3)$$

where N_{T_E} is the number of samples in test data set and N_y is the number of neural model outputs. When $q = 2$, the q^{th} norm is the Euclidean distance between neural model prediction and test data. When $q = \text{infinite}$, the q^{th} norm measure is the maximum test error, which is also called as the worst-case error among entire test data and all model outputs.

2.2 Knowledge Architecture

The drive in the electronics industry for manufacturability-driven design and time-to-market demands powerful and efficient CAD techniques. Recently, the concept of neural networks and their application to microwave modeling and design [6,58-59] gained popularity. Reliable CAD solutions need accurate neural models, which in turn, require lots of accurate training data. Lots of training data generation often takes a lot of human as well as CPU time. As a result, the model development process is stretched and model is not available quickly to the designer. Lots of data could also mean high cost especially if the data has to come from highly CPU-expensive 3D-EM or any other detailed theory based simulators. Several knowledge neural network architectures such as Difference Method (DM) [29], Knowledge Based Neural Networks (KBNN) [1,18], Prior Knowledge Input (PKI) [7] and Space Mapped Neural Networks (SMNN) [26,59,60] have been proposed which aims at generation of neural models of high accuracy with least use of the expensive 3D-EM training data.

The DM architecture is based on the concept that difference between the coarse data and the fine data is simpler to model than the fine data itself. The KBNN architecture incorporates existing microwave knowledge in the form of hidden neurons. The PKI architecture links the information from coarse data with the fine data. It is based on the fact that the relationship between the coarse data and the fine data is simpler to model than the fine data itself. The SM architecture is based on the idea that the coarse data can generate the same response as the fine data but with a different value of the input. In other words for a given original fine input, a corresponding coarse input can be achieved which will give the same response as the fine output. These structures are explained briefly in Chapter 4.

2.3 Circuit Application of Neural Model

Artificial neural networks recently gained attention as fast and flexible vehicle to microwave circuit application and statistical design. This is mainly attributed to the ability of neural network to compute circuit parameters nearly as accurate as those obtained from full-wave EM simulations. Neural models trained from EM data of microwave components act as a bridge connecting EM simulation with CAD tools. The drive in the microwave industry to meet the demands of high manufacturability and faster design cycles created a need for efficient statistical design techniques. Statistical analysis and yield optimization that take into account manufacturing tolerances, model uncertainties, variations in process parameters etc., are widely used in circuit design methodologies. So, they are extensively used in industry now a days to improve the yield of microwave circuits.

As signal frequency is increased, non-ideal EM effects such as discontinuities and radiation losses becomes more prominent in circuit design which results in many signal integrity problems in switching circuits and high frequency analog circuit. The Delta-I noise also called, as simultaneous switching noise or power/ground bounce is one of the major factors concerning the efficiency of the switching circuit. It is mainly caused due to the inductance of the power supply distribution system of the machine's package and its inability to change the supply of the current at the same rate, as the circuits require and tends to get worse with faster device switching speed, more input/outputs and higher clock frequency. The noise consists of voltage spikes appearing at the power supply terminals of the chip and also propagates through the quiet drivers on this chip and appears at the input of the corresponding receiver circuits [61-62]. The cause of the

voltage fluctuation in the power/ground reference plane is the transient current drawn by a large number of drivers switching simultaneously. The noise could degrade the buffer output waveforms, falsely set the latches, and influence the quiet chips due to the coupling between the power distribution and the signal distribution. The impedance of a power and ground system typically increases with frequency, and can reach quite significant magnitudes if there are significant resonances inside the package. Under these conditions minimization of mesh inductance effect in power/ground path and integration of de-coupling capacitance into substrate will improve the circuit performance. Adding decoupling capacitors ensures the system reliability by suppressing the voltage fluctuations on the power/ground planes. The decoupling capacitor provides AC ground for the noise and maintains the dc supply voltage [63]. In order to function the circuits properly, evaluation of the number of decoupling capacitor and their efficient placement in the power/ground plane is absolutely essential. By optimal placement of individual components in the power plane, the power supply noise can be reduced significantly. As a result various signal integrity problems, like time-delay, crosstalk can be overcome. The reduction of power supply noise also increases the power gain of the circuit.

Chapter 3

Automatic Driving of Data Generators: A Concept

Every modeling approach requires data and so does the neural modeling approach. Neural network models are developed from measured/simulated microwave data through a process known as training. The development process involves several subtasks like the data generation, neural-network selection, training, and validation. Conventionally, these subtasks are manually carried out in a sequential manner independent of one another. This approach requires intensive human effort. There is a definite need for automation of the neural model development process. One of the most critical steps to make the entire process of model generation automated is the automatic generation of data by the simulators. Typical example includes simulator such as *Sonnet-Lite*, *Ansoft-HFSS*, *HSIPCE*, *Agilent-ADS*, etc. In this chapter we will discuss about the automatic driving of some of the simulation software by the *Neuro-Modeler* [64] simulator driver. Automatic generation of data by simulator driver is one of the key aspects, which contributes the most towards model generation by KAMG algorithm and their application to circuit optimization and statistical design.

3.1 Automation of Data Generators

AMG is a robust algorithm aimed at automating the neural network based RF/microwave model development process. In our proposed algorithm KAMG we have

utilized the stage-wise training and adaptive data sampling of AMG to automate neural model generation. So, initially we start with a zero amount of training data and then proceed in a stage-wise manner to generate the neural model [6]. The algorithm dynamically generates the data for training and testing by automatically driving the simulation tools e.g., *Sonnet-Lite*, *Ansoft-HFSS*, etc. In the sections to follow we will discuss about the automatic driving of the simulation software used to generate neural models for our examples.

There are several advantages by making the entire process of data generation automatic. The main advantage is that it reduces manual labor thereby reducing the time for data generation and any possible chances of human error. However, unlike the manual method, where all the tasks are carried out manually and sequentially but independent of each other, a successful automation technique requires addressing multiple complicated and interdependent challenges. A comparison between the conventional method of model generation and the proposed automated technique is shown in figure 3.1.

In the manual method after every simulation the physical/geometrical or the electrical parameters are changed manually. Then the simulation is restarted again to obtain output for the new set of changes. This assignment becomes highly tedious and error prone when extensive data are generated. But in the proposed method once data generation begins no human effort is necessary to change any parameters. The input data file is composed of the original input vectors and their range is fed to the *Neuro-Modeler* simulator driver. In the next step we run the EM simulator. After solving and exporting the results, the *Neuro-Modeler* simulator driver automatically changes one parameter in the input data file. This entire process is repeated until results for all sets of inputs are obtained.

Figure 3.1: Comparison between the conventional and the proposed method of data generation. In the proposed method intensive PC work replaces manual input.

3.1.1 Automatic Driving of *Sonnet-Lite*

Sonnet-Lite is a 2.5D EM simulator. The *Sonnet-Lite* version has a limited memory but it is very fast. The cell size is fixed and it has a discontinuous geometry. In the adjoined figure 3.2 the automatic driving of *Sonnet-Lite* is explained. It require two kinds of files namely the *geo* files and the *net* files. In the *geo* file the geometry of the structure to be simulated is defined. In the *net* file the parameter required to be swept is included.

Figure 3.2: Automatic Driving of *Sonnet-Lite*.

The first step is to create the input data file. This file contains all the geometrical and electrical parameters, which are changed during simulation. After the completion of one simulation, the *Neuro-Modeler* simulator driver automatically changes the value of one of the variables. Following this a new simulation is carried out again with the changed parameter to obtain a new set of outputs.

In the next step, the geo file is created. In the geo file we need to define the units of measurements (e.g. *mils*, *mm* etc), electrical/physical nature of the materials to be used (e.g. *lossless metal*, *lossy metal*, *resistivity*, *permittivity*, *height of substrate*, etc), the boundary conditions (e.g. *number of ports*, *location of ports*, etc) and the physical location of each structure used in our formulation (in the form of *x-y* coordinate). All these parameters are defined in the geo file in the format shown below.

```

**Units of Measurement**
LEN mils 2.54e-005
**Electrical Parameter**
TON 0 Lossless
BON 0 Lossless
RES 600 0 0 0
REN 1 Metall
** Physical Parameter**
BOX 1 64 48 128 96 20 0
100 1 1 0.0 0 0 2
10 8.5 1 0.0 0 0 2
** Boundary Condition**
POR 3 3 0 1 50 0 0 0 0 0
** Physical Location **
NUM 1
0 5 -1 N 9 1 1 100 100 0 0 0
8 12
12 12
12 36
8 36
8 12
END

```

Thereafter the net file is created which includes information about the parameters to be swept. In the present version of *Sonnet-Lite* used we can sweep only the frequency, however in higher versions any geometrical or electrical parameters can also be swept. We also need to define our desired EM outputs like the S-parameter, Z-parameter or Y-parameter. The format of the net file is shown below.

```
**Frequency Sweep**
!< FTYP NET
! Analyze comp
DIM
FREQ    GHZ
CKT
OPT=vd      CTL=internal
DEF2P 1 2 ATTN
FILEOUT
comp.rsp    S RI R 50
FREQ
SWEEP 0 10 0.1
```

After this step the *Sonnet-Lite* solves the structure and generates the output in the form of S, Y or Z parameters as specified in the net file. It generates a separate file known as the *.rsp file where all the results of simulation are stored. We can retrieve our desired output from this file very easily with the help of a simple C-program.

It is also very convenient to generate the geo and net files repetitively with the help of the C-program as the fixed coordinate describing the physical location of the structure can be replaced by a variable and can be changed eventually during repetitive simulation. A complete C-program for the generation of a capacitor model in *Sonnet-Lite* is attached in Appendix A.

3.1.2 Automatic Driving of *Ansoft-HFSS*

Ansoft-HFSS is a 3D EM simulator. There are no memory limitations for this software but it may become very slow depending upon the accuracy criterion. The cell size is automatic and it has continuous geometry. To automate the entire process of data generation using *HFSS* as the EM simulator we have to create a macro. This macro contains all the *HFSS* executables and by driving it we can get the appropriate outputs for all the sets of input data. In the adjoined figure 3.3 the automatic driving of *Ansoft-HFSS* is explained.

Figure 3.3: Automatic Driving of *Ansoft-HFSS*.

The development of macro is the most critical step involved in the automatic driving of *HFSS*. The macro file consists of six consecutive steps. The steps are shown in figure 3.4. A macro for the generation of a resistor model in *Ansoft-HFSS* is attached in Appendix B.

Figure 3.4: Steps for generation of *Ansoft-HFSS* macro.

Step 1: Define Structure Geometry

In this step the geometry of the structure to be simulated are defined. The modeler consists of two parts. The first part contains the entire input vector. In the second part the geometrical structure of the element is defined. When the *Neuro-Modeler* simulator driver changes any input variable, it automatically updates the change with in the macro.

Step 2: Setup Materials

In this step materials are assigned for all the 3D structures. Various properties of the materials like conductivity, permeability, permittivity, resistivity etc. are defined.

Step 3: Setup Boundary

In this step boundary condition for all the 2D objects are specified. After assigning material properties, the ports and the boundary condition must be defined. This specifies the excitation signals entering the structure, the behavior of electric and magnetic fields at various surfaces in the model, and any special surface characteristics.

Step 4: Setup Solution

In this step the main goal is to set up the solution parameters for the problems. We need to specify the frequency range of operation and also define our desired EM outputs like the S-parameter, Z-parameter or Y-parameter in this step.

Step 5: Simulate

In this step the effect is the execution of the simulator. The EM solver solves for the specified output parameters.

