

# **Advanced Neural Network Modeling Techniques for Efficient CAD of Microwave Filters**

By

**Humayun Kabir, B.Sc. EEE, MSEE**

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

Ottawa-Carleton Institute for  
Electrical and Computer Engineering

Department of Electronics  
Carleton University  
Ottawa, Ontario, Canada  
September 2009

Copyright ©  
2009 – Humayun Kabir



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-60117-4  
*Our file* *Notre référence*  
ISBN: 978-0-494-60117-4

**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.

---

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

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

**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.

---

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

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

  
**Canada**

## **Abstract**

This thesis presents advanced neural network modeling techniques for computer aided design (CAD) of RF/microwave filters. The overall objective is to increase the efficiency of modeling and design. The first contribution is made by proposing a systematic neural network inverse modeling technique where the inputs to the inverse model are electrical parameters and outputs are geometrical parameters. Training the neural network inverse model directly may become difficult due to the non-uniqueness of the input-output relationship in the inverse model. We propose a new method to solve such problems by detecting multivalued solutions in training data. A comprehensive modeling methodology is proposed which utilizes various newly developed techniques such as detection of multivalued solution, derivative division, and submodel combining techniques to develop inverse models. Furthermore, a design methodology for microwave filters is presented using the inverse models. The methodology is validated by applying it to waveguide filter modeling and design. Full electromagnetic (EM) simulation and measurement results of a Ku-band circular waveguide dual-mode pseudo-elliptic bandpass filter are presented to demonstrate the efficiency of the proposed techniques.

The RF/microwave computer aided design is further enhanced by proposing a new method for high dimensional neural network modeling of microwave filters. Although neural network is useful for fast and accurate EM modeling, existing techniques are not

suitable for high dimensional modeling, because data generation and model training become too expensive. To overcome this limitation, we propose an efficient method for EM behavior modeling of microwave filters that have many design variables. Decomposition approach is used to simplify the overall high-dimensional modeling problem into a set of low-dimensional subproblems. We formulate a set of neural network submodels to learn filter subproblems. A method is then proposed to combine the submodels with filter empirical/equivalent model. An additional neural network mapping model is formulated and combined with the neural network submodels and empirical/equivalent model to produce the final overall filter model. Results of high dimensional model development show that the proposed method is advantageous over conventional neural network method and the resulting model is much faster than the EM model.

*To My Parents*

## **Acknowledgements**

I would like to express my sincere appreciation to my supervisor Dr. Qi-Jun Zhang, Professor of Department of Electronics, Carleton University, for his professional guidance, continued assistance, and supervision throughout the course of this work. I am grateful for giving me the opportunity to work on this research work. His in depth knowledge, high class expertise on computer aided design and modeling area, continuous encouragement, and inspiration motivated me to stay on course and led to a successful outcome of the research. Working as a PhD student with him, greatly enhanced my knowledge and expertise on computer aided design area. I have learnt in depth knowledge about research and development which I believe will help me in my future career.

I would like to thank Dr. Ming Yu, Director of R&D, COMDEV Ltd., for technical collaboration and providing support on this research work. I express gratitude for providing me the opportunity to work on the modeling project and onsite research opportunity at COMDEV Ltd. The opportunity provided me with confidence and motivation to produce high quality work. His assessment, guidance, and advice have contributed to this work significantly. I really appreciate for making the filter data available for the research and facilitate the filter fabrication and measurement for validating the proposed techniques.

Special thanks to Dr. Ying Wang of University of Ontario Institute of Technology for her technical collaboration and advice. Her expert advice on waveguide filters greatly assisted the research work. I am grateful for providing me time for discussions on many occasions. Her assistance in data generation for the filter models is also greatly appreciated.

I am also thankful to Faculty of Graduate Studies and Research and Department of Electronics for financial support in terms of scholarships and teaching assistantship. The experience in teaching provided me with great confidence which I believe would help me in my future career.

Many thanks to Blazenka Power, Peggy Piccolo, Jacques Lemieux, Scott Bruce and other DOE staff for offering a helpful and friendly yet professional service in the department.

# Table of Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Table of Contents</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Symbols</b>	<b>xvi</b>
<b>Nomenclature</b>	<b>xxvi</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Background .....	1
1.2 Motivations.....	3
1.3 Contributions of the Thesis .....	5
1.4 Thesis Organization.....	9
<b>Chapter 2: Literature Review</b>	<b>11</b>
2.1 Introduction .....	11
2.2 Neural Networks.....	12
2.2.1 Concept of Neural Network Model.....	12
2.2.2 Neural Network Structure .....	13
2.2.3 Neural Network Model Development.....	18
2.3 Neural Network Modeling for EM Applications .....	26

2.4 Neural Network Modeling for Microwave Filter .....	33
2.5 Summary .....	39
<b>Chapter 3: Neural Network Inverse Modeling and Applications to Microwave Filter Design</b>	<b>41</b>
3.1 Introduction .....	42
3.2 Inverse Modeling: Formulation and Proposed Neural Network Methods .....	45
3.2.1 Formulation .....	45
3.2.2 Non-Uniqueness of Input-Output Relationship in Inverse Model and Proposed Solutions .....	49
3.2.3 Proposed Method to Divide Training Data Containing Multivalued Solutions .....	52
3.2.4 Proposed Method to Combine the Inverse Sub-Models.....	54
3.2.5 Accuracy Enhancement of Sub-Model Combining Method .....	59
3.2.5.1 Competitively Trained Inverse Sub-Model.....	60
3.2.5.2 Forward Sub-Model .....	60
3.3 Overall Inverse Modeling Methodology .....	61
3.4 Examples and Applications to Filter Design.....	64
3.4.1 Example 1: Inverse Spiral Inductor Model .....	64
3.4.2 Example 2: Filter Design Approach and Development of Inverse Coupling Iris and IO Iris Models.....	67
3.4.3 Example 3: Inverse Tuning Screw Model.....	71
3.4.4 Example 4: A 4-pole Filter Design for Device Level Verification Using the Three Developed Inverse Models.....	77
3.4.5 Example 5: A 6-pole Filter Design for Device Level Verification of Proposed Methods .....	80
3.5 Additional Discussion on Examples.....	85
3.6 Summary .....	88

<b>Chapter 4: High Dimensional Neural Network Techniques and Application to Microwave Filter Modeling</b>	<b>89</b>
4.1 Introduction .....	90
4.2 Proposed High Dimensional Modeling Approach .....	92
4.2.1 Problem Statement .....	92
4.2.2 Neural Network Submodels .....	94
4.2.3 Integration of Neural Network Submodels with Empirical/Equivalent Circuit Model.....	96
4.2.4 Neural Network Mapping Model for Accuracy Improvement.....	97
4.2.5 Overall Modeling Structure.....	99
4.3 Algorithm for Proposed High Dimensional Model Development .....	104
4.4 Modeling Examples.....	107
4.4.1 Illustration of the Proposed Modeling Techniques for an H-Plane Filter ...	107
4.4.2 Development of a Side-Coupled Circular Waveguide Dual-Mode Filter Model with the Proposed High Dimensional Modeling Technique .....	116
4.5 Summary .....	127
<b>Chapter 5: Conclusion and Future Work</b>	<b>128</b>
5.1 Conclusion.....	128
5.2 Future Work .....	130
<b>Bibliography</b>	<b>133</b>

## List of Tables

Table 3.1: Comparison of model test errors between direct and proposed methods for tuning screw model.....	74
Table 3.2: Comparison of dimensions of the 4-pole filter obtained by the neural network inverse model and measurement.....	80
Table 3.3: Comparison of dimensions of the 4-pole filter obtained by the neural network inverse model and measurement.....	84
Table 3.4: Comparison of time to obtain the dimensions by neural network inverse models and EM models.....	85
Table 4.1: Comparison of test errors of 4-pole H-plane filter models developed using conventional and proposed high dimensional modeling approach.....	113
Table 4.2: Comparison of test error of side-coupled circular waveguide dual-mode filter models developed with conventional and proposed high dimensional modeling approach.....	121
Table 4.3: Comparison of CPU time of EM and neural network model of a side-coupled circular waveguide dual-mode filter.....	125

## List of Figures

Figure 2.1:	Diagram of an MLP neural network structure. An MLP is consists of one input layer, one or more hidden layer and one output layer.....	15
Figure 2.2:	Flowchart demonstrating major steps in neural network training, validation and testing [1].....	24
Figure 2.3:	Fast optimization process of a spiral inductor using neural network CAD technique. ....	28
Figure 3.1:	Example illustrating neural network forward and inverse models, (a) forward model (b) inverse model. The inputs $x_3$ and $x_4$ (output $y_2$ and $y_3$ ) of the forward model are swapped to the outputs (inputs) of the inverse model respectively.....	46
Figure 3.2:	Diagram of inverse sub-model combining technique after derivative division for a two sub-model system. Inverse sub-model 1, and inverse sub-model 2 in set (A) are competitively trained version of the inverse sub-models. Inverse sub-model 1 and inverse sub-model 2 in set (B) are trained with the divided data based on derivative criteria (3.15) – (3.16). The input and output of the overall combined model is $\bar{x}$ and $\bar{y}$ respectively.....	56
Figure 3.3:	Flow diagram of overall inverse modeling methodology consisting of direct, segmentation, derivative dividing and model combining techniques. ....	63
Figure 3.4:	Non-uniqueness of input-output relationship is observed when $Q_{eff}$ vs. $C_{D_i}$ data of a forward spiral inductor model is exchanged to formulate an inverse model. (a) Unique relationship between input and output of a forward model, (b) Non-unique relationship of input-output of an inverse model obtained from forward model of (a). Training data containing	

multivalued solutions of Figure 3.4(b) are divided into groups according to derivative. (c) Group I data with negative derivative, (d) group II data with positive derivative. Within each group, the data are free of multivalued solutions, and consequently the input-output relationship becomes unique.

	.....	66
Figure 3.5:	Comparison of inverse model using the proposed methodology and the direct inverse modeling method for the spiral inductor example.....	67
Figure 3.6:	Diagram of the filter design approach using the neural network inverse models. ....	68
Figure 3.7:	Original data showing variation of phase angle ( $P$ ) with respect to horizontal screw length ( $L_h$ ) describing unique relationship of forward tuning screw model. ....	75
Figure 3.8:	Comparison of output ( $L_h$ ) of inverse tuning screw model trained using direct and proposed methods at two different frequencies (a) $\omega_o = 10.8$ GHz, $C_D = 1.11$ inch (b) $\omega_o = 12.5$ GHz, $C_D = 1.11$ inch. It is evident that this inverse model has non-unique outputs. The proposed method produced more accurate inverse model than that of direct inverse method. Inverse data are plotted for two different diameters (c) $\omega_o = 11.85$ GHz, $C_D = 1.09$ and (d) $\omega_o = 11.85$ and $C_D = 0.95$ . Figure 3.8(c) contains multivalued data whereas 3.8(d) does not contain any multivalued data. This demonstrates the necessity of automatic algorithms to detect and handle multivalued scenarios in different regions of the modeling problem.....	76
Figure 3.9:	Training error of inverse tuning screw model following direct inverse modeling approach and proposed derivative division approach. The training errors of both the inverse sub-models are lower than that of direct inverse model.....	77

Figure 3.10: Comparison of the ideal 4-pole filter response with the measured filter response after tuning. The dimensions of the measured filter were obtained from neural network inverse models. ....	79
Figure 3.11: Picture of the 6-pole waveguide filter designed and fabricated using the proposed neural network method. ....	82
Figure 3.12: Comparison of the 6-pole filter response with ideal filter response. The filter was designed, fabricated, tuned and then measured to obtain the dimensions. ....	83
Figure 4.1: Diagram of the proposed high dimensional modeling structure. ....	100
Figure 4.2: Flow diagram of the proposed high dimensional neural network modeling approach. ....	106
Figure 4.3: Diagram of a 4-pole H-plane filter. The filter model holds eight input variables including five geometrical dimensions, bandwidth, center frequency, and sweeping frequency. ....	108
Figure 4.4: High dimensional modeling structure for the 4-Pole H-plane filter. Two neural network submodels: input-output iris model (IO iris) and coupling iris model (Co iris) are developed decomposing the filter. Five submodels required by the overall filter as shown in this figure are obtained by training only 2 neural network submodels. Equivalent circuit model of a filter are used to obtain the approximate S-parameter. A neural network mapping model is then used to obtain the accurate S-parameter of the 4-pole H-plane filter. ....	112
Figure 4.5: Comparison of approximate solution with EM- solution of a 4-pole H-plane filter. The approximate solution is obtained without using the mapping model of the proposed method. The similarity between the solutions confirms that a simple mapping using a few training data of overall filter can map the $y^a$ to accurate EM solution. Filter geometry: $L_{b1} = 0.54"$ , $L_{b2} = 0.60"$ , $W_1 = 0.37"$ , $W_2 = 0.23"$ , $W_3 = 0.21"$ , and $\omega_o = 12\text{GHz}$ . ....	114

Figure 4.6: Comparison of S-parameter of conventional neural network and proposed model of a 4-pole H-plane filter. (a) Filter geometry 1:  $L_{b1} = 0.52''$ ,  $L_{b2} = 0.58''$ ,  $W_1 = 0.38''$ ,  $W_2 = 0.25''$ ,  $W_3 = 0.22''$ ,  $\omega_o = 11.8$  GHz, (b) Filter geometry 2:  $L_{b1} = 0.54''$ ,  $L_{b2} = 0.60''$ ,  $W_1 = 0.37''$ ,  $W_2 = 0.23''$ ,  $W_3 = 0.21''$ ,  $\omega_o = 12$ GHz. Output of the conventional model is not accurate, because the amount of data used for training is not enough for the conventional method. However, the same data is enough for the proposed method..... 115

