Advanced Microwave Modeling Framework Exploiting Automatic Model Generation, Knowledge Neural Networks, and Space Mapping

Vijay K. Devabhaktuni, *Student Member, IEEE*, Biswarup Chattaraj, *Student Member, IEEE*, Mustapha C. E. Yagoub, *Member, IEEE*, and Qi-Jun Zhang, *Senior Member, IEEE*

***Abstract*—In this paper, we propose an efficient knowledge-based automatic model generation (KAMG) technique aimed at generating microwave neural models of the highest possible accuracy using the fewest accurate data. The technique is comprehensively derived to integrate three distinct powerful concepts, namely, automatic model generation, knowledge neural networks, and space mapping. For the first time, we simultaneously utilize two types of data generators, namely, coarse data generators that are approximate and fast (e.g., two-and-one-half-dimensional electromagnetic), and fine data generators that are accurate and slow (e.g., three-dimensional electromagnetic). Motivated by the space-mapping concept, the KAMG technique utilizes extensive coarse data, but fewest fine data to generate neural models that accurately match the fine data. Our formulation exploits a variety of knowledge neural-network architectures to facilitate reinforced neural-network learning from coarse and fine data. During neural model generation by KAMG, both coarse and fine data generators are automatically driven using adaptive sampling. The KAMG technique helps to increase the efficiency of neural model development by taking advantage of a microwave reality, i.e., availability of multiple sources of training data for most high-frequency components. The advantages of the proposed KAMG technique are demonstrated through practical microwave examples of MOSFET and embedded passive components used in multilayer printed circuit boards.**

***Index Terms*—Computer-aided design (CAD), modeling, neural networks, optimization.**

I. Introduction

The drive in the electronics industry for manufacturability driven design and time to market, coupled with ever-increasing circuit complexities and operating frequencies, demands powerful computer-aided design (CAD) methodologies [1], [2]. Modeling still remains a major bottleneck for efficient high-frequency CAD [3]. Artificial neural networks (ANNs) recently gained popularity as a fast and flexible vehicle to microwave modeling [1], [4]–[23]. Neural-network models are developed from measured or simulated microwave data through a process called training. Resulting neural models are used in place of CPU-intensive theoretical models for fast and accurate microwave design and optimization. Neural-network techniques have been used in the CAD of a variety of microwave components and circuits, e.g., embedded passives [4], striplines [5], transistors [6], [7], microstrip lines [8], CPW components [9], vias [10], bends [11], [13], filters [14]–[17], spiral inductors [18], high electron-mobility transistors (HEMTs) [19], amplifiers [20]–[22], mixers [23], etc.

Detailed theoretical models (e.g., three-dimensional electromagnetic (3-D EM) models of passive components, physics models of active devices) are accurate, but can be CPU intensive and slow. On the other hand, approximate models (e.g., empirical models) are fast, but their accuracy can be limited, i.e., accurate only in a specified frequency range or in a particular region of operation. High-level CAD operations such as Monte Carlo simulation and optimization of microwave circuits utilizing detailed models of individual circuit elements can be CPU prohibitive, while those using approximate models could yield imprecise solutions. Neural models are as fast as approximate models and as accurate as detailed EM/physics models [8], thus making ANN-based CAD an efficient alternative.

Reliable ANN-based CAD solutions need accurate neural models, which, in turn, require lots of accurate training data. For example, lots of accurate training data from a detailed 3-D EM simulator is needed for developing an embedded capacitor neural model with 3-D EM accuracy to be able to use the capacitor neural model for reliable circuit CAD [4]. Training-data generation is expensive as it involves both CPU time (for detailed model computations) and human time (for repetitive geometry changes), and can slow down neural model development. There is a recent trend in the EM-ANN area for investigating techniques that could potentially lower model development time by reduced use of accurate training data and lessened human involvement through automation of model development.

Several techniques have been developed to reduce the need for expensive data. Neural networks with knowledge such as the difference method (DM) [10], knowledge-based neural networks (KBNNs) [8], prior-knowledge input (PKI) network

Manuscript received September 16, 2002; revised February 14, 2003. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by Micronet—A Canadian Network of Centers of Excellence on Microelectronic Devices, Circuits, and Systems, and in part by Nortel Networks.

V. K. Devabhaktuni, B. Chattaraj, and Q.-J. Zhang are with the Department of Electronics, Carleton University, Ottawa, ON, Canada K1S 5B6.

M. C. E. Yagoub is with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N5.

Digital Object Identifier 10.1109/TMTT.2003.814318

[11], and space mapped neural networks (SMNNs) [13]–[15] utilize existing knowledge (e.g., empirical equations or models), thereby reducing the need for expensive training data. The automatic model generation (AMG) algorithm [4] shortens human time through automation of a multilayer perceptrons (MLPs) neural-network model development process.

For the first time, we propose a robust knowledge-based automatic model generation (KAMG) technique that takes advantage of multiple data generators and knowledge neural networks to generate microwave neural models of the highest possible accuracy using the fewest accurate training data. Motivated by the space-mapping concept [24]–[26], the proposed KAMG achieves efficient neural model generation through extensive use of approximate data (also known as coarse data) together with minimized use of accurate data (also referred to as fine data). A variety of knowledge neural-network architectures are exploited to enable both coarse and fine data to contribute toward reinforced neural-network learning of detailed or fine microwave behaviors. The KAMG technique incorporates stage-wise training and adaptive training-data sampling to facilitate automatic generation of neural models. Examples presented in this paper demonstrate increased efficiency of microwave neural model development by the proposed KAMG owing to its ability to simultaneously utilize multiple microwave simulators.

II. Overview of Neural Modeling

Let $\boldsymbol{x}$ represent an n-vector containing the external inputs (stimuli) and $\boldsymbol{y}$ represent an m-vector containing the outputs (responses) of a microwave-modeling problem. For example, $\boldsymbol{x}$ could contain length of an embedded capacitor and signal frequency, and $\boldsymbol{y}$ could contain corresponding S-parameters. Let $\boldsymbol{f}$ such that $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ represent the detailed theoretical microwave relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ to be modeled by a neural network. For example, $\boldsymbol{f}$ could be an EM relationship in the case of passive components or a semiconductor physics relationship in the case of active devices.

In general, the detailed theoretical relationship $\boldsymbol{f}$ can be accessible to microwave engineers in the form of simulation software $\boldsymbol{d} = \boldsymbol{g}(\boldsymbol{x})$, also referred to as the simulator. We introduce a new terminology for the simulator "$\boldsymbol{g}$" and refer to it as a "*data generator*." For a given input $\boldsymbol{x}$, the data generator $\boldsymbol{g}(\boldsymbol{x})$ can be used to compute the outputs $\boldsymbol{d}$. Data generation involves repetitive use of $\boldsymbol{g}$ to obtain sample pairs $(\boldsymbol{x}_k, \boldsymbol{d}_k)$, where k is the sample index. These sample pairs are split into training and validation data sets [10], [18]. We define L and V as index sets of training (learning) and validation data, respectively.

The purpose of neural-network modeling is to develop a fast neural-network model $\tilde{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{w})$ that accurately represents the input–output relationship $\boldsymbol{g}$ of interest. Here, $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{w})$ is a neural network trained to learn $\boldsymbol{g}$ from training data $(\boldsymbol{x}_k, \boldsymbol{d}_k)$, $k \in L$, and $\boldsymbol{w}$ is a weight vector containing the adjustable parameters inside the neural network. The objective of neural-network training is to adjust $\boldsymbol{w}$ such that the error between neural model responses $\tilde{\boldsymbol{y}}$ and training data outputs $\boldsymbol{d}$ is minimized. Validation data $(\boldsymbol{x}_k, \boldsymbol{d}_k)$, $k \in V$, is used to monitor the quality of the neural-network model during training.

Neural model development involves several sub-tasks like data generation, data preprocessing, neural network training, and validation. Previously, these sub-tasks have always been carried out manually in a sequential manner independent of one another. Such an approach, referred to as conventional training, requires both human experience and time. Our recent AMG algorithm [4] integrated all the sub-tasks in neural modeling into a unified process. Starting with zero training and validation data, neural model generation proceeds automatically in a stage-wise manner. It has been demonstrated in the case of MLP neural networks that AMG requires fewer training data and shorter model development time.