Step 6: Results

In this step the results of the solution are analyzed. *Ansoft-HFSS* provides us with three post processors to analyze our results. They are

Fields:	This allows displaying the contour, shaded, vector, value versus distance and animated plots of fields.
Matrix Data:	This allows viewing the matrices computed for the S-parameters, impedance, and propagation constants.
Matrix Plot:	This allows plotting of various parameters versus time or frequency.

Retrieve Data:

Ansoft-HFSS solves for the specified output parameter and stores the results of simulation in a separate file known as the *.szg. In this case also we can retrieve our desired output from this file very easily with the help of a simple C-program. A complete macro for the generation of a resistor in *Ansoft-HFSS* simulator is shown in Appendix B.

Chapter 4

A Knowledge Based Approach for Neural Model Generation

In this chapter, we will explain the robust KAMG technique that takes advantage of multiple data generators. The proposed technique is developed by the combination of three separate powerful concepts namely automatic model generation, knowledge based neural networks and space mapping. Two types of data generators are used for this purpose namely fine data generators which is accurate and slow (e.g., CPU intensive 3D-EM simulators) and coarse data generators which is approximate and fast (e.g. inexpensive 2D/2.5D-EM simulators). Our objective is to generate accurate neural models by extensive use of the coarse data together with fewest fine data. All the coarse data generator and fine data generator are driven automatically with the help of simulator driver explained in Chapter 3.

Let $\mathbf{x}$ be an n -vector containing the external inputs to the model and $\mathbf{y}$ be an m -vector containing the output responses of the model. As mentioned earlier the relationship between the input vector $\mathbf{x}$ and the output vector $\mathbf{y}$ can be represented by $\mathbf{y} = \mathbf{g}(\mathbf{x})$ where $\mathbf{g}$ represents theoretical physics/EM relationship for any components. The neural network can be mathematically represented by $\tilde{\mathbf{y}} = \mathbf{h}(\mathbf{x}, \mathbf{w})$, where $\mathbf{h}$ is a neural

network trained to learn the $\mathbf{x}$ - $\mathbf{y}$ relationship, and $\mathbf{w}$ is the neural model weight vector. During neural network training, $\mathbf{w}$ is adjusted such that the error between $\mathbf{y}$ and $\tilde{\mathbf{y}}$ is minimized.

Neural network model $\mathbf{h}$ can be trained to learn $\mathbf{g}$ from a set of $\mathbf{x}$ - $\mathbf{y}$ sample pairs. We introduce a new terminology for “ $\mathbf{g}$ ” and refer to it as “data generator”. For example, $\mathbf{g}$ could be 3D-EM *Ansoft-HFSS* simulator, physics-based *MINIMOS* simulator. For a given input $\mathbf{x}$, $\mathbf{g}$ can be used to generate a corresponding output $\mathbf{y}$. We define coarse and fine data generators as $\mathbf{g}_c$ and $\mathbf{g}_f$ respectively. For example, $\mathbf{x}$ could represent any physical parameters (length, width, resistivity, permittivity, etc) of embedded passives like resistors, capacitors, etc; $\mathbf{g}_c(\mathbf{x})$ and $\mathbf{g}_f(\mathbf{x})$ could represent the S -parameters computed from 2D and 3D-EM respectively. The objective of KAMG is to produce a neural model $\mathbf{h}_f$ that accurately fits the fine data from $\mathbf{g}_f(\mathbf{x})$, by extensive use of training data from $\mathbf{g}_c(\mathbf{x})$ but minimally using data from $\mathbf{g}_f(\mathbf{x})$. We define $\mathbf{h}_c$, $\mathbf{h}_s$, and $\mathbf{h}_f$ as coarse, sub, and overall (fine) neural models.

The proposed KAMG technique is implemented following three major steps.

STEP 1: In this step a coarse model is developed using coarse data generator (e.g. 2D EM simulator) extensively.

STEP 2: In this step a sub neural model is initialized by simple training. The training differs for different neural networks. Besides an initial overall neural network is constructed using coarse and sub neural model.

STEP 3: This is the final stage of training. In this stage the overall neural network is refined (i.e. further trained) using fine data generator (e.g. 3D EM simulator).

4.1 Coarse Model Generation

In this phase we generate a coarse neural model $\tilde{y}_c = h_c(x, w_c)$, where w_c denotes weight parameters of the coarse model. The training objective here is to minimize the difference between coarse data generator outputs and coarse neural model outputs, i.e.,

$$\min_{w_c} \sum_{x \in L_c} \|g_c(x) - h_c(x, w_c)\| \quad (4.1)$$

by adjusting w_c , and adaptively sampling the set of coarse training data L_c . Coarse neural model h_c is trained using g_c extensively, because it is inexpensive. This training ensures that neural model h_c captures the coarse x - y relationship available in the form of g_c .

4.2 Sub-Model Training and Overall Model Initialization

The objective in the second phase is to systematically construct a framework that facilitates reinforced neural network learning of fine behaviors from coarse information $h_c(x, w_c)$ together with fine data from g_f . The KAMG framework allows the use of a variety of knowledge neural network architectures such as DM, KBNN, PKI and SMNN. Depending upon the knowledge architecture a sub neural model $\tilde{y}_s = h_s(x, w_s)$ is initialized through a simple training of h_s re-using L_c from the first phase. KAMG then utilizes h_c and h_s to construct an initial overall neural model as

$$\tilde{y}_f = h_f(x, w_f) = h_f(x, w_c, w_s) \quad (4.2)$$

The initial overall neural model will have a fine accuracy as good as that of the coarse model. In the following section a brief explanation about the sub-model training and overall model initialization specific to each architecture, are presented.

4.2.1 Difference Method

In DM h_s models the difference between g_c and g_f . The best possible initialization in such a structure would thus be to make h_s to give an output 0 for all inputs, as we have no fine data available until this phase. By definition, the difference method outputs are computed as a simple summation of outputs from h_c and h_s . As such, initial neural model is given by,

$$h_f(x, w_c, w_s) = h_c(x, w_c) + h_s(x, w_s) \quad (4.3)$$

The proposed DM architecture is illustrated in figure 4.1.

4.2.2 Prior Knowledge Input

PKI architecture (Figure 4.2) is based on the fact that the relationship between (x, g_c) and g_f , could be simple than x - g_f relationship. In PKI method, the coarse neural model h_c developed in step 1 is used at the lower level of hierarchy. Original inputs x and outputs $\tilde{y}_c$ from h_c become inputs to the sub neural model h_s at the higher level, which is forced to learn the relation between (x, g_c) and g_c itself. In other words, h_s is not only a function of x but also a function of h_c . Initial neural model for PKI is given by,

$$h_f(x, w_c, w_s) = h_s(x, h_c(x, w_c), w_s) \quad (4.4)$$

and our initial training of h_s ensures that the initial overall model h_f will have a fine accuracy very close to that of h_c .

4.2.3 Space Mapping

In space mapping [60] it is considered that for a given original input $\mathbf{x}_f$, a corresponding $\mathbf{x}_c$ can be found such that $\mathbf{g}_f(\mathbf{x}_f) = \mathbf{g}_c(\mathbf{x}_c)$. In the SMNN architecture (Figure 4.3) $\mathbf{h}_s$ is used at the lower level of the hierarchy. Original input $\mathbf{x}_f$ are fed to the network $\mathbf{h}_s$, and output $\mathbf{x}_c$ from $\mathbf{h}_s$ is inputted to $\mathbf{h}_c$ at the higher level. In our initial training since $\mathbf{h}_s$ is trained to learn $\mathbf{x}_c$ - $\mathbf{x}_f$ relationship, the construction of the initial network can be carried out as

$$\mathbf{h}_f(\mathbf{x}, \mathbf{w}_c, \mathbf{w}_s) = \mathbf{h}_c(\mathbf{h}_s(\mathbf{x}, \mathbf{w}_s), \mathbf{w}_c) \quad (4.5)$$

Initial training of $\mathbf{h}_s$ ensures that the initial overall model $\mathbf{h}_f$ will have a fine accuracy very close to that of $\mathbf{h}_c$ to start with before using any fine data.

4.2.4 Knowledge Based Neural Network

In this approach it is believed that by multiple use of $\mathbf{h}_c$ with a different weightage we can easily achieve $\mathbf{h}_f$. The KBNN architecture [18] shown in figure 4.4 is exploited, and coarse neural model $\mathbf{h}_c$ generated in the first phase is incorporated into several knowledge layer neurons. During training, KBNN parameters will adjust so as to divide the input space into a number of sub regions. The weightage of each coarse model in each sub-region is available. As such, the best possible initialization in such a structure would thus be to implement

$$\mathbf{h}_f(\mathbf{x}, \mathbf{w}_c, \mathbf{w}_s) = \mathbf{h}_s(\mathbf{h}_c(\mathbf{x}, \mathbf{w}_c), \mathbf{w}_s) \otimes \mathbf{h}_r(\mathbf{x}, \mathbf{w}_s) \quad (4.6)$$

Here, $\mathbf{h}_r(\mathbf{x}, \mathbf{w}_s)$ is region network that defines boundaries in $\mathbf{x}$ -space and $\otimes$ is a gating operator [1].

Figure 4.1: Schematic Diagram of the proposed Difference Method.

Figure 4.2: Schematic of the proposed PKI Method.

Figure 4.3: Schematic of the proposed SM Method.

Figure 4.4: Schematic of the proposed KBNN Method.

4.3 Final Training:

In the final phase, initial overall neural model h_f is further trained (refined) using fine data from g_f . The initial version of the fine (overall) neural network h_f is further trained using fewest fine data from g_f , to capture the finer microwave behaviors missed in the coarse neural network model h_c or in coarse data generator g_c . The training objective is to minimize the difference between fine data generator outputs and overall neural model outputs, while keeping the coarse portion of h_f fixed, i.e.,

$$\min_{w_s} \sum_{k \in L_f} \|g_f(x) - h_f(x, w_c, w_s)\|, \quad (4.7)$$

by adjusting w_s , while keeping w_c fixed. L_f represents the set of fine training data.

4.4 Examples by KAMG Approach

In this section the advantages of the proposed KAMG technique are demonstrated through examples of MOSFET and embedded passive components used in multilayer PCB's. In the examples under consideration, comparison of accuracy of neural model generated by KAMG structure and the conventional method and AMG are presented. Besides this a table is also provided for each example, which contains all the notations for input and output parameters and the effective range of the input parameters.

Example 1

A 2-Input MOSFET Neural Model Development Using Circuit Based and Physics Based Data Generators.

This example illustrates the development of a 2-input MOSFET neural model (figure 4.5 and figure 4.6). Input vector x contains drain voltage (V_D) and gate voltage (V_G). Drain current (I_D) is the neural model output y . An approximate and fast equivalent circuit model *HSPICE* is used as coarse data generator g_c , while a CPU-intensive (slow) but accurate physics-based *MINIMOS* simulator is used as fine data generator g_f . Table 4.1 represents the model input and output parameters and its effective range.

The KAMG technique is applied to various knowledge architectures. In the first phase, a coarse neural model h_c is generated by extensive use of coarse training data from g_c . A total of 84 coarse training data are used since coarse data is inexpensive. Neural model h_c predicts MOSFET drain current with an average validation error of 3.80%. In the second phase, sub neural model h_s is developed and initial fine (overall) neural network is constructed. It is ensured that the initial fine (overall) neural network model is as good as h_c , i.e., average validation error of initial fine neural model is 3.80%. Following this, the initial fine (overall) network is refined using fewest fine training data from g_f through automatic sampling and stage-wise training.