Figure 4.7: Diagram of a side-coupled circular waveguide dual-mode filter..... 116

Figure 4.8: Reflection coefficients of two different side-coupled circular waveguide dual-mode filters obtained using the proposed model. Geometry 1:  $B = 27$  MHz,  $\omega_o = 11.627$  GHz; Geometry 2:  $B = 35$  MHz,  $\omega_o = 11.627$  GHz... 122

Figure 4.9: Reflection coefficient of a side-coupled circular waveguide dual-mode filter with  $B = 54$  MHz,  $\omega_o = 11.627$  GHz showing the effectiveness of the neural network mapping in the coupling parameter space. .... 123

Figure 4.10: Comparison of average model test error vs. the number of filter geometry used for model training in conventional and proposed method of the side-coupled circular waveguide dual-mode filter..... 124

## List of Symbols

$B$	Bandwidth
$C_D$	The circular cavity diameter
$C_{D_i}$	Inner mean diameter of a spiral inductor
$\mathbf{d}_k$	An $m$ - vector representing the desired outputs of $k$ th sample of a neural network
$d_{jk}$	The $j$ th element of $\mathbf{d}_k$
$d_{\min,j}$ and $d_{\max,j}$	The minimum and maximum values of the $j$ th element of all $\mathbf{d}_k$
$\mathbf{d}_i^k$	The $k^{\text{th}}$ sample of the training data for output neurons which contains the EM solution of the $i^{\text{th}}$ substructure
$D_g$	Generated data
$D_r$	Training data set
$\mathbf{D}^k$	The $k^{\text{th}}$ sample of training data for output neurons and which is the EM solution of the overall filter
$E_i^s$	The training error of submodel $i$
$E^M$	Training error of neural network mapping model
$E_k$	Per sample error function

$E_p$	Error of inverse-forward submodel pair
$E_T$	A threshold value for $E_p$
$\hat{E}_{D_r}$	The normalized training error
$E_{D_r}(\mathbf{w})$	Training error of a neural network
$\hat{E}_v$	The normalized validation error
$f$	Input-output relationship of a neural network
$\bar{f}$	The input-output relationship of an inverse model
$f_i$	The geometrical to electrical relationship of the $i^{\text{th}}$ submodel
$f_M$	The input-output relationship of the mapping model
$f_q$	The empirical/equivalent circuit function
$G_{\min}^{(l)}$ and $G_{\max}^{(l)}$	Minimum and maximum slope between samples within the neighborhood of $\bar{\mathbf{x}}^{(l)}$
$G_M$	The maximum allowed slope between two samples within a neighbourhood
$G_R$	The maximum allowed slope change within a neighbourhood
$h$	Direction vector for neural network training
$H$	Substrate height

$I_x$	An index set containing the indices of inputs of forward model that are swapped
$I_y$	An index set containing the indices of outputs of forward model that are swapped
$I$	A $g \times g$ identity matrix
$J$	Number of layers of an MLP
$k$	Index of training samples
$L_S$	Spacing of spiral inductor
$L_C$	Coupling screw length
$L_h$	Horizontal tuning screw length
$L_t$	Microstrip length
$L_r$	The iris length of an IO iris model
$L_{r1}$ and $L_{r2}$	Lengths of input iris and output iris of side-coupled filter
$L_{11}$ , $L_{22}$ , and $L_{12}$	Screw lengths of tuning and coupling screw model
$L_{11b1}$ , $L_{22b1}$ , and $L_{12b1}$	Lengths of three screws of cavity 1 of a side-coupled filter
$L_{11b2}$ , $L_{22b2}$ , and $L_{12b2}$	Lengths of three screws of cavity 2 of a side-coupled filter
$L_{23}$	Length of the sequential coupling iris of a side-coupled filter
$L_{14}$	Length of cross coupling iris of a side-coupled filter
$L_{b1}$ and $L_{b2}$	Lengths of cavity 1 and cavity 2
$L_v$ and $L_h$	The vertical and horizontal coupling slot lengths
$m$	Number of outputs of a neural network

$M$	Ideal coupling matrix
$M^a$	A $g \times g$ approximate coupling matrix
$M_{11}$ to $M_{44}$	Self coupling bandwidths
$M_{12}, M_{23}, M_{34},$ and $M_{14}$	Sequential and cross-coupling bandwidths
$M_i^a$	Approximate values of coupling parameter of the $i^{\text{th}}$ submodel
$M_{ij}$	Coupling bandwidth, for $i \neq j$
$M_j$	$j^{\text{th}}$ coupling parameter
$M_j^a$	$j^{\text{th}}$ approximate coupling parameter obtained from neural network submodel
$n$	Number of inputs of a neural network
$N$	Number of inverse submodels divided from an inverse model
$N_{sub}$	Number of different types of substructures decomposed form an overall structure
$N_{D_r}$	The number of data samples in $D_r$
$N_l$	Number of neurons in layer $l$ of an MLP
$N_J$	Number of outputs in layer $J$
$N_o$	Number of neural network submodels needed to form the overall filter model

$N^c$	Number of samples of data of the overall filter required for the neural network model in the conventional approach
$N^M$	Number of samples of the overall filter required to train the mapping model accurately
$N_s$	The total number of training samples
$N_i^s$	The number of training samples required to develop neural network submodel $i$
$g$	Filter order
$p$	Index of inverse submodels
$P$	The phase shift of the vertical mode and that of the horizontal mode across the tuning screw
$P_{in}$	The phase loading on the input rectangular waveguide for IO iris
$P_v$ and $P_h$	Phases corresponding to the loading effect of the coupling iris on the two orthogonal modes
$P_i^a$	Approximate values of phase length of the $i^{\text{th}}$ submodel
$q$	Index of inverse submodels with values other than $p$
$Q_i$	A selection matrix containing 1s and 0s
$Q_{eff}$	Effective quality factor of a spiral inductor
$R$	The coupling bandwidth
$R_1$	The input coupling bandwidth

$R_2$	The output coupling bandwidth
$\mathbf{R}^a$	A $g \times g$ matrix
$R_1^a$ and $R_2^a$	Approximate values of input and output coupling parameters of a filter
$S_{11}^R$ and $S_{11}^I$	Real and imaginary part of $S_{11}$
$S_{12}^R$ and $S_{12}^I$	Real and imaginary part of $S_{12}$
$S_{11}, S_{12}$	S-parameter
$S_{11}^a$ and $S_{12}^a$	Approximate S-parameters
$t_o$	Data generation time per sample of an overall filter
$t_i$	Data generation time per sample for submodel $i$
$t_1, t_2,$ and $t_3$	Data generation time per sample of input-output iris, internal coupling iris, and coupling and tuning screw substructures
$T^c$	Cost of data generation in the conventional method
$T^p$	Cost of data generation in the proposed method
$u^{(k,l)}$	The distance between two samples of training data
$U_i^{(p)}$	Distance outside the training range of $i$ th output parameter of $p$ th inverse submodel
$U_p$	Total distance outside the training range for the $p$ th inverse submodel
$V_g$ and $V_d$	Gate voltage and drain voltage of a transistor

$\boldsymbol{w}$	Vector containing neural network weight parameters
$w_{ij}^l$	The weight of the link between the $j$ th neuron of the $(l-1)$ th layer and the $i$ th neuron of the $l$ th layer
$w_{i0}^l$	An additional weight parameter for each neuron for the $i$ th neuron of the $l$ th layer
$\boldsymbol{w}_M$	A vector containing internal weight parameters of neural network mapping model
$\boldsymbol{w}_{\text{now}}$	Present point of $\boldsymbol{w}$ during training process
$\boldsymbol{w}_{\text{next}}$	Next point of $\boldsymbol{w}$ during training process
$\Delta\boldsymbol{w}$	Gradient of weight vector
$\boldsymbol{w}_i$	A vector containing neural network weight parameters for the $i^{\text{th}}$ submodel
$W$	Microstrip width
$W_1, W_2, \text{ and } W_3$	Iris widths of an H-plane filter
$\boldsymbol{x}$	Input vector of a neural network
$\boldsymbol{x}_k$	An $n$ - vector representing the inputs of $k$ th sample of a neural network
$\tilde{\boldsymbol{x}}$	Vector containing input values of scaled data
$x_i$	The $i$ th external input to an MLP
$x_{\min}$ and $x_{\max}$	The minimum and maximum value of the input parameter space

$[\tilde{\mathbf{x}}_{\min}, \tilde{\mathbf{x}}_{\max}]$	The input parameter range of training data after scaling
$\tilde{x}$	A generic element in the vector $\tilde{\mathbf{x}}$
$\bar{\mathbf{x}}$	Vector of inputs of an inverse model
$\bar{x}_i$	Value of $i$ th input parameter of inverse model
$\tilde{\mathbf{x}}_i$	A vector containing the design variables of the $i^{\text{th}}$ substructure
$\tilde{\mathbf{x}}_i$	A vector containing the inputs of $i$ th submodel, which is a subset of the overall input vector $\mathbf{x}$
$\bar{x}_i^{\max}$ and $\bar{x}_i^{\min}$	The maximum and minimum value of $\bar{x}_i$
$\bar{x}_i^{(k)}$ and $\bar{y}_i^{(k)}$	Values of $\bar{x}_i$ and $\bar{y}_i$ in the $k^{\text{th}}$ training data
$\bar{\mathbf{x}}^{(l)}$	$l$ th sample of input parameters of inverse model
$\tilde{\mathbf{x}}_i^k$	The $k^{\text{th}}$ sample of the training data for input neurons of the $i^{\text{th}}$ submodel
$\mathbf{y}$	Output vector of a neural network
$y_j(\mathbf{x}_k, \mathbf{w})$	The $j$ th neural network output for input $\mathbf{x}_k$
$\bar{\mathbf{y}}$	Vectors of outputs of an inverse model
$\bar{y}_i$	Value of $i$ th output parameter of inverse model corresponding to $\bar{x}_i$
$\bar{y}_i^{\max}$ and $\bar{y}_i^{\min}$	The maximum and minimum value of $\bar{y}_i$

$y^a$	A vector containing approximate values of the outputs of the overall filter
$z_i$	A vector containing the output parameters of the $i^{\text{th}}$ substructure
$z_1$ to $z_{N_o}$	Electrical parameters obtained from $N_o$ submodels
$z_i^l$	The output of the $i$ th neuron of the $l$ th layer
$\alpha$	A user defined threshold value for $u^{(k,l)}$
$\beta$	Threshold of derivative to divide contradictory training data
$\epsilon_r$	Dielectric constant
$\gamma_i^l$	Weighted sum of a neuron
$\sigma(.)$	Activation function of a neuron
$\eta$	Neural network learning rate
$\delta_i^l$	The error between the $i$ th neural network output and the $i$ th output in the training data
$\delta_i^l$	The local error at the $i$ th neuron in the $l$ th layer
$\Gamma^c$	Model training time for conventional neural network approach
$\Gamma^p$	Model training time for the proposed neural network approach

$\lambda$	Normalized frequency for S-parameter calculation of filter
$\omega$	Frequency
$\omega_0$	Center frequency

## Nomenclature

3D	Three dimensional
ADS	Advanced design system
ANN	Artificial neural network
ARMA	Autoregressive moving-average
BP	Backpropagation
CAD	Computer-aided design
Co	Coupling
CPU	Central processing unit
CPW	Coplanar waveguide
DNN	Dynamic neural networks
DOE	Design of experiments
DR	Dielectric resonator
EBP	Error backpropagation
EM	Electromagnetic
FDTD	Finite difference time domain
FEM	Finite element method
GaAs	Gallium Arsenide

GSM	Generalized scattering matrix
IO	Input-output
L2	Least square
MEMS	Micro-electro-mechanical system
MLP	Multilayer perceptron
MoM	Method of moments
NISM	Neural inverse space mapping
NN	Neural network
PBG	Periodic band gap
Q	Quality factor
RBF	Radial basis function
RBF-NN	Radial basis function neural network
RF	Radio frequency
RNN	Recurrent neural networks
SFE	Segmentation finite element
SFNN	Sample function neural network
SPWL	Smooth piecewise linear
WNN	Wavelet neural network

# **Chapter 1: Introduction**

## **1.1 Background**

In the last decade, the RF/microwave circuits and their applications have grown rapidly. RF/microwave circuits find applications in many areas such as satellite and terrestrial communications, aircraft, warfare, many applications in our daily lives such as cell phones, vehicles, wireless devices, etc. The highly competitive microwave industry requires products with high functionality, better reliability, and lower cost. Additionally, short design cycle and lower time to market is desired for better cost benefit. For these reasons microwave circuit design and optimization has become more challenging than before. The computer aided design (CAD) has been utilized for design optimization of microwave circuits and structures for many years [1], [2]. These CAD tools are very useful to simulate structures and optimize circuit behaviour before building a prototype. This allows faster design cycle resulting in higher yield.

The success of CAD tools depends largely on models for circuit components that are used for simulation. Accuracy is one of the most important factors for a model. Simulation speed is another important factor which makes the CAD tools effective. Various models have been introduced for high frequency circuits including theoretical models, empirical/equivalent circuit models. Detailed theoretical models which are also

commonly known as electromagnetic (EM) models are developed based on the circuit theory and they can provide accuracy. However, the EM model becomes expensive especially when iterative design process is followed. The empirical/equivalent models are useful for fast estimation of the device behaviour but those are limited by the accuracy. For these reasons, there is a need for modeling techniques which can deliver accurate and fast models to meet the constant challenges for the RF/microwave design and optimization.

