

# **Neuro-Space Mapping Technique for Microwave Device Modeling and Its Use in Circuit Simulation and Statistical Design**

By

**Lei Zhang, M.A.Sc.**

A thesis submitted to  
The Faculty of Graduate Studies and Research  
in partial fulfilment of  
the degree requirements of  
**Doctor of Philosophy**

Ottawa-Carleton Institute for  
Electrical and Computer Engineering

Department of Electronics  
Carleton University  
Ottawa, Ontario, Canada

August 2008

Copyright ©  
2008 – Lei Zhang



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file    Votre référence*  
*ISBN: 978-0-494-43920-3*  
*Our file    Notre référence*  
*ISBN: 978-0-494-43920-3*

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

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

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

## **Abstract**

Nonlinear device modeling is an important area of computer-aided design for fast and accurate microwave design and optimization. The purpose of this thesis is to develop advanced modeling techniques for efficient generation of microwave device models. The proposed techniques combine the universal fitting capability of neural networks and the cost-effective optimization concept of space mapping, to achieve reliable device models for nonlinear circuit simulation and statistical design.

To meet the constant need for new device models due to the rapid progress in semiconductor technology, a neuro-space mapping (Neuro-SM) technique is firstly proposed. It automatically modifies the behavior of existing models to match new device behavior. Neuro-SM models improve the accuracy of existing device models while retaining the model speed. An advanced Neuro-SM formulation is proposed with analytical mapping representations and exact sensitivity analysis for efficient model training and evaluation. The analytical Neuro-SM model can be incorporated into high-level simulators to increase the speed and accuracy of circuit design. By mapping the existing equivalent circuit models to detailed device physics data, the Neuro-SM can also efficiently expand the scope of models in existing circuit simulators to include device physics behavior.

This Neuro-SM concept is expanded for efficient large-signal statistical modeling of nonlinear microwave devices. A linear statistical space mapping technique and a statistical neuro-space mapping technique are proposed. The proposed techniques introduce a new statistical space mapping concept that can expand a large-signal nominal model into a large-signal statistical model. The nominal model is extracted or trained from one complete set of dc, small-signal, and large-signal data. The behavior of a random device in the population is obtained by a mapping from that of the nominal device. In the linear statistical space mapping technique, we propose to use a simple linear dynamic mapping. In the statistical Neuro-SM, this mapping is nonlinear and represented by neural networks to overcome the accuracy limitations of the linear mapping in modeling large statistical variations among different devices. The statistical parameters of the model are extracted from dc and small-signal S-parameter data of many device samples. In this way, the proposed techniques allow efficient large-signal statistical model development with reduced expense of otherwise massive large-signal measurements for many devices.

Dedicated to

*my grandfather*

for all the love, patience, and understanding

## **Acknowledgements**

First of all, I would like to express my sincere thanks to my thesis supervisor, Dr. Q. J. Zhang, for his professional guidance, continued assistance, invaluable inspiration, motivation, and suggestions throughout the research work and the preparation of this thesis. This thesis would not have been complete without his expert advices and unfailing patience. I am also most grateful for his faith in this study especially in the sometimes-difficult circumstances in which it was written.

Special thanks to Dr. John W. Bandler and his research team from McMaster University. Their pioneering innovations and excellent developments on the space mapping technique provide strong fundamentals for this thesis. Dr. John Wood and Dr. Peter Aaen from the RF Division, Freescale Semiconductor, Inc., are also thanked for their active collaboration and valuable conversations in conducting this research.

My deep appreciation is given to Dr. Mustapha C.E. Yagoub from the University of Ottawa for his knowledgeable instruction and invaluable counsel. I would also like to thank all present and former colleagues in our research group for their enthusiasm, promotional skills, and helpful discussions.

Many thanks to Sylvie Beekmans, Scott Bruce, Lorena Duncan, Jacques Lemieux, Nagui Mikhail, Peggy Piccolo, Blazenka Power, Betty Zahalan, and all other staff and

faculty for providing excellent lab facilities and a friendly environment for my research and study in the department.

Finally, I wish to thank my parents for their endless love, trust, and encouragement throughout the years of my study. This thesis is dedicated to my grandfather, whom I will always love and wish to share happiness with, although he is no longer here to show his pride at my achievements. Thanks are also due to Xiaochen Liu, Yue Liu, Chao Lu, Zhen Xu, and others, too many to name here, for their lasting friendship and hearty support in fulfilling my dreams.

# Table of Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Symbols</b>	<b>xx</b>
<b>Nomenclature</b>	<b>xxvii</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Background and Motivation .....	1
1.2 Contributions of the Thesis.....	4
1.3 Outline of the Thesis.....	6
<b>Chapter 2: Literature Review</b>	<b>9</b>
2.1 Review of Neural Network Applications in RF/Microwave Modeling and Design.	9
2.2 Neural Network Based Microwave Modeling.....	11
2.2.1 Neural Network Structures .....	11
2.2.2 Neural Network Approaches for Linear/Nonlinear Microwave Modeling.....	14
2.2.3 Neural Network Model Development.....	16
2.2.4 Use of the Neural Models .....	19
2.3 Neural Networks With Prior Knowledge.....	21
2.3.1 Source Difference Method.....	22
2.3.2 Prior Knowledge Input.....	24

2.3.3 Knowledge-Based Neural Network .....	25
2.4 Space Mapping for Microwave Modeling and Design.....	27
2.4.1 Space Mapping Optimization .....	27
2.4.2 Space Mapping Based Neuromodeling.....	28
2.5 Nonlinear Microwave Device Modeling by Conventional Techniques .....	31
2.5.1 Physics-based Modeling Technique .....	31
2.5.2 Equivalent Circuit Modeling Technique.....	32
2.5.3 Table-Based Modeling Technique .....	34
2.5.4 Statistical Modeling of Nonlinear Microwave Devices.....	34
2.6 Nonlinear Microwave Device Modeling by Neural-Based Techniques .....	36
2.6.1 Neural Network Based Direct Modeling Approach.....	36
2.6.2 Neural Network Based Indirect Modeling Approach .....	38
2.6.3 Neural Network Based Statistical Modeling.....	40
2.7 Conclusions.....	41

**Chapter 3: Analytical Neuro-Space Mapping Technique for Nonlinear Microwave Device Modeling** **43**

3.1 Introduction.....	43
3.2 Problem Formulation .....	44
3.3 Proposed Analytical Formulation and Exact Sensitivity of the Neuro-SM Technique.....	47
3.3.1 Proposed Analytical Formulation of the Neuro-SM Model.....	47
3.3.2 Sensitivity Analysis of the Analytical Neuro-SM Model w.r.t. Mapping Neural Network Weights .....	51
3.3.3 Exact Sensitivity Analysis of the Analytical Neuro-SM Model w.r.t. Coarse Model Parameters .....	54
3.4 Proposed Training Algorithm for the Analytical Neuro-SM Model .....	58
3.4.1 Initialization of the Mapping Neural Network.....	58
3.4.2 Formal Training of the Mapping Neural Network.....	59

3.4.3 Use of the Trained Analytical Neuro-SM Model.....	63
3.5 Discussions .....	63
3.6 Application Examples.....	66
3.6.1 Analytical Neuro-SM Modeling of SiGe HBT .....	66
3.6.2 Analytical Neuro-SM Modeling of GaAs MESFET .....	70
3.6.3 Analytical Neuro-SM Modeling of a HEMT Trained with Physics-Based Device Data .....	77
3.6.4 Use of Neuro-SM Models in a Frequency Doubler Circuit.....	83
3.7 Conclusions.....	88
<b>Chapter 4: Statistical Space Mapping for Nonlinear Device Modeling: Linear Mapping Method</b>	<b>89</b>
4.1 Introduction.....	89
4.2 Proposed Statistical Space Mapped Model.....	90
4.2.1 Nominal Model .....	90
4.2.2 Statistical Space Mapping.....	90
4.2.3 Modeling Procedure.....	92
4.3 Application Examples.....	92
4.3.1 Large-Signal Statistical Model of a MESFET Device.....	92
4.3.2 Use of Statistical Space-Mapped Model in Amplifier Simulation .....	97
4.4 Conclusions.....	99
<b>Chapter 5: Statistical Neuro-Space Mapping (Neuro-SM) Technique for Large- Signal Statistical Modeling of Nonlinear Devices</b>	<b>100</b>
5.1 Introduction.....	100
5.2 Proposed Statistical Neuro-SM Technique.....	101
5.2.1 Proposed Statistical Neuro-SM Formulation .....	101
5.2.2 Proposed Statistical Neuro-SM for FET Modeling .....	106
5.2.3 Proposed Training of the Statistical Neuro-SM Model .....	108
5.2.4 Normality Mapping.....	112

5.2.5 Discussion .....	113
5.3 Proposed Training Algorithm for the Statistical Neuro-SM Model .....	115
5.4 Application Examples .....	118
5.4.1 Statistical Neuro-SM Modeling of A MESFET Device .....	118
5.4.2 Statistical Neuro-SM Modeling of a HEMT Device from a Physics-Based Device Simulator .....	130
5.4.3 Use of Statistical Neuro-SM Models in Two-Stage Amplifier Simulation.....	142
5.5 Conclusions.....	148
<b>Chapter 6: Conclusions and Future Work</b>	<b>149</b>
6.1 Conclusions.....	149
6.2 Suggestions on Future Directions .....	151
<b>Bibliography</b>	<b>155</b>

## List of Tables

Table 3.1:	Examples of sensitivity comparison in the HBT example. Sensitivity is done w.r.t. the mapping neural network weights and coarse model parameters. The Gummel-Poon model is used for mapping. ....	68
Table 3.2:	Comparison of model accuracy in the HBT example. The values are average errors between the model and training/testing data. The proposed analytical Neuro-SM can retain the same accuracy as the circuit-based Neuro-SM...	69
Table 3.3:	Neuro-SM Training time comparison between several training techniques for the HBT example. Training was done with dc data only. The proposed technique is the most efficient.....	69
Table 3.4:	Model evaluation time for 1000 Monte-Carlo analysis of 100 dc biases in the HBT example. Relative to the original coarse model, the computational overhead of the proposed analytical Neuro-SM is much less than the circuit-based Neuro-SM.....	69
Table 3.5:	Sensitivity comparison in the MESFET example. Sensitivity is calculated w.r.t. the mapping neural network weights and coarse model parameters. The Curtice model is used for the mapping. ....	75
Table 3.6:	Comparison of model accuracy in the MESFET example. The values are average errors between the model and training/testing data. The proposed analytical Neuro-SM can retain the same accuracy as the circuit-based Neuro-SM.....	76
Table 3.7:	Neuro-SM training time comparison between several training techniques for the MESFET example. Training was done with dc and S-parameter/harmonic data. The proposed technique is the most efficient....	76

Table 3.8:	Model evaluation time of dc and S-parameter sweeps at 150 biases, repeated for 1000 Monte-Carlo analyses in the MESFET example. Relative to the original coarse model, the computational overhead of the proposed analytical Neuro-SM is only marginal. ....	76
Table 3.9:	Sensitivity comparison in the HEMT example. Sensitivity is calculated w.r.t. the mapping neural network weights and coarse model parameters. The Chalmers model is used for mapping. ....	81
Table 3.10:	Comparison of model accuracy in the HEMT example. The values are average errors between the model and training/testing data. The proposed analytical Neuro-SM can retain the same accuracy as the circuit-based Neuro-SM. ....	82
Table 3.11:	Neuro-SM training time comparison between several training techniques for the HEMT example. Training was done with dc and bias-dependent S-parameter data. The proposed technique is the most efficient. ....	82
Table 4.1:	Means and standard deviations of the statistical space mapping parameters. ....	93
Table 4.2:	Correlation coefficients of the statistical space mapping parameters. ....	94
Table 5.1:	Correlation coefficients of the Gaussian variables $\bar{\phi}$ for the MESFET example. ....	120
Table 5.2:	Comparison of statistical accuracy and modeling efficiency between three different techniques for the MESFET example. ....	125
Table 5.3:	Cumulative probability error between the statistical model responses and the test data for the MESFET example. The proposed model has significantly better accuracy than the linear statistical space-mapped model as statistical variations become large. ....	129
Table 5.4:	Mean values of geometrical/physical parameters For HEMT device. ....	131
Table 5.5:	Correlation coefficients of the Gaussian variables $\bar{\phi}$ for the HEMT example. ....	133

Table 5.6: Comparison of statistical accuracy and modeling efficiency between three different techniques for the HEMT example..... 137

Table 5.7: Cumulative probability error between the statistical model responses and the test data for the HEMT example. The proposed model has significantly better accuracy than the linear statistical space-mapped model as statistical variations become large..... 141

## List of Figures

Figure 2.1:	Illustration of the feedforward multilayer perceptron (MLP) structure (from Zhang and Gupta [2]). Typically, the neural network consists of one input layer, one or more hidden layers, and one output layer. ....	13
Figure 2.2:	Physical models along with corresponding neural models of (a) a microstrip line and (b) a FET.....	14
Figure 2.3:	Flowchart of ANN approach for microwave modeling and circuit/system design. ....	20
Figure 2.4:	Structure of hybrid EM-ANN model utilizing the source difference method (from Zhang and Gupta [2])......	23
Figure 2.5:	Structure of the PKI model (from Zhang and Gupta [2])......	24
Figure 2.6:	The structure of the knowledge-based neural network (KBNN) (from Wang and Zhang [10]). Typically the KBNN model includes six layers: the input layer, the boundary layer, the region layer, the normalized region layer, the knowledge layer, and the output layer.....	26
Figure 2.7:	Structure of the space-mapped neural model (from Zhang and Gupta [2]).	29
Figure 2.8:	The conventional large-signal equivalent circuit model for MESFET (from Golio [67]). The resistances and inductances of the extrinsic components are constant. $I_{ds}$ , $I_{gs}$ , and $I_{dg}$ are nonlinear voltage controlled current sources. $C_{gd}$ and $C_{gs}$ are nonlinear capacitances.....	33
Figure 2.9:	(a) A physics-based MESFET modeled by (b) a neural network (from Zaabab, Zhang, and Nakhla [5]). The terminal currents and charges of the MESFET in (a) are represented by the neural network model as illustrated in (b) with the geometrical/physical and electrical parameters as the neural network inputs. ....	37

Figure 2.10: Large-signal FET modeling including adjoint neural networks trained by dc and bias-dependent S-parameters (from Xu, Yagoub, Ding, and Zhang [7]). .....	39
Figure 3.1: Structure of the general 2-port Neuro-SM nonlinear model, where a neural network $f_{ANN}$ is used to provide a mapping between coarse input signals and fine input signals. (a) Circuit-based Neuro-SM using neural network equations in controlled sources for the mapping. (b) Illustration of the proposed analytical Neuro-SM model for efficient model development without introducing extra equations to circuit simulation.....	45
Figure 3.2: Block diagram for dc and small-signal training of the proposed analytical Neuro-SM model.....	60
Figure 3.3: Block diagram for large-signal training of the proposed analytical Neuro-SM model. As observed, the input voltages are firstly passed to the mapping neural network to be mapped (modified) before being applied to the coarse model. FFT denotes fast Fourier transform, and IFFT means inverse FFT.	61
Figure 3.4: Comparison of the device dc data, the dc responses of the existing models (without mapping), and the Neuro-SM models in the HBT example. ....	67
Figure 3.5: Comparison of the dc current between the original ADS solution (device data), existing models (without mapping), and Neuro-SM models in the MESFET example. ....	72
Figure 3.6: Comparison of the S-parameters between the original ADS solution (device data), existing models (without mapping), and Neuro-SM models in the MESFET example. The S-parameters are at 2 biases of $(V_g, V_d)$ of $(-0.8V, 4V)$ and $(-0.2V, 1V)$ . ....	73
Figure 3.7: Comparison between the first three harmonic data and the HB response of the Neuro-SM models before/after HB refinement training in the MESFET example. Neuro-SM is applied to (a) Curtice model, and (b) Materka model. .....	74

Figure 3.8: Physical structure of a HEMT device used for generating fine data in *MINIMOS* to train Neuro-SM models..... 77

Figure 3.9: The dc comparison between the original HEMT data from *MINIMOS*, existing models (without mapping), and the Neuro-SM models in the HEMT example. The gate voltage  $V_g$  for all three models is from  $-0.5V$  to  $-0.1V$ . Existing models used for Neuro-SM are (a) Statz, (b) Curtice, and (c) Chalmers model. Training of Neuro-SM models was done using such dc data and the bias-dependent S-parameter data in Figure 3.10 simultaneously..... 78

Figure 3.10: S-parameter comparison between the original HEMT data from *MINIMOS*, existing models (without mapping), and the Neuro-SM models in the HEMT example. All plots show S-parameters in dB versus frequency in GHz. Comparison was done at 4 different dc biases at gate voltage ( $-0.4V$ ,  $-0.2V$ ) and drain voltage ( $0.2V$ ,  $2.4V$ ). Existing models used as coarse models for mapping are (a) the Statz model, (b) the Curtice model, and (c) the Chalmers model..... 80

Figure 3.11: A frequency doubler circuit. Both the MESFET models and the HEMT models developed with the Neuro-SM technique will be used in this circuit. .... 84

Figure 3.12: Comparison of the frequency doubler (with the MESFET models) HB solutions between the original ADS model, the coarse model, and the Neuro-SM model. (a) Second harmonic output power and conversion gain versus input power level at input frequency of 4 GHz. (b) Second harmonic output power versus output frequency with input power level of 1dBm. (c) Fundamental signal suppression at input power level of 1 dBm. Before mapping, the existing device model led to an inaccurate doubler solution. The Neuro-SM model improved the solution to be consistent with the original ADS solution..... 85

Figure 3.13: Comparison of the frequency doubler (with the MESFET models) HB solution between the original ADS model, the coarse model, and the Neuro-SM model. (a), (b), and (c) are similarly defined as in Figure 3.12, except that the coarse model used here for mapping here is the Materka model instead of the Curtice model of Figure 3.12.....	86
Figure 3.14: Frequency doubler (with the HEMT models) HB solutions using three Neuro-SM models (mapping of the Statz, the Curtice, and the Chalmers models). All the doubler solutions were obtained by <i>ADS</i> simulation. (a), (b), and (c) are similarly defined as in Figure 3.12, except that the transistor models used here were trained from the HEMT data generated from <i>MINIMOS</i> . Even though the original HEMT represented by the physics-based device simulator <i>MINIMOS</i> cannot be directly used in circuit simulators such as <i>ADS</i> , the proposed Neuro-SM technique makes it possible to have a HEMT model with device physics behavior in an <i>ADS</i> simulation.....	87
Figure 4.1: Two-port statistical space-mapped model.....	91
Figure 4.2: Example of output power (fundamental to third harmonics) v.s. input power of Monte-Carlo simulations with 100 devices using (a) the original ADS MESFET and (b) the proposed statistical space-mapped model.....	95
Figure 4.3: Example of output current of Monte-Carlo simulations with 100 devices using (a) the original ADS MESFET and (b) the proposed statistical space-mapped model.....	96
Figure 4.4: Three-stage amplifier circuit.....	97
Figure 4.5: Gain comparison of 1000 amplifier circuits using (a) the original ADS MESFET and (b) the proposed statistical space-mapped model. The distribution of the amplifier responses using our proposed statistical space-mapped model matches that of the original ADS results well, confirming our proposed method.....	98
Figure 5.1: Illustration of the proposed large-signal statistical Neuro-SM model. ....	103

Figure 5.2:	Calculation of the training error of the proposed statistical Neuro-SM model. Note that for different device samples, the proposed model uses the same $x$ - and $y$ -mapping neural networks but different values of the statistical variables to alter the nonlinear mapping. ....	110
Figure 5.3:	Flowchart for developing the proposed statistical Neuro-SM model. ....	116
Figure 5.4:	Mean values of the real part S-parameters at 2 biases from Monte-Carlo analyses of 300 small-signal simulations for the MESFET example. The comparison is done between the MESFET test data (o) and Monte-Carlo results using the statistical Neuro-SM model (—), the statistical model with no mapping (.), and the linear statistical space-mapped model (—). ....	122
Figure 5.5:	Standard deviations of the real part S-parameters at 2 biases from Monte-Carlo analyses of 300 small-signal simulations for the MESFET example. The comparison is done between the MESFET test data (o) and Monte-Carlo results using the statistical Neuro-SM model (—), the statistical model with no mapping (.), and the linear statistical space-mapped model (—). ....	123
Figure 5.6:	Cumulative probability distributions (CPD) of real part S-parameters at 1 GHz for 4 biases from Monte Carlo analyses of 300 small-signal simulations for the MESFET example. Such CPDs are used for a K-S test between the MESFET test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (—). ....	127
Figure 5.7:	Cumulative probability distributions (CPD) of (a) third order intermodulation interception (IP3), (b) power added efficiency, and (c) power gain at 4 input power levels from Monte Carlo analyses of 300 two-tone HB simulations for the MESFET example. Such CPDs are used for a K-S test between the MESFET test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (—). ....	128

- Figure 5.8: HEMT structure in *Medici* used for data generation of random device samples, where 10 process parameters are subject to random variations. 131
- Figure 5.9: Mean values of the real part S-parameters at 2 biases from Monte-Carlo analyses of 250 small-signal simulations for the HEMT example. The comparison is done between the HEMT test data (o) and the Monte-Carlo results using the statistical Neuro-SM model (—), the statistical model with no mapping (.), and the linear statistical space-mapped model (—). ..... 135
- Figure 5.10: Standard deviations of the real part S-parameters at 2 biases from Monte-Carlo analyses of 250 small-signal simulations for the HEMT example. The comparison is done between the HEMT test data (o) and the Monte-Carlo results using statistical Neuro-SM model (—), the statistical model with no mapping (.), and the linear statistical space-mapped model (—). ..... 136
- Figure 5.11: Cumulative probability distributions (CPD) of real part S-parameters at 10 GHz for 5 biases from Monte Carlo analyses of 250 small-signal simulations for the HEMT example. Such CPDs are used for the K-S test between the HEMT test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (—). ..... 139
- Figure 5.12: Cumulative probability distributions (CPD) of output power in dB at (a) fundamental, (b) second, and (c) third harmonic frequencies for 5 input power levels from Monte Carlo analyses of 250 HB simulations for the HEMT example. Such CPDs are used for the K-S test between the HEMT large-signal test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (—). ..... 140
- Figure 5.13: A two-stage amplifier whose transistors are represented by our statistical models. .... 142
- Figure 5.14: Transducer gain, main channel output power, and power added efficiency (PAE) from Monte Carlo analyses of 100 circuit envelop simulations of the

two-stage amplifier using (a) the original MESFET, (b) the proposed statistical Neuro-SM model, and (c) the linear statistical space-mapped model. The statistical behavior of the amplifier can be reproduced more accurately using the proposed model than using the linear statistical space-mapped model. .... 144

Figure 5.15: The output power spectrum of the two-stage amplifier for Monte Carlo analyses of 100 circuit envelope simulations using (a) the original MESFET, (b) the proposed statistical Neuro-SM model, and (c) the linear statistical space-mapped model. .... 145

Figure 5.16: Comparison of cumulative probability distributions for transducer gain and power added efficiency from 100 circuit envelope simulations between using the original MESFET (—) and the statistical models: (a) the proposed statistical Neuro-SM model (--) and (b) the linear statistical space-mapped model (--). .... 146

Figure 5.17: (a) Transducer gain and main channel output power, (b) power added efficiency (PAE), and (c) output power spectrum from Monte Carlo analyses of 100 circuit envelope simulations of the two-stage amplifier using the statistical Neuro-SM model for the HEMT. .... 147

## List of Symbols

$A$	Diagonal matrix containing the scaling factors defined as the inverse of the minimum-to-maximum range of the $I_D$ data
$B$	Diagonal matrix containing the scaling factors defined as the inverse of the minimum-to-maximum range of the $S_D$ data
$C_c$	Capacitance matrix of the coarse model
$d$	$N_y$ -vector containing the response of the device or the circuit under consideration
$d_{pk}$	$k^{th}$ device output of $d_p$ with the inputs $x$ of $p^{th}$ training sample
$e_p$	Error vector at input sample of $x_f = x_p$ , $p = 1, 2, \dots, N_p$ , for training of the space-mapped neural model
$E(w)$	Training error as a function of neural network internal weights $w$ . It is used to represent the difference between the outputs of the neural model and the device data. In Chapter 5, $E$ is used to represent the total training error of the statistical Neuro-SM model.
$f(x)$	Relationship between inputs $x$ and outputs $d$ in the original circuit or component problems

$f_{ANN}(\mathbf{x}, \mathbf{w})$	Neural network model representing the relationship between the inputs $\mathbf{x}$ and outputs $\mathbf{y}$
$f_{emp}(\mathbf{x})$	Empirical functions representing the relationship between the inputs $\mathbf{x}$ and the empirical/equivalent circuit model responses $\mathbf{y}$
$f_{stat}(\cdot)$	Statistical mapping equation
$g_{NN}(\cdot)$	$x$ -mapping neural network which maps the input signals of a random device sample to those of the nominal model
$G_c$	Conductance matrix of the coarse model
$h_{NN}(\cdot)$	$y$ -mapping neural network which further refines the output signals from the nominal model to produce the final model outputs
$i$	Terminal current signals (original signals) of the statistical model
$i_c$	Terminal current signals of the coarse model
$i_{c,NL}(t) _{v_c(t)}$	Nonlinear terminal current of the coarse model in terms of coarse input voltage signals $v_c(t)$
$i_f$	Terminal current signals of the fine model
$i_{f,NL}(t)$	Nonlinear terminal current of the fine model
$i_{nom}$	Terminal current signals (mapped signals) of the nominal model
$I(\cdot)$	The dc currents evaluated from the statistical model
$I_{c,L}(\omega_k)$	Currents of the coarse model at the generic harmonic frequency $\omega_k$ due to linear subcircuit

$I_{c,NL}(\omega_k)$	Currents of the coarse model at the generic harmonic frequency $\omega_k$ due to nonlinear subcircuit
$I_D$	The dc currents of the device
$I_f$	The dc response of the Neuro-SM model
$I_f(\omega_k)$	Currents of the Neuro-SM model at the generic harmonic frequency $\omega_k$
$I_{gc}, I_{dc}, I_{sc}$	Gate, drain, and source currents of a FET device
$I_{nom}$	The dc current signals of the nominal model evaluated at $V_{nom}$
$N_{bias}$	Total numbers of biases
$N_{freq}$	Total numbers of frequencies
$N_H$	Number of harmonics considered in HB simulation
$N_i$	Derivative orders of the voltage signals at port $i$ ( $i=1,2$ ) of the statistical model
$N_l$	Number of neurons in $l^{th}$ layer of MLP neural network
$N_{nomi}$	Derivative orders of the voltage signals at port $i$ ( $i=1,2$ ) of the nominal model
$N_p$	Number of data samples for neural network training
$N_T$	Number of time points
$N_x$	Number of external inputs to the neural network
$N_y$	Number of outputs from neural network
$P$	Approximate mapping from the fine model parameter space $x_f$ to the coarse model parameter space $x_c$

$P_o$	Index set for all training data for the Neuro-SM model training
$\mathbf{q}_{c,NL}(t) _{\mathbf{v}_c(t)}$	Nonlinear terminal charge of the coarse model in terms of coarse input voltage signals $\mathbf{v}_c(t)$
$\mathbf{q}_{f,NL}(t)$	Nonlinear terminal charge of the fine model
$Q_g, Q_d, Q_s$	Total charges of the gate, drain, and source electrodes of a FET device
$\mathbf{R}_c(\mathbf{x}_c)$	Corresponding responses of the coarse model
$\mathbf{R}_f(\mathbf{x}_f)$	Corresponding responses of the fine model
$\mathbf{S}(\cdot)$	S-parameters evaluated from the statistical model
$\mathbf{S}_D$	S-parameter data of a device
$\mathbf{S}_{ij}$	S-parameter from port $j$ to port $i$ of a device
$\mathbf{v}$	Terminal voltage signals (original signals) of the statistical model
$\mathbf{v}_c$ and $\mathbf{v}_c(t)$	Terminal voltage signals of the coarse model and its function representation in terms of time $t$
$\mathbf{v}_f, \mathbf{v}_f(t), \mathbf{v}_f(t_n)$	Terminal voltage signals of the fine model and its function representation in terms of continuous time $t$ as $\mathbf{v}_f(t)$ , and discrete time point $t_n$ as $\mathbf{v}_f(t_n)$
$\hat{\mathbf{v}}_f$	Adjoint voltages at the terminals of the analytical Neuro-SM model obtained by performing small-signal simulation of the nonlinear circuit
$\mathbf{v}_{in}(t)$	Dynamic input voltage of the DNN model
$\mathbf{v}_{in}^{(i)}(t)$	$i^{th}$ order derivative of $\mathbf{v}_{in}(t)$ for DNN modeling
$\mathbf{v}_{nom}$	Terminal voltage signals (mapped signals) of the nominal model
$\mathbf{v}_{out}(t)$	Dynamic output voltage of the DNN model

$v_{out}^j(t)$	$i^{th}$ order derivative of $v_{out}(t)$ for DNN modeling
$V$	The dc voltage signals of the $k^{th}$ device sample
$V_{c,Bias}$	Mapped dc bias for the coarse model through the neuro-space mapping
$V_{c,dc}$	Coarse dc voltage signals supplied to the coarse model
$V_{f,Bias}$	Bias of the fine model
$V_{f,dc}$	Fine dc voltage signals supplied to the fine (Neuro-SM) model
$\hat{V}_{f,dc}$	Adjoint dc port voltages of the analytical Neuro-SM model obtained by solving the original nonlinear circuit and its linear adjoint circuit
$V_f(\omega_k)$	Input voltages of the fine model at the generic harmonic frequency $\omega_k$
$\hat{V}_f(\omega_k)$	Adjoint voltages of the analytical Neuro-SM model at harmonic frequency $\omega_k$
$V_{nom}$	Mapped dc voltages from $V$ by the $x$ -mapping neural network
$w$	Vector containing all neural network weights, i.e., the training parameters of the neural network
$w_{nt}$	Weighting parameters of the normality mapping neural network
$w_x$	Weighting parameters of the $x$ -mapping neural network
$w_y$	Weighting parameters of the $y$ -mapping neural network
$W_N(n, k)$	Fourier coefficient for the $n^{th}$ time sample and the $k^{th}$ harmonic frequency
$W_N^*(n, k)$	Conjugate of $W_N(n, k)$

$\mathbf{x}$	$N_x$ -vector containing the inputs to the neural network or $N_x$ -vector containing parameters of a given device or a circuit
$\mathbf{x}'$	Input vector to the neural network containing the target model inputs and source model outputs in the PKI method
$\mathbf{x}_c$	Design parameters of the coarse model
$\mathbf{x}_f$	Design parameters of the fine model
$\mathbf{x}_{nom}$	Input signals of the nominal model
$\mathbf{y}$	$N_y$ -vector containing the outputs from the neural network
$\mathbf{y}'$	Outputs from the empirical model
$\Delta\mathbf{y}$	Difference between empirical approximation $\mathbf{y}'$ and training data $\mathbf{d}$
$\mathbf{y}_{nom}$	Output signals of the nominal model
$\mathbf{y}_{pk}(\mathbf{x}_p, \mathbf{w})$	$k^{th}$ neural model output with the inputs $\mathbf{x}$ of $p^{th}$ training sample
$\mathbf{Y}_c$	Small-signal Y-parameters of the coarse model
$\mathbf{Y}_{c,L}(\omega_k)$	Admittance matrix of the coarse linear subcircuit at $\omega_k$
$\mathbf{Y}_f$	Small-signal Y-parameters of the fine model
$\mathbf{Y}_{nom}(\omega)$	Y-parameters of the nominal model at frequency $\omega$
$z_{NN}$	Neural network function of the normality mapping neural network
$\mu$	Mean values
$\sigma$	Standard deviations
$\rho$	Correlation coefficients
$\phi$	Vector containing statistical parameters in statistical modeling

$\bar{\phi}$	Vector containing statistical variables with Gaussian distribution
$\phi^k$	$k^{\text{th}}$ random outcome of the statistical variables $\phi$ corresponding to the $k^{\text{th}}$ device
$\omega$	A frequency point
$\omega_k$	A generic harmonic frequency
$\mathcal{R}$	A circuit response
$\mathcal{R}(\cdot)$	Dynamic input-output relationship of the large-signal nominal model

## Nomenclature

AMG	Automated model generation
ANN	Artificial neural network
BPTT	Backpropagation through time
CAD	Computer-aided design
CPD	Cumulative probability distribution
CPU	Central processing unit
DNN	Dynamic neural network
DQPSK	Differential quaternary phase shift keying
EM	Electromagnetic
FET	Field effect transistor
GaAs	Gallium arsenide
GaN	Gallium Nitride
HB	Harmonic balance
HBT	Heterojunction bipolar transistor
HEMT	High electron mobility transistor
IP3	Third order intermodulation interception
KBNN	Knowledge-based neural network
K-S test	Kolmogorov-Smirnov goodness-of-fit test

LCP	Liquid crystalline polymer
MESFET	Metal-semiconductor field effect transistor
MLP	Multilayer Perceptron
Neuro-SM	Neuro-space mapping
PAE	Power added efficiency
PKI	Prior knowledge input
RBF	Radial basis function
RF	Radio frequency
RNN	Recurrent neural network
SiGe	Silicon Germanium
SM	Space mapping
TDNN	Time-delayed neural network
VLSI	Very large scale integration

# **Chapter 1: Introduction**

## **1.1 Background and Motivation**

Today's radio frequency (RF)/microwave design faces the challenges of increasing complexity, tighter component tolerances, and shorter design cycles. As a result, the demand for faster, more accurate, and cost-effective computer-aided design (CAD) techniques in the RF/microwave area becomes more and more urgent. It is desired to use electromagnetic (EM)/physics based simulations to achieve design accuracy. However, such simulations are in general computationally expensive. Modeling with EM/physics accuracy but much faster than direct EM/physics simulations has become an important research direction [1]. This thesis addresses an important aspect in this direction, i.e., large-signal modeling of nonlinear microwave devices.

Nonlinear device modeling is an important area of CAD, and many device models have been developed [1]. Due to rapid development in the semiconductor industry, new devices constantly evolve. Models that were developed to fit previous devices may not fit new devices well. There is an ongoing need for new models. The challenges for CAD researchers are not only to develop more models, but also to introduce new CAD methods, so the task of developing models becomes more efficient and systematic. The latter aspect is the motivation of this thesis.

Most traditional RF/microwave modeling techniques for active devices are based on equivalent-circuit models or parametric characterization (black-box models) [1], requiring trial and error based topology modification or extensive dc and RF characterization. On the other hand, the use of more detailed models, e.g., physical models based on analytical expressions or numerical algorithms [1], is necessary to achieve improved large-signal designs and to relate the yield and performance of the designs to the fabrication process. However, many of the analytical models lack the detail and fidelity of their numerical counterparts, and the computation of the numerical models is usually slow. Microwave technology innovation desires generalized and efficient techniques that can accurately model different types of newly evolved devices.

Recently, artificial neural networks (ANNs) have been recognized as an important vehicle in the microwave computer-aided design area in addressing the growing challenges of designing next generation microwave devices, circuits, and systems [2]-[4]. ANN models can be trained to learn electromagnetic and physics behavior. Furthermore, trained ANNs can be used in high-level circuit and system design allowing fast optimization including electromagnetic and physics effects in components. In the nonlinear modeling area, there has been a significant amount of interest in exploring the potential of ANN based methods for modeling active devices and nonlinear circuits. The I-Q model in [5] uses pure ANN to provide transistor terminal currents and charges. It has been applied to metal-semiconductor field effect transistor (MESFET) modeling [5] and large-signal high electron mobility transistor (HEMT) modeling [6]. The adjoint neural network method in [7] enables neural network model for large-signal devices to be

trained with dc and bias-dependent S-parameter data. The recurrent neural network method in [8] uses a discrete time-domain formulation to model nonlinear circuits and devices. These methods represent important steps towards automating the device modeling process. However, because the neural networks have to learn the device behavior from scratch without using an existing device formula, more training data is needed, or the reliability of the model will be low. Since there already exist a vast body of device models, how to utilize the existing models and to use neural networks to complement what is missing in the existing models becomes an important research topic.

One solution is knowledge-based neural networks [2], [9]-[12], which exploit existing microwave empirical functions or equivalent circuit models (the knowledge) together with the neural network to create an overall model. The empirical function/equivalent circuit part is computationally efficient, and can be used to simplify neural network learning. The neural network part will be trained from accurate microwave data to recover any characteristics, which may have been missed by the knowledge part. Such models need less data to achieve the same accuracy than pure neural networks without any knowledge. Extrapolation capability can also be improved because of the embedded knowledge.

Among different knowledge-based techniques, the space-mapped neural network [12] is one of the most efficient structures. It is based on an advanced optimization concept, the space-mapping (SM) concept proposed in [13], which has successfully achieved substantial computational speedup in otherwise expensive optimizations of microwave components and circuits. By establishing a mathematical link between the coarse and fine

models, the space-mapped neuromodeling directs the bulk of the CPU-intensive computations to the coarse model, while preserving the accuracy offered by the fine model. The technique has been applied to passive modeling or small-signal device modeling, achieving fast and accurate models for devices such as bends, high temperature superconductor filters, embedded passives in multilayer printed circuits, and other linear components [12]. However, the application of such space-mapped neuromodeling into large-signal modeling of nonlinear microwave devices still remains an open topic.

## **1.2 Contributions of the Thesis**

The main objective of this thesis is to develop efficient and systematic techniques for nonlinear microwave device modeling, where accurate device models can be automatically achieved from a computational process to maximally reduce human trial and error efforts. The developed model should be able to correctly represent the dc, small-signal, and large-signal characteristics of the nonlinear device, and effectively capture the statistical behavior of the device due to process variations. In addition, the model should be formulated such that it can be conveniently incorporated into existing circuit simulators for high-level circuit simulation and yield design. In this thesis, the following substantial works are presented, combining neural networks with the space mapping concept for efficient nonlinear device modeling:

- (1) A new neuro-space mapping (Neuro-SM) technique [14]-[16] is presented to meet the constant need for new device models due to rapid progress in semiconductor technology. It aims to automatically modify the behavior of existing models to match new device behavior. The proposed Neuro-SM model

retains the speed of the existing device model while improving model accuracy. An advanced formulation of Neuro-SM is proposed with analytical mapping representations and exact sensitivity analyses to achieve fast model training and evaluation. A 2-phase training algorithm utilizing gradient optimization with analytical sensitivity is also developed for efficient training of the analytical Neuro-SM models. After being trained, the analytical Neuro-SM model can be incorporated into high-level simulators to increase the speed and accuracy of circuit design. By mapping the existing equivalent circuit models to detailed device physics data, the Neuro-SM can also efficiently expand the scope of models in existing circuit simulators to include device physics behavior.

- (2) For the first time, an efficient statistical space mapping concept is introduced for large-signal statistical modeling of nonlinear microwave devices [17]. It can expand a large-signal nominal model for a nominal device into a large-signal statistical model for a given statistical device population. The nominal model is extracted or trained from one complete set of large-signal data of the nominal device. The statistical property is achieved by a dynamic mapping between the behavior of the nominal model and that of the statistical samples of the given population of devices. The parameters in the mapping, which are statistical parameters, can be extracted from dc and small-signal S-parameter data of many device samples. The proposed statistical space-mapped model can approximate the large-signal statistical characteristics using only one set of large-signal data. This helps

to efficiently develop large-signal statistical models while reducing the expense of otherwise massive large-signal measurements for many devices.

- (3) Another statistical modeling technique, called statistical neuro-space mapping [18], is also proposed for large-signal statistical modeling of nonlinear microwave devices. It is an advance over the linear statistical mapping technique, using nonlinear mapping to overcome the accuracy limitations of the linear mapping in modeling large statistical variations among different devices. For a given population of device samples, the nominal device model is determined from dc, small-signal, and large-signal data. The behavior of a random device in the population is obtained by a nonlinear mapping from that of the nominal device. The unknown mapping function is represented by neural networks trained using dc and small-signal data of various devices in the population. A novel statistical mapping is formulated by introducing a compact set of statistical variables to control the mapping to map from the nominal device towards different devices in the population. A new training method is proposed for simultaneous statistical parameter extraction and neural network training. It is demonstrated that, for small or large statistical variations, the proposed technique is able to provide accurate large-signal statistical models using a minimal amount of expensive large-signal data.

### **1.3 Outline of the Thesis**

The thesis is organized as follows.

In chapter 2, an overview of the literature is presented. Neural network applications for RF/microwave modeling and design are firstly reviewed. The procedure of neural model development is described. Knowledge-based neural networks for efficient and robust neural modeling are presented. Space mapping methodology and applications in RF/microwave area are discussed, and reviews of existing nonlinear device modeling techniques are conducted.

Chapter 3 introduces the analytical neuro-space mapping technique for efficient large-signal modeling of nonlinear microwave devices. A novel analytical formulation is proposed, where the mapping between the existing device model with coarse accuracy and the overall Neuro-SM model with fine accuracy is analytically achieved for dc, small-signal, and large-signal simulation and sensitivity analysis. A 2-phase training algorithm is derived for efficient model development. Application examples on modeling heterojunction bipolar transistor (HBT), MESFET, and HEMT devices and the use of Neuro-SM models in harmonic balance simulations are demonstrated.

In chapter 4, a new statistical space mapping technique is presented for large-signal statistical modeling of nonlinear microwave devices. The use of a large-signal nominal model to represent the average performance of a population of random device sample and a dynamic mapping network to characterize the statistical variations around the nominal is presented. The statistical parameters of the proposed model are defined as the mapping coefficients of the dynamic mapping network, and their extractions from dc and bias-dependent S-parameter data are described. Preliminary examples of a MESFET device modeling and its use in the statistical design of a three-stage amplifier circuit are included,

demonstrating that the statistical space-mapped model can approximate the large-signal statistical characteristics using only one set of large-signal data.

Another advanced large-signal statistical modeling technique, statistical neuro-space mapping, is discussed in detail in Chapter 5. To overcome the accuracy limitations of the linear dynamic mapping in modeling large statistical variations among different devices, the proposed statistical Neuro-SM uses nonlinear mapping represented by neural networks, which are trained using dc and small-signal data of various devices in the population. A novel statistical mapping is formulated, and a new training algorithm is proposed for simultaneous statistical parameter extraction and neural network training. The proposed technique is confirmed by statistical modeling of microwave transistor examples, and use of the models in statistical analyses of a two-stage amplifier.

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

## **Chapter 2: Literature Review**

### **2.1 Review of Neural Network Applications in RF/Microwave Modeling and Design**

The drive in the microwave industry to meet the demands of high manufacturability and fast design cycles has created a need for efficient modeling and design techniques. Statistical analysis and yield optimization that take into account manufacturing tolerances, model uncertainties, variations in the process parameters, etc., are also widely accepted as indispensable components of circuit design methodology [19]-[22]. Detailed EM/physics models of passive/active components can be an important step towards a design for first-pass success, but the models are computationally intensive. In the past decade, significant advances have been made in the exploitation of artificial neural networks [2] as an unconventional alternative to modeling and design tasks in RF/microwave CAD. Neural networks are computationally efficient, and their ability to learn and generalize from data allows model development even when component formulas are unavailable. Neural models are much faster than detailed EM/physics models [2], [23]-[25], more accurate than polynomial and empirical models [26], allow more dimensions than table lookup models [27], and are easier to develop when a new device/technology is introduced [28]. Once developed, these neural network models can

be used in place of computationally intensive EM/physics models of passive and active components [5], [6], [10], [29] to speed up microwave circuit design.

Important work has been done by microwave researchers demonstrating the ability of neural networks to accurately model a variety of microwave components, such as microstrip interconnects [10], vias [9], spiral inductors [29], field effect transistor (FET) devices [10], [30], HBT devices [31], HEMT devices [32], filters [33], [34], amplifiers [5], [8], coplanar waveguide (CPW) circuit components [35], mixers [8], antennas [36], embedded passives [23], [24], packaging and interconnects [37], etc. Neural networks have also been used in circuit simulation and optimization [5], [38], [39], signal integrity analysis and optimization of very-large-scale-integration (VLSI) interconnects [37], [40], microstrip circuit design [41], process design [42], synthesis [5], [43], microwave impedance matching [44], and behavioural modeling of nonlinear RF/microwave subsystems [45]. These pioneering works have established the framework of neural modeling technique in both component and the circuit/system level of microwave applications.