III. Proposed KAMG Technique

A. KAMG Notation

In practice, a given microwave input–output relationship to be modeled could be accessible in the form of different simulation tools (i.e., different $\boldsymbol{g}$'s). Some $\boldsymbol{g}$'s are implemented using detailed theories such as the full-wave EM equations, and are accurate, but can be slow. On the other hand, some $\boldsymbol{g}$'s are based on simplified theories, and are, therefore, approximate and fast. We define $\boldsymbol{g}_c$ and $\boldsymbol{g}_f$ as a coarse data generator and fine data generator, respectively. For example, $\boldsymbol{x}$ could include a length of an embedded capacitor and signal frequency, and $\boldsymbol{d}_c = \boldsymbol{g}_c(\boldsymbol{x})$ and $\boldsymbol{d}_f = \boldsymbol{g}_f(\boldsymbol{x})$ could represent corresponding S-parameters computed from the coarse, e.g., two-and-one-half-dimensional electromagnetic (2.5-D EM) and the fine (e.g., 3-D EM) data generators (simulators), respectively. We also define

$$\tilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c) \tag{1}$$

$$\tilde{\boldsymbol{y}}_s = \boldsymbol{h}_s(\boldsymbol{x}, \boldsymbol{w}_s) \tag{2}$$

$$\tilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_f) \tag{3}$$

as coarse, sub, and fine (overall) neural-network models, respectively. Here, $\boldsymbol{h}_c$, $\boldsymbol{h}_s$, and $\boldsymbol{h}_f$ are coarse, sub, and fine (overall) neural networks, and $\boldsymbol{w}_c$, $\boldsymbol{w}_s$, and $\boldsymbol{w}_f$ are corresponding weight vectors. The main objective of our proposed KAMG technique is to automatically generate a neural-network model $\tilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_f)$ that accurately matches fine data from $\boldsymbol{g}_f$ over the entire $\boldsymbol{x}$ space (input space) of interest. This is accomplished through extensive use of coarse training data from $\boldsymbol{g}_c$ and minimized use of expensive fine training data from $\boldsymbol{g}_f$.

B. KAMG Formulation

Automatic neural-model generation using the proposed KAMG technique is accomplished through three major phases described in the following sections.

1) Phase I: Generation of Coarse Neural-Network Model Using Coarse Data Generator: In the first phase, the objective of KAMG is to generate a coarse neural-network model $\tilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$. A neural network $\boldsymbol{h}_c$ is trained to learn the coarse microwave relationship using coarse training data from

$\boldsymbol{g}_c$. The training objective here is to minimize the difference between coarse neural model outputs and coarse data generator outputs, i.e.,

$$\min_{\boldsymbol{w}_c} \sum_{k \in L_c} \left\| \boldsymbol{h}_c(\boldsymbol{x}_k, \boldsymbol{w}_c) - \boldsymbol{d}_c^k \right\| \tag{4}$$

by adjusting $\boldsymbol{w}_c$ using neural-network training (optimization) algorithms, e.g., quasi-Newton [1]. The vector $\boldsymbol{d}_c^k = \boldsymbol{g}_c(\boldsymbol{x}_k)$ is the coarse training data output corresponding to an input $\boldsymbol{x}_k$, and L_c is the index set of coarse training data. In this phase, coarse training data from $\boldsymbol{g}_c$ is used extensively because $\boldsymbol{g}_c$ is fast and inexpensive. The training ensures that $\boldsymbol{h}_c$ captures the coarse $\boldsymbol{x}$–$\boldsymbol{y}$ relationship available in the form of $\boldsymbol{g}_c$, i.e., the neural-network model $\widetilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ accurately matches coarse data from $\boldsymbol{g}_c$ over the entire $\boldsymbol{x}$ space of interest.

The process of coarse neural model development including data generation using $\boldsymbol{g}_c$ and training of neural network $\boldsymbol{h}_c$ is achieved automatically through adaptive sampling and stage-wise training [4]. Data sets L_c and V_c are empty initially and are updated in every training-stage by automatic driving of $\boldsymbol{g}_c$ using a suitable simulator driver [4]. Initially, the n-dimensional $\boldsymbol{x}$ space of interest (i.e., neural model utilization range) is considered as a set of sub-regions. For the first stage of multistage training, coarse training data (L_c) and coarse validation data (V_c) are systematically generated covering all the sub-regions in a predefined way (e.g., star distribution [13]). The neural network $\boldsymbol{h}_c$ is trained with data samples in L_c, i.e., $(\boldsymbol{x}_k, \boldsymbol{d}_k)$ and $k \in L_c$, and is validated with samples in V_c, i.e., $(\boldsymbol{x}_k, \boldsymbol{d}_k)$ and $k \in V_c$. The worst-case validation sample index $k^* \in V_c$ with maximum error between the coarse neural model and coarse data is identified by

$$k^* = \arg\max_{k \in V_c} \left\| \boldsymbol{h}_c(\boldsymbol{x}_k, \boldsymbol{w}_c) - \boldsymbol{d}_c^k \right\|. \tag{5}$$

The sub-region to which this worst sample belongs is further divided into smaller sub-regions. Additional training and validation data are generated in the new regions again by automatic driving of $\boldsymbol{g}_c$. The training and validation index sets are augmented with new coarse data samples as

$$L_c^{\text{next}} = L_c \bigcup L_c^{\text{new}} \tag{6}$$

$$V_c^{\text{next}} = V_c \bigcup V_c^{\text{new}} \tag{7}$$

where L_c^{new} and V_c^{new} are index sets of newly generated coarse training and validation data, and L_c^{next} and V_c^{next} are used for the subsequent stage of the coarse neural network $(\boldsymbol{h}_c)$ training. The automatic coarse neural model development process including incremental data generation and stage-wise neural-network training is continued until $\widetilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ exceeds the user-specified coarse validation error [1], [4], i.e., $\boldsymbol{h}_c$ satisfactorily captures the coarse relationship $\boldsymbol{g}_c$.

2) Phase II: Initialization of Fine Neural-Network Model: The key idea behind the proposed KAMG technique is to systematically establish a framework that could enable simultaneous utilization of $\boldsymbol{g}_c$ and $\boldsymbol{g}_f$ toward efficient neural-network learning of fine behaviors. After the first phase, a satisfactorily close representation of $\boldsymbol{g}_c$ is available in the form of a coarse neural model $\widetilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$. What now needs to be created is a framework that facilitates reinforced neural-network learning from coarse information $\widetilde{\boldsymbol{y}}_c$ together with fine data from $\boldsymbol{g}_f$. Toward this end, we exploited a variety of knowledge neural-network architectures including DM, KBNN, PKI, and SMNN, and created a set of novel formulations for combining coarse and fine information. The objective of the second phase is to construct an initialized fine (overall) neural model, combining $\widetilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ with a suitable sub-neural model before using any fine data. To accomplish this, we generate a sub-neural model $\widetilde{\boldsymbol{y}}_s = \boldsymbol{h}_s(\boldsymbol{x}, \boldsymbol{w}_s)$ through a simple training of $\boldsymbol{h}_s$ reusing L_c and V_c from the first phase. The coarse and sub-neural networks are then combined to construct a fine (overall) neural network as

$$\widetilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_f) = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s) \tag{8}$$

where $\boldsymbol{w}_f$ includes both $\boldsymbol{w}_c$ and $\boldsymbol{w}_s$. The manner in which the fine (overall) neural network is constructed from $\boldsymbol{h}_c$ and $\boldsymbol{h}_s$ depends upon the KBNN architecture being used. However, since fine data from $\boldsymbol{g}_f$ has not been used thus far, $\widetilde{\boldsymbol{y}}_f$ in (8) is merely an initialized version of the fine (overall) neural-network model. This initial $\widetilde{\boldsymbol{y}}_f$ closely matches $\boldsymbol{g}_c$, and the KAMG can improve $\widetilde{\boldsymbol{y}}_f$ to accurately match $\boldsymbol{g}_f$ by further training of $\boldsymbol{h}_f$ using the fewest fine data from $\boldsymbol{g}_f$. In this section, fine (overall) neural model initialization through a simple sub-neural-network training is described for various knowledge ANN architectures.

Case 1: Initialization of KAMG–DM: Coarse data generator $\boldsymbol{g}_c$ could miss finer details in the original $\boldsymbol{x}$–$\boldsymbol{y}$ relationship to be modeled. Consequently, $\boldsymbol{g}_c$ and $\boldsymbol{g}_f$ differ over the entire $\boldsymbol{x}$ space of interest, and the difference could vary with $\boldsymbol{x}$. Here, we focus on this difference in an attempt to use coarse and fine information simultaneously toward fine learning. The DM-based neural-network architecture [10] is exploited. The coarse neural-network model $\widetilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ closely representing $\boldsymbol{g}_c$ has already been generated in the first phase. Considering this, the sub-neural network $\boldsymbol{h}_s$ is expected to learn the difference between coarse and fine data.