As can be seen in figure 4.7, all the neural models from KAMG achieved better accuracy with fewer expensive data, as compared to neural model from AMG. For a given model accuracy, the KAMG requires fewer fine data than AMG and conventional grid-based manual neural modeling approach, as shown in Table 4.2.

Table 4.3 shows that the proposed KAMG models outperform corresponding conventional knowledge models, when only a few accurate data are available.

Figure 4.5: 3-D layout of the MOSFET modeled using proposed KAMG technique.

Figure 4.6: Neural Model of 2-input MOSFET model.

Figure. 4.7: Accuracy comparison of 2-input MOSFET neural model generated by proposed KAMG and existing AMG techniques. KAMG achieved better accuracy with fewer fine data.

TABLE 4.1. NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 2-INPUT MOSFET MODELS AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
GATE VOLTAGE	V_G	0V – 2V
DRAIN VOLTAGE	V_D	-2V – 0V
DRAIN CURRENT	I_D	N/A

TABLE 4.2. COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE 2-INPUT MOSFET MODELS WITH 0.50% TEST ERROR.

Neural Modeling Technique	No. of Fine Data Used
Conventional training	66
AMG	49
Proposed KAMG-DM	32
Proposed KAMG-KBNN	23
Proposed KAMG-PKI	25
Proposed KAMG-SMNN	14

TABLE 4.3. ACCURACY COMPARISON BETWEEN 2-INPUT MOSFET NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 23 FINE DATA.

Neural Modeling Technique	Test Error
Conventional training	3.82%
AMG	1.00%
Proposed KAMG-DM	0.65%
Proposed KAMG-KBNN	0.48%
Proposed KAMG-PKI	0.52%
Proposed KAMG-SMNN	0.25%

Example 2

A 5-Input MOSFET Neural Model Development:

As the number of the inputs increases the responses of the circuit becomes more complex. It then becomes more difficult and challenging to model such structures. In order to prove the robustness of our knowledge-based approach we developed a more complex model by taking into consideration five input parameters for the MOSFET (Figure 4.5). Figure 4.8 represents the neural network model of 5-input MOSFET. The input vector $\mathbf{x}$ consists of the gate voltage (V_G), the drain voltage (V_D), the thickness of oxide (T_{ox}), effective channel length (L) and effective channel width (W). Drain current is the output y of this MOSFET model. Similar to the previous example *HSPICE* is used as coarse data generator g_c , while *MINIMOS* is used as fine data generator g_f .

Table 4.4 represents the model input and output parameters and their effective range.

In this example, a total of 3265 coarse training data are used to generate a coarse neural model h_c in the first phase. Neural model h_c predicts MOSFET drain current with an average validation error of 4.50%. Thereafter a sub neural model h_s is developed and initial fine (overall) neural network is constructed with an initial fine (overall) neural network model as good as h_c , i.e., average validation error of initial fine neural model is 4.50%. Figure 4.9 illustrates that our neural models from KAMG achieved much better accuracy with fewer expensive data. Table 4.5 and Table 4.6 reveals that the KAMG models achieve better accuracy with less expensive data and less computational time.

Figure 4.8: Neural Model of the 5-input MOSFET model.

Figure 4.9: Accuracy comparison of the 5-input MOSFET neural models generated by proposed KAMG and existing AMG techniques.

TABLE 4.4. NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 5-INPUT MOSFET MODELS AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
GATE VOLTAGE	V_G	0v – 2v
DRAIN VOLTAGE	V_D	-2v – 0v
THICKNESS OF OXIDE	T_{ox}	50nm – 100nm
CHANNEL LENGTH	L	0.25 μ m – 1.0 μ m
CHANNEL WIDTH	W	5 μ m – 25 μ m
DRAIN CURRENT	I_D	N/A

TABLE 4.5: COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE 5-INPUT MOSFET MODELS WITH 0.50% TEST ERROR.

Neural Modeling Technique	No. of Fine Data Used
Conventional training	1495
AMG	1190
Proposed KAMG-DM	724
Proposed KAMG-KBNN	562
Proposed KAMG-PKI	645
Proposed KAMG-SMNN	454

TABLE 4.6: ACCURACY COMPARISON BETWEEN 5-INPUT MOSFET NEURAL MODELS
DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 454 FINE DATA.

Neural Modeling Technique	Test Error
Conventional training	3.24%
AMG	2.52%
Proposed KAMG-DM	0.95%
Proposed KAMG-KBNN	0.65%
Proposed KAMG-PKI	0.60%
Proposed KAMG-SMNN	0.48%

Example 3

Embedded Resistor Neural Model Development Using Planar-EM and 3D-EM Data Generators

Fast and accurate modeling of 3D-EM behaviors of embedded passives is important for multi-layer PCB design. In this example we have modeled the embedded resistor shown in figure 4.10. Figure 4.11 represents the neural model of the Resistor. The input vector $\mathbf{x}$ contains resistor length (L) and signal frequency (f), and output vector $\mathbf{y}$ contains real and imaginary parts of S-parameters RS_{11} , IS_{11} , RS_{21} , and IS_{21} . Resistor width is 10 mils and resistivity is 100 ohm-cm. The coarse data generator g_c used for the embedded resistor is 2.5D Planar EM simulator *Sonnet-Lite*. The fine data generator

g_f used for the example is 3D-EM simulator *Ansoft-HFSS*. Table 4.7 summarizes the model input and output parameters and their effective ranges.

In figure 4.12 the accuracy comparison between the different neural models generated by our proposed KAMG and the conventional method is illustrated. In figure 4.13 the comparison between real and imaginary parts of the S_{11} parameter of embedded resistor obtained from proposed KAMG and EM simulators is illustrated. Table 4.8 demonstrates that the proposed KAMG technique uses fewer fine data to achieve a given neural model accuracy, as compared to both AMG without knowledge and conventional training. In a worst situation, i.e., when fine data is unavailable to the user, neural models from KAMG are at least as accurate as the coarse neural model. For the embedded resistor, all the KAMG neural models exhibit 7.02% average test error, after training with 220 coarse data, and before training with any fine data.

Figure 4.10: 3-D layout of Embedded Resistor modeled using proposed KAMG technique.

Figure 4.11: Neural Model of Embedded Resistor.

Figure 4.12: Accuracy comparison of embedded resistor neural models generated by proposed KAMG and existing AMG.

(a)

(b)

Figure 4.13: Comparison of real and imaginary parts of the S_{11} parameter of embedded resistor obtained from proposed KAMG (—) and from EM simulators (o).

TABLE 4.7: NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED RESISTOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
LENGTH	L	5MILS – 20MILS
FREQUENCY	f	1GHZ – 20GHZ
REAL S11	RS11	N/A
IMAGINARY S11	IS11	N/A
REAL S21	RS21	N/A
IMAGINARY S21	IS21	N/A

TABLE 4.8: ACCURACY COMPARISON BETWEEN RESISTOR NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 18 FINE DATA.

Neural Modeling Technique	Test Error
Conventional training	2.85%
AMG	1.65%
Proposed KAMG-DM	1.02%
Proposed KAMG-KBNN	0.65%
Proposed KAMG-PKI	0.42%
Proposed KAMG-SMNN	0.95%

Example 4

Embedded Capacitor Neural Model Development Using Planar-EM and 3D-EM Data Generators

In this example we have modeled an embedded capacitor (Figure. 4.14) used in multi-layer PCBs. The coarse and the fine data simulators are identical to that of the resistor. The input vector x includes capacitor length (L) and signal frequency (f). Real and imaginary parts of S-parameters RS_{11} , IS_{11} , RS_{21} , and IS_{21} are the model outputs y . The neural model representing the embedded capacitor is shown in figure 4.15. Table 4.9 represents the model input and output parameters and their effective range.

The first and the second stages of training using 180 coarse data results in initial (fine) neural models with an average validation error of 3.02%. Figure 4.16 demonstrates that the proposed KAMG technique yields neural models with better accuracies as compared to AMG, when same amounts of fine data are used. As a result, data generation time (CPU) is significantly reduced as can be seen in Table 4.10.

Figure 4.14: 3-D layout of Embedded Capacitor modeled using proposed KAMG technique.

Figure 4.15: Neural Model of Embedded Capacitor.

TABLE 4.9: NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED CAPACITOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
LENGTH	L	5MILS – 20MILS
FREQUENCY	f	1GHZ – 20GHZ
REAL S11	RS11	N/A
IMAGINARY S11	IS11	N/A
REAL S21	RS21	N/A
IMAGINARY S21	IS21	N/A

Figure 4.16 Accuracy comparison of embedded capacitor neural models generated by proposed KAMG and existing AMG techniques.

TABLE 4.10: COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE CAPACITOR MODELS WITH 1% TEST ERROR.

Neural Modeling Technique	No. of Fine Data	CPU for Data Generation
Conventional training	125	625 min
AMG (without knowledge)	96	480 min
Proposed KAMG-DM	48	240 min
Proposed KAMG-KBNN	14	70 min
Proposed KAMG-PKI	24	120 min
Proposed KAMG-SMNN	23	115 min

Example 5:

2-Loop Spiral Inductor Neural Model Development Using Planar-EM and 3D-EM Data Generators:

Integration of passive components in wireless systems remains a challenging endeavor. Passives such as inductors on a chip are the main limitation for a successful highly integrated microwave system design [65]. Although full-wave EM simulators can be used to analyze spiral inductors, they are computationally prohibitive especially if the inductor's geometrical/physical parameters need to be repetitively changed during CAD and they are also computationally very expensive. In fact, the value of an inductance is often critical for a good design of compact high quality factor (Q) filters, matching networks, bias chokes, resonator elements or circuit components, but there is no simple equation to determine it accurately [65-66].

In this example, a 2-loop spiral inductor (figure 4.17) is modeled without using too much EM-based expensive training data. A bridge drawn at 0.1 mills on the spiral pattern is used as the inductor connector pass. In figure 4.18 the neural model of the 2-loop spiral inductor is shown. Input $\mathbf{x}$ contains width (W), space (S), dielectric constant (ϵ_r) and frequency (f). Real and imaginary parts of S-parameters RS_{11} , IS_{11} , RS_{21} , and IS_{21} are the model outputs $\mathbf{y}$. Table 4.11 represents the model input and output parameters and their effective range. 2.5D Planar EM simulator *Sonnet-Lite* is used as coarse data generator $\mathbf{g}_c$ and 3D-EM simulator *Ansoft-HFSS* is used as fine data generator $\mathbf{g}_f$. Extensive coarse data (540 samples) are used in first and second stages of KAMG algorithm.

The KAMG technique is applied to various knowledge architectures. Figure 4.19 reveals all neural models from KAMG achieved better accuracy's with fewer expensive

data, as compared to neural model from AMG. For a given model accuracy, the KAMG requires fewer fine data and less CPU time than AMG and conventional manual neural modeling approach, as shown in Table 4.12 and Table 4.13.

Figure 4.17: 3-D layout of the 2-loop spiral inductor.

Figure 4.18: Neural Model of 2-loop Spiral Inductor.

TABLE: 4.11: NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 2-LOOP SPIRAL

INDUCTOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
WIDTH	W	8 MILS – 10 MILS
SPACE	S	2 MILS – 4 MILS
PERMITTIVITY	ϵ_r	6 – 10
FREQUENCY	f	1 GHZ – 20 GHZ
REAL S11	RS11	N/A
IMAGINARY S11	IS11	N/A
REAL S21	RS21	N/A
IMAGINARY S21	IS21	N/A

Figure 4.19: Accuracy comparison of 2-loop Spiral Inductor neural models generated by proposed KAMG and existing AMG techniques.