A variety of ANN structures has been developed to address different modeling scenarios [2]. Pure ANNs such as multilayer perceptron (MLP) neural network [2], recurrent neural network (RNN) [8][33], and time-delayed neural network (TDNN) [46] are used to directly model the linear/nonlinear behavior while knowledge-based neural network modeling techniques [2], [9]-[12] are suitable for efficient behavior modeling when empirical models exist as prior knowledge. Recent developments in ANN techniques for microwave modeling include automated model generation (AMG) [47],

[48], neural network enabled sensitivity analysis [7], dynamic behavioral modeling of nonlinear devices and circuits [49], and neuro-space mapping for nonlinear device modeling [14]. A new family of neural-network based bidirectional and dispersive behavioral models for nonlinear RF/microwave subsystems were proposed in [45] using a non-recursive MLP neural network in the frequency domain. Recursive MLP neural networks have been used in nonlinear time-series analysis as in [50], [51]. Other applications of ANNs are accurate parameter estimation for nonlinear device modeling [6], layout-level synthesis of RF inductors and filters in liquid crystalline polymer (LCP) substrates for Wi-Fi applications [52], and wide-band dynamic modeling of power amplifiers using radial-basis function (RBF) neural networks [53]. Research and development in the area are still continuing in developing ANN based methodologies for advanced linear/nonlinear microwave modeling and circuit optimization. More recently, ANN techniques for analysis of multilayered shielded microwave circuits [54], effective design of waveguide dual mode filter [55], state-space DNN modeling for high-speed IC applications [56], parallel automatic model generation technique for neural network based microwave modeling [57], and modeling nonlinear circuit/system behavior [58], [59] have also been proposed.

## **2.2 Neural Network Based Microwave Modeling**

### **2.2.1 Neural Network Structures**

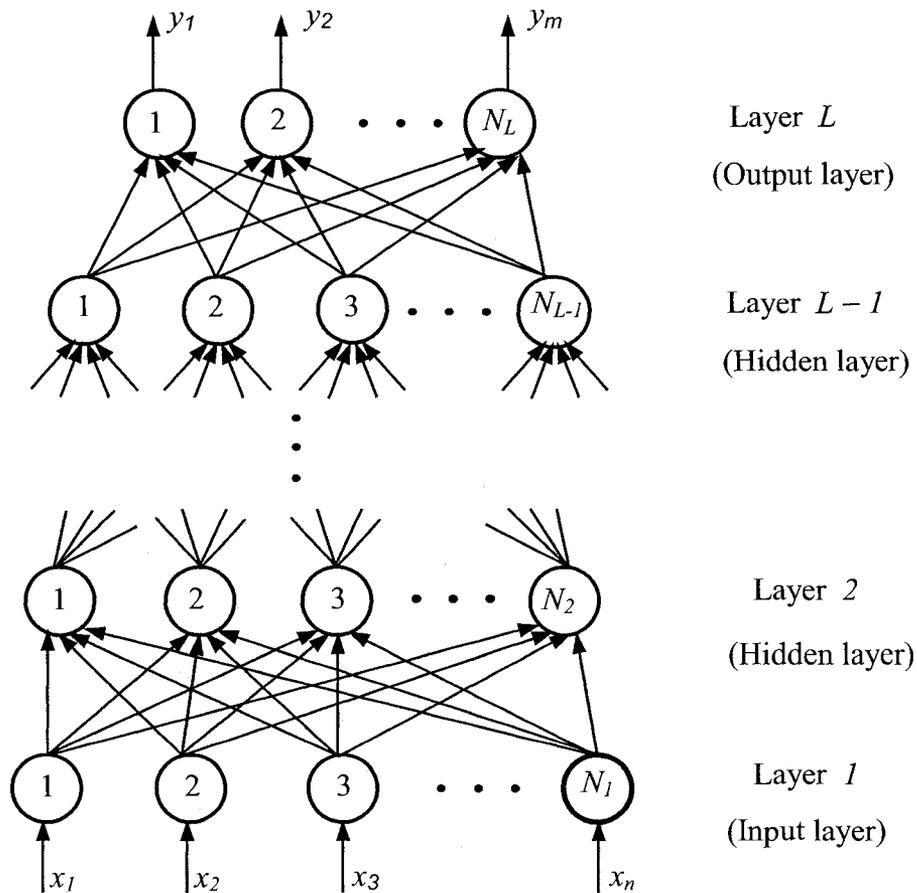
Various neural network structures, such as MLP neural networks [2], RBF neural networks [2], wavelet neural networks [2], recurrent neural networks [8], and dynamic

neural networks [49], have been developed in the neural network community to deal with different scenarios of microwave modeling. Among these variations, the feedforward neural network is a basic type of neural network capable of approximating generic continuous and integrable functions. An important class of feedforward neural networks is multilayer perceptrons [2]. MLP neural models are widely used in microwave device modeling and circuit/system design. Typically, the MLP neural network consists of an input layer, one or more hidden layers, and an output layer, as shown in Figure 2.1.

Figure 2.2 illustrates the application of MLP neural networks in modeling a microstrip line and a FET device. The geometrical/physical parameters are considered as the inputs of the neural network model. The outputs of the neural network model are the electrical parameters (i.e., the inductances and capacitances defined as  $L_s$ ,  $L_m$ ,  $C_s$ ,  $C_m$ ) of the resulting equivalent circuit model for the microstrip line, or the electrical responses (i.e., the S-parameters defined as  $S_{11}$ ,  $S_{12}$ ,  $S_{21}$ ,  $S_{22}$ ) for the FET device.

In general, the selection of an appropriate neural network structure normally starts by identifying the nature of the input–output relationship of a given application. The modeling of microwave components in the frequency domain is usually formulated with static parameters for neural network inputs and outputs. Such problems can be solved using MLP, RBF, and wavelet networks [2]. RBF and wavelet networks can be used when the microwave problem exhibits highly nonlinear and localized phenomena (e.g., sharp variations). Time-domain dynamic behavior of nonlinear microwave devices or circuits can be represented using recurrent neural networks [8], [33] and dynamic neural networks [49]. One of the most recent research directions in the area of microwave-

oriented ANN structures is the knowledge-based network [8]-[12], which combines existing engineering knowledge (e.g., empirical equations and equivalent circuit models) with neural networks.



**Figure 2.1: Illustration of the feedforward multilayer perceptron (MLP) structure (from Zhang and Gupta [2]). Typically, the neural network consists of one input layer, one or more hidden layers, and one output layer.**

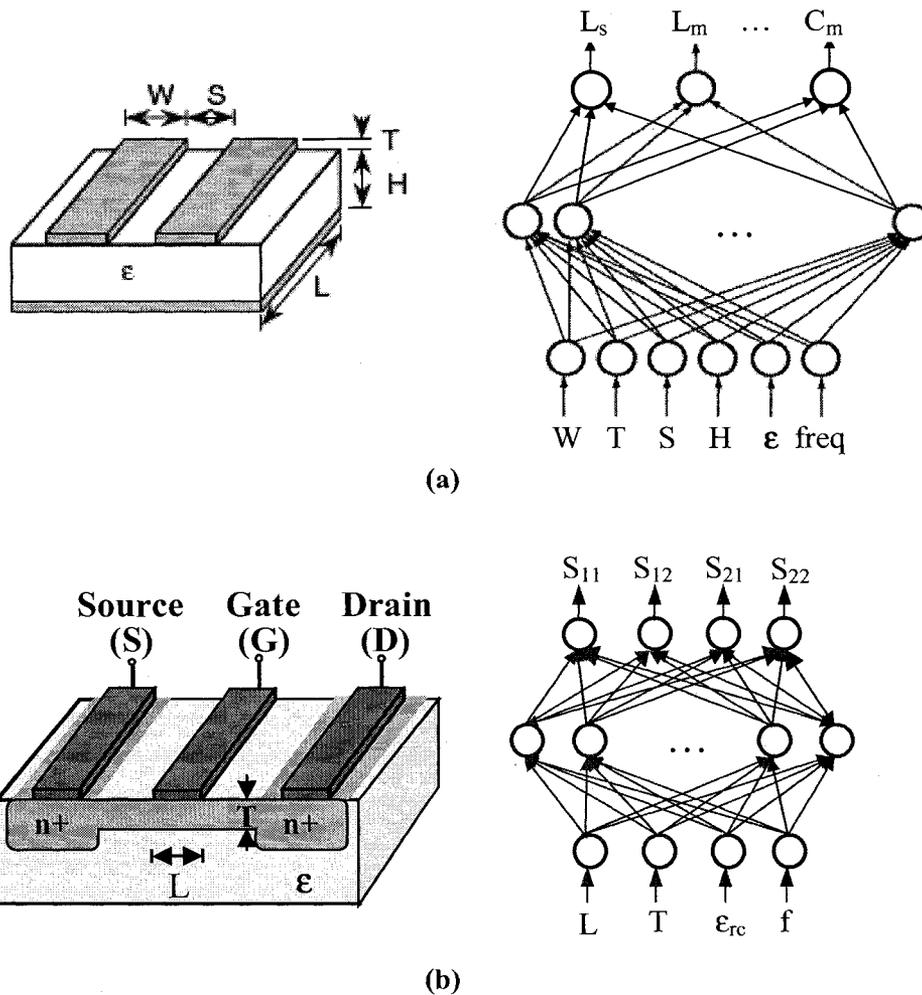


Figure 2.2: Physical models along with corresponding neural models of (a) a microstrip line and (b) a FET.

### 2.2.2 Neural Network Approaches for Linear/Nonlinear Microwave Modeling

ANN models can be developed in either the frequency domain [23], [39], [45] or time domain [8], [24], [33], [46], [49] to catch the steady state or transient responses. For EM modeling, a frequency-domain neural model usually uses geometrical or physical parameters (e.g., length and dielectric permittivity of an embedded capacitor) and signal frequency as model inputs, and the corresponding Y-parameters or S-parameters as

model outputs. The trained model can represent Y-parameters or S-parameters of a passive component with the speed of empirical models but with accuracy near detailed EM models, thus can be used in place of a CPU intensive EM simulator during frequency-domain simulation and optimization. Time-domain models are also important for CAD applications such as minimization of signal delay and crosstalk in high-speed VLSI interconnects. A time-domain neural model can produce current/voltage relationships of a given passive component/circuit in terms of the geometrical/physical parameters and time. For example, an ANN model can be trained to learn the coefficient values of the Y-parameter transfer functions of an EM structure, with physical/geometrical parameters as inputs. A state space model is then formulated to express the time-domain responses using coefficients estimated by the trained ANN model [24]. Recently, a time-domain modeling approach using recurrent neural networks has also been addressed for 2-port passive EM structures such as rectangular waveguide and microstrip low-pass filter [33].

For general nonlinear modeling problems, the inputs and outputs of nonlinear components/circuits are related by differential equations and the relationship is not algebraic. There are two different categories of ANN based approaches to model such a relationship. The first category uses a combination of equivalent circuit and ANN model [7], where we rely on the known circuit topology to define the dynamics and the ANN to define the unknown nonlinearity. The training data for the combined circuit/ANN model can be dc, bias-dependent S-parameters, or large-signal harmonic data. In the second category, the continuous time-domain formulation of dynamic neural networks [49] and

the discrete time-domain formulation of recurrent neural networks [8] are used to directly represent the entire model, including both the dynamic effect and nonlinearity. The DNN is formulated as a reduced order representation of the original circuit, and is trained with large-signal harmonic data to capture the nonlinear dynamics of the circuit. The RNN is formulated in discrete time domain with feedback from output to input. RNN models can be trained to learn the dynamics in both transient and steady state stages, using the backpropagation through time (BPTT) method with the input and output waveforms of the original circuit as the training data.

Although different types of neural networks exist for different applications, they have to be trained by component/circuit data before being used in circuit simulation and design. A general flow of neural network model development will be discussed in the following subsection.

### 2.2.3 Neural Network Model Development

Let  $\mathbf{x}$  be an  $N_x$ -vector containing parameters of a given device or a circuit, e.g., gate length and gate width of a FET device; or geometrical and physical parameters of a transmission line circuit. Let  $\mathbf{d}$  be an  $N_y$ -vector containing the response of the device or the circuit under consideration, e.g., drain current of the FET; or S-parameters of the transmission line circuit. The relationship between  $\mathbf{x}$  and  $\mathbf{d}$  may be nonlinear and multidimensional. This kind of relationship is represented by

$$\mathbf{d} = \mathbf{f}(\mathbf{x}) \tag{2.1}$$

in the original EM/physics problems.

A neural network model can be used to represent such a relationship, by being trained through a set of  $\mathbf{x}$ - $\mathbf{d}$  sample pairs, called training data, which is generated from original EM/physics simulations or measurement. Let the neural network model be represented by a nonlinear function  $f_{ANN}$  as

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

where  $\mathbf{y}$  is a vector containing the outputs from the neural network model, and  $\mathbf{w}$  is a vector containing all neural network weights, which will be adjusted during the neural network training process to make it best match the training data [2].

Neural network training [2] is an important step in neural model development. ANN models cannot accurately represent the component/circuit behavior until they are trained by  $\mathbf{x}$ - $\mathbf{d}$  data. As defined above, the inputs  $\mathbf{x}$  are the component/circuit parameters (e.g., geometric, physical, bias, frequency, etc.) that affect the responses, while the outputs  $\mathbf{y}$  are normally characterized as real and imaginary parts of S-parameters for passive component models, or currents and charges for large-signal device models. A basic description of the training objective is to determine  $\mathbf{w}$  such that the difference between neural model outputs  $\mathbf{y}$  and desired outputs  $\mathbf{d}$  from simulation/measurement,

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

is minimized. Here  $d_{pk}$  is the  $k^{th}$  element of vector  $\mathbf{d}_p$ ,  $y_{pk}(\mathbf{x}_p, \boldsymbol{w})$  is the  $k^{th}$  output of the neural network model when the input  $\mathbf{x}$  presented to the network is  $\mathbf{x}_p$ , where  $p$  is the index of the training samples. Since  $E(\boldsymbol{w})$  is a nonlinear function of the adjustable (i.e., trainable) weight parameters  $\boldsymbol{w}$ , iterative algorithms are often used to efficiently explore the  $\boldsymbol{w}$  space, beginning with an initialized value of  $\boldsymbol{w}$  and then iteratively updating it.

Neural network training algorithms commonly used in RF/microwave applications include gradient-based training techniques [2] such as backpropagation, conjugate-gradient, and quasi-Newton methods. Global optimization methods [2] such as simulated annealing and genetic algorithms can be used for increased quality of neural network training but at the cost of increased training time. There are two categories of the neural network training process known as sample-by-sample training and batch-mode training. In sample-by-sample training, also called online training,  $\boldsymbol{w}$  is updated each time a training sample is presented to the network. In batch-mode training, or offline training,  $\boldsymbol{w}$  is updated after all the training data (or samples) are used. In the RF/microwave case, batch-mode training is usually more effective.

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

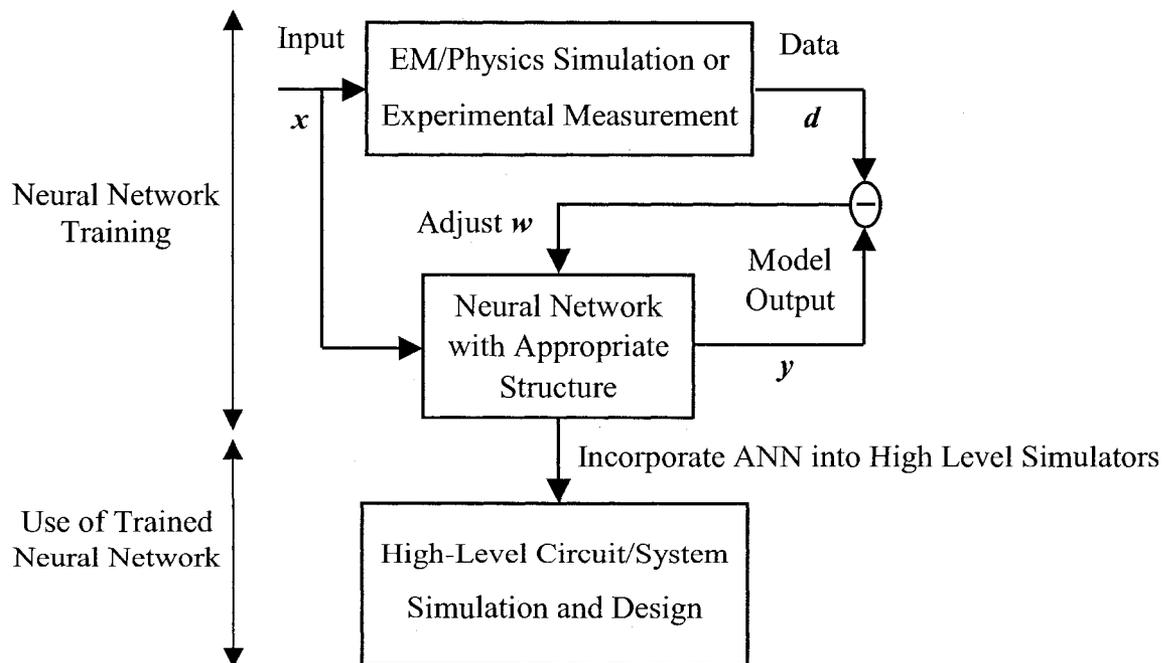
seen in the training stage. The ability of neural models to predict  $y$  with  $x$  values different from that of the training data is called the generalization ability.

As described above, the overall ANN model development involves data generation, training, and testing. An automated model generation algorithm [47] has been recently introduced that automatically drives all the sub-tasks involved in neural modeling process in a unified way. Neural model development can start with zero amount of training/test data. As the stage-by-stage training proceeds, the algorithm can (i) Determine the number of additional training/test samples required and the sampling distribution in the model input parameter space based on the neural network test error, and (ii) Adjust the size of the neural network (i.e., add more neurons in the hidden layer) based on the neural network training error. The algorithm identifies nonlinear sub-regions (if any) in the model input space and adds relatively more samples in such regions, while fewer samples are generated in smooth regions, thus making judicious use of data. The AMG algorithm can automatically drive simulators to generate enough data to train a model to meet a user-desired accuracy. AMG is designed to integrate all the subtasks involved in neural modeling, thereby facilitating a more efficient and automated model development framework. It can significantly reduce the intensive human effort demanded by the conventional step-by-step neural modeling approach.

#### **2.2.4 Use of the Neural Models**

The trained and tested neural model can then be used online during the microwave design stage, providing fast model evaluation replacing the original detailed/slow EM/physics simulators. Figure 2.3 shows the flow of the ANN approach for microwave modeling and

circuit/system design. ANN models can be integrated into a microwave circuit simulator, interconnecting each other or connecting to other components or models in the simulator to form a high-level circuit for design and optimization [2]-[4]. During simulation, the circuit simulator passes input variables, the physical/geometrical parameters of a component/circuit and electrical parameters such as bias and frequency, to the ANN model trained with the component/circuit behavior. The ANN model then computes and returns the corresponding outputs back to the simulator. The use of such ANN models helps greatly to improve design speed while maintaining the EM/physics accuracy of the detailed EM/physics model. The benefit of the neural network modeling is especially significant when the model is highly repetitively used in design processes such as optimization, Monte Carlo analysis, and yield maximization.



**Figure 2.3: Flowchart of ANN approach for microwave modeling and circuit/system design.**

As the ANN models being incorporated into the circuit environment, the component/circuit they represent can also be optimized along with the rest of the circuit with the model parameters as optimization variables [2]. This neural-based optimization allows the neural network inputs to be optimization variables, for instance, physical/geometrical parameters of the component/circuit. It addresses the challenges due to computational expenses of evaluating EM/physics effects in circuit components, and the need to repetitively vary physical/geometrical parameters and re-evaluate EM/physics behavior of all the components during design optimization. The use of neural network models helps to significantly accelerate such EM/physics based optimization.

### **2.3 Neural Networks With Prior Knowledge**

As reviewed in the previous section, different neural network structures, such as MLP neural networks [2], RBF neural networks [2], wavelet neural networks [2], recurrent neural networks [8], and dynamic neural networks [49] have been developed to meet various kinds of modeling requirements. However, the pure neural network (without using any approximation model) is a kind of black-box model structurally with no problem dependent information embedded. In this case, a large amount of training data is usually needed to ensure model accuracy. In reality, generating large amounts of training data could be very expensive for microwave problems, e.g., physics based device simulations and detailed device measurements could be very expensive to perform at many points in the model input parameter space. The conflict between the requirement of fast model development and the present status of expensive (time-consuming) data generation becomes a problem that the pure neural network cannot solve.

The key to solve this problem is a concept called the knowledge-based neural network [10]. The idea of knowledge-based neural network is to exploit existing knowledge in the form of empirical functions or equivalent circuit model together with a neural network model to develop faster and more accurate models. Existing microwave knowledge can provide additional information to the original problem that may not be adequately represented by the limited training data, and the neural network can help bridge the gap between the empirical model and the actual device behavior. Extrapolation capability is also enhanced because of the embedded knowledge in the model.

Recent publications address four methods referring to this advanced technique: source difference method [9], prior knowledge input (PKI) method [11], knowledge-based neural networks (KBNN) of [10], and space-mapped neuromodeling [12].

### **2.3.1 Source Difference Method**

The source difference method, also known as the hybrid EM-ANN modeling method [9], is one of the earlier methods utilizing the knowledge-based concept. Its structure is shown in Figure 2.4. The hybrid EM-ANN model is formed by generating the difference between the existing approximate model (source model) and the EM simulation results (target model). The difference data is then used to train the neural network. This results in a smaller range of the output variables and a simpler input-output relationship. This method is expected to give good results when the difference has a simpler input-output relationship as a function of the inputs than the target data. This simpler input-output relationship requires less EM simulation points to capture important data trends. This simplification is very desirable since EM simulations consume a major portion of the

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

As shown in Figure 2.4, for each input sample  $x$ , the corresponding output  $y' = f_{emp}(x)$  is computed from the approximate model, which could be empirical functions or an equivalent circuit model [9]. The difference between the empirical approximation and the training data  $\Delta y$  is represented by a neural network [9], say, a three layer MLP, as  $\Delta y = f_{ANN}(x, w)$ , where  $w$  is the internal weight vector of the neural network. The overall output of hybrid EM-ANN model is [9]

$$y = y' + \Delta y = f_{emp}(x) + f_{ANN}(x, w) \quad (2.4)$$

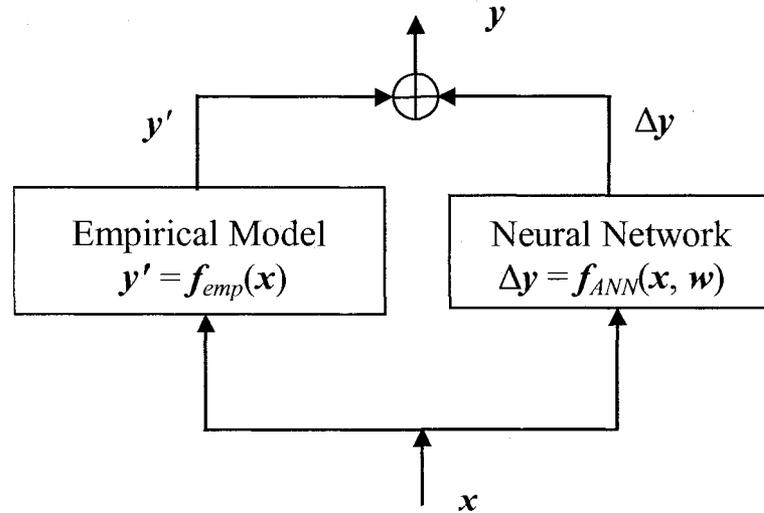


Figure 2.4: Structure of hybrid EM-ANN model utilizing the source difference method (from Zhang and Gupta [2]).

### 2.3.2 Prior Knowledge Input

In the prior knowledge input method [11], the outputs of the existing empirical model (source model) are used as inputs to the neural network model, in addition to the original problem (target model) inputs, shown in Figure 2.5. In this case, the input-output mapping to be learned by the neural network is that between the output response of the existing approximate model and that of the target model.

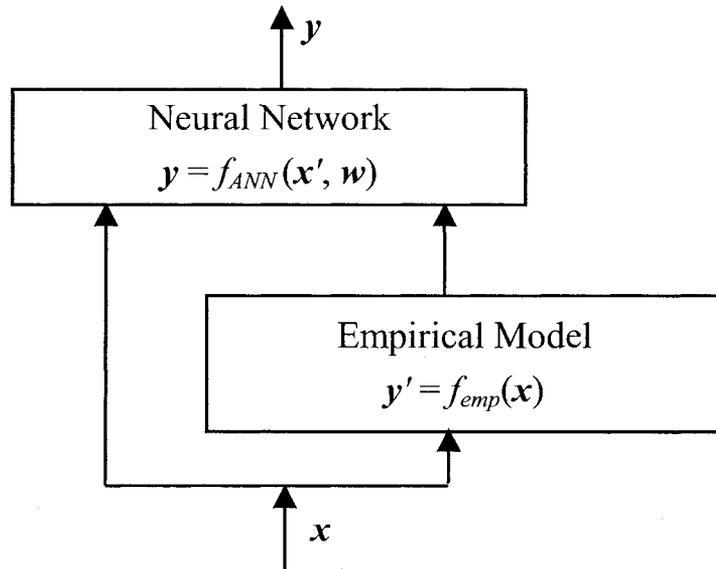


Figure 2.5: Structure of the PKI model (from Zhang and Gupta [2]).

For each  $x$  in the training data, a corresponding  $y' = f_{emp}(x)$  is computed by the empirical functions or equivalent circuit response. The neural network will then learn the mapping from target model inputs and source model outputs  $x' = (x, y')$ , to the target data, resulting in a simpler input-output relationship as compared to the original problem, which requires less training data [11]. After training, given input  $x$ , first using an

empirical function to get an approximation  $y'$ , then the neural network will predict the final result as [11]

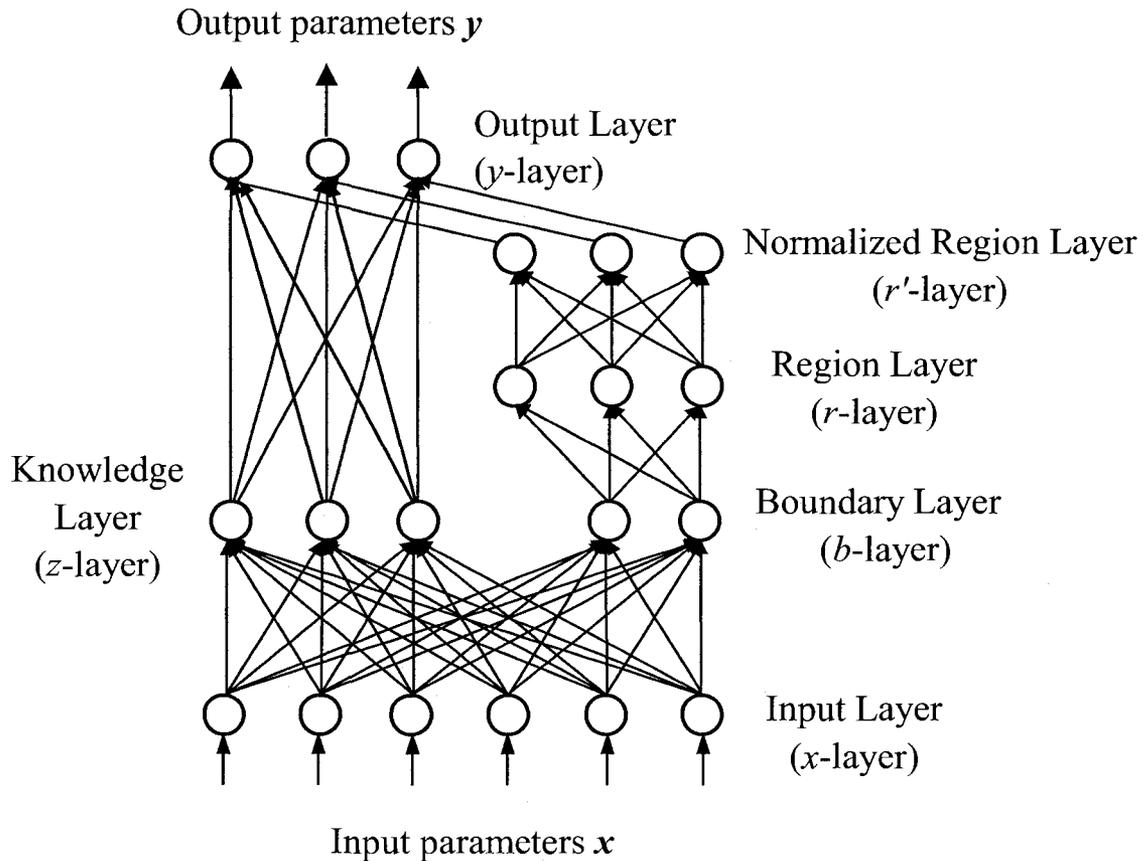
$$y = f_{ANN}(x', \mathbf{w}) = f_{ANN}(x, f_{emp}(x), \mathbf{w}) \quad (2.5)$$

### 2.3.3 Knowledge-Based Neural Network

The knowledge-based neural network [10] is a modeling approach combining microwave empirical experience with the learning power of neural networks by incorporating microwave empirical or semi-analytical information into the internal structure of neural networks. The comprehensive structure of the knowledge-based neural network [10] is illustrated in Figure. 2.6.

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

to normalize the outputs of the region layer. The output layer contains second-order neurons combining knowledge neurons and normalized region neurons.



**Figure 2.6: The structure of the knowledge-based neural network (KBNN) (from Wang and Zhang [10]). Typically the KBNN model includes six layers: the input layer, the boundary layer, the region layer, the normalized region layer, the knowledge layer, and the output layer.**

As a summary of Section 2.3, the prior knowledge gives the neural network more information about the original microwave problem, besides the information included in the training data. Consequently, neural network models with prior knowledge have better

reliability when the training data is limited or when the model is used beyond the training range. Another neural network modeling technique with prior knowledge, utilizing an advanced optimization concept called space mapping, is discussed in next section.

## 2.4 Space Mapping for Microwave Modeling and Design

### 2.4.1 Space Mapping Optimization

Space mapping (SM) is an advanced optimization concept, proposed by Bandler *et al.* [13], for modeling and design of engineering devices and systems, allowing expensive EM optimizations to be performed efficiently with the help of fast and approximate “coarse” or surrogate models [13], [60]-[64]. SM intelligently links companion “coarse” (ideal, fast, or low fidelity) and “fine” (accurate, practical, or high fidelity) models of different complexities, e.g., empirical circuit-theory based simulations and full-wave EM simulations, to accelerate iterative design optimization. Through SM optimization, the surrogates of the fine models are iteratively refined to achieve the accuracy of EM simulations with the speed of circuit-theory based simulations.

The mathematical representation of space mapping methodology presented in [13] is recalled as follows. Let vectors  $\mathbf{x}_c$  and  $\mathbf{x}_f$  represent the design parameters of the coarse and fine models, respectively. Let  $\mathbf{R}_c(\mathbf{x}_c)$  and  $\mathbf{R}_f(\mathbf{x}_f)$  represent the corresponding responses of the coarse and fine models, respectively. The response of the coarse model  $\mathbf{R}_c$  is much faster to calculate, but less accurate than the response of the fine model  $\mathbf{R}_f$ . The aim of space mapping optimization is to find an approximate mapping  $\mathbf{P}$  from the fine model parameter space  $\mathbf{x}_f$  to the coarse model parameter space  $\mathbf{x}_c$ , i.e.,  $\mathbf{x}_c = \mathbf{P}(\mathbf{x}_f)$  such that

$$R_c(\mathbf{P}(\mathbf{x}_f)) \approx R_f(\mathbf{x}_f) \text{ [60].}$$

The space mapping technique uses coarse model optimization with the mapping  $\mathbf{P}$  to find a good estimate for the optimal solution of the fine model, enabling effective use of the surrogate's fast evaluation to sparingly manipulate the iterations of the fine model in design optimization. It has been applied with great success to otherwise expensive direct EM optimizations of microwave components and circuits with substantial computation speedup [13], [60]-[64].

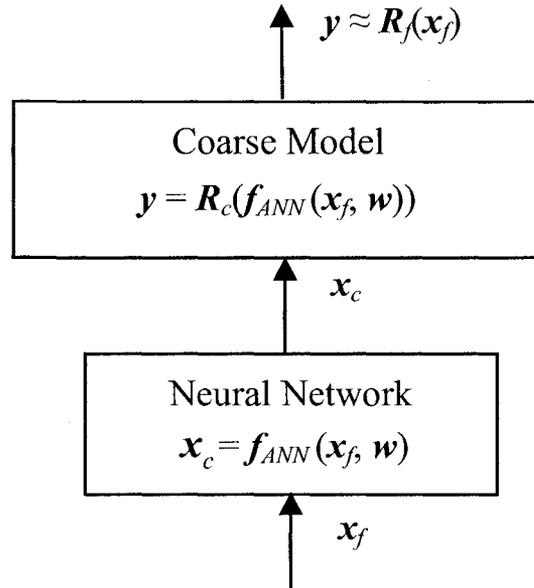
#### 2.4.2 Space Mapping Based Neuromodeling

Recently a space mapping based neuromodeling technique combining neural networks with space mapping (Bandler, Ismail, Rayas-Sánchez, and Zhang [12]) was developed, using neural networks to map the coarse model to the fine model. It retains the efficiency of space mapping optimization by directing the bulk of CPU intensive evaluations to the coarse model, while preserving the accuracy and confidence offered by the fine model. The coarse model is typically an empirical or equivalent circuit model, which is fast but often has a limited validity range for its parameters, beyond which the simulation results may become inaccurate. The fine model is from a detailed physics/EM simulator or measurement, which is accurate but CPU intensive. Neural networks are used to provide the mapping between the coarse model parameter space and the fine model parameter space, subsequently establishing the mathematical link between the coarse and fine models.

The space mapping based neuromodeling is illustrated in Figure 2.7, where the mapping  $\mathbf{P}$  from the fine input space  $\mathbf{x}_f$  to the coarse input space  $\mathbf{x}_c$  is realized by a neural

network as  $\mathbf{x}_c = f_{ANN}(\mathbf{x}_f, \mathbf{w})$  [12], where  $\mathbf{w}$  represents the weighting parameters of the neural network. The overall model output  $\mathbf{y}$  is then produced through the coarse model with the mapped input  $\mathbf{x}_c$  as [12]

$$\mathbf{y} = \mathbf{R}_c(\mathbf{x}_c) = \mathbf{R}_c(P(\mathbf{x}_f)) = \mathbf{R}_c(f_{ANN}(\mathbf{x}_f, \mathbf{w})) \quad (2.6)$$



**Figure 2.7: Structure of the space-mapped neural model (from Zhang and Gupta [2]).**

The mapping  $P$  is determined by neural network training to solve the optimization problem of [12]

$$\min_{\mathbf{w}} \left\| \begin{bmatrix} \mathbf{e}_1^T & \mathbf{e}_2^T & \cdots & \mathbf{e}_{N_p}^T \end{bmatrix}^T \right\| \quad (2.7)$$

where  $N_p$  is the total number of training samples, and  $e_p$  ( $p=1,2,\dots,N_p$ ) is the error vector at input sample of  $\mathbf{x}_f = \mathbf{x}_p$  given by [12]

$$\mathbf{e}_p = \mathbf{R}_f(\mathbf{x}_p) - \mathbf{R}_c(\mathbf{f}_{ANN}(\mathbf{x}_p, \mathbf{w})) \quad (2.8)$$

Once the neural network is trained, i.e., the mapping  $\mathbf{P}$  is found, the space-mapped neuromodel, whose responses should match the training data, becomes an accurate representation of the fine model for efficient evaluations in simulation and optimization.

Space-mapped neuromodeling has demonstrated its efficiency by passive modeling or small-signal device modeling, achieving fast and accurate models for devices such as bends, high temperature superconductor filters, embedded passives in multilayer printed circuits, and other linear components [12]. Recently, space mapping based neuromodeling has been expanded to new formulations for nonlinear device modeling [14], and the concept of combined space mapping and neural networks has also been applied to statistical modeling of passive EM structures [65] and small-signal linearized devices [66].

## **2.5 Nonlinear Microwave Device Modeling by Conventional Techniques**

In previous sections, we have reviewed ANN based techniques and their applications to microwave modeling and design, advanced neural modeling techniques with prior knowledge, and the space mapping methodology. In this section, we will review conventional techniques in modeling the nonlinear behavior of microwave devices.

The fast evolution of semiconductor device technologies perseveres an ongoing need for accurate new models in the computer-aided design of RF/microwave circuits and systems. A number of approaches for semiconductor device modeling have been developed as reviewed in [1] by Steer, Bandler, and Snowden. However, most of them are for specific modeling purposes [1], [67]-[69], falling into different model categories such as physical models [70]-[77], equivalent circuit models [78]-[84], and black-box models (e.g., table lookup models [85]).

### **2.5.1 Physics-based Modeling Technique**

Physical models are important to relate the yield and performance of the design to the fabrication process, material properties, and device geometry, to achieve improved reliability and significantly reduced cost in large-signal design. Two principal types of physical models that are applied to device design and characterization are presented in [1]. The most straightforward of these is based on a derivative of equivalent-circuit models, where the circuit element values are quantitatively related to the device geometry, material structure, and physical processes [70]-[73]. This model is an analytical model but more useful device information may be gleaned from the value of the circuit elements. The second approach is more fundamental in nature and is based on

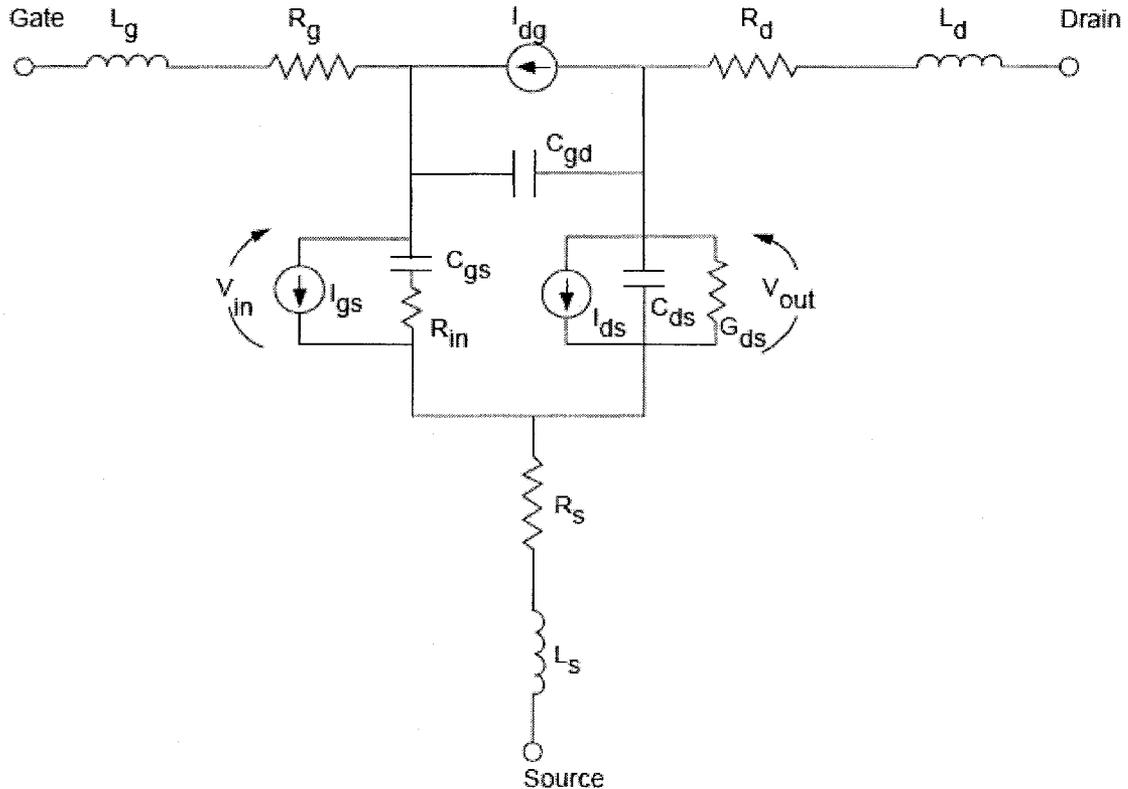
a rigorous solution of the carrier transport equations over a representative geometrical domain of the device [74]-[77]. These models use numerical solution schemes to solve the carrier transport equations in semiconductors often accounting for hot electrons, quantum mechanics, EM, and thermal interaction.

However, in design optimization, physical models based on numerical algorithms may require too much computational time to be used to any extent in circuit design work, and physical models based on analytical expressions may lack the detail and fidelity of their numerical counterparts. Another disadvantage of physics-based modeling is that it usually takes long time to develop a good model for a new device. These become major reasons to explore alternative modeling techniques.

### **2.5.2 Equivalent Circuit Modeling Technique**

Equivalent circuit approaches [1] are commonly used in microwave device modeling because they are formulated to be efficiently exercised in existing circuit simulators, and thus are efficient for circuit design and optimization. The equivalent circuit model is usually composed of nonlinear controlled voltage or current sources, together with (linear or nonlinear) parasitic resistors, inductors, and capacitors, which the existing simulators are accustomed to dealing with. It is conventional to separate the extrinsic parameters from the intrinsic device parameters. The intrinsic parameters are assumed to contain all the bias-dependent behavior and the extrinsic parameters are assumed to have constant values. The nonlinear elements in the equivalent circuit are represented by empirical functions containing several so-called “model parameters”. Dedicated procedures allow extracting the value of these parameters out of dc and small-signal S-parameter

measurements. As an example, an equivalent circuit model of a MESFET [67] is shown in Figure 2.8.



**Figure 2.8: The conventional large-signal equivalent circuit model for MESFET (from Golio [67]). The resistances and inductances of the extrinsic components are constant.  $I_{ds}$ ,  $I_{gs}$ , and  $I_{dg}$  are nonlinear voltage controlled current sources.  $C_{gd}$  and  $C_{gs}$  are nonlinear capacitances.**

Equivalent circuit modeling is a convenient way to interpret the behavior of the transistor characteristics, and is computationally efficient. However, it is only accurate in specific cases and often inadequate to describe the full behavior of the device. The development of such a model also requires experience and involves a trial-and-error

process to find appropriate circuit topology and the values of the circuit elements. Compared to physical models, an equivalent circuit model may not have direct links with the physical process parameters of the device. Empirical formulas for such links may exist, but the accuracy cannot be guaranteed when applied to different devices.

### **2.5.3 Table-Based Modeling Technique**

Another type of device model is the table-based model, proposed by Root, Fan, and Meyer [85], which has no assumption of particular analytical functions. The table-based models have some properties of black-box models. The equations used result from fitting to data, using splines, or other such functions. These models can therefore “learn” the behavior of the nonlinear device and are ideal for applications where the functional form of the behavior is unknown. Table-based models are efficient but do not provide the user with any insight, since there is a minimal “circuit model.” They have difficulty incorporating dispersive effects, such as “parasitic gating” due to traps and do not accommodate self-heating effects [67]. They cannot be accurately extrapolated into regions where data was not taken, and the models are often restricted in their application due to the limited information within the model. Furthermore, such models can also be slow if the dimension of the lookup table is high.

### **2.5.4 Statistical Modeling of Nonlinear Microwave Devices**

Accurate statistical models for nonlinear devices are essential to the success in costly and time-consuming RF and microwave circuit design, where process parameter variations of active devices have a strong impact on overall yield [19]-[22]. Most of the existing

statistical modeling approaches are based on dc and small-signal S-parameter data for linear [86]-[89] or nonlinear [90]-[92] modeling of microwave devices. Usually, the data generation (either from detailed device simulation or measurement) has to be performed on many devices in order to obtain statistical information. Each set of dc and S-parameter data, corresponding to one device, is converted to parameters of an equivalent circuit through a parameter extraction procedure. The statistical properties of the equivalent circuit parameters are then examined, and the estimates of the means ( $\mu$ ), standard deviations ( $\sigma$ ), and correlation coefficients ( $\rho$ ) are calculated. Principle component and factor analysis [93], or sensitivity analysis [22] can be used to identify critical factors to reduce the dimension of the statistical parameters. If needed, normality transformation [94] can also be applied to the statistical parameters to convert the extracted distribution, which may be arbitrary, to a better approximation of the Gaussian (normal) distribution [94]. Finally, statistical models based on some multivariate or heuristic techniques capable of recreating those distributions or those means, standard deviations, and correlations can be developed.

Nowadays, nonlinear device modeling directly using large-signal data has gained recognition due to the increasing need for accuracy in characterizing large-signal behavior. Nonlinear statistical models are required in circuit applications where bias point variations of active devices have a strong impact on overall yield. Another example is the power amplifier design where the large-signal statistical property of the devices needs to be represented for yield-driven design. However, direct large-signal statistical modeling

remains prohibitive by the conventional techniques because complete large-signal measurement for many devices is too expensive and time consuming.