Neural-network $\boldsymbol{h}_s$ is initialized through a training process whose objective is to

$$\min_{\boldsymbol{w}_s} \sum_{k \in L_c} \|\boldsymbol{h}_s(\boldsymbol{x}_k, \boldsymbol{w}_s) - \boldsymbol{0}\| \tag{9}$$

by adjusting $\boldsymbol{w}_s$. In other words, $\boldsymbol{h}_s$ is forced to learn a $\boldsymbol{0}$ difference, which is a reasonable initialization for the difference, in the absence of any fine data. We then construct an initial fine (overall) neural-network model utilizing the coarse and sub-neural models as

$$\widetilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s) = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c) + \boldsymbol{h}_s(\boldsymbol{x}, \boldsymbol{w}_s) \tag{10}$$

where outputs of $\boldsymbol{h}_f$ are computed through a vector summation of outputs from $\boldsymbol{h}_c$ and $\boldsymbol{h}_s$. In this way, the initial fine (overall) network $\boldsymbol{h}_f$ would be as good as $\boldsymbol{h}_c$ (i.e., $\boldsymbol{g}_c$) in representing the given $\boldsymbol{x}$–$\boldsymbol{y}$ relationship, even before any expensive fine data is used.

Case 2: Initialization of KAMG–KBNN: In general, the coarse data generator $\boldsymbol{g}_c$ (e.g., empirical equations) can at best represent fine microwave behaviors $\boldsymbol{g}_f$ accurately in a sub-region of the $\boldsymbol{x}$ space of interest. In order to develop a fine (overall) neural-network model that can accurately represent $\boldsymbol{g}_f$

over the entire $\boldsymbol{x}$ space of interest, we propose using multiple coarse neural-network models. The KBNN architecture [8] is exploited, and the coarse neural model $\boldsymbol{h}_c$ generated in the first phase is incorporated into several knowledge layer neurons. The idea here is that the KBNN's weight parameters associated with several instances of $\boldsymbol{h}_c$ could be tweaked to build an accurate fine (overall) neural model.

Neural network $\boldsymbol{h}_s$ is initialized through a training process whose objective is to

$$\min_{\boldsymbol{w}_s} \sum_{k \in L_c} \left\| \boldsymbol{h}_s\left(\boldsymbol{x}_k, \boldsymbol{h}_{c1}(\boldsymbol{x}_k, \boldsymbol{w}_c), \boldsymbol{h}_{c2}(\boldsymbol{x}_k, \boldsymbol{w}_c), \ldots, \boldsymbol{h}_{cp}(\boldsymbol{x}_k, \boldsymbol{w}_c), \boldsymbol{w}_s\right) - \tilde{\boldsymbol{y}}_c^k \right\| \quad (11)$$

by adjusting $\boldsymbol{w}_s$ and keeping $\boldsymbol{w}_c$ fixed. Here, p is the number of knowledge layer neurons (i.e., the number of times the coarse neural model is repeated in the KBNN sub-network), and $\tilde{\boldsymbol{y}}_c^k = \boldsymbol{h}_c(\boldsymbol{x}_k, \boldsymbol{w}_c)$ is the coarse neural model output corresponding to an input $\boldsymbol{x}_k$. The sub-network $\boldsymbol{h}_s$ is used together with a region network $\boldsymbol{h}_r$ and a gating operator $\otimes$ [8] to construct the initial version of the fine (overall) neural-network model as

$$\begin{aligned}\tilde{\boldsymbol{y}}_f &= \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s) \\ &= \boldsymbol{h}_s\big(\boldsymbol{x}, \boldsymbol{h}_{c1}(\boldsymbol{x}, \boldsymbol{w}_c), \boldsymbol{h}_{c2}(\boldsymbol{x}, \boldsymbol{w}_c), \ldots, \boldsymbol{h}_{cp}(\boldsymbol{x}, \boldsymbol{w}_c), \boldsymbol{w}_s\big) \\ &\quad \otimes \boldsymbol{h}_r(\boldsymbol{x}, \boldsymbol{w}_r).\end{aligned} \quad (12)$$

Weight parameters $\boldsymbol{w}_r$ in $\boldsymbol{h}_r$ associated with each of the $\boldsymbol{h}_c$'s used in the fine (overall) network are initialized with equal values to ensure equal contribution of each $\boldsymbol{h}_c$ toward fine outputs $\tilde{\boldsymbol{y}}_f$ throughout the $\boldsymbol{x}$ space. The initial fine (overall) neural model $\tilde{\boldsymbol{y}}_f$ is as good as the coarse neural model $\tilde{\boldsymbol{y}}_c$ in representing the given $\boldsymbol{x}$–$\boldsymbol{y}$ relationship prior to using any fine data. It may be noted that, conceptually, the region network weights $\boldsymbol{w}_r$ are part of the sub-neural-network weights $\boldsymbol{w}_s$.

Case 3: Initialization of KAMG–PKI: In this case, the KAMG focuses on the relationship between coarse data generator outputs $\boldsymbol{d}_c$ and fine data generator outputs $\boldsymbol{d}_f$. The $\boldsymbol{d}_c$–$\boldsymbol{d}_f$ relationship could be influenced by the input vector $\boldsymbol{x}$ as well. The PKI architecture [11] is exploited and the purpose is to train a sub-neural-network $\boldsymbol{h}_s$ to learn the relationship $(\boldsymbol{x}, \boldsymbol{d}_c)$ versus $\boldsymbol{d}_f$. Since the fine data generator $\boldsymbol{g}_f$ has not been used thus far, $\boldsymbol{h}_s$ is made to learn the relationship $(\boldsymbol{x}, \boldsymbol{d}_c)$ versus $\boldsymbol{d}_c$, which is a reasonable initialization. Data $\boldsymbol{d}_c$ is available in the form of the coarse neural model $\tilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ generated in the first phase. The neural network $\boldsymbol{h}_s$ is initialized through a training process whose objective is to

$$\min_{\boldsymbol{w}_s} \sum_{k \in L_c} \left\| \boldsymbol{h}_s\left(\boldsymbol{x}_k, \tilde{\boldsymbol{y}}_c^k, \boldsymbol{w}_s\right) - \tilde{\boldsymbol{y}}_c^k \right\| \quad (13)$$

by adjusting $\boldsymbol{w}_s$.

Following this, an initial version of the fine (overall) neural network is constructed. The coarse neural network $\boldsymbol{h}_c$ trained in the first phase is used at the lower level of the hierarchical PKI structure. Inputs $\boldsymbol{x}$ together with outputs $\tilde{\boldsymbol{y}}_c$ from $\boldsymbol{h}_c$ are then supplied to the sub-network $\boldsymbol{h}_s$ at the higher level of the hierarchy. The initial fine (overall) neural-network model given by

$$\tilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s) = \boldsymbol{h}_s\big(\boldsymbol{x}, \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c), \boldsymbol{w}_s\big) \quad (14)$$

is as good as $\boldsymbol{h}_c$ (i.e., $\boldsymbol{g}_c$) in representing the $\boldsymbol{x}$–$\boldsymbol{y}$ relationship before using fine data from $\boldsymbol{g}_f$.

Case 4: Initialization of KAMG–SMNN: To better formulate the KAMG–SMNN problem, we define coarse and fine input vectors $\boldsymbol{x}_c$ and $\boldsymbol{x}_f$ (original input $\boldsymbol{x}$ itself), respectively. When a fine input $\boldsymbol{x}_f = \boldsymbol{x}$ is presented to the coarse data generator $\boldsymbol{g}_c$, an approximate or coarse output is computed. If a coarse input $\boldsymbol{x}_c$ corresponding to the fine input $\boldsymbol{x}_f$ can be found, such that $\boldsymbol{g}_f(\boldsymbol{x}_f) \cong \boldsymbol{g}_c(\boldsymbol{x}_c)$, then fast $\boldsymbol{g}_c$ could be used in place of CPU-intensive $\boldsymbol{g}_f$. The challenge here is to find a space mapping between $\boldsymbol{x}_f$ and $\boldsymbol{x}_c$ [24]–[26]. The SMNN architecture [13]–[15] is exploited and the sub-neural network $\boldsymbol{h}_s$ is made to learn the mapping between the above-mentioned input-spaces. In the absence of any fine data, a logical initialization for the space mapping can be $\boldsymbol{x}_c = \boldsymbol{x}_f$. As such, the neural network $\boldsymbol{h}_s$ is initialized through a training process whose objective is to

$$\min_{\boldsymbol{w}_s} \sum_{k \in L_c} \|\boldsymbol{h}_s(\boldsymbol{w}_k, \boldsymbol{w}_s) - \boldsymbol{x}_k\| \quad (15)$$

by adjusting $\boldsymbol{w}_s$.