In recent years, neural network has been recognized as a useful alternative for model development where a mathematical model is not available. Neural network is an information processing system, which can learn from observation and generalize arbitrary multidimensional nonlinear input-output relationship [2]. The evaluation time of a neural network model is fast. Once developed, neural network model can be incorporated into CAD tools for fast simulation and optimization [3]. For these reasons, neural network model has been utilized for many areas of engineering and scientific applications such as biometric, remote sensing, communication, etc. Recently neural network has gained popularity in microwave modeling. Many results of microwave modeling has been reported which shows the benefits of this technique [4]-[12]. Introduction of neural network has enriched the microwave modeling and CAD area. More research on neural modeling techniques are needed to meet new challenges of microwave computer aided design.

## 1.2 Motivations

The drive for microwave circuit design at a large scale on regular basis demands good quality model. Quality of a model can be assessed against some criteria such as: accuracy, reliability, reusability, ease of incorporation in the existing CAD tools, ability to modification, and cost of model development [1]. A model should represent the original device behaviour accurately for the range of input variables it is developed for. For a first pass design success, an accurate model is a must. The model should be usable for various applications and should have the flexibility to make changes, improvement, and extend the operation range easily such that a model initially developed for a smaller range of variables can be modified to cover a larger range. The model should also be available for incorporating into commonly used simulation software. An important criterion is the cost for model development and operation. The model must be affordable to develop. However the model may become ineffective if it becomes CPU intensive to evaluate. The cost of design optimization increases significantly for using a CPU intensive model.

A high quality model should follow the above mentioned criteria for effective design optimization. There are a few choices of models available, but none of them can provide all the qualities together. One option is to develop empirical/analytical model. Analytical model can provide most of the criteria. However, it becomes challenging to develop an analytical circuit model quickly. Often they are developed based on certain assumptions and thus suffer from accuracy limitations. For this reason, computer aided design of microwave filters is done using electromagnetic (EM) -based model through classical

synthesis process. The purpose of a synthesis process is to find optimum values of the geometrical parameters for a specific electrical specification. The model is evaluated several times while varying the geometrical parameters until the optimum values of the parameters are found that matches the electrical specification. This process is time consuming and becomes too expensive specially using the EM. For these reasons, there is a constant demand for new modeling solutions which can deliver fast, and accurate solutions without the high cost of conventional models.

Recently, neural network models have been proven useful for microwave modeling including waveguide filters [4]-[12]. CAD techniques based on neural models have shown advantageous over the CAD techniques based on EM model. Conventionally, most neural models are developed formulating geometrical parameters as inputs to the model and response of the model such as S-parameters as outputs of the model [4]-[10]. The neural network model can provide solutions quickly for various values of geometrical input variables. Design synthesis is performed using several evaluations of the model which become inefficient. To overcome this limitation, we need a better method where repetitive model evaluations can be avoided unlike the conventional synthesis approach where multiple evaluations of the model are needed. For this purpose, we need new techniques to develop neural models that can provide geometric parameters for some given electrical parameters without repetitive model evaluations. This will improve the capability of microwave CAD by reducing design time.

Another strong motivation is that we need to find a realistic and effective way to develop models for microwave structures including filters that have many design

variables. Due to the increased complexity and variety of microwave structures, the number of design variables per structure is on the rise. Design optimization based on EM models is slow and expensive. Various modified neural network structures have been investigated for microwave modeling such as knowledge-based neural network [13], [14], modular neural network [15] for improving neural networks' learning capability. But none of the techniques is directly suitable to address the challenges of high dimensional neural network modeling. In order to develop an accurate neural network model that can represent EM behavior of filters over a range of values of geometrical variables, we need to provide EM data at sufficiently sampled points in the space of geometrical variables [2]. The amount of data required increases very fast with the number of input variables of the model. For this reason, data generation for a high dimensional RF/microwave structure becomes too expensive. The training time of a neural network using the massive data also becomes impractical. Therefore, we need a new method to develop high dimensional neural network which can be used without high cost of data generation and model training. Once the model is developed it can be used as a fast model alternative to the computationally intensive EM model.

### **1.3 Contributions of the Thesis**

The scope of this work is to develop fast and accurate modeling techniques for efficient computer aided design and optimization of complex RF/microwave structures including filters such that the use of computationally expensive EM models can be relaxed. Conventionally models are developed whose inputs are design parameters such as physical or geometrical dimensions of a structure. The outputs are electrical responses

such as S-parameters [4]-[10], [16]. Using a conventional model, we change the geometrical parameters and match the desired electrical response. This process becomes too expensive using EM based modeling and optimization method. A major contribution is made by proposing the neural network inverse modeling approach [17]-[22]. We develop neural inverse models whose inputs are electrical parameters and outputs are geometrical or physical parameters. From a given electrical response, we can obtain the corresponding design parameters without repetitive model evaluation. In this way, the cost of design and optimization is reduced. The important steps towards the goal of reducing the cost are: (a) develop neural network inverse models for RF/microwave structures, (b) develop design method using the neural network inverse models. We further improve the CAD of microwave structures by developing a new modeling method for filter structures that have many design variables [23]. The resulting models provide more accurate results than those developed using conventional approach. The models also become faster than the EM models with comparable accuracy. By accomplishing these objectives, neural network based computer aided design and optimization become more attractive and useful for designing complex nonlinear microwave structures. These techniques offer benefits of accuracy over conventional neural network method and speed over conventional EM-based design method.

The major contributions of this thesis towards making the microwave CAD more efficient are summarized as follows:

- (1) An efficient neural network inverse modeling technique is developed for microwave filters. A general formulation of the neural network inverse model

is proposed. Non-uniqueness of input-output relationship of neural network inverse model is addressed. A method is developed to check the existence of multivalued solutions in the training data of the inverse model. A method to divide the training data that contain multivalued solutions is developed. The inverse submodels are developed using the divided data. A method is then proposed to combine the submodels to form the complete inverse model. We also propose techniques to enhance the accuracy of model combining technique. A comprehensive modeling algorithm is presented to develop the inverse models efficiently. The algorithm increases efficiency by developing the inverse model following right steps in right order. The neural network inverse models developed using the proposed techniques provide much better accuracy than those developed using conventional direct modeling approach.

- (2) An inverse design approach is developed using the inverse neural network models. This approach uses electrical parameters obtained from design specification and provides design variables without having to use repetitive model evaluations. The proposed inverse modeling technique is illustrated by simple spiral inductor model development example. In order to validate the proposed techniques, we apply those to develop inverse cavity filter models where a 4-pole filter and a 6-pole filter are designed. In addition to the design, the 6-pole filter is fabricated. Results and comparisons are also presented to show the accuracy and effectiveness of the proposed techniques. The

proposed approach provides fast solution to a design problem with comparable accuracy of EM design approach.

- (3) An efficient method for high dimensional neural network modeling is proposed. Conventional neural network modeling is not suitable for modeling devices/structures that have many input variables. In the proposed method, we decompose the overall structure in to substructures. Neural network models are developed for each of the substructures. We propose a method to combine the empirical/equivalent circuit models with neural network models to produce an approximate solution of the overall structure. To improve the accuracy, we propose to use another neural network which maps the approximate solution to the accurate solution. The combined neural network submodels, equivalent circuit model and neural network mapping model forms the accurate model of the overall structure. An overall algorithm is proposed to develop high dimensional neural network models efficiently.
- (4) The proposed high dimensional modeling techniques are verified by developing models for complex microwave filters that have many design variables. Neural network models are developed using data from EM simulation. Data generation of an overall filter is expensive. Conventional neural network method requires many samples of an overall filter during training to achieve good accuracy, whereas the proposed method requires only a few samples of the overall filter to achieve the same. For this reason, the proposed method becomes significantly less expensive than the conventional

method. The evaluation time of the proposed neural network model is faster than that of the EM model. We describe the reason why the proposed method is more accurate and less expensive than the conventional neural network modeling approach to develop models with many input variables. The comparison results between the two methods are presented in this thesis.

## **1.4 Thesis Organization**

The rest of the thesis is organized as follows:

Chapter 2 provides an overview of neural network modeling. Various neural network structures and development of neural network model is presented. An overview of recent advances of neural network based EM modeling techniques is presented. State-of-the-art microwave filter modeling using neural network modeling techniques is also described.

In Chapter 3, the neural network inverse modeling technique is presented. First the formulation of the inverse model is proposed. Generation of training data for the inverse neural network is proposed. The problem of non-uniqueness of input-output relationship, which introduces the contradictory samples, is discussed. A method to check the existence of multivalued solutions in training data and a method to divide the data into groups are proposed. Then a method is proposed to combine the inverse sub-models to construct the overall inverse model. Additional accuracy enhancement techniques of the model combining method are also presented. A comprehensive methodology for neural network inverse modeling incorporating various steps is proposed. A microwave example illustrating the problem of non-uniqueness in inverse modeling, and its solution is presented. An overview of the methodology of designing waveguide filter using neural

network inverse models is presented. Development of neural network inverse models for various waveguide filter junctions is presented. The effectiveness and usefulness of the proposed modeling and design approach is validated by designing four-pole and six-pole filter design. Various results and statistics are presented to support the proposed inverse modeling techniques.

In Chapter 4, the high dimensional neural network modeling for microwave filters is proposed. Development of neural network submodels from an overall structure is presented. A method to combine submodels with equivalent circuit/empirical model is formulated. A method for developing neural network mapping model to produce accurate result is presented. A comprehensive algorithm to develop high dimensional neural network model is presented. The proposed method is validated with filter modeling examples. Results of H-plane filter are described to illustrate the proposed modeling technique. A high dimensional model for a side-coupled circular waveguide dual-mode filter which holds fifteen design variables is developed using the proposed technique to validate the proposed techniques. Results and comparisons are presented to show the accuracy and effectiveness of the proposed technique over conventional neural network modeling approach.

Finally, in Chapter 5, a summary of the thesis is presented highlighting the key contributions of the proposed neural network inverse and high dimensional modeling techniques. Future research directions in relations with the proposed techniques are also outlined.

## Chapter 2: Literature Review

### 2.1 Introduction

Artificial neural network (ANN) or simply neural network (NN) is an information processing system which was developed inspired by human brain function. It has the ability to learn from observation and generalize arbitrary input-output relationships [1]. It can be used to provide projections for new situations of interest and answer “what if” questions. Due to its learning and generalization ability, it has been used for many engineering and scientific applications such as pattern recognition [24], control systems [25], biomedical [26]-[27], remote sensing [28]-[29], etc.

Neural models are first trained with the data generated from the components or structures that they are built for. Once developed, they can be used for high level simulation where the components are replaced with their EM model [30]-[35]. The evaluation of a neural model is fast. As a result the circuit optimization using neural models becomes advantageous and computationally inexpensive. For these reasons, it has gained popularity in RF/microwave computer aided design area and has been proven to be very useful for modeling passive microwave structures and components [4], [36]-[37], library of model [38], vias and interconnects [39], coplanar wave-guide components [40], transistor modeling [41]-[43], noise modeling [44]-[46], electromagnetic CAD [47]-

[49], integrated circuits [50], [51], amplifiers [52], [53], microwave filters [7]-[10], [54], microwave optimization [55], [56], loaded cylindrical cavity [57], shielded microwave circuit [58], frequency selective surface [59], etc.

The application of neural network model enhances the design speed. Conventionally, the RF/microwave design is accomplished by using EM based model. The EM analysis tool can provide accurate result. However, the computational cost is high and in general evaluation is slow [3]. Neural network model thus becomes an attractive choice for RF/microwave design since its evaluation is fast. Neural models are generated from EM simulation data and thus are capable of providing the same accurate results as EM models.

## 2.2 Neural Networks

### 2.2.1 Concept of Neural Network Model

The input-output relationship of a device or structure is represented by a neural network. Let us assume  $\mathbf{x}$  is an  $n$ -vector containing external inputs to the neural network and  $\mathbf{y}$  is an  $m$ -vector containing all outputs from the output neurons, and  $\mathbf{w}$  is a vector containing all the weight parameters of various interconnections of the network. The variables of a device for example a microstrip line are represented by input neurons and the output response for example S-parameters are represented by output neurons. Then the input and output vector is derived as

$$\mathbf{x} = [L, W, H, \epsilon_r, \omega]^T \quad (2.1)$$

and

$$\mathbf{y} = [S_{11} \ S_{12}]^T \quad (2.2)$$

where  $L_t$  represents length,  $W$  is width,  $H$  is the substrate height,  $\epsilon_r$  is the dielectric constant,  $\omega$  is the frequency,  $S_{11}$  and  $S_{12}$  represent the S-parameters of the transmission line. The original physics based model is expressed as

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad (2.3)$$

where  $\mathbf{f}$  defines the physics based input-output relationship. The neural network model is defined as

$$\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{w}). \quad (2.4)$$

The neural network model can produce the same result after the learning process called training from the data generated from the original physics based device measurement or EM simulation.