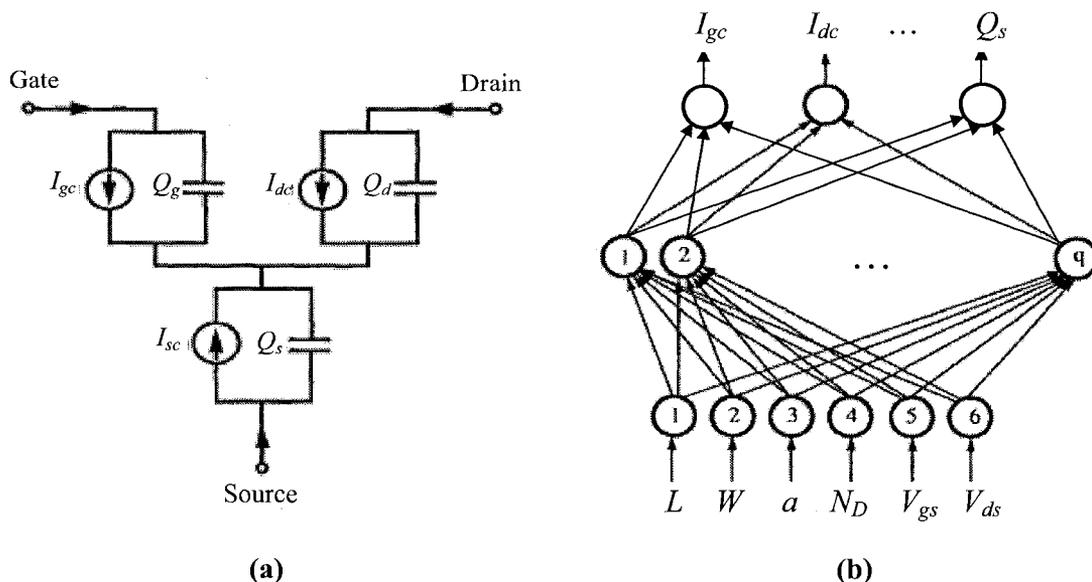
## **2.6 Nonlinear Microwave Device Modeling by Neural-Based Techniques**

Recently there emerged an advanced requirement for nonlinear device modeling, regarding development cost, speed and accuracy, statistical capability, and modeling automation. However, detailed physical models [70]-[77] are usually computationally slow. Existing equivalent-circuit modeling techniques [78]-[84] require trial and error of the model topology and component empirical equations. Table lookup models are easy to develop but suffer from the curse of dimensionality [85]. With the universal approximation capability, neural networks have become flexible alternatives to meet such requirement for nonlinear device modeling [2]-[4]. Several modeling methods have been developed, including two major categories as direct modeling and indirect modeling approaches.

### **2.6.1 Neural Network Based Direct Modeling Approach**

In the direct modeling approach, the external behavior of the nonlinear device is directly modeled by neural networks. This approach has been applied to model dc characteristics of a physics-based MESFET [10], small-signal HBT device [31], and large-signal MESFET devices [5], [95]. As an example, the work by Zaabab, Zhang, and Nakhla in [5] presents a straightforward formulation of large-signal models to describe terminal currents and charges of nonlinear devices as nonlinear functions of the device parameters and the bias conditions. In this example, a CPU intensive physics-based device model is

re-modeled by neural networks to efficiently speed up simulation and optimization [5]. The physical MESFET model chosen is the Khatibzadeh and Trew model [73]. Figure 2.9 shows the representation of the MESFET using a neural network model for the terminal currents and charges.



**Figure 2.9: (a) A physics-based MESFET modeled by (b) a neural network (from Zaabab, Zhang, and Nakhla [5]). The terminal currents and charges of the MESFET in (a) are represented by the neural network model as illustrated in (b) with the geometrical/physical and electrical parameters as the neural network inputs.**

The neural network model has six inputs namely gate length  $L$ , gate width  $W$ , channel thickness  $a$ , doping density  $N_D$ , gate voltage  $V_{gs}$  and drain voltage  $V_{ds}$ . The model outputs are the gate, drain, and source currents  $I_{gc}$ ,  $I_{dc}$ , and  $I_{sc}$ , and the total charges  $Q_g$ ,  $Q_d$ , and  $Q_s$  on the gate, drain, and source electrodes, respectively. The training data for the neural network is achieved from simulation of the physics-based MESFET for different

configurations at a number of bias points using OSA90 [96]. Since this large-signal MESFET neural model directly describes terminal currents and charges as nonlinear functions of device parameters, it can be conveniently used in a circuit simulator to satisfactorily perform dc, small-signal, and large-signal harmonic balance (HB) simulations. The work by Schreurs, *et. al*, in [6] has successfully demonstrated this technique to model pHEMTs and MOSFET devices, using full two-port vectorial large-signal measurements as training data.

### **2.6.2 Neural Network Based Indirect Modeling Approach**

The indirect modeling approach combines known equivalent circuit models together with neural network models to develop more efficient and flexible models for nonlinear microwave devices. As described in Subsection 2.5.2, the lumped equivalent circuit approach is a traditional way for transistor modeling. Developing such models requires experience and involves a trial-and-error process to determine a matching topology. Moreover, equivalent circuit parameters may not be related to the physical/geometrical parameters of the device under consideration. Empirical formulas for such relations exist, and neural networks can easily learn these relationships. A hybrid approach that utilizes existing knowledge in the form of known equivalent circuit and empirical formulas, together with the powerful learning and generalization abilities of neural networks has been demonstrated for modeling large-signal behavior of MESFET [97] and HEMT [98]-[100].

Another example is to use the adjoint neural networks for large-signal FET modeling [7] as illustrated in Figure 2.10. Instead of using the maybe unknown terminal currents

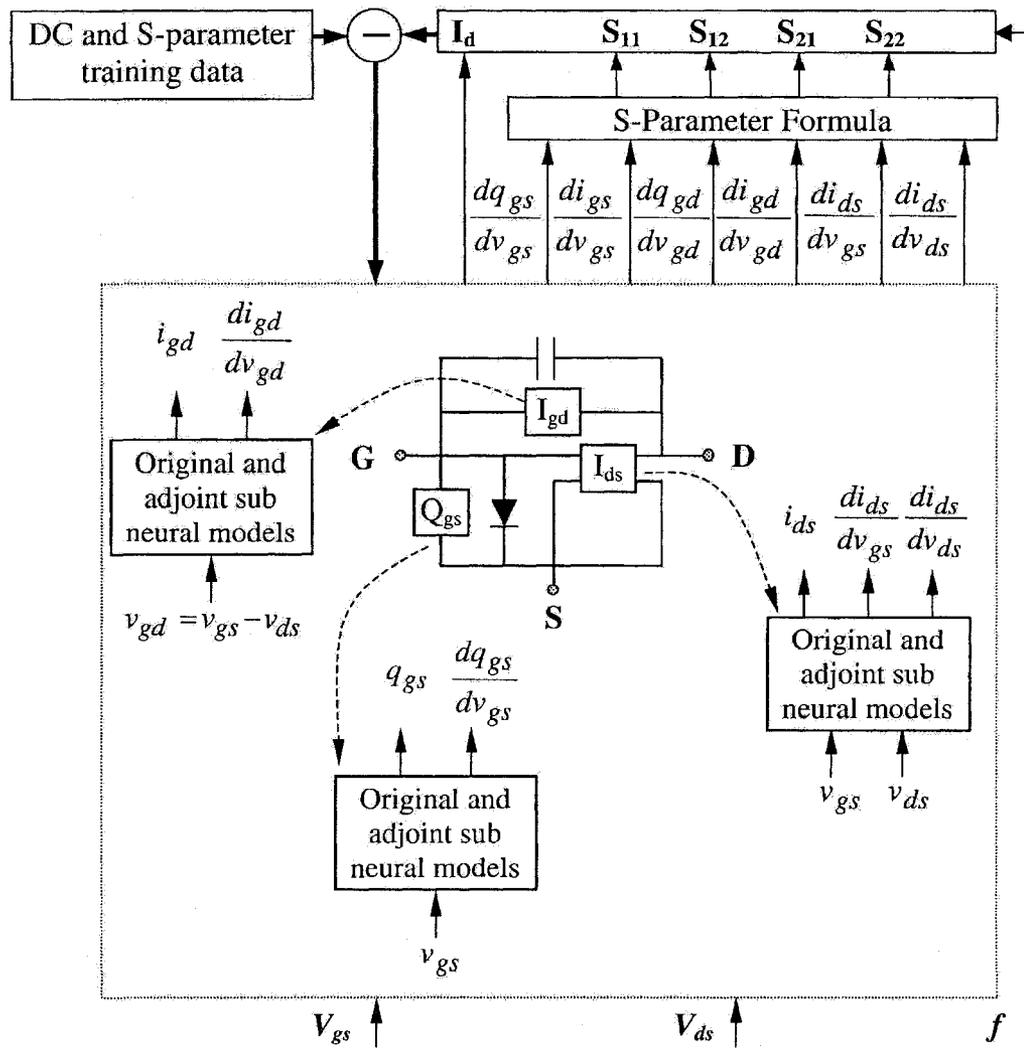


Figure 2.10: Large-signal FET modeling including adjoint neural networks trained by dc and bias-dependent S-parameters (from Xu, Yagoub, Ding, and Zhang [7]).

and charges as the outputs for neural network training, the adjoint neural networks are trained directly by dc and bias-dependent S-parameters. Microwave knowledge of a basic equivalent circuit is combined with sub-neural models leading to a knowledge-based approach for the FET modeling. Here, adjoint neural networks complement the intrinsic FET equivalent circuit by providing the unknown nonlinear currents ( $I_{ds}$ ,  $I_{gd}$ ) and charge ( $Q_{gs}$ ) [7]. The adjoint neural networks in enhancing conventional FET models through adding trainable nonlinear current or charge relationships to the model. Such a trainable nonlinear relationship is especially beneficial when analytical formulas in the FET problem are unknown or available formulas are not suitable. By combining adjoint neural networks with existing FET models, one can improve the models efficiently without having to go through the trial-and-error process typically needed during manual creation of empirical functions.

### **2.6.3 Neural Network Based Statistical Modeling**

Neural network has also been used in statistical modeling as an accurate and efficient statistical extraction method for small-signal equivalent circuit model parameters of a HBT [66], to solve the problem of noisy statistical properties caused by using the optimization-based parameter extraction techniques. In [66], a neural network is used to learn the required relation between model parameters domain and performance (measured or simulated quantities) domain. Eight complex means of the real and imaginary parts of the S-parameters over the considered frequency range are considered as the inputs for the neural network, and the most sensitive equivalent circuit model parameters are used as the outputs for the neural network. A nominal device is firstly determined, by taking the

device which has average performance over 23 device samples from different wafers with a geometry of  $0.8 \mu\text{m} \times 9.6 \mu\text{m}$  on a  $S_i/S_iG_e$  HBT in the frequency range from 1 GHz to 20 GHz, biased at  $V_{BE} = 0.9 \text{ V}$  and  $V_{CE} = 1.5 \text{ V}$  [66]. A small-signal equivalent circuit model for the nominal device is extracted. The training data is generated by performing 100 Monte Carlo simulations randomly around the vicinity ( $\pm 10\%$ , the maximum limit the model parameter can deviate from the nominal value) of the nominal model parameters to get the corresponding performances. It has been proved that the extraction by neural networks is more effective to obtain a better statistical model than using the conventional optimization-based extraction methods, providing a more robust statistical model for the device.

The above reviewed applications of the neural based device modeling techniques have demonstrated that trained neural models from measurement data can represent dc, small-signal, and large-signal behavior of a new device, even if the device theory/equations are still unavailable. Because a neural network can learn the nonlinearity much more automatically and easily than manually formulating a nonlinear function as by equivalent circuit model development, it is very suitable and efficient for such modeling activities. In this sense, the neural based modeling methods provide useful alternatives for efficient generation of nonlinear device models for use in large-signal simulation and statistical design.

## **2.7 Conclusions**

In this chapter, existing conventional and neural network based techniques for RF/microwave modeling and design, which are relevant to this thesis work, have been

reviewed. Neural network based models can be used to achieve a significant speedup of RF/microwave simulation and optimization by replacing electronic and microwave component models, which are represented by detailed EM/physics equations. These neural models can be trained with the corresponding EM/physics data. However, most of the existing neural network structures are of black-box type without any problem dependent information embedded, and need a large amount of training data to get an accurate model, which results in high cost model development. Neural network based modeling techniques utilizing prior knowledge have been introduced to address this issue. With the rapid development in semiconductor technology, new devices constantly emerge. Although there have been existing techniques including neural network based techniques to meet different requirements in nonlinear device modeling, the challenges for more efficient, accurate, cost-effective, and systematic model development still continue to exist.

## **Chapter 3: Analytical Neuro-Space Mapping Technique for Nonlinear Microwave Device Modeling**

### **3.1 Introduction**

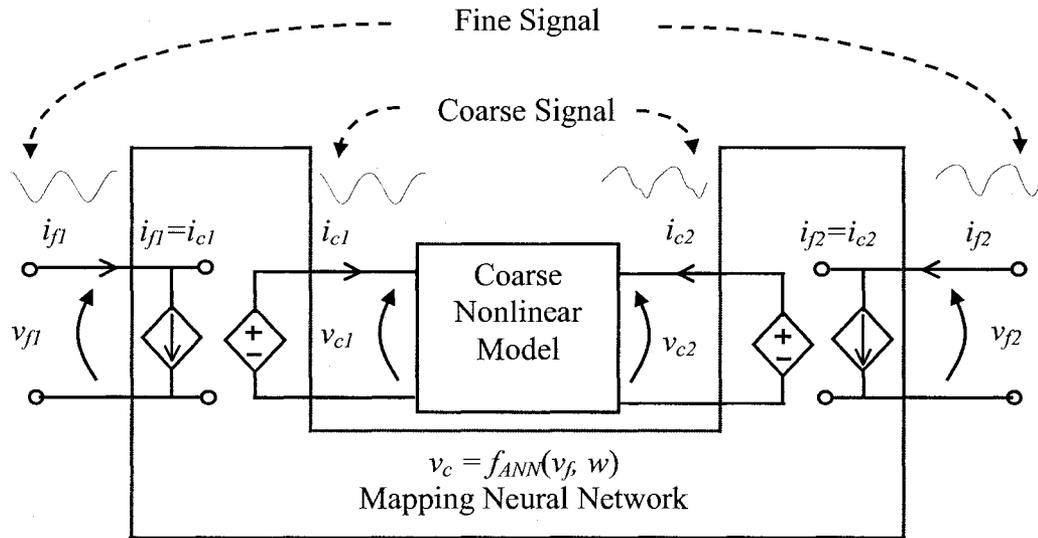
In the field of nonlinear device modeling, the neuro-space mapping (Neuro-SM) technique [14] has been recently proposed, using a novel formulation of space mapping together with a neural network to automatically modify the voltage and current signals of an existing device model (coarse model) to accurately match new device data (fine model). It is an advance over several previous ANN based methods for device modeling such as the I-Q model in [25] and the adjoint neural network method in [7]. The Neuro-SM in [14] is the first time that a complete large-signal device model from an existing circuit simulator library can be combined with neural network architecture. With the neural network represented by controlled sources, the Neuro-SM model can be conveniently incorporated into existing circuit simulators for nonlinear circuit design [14]. However, the controlled sources introduce additional variables and Kirchhoff equations to the overall circuit. Thus such a circuit-formulation of Neuro-SM (circuit-based Neuro-SM) model improves accuracy but at the cost of computational overhead from the extra equations in circuit simulation.

In this chapter, a new analytical formulation is derived [15] allowing efficient Neuro-SM model evaluation and sensitivity analysis for dc, small-signal, and large-signal applications. In the proposed technique, the mapping mechanisms are incorporated by directly modifying the signals in existing device equations. In this approach, there are no extra unknown variables or equations introduced into circuit simulation equations. This increases simulation efficiency especially when the Neuro-SM model is later used in circuit and system designs. Based on the proposed analytical formulation, a 2-phase training algorithm utilizing gradient optimization is developed for efficient training of the Neuro-SM models. The proposed analytical Neuro-SM model is more efficient in both model training and circuit simulation/optimization than the equivalent circuit formulation of Neuro-SM in [14].

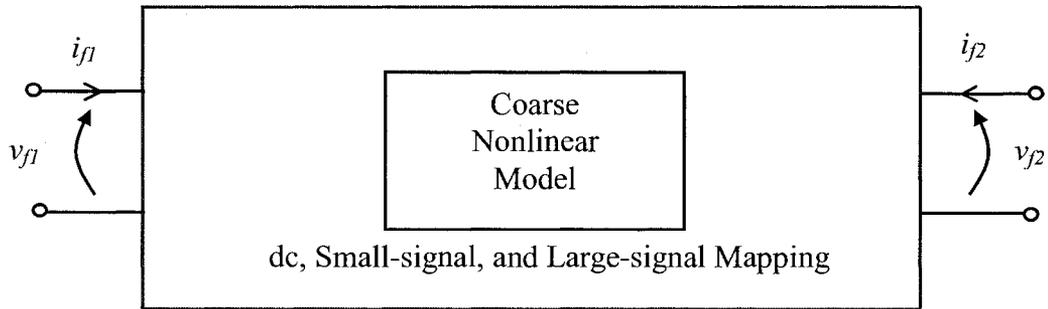
### 3.2 Problem Formulation

The starting point for the Neuro-SM technique is when the existing/available device model cannot match the data of a new device. Let the existing/available nonlinear device model be called the coarse model. Let the fine model be a fictitious model implied by actual device data from measurement or detailed/expensive device simulations. Suppose that the gap between the coarse and fine models cannot be overcome by simply optimizing the parameters in the coarse model. To achieve a model that can best match the device data, the model structure or the nonlinear equations of the coarse model need to be modified.

Figure 3.1(a) shows the structure of a 2-port circuit-based Neuro-SM model [14]. We define the terminal voltage and current signals of the coarse model as  $\mathbf{v}_c = [v_{c1}, v_{c2}]^T$  and



(a)



(b)

**Figure 3.1: Structure of the general 2-port Neuro-SM nonlinear model, where a neural network  $f_{ANN}$  is used to provide a mapping between coarse input signals and fine input signals. (a) Circuit-based Neuro-SM using neural network equations in controlled sources for the mapping. (b) Illustration of the proposed analytical Neuro-SM model for efficient model development without introducing extra equations to circuit simulation.**

$\mathbf{i}_c = [i_{c1}, i_{c2}]^T$ , respectively. Similarly, we define the terminal voltage and current signals of the fine model as  $\mathbf{v}_f = [v_{f1}, v_{f2}]^T$  and  $\mathbf{i}_f = [i_{f1}, i_{f2}]^T$ , respectively. Here  $\mathbf{v}_c$  and  $\mathbf{i}_c$  are called coarse signals, and  $\mathbf{v}_f$  and  $\mathbf{i}_f$  are called fine signals. In the Neuro-SM model, the fine voltage signals  $\mathbf{v}_f$  are mapped into the coarse voltage signals  $\mathbf{v}_c$  by a neural network through  $\mathbf{v}_c = f_{ANN}(\mathbf{v}_f, \mathbf{w})$ , where  $f_{ANN}$  represents a multilayer feedforward neural network, and  $\mathbf{w}$  is a vector containing all internal synaptic weights of the neural network. This neural network is embedded as functions of the voltage controlled voltage sources in the circuit-based Neuro-SM of [14]. Current controlled current sources are used to pass  $\mathbf{i}_c$  to  $\mathbf{i}_f$ .

Such a circuit-based Neuro-SM model will match the fine device data more closely than that is possible from the coarse model alone. This is due to (a) the additional degrees of freedom in device modeling from the mapping neural network, and (b) the use of this freedom where it is needed: the flexible transformation of terminal signals. The circuit-based structure also allows the Neuro-SM model to be conveniently implemented in existing circuit simulators for circuit design.

The overall circuit-based Neuro-SM model has two external ports as in Figure 3.1(a). The mapping neural network adds two internal ports in the model. Thus additional nodal variables and nonlinear circuit equations [101] have to be solved for each use of Neuro-SM, for example, for each bias and each frequency. This computational overhead occurs not only in simulation but also in sensitivity analysis.

### 3.3 Proposed Analytical Formulation and Exact Sensitivity of the Neuro-SM Technique

#### 3.3.1 Proposed Analytical Formulation of the Neuro-SM Model

We propose a new analytical formulation for efficient Neuro-SM modeling as illustrated in Figure 3.1(b). In the new formulation, the mapping mechanisms are analytically derived instead of being indirectly represented by controlled sources and Kirchhoff equations. The port voltage and current signals of the device are modified explicitly in the original circuit equations through the mapping neural network. In this way, neural network and space mapping become an integral part of the model equations, without adding any extra nodal variables or equations. We examine how to achieve this analytical formulation within the environment of dc, small-signal, and large-signal simulations. We then further examine an analytical formulation of Neuro-SM sensitivity analysis for dc, small-signal, and large-signal cases.

**Analytical dc mapping:** The Neuro-SM model is a full large-signal nonlinear model. The mapping for coarse dc voltage signals  $V_{c,dc}$  and fine dc voltage signals  $V_{f,dc}$  is directly achieved by the neural network. Let the dc response of the coarse model be a nonlinear function evaluated at coarse dc voltages, i.e.,  $I_c|_{V_{c,dc}}$ . Let the dc response of the Neuro-SM model be  $I_f$ . Neuro-SM requires that after receiving the modified signal, the coarse model output signal should become an approximation of the fine output signal. Thus the dc output current of the analytical Neuro-SM model as a function of the fine dc input voltage signal  $V_{f,dc}$  is

$$\mathbf{I}_f = \mathbf{I}_f(\mathbf{V}_{f,dc}) = \mathbf{I}_c \Big|_{\mathbf{V}_{c,dc} = f_{ANN}(\mathbf{V}_{f,dc}, \mathbf{w})} \quad (3.1)$$

**Analytical small-signal mapping:** The small-signal S-parameters are mapped via the analytical mapping of the Y matrices between the coarse model  $\mathbf{Y}_c$  and fine model  $\mathbf{Y}_f$  as

$$\mathbf{Y}_f = \mathbf{Y}_c \Big|_{\mathbf{V}_{c,Bias} = f_{ANN}(\mathbf{V}_{f,Bias}, \mathbf{w})} \cdot \left( \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f} \Big|_{\mathbf{v}_f = \mathbf{V}_{f,Bias}} \right)^T \quad (3.2)$$

where  $\mathbf{Y}_c$  is evaluated at the mapped bias  $\mathbf{V}_{c,Bias}$ , and the derivative of  $\mathbf{f}_{ANN}$  is obtained at the bias of the fine model  $\mathbf{V}_{f,Bias}$  using the adjoint neural network method [7]. Notice that  $\mathbf{Y}_c$  is complex and has contributions of all elements in the coarse model including capacitors. Equation (3.2) represents a transformation (mapping) of  $\mathbf{Y}_c$  through the derivatives of  $\mathbf{f}_{ANN}$ .

**Analytical large-signal mapping:** For large-signal simulation, we formulate the 2-port Neuro-SM as a current-charge model. The analytical large-signal mapping is derived using the harmonic balance environment, which requires nonlinear models in the time domain and circuit equations in the frequency domain [102]. Let  $\mathbf{i}_{c,NL}(t) \Big|_{\mathbf{v}_c(t)}$  and  $\mathbf{q}_{c,NL}(t) \Big|_{\mathbf{v}_c(t)}$  represent the nonlinear terminal current and charge of the coarse model in terms of coarse input voltage signals  $\mathbf{v}_c(t)$ . In the proposed analytical Neuro-SM model, given the fine input signals  $\mathbf{v}_f(t)$ , the fine output current and charge are computed by (3.3)

in the time domain

$$\begin{aligned}\mathbf{i}_{f,NL}(t) &= \mathbf{i}_{c,NL}(t) \Big|_{\mathbf{v}_c(t)=f_{ANN}(\mathbf{v}_f(t),\boldsymbol{\omega})} \\ \mathbf{q}_{f,NL}(t) &= \mathbf{q}_{c,NL}(t) \Big|_{\mathbf{v}_c(t)=f_{ANN}(\mathbf{v}_f(t),\boldsymbol{\omega})}\end{aligned}\quad (3.3)$$

For the frequency domain case, let the currents of the Neuro-SM model and the coarse model at a generic harmonic frequency  $\omega_k$  be  $\mathbf{I}_f(\omega_k)$  and  $\mathbf{I}_{c,NL}(\omega_k)$ , respectively. The subscript  $k$  represents the index of the harmonic frequency,  $k = 0, 1, 2, \dots, N_H$ , where  $N_H$  is the number of harmonics considered in HB simulation. Given fine input  $\mathbf{V}_f(\omega_k)$  for all  $k$ , fine output  $\mathbf{I}_f(\omega_k)$  is computed as

$$\begin{aligned}\mathbf{I}_f(\omega_k) &= \mathbf{I}_{c,NL}(\omega_k) \\ &= \frac{1}{N_T} \sum_{n=0}^{N_T-1} \left( \mathbf{i}_{c,NL}(t_n) \Big|_{\mathbf{v}_c(t_n)=f_{ANN}(\mathbf{v}_f(t_n),\boldsymbol{\omega})} + j\omega_k \cdot \mathbf{q}_{c,NL}(t_n) \Big|_{\mathbf{v}_c(t_n)=f_{ANN}(\mathbf{v}_f(t_n),\boldsymbol{\omega})} \right) \cdot \mathcal{W}_N(n,k)\end{aligned}\quad (3.4)$$

where  $\mathbf{v}_f(t_n) = \sum_{k=0}^{N_H} \mathbf{V}_f(\omega_k) \cdot \mathcal{W}_N^*(n,k)$  is the fine input signal at time point  $t_n$ ,  $N_T$  is the number of time points,  $\mathcal{W}_N(n,k)$  is the Fourier coefficient for the  $n^{\text{th}}$  time sample and the  $k^{\text{th}}$  harmonic frequency, and superscript \* denotes complex conjugate.

In addition, if the coarse model has separate linear and nonlinear parts [102], we can implement an even more efficient analytical Neuro-SM model for large-signal simulation

by directly mapping the linear part in the frequency domain and the remaining nonlinear part in the time domain. Let  $Y_{c,L}(\omega_k)$  represent the admittance matrix of the coarse linear subcircuit at  $\omega_k$ . Since the signals applied to the linear part are from the nonlinear mapping, we need to add the contribution of  $Y_{c,L}(\omega_k)$  to the harmonic balance (HB) equation in the form of harmonic current

$$\mathbf{I}_{c,L}(\omega_k) = Y_{c,L}(\omega_k) \cdot \frac{1}{N_T} \sum_{n=0}^{N_T-1} f_{ANN}(\mathbf{v}_f(t_n), \mathbf{w}) \cdot W_N(n, k) \quad (3.5)$$

The nonlinear subcircuit in general consists of nonlinear current and charge elements. The effect of neural network mapping on the response of the nonlinear subcircuit can be computed by (3.4), where the nonlinear current and charge are due to nonlinear components such as controlled current sources and nonlinear capacitors in the coarse model. The overall Neuro-SM model response is

$$\mathbf{I}_f(\omega_k) = \mathbf{I}_{c,L}(\omega_k) + \mathbf{I}_{c,NL}(\omega_k) \quad (3.6)$$

We described above how to systematically modify the device equations used in dc, small-signal, and large-signal simulation. Such modification is achieved by using a neural network to map the input voltage signals in the equations. Because of the neural network universal approximation capability [2], such mapping allows the model to achieve an extra degree of freedom beyond the limitation of the coarse model in matching the device

data. The mapping effect is achieved by modifying the existing circuit equations only, thus no additional equations are introduced. For example, if we remove the mapping of  $f_{ANN}$  and evaluate the model at the original input signals (i.e.,  $\mathbf{v}_c = \mathbf{v}_f$ ) instead of the mapped input, (3.1)–(3.5) would become similar to the original circuit equations. These equations are needed to solve the coarse model in dc, small-signal, and large-signal cases. By introducing the mapping neural network into these equations, we alter the signals in the coarse model to improve model accuracy. The same mapping, i.e.,  $f_{ANN}$ , is used in all cases of derivations in order to ensure the analytical consistency of the Neuro-SM model among dc, small-signal, and large-signal simulations.

### 3.3.2 Sensitivity Analysis of the Analytical Neuro-SM Model w.r.t. Mapping Neural Network Weights

Let  $w_i$  be a generic symbol representing an internal weight of the mapping neural network. The sensitivity of the Neuro-SM model w.r.t.  $w_i$  provides gradient information for efficient training of the Neuro-SM model. Here we derive the sensitivity formulas for the proposed analytical Neuro-SM.

**DC sensitivity:** In dc case, the sensitivity of the output current of the analytical Neuro-SM model  $I_f$  w.r.t.  $w_i$  is

$$\frac{\partial I_f}{\partial w_i} = \left( \frac{\partial \mathbf{I}_c^T}{\partial \mathbf{V}_{c,dc}} \right)^T \cdot \frac{\partial \mathbf{V}_{c,dc}}{\partial w_i} = \mathbf{G}_c \cdot \frac{\partial f_{ANN}(\mathbf{V}_{f,dc}, \mathbf{w})}{\partial w_i} \quad (3.7)$$

where  $\mathbf{G}_c$  is the dc conductance matrix of the coarse model, and  $\partial f_{ANN}(\mathbf{V}_{f,dc}, \mathbf{w}) / \partial w_i$  is

the first order derivative computed by neural network backpropagation [2].

**Small-signal sensitivity:** The sensitivity for Y-parameters of the analytical Neuro-SM model due to changes in the mapping neural network can be derived as

$$\begin{aligned} \frac{\partial \mathbf{Y}_f}{\partial w_i} = & \mathbf{Y}_c \Big|_{V_c, Bias} \cdot \left( \frac{\partial^2 \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f \partial w_i} \Big|_{\mathbf{v}_f = V_f, Bias} \right)^T + \\ & \left( \sum_{j=1,2} \frac{\partial \mathbf{Y}_c}{\partial v_{cj}} \Big|_{v_c = V_c, Bias} \cdot \frac{\partial f_{ANNj}(\mathbf{v}_f, \mathbf{w})}{\partial w_i} \Big|_{\mathbf{v}_f = V_f, Bias} \right) \cdot \left( \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f} \Big|_{\mathbf{v}_f = V_f, Bias} \right)^T \end{aligned} \quad (3.8)$$

This equation includes two derivative terms. The first term has the second order derivative of the neural network  $\mathbf{f}_{ANN}(\mathbf{v}_f, \mathbf{w})$ , which is the differentiation of the Jacobian matrix  $\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w}) / \partial \mathbf{v}_f$  w.r.t. the mapping neural network weight  $w_i$ . This second order derivative can be achieved by the adjoint neural network sensitivity analysis [7]. The second term is the sensitivity of the coarse model Y-parameter, which is dependent on the mapped dc bias voltages thus the neural network weights. Here  $v_{cj} = f_{ANNj}(\mathbf{v}_f, \mathbf{w})$ ,  $j = 1, 2$ , represents the coarse input signal at the coarse input port 1 (if  $j = 1$ ) or port 2 (if  $j = 2$ ). By converting Y-parameters to S-parameters, sensitivity for S-parameters can be subsequently obtained.

**Large-signal sensitivity:** Equation (3.9) shows the sensitivity of the output current of the proposed analytical Neuro-SM model at a generic harmonic frequency  $\omega_k$  ( $k = 0, 1, 2, \dots, N_H$ ). It is achieved by differentiating (3.4) w.r.t. the mapping neural network weight

$w_i$ .

$$\begin{aligned} \frac{\partial \mathbf{I}_f(\omega_k)}{\partial w_i} &= \frac{\partial \mathbf{I}_{c,NL}(\omega_k)}{\partial w_i} \\ &= \frac{1}{N_T} \sum_{n=0}^{N_T-1} \left( \mathbf{G}_c \Big|_{\mathbf{v}_c = \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})} + j\omega_k \cdot \mathbf{C}_c \Big|_{\mathbf{v}_c = \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})} \right) \cdot \frac{\partial \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})}{\partial w_i} \cdot \mathbf{W}_N(n, k) \end{aligned} \quad (3.9)$$

In general, using a standard sensitivity technique, the sensitivity of a circuit current w.r.t. any parameter in the nonlinear circuit would require either perturbation or adjoint sensitivity. Here we derive a much simpler sensitivity formulation for training shown in (3.9) without involving perturbation or adjoint sensitivity analysis. This is made possible by our formulation of training, where the fine input signals  $\mathbf{v}_f$  would be fixed by training data. In (3.9),  $\mathbf{G}_c = (\partial \mathbf{i}_c^T / \partial \mathbf{v}_c)^T$  and  $\mathbf{C}_c = (\partial \mathbf{q}_c^T / \partial \mathbf{v}_c)^T$  are the nonlinear conductance and capacitance matrices of the coarse model evaluated at the time point  $t_n$  at the mapped signal  $\mathbf{v}_c = \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})$ . In the case where the coarse model has separate linear and nonlinear parts, the sensitivity is the summation of that of the linear and nonlinear parts

$$\frac{\partial \mathbf{I}_f(\omega_k)}{\partial w_i} = \frac{\partial \mathbf{I}_{c,L}(\omega_k)}{\partial w_i} + \frac{\partial \mathbf{I}_{c,NL}(\omega_k)}{\partial w_i}, \quad k = 0, 1, \dots, N_H \quad (3.10)$$

where  $\partial \mathbf{I}_{c,L}(\omega_k) / \partial w_i$  is derived from (3.5) as

$$\frac{\partial \mathbf{I}_{c,L}(\omega_k)}{\partial w_i} = \mathbf{Y}_{c,L}(\omega_k) \cdot \frac{1}{N_T} \sum_{n=0}^{N_T-1} \frac{\partial \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})}{\partial w_i} \cdot \mathbf{W}_N(n, k) \quad (3.11)$$

In (3.11),  $\mathbf{Y}_{c,L}(\omega_k)$  is the admittance matrix of the coarse linear subcircuit at  $\omega_k$ , and  $\partial \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w}) / \partial w_i$  is the result computed from neural network backpropagation [2].

### 3.3.3 Exact Sensitivity Analysis of the Analytical Neuro-SM Model w.r.t. Coarse Model Parameters

The analytical Neuro-SM model can be incorporated into a circuit simulator after being trained utilizing the sensitivity formulation discussed in Subsection 3.3.2. When coarse model parameters need to be treated as variables during circuit optimization, the sensitivity of the circuit response w.r.t. coarse model parameters becomes useful. Now we consider the sensitivity of the circuit response, denoted by  $\mathcal{R}$ , with respect to a generic design variable  $x$  in the coarse model part of the Neuro-SM model.

**DC sensitivity:** Let  $V_{f,dc}$  and  $\hat{V}_{f,dc}$  be the original and adjoint dc port voltages of the analytical Neuro-SM model obtained by solving the original nonlinear circuit and its linear adjoint circuit [103], respectively. The dc sensitivity is

$$\frac{\partial \mathcal{R}}{\partial x} = -\hat{V}_{f,dc}^T \cdot \frac{\partial \mathbf{I}_f}{\partial x} = -\hat{V}_{f,dc}^T \cdot \left. \frac{\partial \mathbf{I}_c}{\partial x} \right|_{V_{c,DC} = f_{ANN}(V_{f,dc}, \mathbf{w})} \quad (3.12)$$

where  $\partial \mathbf{I}_c / \partial x$  is the sensitivity in the coarse model evaluated after the original voltage

$V_{f,dc}$  has been mapped by the mapping neural network. The contribution of the Neuro-SM model to the adjoint circuit is an admittance matrix evaluated at the mapped dc voltage

$$\frac{\partial \mathbf{I}_{f,DC}^T}{\partial \mathbf{V}_{f,DC}} = \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{V}_{f,DC}, \mathbf{w})}{\partial \mathbf{V}_{f,DC}} \cdot \frac{\partial \mathbf{I}_{c,DC}^T}{\partial \mathbf{V}_{c,DC}} \Bigg|_{V_{c,DC} = f_{ANN}(\mathbf{V}_{f,DC}, \mathbf{w})} \quad (3.13)$$

**Small-signal sensitivity:** Let  $\mathbf{v}_f$  and  $\hat{\mathbf{v}}_f$  be the original and adjoint voltages [103] at the terminals of the analytical Neuro-SM model obtained by performing small-signal simulation of the nonlinear circuit. Let  $v_{fi}$  be the fine voltage signal for port 1 (if  $i = 1$ ) or port 2 (if  $i = 2$ ). The sensitivity is evaluated by

$$\frac{\partial \mathcal{R}}{\partial x} = -\hat{\mathbf{v}}_f^T \cdot \left( \frac{\partial \mathbf{Y}_f}{\partial x} + \sum_{i=1,2} \frac{\partial \mathbf{Y}_f}{\partial v_{fi}} \cdot \frac{\partial v_{fi}}{\partial x} \right) \cdot \mathbf{v}_f \quad (3.14)$$

In (3.14),  $\partial \mathbf{Y}_f / \partial x$  is computed at the mapped voltage signal as

$$\frac{\partial \mathbf{Y}_f}{\partial x} = \frac{\partial \mathbf{Y}_c}{\partial x} \Bigg|_{V_{c,Bias}} \cdot \left( \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f} \Bigg|_{\mathbf{v}_f = \mathbf{V}_{f,Bias}} \right)^T \quad (3.15)$$

where  $\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w}) / \partial \mathbf{v}_f$  is achieved by extending backpropagation towards the input neurons of the mapping neural network [2].  $\partial \mathcal{R} / \partial x$  is also affected by the bias

dependency of the small-signal solution of the Neuro-SM shown by the second term in the bracket of (3.14), where  $\partial Y_f / \partial v_{fi}$  is derived in (3.16), and  $\partial v_{fi} / \partial x$  is the sensitivity of the dc bias of  $v_f$  obtained using (3.12) with  $\mathcal{R}$  replaced with  $v_{fi}$ .

$$\begin{aligned} \frac{\partial Y_f}{\partial v_{fi}} = & \mathbf{Y}_c \Big|_{\mathbf{v}_c = \mathbf{V}_{c,Bias}} \cdot \left( \frac{\partial^2 \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f \partial v_{fi}} \Big|_{\mathbf{v}_f = \mathbf{V}_{f,Bias}} \right)^T + \\ & \left( \sum_{j=1,2} \frac{\partial \mathbf{Y}_c}{\partial v_{cj}} \Big|_{\mathbf{v}_c = \mathbf{V}_{c,Bias}} \cdot \frac{\partial \mathbf{f}_{ANNj}(\mathbf{v}_f, \mathbf{w})}{\partial v_{fi}} \Big|_{\mathbf{v}_f = \mathbf{V}_{f,Bias}} \right) \cdot \left( \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f} \Big|_{\mathbf{v}_f = \mathbf{V}_{f,Bias}} \right)^T \end{aligned} \quad (3.16)$$

Here  $v_{cj}$  is defined as the coarse voltage signal of the Neuro-SM model for port 1 (if  $j = 1$ ) or port 2 (if  $j = 2$ ).

**Large-signal sensitivity:** Define complex vectors  $\mathbf{V}_f(\omega_k)$  and  $\hat{\mathbf{V}}_f(\omega_k)$  as the original and adjoint voltages of the analytical Neuro-SM model at harmonic frequency  $\omega_k$  ( $k = 0, 1, 2, \dots, N_H$ ). Utilizing harmonic balance sensitivity [104], the sensitivity of large-signal response w.r.t.  $x$  is derived as in (3.17), where  $\partial \mathbf{i}_{c,NL} / \partial x$  and  $\partial \mathbf{q}_{c,NL} / \partial x$  are the sensitivities of the nonlinear current and charge of the coarse model evaluated at time  $t_n$  from the mapped voltage signals  $\mathbf{v}_c = \mathbf{f}_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})$ .

$$\frac{\partial \mathcal{R}}{\partial x} = \begin{cases} -\sum_{k=0}^{N_H} \text{Real} \left( \hat{\mathbf{V}}_f^T(\omega_k) \cdot \left( \frac{1}{N_T} \sum_{n=0}^{N_T-1} \frac{\partial \mathbf{i}_{c,NL}}{\partial x} \Big|_{\mathbf{v}_c=f_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})} \cdot \mathbf{W}_N(n, k) \right)^* \right) \\ \text{if } x \text{ belongs to a nonlinear current branch in the coarse model} \\ -\sum_{k=0}^{N_H} \text{Imag} \left( \hat{\mathbf{V}}_f^T(\omega_k) \cdot \left( \frac{1}{N_T} \sum_{n=0}^{N_T-1} \left( j\omega_k \cdot \frac{\partial \mathbf{q}_{c,NL}}{\partial x} \Big|_{\mathbf{v}_c=f_{ANN}(\mathbf{v}_f(t_n), \mathbf{w})} \right) \cdot \mathbf{W}_N(n, k) \right)^* \right) \\ \text{if } x \text{ belongs to a nonlinear charge branch in the coarse model} \end{cases} \quad (3.17)$$

The contribution of the Neuro-SM model to the adjoint HB equations is the admittance matrix shown in

$$\frac{\partial \mathbf{I}_f^T(\omega_k)}{\partial \mathbf{V}_f(\omega_l)} = \frac{1}{N_T} \sum_{n=0}^{N_T-1} \left( \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f(t_n), \mathbf{w})}{\partial \mathbf{v}_f} \right) \times \left( \mathbf{G}_c \Big|_{\mathbf{v}_c=f_{ANN}(\mathbf{v}_f, \mathbf{w})} + j\omega_k \cdot \mathbf{C}_c \Big|_{\mathbf{v}_c=f_{ANN}(\mathbf{v}_f, \mathbf{w})} \right)^T \cdot \mathbf{W}_N(n, k) \cdot \mathbf{W}_N^*(n, l) \quad (3.18)$$

which is to be added into the admittance matrix of the overall adjoint circuit.  $\mathbf{G}_c$  and  $\mathbf{C}_c$  are the same as those in (3.9). If the coarse model has separate linear and nonlinear parts, the contribution from the nonlinear subcircuit is the same as in (3.18), and the contribution of the linear part is

$$\frac{\partial \mathbf{I}_{C,L}^T(\omega_k)}{\partial \mathbf{V}_f(\omega_l)} = \frac{1}{N_T} \sum_{n=0}^{N_T-1} \left( \frac{\partial \mathbf{f}_{ANN}^T(\mathbf{v}_f, \mathbf{w})}{\partial \mathbf{v}_f} \right) \mathbf{W}_N(n, k) \cdot \mathbf{W}_N^*(n, l) \cdot \mathbf{Y}_{c,L}^T(\omega_k) \quad (3.19)$$

In the derivations of both the model sensitivity and the circuit sensitivity in Subsections 3.3.2 and 3.3.3, we notice that the sensitivity computations in the dc, small-signal, and large-signal cases involve derivatives of the coarse model evaluated at mapped voltage signals, and the derivatives of the mapping neural network achieved by backpropagation [2] and the adjoint neural network sensitivity analysis [7].

### **3.4 Proposed Training Algorithm for the Analytical Neuro-SM Model**

The Neuro-SM model will not be good unless the mapping neural network is trained by fine data. The purpose of Neuro-SM model training is to let the mapping neural network  $f_{ANN}$  learn the necessary relationship between the coarse and fine signals, such that the response of the Neuro-SM model matches that of the fine model (device data). However, we may not have the voltage and current signals as direct training data as required by conventional neural network training algorithms. In this section, we formulate the training algorithm using dc, bias dependent S-parameter data, and optionally large-signal harmonic data from the fine model.

Our training technique extends the 2-phase training of [48] from linear/passive device modeling to nonlinear/active device modeling for the analytical Neuro-SM model. The overall training has two phases, initialization and formal training.

#### **3.4.1 Initialization of the Mapping Neural Network**

The mapping neural network is firstly initialized by a preliminary training to learn unit mapping, where the weights  $\mathbf{w}$  are adjusted in order to

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{p \in P_o} \|\mathbf{v}_f^p - \mathbf{v}_c^p\|^2 = \min_{\mathbf{w}} \frac{1}{2} \sum_{p \in P_o} \|\mathbf{v}_f^p - \mathbf{f}_{ANN}(\mathbf{v}_f^p, \mathbf{w})\|^2 \quad (3.20)$$

where  $p$  is a data index, and  $P_o$  is an index set for all training data. Training data can be obtained by assigning  $[v_{f1}, v_{f2}]$  in a grid form across the entire operation range of the device. The initialization phase leads to  $\mathbf{v}_c = \mathbf{v}_f$ , making the overall Neuro-SM model to be equal to the coarse model, before actual device data is used in the training of the neural network.

### 3.4.2 Formal Training of the Mapping Neural Network

The mapping neural network needs to be further trained by actual device data in a formal training phase, in order to exceed the performance of the given coarse model. Formal training can be done with either dc and bias dependent S-parameter data or harmonic data. The exact sensitivity analysis described in Subsection 3.3.2 provides the gradient information required by the training algorithm. Compared to the circuit-based Neuro-SM, the analytical Neuro-SM is a more compact model. The number of circuit equations for both model simulation and sensitivity analysis used in the analytical Neuro-SM is less than that of the circuit-based Neuro-SM. Thus the proposed analytical formulation and sensitivity help achieve more efficient training than the circuit-based Neuro-SM in [14]. Figures 3.2 and 3.3 show the training diagrams of the analytical Neuro-SM model.