Utilizing $\boldsymbol{h}_c$ and $\boldsymbol{h}_s$, an initial version of the fine (overall) neural network can now be constructed. The original or fine input $\boldsymbol{x}_f$ is supplied to $\boldsymbol{h}_s$ and corresponding $\boldsymbol{x}_c$ is computed. The coarse neural-network model $\boldsymbol{h}_c$ developed in the first phase is then used to compute the output vector corresponding to $\boldsymbol{x}_c$. The initial fine (overall) neural network model is given by

$$\tilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}_f, \boldsymbol{w}_c, \boldsymbol{w}_s) = \boldsymbol{h}_c\big(\boldsymbol{h}_s(\boldsymbol{x}_f, \boldsymbol{w}_s), \boldsymbol{w}_c\big). \quad (16)$$

Training of $\boldsymbol{h}_s$ in (15) ensures that the initial fine (overall) neural model $\tilde{\boldsymbol{y}}_f$ can be meaningful even without using any fine data, and is as good as the coarse neural model $\tilde{\boldsymbol{y}}_c$ (i.e., $\boldsymbol{g}_c$) in representing the given microwave input–output relationship.

Discussion: Through systematic training and construction of an initial fine (overall) neural network in the second phase, the KAMG technique ensures that

$$\boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s) \cong \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c) \quad (17)$$

when the sub-neural-network weight vector $\boldsymbol{w}_s$ is at its current or initialized value. In other words, the initialized $\tilde{\boldsymbol{y}}_f = \boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s)$ can be as good as the available coarse data $\boldsymbol{d}_c = \boldsymbol{g}_c(\boldsymbol{x})$ in representing the given $\boldsymbol{x}$–$\boldsymbol{y}$ relationship prior to using any fine data from $\boldsymbol{g}_f$.

3) Phase III: Refined Training of Fine (Overall) Neural Network Using Fine Data Generator: In the third phase, the KAMG technique emphasizes on capturing the finer microwave behaviors of the given $\boldsymbol{x}$–$\boldsymbol{y}$ relationship missed in the coarse neural-network model $\boldsymbol{h}_c$ (i.e., those missed by $\boldsymbol{g}_c$). The initial version of the fine (overall) neural network $\boldsymbol{h}_f$ is further trained using the fewest fine data from $\boldsymbol{g}_f$ to achieve the highest possible fine accuracy. This is required in order to elevate the neural model performance $\tilde{\boldsymbol{y}}_f$ currently close to $\boldsymbol{g}_c$ to an accuracy closest to that of $\boldsymbol{g}_f$. The training objective here is to minimize the difference between fine (overall) neural-network model outputs and fine data outputs, i.e.,

$$\min_{\boldsymbol{w}_s} \sum_{k \in L_f} \left\| \boldsymbol{h}_f(\boldsymbol{x}_k, \boldsymbol{w}_c, \boldsymbol{w}_s) - \boldsymbol{d}_f^k \right\| \quad (18)$$

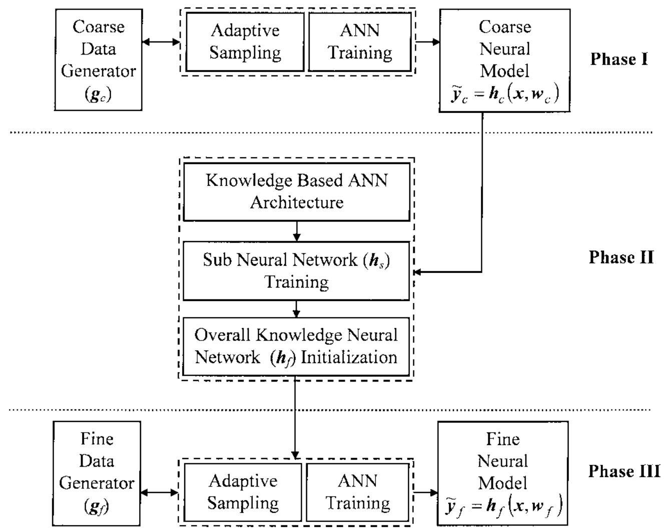


Fig. 1. Overall framework of the proposed KAMG technique showing various phases of neural model development simultaneously utilizing coarse and fine data generators.

by adjusting $\boldsymbol{w}_s$, keeping $\boldsymbol{w}_c$ fixed. In (18), $\boldsymbol{d}_f^k = \boldsymbol{g}_f(\boldsymbol{x}_k)$ represents the fine training data output corresponding to an input $\boldsymbol{x}_k$, $k \in L_f$, where L_f represents the index set of fine training data.

Similar to training in the first phase, the $\boldsymbol{x}$ space of interest is considered as a set of sub-regions. For the first stage, fine training data (L_f) and fine validation data (V_f) are systematically generated covering all the sub-regions. The initial fine (overall) neural network $\boldsymbol{h}_f$ is trained with data samples in L_f and validated with those in V_f. For subsequent stages of training, new data sample locations are selected by error-based sampling following (5), and L_f and V_f are augmented as in (6) and (7) by automatically driving $\boldsymbol{g}_f$ using a suitable simulator driver [4]. The overall framework of the proposed KAMG technique showing neural model development through simultaneous utilization of coarse and fine data generators is shown in Fig. 1. Our novel KAMG formulations specific to knowledge neural-network architectures DM, KBNN, PKI, and SMNN are illustrated in Figs. 2–5, respectively.

C. KAMG Algorithm

In this section, various steps involved in the proposed KAMG technique are summarized into a computational algorithm as follows.

Step 1) For a given microwave-modeling problem, select coarse and fine data generators $\boldsymbol{g}_c$ and $\boldsymbol{g}_f$. For example, 2.5-D and 3-D EM simulators can be used as $\boldsymbol{g}_c$ and $\boldsymbol{g}_f$, respectively, for neural-network modeling of EM components such as the embedded passives (e.g., capacitors and spiral inductors).

Step 2) Select an MLP neural network $\boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$. Perform error-based adaptive sampling and incremental coarse data generation together with stage-wise training [4] with a training objective $\min_{\boldsymbol{w}_c} \sum_{k \in L_c} \|\boldsymbol{h}_c(\boldsymbol{x}_k, \boldsymbol{w}_c) - \boldsymbol{d}_c^k\|$ to automatically generate a coarse neural-network model $\tilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$. Neural model development process

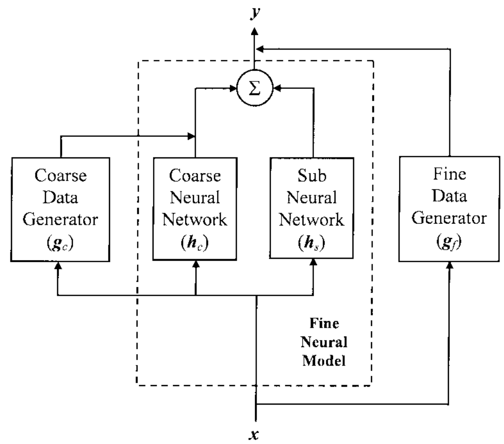


Fig. 2. Neural model development using the proposed KAMG formulation for the DM-based neural-network architecture resulting in our KAMG–DM. The portion within the dotted boundary represents the fine (overall) neural model for use in CAD.

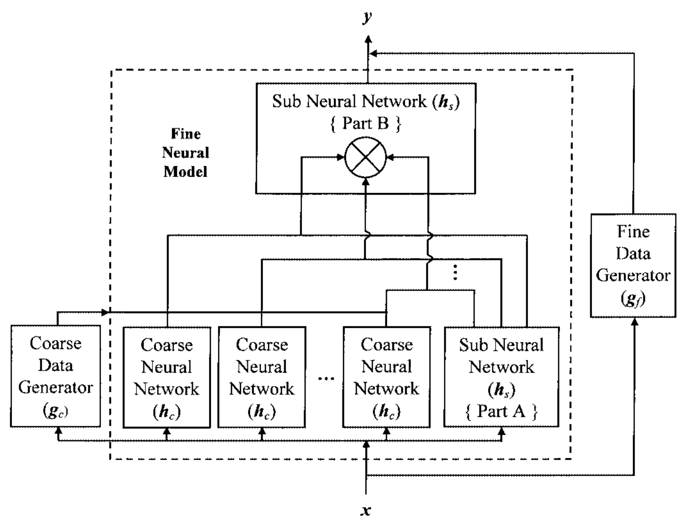


Fig. 3. Neural model development using the proposed KAMG formulation for the KBNN architecture resulting in our KAMG–KBNN. Part A of the sub-network represents the region and boundary layers, and Part B represents the gating network. The portion within the dotted boundary represents the fine (overall) neural model for use in CAD.