### 2.2.2 Neural Network Structure

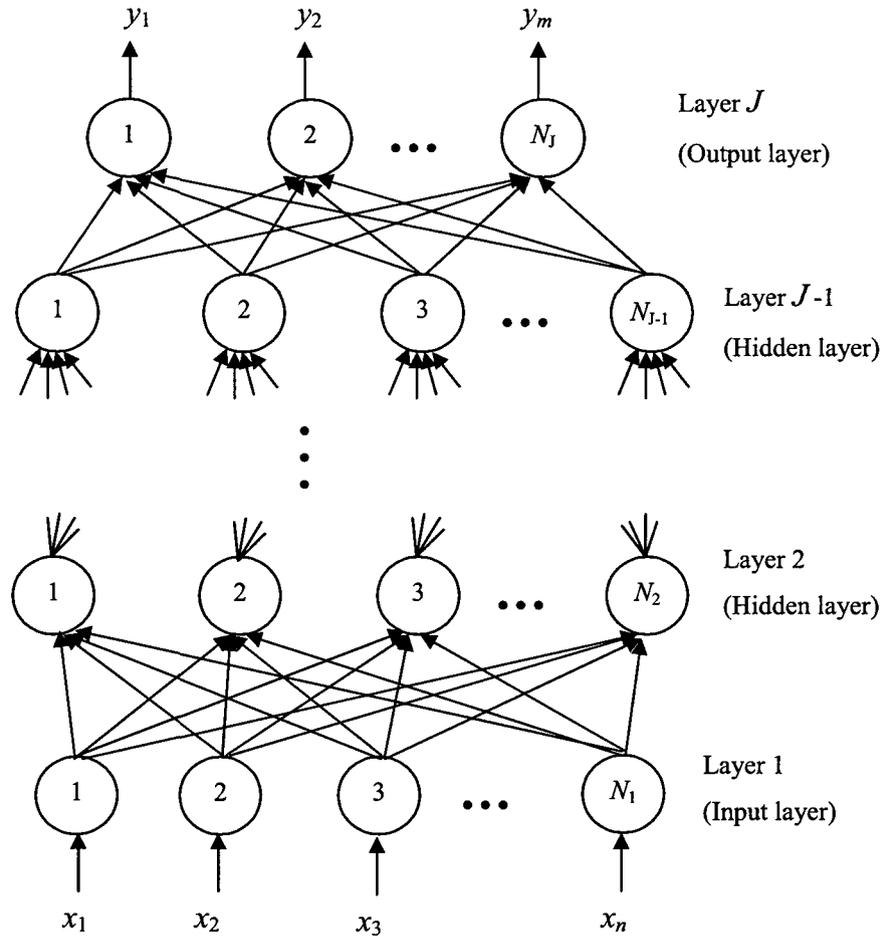
Multilayer perceptron (MLP) is a popular neural network structure [60]. The neurons are arranged in layers and thus neural network is known as multilayer perceptron neural network. There are three types of layers: (1) input layer, (2) output layer, and (3) hidden layer. The connections between neurons of different layers are known as links or synapses. Each neuron is associated with a weight parameter. The input neurons receive

stimuli from outside the network. The neurons of hidden layers receive the signal and compute responses and send the information to the output neurons. Thus the response of the model is determined by the inputs and weight parameter of the network. Figure 2.1 represents a diagram of an MLP. Let us assume that the MLP has  $J$  layers. Layer 1 is the input layer, Layer  $J$  is the output layer and middle layers are hidden layers. Let the number of neurons in the  $l$ th layer be  $N_l$ ,  $l = 1, 2, \dots, J$ . Let  $w_{ij}^l$  represent the weight of the link between the  $j$ th neuron of the  $(l - 1)$  th layer and the  $i$ th neuron of the  $l$ th layer. Let  $x_i$  represent the  $i$ th external input to the MLP and  $z_i^l$  be the output of the  $i$ th neuron of the  $l$ th layer. An additional weight parameter exists for each neuron ( $w_{i0}^l$ ) representing the bias for the  $i$ th neuron of the  $l$ th layer. Therefore,  $\mathbf{w}$  of the MLP includes  $w_{ij}^l, j = 0, 1, \dots, N_{l-1}, i = 1, 2, \dots, N_l$ , and  $l = 2, 3, \dots, J$ . Thus the weight vector of the MLP is expressed as follows [1]

$$\mathbf{w} = \left[ w_{10}^2 \ w_{11}^2 \ w_{12}^2 \ \dots \ w_{N_J N_{J-1}}^J \right]^T. \quad (2.5)$$

Each neuron processes the input information received from other neurons. This process is done through a function called the activation function of the neuron. The processed information becomes output of the neuron. A typical  $i$ th neuron in the  $l$ th layer processes this information in two steps. Firstly, each of the inputs is multiplied by the corresponding weight parameter and the products are added to produce a weighted sum  $\gamma_i^l$ , i.e., [61]

$$\gamma_i^l = \sum_{j=0}^{N_{l-1}} w_{ij}^l z_j^{l-1}. \quad (2.6)$$



**Figure 2.1: Diagram of an MLP neural network structure. An MLP is consists of one input layer, one or more hidden layer and one output layer.**

In order to create the effect of bias parameter  $w'_{i0}$ , we assume a fictitious neuron in the ( $l-1$ ) the layer whose output is  $z_0^{l-1} = 1$ . Secondly, the weighted sum of (2.6) is used to activate the neuron's activation function  $\sigma(\cdot)$  [62], [63] to produce the final output of the

neuron  $z_i^l = \sigma(\gamma_i^l)$ . This output can become the stimulus to neurons in the  $(l + 1)$ th layer.

The most common activation function for the hidden neurons is sigmoid function and is given by

$$\sigma(\gamma) = \frac{1}{1 + e^{-\gamma}}. \quad (2.7)$$

Arc tangent function, hyperbolic-tangent function are also used as activation functions.

Input neurons use relay activation function which only relay the input information to the hidden neurons. An output neuron computation is given by [2]

$$\sigma(\gamma_i^J) = \gamma_i^J = \sum_{j=0}^{N_{J-1}} w_{ij}^J z_j^{J-1}. \quad (2.8)$$

Neural network follows feedforward computation [61]. The external inputs  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$  are fed to the input neurons and the outputs of the input neurons are fed to the hidden neurons of the second layer. Continuing this way, the outputs of the  $(J - 1)$ th layer neurons are fed to the output neurons. During feedforward computation  $w$  remains fixed. The feedforward computation is expressed as [1]

$$z_i^1 = x_i, \quad i = 1, 2, \dots, N_1; \quad n = N_1 \quad (2.9)$$

$$z_i^l = \sigma \left( \sum_{j=0}^{N_{l-1}} w_{ij}^l z_j^{l-1} \right), \quad i = 1, 2, \dots, N_l; \quad l = 2, 3, \dots, J \quad (2.10)$$

$$y_i = z_i^J, \quad i = 1, 2, \dots, N_J; \quad m = N_J. \quad (2.11)$$

It is evident that the formulas in (2.9) to (2.11) are simpler to compute than solving the theoretical EM of physics equations. For this reason the neural network model becomes faster than EM model.

The theoretical basis of neural network to approximate arbitrary input-output relationship is based on universal approximation theory [64], which states that there always exists a three-layer MLP neural network that can approximate any arbitrary nonlinear continuous multidimensional function to any desired accuracy. Thus neural network has the ability to accurately relate geometrical variables of a RF/microwave structure/device to its electrical response. In order to model an  $x$ - $y$  relationship, neural network needs a suitable number of hidden neurons. The number depends on the degree of nonlinearity of the input output relationship  $f$ , and the dimensions of  $x$  and  $y$ . Highly nonlinear and high dimensional model requires many hidden neurons. The precise number of hidden neurons required for a given modeling task remains an open question. This number can be determined by automated trial and error process [39], [65]. The number of layers that should be used in the MLP structure is determined by the hierarchical information of the modeling problem. In general for RF/Microwave modeling problems, a three or four layer MLP [66] is commonly used.

In addition to the MLP, there are other neural network structures such as radial basis function (RBF) neural networks [67], wavelet networks [67], recurrent neural networks (RNN) [68], [69], dynamic neural networks (DNN) [70], etc. The selection of neural

network depends on the nature of x-y relationship. The most popular type is MLP since its training and structure is well established. For sharp variation in the x-y relationship, radial basis function (RBF) [71], [72] and wavelet [73] is suitable. For time domain modeling recurrent neural networks (RNN) and dynamic neural networks (DNN) are suitable. In addition to the basic neural network structures, there exist hybrid neural network structures which are known as knowledge based neural networks [13], [14], [38], [74]. It uses existing knowledge of empirical/equivalent circuits and combines with neural network for superior performances.

### **2.2.3 Neural Network Model Development**

In order to develop a neural network model, we need to train it with input-output data of the device/structure. The first step is to identify the inputs and outputs of the model. The input parameters are usually the device parameters such as physical and geometrical parameters of device along with frequency and other electrical parameters. For RF/microwave model, the outputs are usually S-parameters of the devices/structures [4]-[10]. The choice of inputs and outputs are selected based on the intention and purpose of the models. Other factors include ease of data generation, ease of incorporation into circuit simulator, etc.

The next step is to define the range of data to be used during neural network model training and distribute x-y samples within the range. Let  $x_{\min}$  and  $x_{\max}$  be the minimum and maximum value of the input parameter space. Training data is sampled little beyond this range to ensure reliability of the model. Once the range of input parameters are selected, a sampling distribution is chosen. Uniform grid distribution, non-uniform grid

distribution, design of experiments (DOE) methodology [74], star distribution [36], and random distribution are commonly used for sampling the input parameter space for data generation. In uniform grid distribution, each input parameter is sampled at uniform intervals. For example, in a transistor modeling problem where  $\mathbf{x} = [V_g \ V_d \ \omega]^T$  and the model is intended to be used for the range of

$$\begin{bmatrix} -2 \text{ V} \\ 0 \text{ V} \\ 1 \text{ GHz} \end{bmatrix} \leq \mathbf{x} \leq \begin{bmatrix} 0 \text{ V} \\ 10 \text{ V} \\ 20 \text{ GHz} \end{bmatrix}, \quad (2.12)$$

then training data can be generated for the range of

$$\begin{bmatrix} -2-0.2 \\ 0 \\ 1-0.5 \end{bmatrix} \leq \mathbf{x} \leq \begin{bmatrix} 0 \\ 10+1 \\ 20+2 \end{bmatrix}. \quad (2.13)$$

In non-uniform grid distribution, each input parameter is sampled at unequal intervals. This ploy is used for nonlinear modeling problems. Smaller steps are used for sampling the non-linear region and larger steps are used in the linear region. Sample distributions based on DOE (e.g.,  $2^n$  factorial experiment of design, central composite experimental design) and star distribution are used where the data generation is expensive.

Data can be generated either by using EM simulation such as HFSS or device measurement using network analyzer. Large number of samples should be generated for

nonlinear problems to obtain sufficient accuracy. The generated data is divided into training, validation, and testing sets. Training data is used during training process of the model. Validation data is used to monitor the quality of the neural network model during training and to determine the stop criteria of the training process. Test data is used to independently examine the final quality of the trained neural model in terms of accuracy and generalization capability. Ideally each data set should adequately represent the original component behaviour  $\mathbf{y} = \mathbf{f}(\mathbf{x})$ . Commonly intermediate points of training data sets are used for validation for better reliability of the model.

The generated data should be pre-processed before it can be used for model development. The orders of magnitude of various input and output parameter values in microwave applications can vary considerable from one another. For this reason scaling of the data is performed for efficient neural network training. Let  $x$ ,  $x_{\min}$ , and  $x_{\max}$  represent a generic input element in the vectors  $\mathbf{x}$ ,  $\mathbf{x}_{\min}$  and  $\mathbf{x}_{\max}$  of original generated data respectively. Let  $\tilde{x}$ ,  $\tilde{x}_{\min}$ , and  $\tilde{x}_{\max}$  represent a generic element in the vectors  $\tilde{\mathbf{x}}$ ,  $\tilde{\mathbf{x}}_{\min}$ , and  $\tilde{\mathbf{x}}_{\max}$  of scaled data where  $[\tilde{x}_{\min}, \tilde{x}_{\max}]$  is the input parameter range after scaling. Linear scaling is given by [1]

$$\tilde{x} = \tilde{x}_{\min} + \frac{x - x_{\min}}{x_{\max} - x_{\min}} (\tilde{x}_{\max} - \tilde{x}_{\min}) \quad (2.14)$$

and corresponding de-scaling is given by

$$x = x_{\min} + \frac{\tilde{x} - \tilde{x}_{\min}}{\tilde{x}_{\max} - \tilde{x}_{\min}} (x_{\max} - x_{\min}). \quad (2.15)$$

Output parameter in training data can also be scaled in a similar way. Another scaling method is the logarithmic scaling [2], which can be applied to outputs with large variations in order to provide balance between small and large values of the same output. At the end of this step, the scaled data is ready to be used for neural network model training.