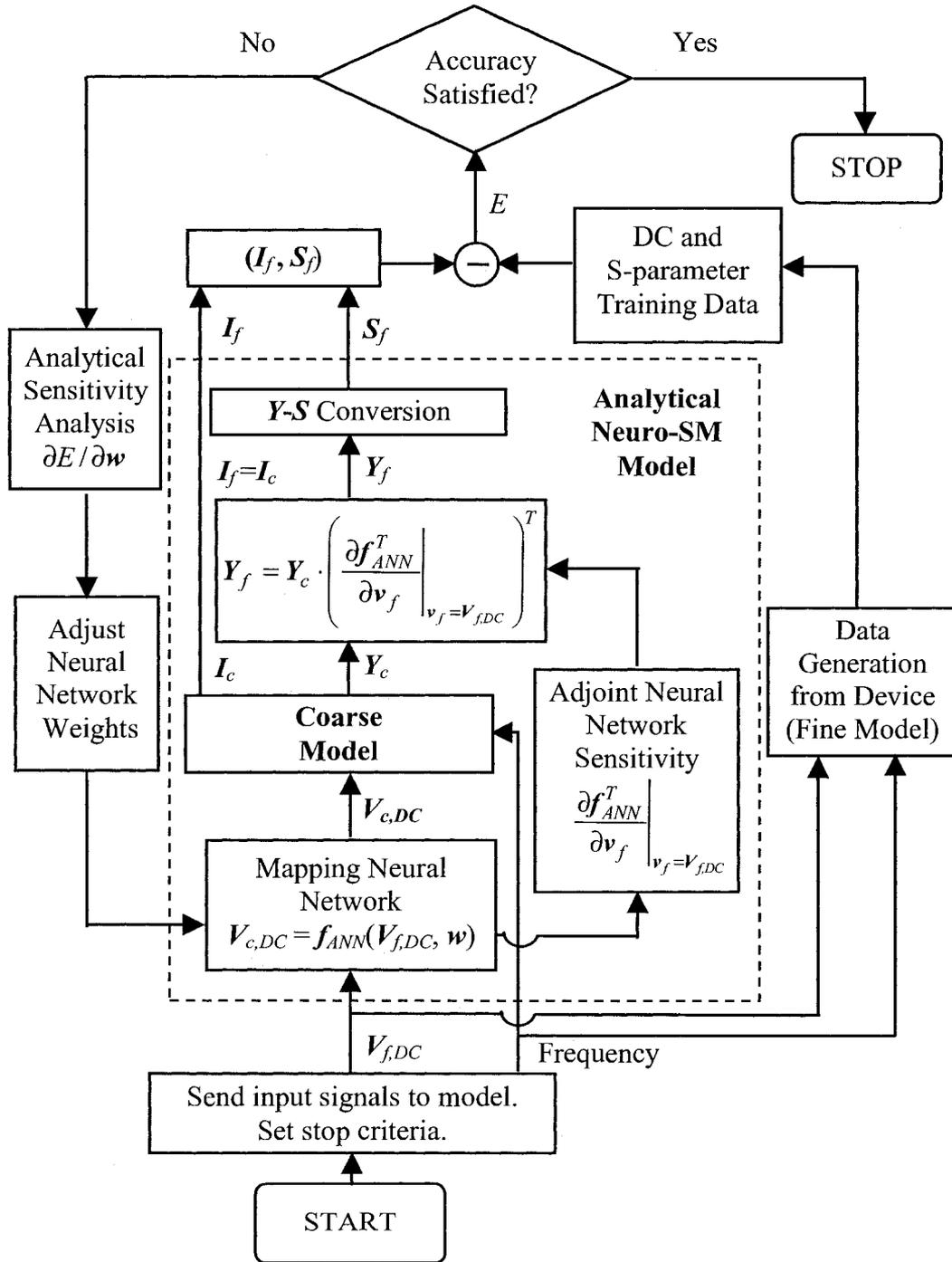


Figure 3.2: Block diagram for dc and small-signal training of the proposed analytical Neuro-SM model.

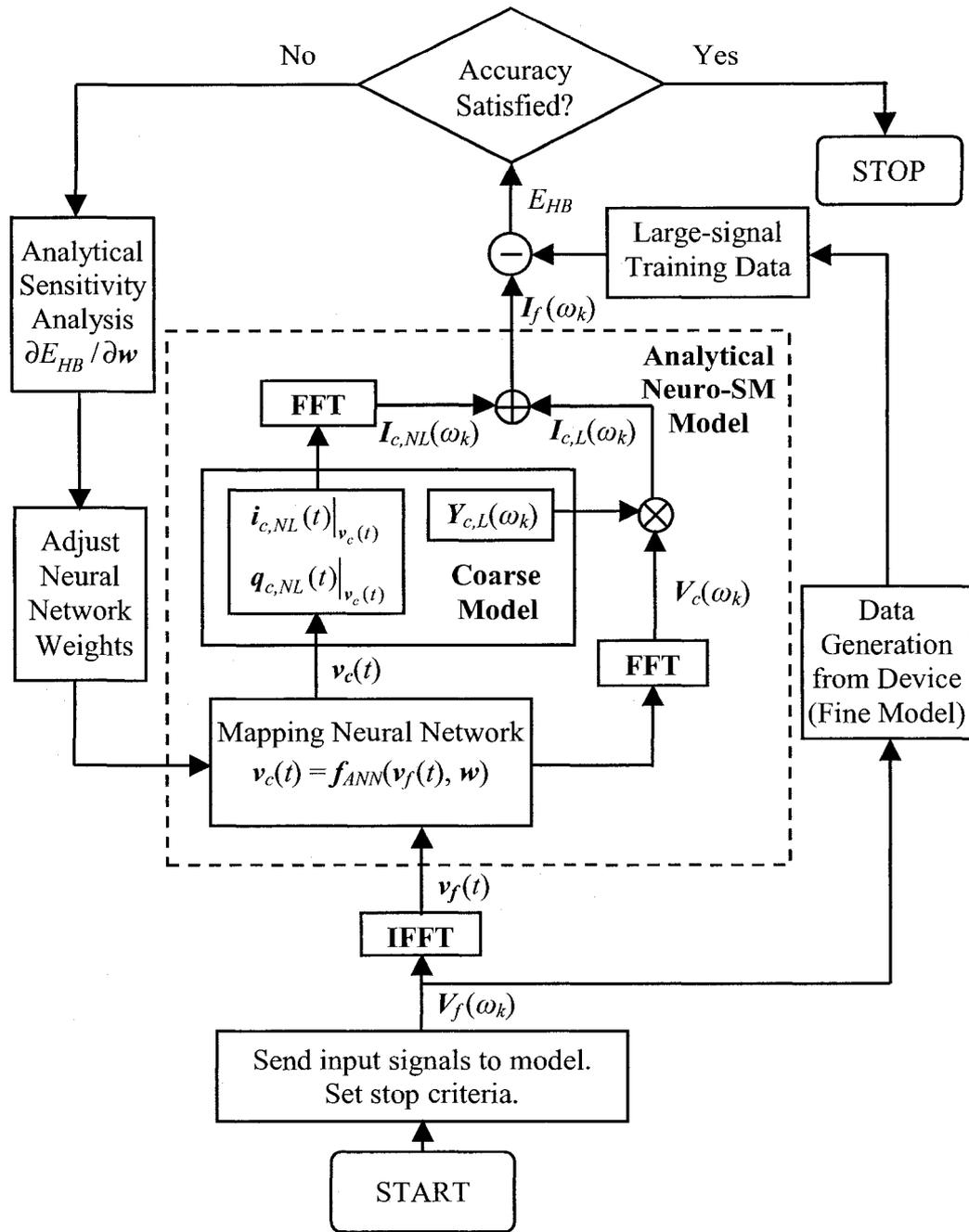


Figure 3.3: Block diagram for large-signal training of the proposed analytical Neuro-SM model. As observed, the input voltages are firstly passed to the mapping neural network to be mapped (modified) before being applied to the coarse model. FFT denotes fast Fourier transform, and IFFT means inverse FFT.

**DC and small-signal training:** The mapping neural network is trained to minimize the dc and S-parameter errors between the model and data at all combinations of dc biases and frequency points. During training, the mapping neural network weights are adjusted according to the gradient information of the training error through sensitivity analysis of the analytical Neuro-SM model. Since the evaluation and sensitivity analysis of Neuro-SM are performed at different dc biases and frequency points, the CPU speedup from each evaluation can be accumulated into a large CPU saving compared to the circuit-based Neuro-SM training in [14].

**Large-signal training:** Large-signal training data contains output power of each harmonic at different combinations of biases, input power levels, and fundamental frequencies. The objective of large-signal training is to minimize the difference between the HB response of the Neuro-SM model and harmonic data for all combinations. Large-signal sensitivity described in the previous section can be used to provide gradient information for training of the analytical Neuro-SM model. The efficiency in evaluating the analytical Neuro-SM model and its sensitivity for each combination of bias, input power level, and fundamental frequency becomes more significant when many such combinations are used in training.

**Accuracy test:** After training, the accuracy of the final model can be tested by comparing the Neuro-SM model with a separate set of data called test data. The test data can be dc, small-signal S-parameters, or large-signal harmonic data.

### 3.4.3 Use of the Trained Analytical Neuro-SM Model

After the analytical Neuro-SM model is trained, it can be plugged into an overall circuit for circuit simulation and design. The Neuro-SM model can be incorporated into a circuit simulator either internally as a new type of device model, or externally as a user-defined model. To implement it internally, we program the neural network mapping to adjust the relationship of the port current and voltage signals of the existing device model using the formulas in the proposed analytical formulation. To implement Neuro-SM externally, we construct the circuit-based form of Figure 3.1(a) using the neural network weights from the trained analytical Neuro-SM model. These neural network weights are passed to the controlling functions of the controlled sources in the circuit-based Neuro-SM. The voltage/current relationship of the Neuro-SM model required by the circuit simulator is that between  $v_f$  and  $i_f$ , which is obtained from the Neuro-SM model through the mapping of coarse model signals as in Figure 3.1.

## 3.5 Discussions

The format of the Neuro-SM model presented so far is to map voltage signals between coarse and fine models. This format can be expanded to a mixed mapping case, where the mapping is for a mixture of port voltage and current signals. For example, the input of an HBT device are base current signal and collector voltage signal, i.e.,  $[i_{f1}, v_{f2}]^T$ . The mapping neural network will map the fine model input signals to the voltage/current input signals of the coarse model, such that the modified coarse model response will match the fine outputs.

For simplification purpose, we used a 2-port device notation in explaining the Neuro-

SM technique. This approach can be further generalized to  $n$ -port networks, where the notation and equations in the previous sections are extended accordingly. For example, the mapping neural network will contain  $n$  input neurons and  $n$  output neurons. The  $n$  external input signals, i.e., fine signals, will be supplied through the mapping neural network to the  $n$ -port coarse model.

The mapping introduced in Sections 3.3 and 3.4 uses only the externally accessible signals of the coarse model, i.e., port voltage or current signals of the 2-port coarse model. The independence from the coarse model internal information makes it convenient for the Neuro-SM to be implemented and used with various coarse models. After being trained by the proposed training algorithm, the Neuro-SM model can be used across different circuit simulators including simulators where the Neuro-SM has not been pre-programmed.

In the formulation described so far, the gap between the Neuro-SM model and the fine data will be minimized, but not necessarily eliminated. This means that the mapping is not necessarily exact. For coarse models such as intrinsic FET (or FET model including parasitic networks) that have fewer (or more) internal nodes, the mapping will make a significant (or incremental) accuracy improvement over that of the coarse model. Such a Neuro-SM is very suitable for intrinsic FET modeling which is usually the major challenge in the development of new FET models.

The Neuro-SM concept can also be extended to alternative formulations exploiting the information inside the coarse model. For example, additional mappings can be performed on the terminal charges of the FET device. Another example is to map the

voltage/current signals of each circuit branch inside the coarse model. Equations for dc, small-signal, and large-signal cases can be derived in a similar way as those in Section 3.3 by involving separate mappings for charge signals or coarse model internal signals. The potential benefit of these formulations is a further increase of the final model accuracy. The flexibility in using the trained model may be reduced because the coarse model internal signals (such as charge signal) may not be accessible in a circuit simulator. Such alternative models may be used only if the mapping is programmed internally in the simulation software.

The neural network used for the mapping is not necessarily unique. In other words, the neural network internal weights can be different if the mapping is trained differently. This does not affect the Neuro-SM as long as the final neural network gives a correct map between the coarse and fine signals.

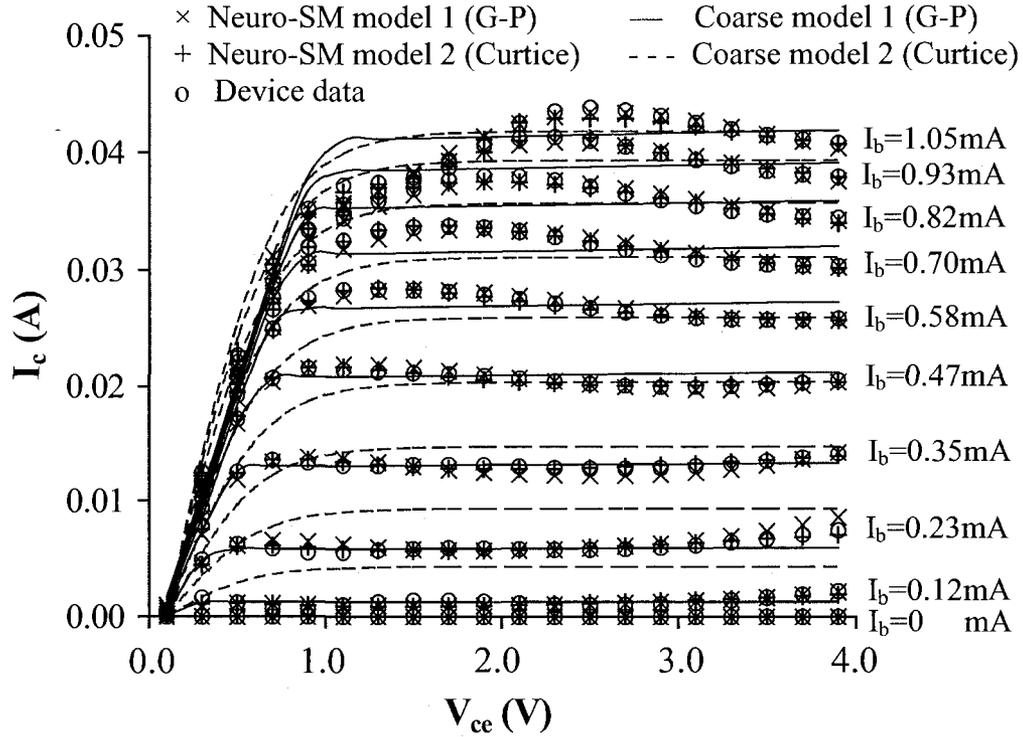
Under certain conditions, the theoretical existence of a Neuro-SM model that exactly matches the fine device behavior can be ensured. Examples of such conditions are: if the coarse model topology is perfect, the mapping is applied to the signals of the individual branches in the coarse model, and the output signal of each branch is controllable by its input signals. In this case, the dc, small-, and large-signal mappings corresponding to those in Section 3.3 are exact. In general, a mapping neural network exists for the Neuro-SM model to match the fine device data more closely (although not necessarily exactly) than possible by the coarse model alone.

## 3.6 Application Examples

### 3.6.1 Analytical Neuro-SM Modeling of SiGe HBT

In this example, the proposed analytical Neuro-SM is used to model a SiGe HBT device with irregular nonlinear measured dc behavior [105]. We use three implementations of the Neuro-SM technique: (a) circuit-based Neuro-SM with perturbation sensitivity implemented in *Agilent-ADS* [106], (b) circuit-based Neuro-SM with adjoint neural network sensitivity used in [14], and (c) proposed analytical Neuro-SM and its sensitivity implemented in *NeuroModelerPlus* [107]. In our *ADS* implementation, the gradient information required for the Neuro-SM model training is achieved by perturbing each weight in the mapping neural network. The circuit-based Neuro-SM in *NeuroModelerPlus* can utilize the exact adjoint sensitivity to train the mapping neural network. For the analytical Neuro-SM in *NeuroModelerPlus*, the sensitivity analyses described in Subsections 3.3.2 and 3.3.3 are implemented, and applied for model training.

Two types of existing models, Gummel-Poon (G-P) model [83] and Curtice cubic model [14], [79], are used as coarse models for mapping. Figure 3.4 shows improved model accuracy by the Neuro-SM technique to map the existing device models. As seen in Figure 3.4, without mapping, the two models at their best provide only an approximation of the device behavior and lack the complicated details seen in the device data. With the mapping neural network, both models can be mapped to the device data with good accuracy. This is because the neural network training can automatically adjust the mapping differently according to the needs of the specific coarse model used.



**Figure 3.4: Comparison of the device dc data, the dc responses of the existing models (without mapping), and the Neuro-SM models in the HBT example.**

Table 3.1 shows that the dc sensitivity of the analytical Neuro-SM model from the analytical sensitivity analysis matches well with the perturbation result, confirming the validity of our new sensitivity technique. Different numbers of hidden neurons (10, 15, 20) for the mapping neural networks have been used in training. The testing accuracy for the Neuro-SM with 10, 15, and 20 hidden neurons by mapping the Gummel-Poon (or Curtice) model were 0.85%, 0.91%, and 1.40% (or 0.88%, 0.74%, and 0.93%), respectively. Mapping neural networks with 10 or 15 hidden neurons are found suitable for this example. In general, fewer (more) hidden neurons are needed if the coarse model is good (poor). Table 3.2 shows a detailed comparison of the training and testing errors

between the coarse and Neuro-SM models with the mapping neural network of 10 hidden neurons. Table 3.3 compares the training time of the three Neuro-SM implementations. Training was done with 200 sets of dc data and the CPU time was recorded for 100 training iterations on a Pentium IV 2.8GHz computer. Table 3.4 shows the model evaluation time comparison between the coarse models, the circuit-based Neuro-SM, and the proposed analytical Neuro-SM by performing 1000 Monte-Carlo analysis for 100 dc bias points.

**Table 3.1: Examples of sensitivity comparison in the HBT example. Sensitivity is done w.r.t. the mapping neural network weights and coarse model parameters. The Gummel-Poon model is used for mapping.**

	Neuro-SM sensitivity by perturbation	Proposed analytical Neuro-SM sensitivity	Difference (%)
$dI_c/dw_{11}$	2.2608e-03	2.2642e-03	0.15
$dI_c/dw_{51}$	6.3170e-03	6.2898e-03	0.43
$dI_c/dw_{42}$	1.5417e-02	1.5588e-02	1.10
$dI_c/dI_{sf}$	2.4239e-04	2.4207e-04	0.13
$dI_c/dN_f$	1.2706e+01	1.2705e+01	0.01
<p><math>I_c</math> is the collector current.  <math>w_{ij}</math> is a neural network synaptic weight.  <math>I_{sf}</math> and <math>N_f</math> are coarse model parameters.</p>			

**Table 3.2: Comparison of model accuracy in the HBT example. The values are average errors between the model and training/testing data. The proposed analytical Neuro-SM can retain the same accuracy as the circuit-based Neuro-SM.**

	Existing model without mapping	Circuit-based Neuro-SM model	Proposed analytical Neuro-SM model
Coarse Model 1 (Gummel-Poon)	1.93 % / 2.27 %	0.81 % / 0.85%	0.81 % / 0.85%
Coarse Model 2 (Curtice)	3.54 % / 4.03 %	0.83 % / 0.88%	0.83 % / 0.88%

**Table 3.3: Neuro-SM Training time comparison between several training techniques for the HBT example. Training was done with dc data only. The proposed technique is the most efficient.**

	Circuit-based Neuro-SM with perturbation	Circuit-based Neuro-SM with adjoint NN sensitivity	Proposed analytical Neuro-SM and sensitivity
Coarse Model 1 (Gummel-Poon)	30 mins	7 mins	2.5 mins
Coarse Model 2 (Curtice)	28.5 mins	6.7 mins	1.7 mins

**Table 3.4: Model evaluation time for 1000 Monte-Carlo analysis of 100 dc biases in the HBT example. Relative to the original coarse model, the computational overhead of the proposed analytical Neuro-SM is much less than the circuit-based Neuro-SM.**

	Coarse model without mapping	Circuit-based Neuro-SM model	Proposed analytical Neuro-SM model
Coarse Model 1 (Gummel-Poon)	19 secs	48 secs	30 secs
Coarse Model 2 (Curtice)	14 secs	27 secs	20 secs

This example extends the study of the Neuro-SM technique beyond that in [14] in three new directions. First, we applied different coarse models to demonstrate the flexibility of Neuro-SM. Second, new analytical sensitivities were utilized and compared with perturbation, validating the proposed sensitivity technique. Third, training of the new analytical Neuro-SM model, and a comparison of model accuracy, training CPU time, as well as evaluation time between the proposed analytical Neuro-SM and the circuit-based Neuro-SM in [14] were done, confirming that the proposed analytical Neuro-SM provides the best efficiency among the three model development methods in Table 3.3.

### **3.6.2 Analytical Neuro-SM Modeling of GaAs MESFET**

In this example, Neuro-SM is used to model the large-signal behavior of an ADS internal GaAs MESFET [14]. The three implementations described in Subsection 3.6.1 are utilized, i.e., circuit-based Neuro-SM with perturbation, circuit-based Neuro-SM with adjoint neural network sensitivity as in [14], and the proposed analytical Neuro-SM. The Neuro-SM models were trained with dc and bias dependent S-parameter data, and refined by large-signal harmonic data. The training data was generated in *ADS* by an internal Statz model [78] for convenient verification purpose. Two existing MESFET models are used as coarse models for mapping: the Curtice cubic model [79] and Materka model [80]. Harmonic data for refinement training was generated at different input power levels (1-5dBm) and fundamental frequencies (2-5GHz) with a harmonic frequency range up to 25GHz. Sensitivity formulas described in Subsection 3.3.2 were implemented and used for training of the analytical Neuro-SM models.

Similar to the example in Subsection 3.6.1, we extend the study of Neuro-SM beyond that in [14] in three new directions: applicability of Neuro-SM for different coarse models, new sensitivity validation, and comparison of the new analytical Neuro-SM with the original Neuro-SM of [14]. Figures 3.5, 3.6, and 3.7 show the comparisons of dc, small-, and large-signal behavior between coarse device models, mapped Neuro-SM models, and data. Notice that the mismatch between the coarse models and the data cannot be simply overcome by optimizing the model parameters alone. A structural change in the nonlinear model formulas is needed. This is achieved by Neuro-SM through the additional degree of freedom beyond that of the existing model due to the neural network mapping. Table 3.5 shows the dc, small-signal, and large-signal sensitivity verification. Different numbers of hidden neurons (10, 15, 20) for the mapping neural networks have been used in the training. The testing accuracy for the Neuro-SM with 10, 15, and 20 hidden neurons by mapping the Curtice (or Materka) model were 1.43%, 1.38%, and 1.72% (or 1.34%, 1.20%, and 1.40%), respectively. Tables 3.6, 3.7, and 3.8 show model accuracy, training time, and model speed comparison, further demonstrating that the proposed analytical Neuro-SM technique with its exact sensitivity can retain the same model accuracy as circuit-based Neuro-SM in [14] while achieving increased efficiency. The neural networks used in the tables are with 10 hidden neurons. In Table 3.7, training CPU time was recorded for 100 training iterations on a Pentium IV 2.8GHz computer.

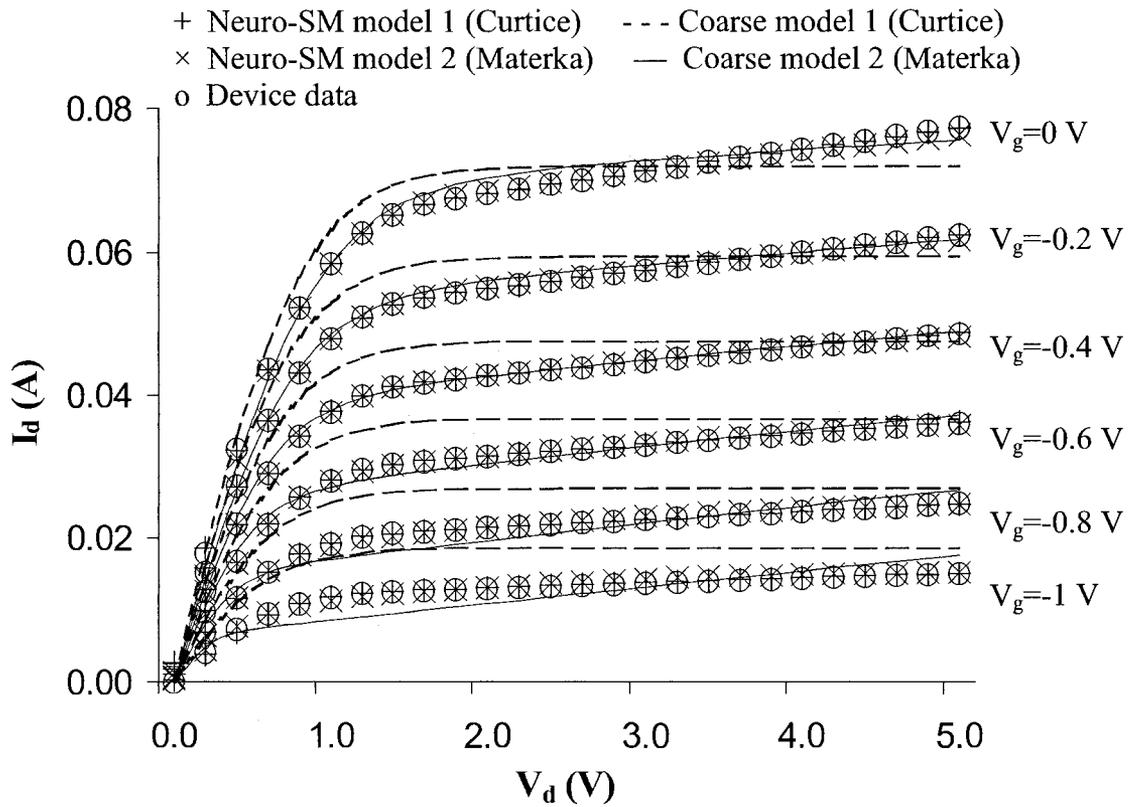


Figure 3.5: Comparison of the dc current between the original ADS solution (device data), existing models (without mapping), and Neuro-SM models in the MESFET example.

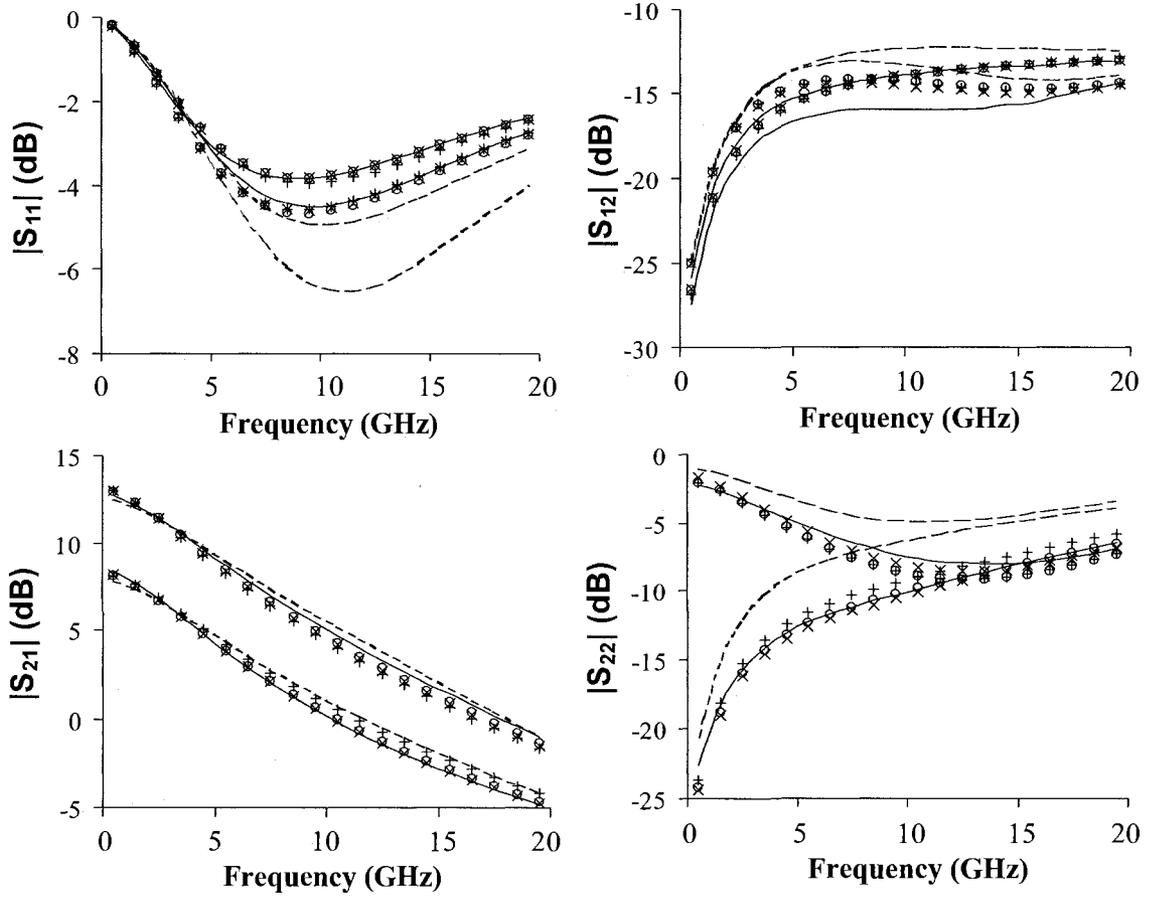


Figure 3.6: Comparison of the S-parameters between the original ADS solution (device data), existing models (without mapping), and Neuro-SM models in the MESFET example. The S-parameters are at 2 biases of  $(V_g, V_d)$  of  $(-0.8V, 4V)$  and  $(-0.2V, 1V)$ .

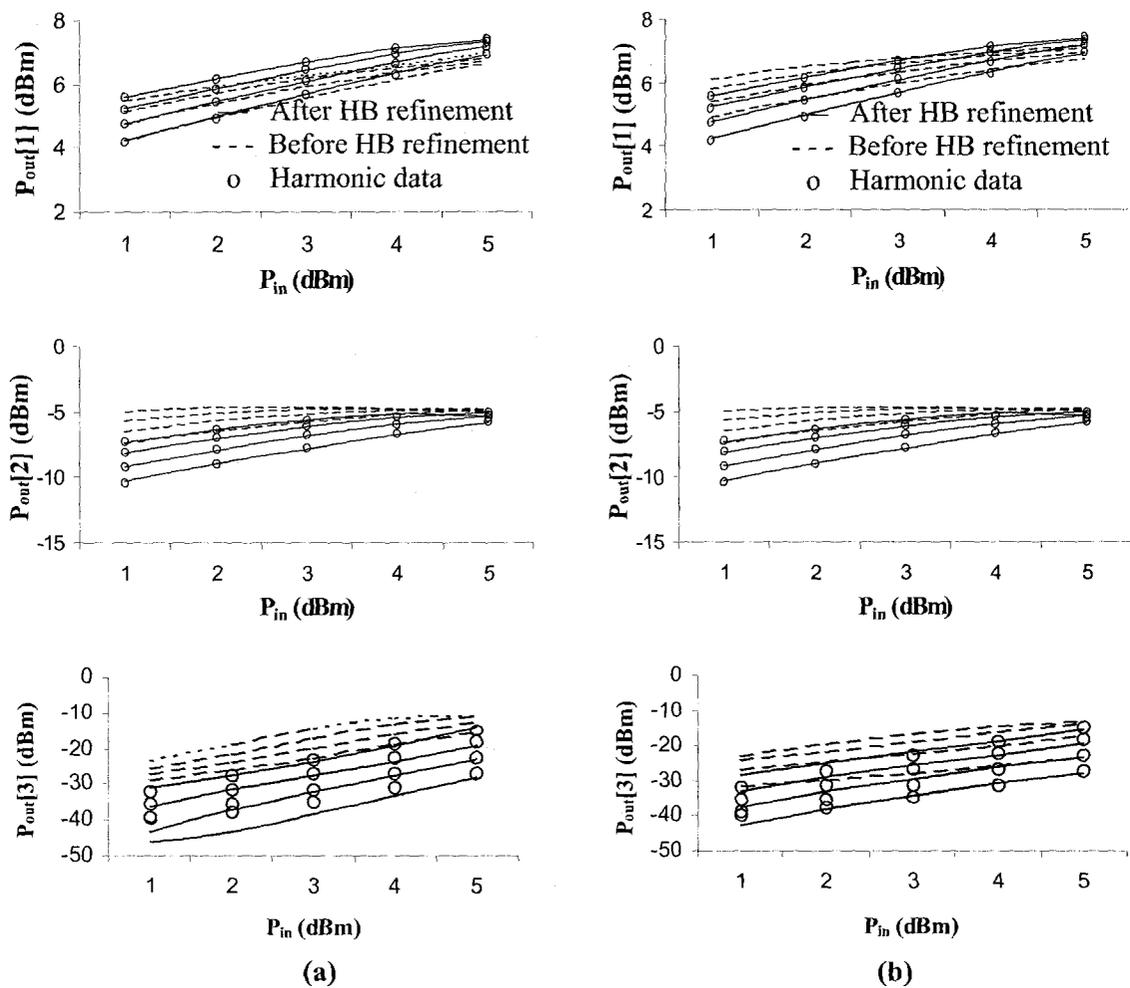


Figure 3.7: Comparison between the first three harmonic data and the HB response of the Neuro-SM models before/after HB refinement training in the MESFET example. Neuro-SM is applied to (a) Curtice model, and (b) Materka model.

**Table 3.5: Sensitivity comparison in the MESFET example. Sensitivity is calculated w.r.t. the mapping neural network weights and coarse model parameters. The Curtice model is used for the mapping.**

	Neuro-SM sensitivity by perturbation	Proposed analytical Neuro-SM sensitivity	Difference (%)
$dI_d/dw_{11}$	1.0234e-01	1.0171e-01	0.62
$dRS_{11}/dw_{31}$	1.2217e-01	1.2214e-01	0.02
$dIS_{11}/dw_{42}$	-4.1562e-02	-4.1541e-02	0.05
$dRS_{12}/dw_{51}$	-1.1381e-02	-1.1379e-02	0.02
$dIS_{12}/dw_{72}$	9.9543e-03	9.9344e-03	0.20
$dRS_{21}/da_3$	-4.1906e+01	-4.1902e+01	0.01
$dIS_{21}/dC_{ds}$	3.3082e+02	3.3248e+02	0.50
$dRS_{22}/dr$	1.2093e-01	1.2101e-01	0.07
$dIS_{22}/dC_{ds}$	-1.5024e+02	-1.5029e+02	0.03
$dI_d[1]/dw_{22}$	-3.1426e-01	-3.1400e-01	0.09
$dI_d[2]/da_2$	1.8342e+00	1.8451e+00	0.59
$dI_d[3]/dC_{gd}$	2.5145e+01	2.6349e+01	0.57

$I_d$  is the drain current.  
 $RS_{ij}$  and  $IS_{ij}$  ( $i, j = 1, 2$ ) are real and imaginary parts of S-parameters.  
 $I_d[k]$  ( $k = 1, 2, 3$ ) is large-signal current at the  $k^{th}$  harmonic frequency.  
 $w_{ij}$  is a neural network synaptic weight.  
 $a_3, C_{ds}, r, a_2,$  and  $C_{gd}$  are coarse model parameters.

**Table 3.6: Comparison of model accuracy in the MESFET example. The values are average errors between the model and training/testing data. The proposed analytical Neuro-SM can retain the same accuracy as the circuit-based Neuro-SM.**

	Existing model without mapping	Circuit-based Neuro-SM model	Proposed analytical Neuro-SM model
Coarse Model 1 (Curtice)	10.53 % / 10.66%	1.41 % / 1.43%	1.41 % / 1.43%
Coarse Model 2 (Materka)	6.44 % / 6.59%	1.26 % / 1.34%	1.26 % / 1.34%

**Table 3.7: Neuro-SM training time comparison between several training techniques for the MESFET example. Training was done with dc and S-parameter/harmonic data. The proposed technique is the most efficient.**

	Circuit-based Neuro-SM with perturbation	Circuit-based Neuro-SM with adjoint NN sensitivity	Proposed analytical Neuro-SM and sensitivity
Coarse Model 1 (Curtice)	120 / 42 mins	23 / 10 mins	11 / 3.3mins
Coarse Model 2 (Materka)	133 / 45 mins	25 / 13.3 mins	15 / 4 mins

**Table 3.8: Model evaluation time of dc and S-parameter sweeps at 150 biases, repeated for 1000 Monte-Carlo analyses in the MESFET example. Relative to the original coarse model, the computational overhead of the proposed analytical Neuro-SM is only marginal.**

	Coarse model without mapping	Circuit-based Neuro-SM model	Proposed analytical Neuro-SM model
Coarse Model 1 (Curtice)	6.7 mins	14 mins	7.5 mins
Coarse Model 2 (Materka)	6 mins	15 mins	6.7 mins

### 3.6.3 Analytical Neuro-SM Modeling of a HEMT Trained with Physics-Based Device Data

The high electron mobility transistor (HEMT) [108] device is important in high frequency circuit design. Physics-based numerical simulators [109] and equivalent circuit models [81] have been used for HEMT modeling. In this example, Neuro-SM is used to learn from physics-based data of a HEMT device. Training data (dc and bias dependent S-parameter data) was generated from a physics-based device simulator, *MINIMOS* [109], by solving the device Poisson equations. The HEMT structure used in setting up the physics-based simulator is shown in Figure 3.8. It was modeled by three Neuro-SM implementations (circuit-based Neuro-SM with perturbation, circuit-based Neuro-SM with the adjoint neural network sensitivity of [14], and the proposed analytical Neuro-SM) with three different coarse models, i.e., the Curtice [79], the Statz [78], and the Chalmers (Angelov) [81] models, resulting in 9 cases for extensive studies of the Neuro-SM technique.

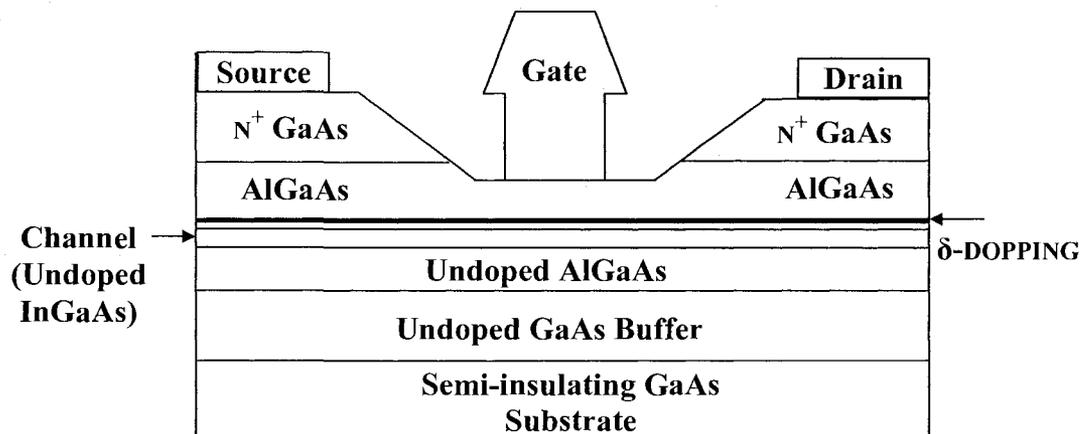
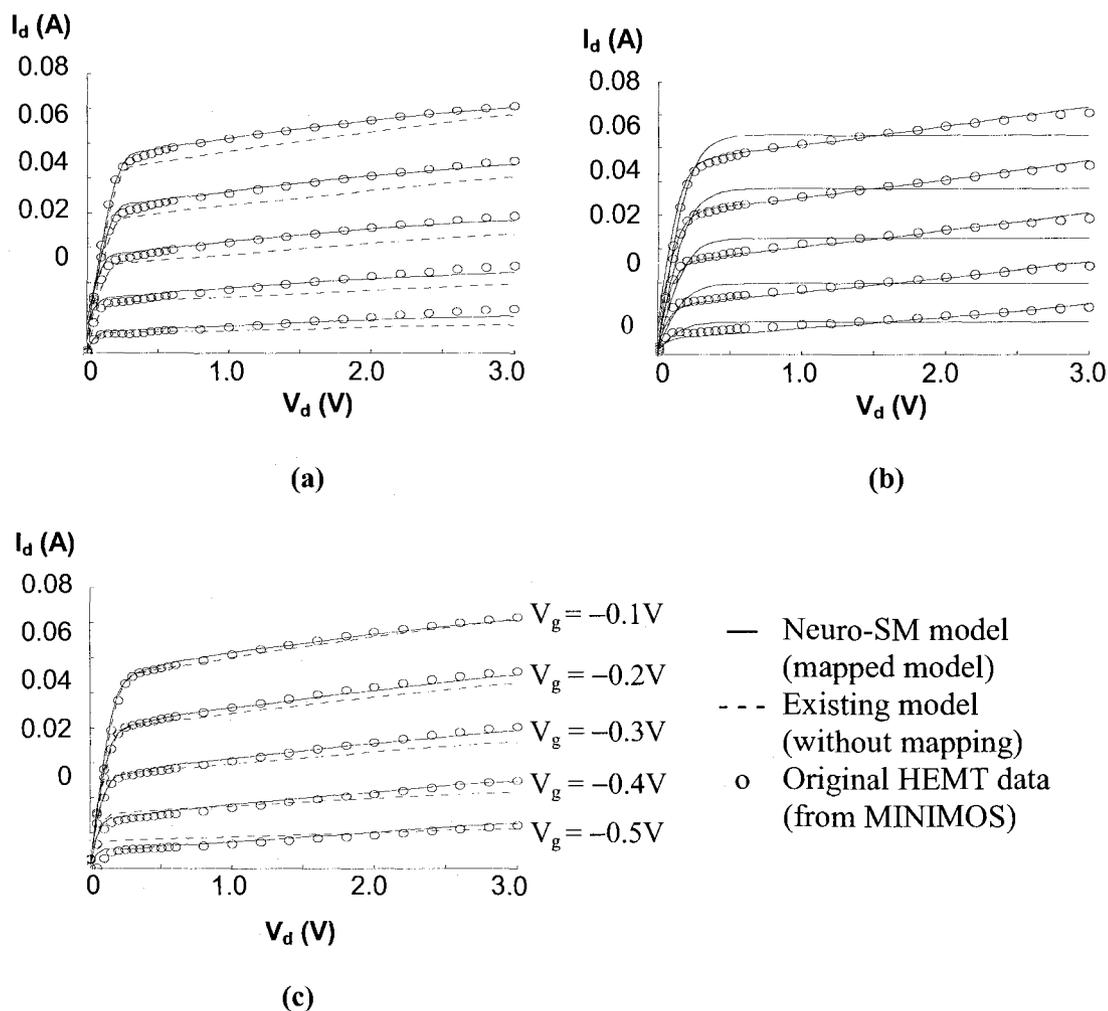
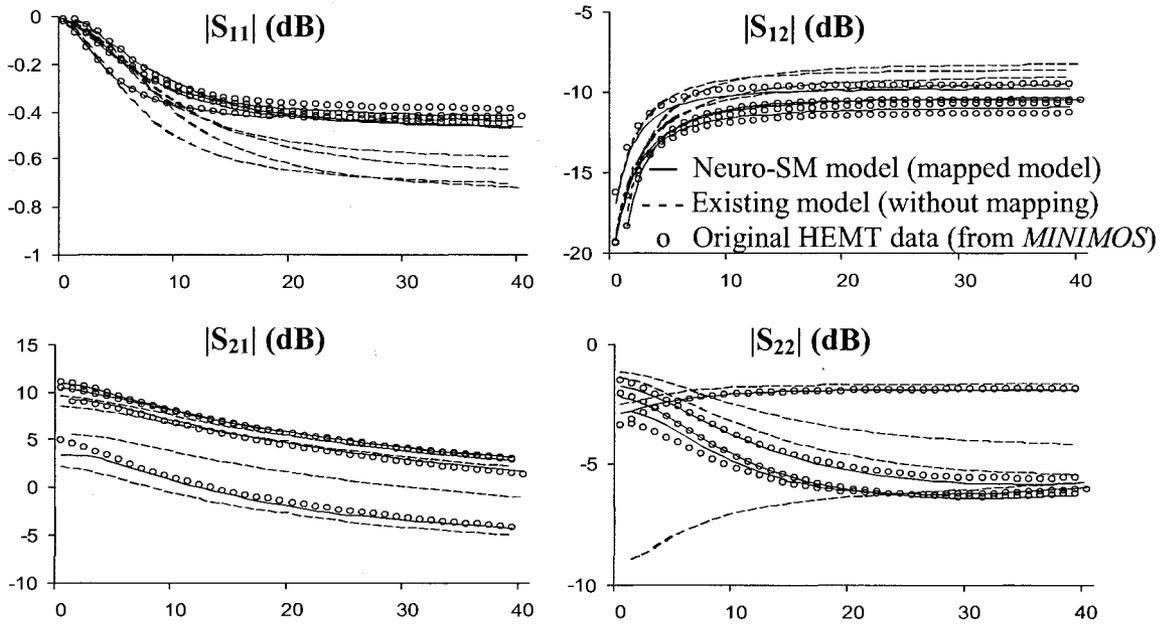


Figure 3.8: Physical structure of a HEMT device used for generating fine data in *MINIMOS* to train Neuro-SM models.