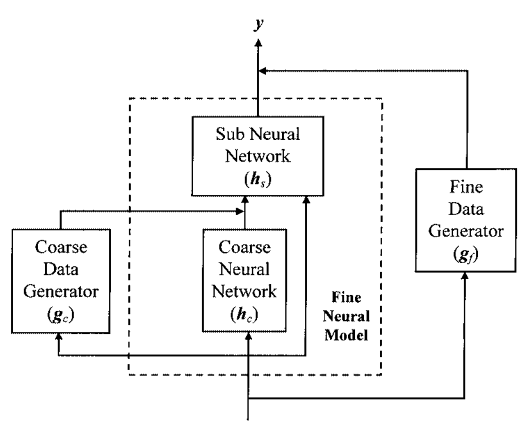


Fig. 4. Neural model development using the proposed KAMG formulation for the PKI neural-network architecture resulting in our KAMG–PKI. The portion within the dotted boundary represents the fine (overall) neural model for use in CAD.

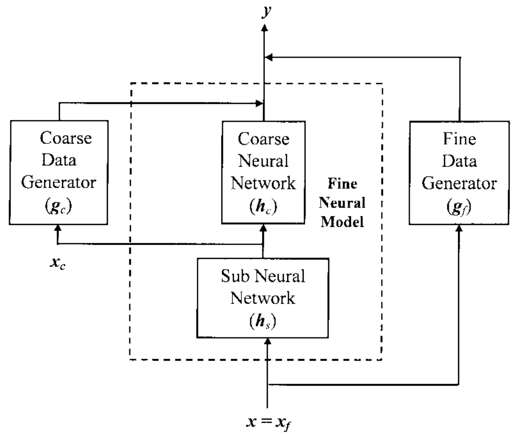


Fig. 5. Neural model development using the proposed KAMG formulation for the SMNN architecture resulting in our KAMG–SMNN. The portion within the dotted boundary represents the fine (overall) neural model for use in CAD.

is terminated when neural model responses from $\boldsymbol{h}_c$ are sufficiently close to coarse data outputs from $\boldsymbol{g}_c$, i.e., $\tilde{\boldsymbol{y}}_c = \boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ and $\boldsymbol{d}_c = \boldsymbol{g}_c(\boldsymbol{x})$ closely match in the $\boldsymbol{x}$ space of interest.

Step 3) Prepare the data index sets L_c and V_c for reuse in training and validation of sub-neural-network model. Data format for sub-neural-network training is different for different knowledge neural-network architectures. Coarse training and validation data are reformatted into ordered pairs $(\boldsymbol{x}_k, \boldsymbol{0})$ for KAMG–DM architecture (ignoring outputs $\boldsymbol{d}_c^k$), into $(\boldsymbol{x}_k, \boldsymbol{d}_c^k)$ for KAMG–KBNN architecture, into $((\boldsymbol{x}_k, \boldsymbol{d}_c^k), \boldsymbol{d}_c^k)$ for KAMG–PKI architecture, and into ordered pairs $(\boldsymbol{x}_k, \boldsymbol{x}_k)$ for KAMG–SMNN architecture.

Step 4) A sub-neural-network $\boldsymbol{h}_s(\boldsymbol{x}, \boldsymbol{w}_s)$ is trained and validated using the corresponding reformatted data sets from step 3. The training objective for sub-neural-network training is different for different knowledge neural-network architectures, as already defined in (9), (11), (13), and (15).

Step 5) Utilizing the coarse neural-network and sub-neural-network models $\boldsymbol{h}_c(\boldsymbol{x}, \boldsymbol{w}_c)$ and $\boldsymbol{h}_s(\boldsymbol{x}, \boldsymbol{w}_s)$ from steps 2 and 4, respectively, an initial version of the fine (overall) neural model $\boldsymbol{h}_f(\boldsymbol{x}, \boldsymbol{w}_c, \boldsymbol{w}_s)$ is constructed. Our formulation ensures that the neural model responses $\tilde{\boldsymbol{y}}_f$ from $\boldsymbol{h}_f$ are as good as $\tilde{\boldsymbol{y}}_c$ from $\boldsymbol{h}_c$ in the entire $\boldsymbol{x}$ space of interest prior to using any fine data.

REPEAT steps 3–5 for various knowledge neural-network formulations including KAMG–DM, KAMG–KBNN, KAMG–PKI, and KAMG–SMNN.

Step 6) Starting with the available version of the fine (overall) neural network $\boldsymbol{h}_f$, perform error-based adaptive sampling, incremental fine data generation, and neural-network training for a single stage. The training objective is $\min_{\boldsymbol{w}_s} \sum_{k \in L_f} \|\boldsymbol{h}_f(\boldsymbol{x}_k, \boldsymbol{w}_c, \boldsymbol{w}_s) - \boldsymbol{d}_f^k\|$, and fine training and validation data are generated by automatically driving $\boldsymbol{g}_f$. At the end of this step, a refined version of the fine (overall) neural-network model is available corresponding to each of the four knowledge architectures formulated within the proposed KAMG framework.

Step 7) Compute the validation error of the available fine (overall) neural model corresponding to each of the four KAMG-based knowledge architectures.

Step 8) If validation error of one or more of the fine models meets user specification, select the fine (overall) neural-network model with the least validation error and exit the algorithm. Provide this best neural-network model to the microwave designer for further use in circuit/system CAD. Otherwise, GO TO step 6 for an additional stage of refined training using fine data.

D. Discussion

The proposed KAMG is a comprehensive algorithm that integrates several advanced concepts in microwave CAD literature including AMG, knowledge neural networks, and space mapping into a unified framework. Our KAMG formulation exploits these concepts in an innovative manner, i.e., space mapping is now formulated for multiple data generators, the knowledge-ANN concept is now used to bridge the gap between coarse and fine information, and the AMG algorithm is applied to automate multiple ANN trainings involved. The KAMG facilitates reinforced neural-network learning from multiple sources of RF/microwave data (coarse and fine data generators) through knowledge-ANN architectures. Here, we provide a further insight to the KAMG algorithm by analyzing various special cases from a microwave perspective. In a case where coarse data equals fine data, the KAMG terminates with phase I alone and, as such, the existing AMG becomes a special case of the proposed KAMG with only a single phase of model development. In cases where coarse data differs from fine data, fine model initialization and refinement of the initial version of fine (overall) neural model using fine data (phases II and III) are essential. There are two possible situations: 1) if the difference is slender or simpler, very few fine data are utilized and phase III terminates quickly or 2) phase III of KAMG gets prolonged and the number of fine data needed depends upon the degree of the difference. An important aspect is that when coarse data retains basic behaviors in the corresponding fine data and misses only higher-order information, the proposed KAMG technique works in its most effective way. In an extreme (worst) case where there is a contradiction between available coarse and fine data generators, the KAMG technique is not recommended and use of AMG with fine data alone is appropriate.

In microwave CAD, detailed EM- or physics-based design and optimization are important, but can be prohibitively expensive in terms of CPU time. Fast neural models that accurately match fine data can effectively address this challenge. The KAMG technique uses a judicious mix of extensive coarse data (e.g., 2.5-D EM) and fewer fine data (e.g., 3-D EM) to automatically generate accurate neural models. In the following section, it is shown through practical microwave examples that the

KAMG neural models exceed user-specified accuracies (e.g., 1%) using fewest fine data compared to any other existing neural modeling techniques including the AMG. The KAMG neural models can be used in place of CPU-intensive detailed models for carrying out fast and reliable high-frequency circuit design and optimization. For even tighter accuracy specifications of a neural model, the KAMG automatically continues the fine training phase by repetitive execution of steps 6–8 and efficient use of additional amounts of fine data. In cases where the highest levels of accuracies are desired, neural-based CAD solutions that are much faster to obtain can serve as an initial design, starting from which original detailed models can be directly used for final refinement of the design. In this way, an otherwise expensive refinement cycle can now be substantially shortened because of a good starting point provided by KAMG neural models.

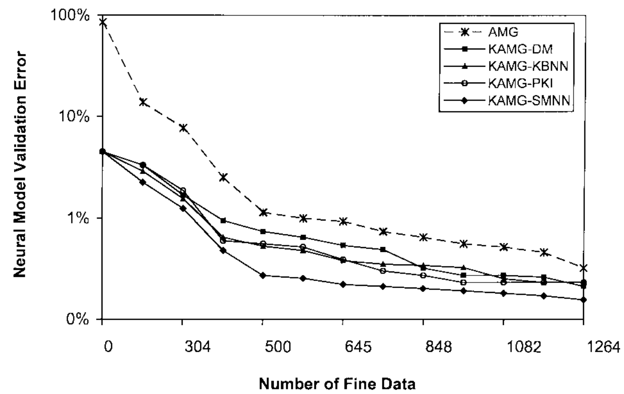


Fig. 6. Accuracy comparison of MOSFET neural models generated by the proposed KAMG and existing AMG techniques. All the KAMG neural models achieved better accuracies with fewer fine data as compared to MLP neural models from the AMG.