The next step we prepare neural network for training. The neural network weight vector  $\mathbf{w}$  is initialized to provide a good starting point for training. Commonly the weight vector is initialized with random small values, e.g., [-0.5, 0.5]. In order to improve the convergence of training, Gaussian distribution, different ranges and different variances for the random number generators can be used [75]. The training data consists of sample pairs  $\{(\mathbf{x}_k, \mathbf{d}_k) \text{ and } k \in D_r\}$ , where  $\mathbf{x}_k$ ,  $\mathbf{d}_k$  are  $n$ - and  $m$ - vectors representing the inputs and desired outputs of the neural network,  $D_r$  is the training data set. The training error of the neural network is defined as

$$E_{D_r}(\mathbf{w}) = \frac{1}{2} \sum_{k \in D_r} \sum_{j=1}^m |y_j(\mathbf{x}_k, \mathbf{w}) - d_{jk}|^2 \quad (2.16)$$

where  $d_{jk}$  is the  $j$ th element of  $\mathbf{d}_k$  and  $y_j(\mathbf{x}_k, \mathbf{w})$  is the  $j$ th neural network output for input  $\mathbf{x}_k$ . During neural network training, the  $\mathbf{w}$  is adjusted such that the error function

$E_{D_r}(\mathbf{w})$  is minimized. Since  $E_{D_r}(\mathbf{w})$  is a nonlinear function of  $\mathbf{w}$ , iterative training techniques are used to update  $\mathbf{w}$  based on error information  $E_{D_r}(\mathbf{w})$  and error derivative information  $\partial E_{D_r} / \partial \mathbf{w}$ . The subsequent point in  $\mathbf{w}$ -space denoted as  $\mathbf{w}_{\text{next}}$  is determined by a step down from the current point  $\mathbf{w}_{\text{now}}$  along a direction vector  $\mathbf{h}$ , i.e.,  $\mathbf{w}_{\text{next}} = \mathbf{w}_{\text{now}} + \eta \mathbf{h}$ . Here,  $\Delta \mathbf{w} = \eta \mathbf{h}$  is called the weight update and  $\eta$  is a positive step size known as the learning rate. As an example backpropagation (BP) training algorithm [67] updates  $\mathbf{w}$  along the negative direction of the gradient of training error as  $\mathbf{w} = \mathbf{w} - \eta (\partial E_{D_r} / \partial \mathbf{w})$ . The computation of the derivatives is done using a standard approach known as error back propagation (EBP) [76]. The EBP is described as follows:

Let us define a per-sample error function  $E_k$  given by [1]

$$E_k = \frac{1}{2} \sum_{j=1}^m (y_j(\mathbf{x}_k, \mathbf{w}) - d_{jk})^2 \quad (2.17)$$

For the  $k$ th data sample  $k \in D_r$ . Let  $\delta_i^J$  represent the error between the  $i$ th neural network output and the  $i$ th output in the training data, i.e.,

$$\delta_i^J = y_i(\mathbf{x}_k, \mathbf{w}) - d_{ik}. \quad (2.18)$$

Starting from the output layer, this error can be backpropagated to the hidden layers as [76]

$$\delta_i^l = \left( \sum_{j=1}^{N_{l+1}} \delta_j^{l+1} w_{ji}^{l+1} \right) z_i^l (1 - z_i^l), \quad (2.19)$$

$$l = J-1, J-2, \dots, 3, 2$$

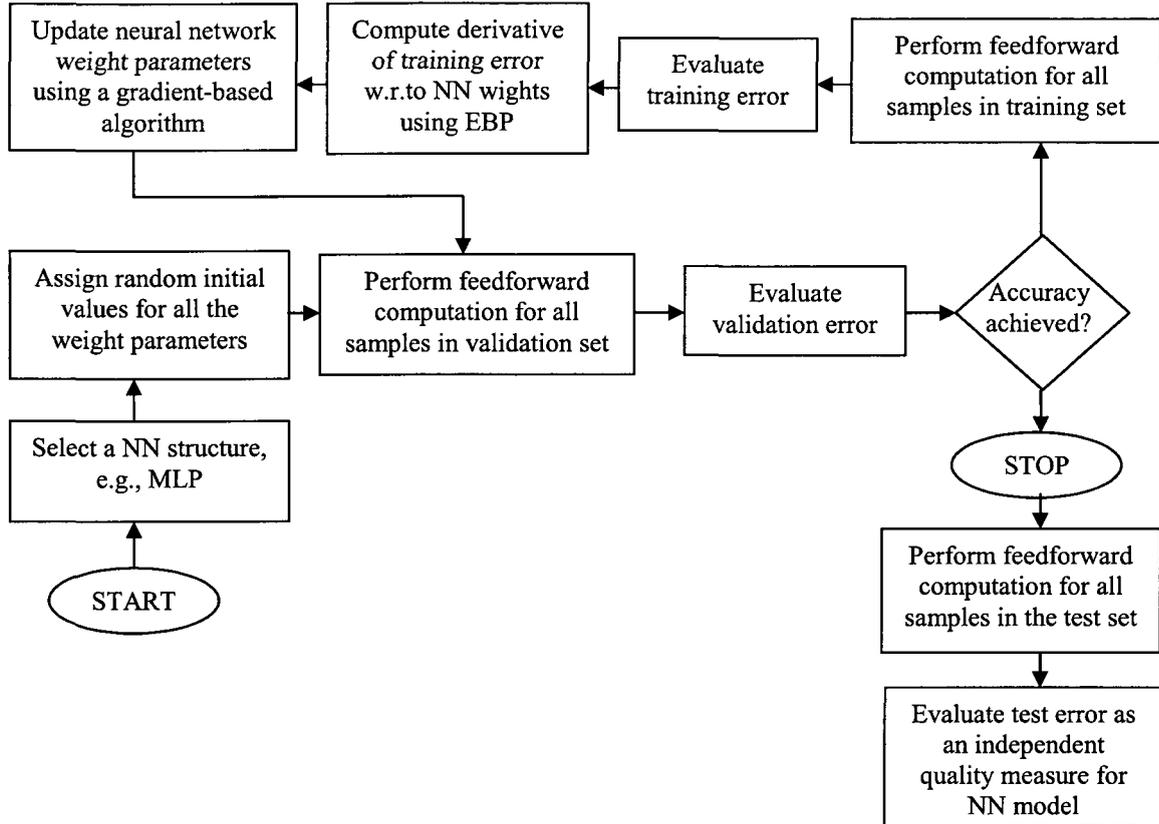
where  $\delta_i^l$  represents the local error at the  $i$ th neuron in the  $l$ th layer. The derivative of the per-sample error in (2.17) with respect to a given neural network weight parameter  $w_{ij}^l$  is given by [1]

$$\frac{\partial E_k}{\partial w_{ij}^l} = \delta_i^l z_j^{l-1} \quad l = J, J-1, \dots, 2. \quad (2.20)$$

Finally, the derivative of the training error in (2.16) with respect to  $w_{ij}^l$  can be computed as [2]

$$\frac{\partial E_{D_r}}{\partial w_{ij}^l} = \sum_{k \in D_r} \frac{\partial E_k}{\partial w_{ij}^l}. \quad (2.21)$$

Using EBP,  $(\partial E_{D_r} / \partial w_{ij}^l)$  can be systematically evaluated for the MLP neural network structure and can be provided to gradient based training algorithms for the determination of weight update  $\Delta w$ . A flowchart of neural network training and testing is presented in Figure 2.2.



**Figure 2.2: Flowchart demonstrating major steps in neural network training, validation and testing [1].**

Gradient based optimization method [77] such as backpropagation (BP), conjugate gradient, quasi-Newton is used for neural network training. Global optimization methods such as simulated annealing [78], genetic algorithms [79] can also be used for globally optimal solutions of neural network weights. However the training time required for global optimization method is much longer than that of gradient-based training techniques. Recently a combined global/local optimization is proposed for fast global training of neural networks [80]. The training process can be categorized into sample-by-

sample training and batch-mode training. The first case is known as online training [81] in which  $\mathbf{w}$  is updated each time a training sample is presented to the network. The second case is known as offline training [82] in which  $\mathbf{w}$  is updated after each epoch, where epoch is defined as a stage of the training process that involves presentation of all training data to the neural network once. In RF/microwave modeling, batch-mode is usually more effective.

The ability of a neural network to estimate output  $y_k$  accurately when presented with input  $\mathbf{x}_k$  never seen during training (i.e.,  $k \in D_r$ ) is called generalization ability. The normalized training error is defined as [1]

$$\hat{E}_{D_r}(\mathbf{w}) = \left[ \frac{1}{mN_{D_r}} \sum_{k \in D_r} \sum_{j=1}^m \left| \frac{y_j(\mathbf{x}_k, \mathbf{w}) - d_{jk}}{d_{\max,j} - d_{\min,j}} \right|^2 \right]^{1/2} \quad (2.22)$$

where  $d_{\min,j}$  and  $d_{\max,j}$  are the minimum and maximum values of the  $j$ th element of all  $\mathbf{d}_k$ ,  $k \in D_g$ ,  $D_g$  is the available (generated) data, and  $N_{D_r}$  is the number of data samples in  $D_r$ .

The normalized validation error  $\hat{E}_v$  can be similarly defined. Good learning of a neural network is achieved when both  $\hat{E}_{D_r}$  and  $\hat{E}_v$  have small values, e.g., 0.5% and are close to each other. Over-learning is phenomenon when neural network memorizes the training data but can not generalize well, i.e.,  $\hat{E}_{D_r}$  is small but  $\hat{E}_v \gg \hat{E}_{D_r}$ . When over-learning happens, deleting a certain number of hidden neurons or adding more samples to the training would improve the result. Under-learning is a phenomenon when neural network

find difficulty to learn the training data, i.e.,  $\hat{E}_D \gg 0$ . Possible remedies of under-learning are: 1) adding more hidden neurons or 2) perturbing the current solution  $\boldsymbol{w}$  to escape from a local minimum of  $\hat{E}_D(\boldsymbol{w})$ , and then continuing training. A robust training algorithm has been presented in [39], which is very useful for automatic model generation with minimum human supervision. More recently a parallel automatic model generation technique has been developed. It takes advantage of the multiprocessor system of modern computer technology [83]. This parallel automated model generation algorithm significantly reduces model development cost.

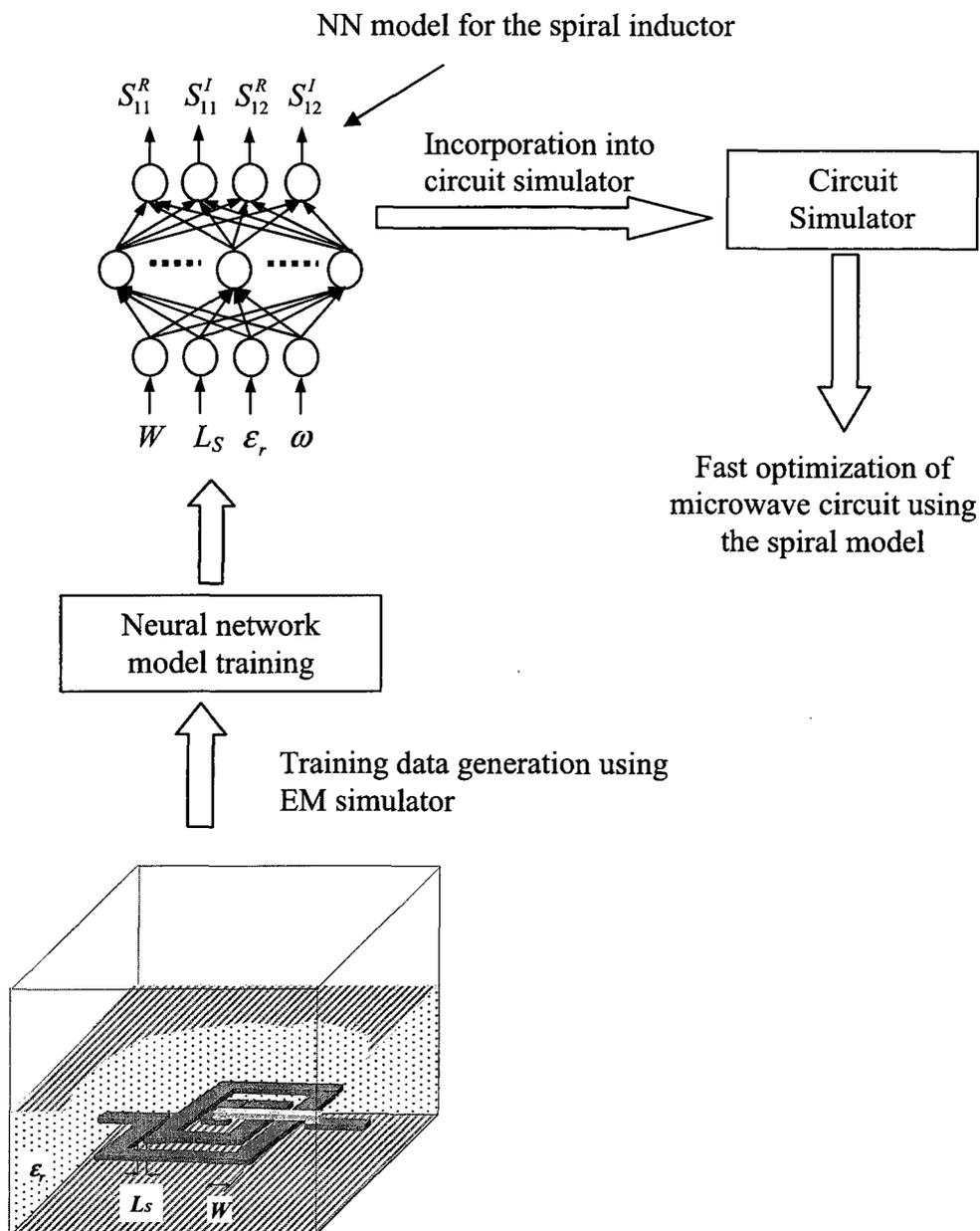
### **2.3 Neural Network Modeling for EM Applications**

For a first pass design success, an accurate model is essential. Accurate solutions can be obtained using electromagnetic simulations. However, the electromagnetic simulation is expensive for its high computational cost [68], [70]. Therefore, a neural network model becomes very useful especially when several model evaluations are required during design and optimization. Neural network model is developed from EM simulation or real device measurement data. For this reason, it can provide solution as accurate as electromagnetic solution [4], [74]. Once the model is developed, it can be incorporated with a circuit simulator for fast and accurate system level simulation and optimization [32]-[35]. In this Section, we review various neural network techniques and their applications in modeling, simulation, design, and optimization of electromagnetic components and structures.