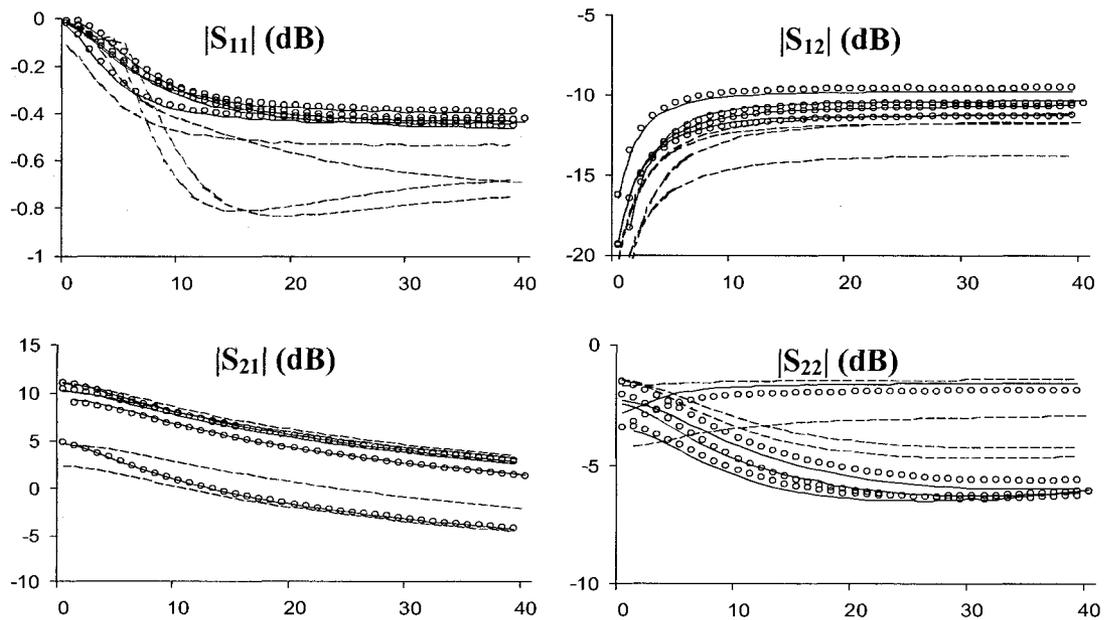
A comparison of the Neuro-SM models and the original physics data is shown in Figures 3.9 and 3.10 for different coarse models (Curtice, Statz, and Chalmers). Mapping neural networks with 10 to 15 hidden neurons are found suitable for this example.



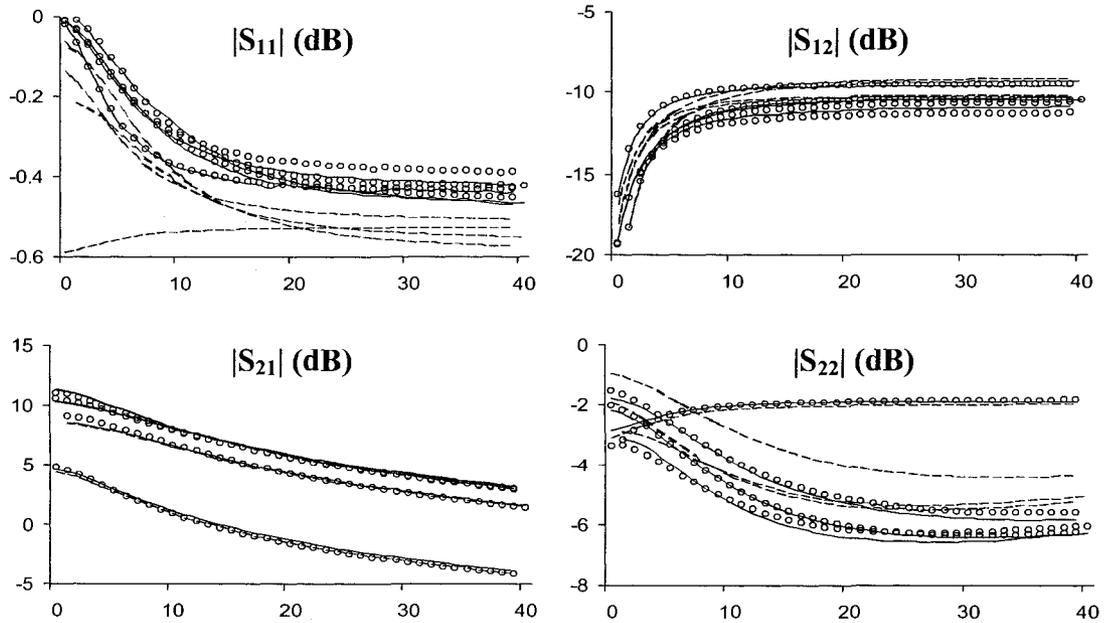
**Figure 3.9:** The dc comparison between the original HEMT data from *MINIMOS*, existing models (without mapping), and the Neuro-SM models in the HEMT example. The gate voltage  $V_g$  for all three models is from  $-0.5V$  to  $-0.1V$ . Existing models used for Neuro-SM are (a) Statz, (b) Curtice, and (c) Chalmers model. Training of Neuro-SM models was done using such dc data and the bias-dependent S-parameter data in Figure 3.10 simultaneously.



(a)



(b)



(c)

**Figure 3.10: S-parameter comparison between the original HEMT data from *MINIMOS*, existing models (without mapping), and the Neuro-SM models in the HEMT example. All plots show S-parameters in dB versus frequency in GHz. Comparison was done at 4 different dc biases at gate voltage ( $-0.4\text{V}$ ,  $-0.2\text{V}$ ) and drain voltage ( $0.2\text{V}$ ,  $2.4\text{V}$ ). Existing models used as coarse models for mapping are (a) the Statz model, (b) the Curtice model, and (c) the Chalmers model.**

Tables 3.9, 3.10, and 3.11 show the sensitivity, model accuracy, and training time for the three implementations of Neuro-SM with 10 hidden neurons, demonstrating the increased efficiency of the proposed analytical Neuro-SM over the circuit-based Neuro-SM of [14]. Training time was recorded for 100 iterations on a Pentium IV 2.8GHz computer. Neuro-SM enables fast and accurate modeling of device physics. To further demonstrate the efficiency of the analytical Neuro-SM, the trained models were

**Table 3.10: Comparison of model accuracy in the HEMT example. The values are average errors between the model and training/testing data. The proposed analytical Neuro-SM can retain the same accuracy as the circuit-based Neuro-SM.**

	Existing model without mapping	Circuit-based Neuro-SM model	Proposed analytical Neuro-SM model
Coarse Model 1 (Statz)	9.82 % / 10.15%	1.34 % / 1.41%	1.34 % / 1.41%
Coarse Model 2 (Curtice)	13.44 % / 13.91%	1.53 % / 1.68%	1.53 % / 1.68%
Coarse Model 3 (Chalmers)	7.53 % / 7.80%	0.96 % / 1.07%	0.96 % / 1.07%

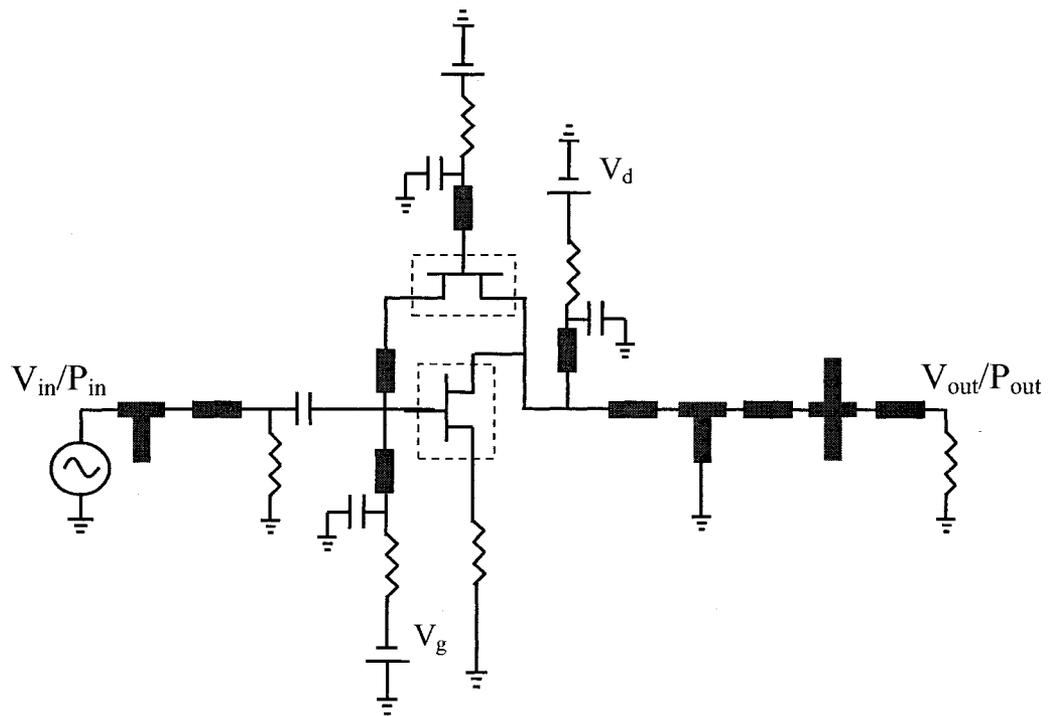
**Table 3.11: Neuro-SM training time comparison between several training techniques for the HEMT example. Training was done with dc and bias-dependent S-parameter data. The proposed technique is the most efficient.**

	Circuit-based Neuro-SM with perturbation	Circuit-based Neuro-SM with adjoint NN sensitivity	Proposed analytical Neuro-SM and sensitivity
Coarse Model 1 (Statz)	220 mins	50 mins	22 mins
Coarse Model 2 (Curtice)	137 mins	35 mins	17 mins
Coarse Model 3 (Chalmers)	350 mins	68 mins	25 mins

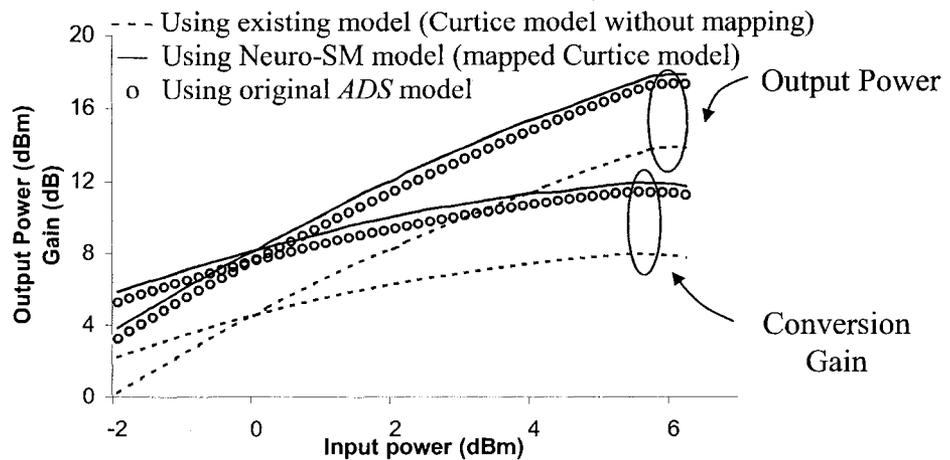
### 3.6.4 Use of Neuro-SM Models in a Frequency Doubler Circuit

This example demonstrates the application of the trained Neuro-SM models in a balanced frequency doubler [110] circuit shown in Figure 3.11. The trained Neuro-SM models for the MESFET device in example of 3.6.2 and the HEMT device in example of 3.6.3 are incorporated into *ADS*, and connected with other *ADS* components to form the overall doubler circuit.

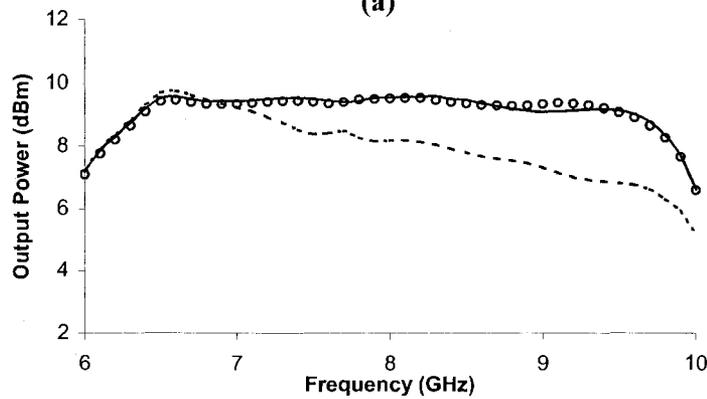
The MESFET Neuro-SM models trained in example of 3.6.2 (mapped Curtice model and mapped Materka model) are first used in the frequency doubler circuit. We performed large-signal harmonic balance simulation of the frequency doubler, and the results including conversion gain, second harmonic output power, and fundamental frequency suppression match well with the original *ADS* solutions, shown in Figures 3.12 and 3.13. This verifies the validity of the large-signal behavior of the proposed Neuro-SM device model. The trained HEMT Neuro-SM models in example of 3.6.3 (i.e., the mapped Statz model, the mapped Curtice model, and the mapped Chalmers model) are then used in the doubler circuit simulation with the results shown in Figure 3.14. In reality, the physics-based device simulator *MINIMOS* cannot be directly combined with other passive/active components in *ADS* for overall circuit design. By the proposed technique, the Neuro-SM models can be first trained to learn the device characteristics from a device physics simulator such as *MINIMOS*. The trained Neuro-SM models can then be conveniently implemented into existing circuit simulators such as *ADS*, thus making the circuit simulation with physics-based device model faster and more convenient.



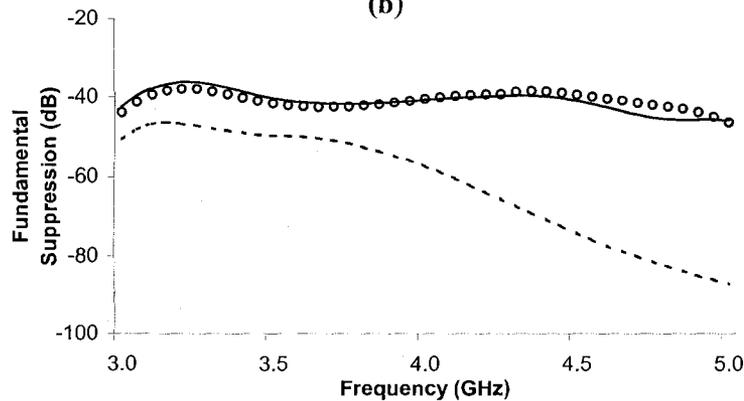
**Figure 3.11: A frequency doubler circuit. Both the MESFET models and the HEMT models developed with the Neuro-SM technique will be used in this circuit.**



(a)

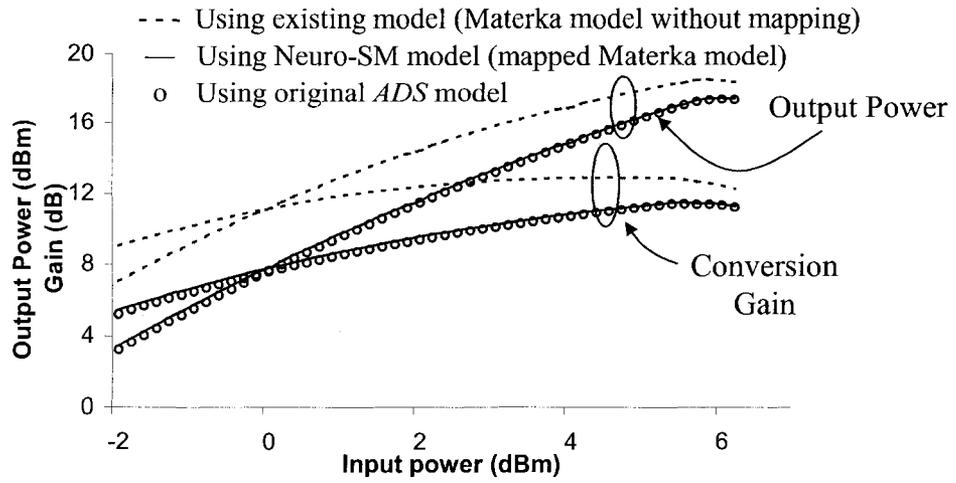


(b)

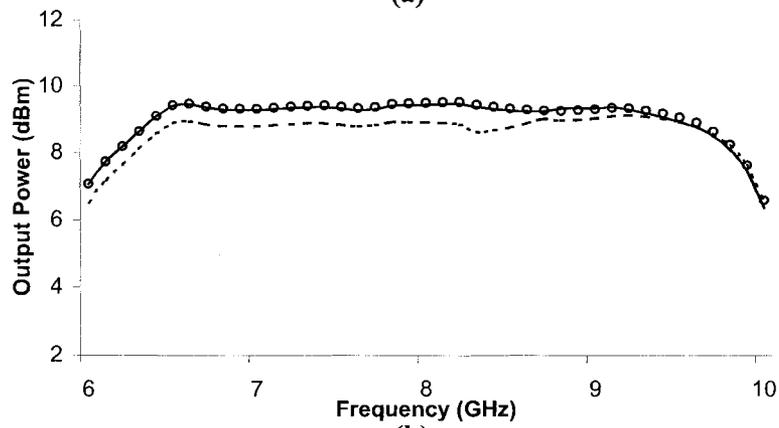


(c)

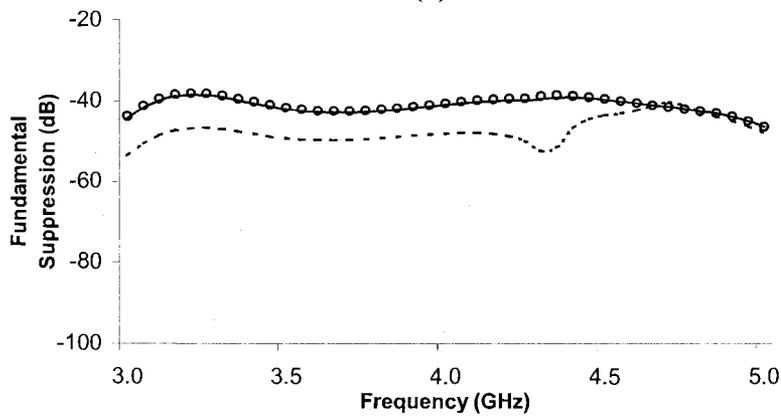
**Figure 3.12: Comparison of the frequency doubler (with the MESFET models) HB solutions between the original ADS model, the coarse model, and the Neuro-SM model. (a) Second harmonic output power and conversion gain versus input power level at input frequency of 4 GHz. (b) Second harmonic output power versus output frequency with input power level of 1dBm. (c) Fundamental signal suppression at input power level of 1 dBm. Before mapping, the existing device model led to an inaccurate doubler solution. The Neuro-SM model improved the solution to be consistent with the original ADS solution.**



(a)



(b)



(c)

Figure 3.13: Comparison of the frequency doubler (with the MESFET models) HB solution between the original ADS model, the coarse model, and the Neuro-SM model. (a), (b), and (c) are similarly defined as in Figure 3.12, except that the coarse model used here for mapping here is the Materka model instead of the Curtice model of Figure 3.12.

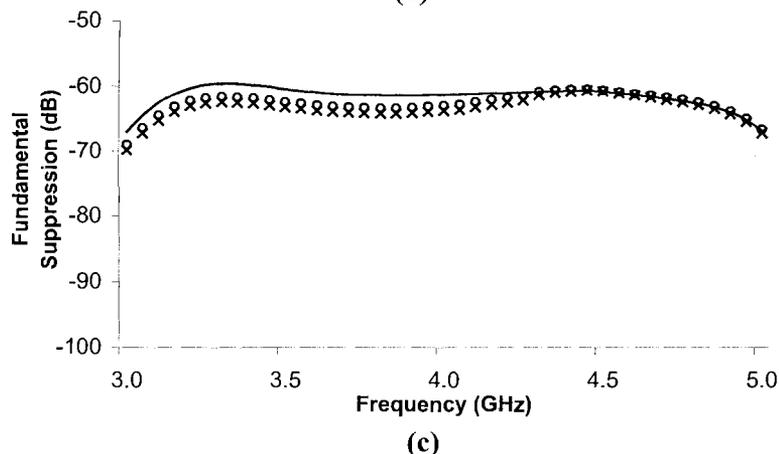
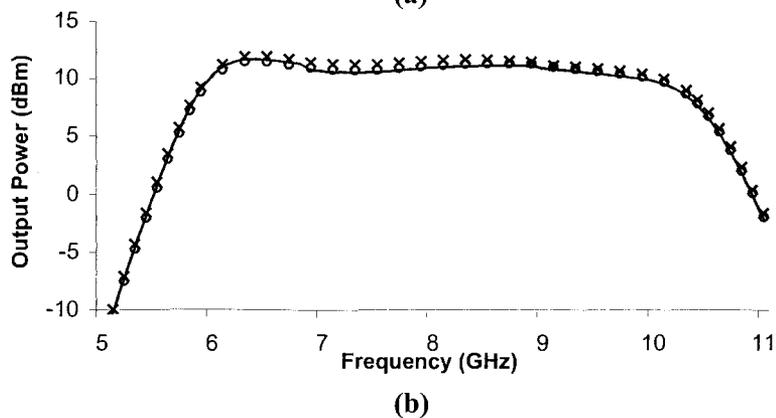
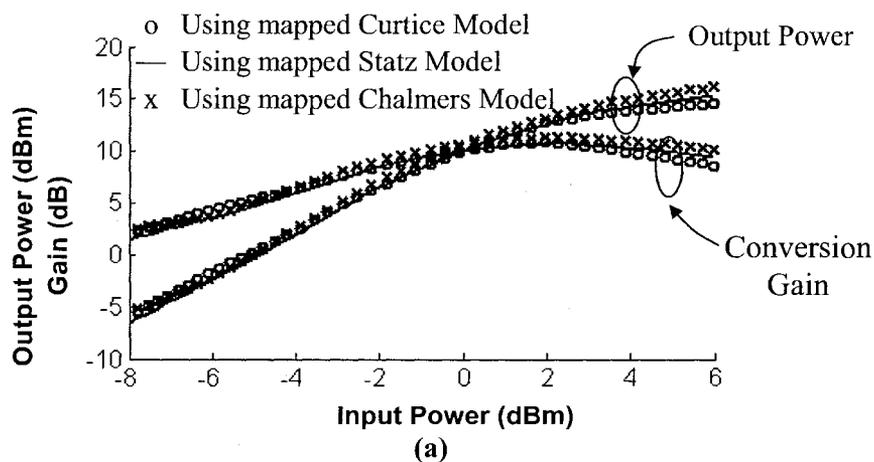


Figure 3.14: Frequency doubler (with the HEMT models) HB solutions using three Neuro-SM models (mapping of the Statz, the Curtice, and the Chalmers models). All the doubler solutions were obtained by *ADS* simulation. (a), (b), and (c) are similarly defined as in Figure 3.12, except that the transistor models used here were trained from the HEMT data generated from *MINIMOS*. Even though the original HEMT represented by the physics-based device simulator *MINIMOS* cannot be directly used in circuit simulators such as *ADS*, the proposed Neuro-SM technique makes it possible to have a HEMT model with device physics behavior in an *ADS* simulation.

### **3.7 Conclusions**

In this chapter, a Neuro-SM technique has been proposed to meet the constant need of new device models due to rapid progress in semiconductor technology. It aims to automatically modify the behavior of existing models to match new device behavior. Neuro-SM models retain the speed of the existing device models while improving the model accuracy. An advanced Neuro-SM formulation has been proposed with analytical mapping representations and exact sensitivity analysis. The proposed technique allows faster model training and evaluation. After being trained, the analytical Neuro-SM model can be incorporated into high-level simulators to increase the speed and accuracy of circuit design. Examples of Neuro-SM modeling by the proposed technique for SiGe HBT, GaAs MESFET, and HEMT devices, and use of the Neuro-SM models in a frequency doubler circuit for harmonic balance simulation have been examined. These examples have demonstrated that the proposed analytical Neuro-SM facilitates efficient model development for nonlinear microwave devices, allowing existing models to exceed their current capabilities. By mapping the existing equivalent circuit models to detailed device physics data, the Neuro-SM can efficiently expand the scope of models in existing circuit simulators to include device physics behavior.

## **Chapter 4: Statistical Space Mapping for Nonlinear Device Modeling: Linear Mapping Method**

### **4.1 Introduction**

This chapter explores the application of ANN and space mapping for large-signal statistical modeling of nonlinear microwave devices. A novel large-signal statistical modeling method is proposed combining one large-signal nominal model and a dynamic space mapping network [17]. The large-signal nominal model is developed using one complete set of large-signal data. It describes the nominal performance of a given device population. A new statistical space mapping concept is introduced to account for the large-signal statistical properties. The mapping contains the statistical parameters estimated by fitting many dc and bias-dependent S-parameter data points of the given device population. In this way, the large-signal nonlinear behavior of the model is mainly represented by the nominal model while the random variations around the nominal model are represented by the space mapping network. With the assumption that the parameter variations of the given device population are usually small percentages of their nominal values, a simple mapping network can be extracted from small-signal data to approximate the large-signal statistical variations. This technique is demonstrated through the modeling of a MESFET device and its use in an amplifier yield analysis.

## 4.2 Proposed Statistical Space Mapped Model

### 4.2.1 Nominal Model

The nominal model is a nonlinear model developed from large-signal measurement data. It can be an extracted equivalent circuit model [78]-[81] or a trained dynamic neural network (DNN) model [49]. This model contains no statistical parameters, therefore requiring the large-signal measurement of only one device. It is used as a coarse representation of the large-signal characteristics for the entire device population.

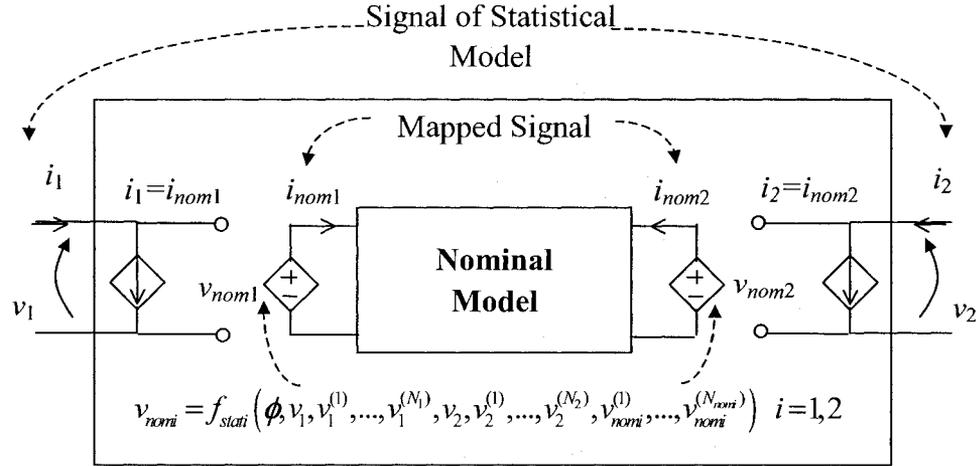
### 4.2.2 Statistical Space Mapping

The statistical space mapping network utilizes the neuro-space mapping concept in [12], [14]-[16] by replacing the mapping neural network with a linear dynamic mapping shown in Figure 4.1. For illustration purpose, a two-port device is examined. Let the terminal voltage and current signals (mapped signals) of the nominal model be defined as  $\mathbf{v}_{nom} = [v_{nom1}, v_{nom2}]^T$  and  $\mathbf{i}_{nom} = [i_{nom1}, i_{nom2}]^T$ , respectively. Similarly define the terminal voltages and currents of the statistical model (original signals) as  $\mathbf{v} = [v_1, v_2]^T$ , and  $\mathbf{i} = [i_1, i_2]^T$ , respectively. A linear dynamic mapping is implemented as the controlling functions of the voltage controlled voltage sources. Current controlled current sources are used to pass  $\mathbf{i}_{nom}$  to  $\mathbf{i}$  in order to make the statistical model consistent with Kirchhoff's Laws, as seen from the external terminals of the overall model.

The mapping equation implemented in the controlled voltage sources is

$$\begin{aligned}
v_{nomi} &= f_{stati} \left( \phi, v_1, v_1^{(1)}, \dots, v_1^{(N_1)}, v_2, v_2^{(1)}, \dots, v_2^{(N_2)}, v_{nomi}^{(1)}, \dots, v_{nomi}^{(N_{nomi})} \right) \\
&= \sum_{k=0}^{N_1} a_{ik} \cdot v_1^{(k)} + \sum_{k=0}^{N_2} b_{ik} \cdot v_2^{(k)} + \sum_{k=1}^{N_{nomi}} c_{ik} \cdot v_{nomi}^{(k)} + d_i, \quad i=1,2
\end{aligned} \tag{4.1}$$

where  $v_i^{(k)}$  and  $v_{nomi}^{(k)}$  ( $i=1,2$ ) are the  $k^{th}$  derivatives of  $v_i$  and  $v_{nomi}$  with respect to time  $t$ , respectively.  $N_i$  and  $N_{nomi}$  are the derivative orders of the voltage signals at port  $i$  ( $i=1,2$ ) of the statistical model and the nominal model, respectively.  $\phi$  is a vector of statistical parameters including  $a_{ik}$  ( $k=1,2,\dots,N_1$ ),  $b_{ik}$  ( $k=1,2,\dots,N_2$ ),  $c_{ik}$  ( $k=1,2,\dots,N_{nomi}$ ), and  $d_i$ , where  $i=1,2$  for all parameters.



**Figure 4.1: Two-port statistical space-mapped model.**

For each device in the statistical population, dc and bias-dependent S-parameter data are measured. Parameter extractions of the  $a$ 's,  $b$ 's,  $c$ 's, and  $d$ 's are performed based on the measurement data for each device. Once  $\phi$  is extracted from all devices in the

population, the means ( $\mu$ ), standard deviations ( $\sigma$ ), and the correlation coefficients ( $\rho$ ) of the statistical parameters  $\phi$  will be calculated.

### 4.2.3 Modeling Procedure

*Step 1.* For a given number ( $N$ ) of devices in the population, use one device to generate a complete set of large-signal data from direct large-signal measurement or device simulation. Develop a large-signal nominal model using this data set.

*Step 2.* Define the mapping function and derivative orders used in the statistical space mapping network.

*Step 3.* Generate dc and S-parameter data for the rest of the devices ( $N-1$ ) in the population. For each set of data, perform parameter extraction to obtain the mapping parameters  $\phi$ .

*Step 4.* From the  $N-1$  sets of extracted parameters, calculate  $\mu$ ,  $\sigma$ , and  $\rho$ . These values represent the statistical properties of the mapping parameters  $\phi$ . The statistical space mapping is formed by applying them into the mapping network.

*Step 5.* Combine the nominal model and the statistical space mapping network to form the large-signal statistical model as shown in Figure 4.1.

## 4.3 Application Examples

### 4.3.1 Large-Signal Statistical Model of a MESFET Device

To demonstrate the proposed technique, we examine the statistical behavior of a population of 50 devices represented by an internal MESFET [78] in ADS [106]. The ADS device parameters are perturbed around given mean values by specified standard

deviations. The nominal model in this example is the MESFET model whose parameters are exactly the mean values. The statistical parameters in the space mapping network are extracted from dc and bias-dependent S-parameters of each device in the population for a reasonable extraction accuracy of 1% error. Each set of dc and S-parameter data is generated at 150 bias points and 20 frequencies. The derivative orders, i.e.,  $N_i$  and  $N_{nomi}$  ( $i=1,2$ ), used in this example are all equal to one. After parameter extraction,  $\mu$ ,  $\sigma$ , and  $\rho$  of the parameters  $\phi$  are calculated as shown in Tables 4.1 and 4.2.

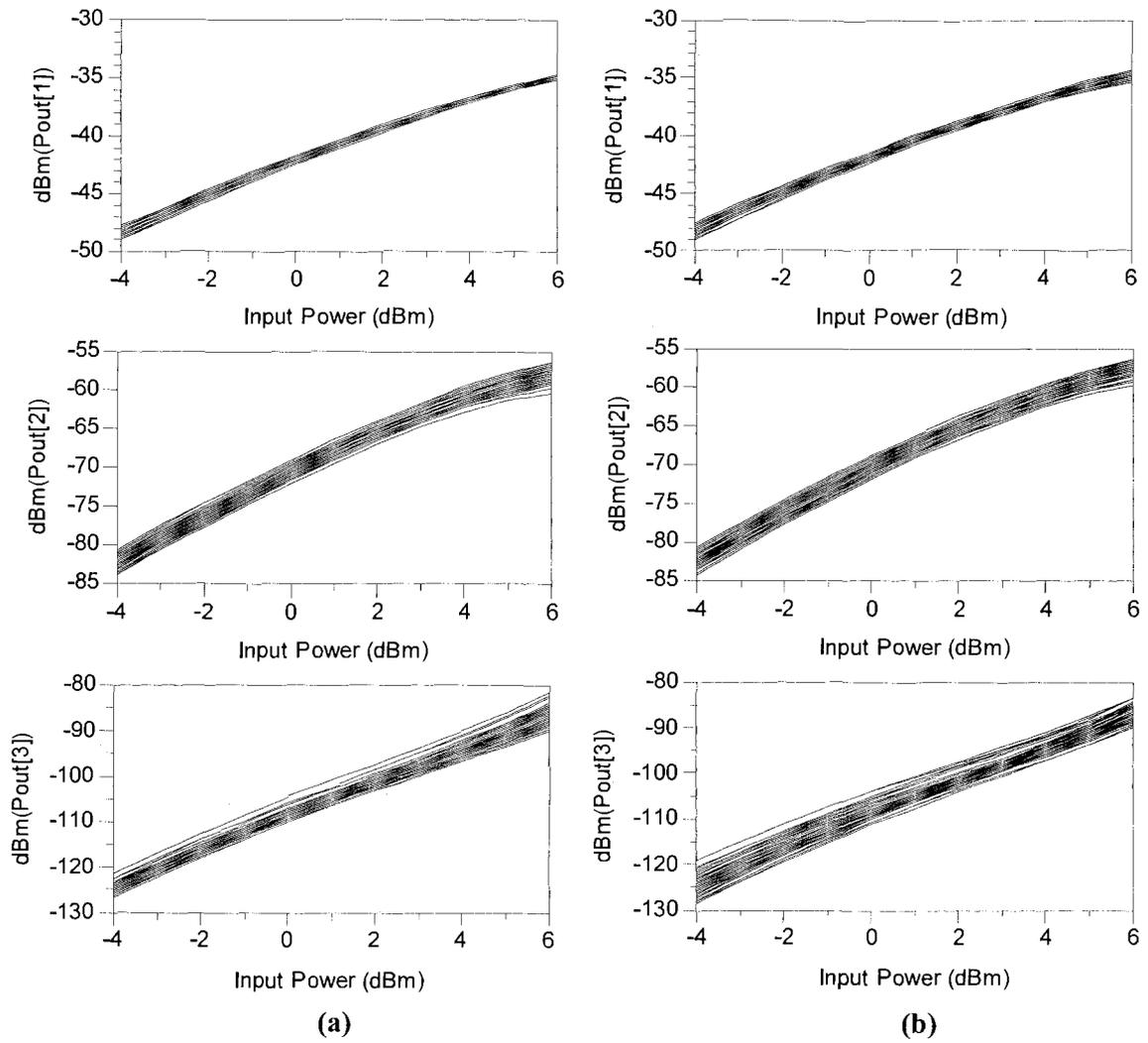
**Table 4.1: Means and standard deviations of the statistical space mapping parameters.**

Parameter ( $\phi$ )	Mean ( $\mu$ )	Standard Deviation ( $\sigma$ )
$a_{10}$	1.004	1.943e-2
$a_{11}$	4.940e-4	5.106e-3
$b_{10}$	-2.968e-3	4.921e-3
$c_{11}$	-4.912e-4	5.095e-3
$d_1$	4.866e-4	5.657e-2
$a_{20}$	-2.178e-2	1.349e-2
$a_{21}$	2.135e-3	6.054e-3
$b_{20}$	1.059	7.872e-2
$b_{21}$	-4.315e-2	8.455e-2
$c_{21}$	3.838e-2	7.181e-2
$d_2$	-4.912e-4	3.297e-2

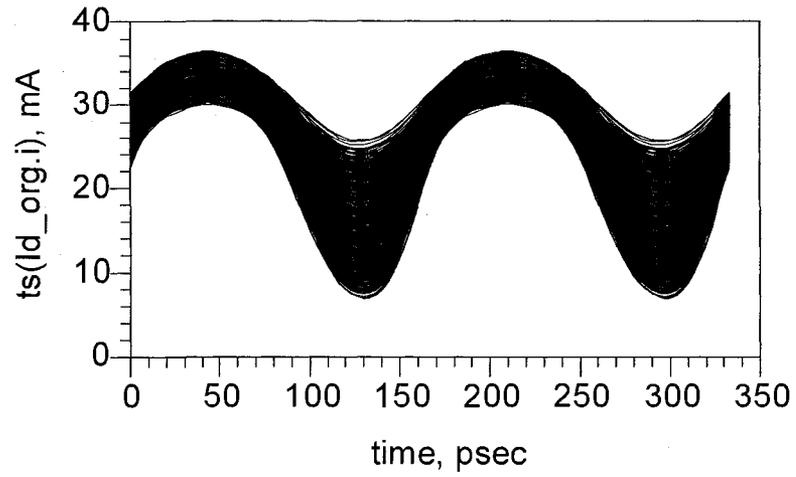
**Table 4.2: Correlation coefficients of the statistical space mapping parameters.**

Correlation Coefficients ( $\rho$ )											
$\phi$	$a_{10}$	$a_{11}$	$b_{10}$	$c_{11}$	$d_1$	$a_{20}$	$a_{21}$	$b_{20}$	$b_{21}$	$c_{21}$	$d_2$
$a_{10}$	1.00										
$a_{11}$	-0.22	1.00									
$b_{10}$	-0.39	-0.29	1.00								
$c_{11}$	-0.30	-0.86	0.48	1.00							
$d_1$	-0.20	0.21	0.16	-0.09	1.00						
$a_{20}$	-0.60	0.12	0.61	0.18	-0.07	1.00					
$a_{21}$	-0.18	0.26	-0.43	-0.14	-0.41	0.09	1.00				
$b_{20}$	0.95	-0.21	-0.44	-0.28	-0.28	-0.67	-0.11	1.00			
$b_{21}$	-0.39	0.02	-0.05	0.16	-0.01	0.23	-0.01	-0.36	1.00		
$c_{21}$	-0.10	0.09	0.28	-0.02	0.15	0.06	0.08	-0.13	-0.87	1.00	
$d_2$	-0.39	0.03	0.38	0.17	0.14	0.71	0.02	-0.62	-0.02	0.25	1.00

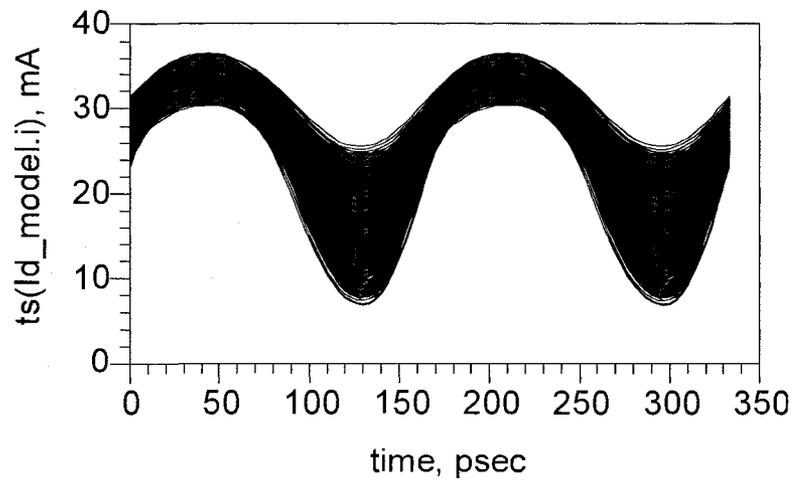
To test the result, the overall statistical model including the nominal model and the statistical space mapping network is then used for large-signal Monte-Carlo analysis with 100 devices. The same analysis is also done to the original MESFET device. The comparisons of the output power and the output current for all the 100 devices are given in Figures 4.2 and 4.3, showing that the proposed model can catch the large-signal statistical properties of the device.



**Figure 4.2: Example of output power (fundamental to third harmonics) v.s. input power of Monte-Carlo simulations with 100 devices using (a) the original ADS MESFET and (b) the proposed statistical space-mapped model.**



(a)



(b)

**Figure 4.3: Example of output current of Monte-Carlo simulations with 100 devices using (a) the original ADS MESFET and (b) the proposed statistical space-mapped model.**

### 4.3.2 Use of Statistical Space-Mapped Model in Amplifier Simulation

To further demonstrate the capability of this technique, we use the statistical space-mapped model from Example *A* in a three-stage amplifier simulation as shown in Figure 4.4. We perform 1000 Monte-Carlo analyses to two amplifier circuits: one uses the original MESFET device in ADS; another uses our proposed statistical model. The yield results are 73.6% and 68.9%, respectively. Figure 4.5 shows a comparison of the gain of the amplifier circuits. This shows that the proposed statistical space-mapped model can be used for statistical design of high-level circuits.

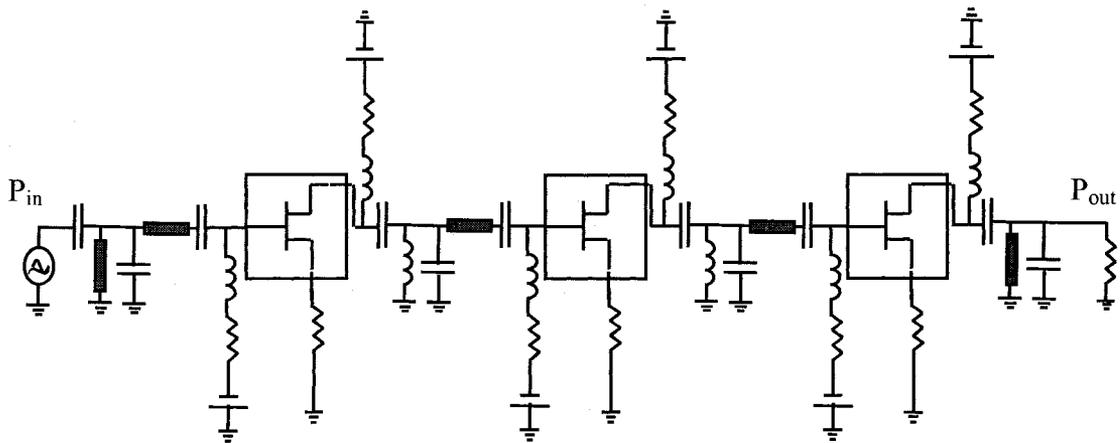
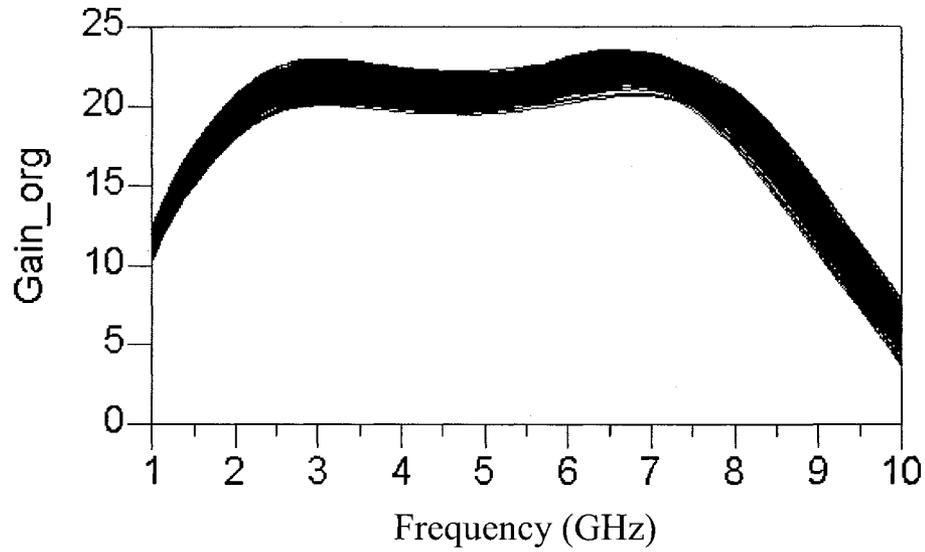
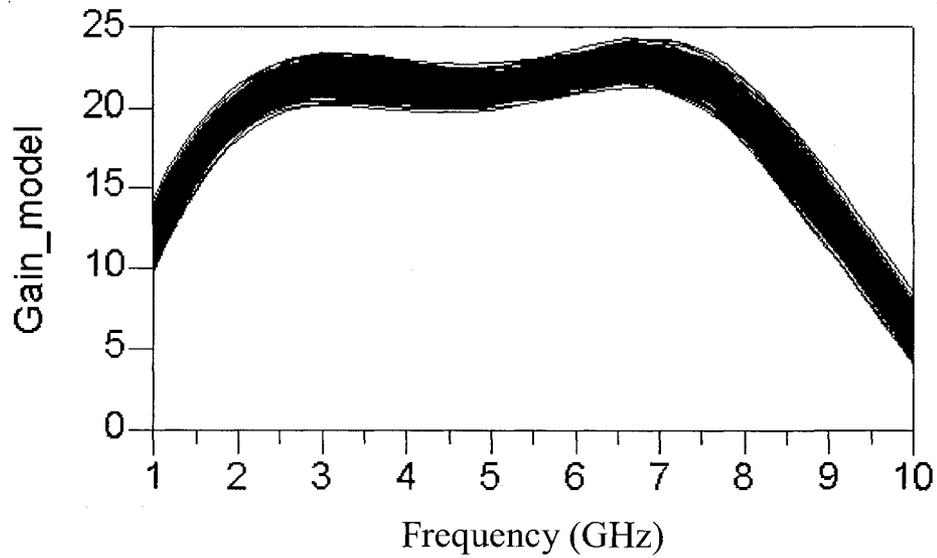


Figure 4.4: Three-stage amplifier circuit.