TABLE 4.12: ACCURACY COMPARISON BETWEEN 2-LOOP SPIRAL INDUCTOR NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 386 FINE DATA.

Neural Modeling Technique	Test Error
Conventional training	6.23%
AMG	2.34%
Proposed KAMG-DM	0.96%
Proposed KAMG-KBNN	1.12%
Proposed KAMG-PKI	0.85%
Proposed KAMG-SMNN	0.99%

TABLE 4.13 COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE 2-LOOP SPIRAL INDUCTOR MODELS WITH 1% TEST ERROR.

Neural Modeling Technique	No. of Fine Data	CPU for Data Generation
Conventional training	1550	10850 min
AMG	687	4809 min
Proposed KAMG-DM	386	2702 min
Proposed KAMG-KBNN	450	3150 min
Proposed KAMG-PKI	321	2247 min
Proposed KAMG-SMNN	386	2702 min

Example 6

8-Loop Spiral Inductor Neural Model Development Using Planar-EM and 3D-EM Data Generators:

In this example we modeled an 8-loop inductor to demonstrate the robustness of our approach. As the numbers of loops of the spiral inductors are increased it becomes more difficult and time consuming to model such structure. The input vector $\mathbf{x}$ consists of space (S) and frequency (f). The output vector $\mathbf{y}$ of the model are the real and imaginary parts of S-parameters. The coarse data generator g_c in this case is the EM simulator *Ansoft-Ensemble* while the 3D EM simulator *Ansoft-HFSS* is used as fine data generator g_f . Figure 4.20 represents the neural model of the 8-loop inductor. Table 4.14 represents the model input and output parameters and its effective range.

Generation of expensive EM data and also computational time is reduced significantly, for all neural models from KAMG. The algorithm uses 184 coarse data for initial training in first and second stages. From figure 4.21 we observe that neural models from KAMG achieve much better accuracy with less expensive EM data. Table 4.15 demonstrates that for a given number of fine data our proposed technique achieves much better accuracy than the conventional AMG method. As a result the overall model generation time is significantly reduced as is evident from Table 4.16. So our proposed technique uses fewer expensive data and less computational time when compared with AMG method without knowledge.

Figure 4.20: Neural Model of 8-Loop Spiral Inductor.

TABLE: 4.14 NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE 8-LOOP SPIRAL INDUCTOR MODEL AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
SPACE	S	0.15 MM – 0.25 MM
FREQUENCY	f	1 GHZ – 20 GHZ
REAL S11	RS11	N/A
IMAGINARY S11	IS11	N/A
REAL S21	RS21	N/A
IMAGINARY S21	IS21	N/A

Figure 4.21: Accuracy comparison of the 8-loop Spiral Inductor neural models generated by proposed KAMG and existing AMG techniques.

TABLE 4.15: ACCURACY COMPARISON BETWEEN 8-LOOP SPIRAL INDUCTOR NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 32 FINE DATA.

Neural Modeling Technique	Test Error
Conventional training	4.45%
AMG	2.65%
Proposed KAMG-DM	0.89%
Proposed KAMG-KBNN	0.97%
Proposed KAMG-PKI	0.72%
Proposed KAMG-SMNN	0.92%

TABLE 4.16: COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE 8-LOOP SPIRAL INDUCTOR MODELS WITH 1% TEST ERROR.

Neural Modeling Technique	No. of Fine Data	CPU for Data Generation
Conventional training	98	1764 min
AMG	52	936 min
Proposed KAMG-DM	27	486 min
Proposed KAMG-KBNN	32	576 min
Proposed KAMG-PKI	23	414 min
Proposed KAMG-SMNN	27	486 min

Chapter 5

Circuit Application and Statistical Design based on Neural Model

In this chapter we will discuss about EM based optimization and statistical design by neural models and their application to power distribution and decoupling optimization. Artificial neural networks recently gained attention as fast and flexible vehicle to microwave circuit design. EM based neural network models, i.e., neural models trained from EM data of microwave components, act as a bridge connecting EM simulation with computer-aided design (CAD) tools. EM based design of multilayer PCB including power/ground planes, drivers and other components is a challenging endeavor for the circuit designer. In this chapter we have described a new neural network formulation of the power plane, which allows automatic adjustment of component and connection geometry and then provides fast estimation of the EM effects with respect to the different connecting positions for circuit optimization and statistical design. The work in this chapter was performed utilizing the simulator driver described in Chapter 3.

5.1 Introduction

The drive in the microwave industry to meet the demands of high manufacturability and faster design cycles created a need for efficient statistical design techniques. Statistical analysis and yield optimization that take into account manufacturing tolerances, model uncertainties, variations in process parameters etc., are widely used in circuit design methodologies. It requires repetitive circuit simulations, which involves solving of complicated physics-based equations. So the overall design process demands huge CPU-time. For practical interactive design, where re-optimization of the circuit may be required after modification in design specifications, there is a clear need for faster techniques. Statistical analysis and optimization with detailed physics/EM models of active/passive components can be an important step towards a design of first-pass success but it is also computationally prohibitive to optimize the geometry using conventional CAD techniques [1].

Design optimization of multilayer printed circuits is important for electronic applications. A multi-layer PCB often contains power-plane and ground plane. As signal frequency is increased, non-ideal EM effects such as discontinuities and radiation losses become more prominent in circuit design which results in many signal integrity problems in switching circuits and high frequency analog circuits. One of the major factors concerning the efficiency of switching circuits is the Delta-I noise or power/ground bounce which increases with faster device switching speed, more I/O's and higher clock frequency. The cause of the voltage fluctuation in the power/ground reference planes is the transient current drawn by a large number of drivers switching simultaneously. This could degrade the buffer output waveforms, falsely set the latches,

and influence the quiet chips due to the coupling between the power distribution and the signal distribution [61,63,67-71].

The main objective in the design of a power/ground distribution system is to ensure the power and ground voltage fluctuation is within certain allowed margins. When the geometry is changed during optimization and statistical design a repetitive set of EM calculations for the power plane inductive mesh is required. This unnecessarily increases the CPU time for optimization and Monte Carlo analysis. In order to design PCBs including the EM effects of various components as well as the power/ground effects we need to develop a fast model for the power/ground plane including the effect of dynamic geometric change.

In this chapter we will explain the combination of two separate concepts of EM based optimization and EM based statistical design using neural networks. Firstly we present a new neural network formulation technique of the power/ground plane, which allows dynamic adjustment of components and connection geometry. By developing such models we avoid repetitive set of EM calculations for the power plane inductive mesh and enhance the computational speed. The models can then be used efficiently for circuit simulation and design optimization. By placing the decoupling capacitor and the power supply planes efficiently in high frequency circuits, maximum amount of power supply noise can be reduced. Thereafter a novel CAD approach based on neural networks is presented, for accurate and efficient statistical design and optimization of high frequency circuits. Our approach combines non-neural EM based techniques and the recent neural network approaches to perform yield optimization. Both these approaches are demonstrated through the CAD of an amplifier circuit and a board-level interconnects.

5.2 Proposed representation of circuit components by neural models

5.2.1 Neural Models of Passive Components

In conventional simulators, R-C components can be modeled as ideal lumped components or physically based semi-analytical components. In high frequency planar circuits, similar R-C models are not so accurate as EM models because they do not adequately include all high frequency EM effects such as discontinuities, radiation losses, etc. So EM representation of circuit components becomes important. But most of the commercial circuit simulators fail to take into account all the EM effects of the passive components. In high frequency the circuit wavelength becomes comparable to the dimension of the component and the conventional lumped models like resistors and capacitors can no longer be used in circuit design. So, to obtain accurate and continuous models covering the entire user defined physical and electrical parameter ranges all the passive models are replaced by their EM based neural models.

Considering embedded resistors and capacitors in multilayer printed circuits, the neural models generated to replace the resistors and the capacitors are trained with data from a 2.5-D EM simulator, namely *Sonnet-Lite*. The notation of the input/output parameters and their range for the embedded resistor and capacitor used are enlisted in Table 5.1 and 5.2 respectively. The training of both the neural models for the resistors and capacitors are done through the *NeuroModeler* software. A wide range of data is considered to create a large library of neural

models for the resistors and capacitors, which can be plugged into any circuit according to user requirement [62].

In figure 5.1, the comparison between the output of ideal lumped 10Ω resistor performed in *ADS* and its EM-based neural model is shown in the form of S-parameters. At low frequency, neural model and ideal model match each other, but at high frequency the neural model exhibit EM effects while ideal model does not.

Figure. 5.1: Comparison of S-parameters of the lumped 10Ω resistor (—) and its EM based neural model (---).

TABLE 5.1. NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED RESISTOR
MODELS AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
LENGTH	L	5MILS – 20MILS
WIDTH	W	5MILS – 20MILS
RESISTIVITY	R	10 Ω CM – 1000 Ω CM
PERMITTIVITY	ϵ_r	2 – 10
FREQUENCY	f	0.1 GHZ – 20 GHZ
REAL S11	RS11	N/A
IMAGINARY S11	IS11	N/A
REAL S21	RS21	N/A
IMAGINARY S21	IS21	N/A

TABLE 5.2. NOTATION FOR INPUT AND OUTPUT PARAMETERS OF THE EMBEDDED
CAPACITOR MODELS AND EFFECTIVE RANGE OF THEIR INPUT PARAMETER.

PARAMETERS	NOTATION	RANGE
LENGTH	L	5 MILS – 20 MILS
THICKNESS	T	0.6 MILS – 1.2 MILS
CAPACITOR PERMITTIVITY	ϵ_{rc}	800-1400
PERMITTIVITY	ϵ_r	2 – 10
FREQUENCY	f	0.1 GHZ – 20 GHZ
REAL S11	RS11	N/A
IMAGINARY S11	IS11	N/A
REAL S21	RS21	N/A
IMAGINARY S21	IS21	N/A

5.2.2 Neural Models of Power Plane

The power planes can be used as the connecting medium in multilayer printed circuit board between any given N numbers of components used in the circuit. Figure 5.2 represents the schematic where different components are connected to each other by power plane. In the figure, V_{dc} represents the DC voltage source, C_{dc} is a DC source decoupling capacitor, and R_1 to R_N represents N components being connected to the power plane. Our main aim is to minimize the power plane noise effects by varying the physical location of any of these N components. Lets define the coordinates (X, Y) of the DC source decoupling capacitor as reference coordinates. Then, the physical position of any k^{th} component can be defined with respect to the reference point (i.e., (dX_k, dY_k)); the subscript 0 is referred to the DC source.

This approach enables us to develop a two-port EM model for the power plane, which can be used recursively to represent N points in power distribution. The neural model of the power plane is trained by driving an EM simulator [48] for the connection between the DC source decoupling capacitor and any component k ($k = 0, \dots, N$). The data generated for training the neural model of the power plane consist of five different control parameters. The parameters are the reference coordinates (X, Y) , the coordinates (dX, dY) of any components connected to the component at the reference nodes and the frequency (f). The neural model of the power plane is shown in the adjoining figure. 5.3.

Figure. 5.2: Physical illustration of a circuit including the power plane for EM based design optimization.

Figure. 5.3: Neural model input-output parameter of the power plane.