IV. EXAMPLES

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

This example illustrates development of MOSFET neural models with physics-level accuracy, but without using too much physics-based (expensive) fine training data. Neural network input $\boldsymbol{x}$ contains five parameters, namely: 1) channel length (L); 2) channel width (W); 3) oxide thickness (T_{ox}); 4) drain voltage (V_d); and 5) gate voltage (V_g). The drain current (I_d) is the only parameter in neural model output $\boldsymbol{y}$. A fast and approximate equivalent-circuit model in *HSPICE*[1] is used as a coarse data generator $\boldsymbol{g}_c$, while a CPU-intensive (slow), but accurate physics-based *MINIMOS* simulator[2] is used as a fine data generator $\boldsymbol{g}_f$.

The proposed KAMG technique is used for automatic generation of MOSFET neural models. In the first phase, a coarse neural model $\tilde{\boldsymbol{y}}_c$ is generated by an extensive use of coarse training data from $\boldsymbol{g}_c$ (3265 coarse samples). The neural model $\tilde{\boldsymbol{y}}_c$ predicts drain current with an average validation error of 4.50%. In the second phase, a sub-neural model $\tilde{\boldsymbol{y}}_s$ is developed and an initial fine (overall) neural network is constructed, corresponding to each of the four knowledge neural-network formulations, namely, KAMG–DM, KAMG–KBNN, KAMG–PKI, and KAMG–SMNN. In each case, it is ensured that the initial fine (overall) neural model is as good as $\tilde{\boldsymbol{y}}_c$ in representing the MOSFETs' $\boldsymbol{x}$–$\boldsymbol{y}$ characteristics, i.e., average validation error of each of the initial (fine) neural models is 4.50% prior to using any fine data. These initial fine (overall) neural networks are further trained using the fewest fine training data from $\boldsymbol{g}_f$ through automatic sampling and stage-wise training, as described in steps 6–8 of the KAMG algorithm.

As can be seen in Fig. 6, all the KAMG neural models achieved better accuracies with fewer fine data, as compared to MLP neural models from the existing AMG. For a user-specified neural model accuracy of 0.50%, the KAMG technique requires fewer fine data than conventional training (manual neural modeling approach) and the existing AMG, as shown in Table I. In this example, the KAMG algorithm terminated with the KAMG–SMNN neural model achieving the specified 0.50% accuracy with only 454 fine data. MOSFET drain currents computed using the fast KAMG–SMNN neural model are compared with physics-based fine data from *MINIMOS* in Fig. 7 and a good agreement is observed. Table II shows that

TABLE I
COMPARISON OF FINE DATA NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE MOSFET MODELS WITH 0.50% VALIDATION ERROR

Neural Modeling Technique	Number 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

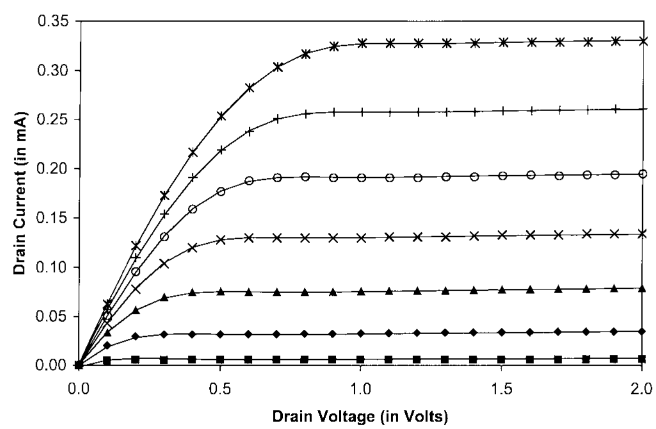


Fig. 7. Comparison of KAMG–SMNN neural model prediction of drain current (—) with physics-based *MINIMOS* data (symbols) for the MOSFET. Responses are shown for a particular geometrical configuration of the MOSFET for different values of drain and gate voltages.

[1] *HSPICE*, ver. 2001.2, Synopsis Inc., Mountainview, CA.

[2] *MINIMOS*, ver. 6.1, Inst. Microelectron., Tech. Univ. Vienna, Vienna, Austria.

TABLE II
ACCURACY COMPARISON BETWEEN MOSFET NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 454 FINE DATA

Neural Modeling Technique	Validation Error
Conventional training	3.24%
AMG	2.52%
Conventional DM	1.48%
Proposed KAMG-DM	0.95%
Conventional KBNN	0.98%
Proposed KAMG-KBNN	0.65%
Conventional PKI	0.92%
Proposed KAMG-PKI	0.60%
Conventional SMNN	0.84%
Proposed KAMG-SMNN	0.48%

the KAMG models outperform corresponding conventional knowledge models when only a few fine data are used.

B. Embedded Resistor Neural Model Development Using 2.5-D and 3-D EM Data Generators

Fast and accurate models representing 3-D EM behaviors of embedded passives are important for multilayer printed circuit board (PCB) design [27]. In this example, neural models of an embedded resistor are developed. The input vector $\boldsymbol{x}$ contains resistor length (L) and signal frequency (f), and the output vector $\boldsymbol{y}$ contains the real and imaginary parts of S-parameters $RS_{11}, IS_{11}, RS_{21}$, and IS_{21}. A fast 2.5- EM simulator[3] is used as the coarse data generator $\boldsymbol{g}_c$ and a CPU-intensive 3-D EM simulator[4] including finer EM effects is used as the fine data generator $\boldsymbol{g}_f$. Embedded resistor neural models are developed using the proposed KAMG, and other existing techniques such as the conventional training technique and AMG.

The first phase of the KAMG algorithm using extensive coarse training data (220 samples) resulted in a coarse neural model with 7.02% average validation error. Training and initialization in the second phase reusing the coarse data yielded initial fine (overall) embedded resistor neural models with accuracies close to that of the coarse neural model. These initial fine neural models are further trained using 3-D EM fine data in the third phase. Fig. 8 and Table III show that the proposed KAMG technique yields relatively accurate neural models, as compared to the existing techniques, when fewer fine data are available. In this particular example, the KAMG algorithm exited with the KAMG–PKI neural model exceeding user-specified accuracy of 0.50% with only 18 fine data. S-parameters of the resistor computed using the fast KAMG–PKI neural model accurately match with those from the slow *Ansoft-HFSS* simulations, as shown in Fig. 9.

[3]*SONNET-LITE*, ver. 7.0, Sonnet Software Inc., Liverpool, NY.
[4]*HFSS*, ver. 7.0.11, Ansoft Corporation, Pittsburgh, PA.

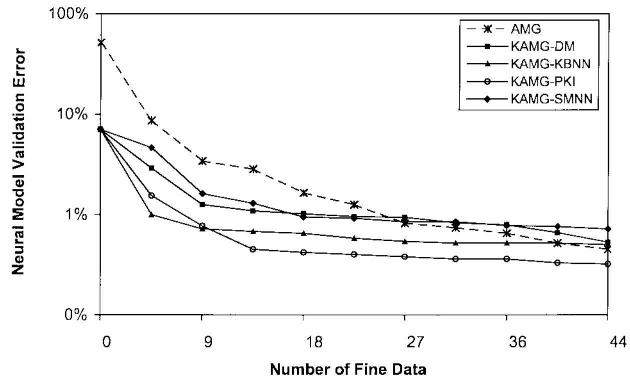


Fig. 8. Accuracy comparison of embedded resistor neural models generated by proposed KAMG and existing AMG techniques. When only a few fine data are available, all the KAMG neural models achieved better accuracies than the MLP neural models from the AMG.

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

Neural Modeling Technique	Validation 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%

C. Embedded Capacitor Neural Model Development Using 2.5-D and 3-D EM Data Generators

As mentioned earlier, consideration of detailed EM effects is mandatory for high-frequency and high-speed CAD. As such, accurate and fast EM-based neural models of embedded capacitors [28] including geometrical parameters as model inputs can significantly speedup design and optimization of RF/microwave circuits and systems. In this example, neural models of an embedded capacitor shown in Fig. 10 are developed. The input $\boldsymbol{x}$ 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 $\boldsymbol{y}$. A 2.5-D EM simulator is used as a coarse data generator and a 3-D EM simulator is used as a fine data generator. Attempts to develop capacitor neural models with 3-D EM accuracy using fine data from a 3-D EM simulator alone have proven to be computationally prohibitive, as each 3-D EM simulation needs a lot of CPU time. This led to a need for examining possibilities of using inexpensive coarse data together with fine data for efficient neural model development. The proposed KAMG algorithm is applied.