(a)



(b)

**Figure 4.5: Gain comparison of 1000 amplifier circuits using (a) the original ADS MESFET and (b) the proposed statistical space-mapped model. The distribution of the amplifier responses using our proposed statistical space-mapped model matches that of the original ADS results well, confirming our proposed method.**

## **4.4 Conclusions**

A new large-signal statistical modeling technique has been presented. The proposed statistical space-mapped model combines a large-signal nominal model with a dynamic mapping network which characterizes the statistical variations around the nominal. The nominal model is extracted from large-signal data while the space mapping network containing the statistical parameters is developed from dc and bias-dependent S-parameter data. This technique allows large-signal statistical model development without large-signal data generation of a massive number of devices, thus reducing the modeling cost.

## **Chapter 5: Statistical Neuro-Space Mapping (Neuro-SM) Technique for Large-Signal Statistical Modeling of Nonlinear Devices**

### **5.1 Introduction**

In this chapter, further progress in large-signal statistical modeling is presented [18], as an expansion over the linear statistical mapping method of Chapter 4. In reality, the statistical variations among the device samples in a given population could be large as well as small [86]-[92]. The use of linear space mapping in a large variation case will result in a large mapping network with high dynamic orders and a large number of mapping coefficients as statistical parameters. This may lead to non-unique solutions in the statistical parameter extraction of some random devices whose behavior is close to that of the nominal device, causing unreliable distributions and uncertainty in the statistical behavior of the model. In the present chapter, we overcome this problem with a new technique, called statistical neuro-space mapping (statistical Neuro-SM) [18]. The proposed technique effectively models large statistical variations by expanding the linear mapping of Chapter 4 to nonlinear mapping, while preserving the use of the large-signal nominal model to minimize the cost of large-signal data generation. Since the analytical formula of the nonlinear mapping is usually unknown, neural networks are used to achieve the mapping. A re-formulated mapping, different from the linear mapping of

Chapter 4, is proposed, allowing the statistical parameters to be defined separately from the coefficients in the mapping function. In this way, the increased complexity of the mapping, required for large statistical variations, can be achieved without having to increase the dimension of the statistical parameters. A new training algorithm is developed to perform simultaneous statistical parameter extraction and neural network training based on dc and bias-dependent S-parameter data of all device samples in the given population. The proposed technique aims at producing better accuracy even when the size of the statistical variations grows, a feature not possible in the linear mapping of Chapter 4.

## **5.2 Proposed Statistical Neuro-SM Technique**

### **5.2.1 Proposed Statistical Neuro-SM Formulation**

To characterize accurately the statistical behavior of nonlinear microwave devices which have large statistical variations, we propose to use a nonlinear mapping function in the statistical space mapping such that the behavior of a randomly selected device can be represented by a mapped version of the nominal device model. Since this mapping function is usually unknown and precise analytical mapping equations may not be available, the neural network becomes a logical choice. However, the formulation of the neural network mapping is not a trivial matter. A straightforward expansion from linear to nonlinear mapping is equivalent to using the “non-statistical” mapping method of Chapter 3 for statistical modeling, where the coarse model is replaced by the nominal model and the fine model is replaced by random device samples. However, this

straightforward approach will lead to similar problems as in linear mapping because the weighting parameters of the mapping neural network have to vary between different device samples. This forces the weighting parameters to become statistical parameters. Since, in general, the solutions of the neural network weighting parameters are almost always non-unique, the statistical distribution estimated from the weighting parameters will be very unreliable. Therefore, the mapping technique in Chapter 3, although good for single device modeling, is not appropriate to be used directly for statistical modeling.

In this chapter, a new formulation dedicated to the special needs of statistical modeling is proposed. We call this new technique “statistical neuro-space mapping”. Figure 5.1 shows the structure of the proposed statistical Neuro-SM model. A large-signal nominal model is used to represent the average large-signal behavior of the given population of devices. Suppose the input and output signals of the device model are represented by  $x$  and  $y$ , respectively. The input signals of a random device sample are firstly mapped to those of the nominal model by an input mapping neural network [14]-[16], and the output signals from the nominal model are further refined by an output mapping neural network to produce the final model outputs using the concept of prior knowledge input [11]. We name the input mapping and the output mapping as  $x$ -mapping and  $y$ -mapping, respectively, for convenience of description in the rest of the chapter. All parameters in the nominal model and the mapping neural networks are defined to be deterministic. In this way, different device samples in the given population will share the same values of the weighting parameters and thus the nonlinear mapping function. However, it is necessary to alter the nonlinear mapping function to capture accurately the

random variations between different device samples. To achieve this, we introduce a new set of input neurons to both the  $x$ - and  $y$ -mapping neural networks. The new input neurons act as control variables to diversify the deterministic nonlinear mapping for various device samples. With different values for these new input neurons, the  $x$ -mapping (and  $y$ -mapping) will map differently for different device samples, allowing the overall model to reach the behavior of all devices in the population. Consequently, the statistical variables in our model are defined to be these new input neurons in the  $x$ - and  $y$ -mapping neural networks. Because the number of input neurons for a neural network can be much less than the number of internal weighting parameters, the dimension of the statistical variables in our proposed model will be more compact, a feature not possible in previous linear mapping of Chapter 4.

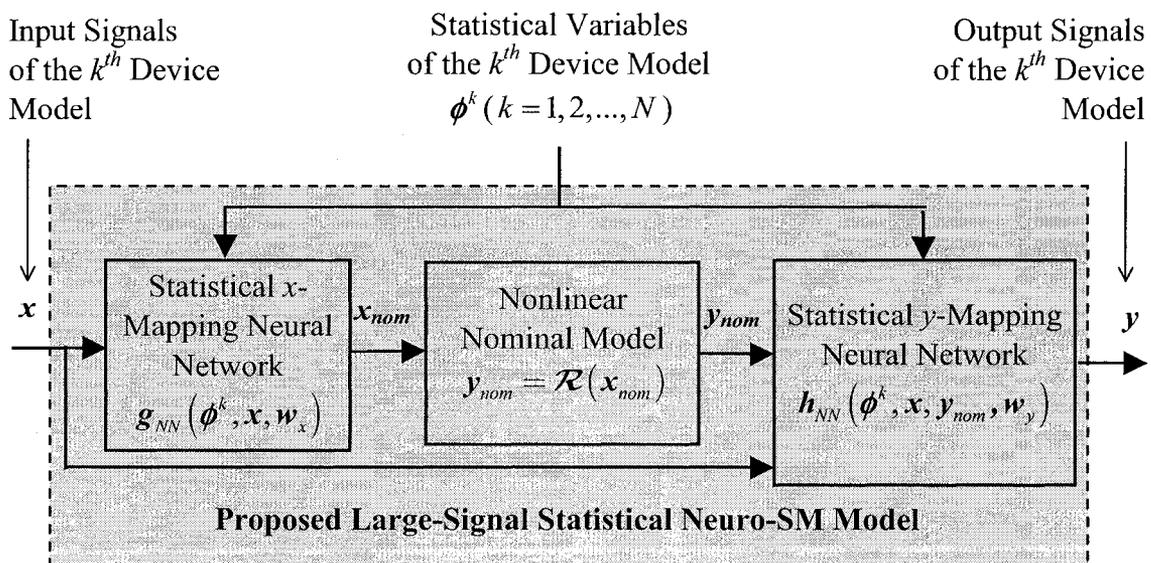


Figure 5.1: Illustration of the proposed large-signal statistical Neuro-SM model.

Following the same notation described in Section 4.2, let  $N$  represent the number of device samples in the device population. Symbols  $\mathbf{x}$  and  $\mathbf{y}$  represent the input and output signals of the overall model of any device in the population. Let  $\mathbf{x}_{nom}$  and  $\mathbf{y}_{nom}$  be the input and the output signals of the nominal model, respectively. Define  $\phi$  as a vector containing the statistical variables, which are implemented as input neurons to control the mapping functions of the  $x$ - and  $y$ -mapping neural networks. Let  $\phi^k$  ( $k=1,2,\dots,N$ ) represent the  $k^{th}$  random outcome of the statistical variables  $\phi$  corresponding to the  $k^{th}$  device. The  $x$ - and  $y$ -mappings are formulated as

$$\mathbf{x}_{nom} = \mathbf{g}_{NN}(\phi^k, \mathbf{x}, \mathbf{w}_x) \quad (5.1)$$

$$\mathbf{y} = \mathbf{h}_{NN}(\phi^k, \mathbf{x}, \mathbf{y}_{nom}, \mathbf{w}_y) \quad (5.2)$$

where  $\mathbf{g}_{NN}(\cdot)$  and  $\mathbf{h}_{NN}(\cdot)$  represent the  $x$ - and  $y$ - mapping neural networks. The input neurons for the  $x$ -mapping neural network include the input signals  $\mathbf{x}$  and the statistical variables  $\phi$ . The input neurons for the  $y$ -mapping neural network include the input signals  $\mathbf{x}$ , the nominal model output signals  $\mathbf{y}_{nom}$ , and the statistical variables  $\phi$ . The deterministic parameters  $\mathbf{w}_x$  and  $\mathbf{w}_y$  are the weighting parameters of the  $x$ - and  $y$ -mapping neural networks, respectively. The responses  $\mathbf{y}_{nom}$  are evaluated by the nominal model at the mapped nominal inputs  $\mathbf{x}_{nom}$  as

$$\mathbf{y}_{nom} = \mathcal{R}(\mathbf{x}_{nom}) \quad (5.3)$$

where the function  $\mathcal{R}(\cdot)$  represents the dynamic input-output relationship of the large-signal nominal model. The dynamic responses of the proposed model are obtained by solving the nominal model together with the nonlinear mapping equations. In other words, given input signals  $\mathbf{x}$ , the output signals  $\mathbf{y}$  can be evaluated from our model this way: map  $\mathbf{x}$  to  $\mathbf{x}_{nom}$  using (5.1), evaluate the nominal model to obtain  $\mathbf{y}_{nom}$  from  $\mathbf{x}_{nom}$ , and finally map  $\mathbf{y}_{nom}$  to  $\mathbf{y}$  using (5.2).

By adding the new statistical variables as input neurons of the mapping neural networks, additional degrees of freedom are achieved to alter the nonlinear mapping function. For a given population of devices, the behavior of different device samples can be individually mapped from that of the nominal model using different values of the statistical variables. To obtain a precise mapping between the nominal model and all device samples in the given population, two questions need to be answered: (i) how to determine the nonlinear mapping, and (ii) how to determine the values of the statistical variables to control the nonlinear mapping for each device sample. To solve these problems, Subsection 5.2.3 proposes a novel training technique, where the nonlinear mapping is determined by allowing the statistical variables (new input neurons) to be optimizable and forcing the  $x$ - and  $y$ -mapping neural networks to simultaneously learn data from all device samples.

### 5.2.2 Proposed Statistical Neuro-SM for FET Modeling

In this subsection, we formulate the proposed technique for statistical modeling of 2-port field effect transistor (FET) devices. For other type of transistors such as heterojunction bipolar transistors (HBT), similar formulations can be deduced with the corresponding input-output relationship.

Let the gate and drain terminal voltage and current signals of the FET be  $\mathbf{v} = [v_g \ v_d]^T$  and  $\mathbf{i} = [i_g \ i_d]^T$ , respectively. Let the terminal voltage and current signals of the nominal model be  $\mathbf{v}_{nom} = [v_{gnom} \ v_{dnom}]^T$  and  $\mathbf{i}_{nom} = [i_{gnom} \ i_{dnom}]^T$ , respectively. Let  $\mathbf{x} = \mathbf{v}$ ,  $\mathbf{y} = \mathbf{i}$ ,  $\mathbf{x}_{nom} = \mathbf{v}_{nom}$ , and  $\mathbf{y}_{nom} = \mathbf{i}_{nom}$ . The neural network mapping equations of (5.1) and (5.2) are implemented as the controlling functions in controlled voltage sources for the  $x$ -mapping and controlled current sources for the  $y$ -mapping.

Before the statistical Neuro-SM model can accurately represent the statistical behavior of a given population of FET devices, it needs to be trained by the input-output data of the device samples. In the proposed technique, the model for a random device is not determined from scratch in a stand-alone fashion. It is determined by a modification (i.e., mapping) of the nominal model, which has already been accurately extracted from dc, small-, and large-signal data. Because of this special formulation, training of the proposed neural networks (i.e., the mapping) does not have to rely on full sets of large-signal data. We use only a reduced set of data, in the form of dc and bias-dependent S-parameter data of each device sample in the given population. This is much more cost-effective compared with the requirement of large-signal data for every device in the

entire population. Here we establish the connection between the statistical Neuro-SM model and the dc and bias-dependent S-parameter data needed for training of the proposed model.

The dc response of the proposed model at  $\phi = \phi^k$  is mapped from that of the nominal model as

$$\mathbf{I}(\phi^k, \mathbf{V}, \mathbf{w}_x, \mathbf{w}_y) = \mathbf{h}_{NN} \left( \phi^k, \mathbf{V}, \mathbf{I}_{nom} \Big|_{\mathbf{V}_{nom} = \mathbf{g}_{NN}(\phi^k, \mathbf{V}, \mathbf{w}_x)}, \mathbf{w}_y \right) \quad (5.4)$$

where  $\phi^k$  contains the statistical variables corresponding to the  $k^{\text{th}}$  device sample.  $\mathbf{V}$  contains the dc voltage signals of the  $k^{\text{th}}$  device sample, and  $\mathbf{I}_{nom}$  contains the dc current signals of the nominal model evaluated at  $\mathbf{V}_{nom}$ , which is mapped from  $\mathbf{V}$  by the  $x$ -mapping neural network.

The small-signal S-parameters of the proposed model at  $\phi = \phi^k$  are achieved by transforming its Y-parameters, which are mapped from the Y-parameters of the nominal model as

$$\begin{aligned} \mathbf{Y}(\phi^k, \mathbf{V}, \omega, \mathbf{w}_x, \mathbf{w}_y) = & \left( \frac{\partial \mathbf{h}_{NN}^T(\phi^k, \mathbf{v}, \mathbf{i}_{nom}, \mathbf{w}_y)}{\partial \mathbf{i}_{nom}} \Big|_{\mathbf{v}=\mathbf{V}} \right)^T \times \mathbf{Y}_{nom}(\omega) \Big|_{\mathbf{V}_{nom}} \times \\ & \left( \frac{\partial \mathbf{g}_{NN}^T(\phi^k, \mathbf{v}, \mathbf{w}_x)}{\partial \mathbf{v}} \Big|_{\mathbf{v}=\mathbf{V}} \right)^T + \left( \frac{\partial \mathbf{h}_{NN}^T(\phi^k, \mathbf{v}, \mathbf{i}_{nom}, \mathbf{w}_y)}{\partial \mathbf{v}} \Big|_{\mathbf{v}=\mathbf{V}} \right)^T \end{aligned} \quad (5.5)$$

where  $V$  contains the bias voltages of the  $k^{\text{th}}$  device sample,  $Y_{nom}(\omega)$  contains the Y-parameters of the nominal model at frequency  $\omega$ , and  $V_{nom} = \mathbf{g}_{NN}(\phi^k, V, \mathbf{w}_x)$  contains the mapped bias of the nominal model by the  $x$ -mapping. The first-order derivatives of the  $x$ - and  $y$ -mapping neural networks required in (5.5) are obtained using the adjoint neural network method [7].

### 5.2.3 Proposed Training of the Statistical Neuro-SM Model

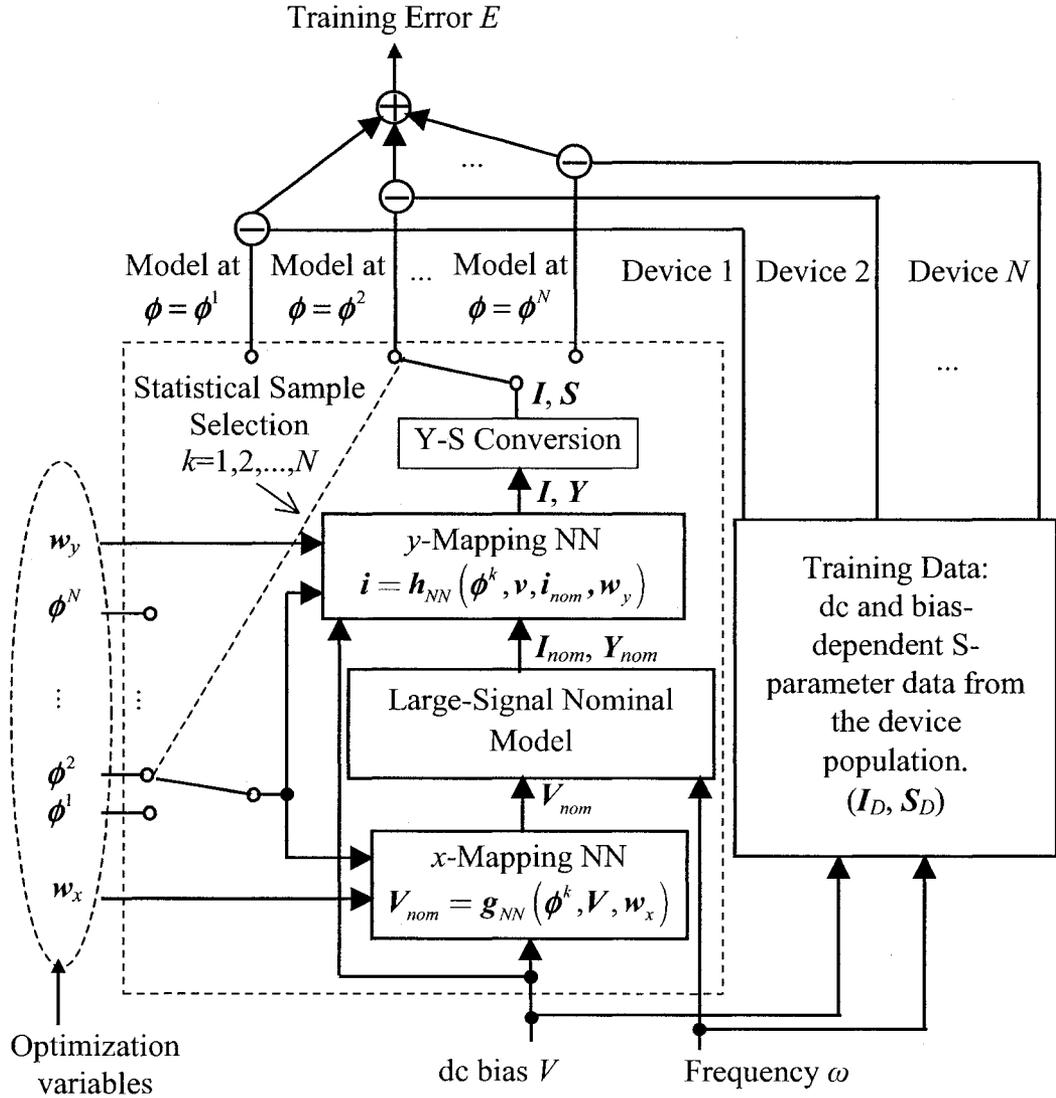
To reproduce accurately the statistical behavior of a given population of devices, the nonlinear mapping functions, i.e.,  $\mathbf{g}_{NN}(\cdot)$  and  $\mathbf{h}_{NN}(\cdot)$ , need to be determined. In reality, the analytical formula of the nonlinear mapping is not available, and how the mapping is controlled by the statistical variables is unknown. The known information are the effects of such controlled mapping, i.e., the statistical variations in the dc and S-parameter data between different device samples in the given population.

The training of the proposed model is to solve  $\mathbf{w}_x$ ,  $\mathbf{w}_y$ , and  $\phi^k$  ( $k=1,2,\dots,N$ ) to find the mapping neural networks  $\mathbf{g}_{NN}(\cdot)$  and  $\mathbf{h}_{NN}(\cdot)$ , such that the mapped model is able to represent the statistical behavior of the device samples in the given population. Note that statistical variables  $\phi^k$  are used to control the nonlinear mapping between the large-signal nominal model and the  $k^{\text{th}}$  device sample, while the weighting parameters  $\mathbf{w}_x$  and  $\mathbf{w}_y$  are common variables in the nonlinear mapping functions shared by all device samples. The existence of common variables means that the conventional device-by-device parameter

extraction is not applicable for training the proposed model. A completely new training technique is proposed here to perform a simultaneous search of the nonlinear mapping functions (i.e., search of the weighting parameters  $\mathbf{w}_x$  and  $\mathbf{w}_y$ ) and their controlling statistical variables  $\phi^k$  ( $k=1,2,\dots,N$ ) using dc and S-parameter data from all device samples in the given population. The training error is formulated as the total difference between model responses and the dc and S-parameter data of all device samples as

$$E(\phi^1, \phi^2, \dots, \phi^N, \mathbf{w}_x, \mathbf{w}_y) = \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^{N_{bias}} \left\| \mathbf{A} \cdot \left( \mathbf{I}(\phi^k, V_l, \mathbf{w}_x, \mathbf{w}_y) - \mathbf{I}_{Dl}^k \right) \right\|^2 + \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^{N_{bias}} \sum_{j=1}^{N_{freq}} \left\| \mathbf{B} \cdot \left( \mathbf{S}(\phi^k, V_l, \omega_j, \mathbf{w}_x, \mathbf{w}_y) - \mathbf{S}_{Dlj}^k \right) \right\|^2 \quad (5.6)$$

where  $\mathbf{I}(\cdot)$  and  $\mathbf{I}_D$  are the dc currents of the model and the device, respectively;  $\mathbf{S}(\cdot)$  and  $\mathbf{S}_D$  are the S-parameters of the model and the device, respectively. The dc and S-parameter responses of the proposed model, i.e.,  $\mathbf{I}(\cdot)$  and  $\mathbf{S}(\cdot)$ , are defined through (5.4) and (5.5), respectively.  $\mathbf{A}$  and  $\mathbf{B}$  are diagonal matrices containing the scaling factors defined as the inverse of the minimum-to-maximum range of the corresponding  $\mathbf{I}_D$  data and  $\mathbf{S}_D$  data, respectively. The superscript  $k$  ( $k=1,2,\dots,N$ ) denotes the index of the random device sample. The subscripts  $l$  ( $l=1,2,\dots,N_{bias}$ ) and  $j$  ( $j=1,2,\dots,N_{freq}$ ) denote the indices of bias and frequency in the dc and S-parameter data of each device sample, respectively.  $N_{bias}$  and  $N_{freq}$  are the total numbers of biases and frequencies, respectively. The calculation of the training error is further illustrated in Figure 5.2.



**Figure 5.2:** Calculation of the training error of the proposed statistical Neuro-SM model. Note that for different device samples, the proposed model uses the same  $x$ - and  $y$ -mapping neural networks but different values of the statistical variables to alter the nonlinear mapping.

The objective of the proposed training is to optimize  $w_x$ ,  $w_y$ , and  $\phi^k$  ( $k=1,2,\dots,N$ ) to minimize the total error  $E$  of (5.6) as

$$\min_{\phi^1, \phi^2, \dots, \phi^N, w_x, w_y} E(\phi^1, \phi^2, \dots, \phi^N, w_x, w_y) \quad (5.7)$$

This comprehensive training process combines the mapping neural network training and the extraction of statistical variables  $\phi^k$  ( $k=1,2,\dots,N$ ) into one gradient-based optimization for efficient model development. The gradient information of (5.6) required for efficient training includes two parts: (i) the derivatives of  $E$  w.r.t. neural network weighting parameters  $w_x$  and  $w_y$ , and (ii) the derivatives of  $E$  w.r.t. statistical variables  $\phi^k$ . The first part is used for updating the weighting parameters of the  $x$ - and  $y$ -mapping neural networks. The second part contributes to the extraction of the statistical variables. Such required derivatives can be analytically achieved through the adjoint neural network sensitivity analysis [7].

After training, the  $k^{th}$  device sample can be represented by the proposed model with  $\phi = \phi^k$  ( $k=1,2,\dots,N$ ). The distribution of the statistical variables can be estimated from the extracted  $\phi^k$ 's. The proposed model with such distributions is able to represent the statistical behavior of a given device population.

### 5.2.4 Normality Mapping

In general, the distribution of the extracted statistical variables  $\phi$  is arbitrary and may not follow standard distributions such as Gaussian distribution. Here we develop a normality mapping to relate the non-Gaussian variables  $\phi$  with Gaussian variables represented by  $\bar{\phi}$ . This mapping should be nonlinear. However, the analytical function of such a nonlinear mapping is usually unknown. Approximate/empirical methods, such as power transformation [94], are one way to solve this problem. Here we use a neural network to achieve this unknown function by learning the exact mapping relationship between the extracted statistical variables  $\phi$  and the Gaussian variables  $\bar{\phi}$ .

A normality mapping neural network is formulated as

$$\phi = z_{NN}(\bar{\phi}, \mathbf{w}_m) \quad (5.8)$$

where  $z_{NN}$  denotes the neural network function, and  $\mathbf{w}_m$  are the weighting parameters of the neural network. The input neurons are represented by  $\bar{\phi}$  and the output neurons are represented by  $\phi$ . The normality mapping neural network is trained by  $N$  samples of data pairs of  $(\bar{\phi}^k, \phi^k)$ ,  $k = 1, 2, \dots, N$ , where  $\phi^k$  is the  $k^{\text{th}}$  sample of the extracted  $\phi$ , and  $\bar{\phi}^k$  is the  $k^{\text{th}}$  sample of  $\bar{\phi}$  from an ideal Gaussian distribution with zero mean values and unit standard deviations. The  $N$  samples of  $\bar{\phi}$  for neural network training are ordered following the sequence of the  $\phi$  samples in such a way that if the largest (or 2<sup>nd</sup> largest,

3<sup>rd</sup> largest,...) value of  $\phi$  occurs at sample number  $k$ , then the largest (or 2<sup>nd</sup> largest, 3<sup>rd</sup> largest,...) value of the  $\bar{\phi}$  samples will be assigned to  $\bar{\phi}^k$ .

The normality mapping transforms the non-Gaussian distribution of the extracted  $\phi$  into an ideal Gaussian distribution of  $\bar{\phi}$  in each dimension of the statistical variables. The correlation coefficients among the different dimensions are computed from the ordered samples of  $\bar{\phi}$  as

$$\rho(\bar{\phi}) = \frac{1}{N} \sum_{k=1}^N \left( \bar{\phi}^k \cdot (\bar{\phi}^k)^T \right) \quad (5.9)$$

The final statistical variables for the overall statistical Neuro-SM model are the Gaussian variables  $\bar{\phi}$ , whose statistical distribution is characterized by the mean values  $\mu(\bar{\phi}) = \mathbf{0}$ , standard deviations  $\sigma(\bar{\phi}) = \mathbf{1}$ , and the computed correlation coefficients  $\rho(\bar{\phi})$  from (5.9).

### 5.2.5 Discussion

Different from the linear statistical mapping in Chapter 4, the proposed statistical Neuro-SM does not directly use the mapping coefficients, i.e., the neural network weighting parameters, as statistical parameters. Instead, it uses separately defined statistical variables to control the mapping functions. In this way, the number of statistical variables is greatly reduced. The dimension of the statistical variables in our formulation means the

number of factors controlling the statistical variations of the overall model. The proposed training process automatically searches for fundamental statistical factors that allow the device to vary to reach all samples in a given population of random devices. The best dimension of the statistical variables is the smallest one that can produce a good match of statistical behavior between the model and the device population. The proposed technique allows the dimension of the statistical variables to be adjusted easily by using more or less input neurons in the  $x$ - and  $y$ -mapping neural networks. This provides flexibility to achieve desired accuracy in statistical modeling.

The  $x$ - and  $y$ -mappings work together to map the nominal device behavior to that of the statistical devices effectively. The role of the  $x$ -mapping, based on the space mapping concept, is to modify the inputs to the nominal model such that the outputs of the nominal model become as close as possible to that of the statistical devices in the population. The role of the  $y$ -mapping is to provide further refinement to the statistical model such that the outputs of the overall statistical Neuro-SM model can match that of the statistical devices without being limited by the output ranges of nominal model.

The aim of the proposed method is to address the difficulties of large-signal statistical modeling, by reducing the need for large-signal data to that for the nominal model only. The starting point for our modeling algorithm is a nominal device model, and dc/small-signal data for the device population. In this sense, the explicit assumption of our proposed method is the availability of an accurate nominal model, regardless of whether the nominal model is extracted from large-signal data or not. For example, if the nominal

model is obtained from extensive dc and multi-bias S-parameters, it can also be used in the proposed method.

### **5.3 Proposed Training Algorithm for the Statistical Neuro-SM Model**

The proposed statistical Neuro-SM process greatly reduces the modeling cost in data generation by using only one set of dc and small-/large-signal data from a nominal device and only dc and bias-dependent S-parameter data from all other device samples in the given population. To generate dc and S-parameter data for each sample, bias points are selected at several locations along the load line to cover the overall usable region of the device. The frequency points are selected around the operating frequency and various harmonics for large-signal operation.

Two populations of devices, called the training population and the test population, are used for data generation. The two populations have the same statistical distribution but different samples of devices. The data generated from the training population, called training data, is used for statistical model development. The data generated from the test population, called test data, is used for validating the statistical performance of the trained model. Large-signal test data, if available, can be used to further test the large-signal statistical behavior of the proposed model. Validation of the statistical model can be done by hypothesis test [111] using the test data. The statistical accuracy of the proposed model can be visually judged by comparing the cumulative probability distribution (CPD) of model responses with that of the test data, and quantitatively represented by the matching error defined as the difference between the two cumulative

probability distributions [87]. Figure 5.3 shows the flowchart for developing the statistical Neuro-SM model.

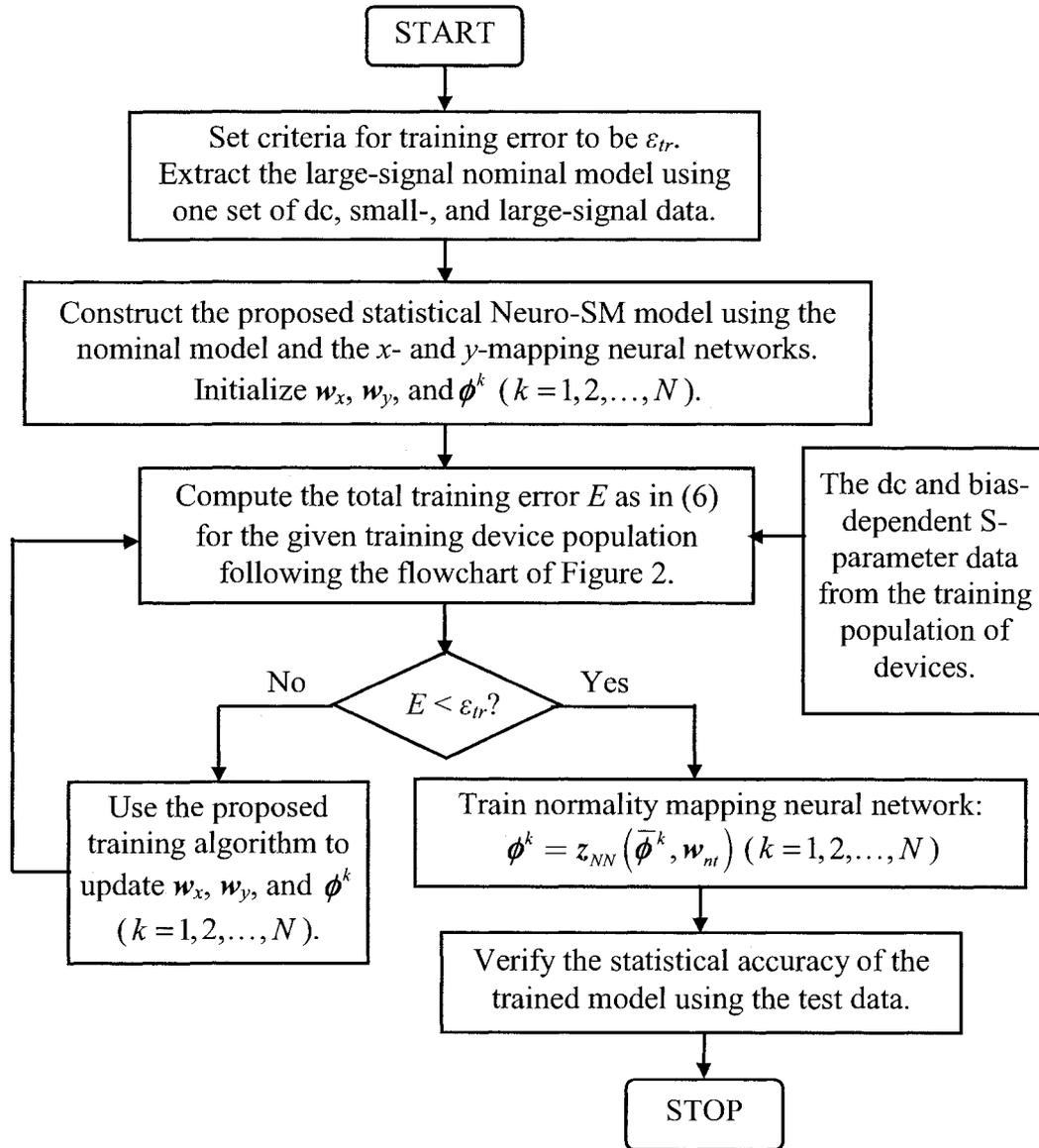


Figure 5.3: Flowchart for developing the proposed statistical Neuro-SM model.

- Step 1.* Select a nominal device as a typical representative of a population of devices. Extract a nominal model using dc, small-, and large-signal data from the nominal device.
- Step 2.* Generate dc and S-parameter data at multiple biases and frequencies for each and every device in the training and test populations.
- Step 3.* Set the criterion for the training error to be  $\varepsilon_{tr}$ . Initialize the weighting parameters of the  $x$ - and  $y$ -mapping neural networks  $w_x$  and  $w_y$ , and the statistical variables  $\phi^k$  ( $k=1,2,\dots,N$ ). Construct the statistical Neuro-SM model by combining the nominal model and the  $x$ - and  $y$ -mapping neural networks as in (5.1) and (5.2).
- Step 4.* Train the statistical Neuro-SM model using dc and S-parameter data of all device samples in the training population. Use an optimization algorithm to adjust  $w_x$ ,  $w_y$ , and  $\phi^k$  ( $k=1,2,\dots,N$ ) to minimize the training error  $E$  of (5.6).
- Step 5.* If the training error  $E$  is less than  $\varepsilon_{tr}$ , go to *Step 6*. Else, add hidden neurons or extra input neurons (statistical variables) to the mapping neural networks, and go to *Step 4* to continue training.
- Step 6.* Train a normality mapping neural network following the description in Subsection 5.2.4. Compute the correlation coefficients  $\rho(\bar{\phi})$  as in (5.9).
- Step 7.* Evaluate the proposed model using new test samples of statistical variables  $\bar{\phi}$  generated from a Gaussian distribution with  $\mu(\bar{\phi}) = \mathbf{0}$ ,  $\sigma(\bar{\phi}) = \mathbf{1}$ , and correlation coefficients  $\rho(\bar{\phi})$ .

*Step 8.* Compare the statistical behavior of the model responses with the dc and S-parameter data, or large-signal data, from the test population. If a satisfactory test accuracy cannot be reached, reduce the number of statistical variables of the  $x$ - and  $y$ -mapping neural networks, and retrain the model. If the retrained model still cannot meet the test accuracy, the statistical information in the training population may be inadequate. Increase the number of device samples in the training population and go to *Step 2*. Otherwise, the statistical Neuro-SM model (including the nominal model,  $x$ - and  $y$ -mapping neural networks, the normality mapping neural network, and the known distribution of  $\bar{\phi}$  from *Step 6*) is obtained and ready to be used for large-signal statistical simulations.

## 5.4 Application Examples

### 5.4.1 Statistical Neuro-SM Modeling of A MESFET Device

Here we illustrate the use of the proposed statistical Neuro-SM technique for statistical modeling and verification through a MESFET example. The training and testing populations of the MESFET are generated using *ADS* [106] statistical analysis on a built-in nonlinear GaAs MESFET [79], whose internal parameters are randomly varied around their nominal values. The nominal device in this example is the MESFET from Chapter 3. A set of dc, small-, and large-signal data is generated for the nominal device by *ADS* simulation. Based on this, a large-signal nominal model is extracted. The Statz model [78], used as the nominal model, is optimized to fit the dc, small-, and large-signal data of the nominal device. It achieves good accuracy in representing the complete behavior of

the nominal device. The training population consists of 100 device samples generated by varying the *ADS* MESFET parameters under a Gaussian distribution with  $\pm 5\%$  variations ( $\sigma/\mu$ ) around the nominal values. Three hundred (300) different device samples are generated under the same statistical distribution to form the test population. The dc and bias-dependent S-parameter data are generated for each device sample in both populations by *ADS* simulations at 40 frequency points (1 to 20 GHz) and 9 biases along the load line of the MESFET ( $V_g$ : -1.5 to 0 V,  $V_d$ : 0 to 4 V).

The proposed statistical Neuro-SM modeling is performed using different numbers, i.e., 4, 6, and 8, of statistical variables. The training algorithm in Section 5.3 is applied to train the proposed model using dc and S-parameter data from the training population to reach 1% training error as defined in (5.6). The statistical accuracy is evaluated using the matching error [87] between the cumulative probability distribution of the model responses (dc and S-parameters) and that of the test data. The matching errors are 4.79%, 2.52%, and 7.15% corresponding to the proposed model with 4, 6, and 8 statistical variables, respectively. This shows that 6 statistical variables are sufficient to represent the factors controlling the statistical variations in the device population. The model with fewer statistical variables may not learn the training data well. The model with too many variables may result in non-unique solutions during extraction of the statistical variables  $\phi$ . For the proposed model with 6 statistical variables, the  $x$ - and  $y$ -mapping neural networks are trained with 15 and 6 hidden neurons, respectively. A normality mapping neural network with 6 hidden neurons is also trained. Table 5.1 lists the correlation coefficients of the Gaussian variables  $\bar{\phi}$  computed by (5.9). The effect of normality

mapping is demonstrated by checking the closeness of statistical distributions of  $\phi$  and  $\bar{\phi}$  with ideal Gaussian distributions. The matching errors are evaluated as the difference between the cumulative probability distribution of each element in  $\phi$  or  $\bar{\phi}$  and the corresponding closest Gaussian distribution obtained by maximum likelihood estimation [94]. The matching errors for  $\phi$  are in the range of 3.19-8.04%, while the matching errors for  $\bar{\phi}$  are around 0.004%. It shows that the normality mapping neural network transforms the non-Gaussian variables  $\phi$  into good Gaussian variables  $\bar{\phi}$ .

**Table 5.1: Correlation coefficients of the Gaussian variables  $\bar{\phi}$  for the MESFET example.**

	$\bar{\phi}_1$	$\bar{\phi}_2$	$\bar{\phi}_3$	$\bar{\phi}_4$	$\bar{\phi}_5$	$\bar{\phi}_6$
$\bar{\phi}_1$	1.0					
$\bar{\phi}_2$	0.5712	1.0				
$\bar{\phi}_3$	-0.3513	-0.6668	1.0			
$\bar{\phi}_4$	-0.1192	-0.6044	0.5189	1.0		
$\bar{\phi}_5$	0.5072	0.0436	-0.0732	0.4391	1.0	
$\bar{\phi}_6$	0.3834	-0.1708	0.5776	0.2871	0.1029	1.0

For comparison purpose, another large-signal statistical model is also developed using the existing nonlinear statistical modeling method (i.e., with no mapping) following the approach in [90]-[92]. The existing approach would have been reliable if extensive

data is available for every device in the population. Here in our work, extensive data is available for only one (i.e., nominal) device. All other devices in the population have limited data. The Statz model, which has been used only for the nominal model in the proposed method, is used here to represent each and every device in the population. This is achieved by performing parameter extraction of the Statz model for each device using the same dc and S-parameter data as used in the proposed method. The principal components of the extracted Statz model parameters are analyzed, and a factor model with 14 common factors is built [93]. A power transformation [94] is performed to achieve an approximate Gaussian distribution for the extracted model parameters. For each device, the Statz model is able to fit accurately all of the data used (dc and S-parameter data). However, for each random device sample, the extracted values of the Statz parameters are not always unique using only the limited data (dc and S-parameter data at a few biases along the load line). This non-uniqueness becomes the main reason affecting the accuracy of the resulting statistical model.

A third model, using the previous linear statistical space mapping method in Chapter 4, is also created with the same large-signal nominal model as in the proposed statistical Neuro-SM model. The linear mapping network is constructed to achieve its optimal dynamic order and best available accuracy. The three statistical models, i.e., the proposed statistical Neuro-SM model, the statistical model with no mapping, and the linear statistical space-mapped model, are implemented into *ADS* for statistical verifications. Monte Carlo analyses of 300 dc and small-signal simulations are performed for each model. We evaluate the quality of the three models by comparing the means and standard

deviations of the real and imaginary parts of S-parameters from the model responses with those from the MESFET test data as illustrated in Figures 5.4 and 5.5. It shows that the means and standard deviations of the S-parameters from the test data can be reproduced more accurately using the proposed model than using the statistical model with no mapping and the linear statistical space-mapped model.