5.2.3 Formulation of overall circuit including power plane

In this approach we have used the power plane neural model shown in figure. 5.3. The output data representing the power plane effect is a two-port, S-parameter matrix. As it is difficult to predict the number of elements in a circuit, it is required to generate a very generic model for the power plane. So, we have modeled the power plane between a reference point (X, Y) with respect to only one set of other coordinates (dX, dY) in the power plane. As because the power plane is a linear component we can use this formulation to represent one connection at a time and use it recursively for multiple component connection. So for N components used in the circuit we will have N power plane neural models, which form the basis of connection between the reference point and the components. The position of any component K (X_k, Y_k) can be defined by the following equations.

$$X_k = X + dX_k \quad (5.1)$$

$$Y_k = Y + dY_k \quad (5.2)$$

where dX_k and dY_k are positions of the component K with respect to the reference point. In figure 5.2 an arbitrary circuit including the power/ground plane is physically illustrated for EM based design optimization. In the figure the coordinates dX_k and dY_k are defined with respect to the reference point X, Y .

5.2.4 Statistical Design Concept

Statistical analysis and yield optimization are basic tools of circuit CAD and a significant amount of work has been reported related to them [3,27,72]. Let ϕ be a

vector containing all the statistical design parameters. In other words, the vector ϕ consists of user-defined input parameters (geometrical/electrical), except frequency. We must formulate the suitable variables using the neural network setup. There are several sets of vectors. For the power plane the ϕ vector consists of the following parameters,

$$\phi_{pp} = [X, Y, dX_1, dY_1, \dots, dX_N, dY_N]^T. \quad (5.3)$$

As we will be using the power plane neural model recursively for N component connection, the ϕ vector for power plane consist of one set of reference point and several sets of dX_k, dY_k , for all the components in the printed circuits.

For other components like the resistor and capacitor model ϕ vector is defined as a function of

$$\phi_R = [L, W, \epsilon_r, \rho]^T \quad (5.4)$$

$$\phi_C = [L, T, \epsilon_r, \epsilon_{rc}]^T \quad (5.5)$$

for the resistor and the capacitor respectively.

Yield analysis is carried out with respect to the parameters contained in ϕ [72-73].

A set of error functions $e_j(\phi)$ is then defined as

$$e_j(\phi) = W_u(\psi_i) \{F(\psi_i, \phi) - S_u(\psi_i)\} \quad (5.6)$$

for upper specifications, and

$$e_j(\phi) = W_l(\psi_i) \{F(\psi_i, \phi) - S_l(\psi_i)\} \quad (5.7)$$

for lower specifications, where ψ_i can be either frequency f_i for frequency domain CAD or time t_i for time domain CAD.

In the above equations the optimization parameters W_u and W_l are positive weighting

factors. The function $F(\psi_i, \boldsymbol{\varphi})$ may represent several responses like power gain or input impedance at frequency f_i in the frequency domain or transient responses like time delay and cross talk at time t_i in the time domain. The factors $S_u(\psi_i)$ and $S_l(\psi_i)$ represent the upper and lower specifications on the circuit response at that frequency or time. In the equations $j = 1, 2, \dots, m$, where m is the total number of error functions resulting from various specifications at selected frequency samples f_i or time t_i .

We define a function $I_a(\boldsymbol{\varphi})$ as

$$I_a(\boldsymbol{\varphi}) = \begin{cases} 1, & \text{if } e_j(\boldsymbol{\varphi}) \leq 0, \text{ for all } j \\ 0, & \text{otherwise} \end{cases} \quad (5.8)$$

If a circuit meets all the specifications, it is acceptable, and $I_a(\boldsymbol{\varphi}) = 1$. If a circuit violates one or more specifications, it is unacceptable, and $I_a(\boldsymbol{\varphi}) = 0$. The yield of the circuit $Y(\boldsymbol{\varphi})$ can be estimated as

$$Y(\boldsymbol{\varphi}) = \frac{\sum_{k=1}^M I_a(\boldsymbol{\varphi}^k)}{M} \quad (5.9)$$

where M is the total number of circuits, which are random outcomes based on the statistical variations chosen during Monte Carlo analysis [12].

After yield analysis, a numerical optimization is performed to find $\boldsymbol{\varphi}_a^0$ such that the yield $Y(\boldsymbol{\varphi})$ is maximized. A formulation based on generalized l_p centering can be expressed as

$$\underset{\boldsymbol{\varphi}_a^0}{\text{Minimize}} U(\boldsymbol{\varphi}_a^0) = \sum_{k \in K} \alpha_k v(\boldsymbol{\varphi}^k) \quad (5.10)$$

where α_k is a weighting factor and $v(\boldsymbol{\varphi}^k)$ is a generalized l_p function over $e_j(\boldsymbol{\varphi}^k)$, $k = 1, 2, \dots, M$. The index set K is defined as

$$K = \overset{\Delta}{\left\{ k \mid v(\boldsymbol{\varphi}^k) > 0, k = 1, 2, \dots, M \right\}}. \quad (5.11)$$

After yield optimization, the CAD tools offer the yield-optimal values of equivalent circuit parameters or any other approximate model parameters used. However, there is a missing gap between the optimized circuit parameter values and the actual physical/geometrical parameters of the microwave device/circuit under consideration. In our neural model based yield optimization technique, this problem is conveniently avoided.

5.3 EXAMPLES

5.3.1 Amplifier

5.3.1.1 EM based amplifier circuit optimization using neural networks

In order to illustrate our proposed method we used an amplifier circuit (Figure. 5.4) in *ADS* where all the passive components are replaced by their equivalent neural models (Figure. 5.5). N_{Cdc} and N_{Cd} are the neural models of the decoupling capacitors of the original circuit. N_{Cc} is the neural model of coupling capacitors. N_{R1} to N_{R4} represent the neural model of the resistor bias network and N_{RL} is the neural model of the resistive load. Here, port-1 of the power plane is connected to the output via of the DC-source decoupling capacitor, which is the reference point and port-2 is connected to the bias source and two resistors (R_1 and R_2). So we have three-power plane models in the circuit namely N_{P1} , N_{P2} and N_{P3} and we will model the connections through four set

of coordinates, namely: (X, Y) , (dX_0, dY_0) , (dX_1, dY_1) and (dX_2, dY_2) for the DC source decoupling capacitor, the DC source, the resistance R_1 and R_2 respectively.

Figure 5.4: RC Amplifier

Figure. 5.5: Amplifier circuit example where all the passive components and power plane are represented by neural models. N_P , N_R and N_C represent the neural models of the power planes, resistors and capacitors respectively.

The objective is to reduce the maximum amount of power source noise by changing the geometrical parameters of each connection to the power plane. But for every change in the values of the geometrical parameters the EM effects need to be recalculated. This step is very difficult to achieve in *ADS* without using neural models which allow a continuous change of the component connection coordinates (X, Y) , (dX_0, dY_0) , (dX_1, dY_1) and (dX_2, dY_2) by determining the S-parameters for every neural model at each optimization iteration.

TABLE 5.3: POSITION OF THE COMPONENT CONNECTIONS TO THE POWER PLANE BEFORE AND AFTER OPTIMIZATION.

Geometrical Coordinates	Before Optimization	After Optimization
X	20.00 mils	12.80 mils
Y	20.00 mils	13.08 mils
dX1	40.00 mils	81.37 mils
dY1	40.00 mils	37.33 mils
dX2	20.00 mils	13.81 mils
dY2	20.00 mils	21.16 mils
dX3	80.00 mils	113.65 mils
dY3	80.00 mils	110.78 mils

Figure. 5.6: Comparison of the amplifier gain obtained by lumped elements (—) and by their neural models before (---) and after optimization (-o-o-).

Our objective function is the amplifier gain in the frequency range 1-20 GHz. The optimization algorithms used for this are least-pth and quasi-Newton. The position of the various components connected to the power plane before and after optimizations are given in the Table 5.3.

Figure. 5.6 shows the gain of the original ideal amplifier simulation and that of neural model amplifier before and after optimization.

5.3.1.2 EM based Monte-Carlo analysis of the amplifier using neural networks

Monte-Carlo analysis and yield optimization of the circuit represented by component neural models trained from 3D-EM data is then carried out. The CAD tools perform optimization over the ϕ -space.

To optimize the gain of the amplifier circuit, a nominal optimization is first performed. The parameters considered in optimization are length and width of all the resistor neural models.

For this example, two separate design constrains were laid at two separate frequency points for yield optimization. For frequency, $f \leq 2$ GHz we define the upper specification for power gain as $S_u(f) = 4$ dB and the lower specification as $S_l(f) = 2.5$ dB whereas for the frequency range of $10 \text{ GHz} < f < 14 \text{ GHz}$, $S_u(f) = 1$ dB and $S_l(f) = -1.5$ dB.

The error function for $f \leq 2$ GHz then can be defined as

$$e_j(\varphi) = F(f_i, \varphi) - 4 \text{ dB} \quad (5.12)$$

for upper specifications, and

$$e_j(\varphi) = 2.5 \text{ dB} - F(f_i, \varphi) \quad (5.13)$$

for lower specifications and for $10 \text{ GHz} < f < 14 \text{ GHz}$, the error function can be defined as

$$e_j(\varphi) = F(f_i, \varphi) - 1 \text{ dB} \quad (5.14)$$

for upper specifications, and

$$e_j(\varphi) = -1.5 \text{ dB} - F(f_i, \varphi) \quad (5.15)$$

for lower specifications.

After nominal optimization, we performed yield analysis for 200 random outcomes and the yield was 29%. In order to improve it, a yield optimization was performed using the nominal design center as the starting point. After yield optimization, the yield of the circuit is observed to be 69%. Figure. 5.7 represents the circuit response after yield optimization [74].

Figure. 5.7: Gain vs. Frequency for *ADS* lumped components (—○—) and Monte-Carlo analysis using Neural models (—).

5.3.2 Board-Level Interconnects

5.3.2.1 EM based circuit optimization of board-level interconnect using neural networks

In our next example we have considered a 4-bit board level interconnects. In this section we describe a systematic formulation for optimization of transient responses such as delay time, crosstalk and power supply noise voltage as is required in high-speed VLSI interconnect design [75-76]. The interconnects and the power plane consist

of sets of coupled transmission lines and several sets of buffers. A 3-D representation of the structure is shown in Figure. 5.8.

In this example, the reference point is the 2V DC source. So the port-1 is referred to that point where as port-2 is connected to the four ICs used in the circuit. So, we have used four power plane models.

Figure. 5.8: Physical representations of the 4-bits board-level interconnect.

This example utilises one of the most recent research topics of generating time-domain models using state space equations or equivalent circuits from S-parameters data for time domain EM simulator [77]. In this example we replaced the buffers and power-plane by their equivalent neural models. All these neural models are trained in

the time domain. The models used for the transmission lines are the internal defined “ustrip” models in H-SPICE simulators. Figure 5.9 represents the schematic of the board level interconnects.

Figure 5.9: 4-Bit board-level interconnect circuit.

Our main objective is to reduce the time delay and the crosstalk of the interconnect circuit and also improve the power/ground bounce of the circuit response by varying the geometrical position of each connection to the power-plane. The excitation signal used is a 1000ps trapezoidal signal.

Let us consider the DC source coordinates (X, Y) as the reference coordinates. The parameters considered for optimization are the geometrical positions of the four ICs (dX_N, dY_N) , $N = 1, \dots, 4$, with respect to the voltage source, the lengths (L_1, L_2, L_3) and distances

(W_1, W_2, W_3) between the four-coupled transmission lines. But, as mentioned earlier for every change in the values of the geometrical parameters the EM effects need to be recalculated. This step is very difficult to achieve in *HSPICE* without using neural models which allow a continuous change of (dX_1, dY_1), (dX_2, dY_2), (dX_3, dY_3) and (dX_4, dY_4) by determining the Y-parameters for every neural model at each optimization iteration.