The first and second phases of training using 180 coarse data resulted in initial fine (overall) neural models with an average validation error of 3.02%. These initial fine neural networks are further refined in the third phase using 3-D EM fine data. In order to generate a neural model with a given accuracy, it can be

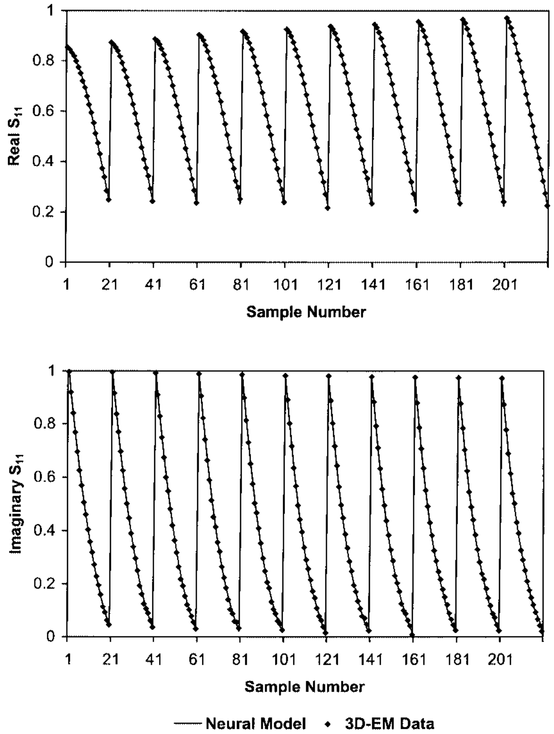


Fig. 9. Comparison of KAMG–PKI neural model prediction of the S-parameters with original 3-D EM *Ansoft-HFSS* data for the embedded resistor example. A total of 220 data samples corresponding to 11 different resistor lengths and 20 frequency samples per length are shown.

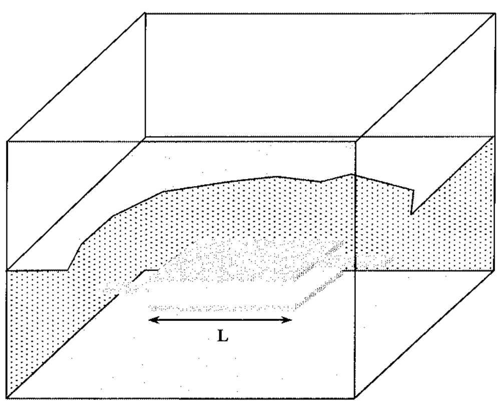


Fig. 10. 3-D EM representation of the embedded capacitor to be modeled using the proposed KAMG technique.

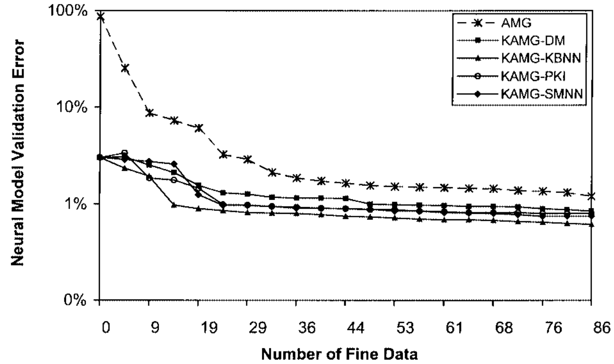


Fig. 11. Accuracy comparison of capacitor neural models generated by proposed KAMG and existing AMG techniques. All the KAMG neural models achieved better accuracies with fewer fine data as compared to MLP neural models from the AMG.

TABLE IV
COMPARISON OF FINE DATA AND CPU TIME NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE CAPACITOR MODELS WITH 1% VALIDATION ERROR

Neural Modeling Technique	Number of Fine Data	CPU-Time for Data Generation (in min)
Conventional training	125	625
AMG	96	480
Proposed KAMG-DM	48	240
Proposed KAMG-KBNN	14	70
Proposed KAMG-PKI	24	120
Proposed KAMG-SMNN	23	115

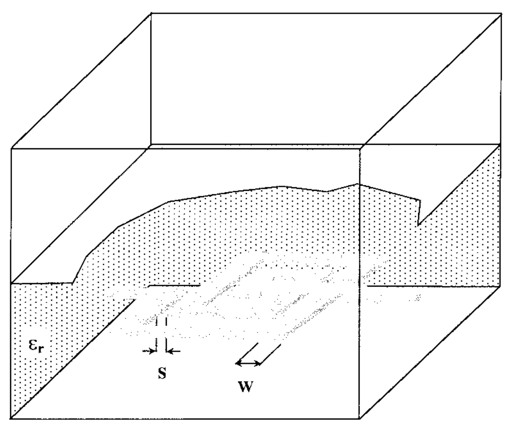


Fig. 12. 3-D EM representation of the spiral inductor to be modeled using the proposed KAMG technique.

seen from Fig. 11 that the proposed KAMG technique requires fewer fine data as compared to the existing AMG technique. Consequently, CPU time for data generation is significantly reduced, as can be seen in Table IV. Since data generation time is the major constituent of the total neural model development time, it can be concluded that neural model development using the proposed KAMG is faster than the existing neural modeling approaches. For a user-specified model accuracy of 1%, the KAMG algorithm stopped with the KAMG–KBNN model exceeding the specification with only 14 fine data.

D. Spiral Inductor Neural Model Development Using 2.5-D and 3-D EM Data Generators

Passives such as spiral inductors [29], [30] on a chip are critical for a successful integrated microwave system design. Although 3-D 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. For the spiral inductor shown in Fig. 12, we illustrate generation of neural models with EM-level accuracy, but without using too much 3-D EM-based

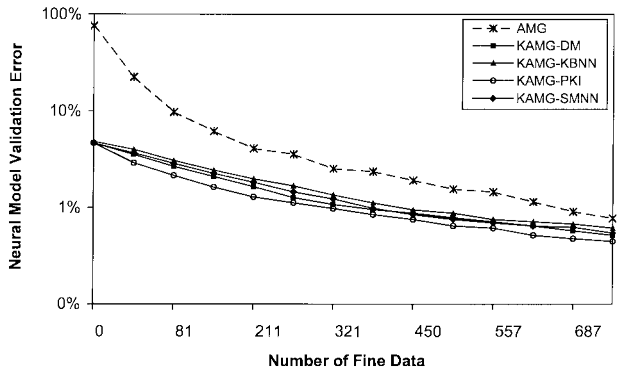


Fig. 13. Accuracy comparison of two-loop spiral inductor neural models generated by proposed KAMG and existing AMG techniques. All the KAMG neural models achieved better accuracies with fewer fine data, as compared to MLP neural models from AMG.

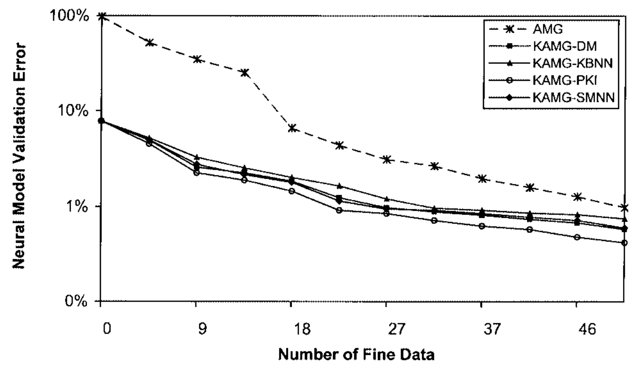


Fig. 14. Accuracy comparison of eight-loop spiral inductor neural models generated by the proposed KAMG and existing AMG techniques. All the KAMG neural models achieved better accuracies with fewer fine data, as compared to MLP neural models from AMG.

TABLE V
ACCURACY COMPARISON BETWEEN TWO-LOOP INDUCTOR NEURAL MODELS DEVELOPED BY VARIOUS MODELING TECHNIQUES USING 386 FINE DATA

Neural Modeling Technique	Validation Error
Conventional training	6.25%
AMG	2.34%
Proposed KAMG-DM	0.96%
Proposed KAMG-KBNN	1.12%
Proposed KAMG-PKI	0.85%
Proposed KAMG-SMNN	0.99%

TABLE VI
COMPARISON OF FINE DATA AND CPU TIME NEEDED BY VARIOUS NEURAL MODELING TECHNIQUES TO ACHIEVE EIGHT-LOOP INDUCTOR MODELS WITH 1% VALIDATION ERROR

Neural Modeling Technique	Number of Fine Data	CPU-Time for Data Generation (in min)
Conventional training	98	1764
AMG	52	936
Proposed KAMG-DM	27	486
Proposed KAMG-KBNN	32	576
Proposed KAMG-PKI	23	414
Proposed KAMG-SMNN	27	486

fine training data. A bridge drawn 0.1 mil above the spiral pattern is used as the inductor's connector pass. A 2.5-D EM simulator (*SONNET*) is used as coarse data generator and a 3-D EM simulator (*HFSS*) is used as fine data generator.