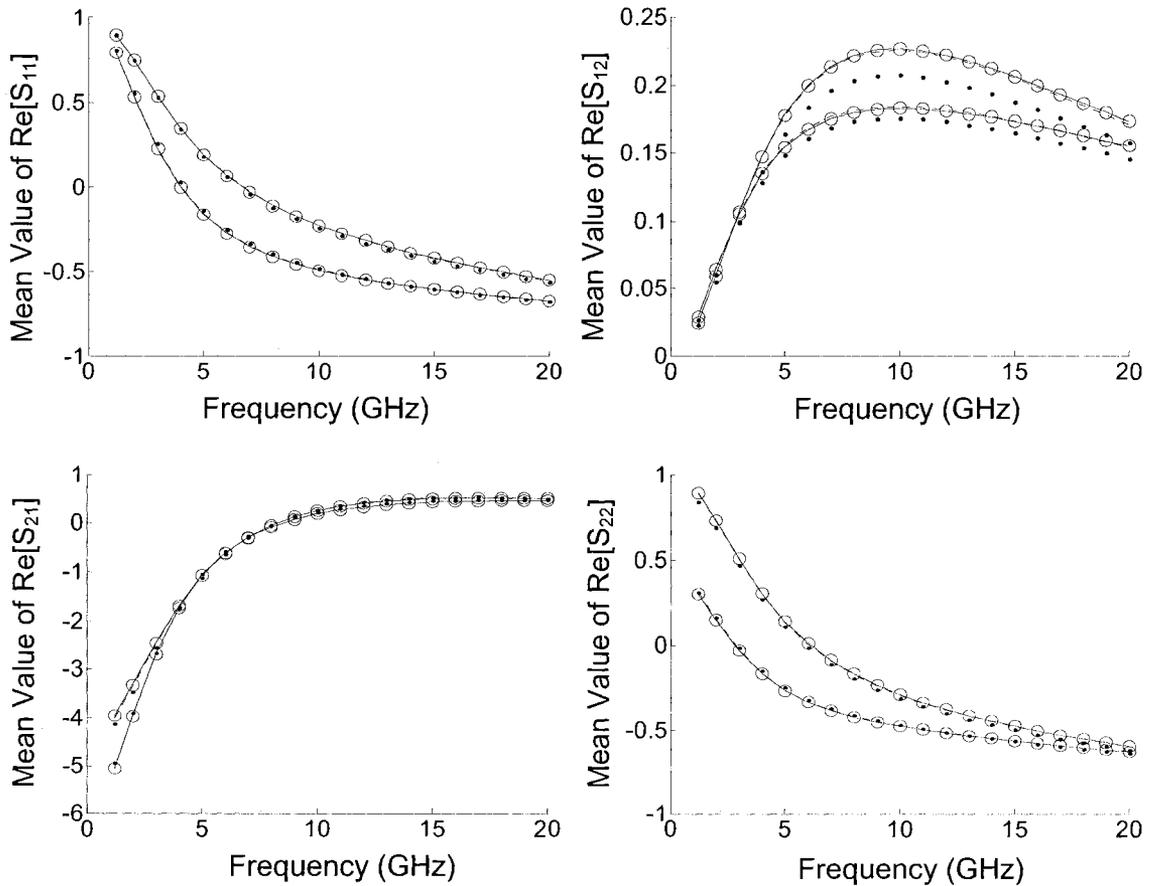
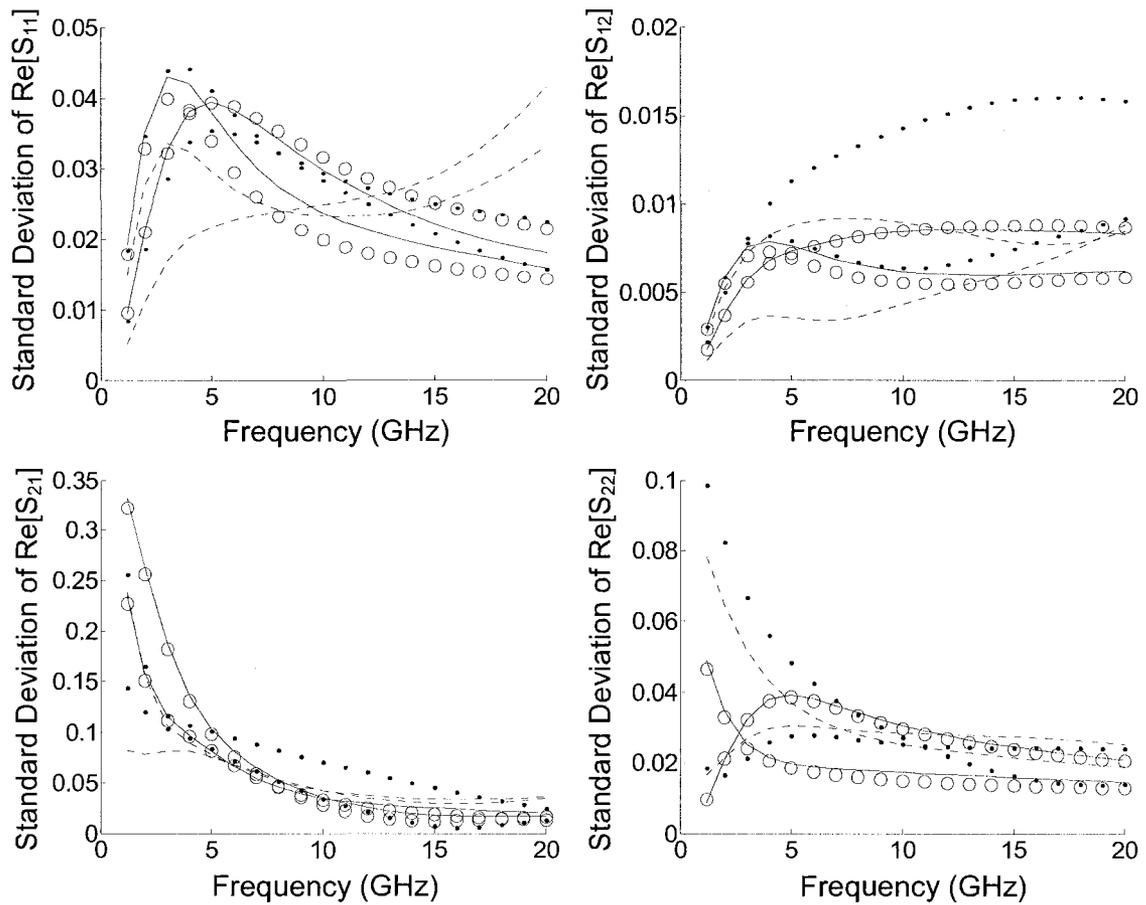


Figure 5.4: Mean values of the real part S-parameters at 2 biases from Monte-Carlo analyses of 300 small-signal simulations for the MESFET example. The comparison is done between the MESFET test data (o) and Monte-Carlo results using the statistical Neuro-SM model (—), the statistical model with no mapping (· ·), and the linear statistical space-mapped model (— —).



**Figure 5.5: Standard deviations of the real part S-parameters at 2 biases from Monte-Carlo analyses of 300 small-signal simulations for the MESFET example. The comparison is done between the MESFET test data (o) and Monte-Carlo results using the statistical Neuro-SM model (—), the statistical model with no mapping (· ·), and the linear statistical space-mapped model (—).**

Additional verification on the large-signal statistical behavior of the proposed model is carried out, by Monte Carlo analyses of harmonic balance simulations using a 2-tone power source with a center frequency of 1 GHz and a frequency spacing of 80 MHz. Three hundred (300) samples of the proposed model are used in the Monte Carlo analyses. The statistical responses, i.e., the third order intermodulation interception (IP3), the power added efficiency (PAE), and the power gain of the proposed model, are evaluated and compared with those from the MESFET test data. The same Monte Carlo analyses are also performed on models from the existing techniques, i.e., the statistical model with no mapping and the linear statistical space-mapped model. Table 5.2 compares the three modeling techniques in terms of statistical accuracy, number of statistical parameters, and model development time. All three models have good training accuracy represented by small errors of cumulative probability between the model responses and dc and S-parameter training data. The proposed technique has the best large-signal test accuracy. The linear statistical space mapping performs better than the existing technique with no mapping because of the embedded large-signal nominal model. The existing technique with no mapping has less accuracy in the large-signal test due to the limited data (say, the lack of large-signal data) for every device during model development. All models are developed on an Intel® Core™ 2 Quad CPU at 2.66 GHz. The proposed statistical Neuro-SM with the proposed training is much more efficient than the other two techniques with the device-by-device parameter extraction.

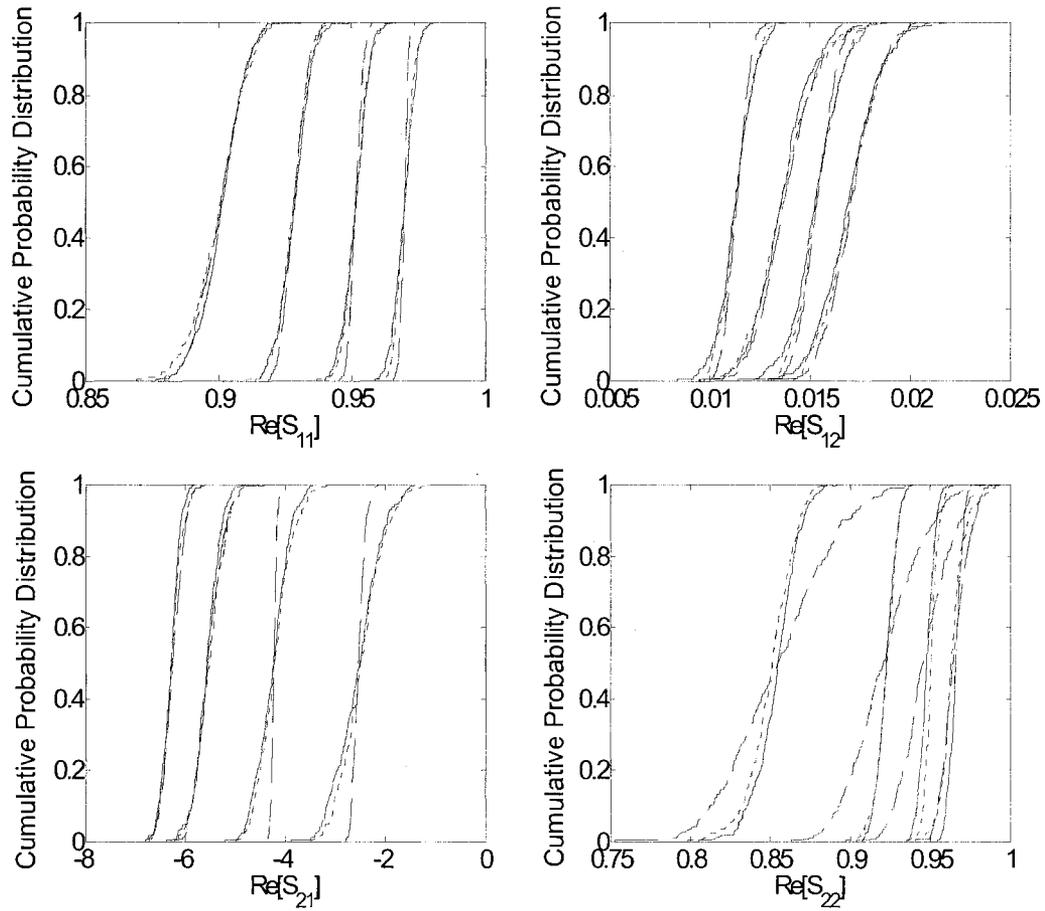
**Table 5.2: Comparison of statistical accuracy and modeling efficiency between three different techniques for the MESFET example.**

	Existing Technique (No Mapping)		Linear Statistical Mapping	Proposed Statistical Neuro-SM
	No Normality Transform	Power Transform		
dc&S Training Error (CPD matching error in %)	1.95	1.95	1.99	1.71
dc&S Test Error (CPD matching error in %)	9.78	8.12	5.34	2.52
IP3 Test Error (CPD matching error in %)	54.67	48.97	10.12	4.32
PAE Test Error (CPD matching error in %)	35.02	31.78	14.04	3.58
Power Gain Test Error (CPD matching error in %)	36.13	32.90	12.54	3.56
Number of Statistical Parameters	14	14	10	6
CPU Time for Training (hrs)	7.47	7.47	3.20	0.58

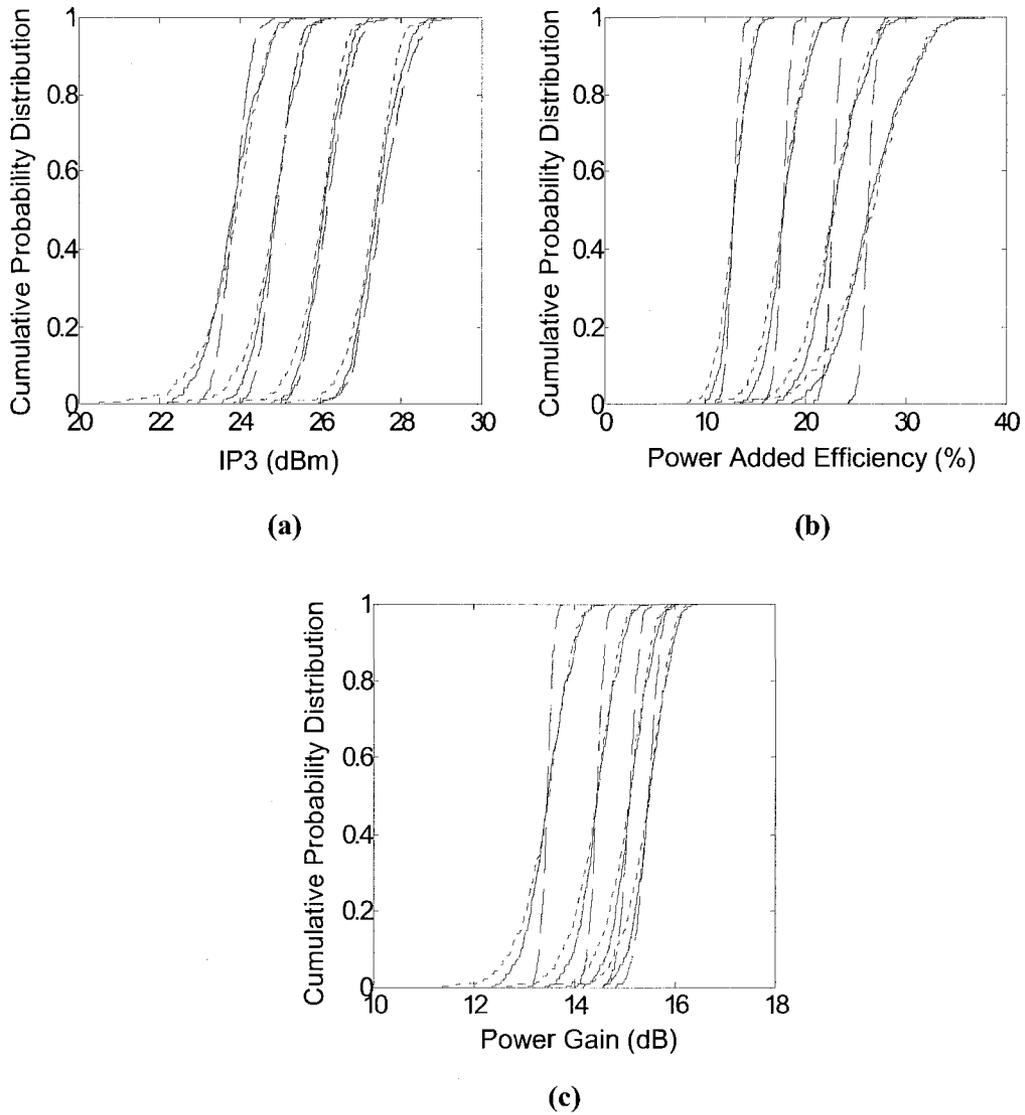
As a further step, we perform a hypothesis test to examine the statistical accuracy of the proposed model and compare it with that of the linear statistical space mapping. Our statistical hypothesis tests are carried out using the *Matlab* statistical toolbox [112]. A cumulative significance level of  $\alpha = 0.05$  is considered to perform hypothesis tests of the models using the real and imaginary parts of S-parameters in the 1 to 20 GHz frequency range. To check the statistical equivalence between the S-parameters computed from the model versus the S-parameter data from the test population, we perform the t-test on the mean values and the F-test on the standard deviations of the S-parameters [91], [111]. We

also perform hypothesis tests on the correlation coefficients after applying the Fisher Z-transform [91], [111]. For all 8 responses (real and imaginary parts of S-parameters), the means, the standard deviations, and the 28 correlation coefficients from the proposed model are tested to be statistically equivalent to those from the test device population. However, at the same significance level of  $\alpha = 0.05$ , 4 out of the 8 responses and 8 out of the 28 correlation coefficients from the linear statistical space-mapped model fail the hypothesis tests. The results of hypothesis sign test show that out of the 28 correlation coefficients, the proposed model has no sign opposite to those obtained from the test data while the linear space-mapped model has one. Another type of hypothesis test, Kolmogorov-Smirnov (K-S) goodness-of-fit test [111], is also performed at a significance level of  $\alpha = 0.05$  using the S-parameters. All 8 responses from the proposed model pass the test while only 3 responses pass for the linear statistical space-mapped model, confirming the statistical equivalence of the proposed model versus the test data. Figures 5.6 and 5.7 compare the cumulative probability distributions of the small- and large-signal responses from the statistical models with those from the test data. It illustrates the statistical equivalence between the proposed statistical Neuro-SM model and the original MESFET, demonstrating the much-enhanced quality of the proposed model over the previous linear statistical space-mapped model.

We also compare the proposed statistical Neuro-SM with the linear statistical space mapping of Chapter 4 for 3 different sizes of statistical variations ( $\sigma/\mu$ ), i.e., 3%, 5%, and 10%, of the MESFET parameters. The linear statistical space-mapped models are developed with different dynamic orders, and the model with the most suitable order and



**Figure 5.6: Cumulative probability distributions (CPD) of real part S-parameters at 1 GHz for 4 biases from Monte Carlo analyses of 300 small-signal simulations for the MESFET example. Such CPDs are used for a K-S test between the MESFET test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (- -).**



**Figure 5.7: Cumulative probability distributions (CPD) of (a) third order intermodulation interception (IP3), (b) power added efficiency, and (c) power gain at 4 input power levels from Monte Carlo analyses of 300 two-tone HB simulations for the MESFET example. Such CPDs are used for a K-S test between the MESFET test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (- -).**

best accuracy is used for comparisons. The Models developed from both techniques use the same nominal model and the same dc and S-parameter training data for the statistical parameter extraction. Monte Carlo analyses of 300 dc, small-signal, and large-signal simulations are performed on the models, and the results are compared with data from the test population. Table 5.3 compares the two techniques in terms of statistical accuracy and number of statistical parameters. It demonstrates that for different sizes of variations, the proposed statistical Neuro-SM models developed from only one set of large-signal data

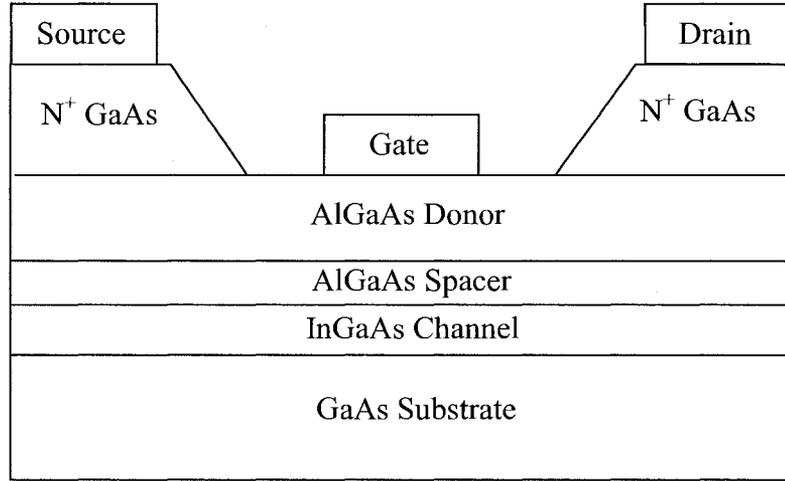
**Table 5.3: Cumulative probability error between the statistical model responses and the test data for the MESFET example. The proposed model has significantly better accuracy than the linear statistical space-mapped model as statistical variations become large.**

$\sigma/\mu$	List of Comparisons	Linear Statistical Space-Mapped Model	Proposed Statistical Neuro-SM Model
3%	No. of Statistical Parameters	7	6
	dc & S (%)	3.01	2.66
	Gain (%)	3.19	2.91
	IP3(%)	3.38	4.76
	PAE (%)	2.79	2.64
5%	No. of Statistical Parameters	10	6
	dc & S (%)	5.34	2.52
	Gain (%)	12.54	3.56
	IP3 (%)	10.12	4.32
	PAE (%)	14.04	3.58
10%	No. of Statistical Parameters	15	6
	dc & S (%)	10.63	3.54
	Gain (%)	44.14	5.61
	IP3 (%)	50.08	6.58
	PAE (%)	46.97	5.02

achieve good statistical accuracy in not only dc and small-signal simulations, but also large-signal simulations. It outperforms the linear statistical space mapping in modeling large statistical variations, while retaining the same accuracy in the small statistical variation case.

#### **5.4.2 Statistical Neuro-SM Modeling of a HEMT Device from a Physics-Based Device Simulator**

This example models process variation induced effects on the electrical behavior of a HEMT device. A HEMT structure from the physics-based device-level simulator *Synopsys Medici* [113], shown in Figure 5.8, is used to generate random devices. Ten geometrical and physical parameters in the HEMT structure are randomly varied using Gaussian distributions with variations ( $\sigma/\mu$ ) of  $\pm 5\%$  around their mean values given in Table 5.4. A large-signal nominal model [114] is first extracted from *Agilent IC-CAP* [115], and further refined in *ADS* [106] using large-signal data generated by *Medici* on a nominal device. A training population of 100 HEMT structures and a test population of 250 HEMT structures with randomly varying geometrical and physical parameters are created. The dc and bias-dependent S-parameter data are generated for the two populations by solving the device Poisson equations in *Medici* at 34 frequencies (10 to 50 GHz) and 10 biases across the load line of the HEMT ( $V_g$ : 0 to 1 V,  $V_d$ : 0 to 5 V). Large-signal HB data is also generated (fundamental frequency: 10 GHz, input power:  $-15$  to 5 dBm) for the test population by performing transient simulations and Fourier transforms in *Medici*. The 100 sets of dc and S-parameter data are used for model training. The 250 sets of dc, S-parameter, and HB data are used for statistical verification.



**Figure 5.8: HEMT structure in *Medici* used for data generation of random device samples, where 10 process parameters are subject to random variations.**

**Table 5.4: Mean values of geometrical/physical parameters For HEMT device.**

Parameter Name		Mean Value ( $\mu$ )
Gate Length ( $\mu\text{m}$ )		0.4
Gate Width ( $\mu\text{m}$ )		100
Thickness ( $\mu\text{m}$ )	AlGaAs Donor Layer	0.025
	AlGaAs Spacer Layer	0.01
	InGaAs Channel Layer	0.01
	GaAs Substrate	0.045
Doping Density ( $1/\text{m}^3$ )	InGaAs Channel Layer	1e2
	AlGaAs Donor Layer	1e18
	Source N+	2e20
	Drain N+	2e20

The proposed statistical Neuro-SM modeling process is performed using 3 different numbers, i.e., 3, 5, and 7, of statistical variables. After training, we evaluate the statistical accuracy using the matching error between the cumulative probability distribution of the model responses (dc and S-parameters) and that of the test data. The matching errors for the proposed model with the 3, 5, and 7 statistical variables are 9.57%, 3.34%, and 8.08%, respectively, showing that the model with 5 statistical variables is the most accurate. For the proposed model with 5 statistical variables, the numbers of hidden neurons used for  $x$ -mapping,  $y$ -mapping, and the normality mapping neural networks are 15, 8, and 15, respectively. The optimization of (5.7) is performed to train the mapping neural networks and extract statistical variables  $\phi$ . The normality mapping neural network is trained to relate the extracted  $\phi$  to Gaussian variables  $\bar{\phi}$ . Table 5.5 shows the correlation coefficients of the Gaussian variables  $\bar{\phi}$  computed by (5.9). To demonstrate the effect of normality mapping, we examine the closeness of the statistical distributions of the extracted statistical variables  $\phi$  (or  $\bar{\phi}$ ) with an ideal Gaussian distribution. The matching errors between the cumulative probability distribution of the extracted  $\phi$  versus that of an ideal Gaussian distribution are in the range of 4.03% to 6.75%, while the corresponding errors for  $\bar{\phi}$  are around 0.004%. It again confirms that the normality mapping neural network efficiently transforms the non-Gaussian distribution of  $\phi$  into an ideal Gaussian distribution of  $\bar{\phi}$ .

**Table 5.5: Correlation coefficients of the Gaussian variables  $\bar{\phi}$  for the HEMT example.**

	$\bar{\phi}_1$	$\bar{\phi}_2$	$\bar{\phi}_3$	$\bar{\phi}_4$	$\bar{\phi}_5$
$\bar{\phi}_1$	1.0				
$\bar{\phi}_2$	0.0491	1.0			
$\bar{\phi}_3$	0.1051	0.8197	1.0		
$\bar{\phi}_4$	-0.5859	0.5053	0.4732	1.0	
$\bar{\phi}_5$	-0.2697	-0.1831	0.3227	0.1430	1.0

For comparison purposes, another large-signal statistical model is developed using the existing nonlinear statistical modeling technique (with no mapping) following the approach in [90]-[92]. The Angelov model [114], which has been used only for the nominal model in the proposed method, is used here to represent each and every device in the HEMT population. This is achieved by performing parameter extraction of the Angelov model for each device from the same set of limited dc and S-parameter data as used in the proposed method. Principal component and factor analysis [93], and power transformations [94] are performed on the extracted parameters to reduce the dimension of the statistical parameters and achieve approximate Gaussian distribution. The Angelov model is shown to be able to fit accurately the limited dc and S-parameter data at a few biases along the load line. However, for each random device sample, the extracted values of the Angelov parameters are not always unique using only the limited data. This non-uniqueness affects the accuracy of the resulting statistical model.

A linear statistical space-mapped model of Chapter 4 is also created using the same nominal model and the same set of dc and S-parameter data as used in the proposed method. It is developed to achieve best available accuracy with optimal dynamic order. All three models, i.e., the proposed statistical Neuro-SM model, the statistical model with no mapping, and the linear statistical space-mapped model are implemented into *ADS* [106] to perform Monte Carlo analyses with 250 dc and small-signal simulations. The quality of the three models are evaluated by comparing the means and standard deviations of the real and imaginary parts of S-parameters from the model responses with those from the HEMT test data. As illustrated in Figures 5.9 and 5.10, the means and standard deviations of the S-parameters from the test data can be reproduced more accurately using the proposed model than using the statistical model with no mapping and the linear statistical space-mapped model.

Additional verifications on the large-signal statistical behavior of the statistical models are performed by Monte Carlo analyses of 250 large-signal HB simulations at 10 GHz. Table 5.6 compares the performance between the proposed model, the statistical model with no mapping, and the linear statistical space-mapped model. All three models have good training accuracy. The proposed statistical Neuro-SM achieves the best test accuracy in dc, small-, and large-signal statistical simulations. The linear statistical space mapping has better test accuracy than the existing technique with no mapping because of the embedded large-signal nominal model. The existing technique with no mapping is less accurate due to limited data (i.e., lack of large-signal data) for every device during

parameter extraction. It also shows that the proposed statistical Neuro-SM modeling with the proposed training is the most efficient among the three modeling techniques.

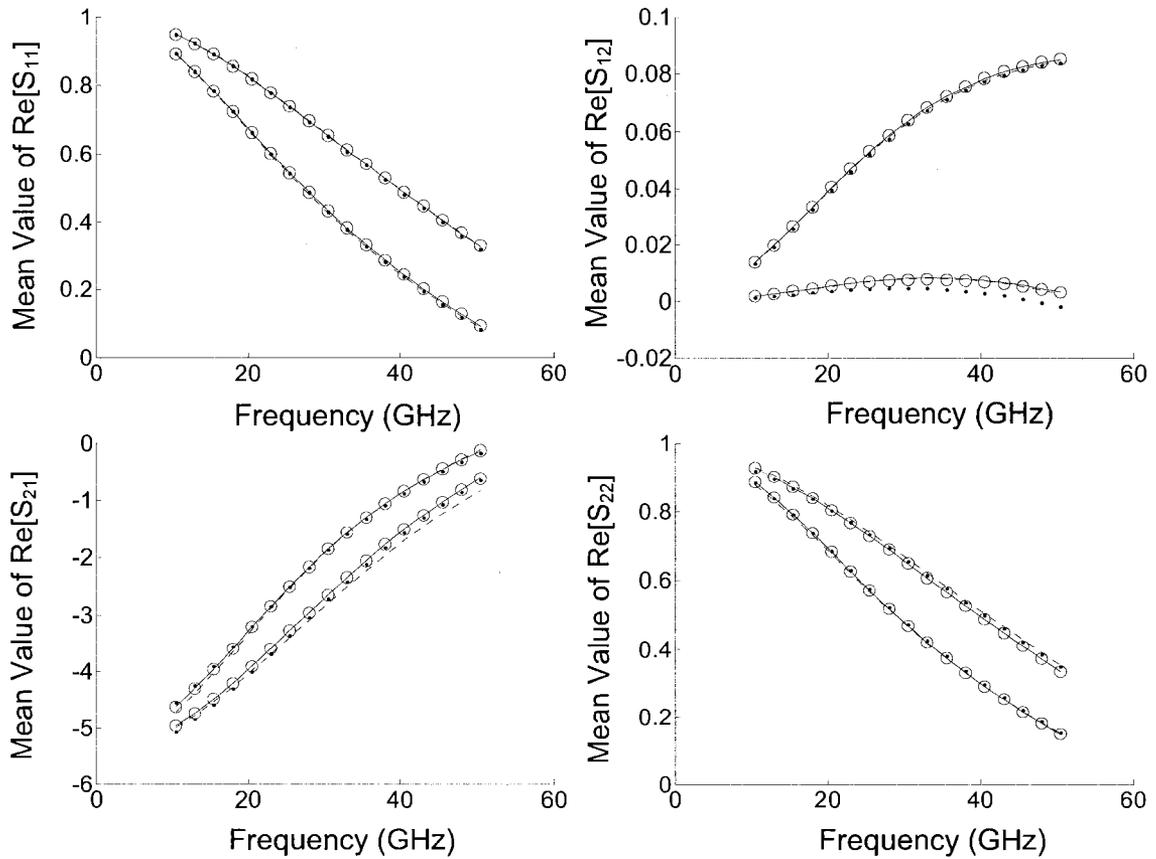
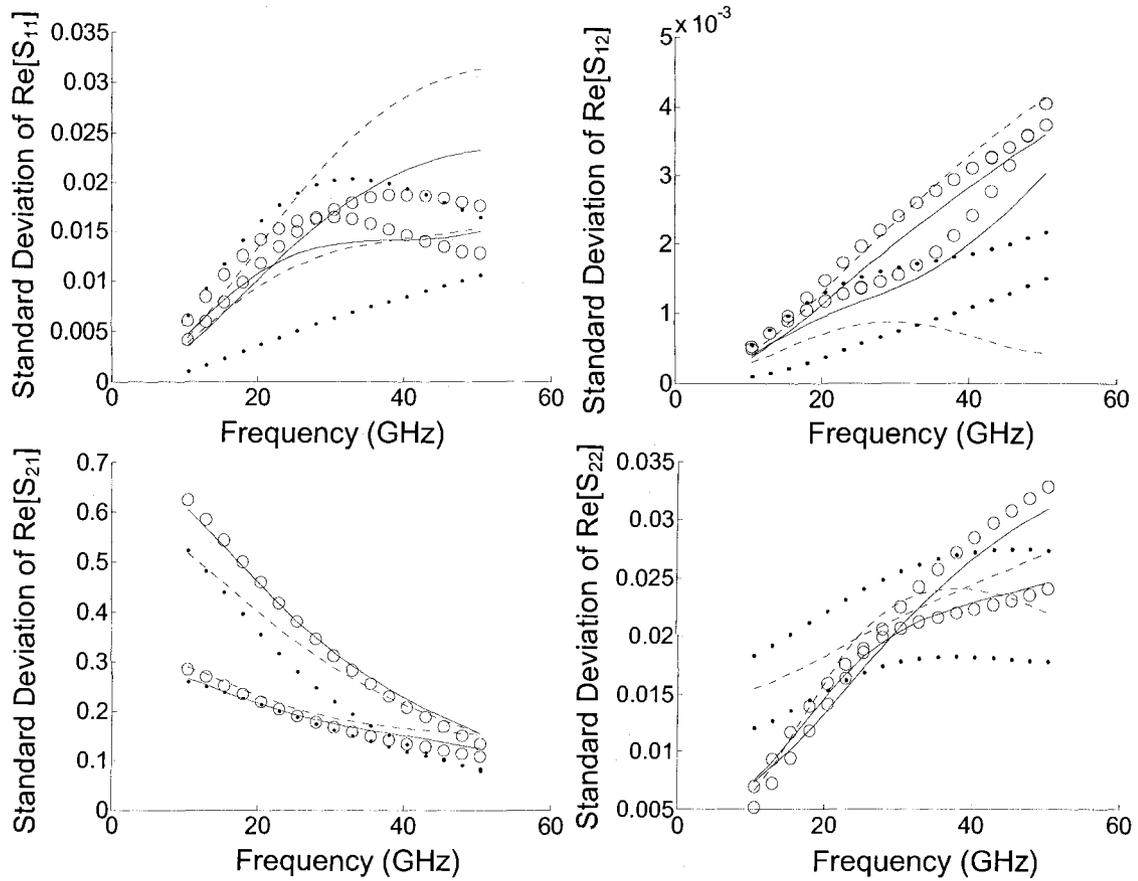


Figure 5.9: Mean values of the real part S-parameters at 2 biases from Monte-Carlo analyses of 250 small-signal simulations for the HEMT example. The comparison is done between the HEMT test data (o) and the Monte-Carlo results using the statistical Neuro-SM model (—), the statistical model with no mapping (.), and the linear statistical space-mapped model (—).



**Figure 5.10: Standard deviations of the real part S-parameters at 2 biases from Monte-Carlo analyses of 250 small-signal simulations for the HEMT example. The comparison is done between the HEMT test data (o) and the Monte-Carlo results using statistical Neuro-SM model (—), the statistical model with no mapping (··), and the linear statistical space-mapped model (—).**

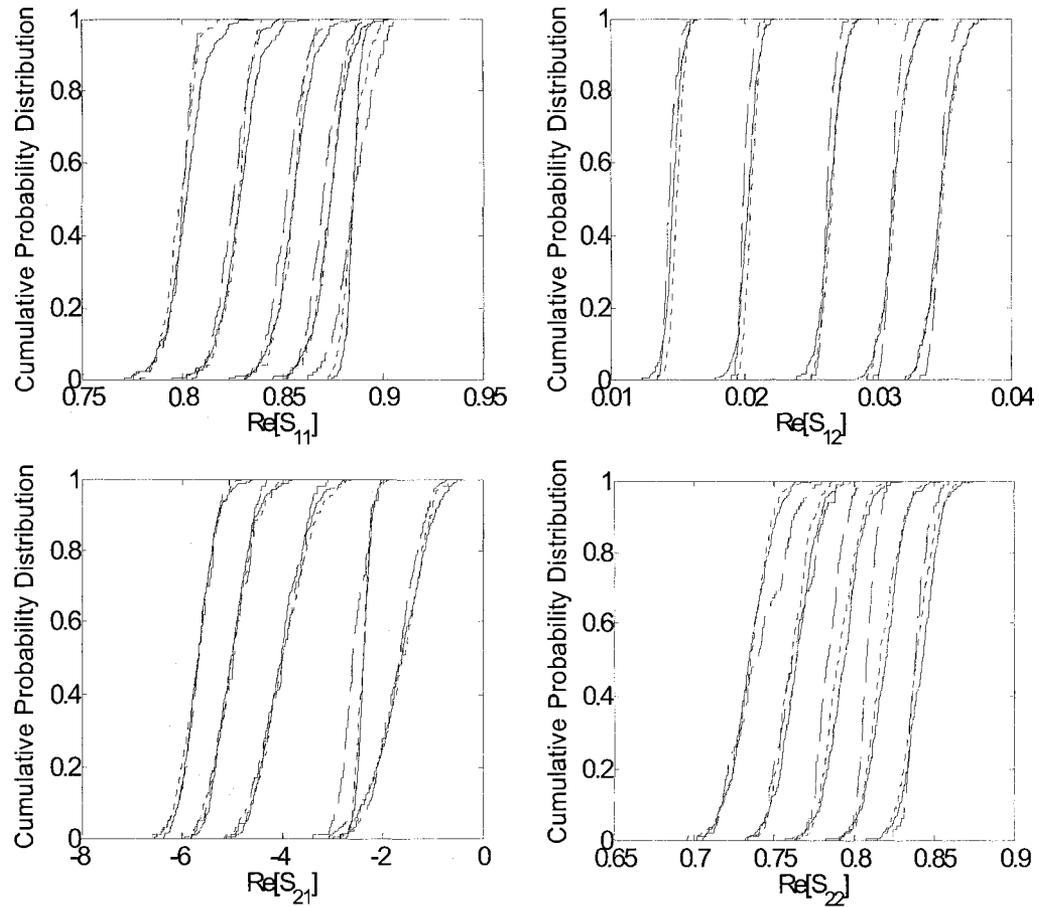
**Table 5.6: Comparison of statistical accuracy and modeling efficiency between three different techniques for the HEMT example.**

	Existing Technique (No Mapping)		Linear Statistical Mapping	Proposed Statistical Neuro-SM
	No Normality Transform	Power Transform		
Training Error (CPD matching error in %)	1.78	1.78	1.86	1.63
dc&S Test Error (CPD matching error in %)	18.84	14.91	12.88	3.34
HB Test Error (CPD matching error in %)	49.22	41.38	19.64	5.48
Number of Statistical Parameters	24	24	18	5
CPU Time for training (hrs)	42.05	42.05	20.18	1.92

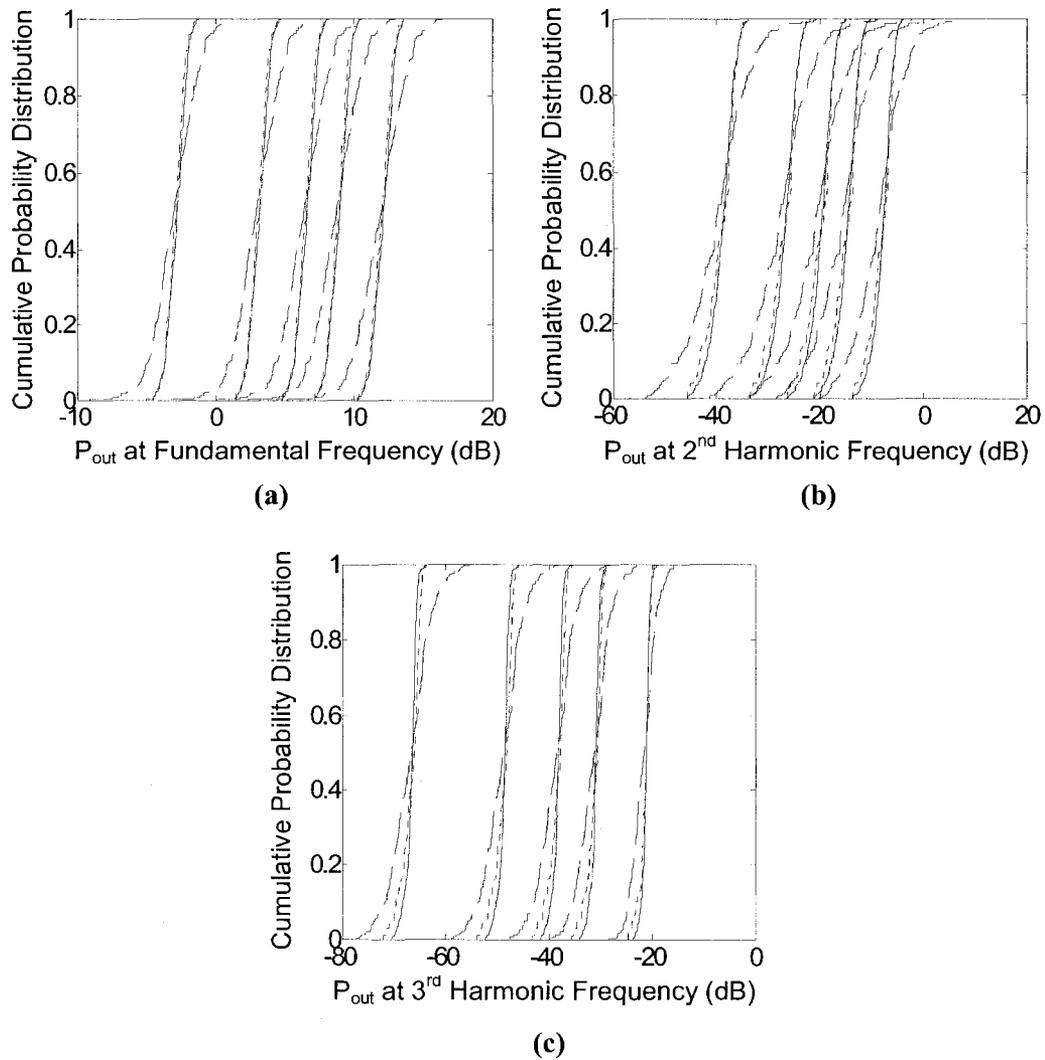
We further perform statistical hypothesis tests to compare the accuracy of the proposed statistical Neuro-SM model with that of the linear statistical space-mapped model. The hypothesis tests are carried out using the *Matlab* statistical toolbox [112]. A cumulative significance level of  $\alpha = 0.05$  is considered to test the models using the real and imaginary parts of S-parameters in the 1-50 GHz frequency range. The same types of hypothesis tests [91], [111] as in Example *A* are performed to test the statistical equivalence between the models and the HEMT test data. It is shown that for the given significance level of test, the proposed model is statistically equivalent to the test device population for all 8 responses (real and imaginary parts of S-parameters) and 28 correlation coefficients, while the linear statistical space-mapped model fails for 5 out of 8 responses and 12 out of 28 correlation coefficients. Sign test shows that out of the 28

correlation coefficients, the proposed model has no sign opposite to those from the original test data, while the linear statistical space-mapped model has 5. A K-S goodness-of-fit test [111] is also performed under a significance level of  $\alpha = 0.05$  using the S-parameters. All 8 responses from the proposed model pass the K-S test while only 2 responses pass for the linear statistical space-mapped model. Figures 5.11 and 5.12 compare the cumulative probability distributions of the small-signal and large-signal responses from the statistical models with those of the test data. It further demonstrates that the proposed statistical Neuro-SM can produce significantly improved accuracy over the linear statistical space mapping.

To illustrate the advantage of the proposed statistical Neuro-SM in modeling large statistical variations, we re-perform statistical modeling for different ranges of random variations in the process parameters of Table 5.4. Statistical modeling for 3 different sizes of statistical variations ( $\sigma/\mu$ ) of 3%, 5%, and 10% are carried out using the proposed statistical Neuro-SM and the linear statistical space mapping of Chapter 4. The models developed from both techniques use the same nominal model and the same dc and S-parameter training data for statistical parameter extraction. Monte Carlo analyses of 250 dc, small-signal, and large-signal simulations are performed on the models, and the results are compared with data from the test population as shown in Table 5.7. It is observed that for small statistical variations, both the proposed model and the linear statistical space-mapped model have good accuracy. It is also demonstrated that the proposed statistical Neuro-SM model has significantly better accuracy than the linear statistical space-mapped model as the statistical variations become large.



**Figure 5.11: Cumulative probability distributions (CPD) of real part S-parameters at 10 GHz for 5 biases from Monte Carlo analyses of 250 small-signal simulations for the HEMT example. Such CPDs are used for the K-S test between the HEMT test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (- -).**



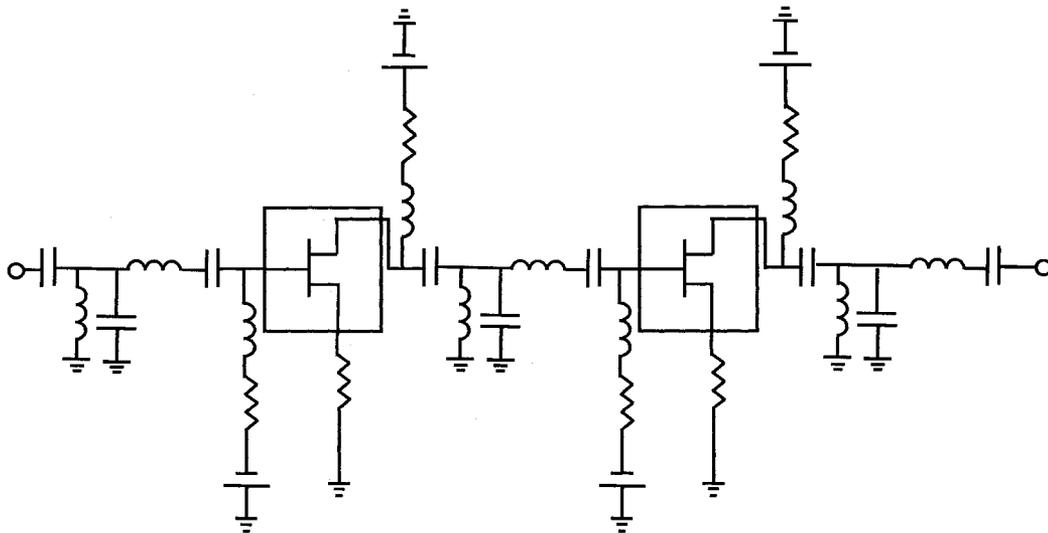
**Figure 5.12: Cumulative probability distributions (CPD) of output power in dB at (a) fundamental, (b) second, and (c) third harmonic frequencies for 5 input power levels from Monte Carlo analyses of 250 HB simulations for the HEMT example. Such CPDs are used for the K-S test between the HEMT large-signal test data (—) and the Monte Carlo results using the proposed statistical Neuro-SM model (--) and the linear statistical space-mapped model (---).**

**Table 5.7: Cumulative probability error between the statistical model responses and the test data for the HEMT example. The proposed model has significantly better accuracy than the linear statistical space-mapped model as statistical variations become large.**

$\sigma/\mu$	List of Comparisons	Linear Statistical Space-Mapped Model	Proposed Statistical Neuro-SM Model
3%	No. of Statistical Parameters	10	5
	dc & S (%)	4.73	3.39
	HB (%)	5.18	4.49
5%	No. of Statistical Parameters	18	5
	dc & S (%)	12.88	3.34
	HB (%)	19.64	5.48
10%	No. of Statistical Parameters	18	5
	dc & S (%)	20.34	4.94
	HB (%)	46.51	8.23

### 5.4.3 Use of Statistical Neuro-SM Models in Two-Stage Amplifier Simulation

In this example, the statistical models developed in the preceding subsections are used in the statistical analysis of the two-stage amplifier circuit shown in Figure 5.13.