The responses of interest are the signals at nodes V_a and V_c and the crosstalks at nodes V_b and V_d as shown in figure 5.10. For optimization of the power supply voltage we have considered the node signal voltages at node V_{ic1} , V_{ic2} , V_{ic3} and V_{ic4} as shown in figure 5.11. The responses before and after optimization are plotted in figure 5.10 and figure 5.11. The delay of V_a and V_c are 1180 ps and 886 ps respectively based on a threshold voltage of 0.9V. We specified the delay to be reduced by 120 ps for both the nodes. A 0.03V upper specification was imposed on the magnitude of V_b and V_d in the interval of 100 ps to 1200 ps. The co-ordinates of the power-plane before and after optimization are given in the Table 5.4. The initial length of the transmission lines (L_1, L_2, L_3) was fixed at 0.6 mm and the width (W_1, W_2, W_3) was fixed at $1.7\mu\text{m}$. After six iterations of minimax optimization, the propagation delay time for V_c and V_a is reduced by 150 ps and 120 ps respectively. The crosstalk signals for V_b and V_d were below our specified 0.03v. The optimized value of the variables is shown in the adjoined Table 5.5. For the power supply noise voltage there is also improvement after optimization of the position of the 4 IC chips connection in the power plane as we can observe from figure. 5.11. The noise voltages at the four nodes are reduced significantly thereby improving the transient response of the circuit by a considerable amount.

TABLE 5.4: POSITION OF THE COMPONENT CONNECTIONS TO THE POWER PLANE BEFORE AND AFTER OPTIMIZATION.

Geometrical coordinates	Before Optimization	After Optimization
dX1	25 mils	25.266 mils
dY1	10 mils	8.632 mils
dX2	35 mils	33.991 mils
dY2	20 mils	14.619 mils
dX3	45 mils	40.754 mils
dY3	10 mils	9.758 mils
dX4	55 mils	51.110 mils
dY4	15 mils	17.320 mils

TABLE 5.5: LENGTH AND WIDTH OF THE TRANSMISSION LINE BEFORE AND AFTER OPTIMIZATION.

Geometrical coordinates	Before Optimization	After Optimization
L1	0.6 mm	0.541 mm
L2	0.6 mm	0.501 mm
L3	0.6 mm	0.617 mm
W1	1.7 μm	2.100 μm
W2	1.7 μm	1.899 μm
W3	1.7 μm	1.913 μm

(a)

(b)

(c)

(d)

Figure 5.10: The board-level interconnect example. Signal responses of the circuit before (—) and after (----) optimization. (a) Signal response V_a . (b) Crosstalk V_b (c) Signal response V_c . (d) Crosstalk V_d .

(a)

(b)

(c)

(d)

Figure 5.11: Power Supply noise of the power plane for each of the 4 IC's used in the board-level interconnect before (—) and after (—) optimization. (a) Signal response at node V (ic_1); (b) Signal response at node V (ic_2); (c) Signal response at node V (ic_3); (d) Signal response at node V (ic_4);

5.3.2.2 EM based Monte Carlo analysis of the board-level interconnect using neural networks

Monte-Carlo analysis of the circuit represented by component neural models trained from 3D-EM data is performed. The CAD tools perform optimisation over ϕ -space. The parameters considered for the Monte-Carlo analysis is length of transmission lines, spacing between them and geometrical position of all the power-plane neural models connected to the four IC's. The Monte Carlo analysis for 100 random outcomes is shown in Fig. 5.12.

Figure 5.12: Monte Carlo analyses of the board-level interconnect using Neural models.

Chapter 6

Conclusions and Future Research

6.1 Conclusion

The potential need for an efficient and faster model generation procedure motivated this thesis. Efficient microwave model generation becomes very costly if the accurate training data generation takes a lot of time. As a result the entire design process is stretched. The work proposed in this thesis combines several advanced concepts (KBNN, PKI, SM, DM) with neural model generation method which is able to reduce the need for expensive training data and thereby speeding up neural model generation significantly. Besides this the neural network approach to circuit applications and statistical design has also been presented. Neural networks has gained an unprecedented popularity in the field of circuit and statistical design mainly because of the ability of neural network to compute circuit parameters nearly as accurate as those obtained from full-wave EM simulations. We presented a novel approach for the formulation of the full EM representation and optimization of passive devices based on neural networks. The proposed models allow dynamic geometric parameter based design and are applied to circuit optimization. Summarily we can say that

(1) We proposed an advanced KAMG technique for automatic generation of knowledge based neural models. The technique integrates powerful concepts such as AMG, knowledge

neural networks, and space mapping, into a framework providing enhanced accuracy and increased speed. Motivated by the space-mapping concept, the KAMG exploits both coarse and fine data generators for efficient neural model development. A variety of knowledge architectures have been utilized to accomplish reinforced neural network training from coarse and fine data. For a given model accuracy, the proposed technique uses fewest fine data as compared to all other existing techniques including AMG and conventional technique. Fewer fine training data translates into significantly reduced CPU time for data generation, thus resulting in faster model development.

(2) A new approach of designing and placement of the high frequency circuits and components have been presented. A frequency domain CAD of an amplifier circuit using power-plane, resistors and capacitor neural models is performed. Minimization of power supply noise and thereby increasing the power gain of the circuit was achieved. High-speed board-level interconnect design using power plane neural models, buffer neural models and transmission lines are carried out by time-domain CAD. Minimization of effects like time delay, crosstalk and power supply noise was formulated into a minimax optimization problem. Considerable improvements on all such effects were obtained by our proposed method. Our design techniques allows EM effects to be used while performing such optimization task and thereby providing an useful enhancement to the existing design criterion such as layout and signal integrity optimization. The proposed approach has a future potential of EM based design including power supply noise problems, time delay, crosstalk and decoupling capacitor optimization for the switching circuits and RF and microwave components in wireless communication systems.

(3) A statistical design technique using neural network models trained from EM data is presented. The results show that neural models provide accurate EM effects under high frequencies, thus enhancing circuit design. In addition, neural models also make Monte-Carlo analysis and Yield optimization possible. The statistical design technique is demonstrated by a complete end-to-end example of an amplifier circuit optimization and board-level interconnect circuit optimization is presented. Our methodology acts as a bridge that connects EM simulation with circuit design, thus contributing to advancement in microwave CAD.

6.2 Suggestions for Future Research

Neural network is emerging as one of the most powerful tools for modeling of microwave devices and systems. Modeling still remains a major bottleneck for CAD of certain classes of RF/microwave circuits. From the viewpoint of future research, complicated passive circuit modeling and more improvement in model generation algorithms will benefit the application of combined neural networks in all level of microwave design from modeling and simulation to optimization and statistical design.

An interesting topic in this regard is the application of KAMG modeling from individual passive components to integrated RC or LC circuits which can be applied to model RC and LC filters respectively. The KAMG algorithm can also be utilized to model non-linear devices and circuits in the future.

The power plane structure modeled has been used so far to model the single amplifier. It can be utilized in distributed amplifier modeling which will be a significant contribution to the existing CAD methodologies.

Appendix A:

This program has enabled us to automate the data generation process in *Sonnet-Lite*. The examples in Chapter 4 and 5 were done using a variation of the following program for different geometry of the components.

Format of the *Sonnet-Lite* C-Program for Capacitor Model:

Input Parameters: Length, Height, Substrate Permittivity and Permittivity of Capacitor.

Output Parameters: Real and Imaginary parts of S-Parameters.

```
#include <stdio.h>
#include <process.h>

void SonnetDriver(double[], double[]);
void main()
{
    double xx[10], yy[8], tmp;

    /*INPUT DATA FILE*/

    FILE *f1;
    f1 = fopen("Sonnet.in", "r");

    fscanf(f1, "%lf", &tmp);
    xx[0] = tmp;           // Length
    fscanf(f1, "%lf", &tmp);
    xx[1] = tmp;           // Height
    fscanf(f1, "%lf", &tmp);
    xx[2] = tmp;           // Substrate Permittivity
    fscanf(f1, "%lf", &tmp);
    xx[3] = tmp;           // Permittivity of Capacitor
    fscanf(f1, "%lf", &tmp);
    xx[4] = tmp;           // Frequency
    fclose(f1);
```

```

SonnetDriver(xx, yy);
}

void SonnetDriver(double x[], double y[])
{
    int l, h, wc, a1, a2, nx, ny;
    double h1, h2, er, erc, freq delx, dely;

    nx = 30;      // Number of Cells along X-axis
    ny = 20;      // Number of Cells along Y-axis
    delx = 1.0;   // Cell Length
    dely = 1.0;   // Cell Width

    // Choose the Variable Inputs

    l    = int (x[0] / dely);
    h    = int (x[1] / delx);
    er   = x[2];
    erc  = x[3];
    freq = x[4];

    ny=2*w;      // Number of Cells along Y-axis
    nx=1+24;     // Number of Cells along X-axis.

    a1=2*nx;
    a2=2*ny;

    // Input Co-ordinate Definitions
    int X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13;
    int X14, X15, X16;
    int Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13;
    int Y14, Y15, Y16, Y17=0;

    // Calculation of the coordinates
    X1 = 0;
    Y1 = ny/2 - wc/2;

    X2 = nx/2 - l/2 - wc;
    Y2 = ny/2 - wc/2;

    X3 = nx/2 - l/2 - wc;
    Y3 = ny/2 + wc/2;

```



```

X4 = 0;
Y4 = ny/2 + wc/2;

X5 = nx/2 - l/2 - wc;
Y5 = ny/2 - w/2;

X6 = nx/2 - l/2;
Y6 = ny/2 - w/2;

X7 = nx/2 - l/2;
Y7 = ny/2 + w/2;

X8 = nx/2 - l/2 - wc;
Y8 = ny/2 + w/2;

X9 = nx/2 + l/2;
Y9 = ny/2 - w/2;

X10 = nx/2 + l/2;
Y10 = ny/2 + w/2;

X11 = nx/2 + l/2 + wc;
Y11 = ny/2 - w/2;

X12 = nx/2 + l/2 + wc;
Y12 = ny/2 - wc/2;

X13 = nx/2 + l/2 + wc;
Y13 = ny/2 + wc/2;

X14 = nx/2 + l/2 + wc;
Y14 = ny/2 + w/2;

X15 = nx;
Y15 = ny/2 - wc/2;

X16 = nx;
Y16 = ny/2 + wc/2;

// File Definitions

FILE *fin;
fin = fopen("qjz.geo", "w");

fprintf(fin, "VER 6.0b-Lite\n");
fprintf(fin, "BUILD 0\n");
fprintf(fin, "LIC SL00000.101\n");