Neural network has been used for passive component modeling. Inputs of models are physical or geometrical parameters such as length, width for a transmission line model. The outputs of the model are electrical parameters such as S-parameter [14], [30]. Many results of passive component modeling using neural network have been reported such as high-speed interconnects [74], CPW components [4], [84] [85], coupler [86], via [33], [34], etc. In [84], models for coplanar waveguide (CPW) components are developed using neural network technique. The inputs of the neural network are the geometrical parameters of the CPW and frequency. The outputs are S-parameters. Training data were generated using EM solver. Similarly in [4], neural network models for transmission line, 90° bends, short-circuit stubs, open-circuit stubs, step-in-width discontinuities and symmetric T-junctions are developed. The trained neural network models represent EM behaviors of respective components. Recently, a combined transfer function and neural network approach is presented in [34]. The via model can become highly nonlinear and the input-output relationship may become very difficult to learn for a neural network. The combined transfer function and neural network concept reduces the learning task significantly as the neural network is trained to learn geometrical parameters to coefficients of a transfer function.

Computer aided design using artificial neural network has become a popular and efficient method for design and optimization of electromagnetic structures [4], [33], [35], [74], [87]-[92]. The general idea of neural network based computer aided design is that we develop neural network models for electromagnetic structures and incorporate the models in a circuit simulator. This allows circuit level simulation speed with

electromagnetic level accuracy. Figure 2.3 illustrates an example of neural network based modeling and optimization of a spiral inductor.



**Figure 2.3: Fast optimization process of a spiral inductor using neural network CAD technique.**

Training data for the neural network model is first generated using EM simulation by varying the width  $W$ , spacing  $L_S$ , dielectric constant  $\epsilon_r$ , and frequency  $\omega$ . A neural network model is then trained using the data. The model is then incorporated into a circuit simulator for fast optimization of a circuit that uses a spiral inductor. Note that the model is developed once and then reused for many different circuit optimizations.

Another example of computer aided design and optimization using neural network is [74], where EM-neural network models for microstrip vias and interconnects are developed. The training data for the neural network models are generated using EM simulations of the vias and interconnects. Once trained, the accurate neural network models are inserted into commercially available microwave circuit simulator. The circuit simulator can provide results faster than that of the EM simulator with comparable accuracy. According to [4], simulation time of a GaAs via using HP-Momentum is 12.48 minutes whereas the simulation time of the via using the proposed method is only 0.3 sec. Similarly, the neural network models that are developed in [4] are incorporated into commercial microwave circuit simulator, HP-MDS. A CPW 50 ohm 3-db power divider is designed and optimized using the neural network models. Optimization time for the power divider circuit is only 2 minutes compared to the EM simulation time of 11 h. A similar EM-neural network model was developed for overlapping open ends in multilayer microstrip lines [30]. The neural network model is then used for design of bandpass filters. The design of gaps using neural network models requires approximately 1 second, whereas without using neural network model the design of coupling gap requires 84 minutes. Similar work has been presented in [31],[87] where neural network models are

developed for embedded passives. All these results show the advantage of using neural network model for design optimization over EM model. The use of neural network reduces the massive computational cost required by the EM analysis tools. It also provides comparable accuracy with EM solution.

Segmentation has been utilized in CAD algorithm [5], [53], [88]. A structure with many design variables requires many training data which may limit learning capability of neural network. In [53] and [88], segmentation is applied to divide the structure into several segments. For each segment, corresponding generalized scattering matrix is computed applying finite element method. Neural network models are then developed for each segment and the models are incorporated in circuit simulator for fast optimization. By using this approach, dielectric resonator filter is designed faster than classical EM optimization method.

Parasitic extraction of interconnects using neural networks is presented in [89]. The models are developed using the EM data of a set of passive interconnect structures. The neural network models improve the parasitic extraction process significantly. In [90], spiral inductor is modeled using neural network where geometrical parameters are taken as inputs, and inductance, quality factor, and resonant frequency are taken as outputs of the neural network model. Particle swarm optimization combined with neural network model for inductor is used to generate multiple sets of layouts that provide the right amount of target inductance with different values of quality factor and resonant frequency. The synthesis process using this method becomes much faster than that using EM simulation.

A computer aided design method for RF micro-electro-mechanical system (MEMS) switches is presented in [91]. Training data characterizing the switch is generated by using finite element method simulation. The developed neural network model is then used to perform circuit level simulation. This method provides fast design optimization.

Neural network models have been used in design optimization of antennas [93]-[96]. In [93], the input resistance of the antenna is first parameterized by a Gaussian model and neural network model is developed to approximate the nonlinear relationship between the antenna geometry and model parameters. The neural network model is incorporated with a genetic algorithm to optimize the antenna structure. In [96], neural network is used for synthesis of microstrip antenna structures. In all cases, the use of neural network models speeds up the optimization process compared to the EM optimization method.

Neural network models have been developed mostly in frequency domain. Time domain formulation of neural network has also been presented through recurrent neural network (RNN) model in [97] and [98]. Formulation is presented utilizing transient responses of the structure to the excitation signals. The training data is generated from time domain EM simulator. The RNN model can be incorporated for transient analysis of EM structure at a circuit simulation speed. RNN model has also been used for modeling interference of internal circuits of electronic devices [99]. Dynamic neural network (DNN) has been presented in [70] which describe continuous time domain behavior modeling of nonlinear microwave devices. DNN retains or enhances the neural modeling speed and accuracy capabilities, and provides additional flexibility in handling diverse needs of nonlinear microwave simulations, e.g., time- and frequency-domain

applications, single-tone and multitone simulations. Recent developments in dynamic modeling techniques shows superior modeling ability and more powerful applications in dynamic behavioral modeling [100], [101].

Neural network has been utilized for speeding up numerical techniques MoM [102] FDTD [103], FEM [104], space mapping [36], [105]. The combined method takes advantage of the high fast speed of neural network and performs a subtask of the overall computation. In [102], radial basis function neural network is used to fill the coupling matrix of the method of moments (MoM). For efficient numerical computation, the matrix fill time is considered as the key issue in the method of moment. The inputs of the neural networks are the distance between a test and a basis function and the angle between the lines connected the centers and a reference line. The outputs are the real and imaginary parts of the weighted functions. This method is used into compute the response of patch antennas. The result shows that the use of neural network speeds up the computation significantly.

Recently, a new EM-field based neural network technique has been developed as described in [106]. Usually neural network models are developed based on S-parameters of external input-output ports. If an electromagnetic 3D-structure is decomposed into substructures and the substructures are modeled based on the S-parameter, the result of the overall circuit simulation using the substructure models becomes inaccurate. The EM-field based model provides much better accuracy than conventionally developed neural network model based on S-parameters of external ports. Also, the proposed neural network models for substructures can be reused as a part of different circuit simulations.

## 2.4 Neural Network Modeling for Microwave Filter

Microwave filters are widely used in satellite and ground based communication systems. The full wave EM solvers have been utilized to design these kinds of filters for a long time. Usually several simulations are required to meet the filter specifications which takes considerable amount of time. In order to achieve first pass success with only minor tuning and adjustment in the manufacturing process, precise electromagnetic modeling is an essential condition. The design procedure usually involves iterating the design parameters until the final filter response is realized. The whole process needs to be repeated even with a slight change in any of the design specifications. The modeling time increases as the filter order increases. With the increasing complexity of wireless and satellite communication hardware, there is a need for faster method to design this kind of filters. Artificial neural network (ANN) has been proven to be a fast and effective means of modeling complex electromagnetic devices. Neural network modeling techniques for EM modeling and optimization have been discussed in the previous Section; this Section reviews the neural techniques dealing with microwave filters.

Waveguide cavity filters are very popular in microwave applications. Several results have been reported using neural network techniques to model cavity filters including E-plane metal-insert filter [107], rectangular waveguide H-plane iris bandpass filter [108]-[109], dual mode pseudo elliptic filter [17], cylindrical posts in waveguide filter [110], combline filter [111], etc. The simplest form of modeling is the direct approach where the geometrical parameters are related to its frequency response. Response of a filter is sampled at different frequency points to generate the training data. Result shows that

ANN can provide accurate design parameters and after learning phase the computational cost is lower than the one associated with full wave model analysis [107]. In a similar work the performance of filter obtained from the ANN was much better than obtained from parametric curve and faster than finite element method (FEM) analysis [108].

Simpler structure or lower order filter is feasible to realize the whole model in a single neural network model. For higher order filter several assumptions and simplifications are required to lower the number of neural network inputs. Filter can be modeled by segmentation finite element (SFE) method and using ANN [7]. Filter structure was segmented into small regions connected by arbitrary cross section and then the smaller sections are analyzed separately. The generalized scattering matrix (GSM) was computed by FEM and the response of the complete circuit was obtained by connecting the smaller sections in proper order. In general the optimization of microwave circuits is time consuming. To attain a circuit response by analytical method is too slow. Therefore, ANN based analytical models were used. The method was applied to a three-cavity filter. The response of the filter rigorously found from SFE was compared with the same response obtained from the GSMs of the irises computed from ANN and excellent agreement was observed. In similar approach smooth piecewise linear (SPWL) neural network model can be utilized for design and optimization of microwave filter [8]. SPWL has the advantage of smooth transitions between linear regions through the use of logarithm of hyperbolic cosine function. This feature suits well for the inductive iris modeling. A rectangular waveguide inductive iris band pass filter was modeled using SPWL neural network model. Several multi section Chebyshev band pass filters in

different bands have been tested and each showed very good agreement with full 3D electromagnetic solution. Again using the neural network model speeds up the design process significantly.

Waveguide dual-mode pseudo-elliptic filters are often used in satellite applications due to their high Q, compact size and sharp selectivity [112]. Recently neural network modeling technique has been applied to design wave-guide dual-mode pseudo-elliptic filter [17]. The coupling mechanism for dual mode filters is complex in nature and the numbers of variables are quite high. This makes the data generation and neural network training an overwhelmingly time-consuming job. Therefore, filter structures were decomposed into different modules each representing different coupling mechanism. This ensures faster data generation, neural network training and better accuracy. This model may be applied to filter with any number of poles as long as the filter structure remains the same. Due to the coupling between orthogonal modes, GSM of the discontinuity junctions in the filter is necessary to characterize most of the modules. Equivalent circuit parameters such as coupling values and insertion phase lengths were extracted from EM data first. Neural network models were then developed for the circuit parameters instead of EM parameters. The method was applied to a four pole filter with 2 transmission zeros. The filter was decomposed into three modules: input-output coupling iris, internal coupling iris and tuning screw. Neural network models were developed for each module and irises and tuning screw dimensions were calculated using the trained neural network models. The dimensions found from the neural network models are within 1% of the ideal ones.

The other popular type of microwave filters is built in planar configuration such as microstrip and strip line. Numerous works have been published modeling microwave filters using ANN including low pass microstrip step filter [5], coupled microstrip band pass filter [24], [113]-[120], microstrip band rejection filter [121], coplanar waveguide low pass filter [82], etc. The trained neural networks become fast filter model so that a designer can get the parameters quickly by avoiding long EM simulations. Wide bandwidth band pass filters were designed using microstrip line coupling at the end [24]. Coupling gaps are critical for designing these kinds of filters and the optimization of gaps require significant amount of time. To speed up the optimization of coupling gaps ANN models were developed and these models were used to design a filter. For a given filter specifications, physical parameters were obtained using ANN models. With these physical dimensions the filter was analyzed using a circuit simulator. A significant improvement in terms of speed has been realized using ANN models. The method can be generalized for low-pass, high pass, band pass or band rejection filters using planar configuration. A little modification is needed if the structure of the filter is changed from microstrip to strip line, but the general process remains the same. ANN models can be developed to model the entire filter if the number of variables is kept low. For larger dimensions some parameters are kept constant to keep the model simple.

Multi-layer asymmetric coupled microstrip line has been modeled using ANN [114]. The ANN replaces the time-consuming optimization routines to determine the physical geometry of multi-conductor multi-layer coupled line sections. ANN models for both synthesis and analysis were developed. The methodology was applied to a two layer

coupled line filter and compared with segmentation and boundary element method (SBEM). Circuit elements were obtained much faster by ANN models than the optimization method. Circuit parameters can also be used as modeling parameters for this kind of filter. For all these cases ANN models are capable of predicting the dimensions or circuit parameters accurately compared to that obtained from the analytical formulas.

Microstrip filter on PBG structure were also designed using neural network models [115]. A new neural network function called sample function neural network (SFNN) was employed for the modeling purpose. The PBG structures are periodic structures that are characterized by the prohibition of electromagnetic wave propagation at some microwave frequencies. A 2 dimensional square lattice consisting of circular holes were considered as the modeling problem. Radius of the circle of the periodic holes and frequency was input and s-parameters were considered as output of the neural network. Regular MLP was unable to converge to right solutions. RBF and wavelet functions improved the result but not accurate enough. Due to these reasons a new activation function called the sample activation function were used. The result shows that the SFNN can produce complex input-output relationship and could model the PBG filters on microstrip circuits accurately.

Neural network has been combined with some other optimization process in order to achieve fast filter design parameters. A design technique combining finite-difference-time domain (FDTD) and neural network was proposed [117]. Two-stage time reduction was realized by utilizing an ARMA signal estimation technique to reduce the computation time of each FDTD run and then the number of FDTD simulations was

decreased using a neural network as a device model. The neural network maps geometrical parameters to autoregressive moving-average (ARMA) coefficients. The trained network was incorporated with an optimization procedure for a microstrip filter design and significant time saving was achieved. Different algorithms can be developed combining neural network and optimization method for faster and accurate filter solution. A Neuro-genetic algorithm was developed for microwave filter [118]. Neural network models were combined with genetic algorithms to synthesize millimeter wave devices. The method has been used to synthesize low pass and band pass filters in microstrip configuration. While the method worked well for low pass filters it showed limited accuracy for band pass filter. In order to overcome the problem some modification is required in the layout and design space.