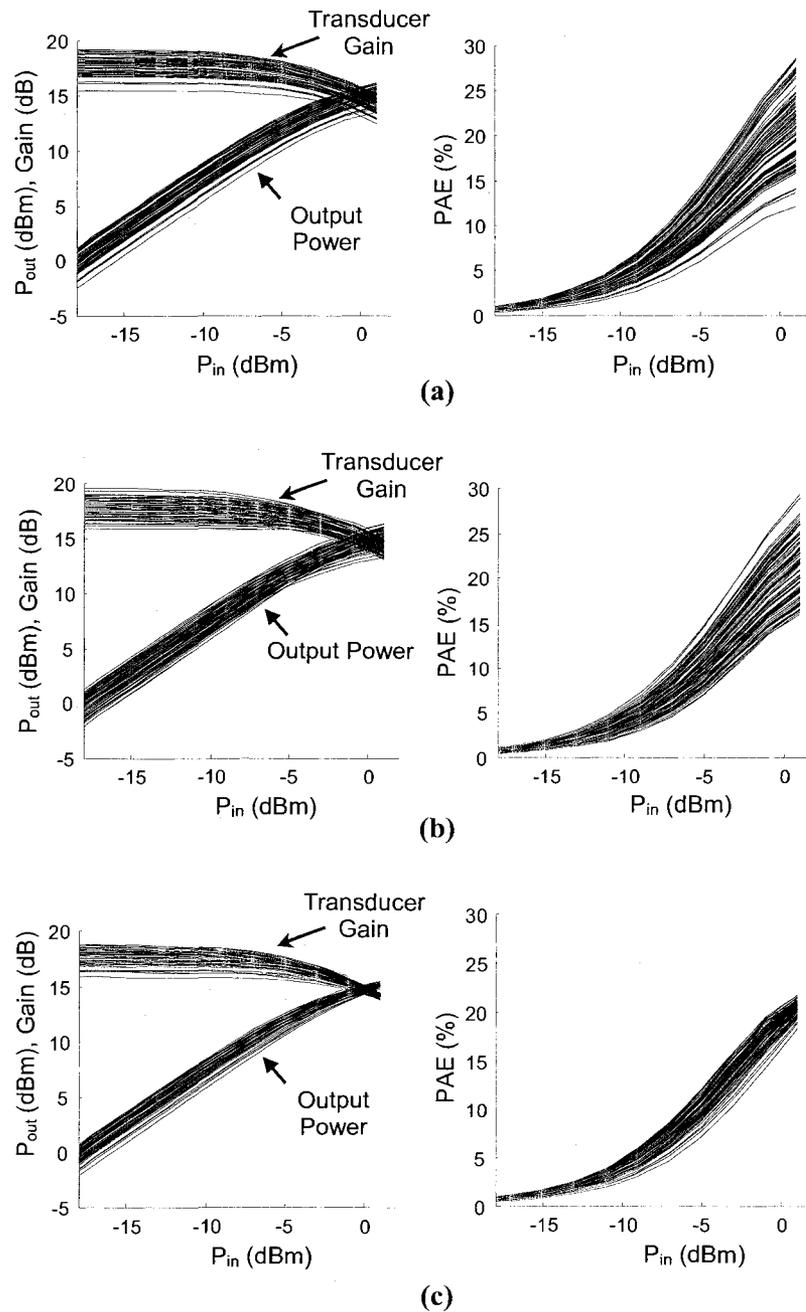


**Figure 5.13:** A two-stage amplifier whose transistors are represented by our statistical models.

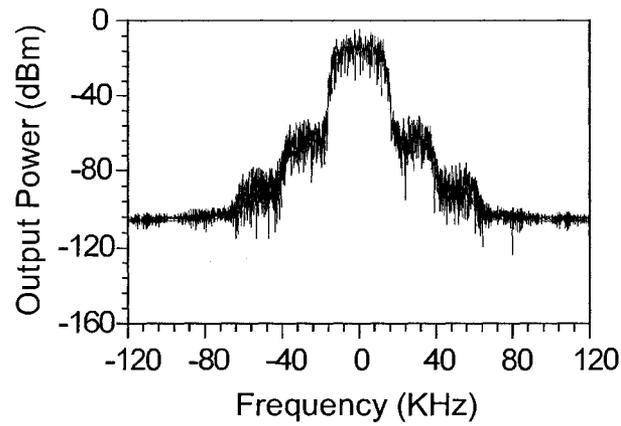
We first use the MESFET device model in the two-stage amplifier. The proposed statistical Neuro-SM model and the linear statistical space-mapped model developed in example of 5.4.1 are incorporated into *ADS* for circuit simulation. A  $\pi/4$  differential-quaternary-phase-shift-keying (DQPSK) modulated signal with symbol rate of 24.3 Ksps at 1 GHz is supplied. The offset frequency is 30 kHz, and the channel bandwidth is 32.8 kHz. Monte Carlo analyses of 100 circuit envelope simulations of the two-stage amplifier are performed. Figures 5.14 and 5.15 compare the statistical distributions of the large-

signal responses, such as transducer gain, main channel output power, power added efficiency, and output spectrum, of the two-stage amplifier between using the original *ADS* MESFET and the statistical models. Figure 5.16 compares the cumulative probability distributions of the amplifier responses. A K-S test is performed at a cumulative significance level of  $\alpha = 0.05$  to confirm the statistical equivalence between the amplifier responses using the original *ADS* MESFET and the proposed model. It demonstrates that the statistical behavior reproduced by the proposed statistical Neuro-SM model is much closer to the original than that by the linear statistical space-mapped model.

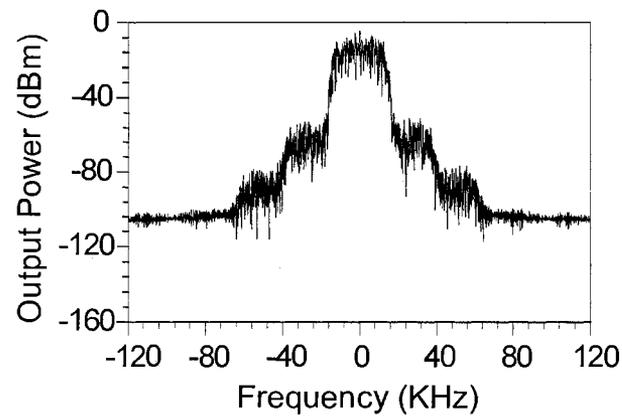
We further demonstrate the use of the statistical Neuro-SM model developed for the HEMT of example of 5.4.2 in the two-stage amplifier simulation, where the amplifier is re-optimized to work in a 10 GHz frequency region. A  $\pi/4$  DQPSK modulated signal with symbol rate of 227 Ksps at 10 GHz, offset frequency of 284 kHz, and a channel bandwidth of 306.5 kHz is used. Monte Carlo analyses of 100 circuit envelope simulations are performed. Figure 5.17 shows the simulation results of the main channel output power, the transducer gain, power added efficiency, and output spectrum. Through statistical Neuro-SM, the proposed approach enables convenient incorporation of the previously unavailable physics-based device of *Medici* into *ADS*, and provides a reliable large-signal statistical model for circuit simulation and design.



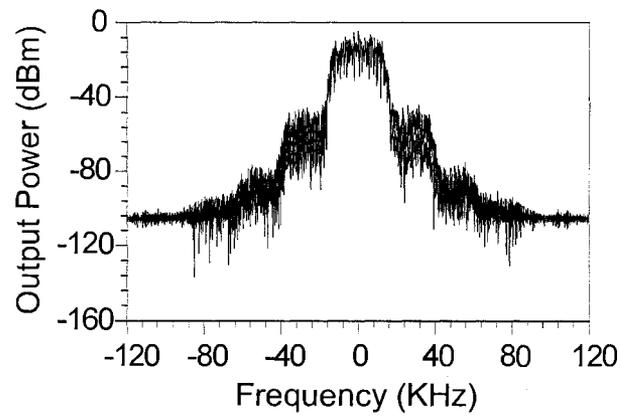
**Figure 5.14: Transducer gain, main channel output power, and power added efficiency (PAE) from Monte Carlo analyses of 100 circuit envelop simulations of the two-stage amplifier using (a) the original MESFET, (b) the proposed statistical Neuro-SM model, and (c) the linear statistical space-mapped model. The statistical behavior of the amplifier can be reproduced more accurately using the proposed model than using the linear statistical space-mapped model.**



(a)

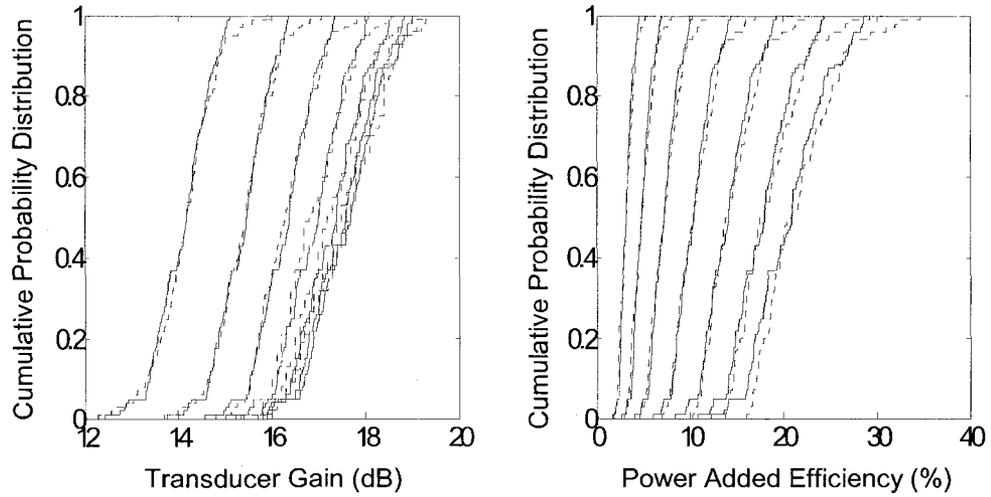


(b)

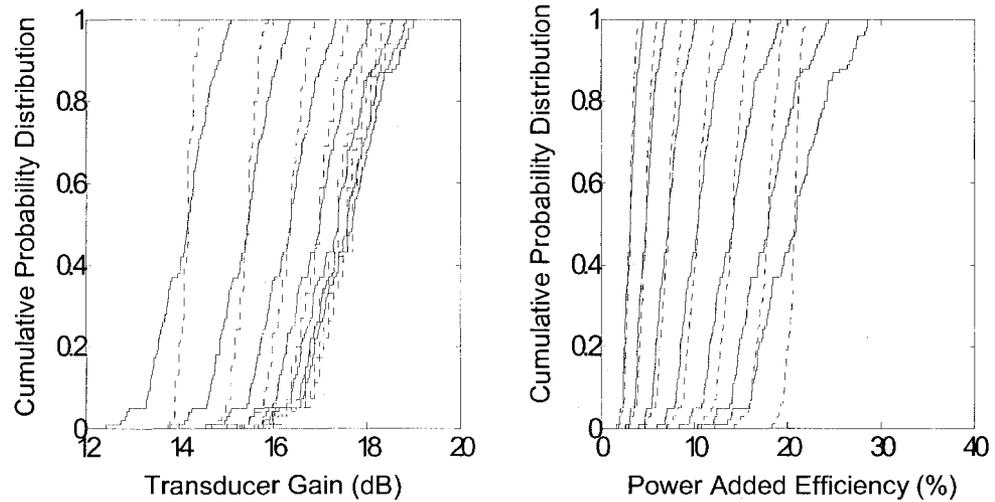


(c)

Figure 5.15: The output power spectrum of the two-stage amplifier for Monte Carlo analyses of 100 circuit envelope simulations using (a) the original MESFET, (b) the proposed statistical Neuro-SM model, and (c) the linear statistical space-mapped model.



(a)



(b)

**Figure 5.16: Comparison of cumulative probability distributions for transducer gain and power added efficiency from 100 circuit envelope simulations between using the original MESFET (—) and the statistical models: (a) the proposed statistical Neuro-SM model (--) and (b) the linear statistical space-mapped model (--).**

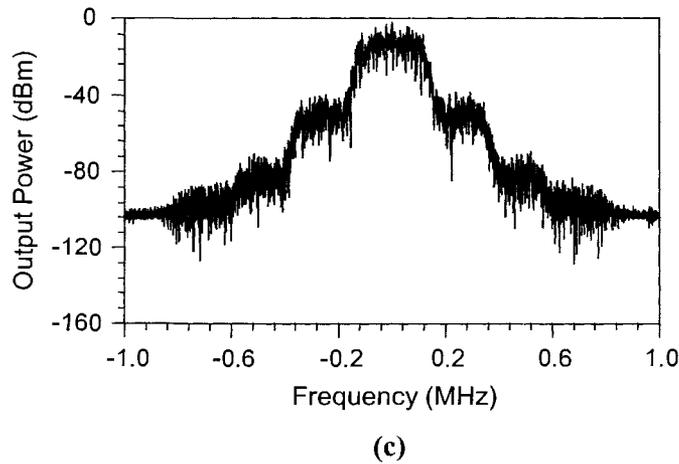
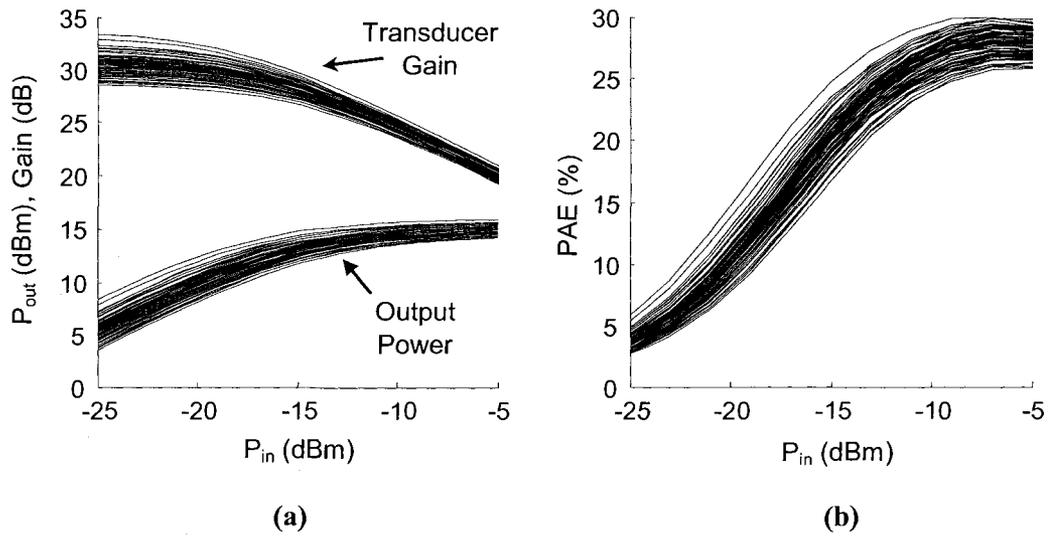


Figure 5.17: (a) Transducer gain and main channel output power, (b) power added efficiency (PAE), and (c) output power spectrum from Monte Carlo analyses of 100 circuit envelope simulations of the two-stage amplifier using the statistical Neuro-SM model for the HEMT.

## 5.5 Conclusions

A new statistical Neuro-SM technique for large-signal statistical modeling of nonlinear microwave devices has been presented. It fills a gap in the previous linear statistical space mapping technique by addressing large statistical variations in devices using nonlinear mapping. A major limitation in the linear mapping has been overcome by a new neural-based formulation where the increased complexity of the mapping, required for large variations, can be achieved without increasing the number of statistical parameters. The mapping in the proposed model enables efficient utilization of large-signal data for nominal model extraction and dc and small-signal data for statistical characterization. The trained model can reliably reproduce the large-signal statistical behavior of a given population of random devices. The proposed technique is applied to statistical modeling of metal-semiconductor field effect transistor (MESFET) and high electron mobility transistor (HEMT) devices, and use of the models in large-signal statistical analyses of a two-stage amplifier. Compared with existing large-signal statistical modeling techniques, the proposed statistical Neuro-SM has demonstrated much-improved performance in terms of accuracy and/or efficiency. It is very useful for statistical design of microwave circuits.

## **Chapter 6: Conclusions and Future Work**

### **6.1 Conclusions**

Recent cutting-edge research has led to automated and optimal transistor modeling based on neural network techniques [2]-[4]. In the meantime, the evolution of the space mapping concept has also resulted in major breakthroughs in microwave modeling area [13], [60], allowing expensive and time-consuming microwave component modeling to be performed economically and efficiently through the use of surrogate (coarse) models.

This thesis has presented systematic research combining the neural-based microwave modeling with the state-of-the-art optimization concept of space mapping [13]. It aims to accomplish efficient and accurate modeling of nonlinear microwave devices.

This thesis first addresses the neuro-space mapping technique for nonlinear microwave device modeling, using neural networks to map an existing device model with coarse accuracy to a final model with fine accuracy. The proposed technique does not require building a new equivalent circuit or empirical device model from scratch. Instead, it is an automated model enhancement process, which is realized by the automatic learning capability of the neural networks. An analytical formulation of the neuro-space mapping technique has been derived to achieve direct derivative information for model training, evaluation, and high-level circuit optimization. Subsequently, modeling

efficiency has been significantly improved. Our proposed approach has led to efficient model building, avoiding otherwise inefficient trial-and-error process in manual adjustment of equivalent circuit topology and nonlinear formulas. By mapping the existing equivalent circuit models to detailed device physics data, the Neuro-SM has also efficiently expanded the scope of models in existing circuit simulators to include device physics.

For the first time, a statistical space mapping concept has been introduced in this thesis for efficient statistical characterization of the large-signal behavior of a population of nonlinear microwave devices. The proposed statistical space-mapped model combines a large-signal nominal model and a space mapping network which characterizes the statistical variations around the nominal. With the assumption that the parameter variations of the given device population are usually small percentages of their nominal values, a simple linear dynamic mapping network can be extracted from small-signal data to approximate the large-signal statistical variations. Preliminary examples have demonstrated that the statistical space-mapped model can approximate large-signal statistical characteristics using only one set of large-signal data from the nominal device. Greatly reduced modeling cost has been obtained by the proposed technique, which allows large-signal statistical model development without performing massive large-signal data generation of many devices.

This thesis has also proposed an advance over the linear statistical space mapping technique, called statistical neuro-space mapping, where nonlinear mapping is used to overcome the accuracy limitations of the linear dynamic mapping in modeling large

statistical variations among different devices. The proposed Neuro-SM retains the advantages of the linear statistical space mapping of using a nonlinear nominal model to represent the average large-signal behavior of a given statistical population of devices. The behavior of a random device in the population is obtained by a nonlinear mapping from that of the nominal device. The unknown mapping function is represented by neural networks trained using dc and small-signal data of various devices in the population. A novel statistical mapping is formulated by introducing a compact set of statistical variables to control the mapping to map from the nominal device towards different devices in the population. With such neural-based mapping formulations, the increased complexity of the mapping, required for large variations, can be achieved without increasing the number of statistical parameters. A new training method has been proposed for simultaneous statistical parameter extraction and neural network training. The proposed statistical Neuro-SM with the proposed training has been demonstrated to outperform the existing methods for small or large statistical variations, using a minimal amount of expensive large-signal data to provide the most accurate large-signal statistical model. It is useful for efficient microwave circuit design involving highly repetitive computations such as design optimization, statistical design, and yield optimization.

## **6.2 Suggestions on Future Directions**

This thesis has proposed and demonstrated the neuro-space mapping concept for efficient modeling of various types of nonlinear microwave devices. Simulated data have been used throughout the verification of the proposed techniques.

One of the future directions is to apply the proposed techniques to the modeling and statistical modeling of high-power transistor devices using measurement data. Recently tremendous interest has been shown in high-power high-frequency electronic devices, whose nonlinear effects and thermal issues are crucial in transistor responses. New models for such devices accounting for these issues are necessary. However, modeling such devices is more challenging due to the increased device complexity and operating frequency. The existing modeling techniques are mostly problem/technology specific, and have certain disadvantages: the physical model may be slow to develop, and the equivalent circuit model may require trial-and-error efforts. As a flexible alternative to the existing techniques, our proposed Neuro-SM technique becomes a suitable candidate for such a modeling challenge, by utilizing the existing device models and using neural network mapping to efficiently accomplish the relationship between the behavior of existing models and that of the new device.

Another direction is to expand the static neural network mapping in the neuro-space mapping techniques of this thesis to dynamic neural network mapping, i.e., Neuro-SM with DNN or statistical Neuro-SM with DNN. The recent development of dynamic neural network techniques has led to great success in accurate and efficient nonlinear behavior modeling of microwave circuits and systems [49]. As device complexity and operating frequency increase, the nonlinearity in the device behavior becomes more dramatic. The dynamic information in an existing equivalent circuit model may not be adequate to accurately represent such nonlinearity. Instead of creating new device models, the DNN can be used to complement the missing dynamics of an existing equivalent circuit model.

This will provide more freedom in the neural-based device modeling to handle more complicated modeling scenarios such as modeling nonlinearity with internal states in intrinsic transistor models and thermal noise effects.

As further work, a generic and automated modeling methodology for semiconductor device modeling can be envisioned, including automatic data generation, automatic topology selection, automatic dynamic order selection, and automatic model modification, for complete automation of device model development. The automatic data generation will determine minimal measurements for device characterization, the automatic topology selection will inspect a list of existing device models to find the one with the closest behavior to that characterized by measurement, the automatic dynamic order selection will choose the most suitable DNN orders for accurate mapping of the input space, and the automatic model modification will be performed under an efficient training algorithm to adjust the DNN weighting parameters in order to accurately capture the nonlinear and thermal noise effects of the new devices.

Automated large-signal statistical modeling following these prospects will also gain attention. How to embed device physics into the automated modeling procedure is another interesting topic. Generic formulations will be required for convenient implementation of the developed model into modern commercial circuit simulators to update the user library of the new devices for use in microwave/millimeterwave circuit design, simulation, and optimization. User-friendly CAD software that can be directly incorporated into circuit simulators for automated model development will be required.

In conclusion, the Neuro-SM techniques proposed in this thesis have the potential of automating the model creation and updating process, contributing to increased efficiency in computer-aided design of microwave circuits and systems. They can be used to cover more varieties of modeling scenarios and applied to more varieties of devices. This work enables computer-based automatic enhancement of existing large-signal device models for efficient and automatic updating of nonlinear device model libraries.

## Bibliography

- [1] M. B. Steer, J. W. Bandler and C. M. Snowden, "Computer-aided design of RF and microwave circuits and systems," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 3, pp. 996-1005, Mar. 2002.
- [2] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA: Artech House, 2000.
- [3] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design: from theory to practice," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 4, pp. 1339-1350, Apr. 2003.
- [4] P. Burrascano, S. Fiori, and M. Mongiardo, "A review of artificial neural networks applications in microwave computer-aided design," *Int. J. RF Microwave CAE*, vol. 9, pp. 158-174, May 1999.
- [5] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microw. Theory Tech.*, vol. 43, no. 6, pp. 1349-1358, June 1995.
- [6] D.M.M.-P. Schreurs, J. Verspecht, S. Vandenberghe, and E. Vandamme, "Straightforward and accurate nonlinear device model parameter-estimation method based on vectorial large-signal measurements," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 10, pp. 2315-2319, Oct. 2002.

- [7] J. J. Xu, M. C. E. Yagoub, R. T. Ding, and Q. J. Zhang, "Exact adjoint sensitivity analysis for neural-based microwave modeling and design," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 1, pp. 226-237, Jan. 2003.
- [8] Y. H. Fang, M. C. E. Yagoub, F. Wang and Q. J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2335-2344, Dec. 2000.
- [9] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in multilayer circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 44, no. 12, pp. 2495-2503, Dec. 1996.
- [10] F. Wang and Q. J. Zhang, "Knowledge based neural models for microwave design," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 12, pp. 2333-2343, Dec. 1997.
- [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 and Microw. CAE*, vol. 9, no. 3, pp. 254-260, May 1999.
- [12] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sánchez and Q. J. Zhang, "Neuromodeling of microwave circuits exploiting space-mapping technology," *IEEE Trans. Microw. Theory Tech.*, vol. 47, no. 12, pp. 2417-2427, Dec. 1999.
- [13] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 12, pp. 2536-2544, Dec. 1994.

- [14] L. Zhang, J.J. Xu, M.C.E. Yagoub, R.T. Ding, and Q.J. Zhang, "Neuro-space mapping technique for nonlinear device modeling and large-signal simulation," *IEEE MTT-S Int. Microw. Symp. Dig.*, Philadelphia, PA, June 2003, pp. 173-176.
- [15] L. Zhang, J. Xu, M. C. E. Yagoub, R. T. Ding, and Q. J. Zhang, "Efficient analytical formulation and sensitivity analysis of neuro-space mapping for nonlinear microwave device modeling," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 9, pp. 2752-2767, Sept. 2005.
- [16] L. Zhang and Q. J. Zhang, "Neuro-space mapping technique for semiconductor device modeling," *Springer J. on Optimization and Engineering*, July 2007 (published online).
- [17] L. Zhang, K. Bo, Q. J. Zhang, and J. Wood, "Statistical space mapping approach for large-signal nonlinear device modeling," *Proc. IEEE 36<sup>th</sup> Eur. Microw. Conf.*, Manchester, U.K., Sept. 2006, pp. 676-679.
- [18] L. Zhang, Q. J. Zhang, and J. Wood, "Statistical Neuro-Space Mapping Technique for Large-Signal Modeling of Nonlinear Devices," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 11, Nov. 2008 (in press).
- [19] 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. Microw. Theory Tech.*, vol. 40, no. 7, pp. 1374-1400, July 1992.
- [20] J. W. Bandler, Q. J. Zhang, J. Song, and R. M. Biernacki, "FAST gradient based yield optimization of nonlinear circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 38, no. 11, pp. 1701-1710, Nov. 1990.

- [21] P. Cox, P. Yang, S. S. Mahant-Shetti, and P. Chatterjee, "Statistical modeling for efficient parametric yield estimation of MOS VLSI circuits," *IEEE Trans. Electron Devices*, vol. 32, no. 2, pp. 471-478, Feb. 1985.
- [22] M. Meehan and J. Purviance, *Yield and Reliability in Microwave Circuit and System Design*. Boston, MA: Artech, 1993.
- [23] X. Ding, J. Xu, M. C. E. Yagoub, and Q. J. Zhang, "A combined state space formulation/equivalent circuit and neural network technique for modeling of embedded passives in multilayer printed circuits," *J. of the Applied Computational Electromagnetics Society*, vol. 18, 2003.
- [24] X. Ding, V. K. Devabhaktuni, B. Chattaraj, M. C. E. Yagoub, M. Doe, J. J. Xu, and Q. J. Zhang, "Neural-network approaches to electromagnetic-based modeling of passive components and their applications to high-frequency and high-speed nonlinear circuit optimization," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 436-449, Jan. 2004.
- [25] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microw. Theory Tech.*, vol. 43, no. 6, pp. 1349-1358, June 1995.
- [26] R. Biernacki, J. W. Bandler, J. Song, and Q. J. Zhang, "Efficient quadratic approximation for statistical design," *IEEE Trans. Circuit Syst.*, vol. 36, no. 11, pp. 1449-1454, Nov. 1989.

- [27] P. Meijer, "Fast and smooth highly nonlinear multidimensional table models for device modeling," *IEEE Trans. Circuit Syst.*, vol. 37, no. 3, pp. 335-346, Mar. 1990.
- [28] Q. J. Zhang, F. Wang, and M. S. Nakhla, "Optimization of high-speed VLSI interconnects: A review," *Int. J. of Microwave and Millimeter-Wave CAD*, vol. 7, pp. 83-107, 1997.
- [29] 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, no. 5, pp. 794-802, May 1997.
- [30] V. B. Litovski, J. I. Radjenovic, Z. M. Mrcarica, and S. L. Milenkovic, "MOS transistor modeling using neural network," *Elect. Lett.*, vol. 28, no. 18, pp. 1766-1768, Aug. 1992.
- [31] 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. 28<sup>th</sup> European Microw. Conf.*, Amsterdam, Netherlands, Oct. 1998, pp. 306-311.
- [32] K. Shirakawa, M. Shimizu, N. Okubo, and Y. Daido, "Structural determination of multilayered large signal neural-network HEMT model," *IEEE Trans. Microw. Theory Tech.*, vol. 46, no. 10, pp. 1367-1375, Oct. 1998.
- [33] H. Sharma and Q. J. Zhang, "Transient electromagnetic modeling using recurrent neural networks," *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, June 2005, pp. 1597-1600.

- [34] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 4, pp. 867-879, Apr. 2008.
- [35] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 12, pp. 2515-2523, Dec. 1997.
- [36] C. Christodoulou, A. E. Zooghy, and M. Georgiopoulos, "Neural network processing for adaptive array antennas," *IEEE-APS Int. Symp.*, Orlando, FL, pp.2584-2587, July 1999.
- [37] 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. Microw. Theory Tech.*, vol. 45, no. 5, pp. 712-723, May 1997.
- [38] G. Kothapali, "Artificial neural networks as aids in circuit design," *Microelectronics J.*, vol.26, pp. 569-678, 1995.
- [39] J. E. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: the state-of-the-art," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 420-435, Jan. 2004.
- [40] Q. J. Zhang and M. S. Nakhla, "Signal integrity analysis and optimization of VLSI interconnects using neural network models," *IEEE Int. Circuits Syst. Symp.*, London, England, pp. 459-462, May 1994.

- [41] T. Hong, C. Wang, and N.G. Alexopoulos, "Microstrip circuit design using neural networks," *IEEE MTT-S Int. Microw. Symp. Dig.*, Atlanta, GA, June 1993, pp. 413-416.
- [42] M. D. Baker, C.D. Himmel, and G.S. May, "In-situ prediction of reactive ion etch endpoint using neural networks," *IEEE Trans. Components, Packaging, and manufacturing Tech. Part A*, vol.18, no. 3, pp. 478-483, Sept. 1995.
- [43] M. Vai and S. Prasad, "Neural networks in microwave circuit design – beyond black box models," *Int. J. RF and Microw. CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, pp. 187-197, 1999.
- [44] M. Vai and S. Prasad, "Automatic impedance matching with a neural network," *IEEE Microwave Guided Wave Letter*, vol. 3, pp. 353-354, 1993.
- [45] V. Rizzoli, A. Neri, D. Masotti, and A. Lipparini, "A new family of neural network-based bidirectional and dispersive behavioral models for nonlinear RF/microwave subsystems," *Int. J. RF Microw. Computer-Aided Eng.*, vol. 12, no. 1, pp. 51-70, Jan. 2002.
- [46] T. J. Liu, S. Boumaiza, and F. M. Ghannouchi, "Applications of neural networks to 3G power amplifier modeling," *Proc. of International Joint Conf. on Neural Networks*, Montreal, QC, Aug. 2005, pp. 2378-2382.
- [47] 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. Microw. Theory Tech.*, vol. 49, no. 12, pp. 2282-2291, Dec. 2001.

- [48] V. K. Devabhaktuni, B. Chattaraj, M. C. E. Yagoub, and Q. J. Zhang, "Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks and space mapping," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 7, pp. 1822-1833, July 2003.
- [49] J. J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, "Neural-based dynamic modeling of nonlinear microwave circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 12, pp. 2769-2780, Dec. 2002.
- [50] J. Wood, and D. Root, "The behavioral modeling of microwave/RFIC's using nonlinear time series analysis," *IEEE MTT-S Int. Microwave Symp. Dig.*, Philadelphia, PA, June 2003, pp. 791-794.
- [51] J. Wood, M. Lefevre, D. Runton, J. C. Nanan, B. H. Noori, and P. H. Aaen, "Envelop-domain time series (ET) behavioral model of a Doherty RF power amplifier for system design," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 8, pp. 3163-3172, Aug. 2006.
- [52] S. Mukherjee, B. Mutnury, S. Dalmia, and M. Swaminathan, "Layout-level synthesis of RF inductors and filters in LCP substrates for Wi-Fi applications," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 6, pp. 2196-2210, June 2005.
- [53] M. Isaksson, D. Wisell, and D. Rönnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 11, pp. 3422-3428, Nov. 2005.
- [54] J. P. García, F. Q. Pereira, D. C. Rebenaque, J. L. G. Tornero, and A. A. Melcón, "A neural-network method for the analysis of multilayered shielded microwave

- circuits,” *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 1, pp. 309-320, Jan. 2006.
- [55] Y. Wang, M. Yu, H. Kabir, and Q. J. Zhang, “Effective design of cross-coupled filter using neural networks and coupling matrix,” *IEEE MTT-S Int. Microwave Symp. Dig.*, San Francisco, CA, June 2006, pp. 1431-1434.
- [56] Y. Cao, R. Ding, and Q. J. Zhang, “State-space dynamic neural network technique for high-speed IC applications: modeling and stability analysis,” *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 6, pp. 2398-2409, June 2006.
- [57] L. Zhang, Y. Cao, S. Wan, H. Kabir, and Q. J. Zhang, “Parallel automatic model generation technique for microwave modeling,” *IEEE MTT-S Int. Microwave Symp. Dig.*, Honolulu, HI, June 2007, pp. 103-106.
- [58] J. Wood, D. E. Root, and N. B. Tuffillaro, “A behavioral modeling approach to nonlinear model-order reduction for RF/microwave ICs and systems,” *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 9, pp. 2274-2284, Sept. 2004.
- [59] V. Rizzoli, A. Costanzo, D. Masotti, P. Spadoni, and A. Neri, “Prediction of the end-to-end performance of a microwave/RF link by means of nonlinear/electromagnetic co-simulation,” *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 12, pp. 4149-4160, Dec. 2006.
- [60] J. W. Bandler, Q. S. Cheng, S. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen and J. Søndergaard, “Space mapping: the state of the art,” *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 337-361, Jan. 2004.

- [61] 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. Microw. Theory Tech.*, vol. 46, no. 12, pp. 2412-2425, Dec. 1998.
- [62] J. W. Bandler, N. Georgieva, M. A. Ismail, J. E. Rayas-Sánchez and Q. J. Zhang, "A generalized space-mapping tableau approach to device modeling," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 1, pp. 67-79, Jan. 2001.
- [63] J. W. Bandler, M. A. Ismail, and J. E. Rayas-Sánchez, "Expanded space-mapping EM-based design framework exploiting preassigned parameters," *IEEE Trans. Circuits and Systems I*, vol. 49, no. 12, pp. 1833-1838, Dec. 2002.
- [64] J. E. Rayas-Sánchez, F. Lara-Rojo, and E. Martínez-Guerrero, "A linear inverse space-mapping (LISM) algorithm to design linear and nonlinear RF and microwave circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 3, pp. 960-968, Mar. 2005.
- [65] J. E. Rayas-Sánchez and V. Gutiérrez-Ayala, "EM-based monte carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 12, pp. 4528-4537, Dec. 2006.
- [66] H. Taher, D. Schreurs, and B. Nauwelaers, "Extraction of small signal equivalent circuit model parameters for statistical modeling of HBT using artificial neural," *Proc. Eur. Gallium Arsenide and Other Semiconductor App. Symp.*, Paris, France, Oct. 2005, pp. 213-216.

- [67] J. M. Golio, Ed., *The RF and Microwave Handbook*. Boca Raton, FL: CRC Press, 2001.
- [68] J. M. Golio, Ed., *Microwave MESFET's & HEMT's*. Norwood, MA: Artech House, 1991.
- [69] C.M. Snowden, *Semiconductor Device Modeling*. Stevenage, UK: Peregrinus, 1988.
- [70] K. Lehovec and R. Zuleeg, "Voltage-current characteristics of GaAs JFET's in the hot electron range," *Solid State Electron.*, vol. 13, pp. 1415–1426, Oct. 1970.
- [71] P. H. Ladbrooke, *MMIC Design: GaAs FET's and HEMT's*. Norwood, MA: Artech House, 1989.
- [72] Q. Li and R. W. Dutton, "Numerical small-signal AC modeling of deep-level-trap related frequency dependent output conductance and capacitance for GaAs MESFET's on semi-insulating substrates," *IEEE Trans. Electron Devices*, vol. 38, no. 6, pp. 1285–1288, June 1991.
- [73] M. A. Khatibzadeh and R. J. Trew, "A large-signal analytical model for the GaAs MESFET," *IEEE Trans. Microw. Theory Tech.*, vol. 36, no. 2, pp. 231–238, Feb. 1988.
- [74] R. J. Trew, "MESFET models for microwave CAD applications," *Int. J. Microwave Millimeter-Wave Computer-Aided Eng.*, vol. 1, pp. 143–158, Apr. 1991.

- [75] C. M. Snowden and R. R. Pantoja, "Quasi-two-dimensional MESFET simulation for CAD," *IEEE Trans. Electron Devices*, vol. 36, no. 9, pp. 1564–1574, Sept. 1989.
- [76] T. R. Cook and J. Frey, "An efficient technique for two-dimensional simulation of velocity overshoot effects in Si and GaAs devices," *COMPEL—Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 1, no. 2, pp. 65, 1982.
- [77] C. G. Morton, J. S. Atherton, C. M. Snowden, R. D. Pollard, and M. J. Howes, "A large-signal physical HEMT model," *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, June 1996, pp. 1759–1762.
- [78] H. Statz, P. Newman, I. W. Smith, R. A. Pucel, and H. A. Haus, "GaAs FET device and circuit simulation in SPICE," *IEEE Trans. Electron Devices*, vol. 34, no. 2, pp. 160-169, Feb. 1987.
- [79] W. R. Curtice, "GaAs MESFET modeling and nonlinear CAD," *IEEE Trans. Microw. Theory Tech.*, vol. 36, no. 2, pp. 220-230, Feb. 1988.
- [80] A. Materka, and T. Kacprzak, "Computer calculation of large-signal GaAs FET amplifier characteristics," *IEEE Trans. Microw. Theory Tech.*, vol. 33, no. 2, pp. 129-135, Feb. 1985.
- [81] I. Angelov, H. Zirath, and N. Rorsman, "A new empirical nonlinear model for HEMT and MESFET devices," *IEEE Trans. Microw. Theory Tech.*, vol. 40, no. 12, pp. 2258-2266, Dec. 1992.

- [82] V. I. Cojocaru and T. J. Brazil, "A scalable general-purpose model for microwave FET's including the DC/AC dispersion effects," *IEEE Trans. Microwave Theory Tech.*, vol. 12, no. 12, pp. 2248–2255, Dec. 1997.
- [83] H. K. Gummel and H. C. Poon, "An integral charge-control relation for bipolar transistors," *Bell Syst. Techn. J.*, vol. 49, pp.115, May 1970.
- [84] C.M. Snowden, "Nonlinear modelling of power FET's and HBT's," *Int. J. Microwave and Millimeter-wave Computer-Aided Eng.*, vol. 6, pp. 219-233, 1996.
- [85] D. E. Root, S. Fan, and J. Meyer, "Technology independent large-signal non quasistatic FET models by direct construction from automatically characterized device data," in *Proc. IEEE 21<sup>st</sup> Eur. Microw. Conf.*, Stuttgart, Germany, Sept. 1991, pp. 927-932.
- [86] J. W. Bandler, R. M. Biernacki, S. H. Chen, J. F. Loman, M. L. Renault, and Q. J. Zhang, "Combined discrete/normal statistical modeling of microwave devices," *Proc. IEEE 19<sup>th</sup> Eur. Microwave Conf.*, London, U.K., September 1989, pp. 205-210.
- [87] J. W. Bandler, R. M. Biernacki, Q. Cai, and S. H. Chen, "A novel approach to statistical modeling using cumulative probability distribution fitting," *IEEE MTT-S Int. Microwave Symp. Dig.*, May 1994, pp. 385-388.
- [88] J. E. Purviance, M. C. Petzold, and C. Potratz, "A linear statistical FET model using principal component analysis," *IEEE Trans. Microw. Theory Tech.*, vol. 37, no. 9, pp. 1389-1394, Sept. 1989.

- [89] J. Carroll, K. Whelan, S. Prichett, and D. R. Bridges, "FET statistical modeling using parameter orthogonalization," *IEEE Trans. Microw. Theory Tech.*, vol. 44, no. 1, pp. 47-55, Jan. 1996.
- [90] J. F. Swidzinski and K. Chang, "Nonlinear statistical modeling and yield estimation technique for use in Monte Carlo simulations," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2316-2324, Dec. 2000.
- [91] A. D. Martino, P. Marietti, M. Olivieri, P. Tommasino, and A. Trifiletti, "Statistical nonlinear model of MESFET and HEMT devices," *IEE Proc. Circuits Devices Syst.*, vol. 150, no. 2, pp. 95-103, Apr. 2003.
- [92] W. Stiebler, F. Rose, and J. Selin, "Nonlinear statistical modeling of large-signal device behavior," *IEEE MTT-S Int. Microw. Symp. Dig.*, Phoenix, AZ, May 2001, pp. 2071-2074.
- [93] A. T. Basilevsky, *Statistical Factor Analysis and Related Methods: Theory and Applications* (Hardcover). New York, NY: Wiley, 1994.
- [94] G. W. Snedecor and W. G. Cochran, *Statistical Methods*. 8th ed. Chap. 15. Ames, IA: Iowa State University Press, 1991, pp. 282-296.
- [95] 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. Microw. Theory Tech.*, vol. 45, no. 10, pp. 1696-1704, Oct. 1997.
- [96] OSA90/hope v2.0, Optimization Systems Associates Inc., Dundas, ON, Canada.

- [97] S. Goasguen, S. M. Hammadi, and S. M. El-Ghazaly, "A global modeling approach using artificial neural network," *IEEE MTT-S Int. Microwave Symp. Dig.*, Anaheim, CA, June 1999, pp. 153-156.
- [98] B. Davis, C. White, M.A. Reece, M.E. Bayne, W.L. Thompson, N.L. Richardson, and L. Walker, "Dynamically configurable pHEMT model using neural networks for CAD," *IEEE MTT-S Int. Microwave Symp. Dig.*, Philadelphia, PA, June 2003, pp. 177-180.
- [99] K. Shirakawa, M. Shimizu, N. Okubo, and Y. Daido, "Structural determination of multilayered large signal neural network HEMT model," *IEEE Trans. Microw. Theory Tech.*, vol. 46, no. 10, pp. 1367-1375, Oct. 1998.
- [100] K. Shirakawa, M. Shimiz, N. Okubo, and Y. Daido, "A large signal characterization of an HEMT using a multilayered neural network," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 9, pp. 1630-1633, Sept. 1997.
- [101] P. J. C. Rodrigues, *Computer-Aided Analysis of Nonlinear Circuits*. Norwood, MA: Artech House, 1997.
- [102] M. S. Nakhla and J. Vlach, "A piecewise harmonic balance technique for determination of periodic response of nonlinear systems," *IEEE Trans. Circuits and Systems*, vol. 23, no. 2, pp. 85-91, Feb. 1976.
- [103] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrand Reinhold, 1994.

- [104] J. W. Bandler, Q. J. Zhang, and R. M. Biernacki, "A unified theory for frequency-domain simulation and sensitivity analysis of linear and nonlinear circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 36, no. 12, pp. 1661-1669, Dec. 1988.
- [105] C. N. Rheinfelder, F. J. Beibwanger, and W. Heinrich, "Nonlinear modeling of SiGe HBT's up to 50 GHz," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 12, pp. 2503-2508, Dec. 1997.
- [106] *Advanced Design System (ADS) 2006A*, Agilent Technologies, 395 Page Mill Road, Palo Alto, CA, U.S.A, 2006.
- [107] *NeuroModelerPlus v.2.0*, Q. J. Zhang, Department of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada, 2008.
- [108] C. Y. Chang and F. Kai, *GaAs High-Speed Devices: Physics, Technology, and Circuit Applications*. New York, NY: John Wiley & Sons, 1994.
- [109] *MINIMOS-NT release 2.0*, Institute for Microelectronics, Technical University, Vienna, Austria, 2003.
- [110] A. S. Yanev, B. N. Todorow, and V. Z. Ranev, "A broad-band balanced HEMT frequency doubler in uniplanar technology," *IEEE Trans. Microw. Theory Tech.*, vol. 46, no. 12, pp. 2032-2035, Dec. 1998.
- [111] A. O. Allen, *Probability, Statistics, and Queueing Theory: with Computer Science Applications*. 2nd ed. Chap. 8. Boston, MA: Academic Press, 1990, pp. 483-547.
- [112] *Matlab R2006b*, The MathWorks, Inc., Natick, MA, 2006.
- [113] *Synopsys Medici 2007A*, Synopsys, Inc., Mountain View, CA, 2007.

- [114] Agilent Angelov (Chalmers) Nonlinear GaAsFET Model, ADS 2006A Manual, "Nonlinear Devices," Chapter 3, Agilent Technologies, Inc., Palo Alto, CA, 2006.
- [115] *IC-CAP 2006B*, Agilent Technologies, Inc., Palo Alto, CA, 2006.