```

```

fprintf(fin, "DAT Mon Oct 16 17:08:43 2000\n");
fprintf(fin, "LEN mils 2.54e-005\n");
fprintf(fin, "REF 0 0 0 0 0 0 0 0\n");
fprintf(fin, "TOP 0 0 0 0\n");
fprintf(fin, "TON 0 Lossless\n");
fprintf(fin, "BOT 0 0 0 0\n");
fprintf(fin, "BON 0 Lossless\n");
fprintf(fin, "RES %g 0 0 0\n", r);
fprintf(fin, "REN 1 Metall\n");
fprintf(fin, "BOX 1 %d %d %d %d 20 0\n", nx, ny, a1, a2);
fprintf(fin, "%g 1 1 0.03 0 0 2\n", h2);
fprintf(fin, "%g %g 1 0.03 0 0 2\n", h1, er);
fprintf(fin, "POR 3 3 0 1 50 0 0 0 0\n");
fprintf(fin, "POR 4 1 0 2 50 0 0 0 0\n");
fprintf(fin, "NUM 6\n");

fprintf(fin, "0 5 -1 N 9 1 1 100 100 0 0 0\n");
fprintf(fin, "%d %d\n", X5, Y5);
fprintf(fin, "%d %d\n", X6, Y6);
fprintf(fin, "%d %d\n", X7, Y7);
fprintf(fin, "%d %d\n", X8, Y8);
fprintf(fin, "%d %d\n", X5, Y5);
fprintf(fin, "END\n");

fprintf(fin, "0 5 -1 N 10 1 1 100 100 0 0 0\n");
fprintf(fin, "%d %d\n", X9, Y9);
fprintf(fin, "%d %d\n", X11, Y11);
fprintf(fin, "%d %d\n", X14, Y14);
fprintf(fin, "%d %d\n", X10, Y10);
fprintf(fin, "%d %d\n", X9, Y9);
fprintf(fin, "END\n");

fprintf(fin, "0 5 0 N 11 1 1 100 100 0 0 0\n");
fprintf(fin, "%d %d\n", X6, Y6);
fprintf(fin, "%d %d\n", X9, Y9);
fprintf(fin, "%d %d\n", X10, Y10);
fprintf(fin, "%d %d\n", X7, Y7);
fprintf(fin, "%d %d\n", X6, Y6);
fprintf(fin, "END\n");

fprintf(fin, "0 5 -1 N 12 1 1 100 100 0 0 0\n");
fprintf(fin, "%d %d\n", X1, Y1);
fprintf(fin, "%d %d\n", X2, Y2);
fprintf(fin, "%d %d\n", X3, Y3);
fprintf(fin, "%d %d\n", X4, Y4);
fprintf(fin, "%d %d\n", X1, Y1);
fprintf(fin, "END\n");

```

```

fprintf(fin, "0 5 -1 N 14 1 1 100 100 0 0 0\n");
fprintf(fin, "%d %d\n", X12, Y12);
fprintf(fin, "%d %d\n", X15, Y15);
fprintf(fin, "%d %d\n", X16, Y16);
fprintf(fin, "%d %d\n", X13, Y13);
fprintf(fin, "%d %d\n", X12, Y12);
fprintf(fin, "END\n");

fprintf(fin, "1 5 -1 N 17 1 1 100 100 0 0 0\n");
fprintf(fin, "%d %d\n", X1, Y17);
fprintf(fin, "%d %d\n", nx, Y17);
fprintf(fin, "%d %d\n", nx, ny);
fprintf(fin, "%d %d\n", X1, ny);
fprintf(fin, "%d %d\n", X1, Y17);
fprintf(fin, "END\n");
fclose(fin);

FILE *fnetlist;
fnetlist = fopen("qjz.net", "w");

fprintf(fnetlist, "!< FTYPE NET\n");
fprintf(fnetlist, "! File: qjz.net\n");
fprintf(fnetlist, "! Date: Sept 26, 2000\n");
fprintf(fnetlist, "! Analyze qjz\n");
fprintf(fnetlist, "DIM\n");
fprintf(fnetlist, "FREQ          GHZ\n");
fprintf(fnetlist, "CKT\n");
fprintf(fnetlist, "GEO 1 2 qjz.geo OPT=vd      CTL=internal\n");
fprintf(fnetlist, "      DEF2P      1 2          ATTEN\n");
fprintf(fnetlist, "FILEOUT\n");
fprintf(fnetlist, "ATTEN TOUCH qjz.rsp  S RI R 50\n");
fprintf(fnetlist, "FREQ\n");
fprintf(fnetlist, "SWEEP %f %f %f\n", freq, freq, freq);
fclose(fnetlist);

system("del C:\\Sonnet\\qjz.d");
system("del C:\\Sonnet\\qjz.pd");
system("del C:\\Sonnet\\qjz.rsp");
system("C:\\Sonnet\\bin\\em C:\\Sonnet\\qjz.net");
}

```

Appendix B:

This program has enabled us to automate the data generation process in *Ansoft-HFSS*. The examples in Chapter 4 and 5 were done using a variation of the following program for different geometry of the components.

Format of the *ANSOFT-HFSS* Macro for Resistor:

Input Parameters: Length, Width, Resistivity, and Permittivity.

Output Parameters: Real and Imaginary parts of S-Parameters.

#STEP 1: DEFINE STRUCTURE GEOMETRY

```
ClearAll
ClearFunctions
SetUnits "mils" "y"
Assign FileID ImportDataBase "C:\Ansoft\in.db"
Assign output_file          "C:/Ansoft/out.tab"
```

#Input File

Assign nrow	DataBaseGetRows	FileID		
# number of input vectors				
Assign ncol	DataBaseGetColumns	FileID		
# number of columns				
Assign length	DataBaseGetField	FileID	0	0
#1st input data	LENGTH			
Assign width	DataBaseGetField	FileID	1	0
#2nd input data	WIDTH			
Assign epsilon	DataBaseGetField	FileID	2	0
#3rd input data	PERMEABILITY			
Assign realres	DataBaseGetField	FileID	3	0
#4th input data	REAL (R)			
Assign groundp	DataBaseGetField	FileID	4	0
#5th input data	HEIGHT			
Assign frequency	DataBaseGetField	FileID	5	0
#6th input data	FREQUENCY			

#Geometry

Box [0, 0, 0] 32 20 100	"air box"
Box [0, 0, 0] 32 20 10	"substrate"
Rectangle [0, 7, 10] 2 6 5	"input line" 1
Rectangle [6, 5, 10] 2 5 10	"input connector" 1
Rectangle [11, 5, 10] 2 10 10	"resistor" 1
Rectangle [21, 5, 10] 2 5 10	"output connector" 1
Rectangle [26, 7, 10] 2 6 5	"output line" 1

#STEP 2: SETUP MATERIALS

```
MaterialSelect "air"
MaterialAssign { "airbox" } "OCS"
{ "0.0000000000" \"0.0000000000" "0.0000000000" }
{ "0.0000000000" \"0.0000000000" "0.0000000000" }
```

```
MaterialAdd
MaterialSetProperty "conductivity" 0
{ "0.00000000000000000000e+000" }
MaterialSetProperty "permittivity" 0
{ epsilon }
MaterialSetProperty "imag_permittivity" 0
{ "0.00000000000000000000e+000" }
MaterialSetProperty "elec_lossstan" 0
{ "0.00000000000000000000e+000" }
MaterialSetProperty "permeability" 0
{ "1.00000000000000000000e+000" }
MaterialSetProperty "imag_permeability" 0
{ "0.00000000000000000000e+000" }
MaterialSetProperty "mag_lossstan" 0
{ "0.00000000000000000000e+000" }
```

#STEP 3: SETUP BOUNDARIES

```
TogglePickObjByName "input line"
TogglePickObjByName "input connector"
TogglePickObjByName "resistor"
SetBoundary "perfect_e"
CreateBoundary
ClearSelection
TogglePickObjByName "output line"
TogglePickObjByName "output connector"
SetBoundary "perfect_h"
CreateBoundary
ClearSelection
```

#STEP 4: SETUP SOLUTION

```
SetSolutionType 1 1 0 1 2
SetSolverComputation "all_fields" "all_modes" 1
SetFrequencyUnit "GHz"
SetExplicitFreqs 0 { }
SetFrequency frequency
SetAdaptiveInfo 6 "conv_delta" 20
SetConvDelta 0.000001
SetConvMatrix "P1M1" "P1M1" 0.002 1
SetConvMatrix "P1M1" "P2M1" 0.002 1
SetConvMatrix "P2M1" "P1M1" 0.002 1
SetConvMatrix "P2M1" "P2M1" 0.002 1
SetMeshType "initial"
SaveEditableSolveParms
ExitSolveSetup
ExitBnds 0
```

#STEP 5: SIMULATOR

```
Solve Nominal
```

#STEP 6: RESULTS

```
PostMatrixData
SzgDataKernelExport "A_6" "spreadsheet" output_file 50
SaveModule
ExitModule 0
```

Bibliography

- [1] Q.J. Zhang and K.C. Gupta, *Neural Networks for RF and Microwave Design*, Norwood, MA: Artech House, 2000.
- [2] J. Purviance and M. Meehan, "CAD for statistical analysis and design of microwave circuits," *Int. J. Microwave and Millimeter-Wave CAE*, vol. 1, pp. 59-76, 1991.
- [3] A.H. Zaabab, Q.J. Zhang, and M.S. Nakhla, "A neural network models approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349-1358, 1995.
- [4] F. Filicori, G. Ghione, and C.U. Naldi, "Physics-based electron device modeling and computer-aided MMIC design," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1333-1352, 1992.
- [5] V.K.Devabhaktuni, B. Chattaraj, M.C.E. Yagoub, Q.J. Zhang, " Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks and space mapping ", *IEEE MTT-S Int. Microwave Symp. (IMS 2002)*, Seattle, WA, USA, June 2002.
- [6] V.K. Devabhaktuni, M.C.E. Yagoub, and Q.J. Zhang, "A robust algorithm for automatic development of neural network models for microwave applications," *IEEE Trans. Microwave Theory Tech.*, vol. 49, pp. 2282-2291, 2001.
- [7] P.M. Watson, K.C. Gupta, and R.L. Mahajan, "Applications of knowledge-based artificial neural network modeling to microwave components," *Int. J. RF and Microwave CAE*, vol. 9, pp. 254-260, 1999.

- [8] 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.
- [9] F. Wang, V.K. Devabhaktuni, and Q.J. Zhang, "A hierarchical neural network approach to the development of a library of neural models for microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 2391-2403, 1998.
- [10] A.H. Zaabab, Q.J. Zhang, and M.S. Nakhla, "Device and circuit level modeling using neural networks with faster training based on network sparsity," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1696-1704, 1997.
- [11] G.L. Creech and J.M. Zurada, "Neural network modeling of GaAs IC material and MESFET device characteristics," *Int. J. RF and Microwave CAE*, vol. 9, pp. 241-253, May 1999.
- [12] V.K. Devabhaktuni, C. Xi, F. Wang, and Q.J. Zhang, "Robust training of microwave neural models," *Int. J. RF and Microwave CAE*, Special Issue: Application of ANN's to RF and Microwave Design, vol. 12, pp. 109-124, 2002.
- [13] G.L. Creech and J.M. Zurada, "Neural modeling of GaAs IC material and MESFET device characteristics," *Int. J. RF and Microwave CAE*, vol. 9, pp. 241-253, 1999.
- [14] J. Rousset, Y. Harkouss, J.M. Collantes, and M. Campovecchio, "An accurate neural network model of FET intermodulation and power analysis," *Proc. European Microwave Conf.*, Prague, Czech Republic, Sept. 1996, pp. 16-19.
- [15] J.A. Garcia, A.T. Puente, A.M. Sanchez, I. Santamaria, M. Lazaro, C.J. Pantaleon, and J.C. Pedro, "Modeling MESFETs and HEMTs intermodulation distortion

- behavior using generalized radial basis function network,” *Int. J. RF and Microwave CAE*, vol. 9 pp. 261-276, 1999.
- [16] 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.
- [17] G. Antonini and A. Orlandi, “Gradient evaluation for neural-network-based electromagnetic procedures,” *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 874-876, 2000.
- [18] F. Wang and Q.J. Zhang, “Knowledge-based neural models for microwave design, ” *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2333-2343, 1997.
- [19] A. Patnaik, R.K. Mishra, G.K. Patra, and S.K. Dash, “An artificial neural network model for effective dielectric constant of microstripline,” *IEEE Trans. Antennas Propagat.*, vol. 45, p. 1697, 1997.
- [20] Q.J. Zhang, F. Wang, and M.S. Nakhla, “Optimization of high-speed VLSI interconnects: A review, ” *Int. J. Microwave Millimeter-Wave CAE*, vol. 7, pp. 83-107, 1997.
- [21] V.K. Devabhaktuni and Q.J. Zhang, “Neural Network training-driven adaptive sampling algorithm for microwave modeling, ” *Proc. European Microwave Conf.*, Paris, France, Oct. 2000, pp. 222-225.
- [22] Q.J. Zhang, “A neural network paradigm for microwave modeling, simulation and optimization,” *IEEE MTT-S Int. Microwave Symp. Workshop on Appl. of ANN to Microwave Design*, Denver, CO, June 1997, pp. 87-107.
- [23] F. Wang and Q.J. Zhang, “Incorporating functional knowledge into neural networks,” *Proc. IEEE Int. Conf. Neural Networks*, Houston, TX, June 1997, pp. 266-269.