Wavelet neural network (WNN) [120] and radial basis function (RBF) [5] can be advantageous for some special applications. Wavelet radical and the entire network construction have a reliable theory, which can avoid the fanaticism of network structure like back propagation (BP) neural network. Also it can radically avoid the non-linear optimization issue such as local most optimized during the network training and have strong function study and extend ability. For these qualities WNN was chosen in [120]. Microstrip band pass filter was optimized where the geometrical parameters were changed to obtain the desired output response. The result was compared with that obtained using ADS optimizer. Fast and accurate results were obtained. In a similar work, radial basis function neural networks (RBF-NN) were used to model microstrip filter.

Segmentation of the structure was employed for a 13 sections microwave step filter. Using the RBF-NN shows much faster and better accurate result than full wave analysis.

Neural network also finds applications in the design of microwave filters consisting of dielectric resonator [56]. A rigorous and accurate EM analysis of the device was performed with FEM and combined with a fast analytical model. The analytical model was derived using segmented EM analysis applying to neural network. The method was then applied to dielectric resonator (DR) filters and good agreement between theoretical and experimental result was achieved within a few iterations.

Neural network has been employed to obtain starting point for optimizer used for yield prediction algorithm [122]. The yield was computed as a ratio of the number of cases passing the specification to the total number of simulations performed. For efficient calculation of yield, the choice of starting point is critical. It requires the knowledge of final solution, which is not available. Neural network was used to predict this solution and then the solution was used as the starting point of the optimization. Different structures realizing the same response was used to calculate the yield. Result suggests that by using neural network models, computational effort can be reduced significantly.

## **2.5 Summary**

In this Chapter, neural network modeling and its use in computer aided design and optimization of various applications have been reviewed. Neural network structures have been briefly described. The neural network model development and training methods have been described summarizing various steps including neural network formulation, data generation and processing, model training and testing. After reviewing neural

network model development, recent advances of microwave modeling and optimization techniques using neural networks have been presented. Following the review of neural network modeling in EM applications, the role of neural network in microwave filter modeling, optimization and design has also been reviewed. The ANN method has provided fast and accurate results and reduced the computational costs associated with a time consuming EM solver in the design of microwave filters.

## **Chapter 3: Neural Network Inverse Modeling and Applications to Microwave Filter Design**

In this Chapter, one of the major contributions of this thesis is presented. A systematic neural network inverse modeling approach is proposed. Various new techniques are proposed to develop neural network inverse models. We address the issue of multivalued solutions which introduces contradictions in the training data and propose mathematical criteria to detect the contradictory data. If the contradictory data exists, the proposed method divides data based on derivatives of the input parameters of the model. Several inverse submodels are then trained accurately with the divided training data. A method is proposed to combine the accurate inverse sub-models and thus obtain the overall accurate inverse model. This approach solves an important issue of inverse modeling in neural network. Without the data pre-processing and proposed technique, conventional approach will yield inaccurate inverse models. Furthermore, a comprehensive algorithm is presented to develop the inverse model combining the various techniques. This algorithm increases the efficiency by using the techniques in the right order. Another important contribution of this thesis is presented in this Chapter. A method is developed to design waveguide filters using the inverse neural network models. Waveguide filters are designed and fabricated using the proposed inverse approach. The filter dimensions are

obtained faster than conventional EM-based design approach and thus inverse approach speeds up the design process significantly.

### **3.1 Introduction**

In recent years, neural network techniques have been recognized as a powerful tool for microwave design and modeling problems [1]-[11]. A neural network trained to model original EM problems can be called the forward model where the model inputs are physical or geometrical parameters and outputs are electrical parameters. For the design purpose, the information is often processed in the reverse direction in order to find the geometrical/physical parameters for given values of electrical parameters, which is called the inverse problem. There are two methods to solve the inverse problem, i.e., optimization method and direct inverse modeling method. In the optimization method, the EM simulator or the forward model is evaluated repetitively in order to find the optimal solutions of the geometrical parameters that can lead to a good match between modeled and specified electrical parameters. An example of such an approach is [123]. This method of inverse modeling is also known as synthesis method.

The formula for the inverse problem, i.e., compute the geometrical parameters from given electrical parameters, is difficult to find analytically. Therefore, the neural network becomes a logical choice since it can be trained to learn from the data of the inverse problem. We define the input neurons of a neural network to be the electrical parameters of the modeling problem and the output neurons as the geometrical

parameters. Training data for the neural network inverse model can be obtained simply by swapping the input and output data used to train the forward model. This method is called the direct inverse modeling and an example of this approach is [124]. Once training is completed, the direct inverse model can provide inverse solutions immediately unlike the optimization method where repetitive forward model evaluations are required. Therefore, the direct inverse model is faster than the optimization method using either the EM or the neural network forward model. A similar concept has been utilized in neural inverse space mapping (NISM) technique where the inverse of the mapping from the fine to the coarse model parameter spaces is exploited in a space-mapping algorithm [125].

Though the neural network inverse model can provide the solution faster than the optimization method, it often encounters the problem of non-uniqueness in the input-output relationship. It also causes difficulties during training, because the same input values to the inverse model will have different values at the output (multivalued solutions). Consequently, the neural network inverse model cannot be trained accurately. This is why training an inverse model may become more challenging than training a forward model.

This Chapter considers application of neural network inverse modeling techniques for microwave filter design. Some results have been reported using neural network techniques to model microwave filters including rectangular waveguide iris bandpass filter [7] [8] [108], low pass microstrip step filter [5], E-plane metal-insert filter [107], coupled microstrip line band pass filter [10], etc. Waveguide dual-mode pseudo-elliptic filters are often used in satellite applications due to its high Q, compact size, and

sharp selectivity [112]. This particular filter holds complex characteristics whose conventional design procedure follows an iterative approach, which is time consuming. Moreover, the whole process has to be repeated even with a slight change in any of the design specifications. The modeling time increases as the filter order increases. Recently the neural network modeling technique has been applied to design wave-guide dual-mode pseudo-elliptic filter [17]. By applying neural network technique, filter design parameters were generated hundreds of times faster than EM-based models while retaining comparable accuracy.

In this Chapter, a new and systematic neural network inverse modeling methodology is developed, and the problem of non-uniqueness in inverse modeling is formally addressed. The proposed methodology uses a set of novel criteria to detect multivalued solutions in training data, and uses adjoint neural network [45] derivative information to separate training data into groups, overcoming non-uniqueness problems in inverse models in a systematic way. Each group of data is used to train a separate inverse sub-model. Such inverse sub-models become more accurate since the individual groups of data do not have the problem of multivalued solutions. A complete methodology to solve the inverse modeling problem efficiently is proposed by combining various techniques including the direct inverse modeling, segmenting the inverse model, identifying multivalued solutions, dividing training data that have multivalued solutions, and combining separately trained inverse sub-models. A significant step is achieved where two actual filters are made following the neural network solutions, and real

measurements from the filters are used to compare and validate the proposed neural network solutions.

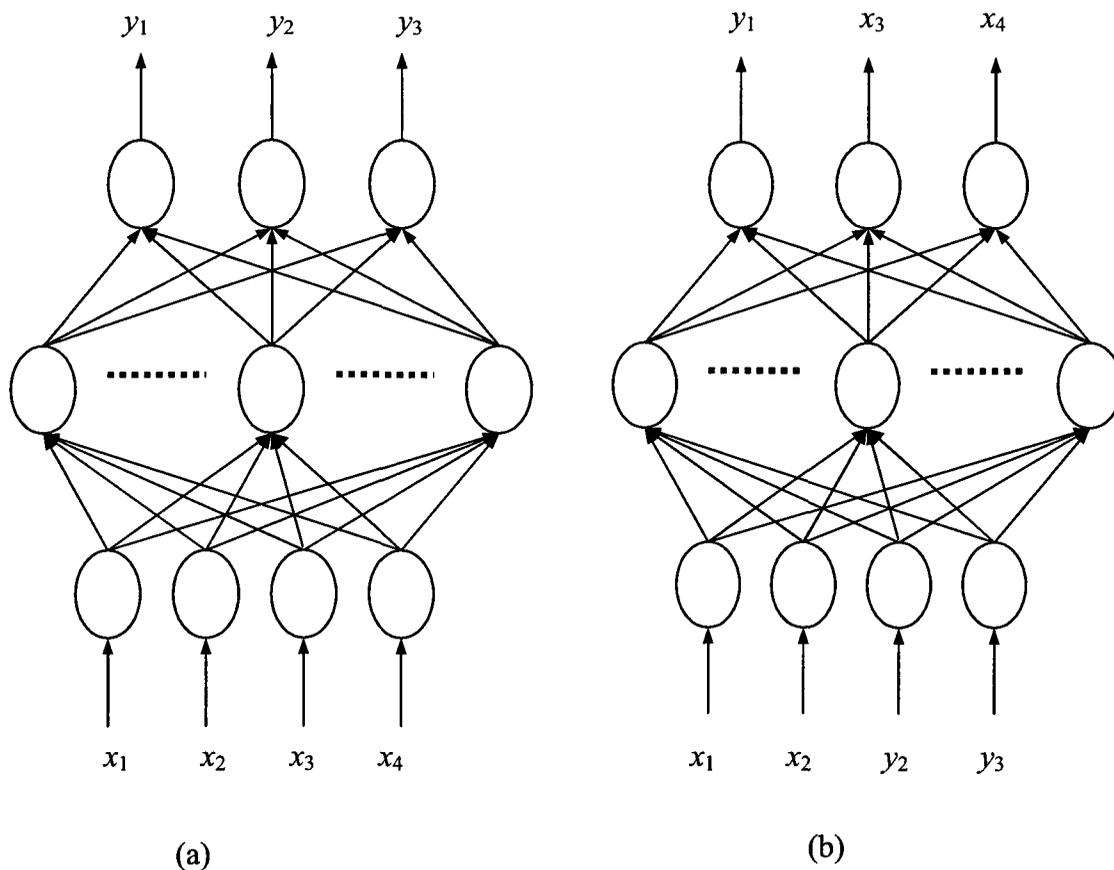
## 3.2 Inverse Modeling: Formulation and Proposed Neural Network Methods

### 3.2.1 Formulation

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

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad (3.1)$$

where  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T$ ,  $\mathbf{y} = [y_1 \ y_2 \ y_3 \ \dots \ y_m]^T$ , and  $\mathbf{f}$  defines the input-output relationship. An example of a neural network diagram of a forward model and its corresponding inverse model is shown in Figure 3.1. Note that, 2 outputs and 2 inputs of the forward model are swapped to the input and output of the inverse model respectively. In general, some or all of them can be swapped from input to output or vice versa. If we swap more inputs with less outputs of the forward model to formulate the inverse model, it might increase the possibility of non-uniqueness (to be described in the next sub-Section) of input-output relationship of the inverse model. The selection of which parameters to be swapped is a problem of specific task and mainly depends on the user.



**Figure 3.1: Example illustrating neural network forward and inverse models, (a) forward model (b) inverse model. The inputs  $x_3$  and  $x_4$  (output  $y_2$  and  $y_3$ ) of the forward model are swapped to the outputs (inputs) of the inverse model respectively.**

Let us define a sub-set of  $x$  and a sub-set of  $y$ . These sub-sets of input and output are swapped to the output and input respectively in order to form the inverse model. Let  $I_x$  be defined as an index set containing the indices of inputs of forward model that are moved to the output of inverse model,

$$I_x = \{i \mid \text{if } x_i \text{ becomes output of inverse model}\}. \quad (3.2)$$

Let  $I_y$  be the index set containing the indices of outputs of forward model that are moved to the input of inverse model,

$$I_y = \{i \mid \text{if } y_i \text{ becomes input of inverse model}\}. \quad (3.3)$$

Let  $\bar{x}$  and  $\bar{y}$  be vectors of inputs and outputs of the inverse model. The inverse model can be defined as

$$\bar{y} = \bar{f}(\bar{x}) \quad (3.4)$$

where  $\bar{y}$  includes  $y_i$  if  $i \notin I_y$  and  $x_i$  if  $i \in I_x$ ;  $\bar{x}$  includes  $x_i$  if  $i \notin I_x$  and  $y_i$  if  $i \in I_y$ ; and  $\bar{f}$  defines the input-output relationship of the inverse model. For example the inputs  $x_3$  and  $x_4$  of Figure 3.1(a) may represent the iris length and width of a filter, and outputs  $y_2$  and  $y_3$  may represent electrical parameter such as coupling parameter and insertion phase. To formulate the inverse filter model we swap the iris length and width with coupling parameter and insertion phase. For the example in Figure 3.1 the inverse model is formulated as

$$I_x = \{3, 4\} \quad (3.5)$$

$$I_y = \{2, 3\} \quad (3.6)$$

$$\bar{\mathbf{x}} = [\bar{x}_1 \ \bar{x}_2 \ \bar{x}_3 \ \bar{x}_4]^T = [x_1 \ x_2 \ y_2 \ y_3]^T \quad (3.7)$$

$$\bar{\mathbf{y}} = [\bar{y}_1 \ \bar{y}_2 \ \bar{y}_3]^T = [y_1 \ x_3 \ x_4]^T \quad (3.8)$$

After formulation is finished model can be trained with the data. Usually data are generated by EM solvers originally in forward way, i.e., given iris length and compute coupling parameter. To train a neural network as an inverse model, we swap the generated data so that coupling parameter becomes training data for neural network inputs and iris length becomes training data for neural network outputs. The neural network trained this way is the direct inverse model.