The KAMG algorithm is applied to automatically generate two-loop inductor neural models. Input $\boldsymbol{x}$ contains width (W), space (S), dielectric constant (ε_r), and frequency (f). Real and imaginary parts of S-parameters are the model outputs $\boldsymbol{y}$. Extensive coarse data (540 samples) are used in the first and second phases of the KAMG algorithm. As seen in Fig. 13, all the two-loop inductor neural models from the KAMG achieved better accuracies with fewer fine data, as compared to MLP neural models from the AMG. When the number of fine data used is the same, the KAMG models offer better accuracies than existing neural modeling techniques, as shown in Table V. Neural models of an eight-loop inductor are also generated using the proposed KAMG. Input $\boldsymbol{x}$ includes space (S) and frequency (f), and output $\boldsymbol{y}$ includes real and imaginary parts of S-parameters. The algorithm uses 184 coarse data for initial training and only a few fine data for model refinement. The KAMG algorithm terminated with the KAMG–PKI model achieving the user-specified 1% accuracy with 23 fine data. Fig. 14 demonstrates that KAMG yields relatively accurate models using fewer fine data compared to the AMG. Table VI shows that the proposed KAMG significantly reduces CPU time for data generation (i.e., model development time).

V. CONCLUSIONS

We have proposed a robust KAMG technique for automatic generation of neural-network models for microwave applications. Our study was aimed at providing the highest levels of efficiency and automation in developing microwave neural models, taking advantage of the fact that multiple sources of microwave training data often exist. The technique has integrated advanced concepts such as the AMG, knowledge neural networks, and space mapping into a powerful and systematic framework. Motivated by the space-mapping concept, the KAMG simultaneously exploits coarse and fine data generators for efficient neural model development. Our unified approach allows a variety of knowledge neural-network architectures to be exploited for accomplishing reinforced neural-network training from coarse and fine data. The KAMG is further strengthened by automation through adaptive sampling. The advantages of the KAMG technique are demonstrated through practical examples of a MOSFET and embedded passives used in multilayer PCBs and, in each example, microwave data from multiple and practical sources are utilized. For a given model accuracy, the proposed technique uses the fewest fine data compared to other existing techniques, including conventional training, AMG, and conventional knowledge methodologies. Fewer fine training data translates into significantly reduced CPU time for data generation, thus resulting in faster model development process. This study is significant for the growing demand of efficient CAD tools for microwave design and optimization.

REFERENCES

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

[2] M. B. Steer, J. W. Bandler, and C. M. Snowden, "Computer-aided design of RF and microwave circuits and systems," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 996–1005, Mar. 2002.

[3] K. C. Gupta, "Emerging trends in millimeter-wave CAD," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 747–755, June 1998.

[4] 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, Dec. 2001.

[5] V. K. Devabhaktuni, C. Xi, F. Wang, and Q. J. Zhang, "Robust training of microwave neural models," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 12, pp. 109–124, 2002.

[6] 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 Computer-Aided Eng.*, vol. 11, pp. 4–21, 2001.

[7] 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, Dec. 1998.

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

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

[10] ——, "EM-ANN models for microstrip vias and interconnects in dataset circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2495–2503, Dec. 1996.

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

[12] J. A. Jargon, K. C. Gupta, and D. C. DeGroot, "Applications of artificial neural networks to RF and microwave measurements," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 12, pp. 3–24, 2002.

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

[14] 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, Dec. 2000.

[15] J. W. Bandler, J. E. Rayas-Sanchez, and Q. J. Zhang, "Yield-driven electromagnetic optimization via space mapping-based neuromodels," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 12, pp. 79–89, 2002.

[16] S. Bila, Y. Harkouss, M. Ibrahim, J. Rousset, E. N'Goya, D. Baillargeat, S. Verdeyme, M. Aubourg, and P. Guillon, "An accurate wavelet neural-network-based model for electromagnetic optimization of microwave circuits," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 9, pp. 297–306, 1999.

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

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

[19] K. Shirakawa, M. Shimizu, N. Okubo, and Y. Daido, "Structural determination of multilayered large-signal neural-network HEMT model," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 1367–1375, Oct. 1998.

[20] 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 Trans. Microwave Theory Tech.*, vol. 48, pp. 2335–2344, Dec. 2000.

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

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

[23] ——, "Neural based dynamic modeling of nonlinear microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 2769–2780, Dec. 2002.

[24] 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, May 1994.

[25] M. H. Bakr, J. W. Bandler, R. M. Biernacki, S. H. Chen, and K. Madsen, "A trust region aggressive space mapping algorithm for EM optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 2412–2425, Dec. 1998.

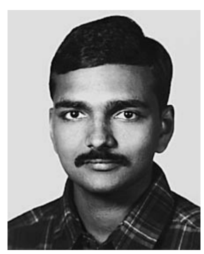
[26] M. H. Bakr, J. W. Bandler, N. Georgieva, and K. Madsen, "A hybrid aggressive space mapping algorithm for EM optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 47, pp. 2440–2449, Dec. 1999.

[27] K. L. Choi and M. Swaminathan, "Development of model libraries for embedded passives using network synthesis," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 249–260, Apr. 2000.

[28] R. Poddar and M. A. Brooke, "Accurate high speed empirically based predictive modeling of deeply embedded gridded parallel plate capacitors fabricated in a multilayer LTCC process," *IEEE Trans. Comp., Packag., Manufact. Technol.*, vol. 22, pp. 26–31, Feb. 1999.

[29] R. E. Collin, *Foundations for Microwave Engineering*, 2nd ed. New York: McGraw-Hill, 1992.

[30] 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, Sept. 1995.



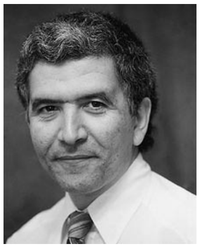
Vijay K. Devabhaktuni (S'97) received the B.Eng. degree in electrical and electronics engineering and the M.Sc. degree in physics from the Birla Institute of Technology and Science, Pilani, Rajasthan, India, in 1996, and is currently working toward the Ph.D. degree in electronics at Carleton University, Ottawa, ON, Canada.

He is currently a Sessional Lecturer with the Department of Electronics, Carleton University. His research interests include ANNs, computer-aided-design methodologies for VLSI circuits, and RF and microwave modeling techniques.

Mr. Devabhaktuni was the recipient of a 1999 Best Student Research Exhibit Award presented by Nortel Networks. He was a two-time recipient of the Ontario Graduate Scholarship for the 1999–2000 and 2000–2001 academic years presented by the Ministry of Education and Training, ON, Canada. He was the recipient of the 2001 John Ruptash Memorial Fellowship, which is awarded annually to an outstanding graduate student of the Faculty of Engineering, Carleton University. He was also the recipient of the 2001 Teaching Excellence Award in Engineering presented by the Carleton University Student's Association.

Biswarup Chattaraj (S'02) received the B.Tech. degree in electronics engineering from Nagpur University, Maharashtra, India, in 2000, the M.A.Sc. degree in electronics from Carleton University, Ottawa, ON, Canada, in 2002, and is currently working toward the Ph.D. degree in electrical and computer engineering from the University of Illinois at Chicago.

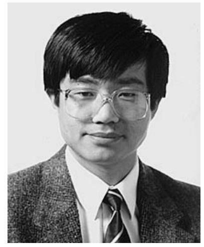
His research interests include microwave device and circuit modeling, CAD of very large scale integration (VLSI) and wireless circuits and systems.



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

He was with the Institute of Electronics, Université des Sciences et de la Technologie Houari Boumédienne, Algiers, Algeria, initially as an Assistant from 1983 to 1991, and then as an Assistant Professor from 1994 to 1999. From 1999 to 2001, he was with the Department of Electronics, Carleton University, Ottawa, ON, Canada, where he was involved with neural network applications in microwave areas. In 2001, he joined the School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada, where he is currently an Assistant Professor. His research interests include neural networks for microwave applications, CAD for linear and nonlinear microwave devices and circuits, and applied electromagnetics. He has authored over 80 publications on these topics in international journals and conferences. He coauthored *Conception de circuits linéaires et non linéaires micro-ondes*(Toulouse, France: Cépadues, 2000).

Dr. Yagoub is a member of the Association of Professional Engineers of the Province of Ontario, Canada.



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

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

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