- [24] M.Nakhla, "Application of Neural networks to simulation and optimization of high speed interconnects, " *IEEE MTT-S Int. Microwave Symp. Workshop on Appl. of ANN to Microwave Design*, Denver Co, June 1997, pp. 49-66.
- [25] C. Cho and K. Gupta, "EM-ANN modeling of overlapping open-ends in multilayer microstrip lines for design of bandpass filter, " *IEEE APS Int. Symp. Dig.*, Orlando, FL, July 1999, pp. 2592-2595.
- [26] J. Bandler, M. Ismail, J. Rayas-Sanchez, and Q. Zhang, "New directions in model development for RF/microwave components utilizing artificial neural networks and space mapping," *IEEE APS Int. Symp. Dig.*, Orlando, FL, July 1999, pp. 2572-2575.
- [27] P.M. Watson and K.C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2495-2503, 1996.
- [28] S. Wang, F. Wang, V.K. Devabhaktuni, and Q.J. Zhang, "A Hybrid neural and circuit-based model structure for microwave modeling", *Proc. European Microwave Conf.*, Munich, Germany, Oct. 1999, pp. 174-177.
- [29] 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.
- [30] 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. European Microwave Conf.*, Amsterdam, Netherlands, Oct. 1998, pp. 306-311.

- [31] K. Shirakawa, M. Shimiz, N. Okubo, and Y. Daido, "A large-signal characterization of an HEMT using a multilayered network," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1630-1633, 1997.
- [32] G. Fedi, A. Gaggelli, S. Manetti, and G. Pelosi, "Direct-coupled cavity filters design using a hybrid feedforward neural network - finite elements procedure," *Int. J. RF and Microwave CAE*, vol. 9, pp. 287-296, 1999.
- [33] P. Burrascano, S. Fiori, and M. Mongiardo, "A review of artificial neural networks applications in microwave computer-aided design," *Int. J. RF and Microwave CAE*, vol. 9, pp. 158-174, 1999.
- [34] Y. 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 MTT-S Int. Microwave Symp. Dig.*, Boston, MA, June 2000, pp. 883-886.
- [35] G. Washington, "Aperture Antenna shape prediction by feedforward neural networks," *IEEE Trans. Antenna Propagat.*, vol. 45, pp.683-688, 1997.
- [36] R. Mishra and A. Patnaik, "Neural Network based CAD model for the design of square-patch antenna," *IEEE Trans. Antenna Propagat.*, vol. 46, pp.1890-1891, 1998.
- [37] A. El Zooghby, C. Christodoulou, and M. Georgiopoulos, "Neural network-based adaptive beamforming for one- and two-dimensional antenna arrays," *IEEE Trans. Antenna Propagat.*, vol. 46, pp.1891-1893, 1998.
- [38] E. Charpentier and J.J. Laurin, "An implementation of a direction-finding antenna for mobile communications using a neural network," *IEEE Trans. Antennas Propagat.*, vol. 47, pp. 1152-1159, 1999.

- [39] S. Sagiroglu, K. Guney, and M. Erler, "Calculation of bandwidth for electrically thin and thick rectangular microstrip antennas with the use of multiplayer perceptrons," *Int. J. RF and Microwave CAE*, vol. 9, pp. 277-286, 1999.
- [40] P.R. Chang, W. H. Yang, and K.K. Chan, "A neural network approach to MVDR beamforming problem," *IEEE Trans. Antenna Propagat.*, vol. 40, pp. 313-322, 1992.
- [41] H.L. Southall, J.A. Simmers, and T.H. O' Donnell, "Direction finding in phased arrays with a neural network beamformer," *IEEE Trans. Antenna Propagat.*, vol. 43, pp. 1369- 1374, 1995.
- [42] F. Wang, Vijaya K. Devabhaktuni, C.Xi, Q.J.Zhang, "Neural Network Structures and Training Algorithms for RF and Microwave application," *Int. Journal of RF and Microwave CAE*. Special Issue: Application of ANN's to RF and Microwave Design, vol. 9, 1999, pp. 216-240.
- [43] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Upper Saddle River, NJ: Prentice Hall, 1994.
- [44] S. Bila, Y. Harkouss, M. Ibrahim, J. Rousset, E. N'Goya, D. Baillargeat, S. Verdeyme, M. Aubourg, P. Guillon, "An accurate wavelet neural-network-based model for electromagnetic optimization of microwave circuits," *Int. Journal of RF and Microwave CAE*. Special Issue: Application of ANN's to RF and Microwave Design, vol. 9, 1999, pp. 297-306.
- [45] T. Kohonen, "Self Organized Formulation of Topologically correct Feature Maps," *Biological Cybermatics*, vol. 43, 1982, pp.59-69.

- [46] *MINIMOS v.6.1*, Institute for Microelectronics, Technical University Vienna, Vienna, Austria.
- [47] *Sonnet-Lite v.7.0*, Sonnet Software Inc., Liverpool, NY, USA.
- [48] *HFSS v.7.0.11*, Ansoft Corporation, Pittsburgh, PA, US.
- [49] *HSPICE v. 2001.2*, Avant! Corporation, Fremont, CA, USA.
- [50] *ADS v 1.5*, Agilent Technologies, Palo Alto, CA, USA.
- [51] *Ensemble*, Ansoft Corporation, Pittsburgh, PA, US.
- [52] Y.K. Kim, and J.B. Ra, "Weight Value Initialization for improving the training speed in the Backpropagation Network," *Proc. IEEE Int. Conf. Neural Networks*, vol.3, 1991, pp. 2396-2401.
- [53] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," in *Parallel Distribution Processing*, vol. I, D. E. Rumelhart, and J.L. McClelland, Eds., Cambridge, MA: MIT Press, 1986, pp. 318-362.
- [54] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge, MA: Cambridge University Press, 1986.
- [55] T.R. Cuthbert Jr., "Quasi-Newton methods and constraints," *Optimization using Personal Computers*, New York, NY: John Wiley & Sons, 1987, pp. 233-314.
- [56] A.J. Shepherd, "Second-order optimization methods," *Second-Order Methods for Neural Networks*, Berlin, NY: Springer-Verlag, 1997, pp. 43-72.
- [57] 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.

- [58] V.K. Devabhaktuni, M.C.E. Yagoub, Y. Fang, J. Xu, and Q.J. Zhang, “Neural networks for microwave modeling: Model development issues and nonlinear modeling techniques”, *Int. J. RF Microwave CAE*, vol. 11, pp. 4-21, 2001.
- [59] M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sanchez, and Q.J. Zhang, “Neural space-mapping optimization for EM-based design”, *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2307-2315, 2000.
- [60] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, and R.H. Hemmers, “Space mapping technique for electromagnetic optimization”, *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 2536-2544, 1994.
- [61] R. Downing, P. Gebler and G. Katopis, “De-coupling capacitor effects on switching noise”, *IEEE Trans. Components, Hybrids Manufacturing Technology*, Vol. 16, pp. 484-489, 1993.
- [62] B. Chattaraj, M.C.E. Yagoub, X. Ding, Q.J. Zhang, “ EM based optimization of the Microwave circuits by neural models and their application to power distribution and decoupling optimization, ” *Proc. European Microwave Conf.*, London, England, Sept. 2001, vol.2, pp. 125-128.
- [63] Y. Chen, Z. Chen, and J. Fang, “Optimum Placement of Decoupling capacitor on Packages and Printed Circuit Boards Under the Guidance of Electromagnetic Field Simulation,” *Proc. IEEE Electronic Components & Technology, Conf.* pp. 756-760, May 1996, Orlando, Florida.

- [64] *NeuroModeler* 1.3, Q.J. Zhang, Dept. of Electronics, Carleton University, Ottawa, ON, Canada.
- [65] A.C. Reyes, S.M. El-Ghazaly, S.T. Dorn, M. Dydyk, D.K. Schroder, and H. Patterson, "Coplanar waveguides and microwave inductors on silicon substrates", *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 2016-2022, 1995.
- [66] B Piernas, K. Nishikawa, K. Kamogawa, T. Nakagawa, and K. Araki, "Improved three-dimensional GaAs inductors", in *IEEE-MTT Symp. Digest*, pp. 189-192, Phoenix, AZ, 2001.
- [67] Y. Chen, Z. Chen, Z. Wu, D. Xue and J. Fang, "A New Approach to Signal Integrity Analysis of High-Speed Packaging," *Proc. of IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, Portland, Oregon, Oct. 1995, pp. 236-238.
- [68] Xuebing Xie and John. L. Prince, "Frequency Response Characteristics of Reference Plane Effective Inductance and Resistance," *IEEE Trans. Advanced Packaging.*, vol. 22, pp. 221-229, 1999.
- [69] J. Fang, Z. Chen, Y. Chen, D. Xue and J. Zhao, "Integrated Electromagnetic and Circuit Simulation for Signal Integrity Analysis of High-Speed Electronic Packages," *Proc. Government Microcircuit Application Conference*, Las Vegas, Nevada, March 1997, pp. 347-350.
- [70] Y. Chen, Z. Wu, A. Agrawal, Y. Liu and J. Fang, "Modeling of Delta-I Noise in Digital Electronics Packaging," *Proc of IEEE Multi-Chip Module Conf.*, Santa Cruz, CA, Mar 1994, pp. 126-131.

- [71] E. Diaz-Alvarez, J.P. Krusius and F. Kroeger, "Modeling and simulation of integrated capacitors for high frequency chip power de-coupling", *IEEE Trans. Components Packaging Technologies*, Vol. 23, pp. 611-619, 2000.
- [72] 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.
- [73] Q.J. Zhang and M.S. Nakhla, "Statistical simulation and optimization of high-speed VLSI interconnects," *Analog Integrated Circuits and Signal Processing*, vol. 5, pp. 95-106, 1994.
- [74] 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*, Montreal, Canada, June, 2001, pp.421-426.
- [75] Q.J. Zhang and M.S. Nakhla, "Signal integrity analysis and optimization of VLSI interconnects using neural network models," *Proc. Int. Symp. Circuits Syst.*, London, England, Oct. 1994.
- [76] Q.J. Zhang, S. Lum, and M.S. Nakhla, "Minimization of Delay and Crosstalk in High-Speed VLSI interconnects," *IEEE Trans Microwave Theory and Tech.*, vol. 40, pp. 1555-1563, 1992.
- [77] X.Ding, J.J. Xu, M.C.E. Yagoub, Q.J. Zhang, "A new modeling approach for embedded passives exploiting state space formulation", *Proc. European Microwave Conf.*, Milan, Italy, Sept. 2002. (Accepted).