The direct inverse modeling method is simple, and is suitable when the problem is relatively easy, for example, when the original input-output relationship is smooth and monotonic, and/or if the numbers of inputs/outputs are small. On the other hand if the problem is complicated and models using direct method are not accurate enough, then segmentation of training data can be utilized to improve the model accuracy. Segmentation of microwave structures has been reported in existing literature such as [7] where a large device is segmented into smaller units. The smaller units are modeled individually and then combined together to obtain the complete device model. We apply the segmentation concepts over the range of model inputs to split data into smaller sections. The complexity of input-output relationships affects the amount of data to be included during neural network training to capture the device behavior completely. The

relationship may contain multiple nonlinear sections, which need dense sampling during data generation. Including the entire training data in a single neural network inverse model may lower the model accuracy. Therefore we split data into multiple sections each covering a smaller range of input parameter space. Neural network models are trained for each section of data. A small amount of overlapping data can be reserved between adjacent sections so that the connections between neighboring segmented models become smooth.

### **3.2.2 Non-Uniqueness of Input-Output Relationship in Inverse Model and Proposed Solutions**

When the original forward input-output relationship is not monotonic, the non-uniqueness becomes an inherent problem in the inverse model. In order to solve this problem, we start by addressing multivalued solutions in training data as follows:

If two different input values in the forward model lead to the same value of output then a contradiction arises in the training data of the inverse model, because the single input value in the inverse model has two different output values. Since we cannot train the neural network inverse model to match two different output values simultaneously, the training error cannot be reduced to a small value. As a result the trained inverse model will not be accurate. For this reason, it is important to detect the existence of multivalued solutions, which creates contradictions in training data.

Detection of multivalued solutions would have been straightforward if the training data were generated by deliberately choosing different geometrical dimensions such that they lead to the same electrical value. However in practice, the training data are not

sampled at exactly those locations. Therefore we need to develop numerical criteria to detect the existence of multivalued solutions.

We assume  $I_x$  and  $I_y$  contain same amount of indices, and that the indices in  $I_x$  (or  $I_y$ ) are in ascending order. Let us define the distance between two samples of training data, sample number  $l$  and  $k$  as

$$u^{(k,l)} = \sqrt{\sum_{i=1}^n (\bar{x}_i^{(k)} - \bar{x}_i^{(l)})^2 / (\bar{x}_i^{\max} - \bar{x}_i^{\min})^2} \quad (3.9)$$

where,  $\bar{x}_i^{\max}$  and  $\bar{x}_i^{\min}$  are the maximum and minimum value of  $\bar{x}_i$  respectively as determined from training data. We use a superscript to denote the sample index in training data. For example,  $\bar{x}_i^{(k)}$  and  $\bar{y}_i^{(k)}$  represent values of  $\bar{x}_i$  and  $\bar{y}_i$  in the  $k^{\text{th}}$  training data respectively. Sample  $\bar{x}^{(k)}$  is in the neighborhood of  $\bar{x}^{(l)}$  if  $u^{(k,l)} < \alpha$ , where  $\alpha$  is a user-defined threshold whose value depends on the step size of data sampling. The maximum and minimum “slope” between samples within the neighborhood of  $\bar{x}^{(l)}$  is defined as

$$G_{\max}^{(l)} = \sqrt{\max_k \frac{\sum_{i=1}^m \{(\bar{y}_i^{(k)} - \bar{y}_i^{(l)}) / (\bar{y}_i^{\max} - \bar{y}_i^{\min})\}^2}{\sum_{i=1}^n \{(\bar{x}_i^{(k)} - \bar{x}_i^{(l)}) / (\bar{x}_i^{\max} - \bar{x}_i^{\min})\}^2}}_{u^{(k,l)} < \alpha} \quad (3.10)$$

and

$$G_{\min}^{(l)} = \sqrt{\min_{u^{(k,l)} < \alpha} \frac{\sum_{i=1}^m \{(\bar{y}_i^{(k)} - \bar{y}_i^{(l)}) / (\bar{y}_i^{\max} - \bar{y}_i^{\min})\}^2}{\sum_{i=1}^n \{(\bar{x}_i^{(k)} - \bar{x}_i^{(l)}) / (\bar{x}_i^{\max} - \bar{x}_i^{\min})\}^2}} \quad (3.11)$$

Input sample  $\bar{x}^{(l)}$  will have multivalued solutions if, within its neighborhood, the slope is larger than maximum allowed or the ratio of maximum and minimum slope is larger than the maximum allowed slope change. Mathematically, if

$$G_{\max}^{(l)} > G_M \quad (3.12)$$

and

$$G_{\max}^{(l)} / G_{\min}^{(l)} > G_R \quad (3.13)$$

then  $\bar{x}^{(l)}$  has multivalued solutions in its neighborhood where  $G_M$  is the maximum allowed slope and  $G_R$  is the maximum allowed slope change.

We employ the simple criteria of (3.12) and (3.13) to detect possible multivalued solutions. A suggestion for  $\alpha$  can be at least twice the average step size of  $y$  in training data. A reference value for  $G_R$  can be approximately the inverse of a similarly defined “slope” between adjacent samples in the training data of the forward model. The value of  $G_M$  should be greater than 1. In the overall modeling method, conservative choices of  $\alpha$ ,  $G_M$  and  $G_R$  (larger  $\alpha$ , smaller  $G_M$  and  $G_R$ ) lead to more use of the derivative division procedure to be described in the next section, while aggressive choices of  $\alpha$ ,  $G_M$  and  $G_R$

lead to early termination of the overall algorithm (or more use of the segmentation procedure) when model accuracy is achieved (or not achieved). In this way, the choices of  $\alpha$ ,  $G_M$  and  $G_R$  mainly affect the training time of the inverse models, rather than model accuracy. The modeling accuracy is determined from segmentation or from the derivative division step to be described in the next section. Sample values of  $\alpha$ ,  $G_M$  and  $G_R$  are given through an example in Section 3.4.1.

### 3.2.3 Proposed Method to Divide Training Data Containing Multivalued Solutions

If existence of multivalued solutions is detected in training data, we perform data preprocessing to divide the data into different groups such that the data in each group do not have the problem of multivalued solutions. To do this, we need to develop a method to decide which data samples should be moved into which group. We propose to divide the overall training data into groups based on derivatives of outputs vs. inputs of the forward model. Let us define the derivatives of inputs and outputs that have been exchanged to formulate the inverse model, evaluated at each sample, as

$$\left. \frac{\partial y_i}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}^{(k)}}, \quad i \in I_y \text{ and } j \in I_x \quad (3.14)$$

where  $k = 1, 2, 3, \dots, N_s$  and  $N_s$  is the total number of training samples. The entire training data should be divided based on the derivative criteria such that training samples satisfying

$$\left. \frac{\partial y_i}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} < \beta \quad (3.15)$$

belong to one group and training samples satisfying

$$\left. \frac{\partial y_i}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}^{(k)}} > -\beta \quad (3.16)$$

belong to a different group. The value for  $\beta$  is zero by default. However to produce an overlapping connection at the break point between the two groups we can choose a small positive value for it. In that case a small amount of data samples whose absolute values of derivative are less than  $\beta$  will belong to both groups. The value for  $\beta$  other than the default suggestion of zero can be chosen as a value slightly larger than the smallest absolute value of derivatives of (3.14) for all training samples. Choice of  $\beta$  only affects the accuracy of the sub-models at the connection region. The model accuracy for the rest of the region will remain unaffected.

This method exploits derivative information to divide the training data into groups. Therefore, accurate derivative is an important requirement for this method. Computation of derivatives of (3.14) is not a straightforward task since no analytical equation is available. We propose to compute the derivatives by exploiting adjoint neural network technique [45]. We first train an accurate neural network forward model. After training is finished, its adjoint neural network can be used to produce the derivative

information used in (3.15) and (3.16). The computed derivatives are employed to divide the training data into multiple smaller groups according to (3.15) and (3.16) using different combinations of  $i$  and  $j$ . Multiple neural networks are then trained with the divided data. Each neural network represents a sub-model of the overall inverse model.

Equations (3.12) and (3.13) play different roles versus Equations (3.15) and (3.16) in our overall algorithm to be described in Section 3.3. Equations (3.12) and (3.13) are used as simple and quick ways to detect the existence of contradictions in training data. But they do not give enough information on how the data should be divided. Equations (3.15) and (3.16), which require more computation (i.e., require training forward neural model) and produce more information, are used to perform detailed task of dividing training data into different groups to solve the multivalued problem.

### 3.2.4 Proposed Method to Combine the Inverse Sub-Models

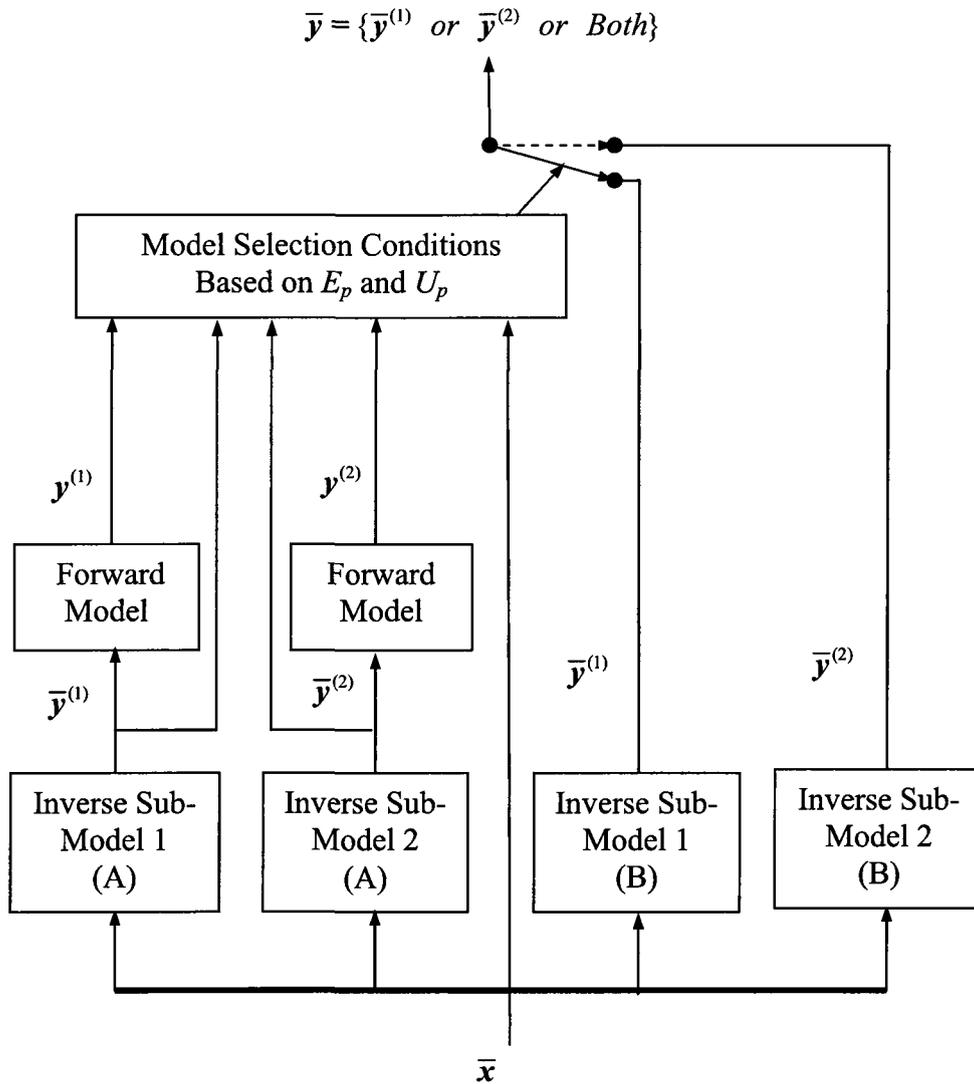
We need to combine the multiple inverse sub-models to reproduce the overall inverse model completely. For this purpose a mechanism is needed to select the right one among multiple inverse sub-models for a given input  $\bar{x}$ . Figure 3.2 shows the proposed inverse sub-model combining method for a two sub-model system. For convenience of explanation, suppose  $\bar{x}$  is a randomly selected sample of training data. Ideally if  $\bar{x}$  belongs to a particular inverse sub-model then the output from it should be the most accurate one among various inverse sub-models. Conversely the outputs from the other inverse sub-models should be less accurate if  $\bar{x}$  does not belong to them. However, when using the inverse sub-models with general input  $\bar{x}$  whose values are not necessarily equal to that of any training samples, the value from the sub-models is the unknown

parameter to be solved. So we still do not know which inverse sub-model is the most accurate one.

To address this dilemma, we use the forward model to help decide which inverse sub-model should be selected. If we supply an output from the correct inverse sub-model to an accurate forward model we should be able to obtain the original data input to the inverse sub-model. For example, suppose  $y = f(x)$  is an accurate forward model. Suppose the inputs and outputs of the inverse sub-model are defined such that  $\bar{x} = y$  and  $\bar{y} = x$ . If the inverse sub-model  $\bar{y} = \bar{f}(\bar{x})$  is true, then

$$f(\bar{f}(\bar{x})) = \bar{x} \quad (3.17)$$

is also true. Conversely, if  $f(\bar{f}(\bar{x})) \neq \bar{x}$  then,  $\bar{f}(\bar{x})$  is a wrong inverse sub-model. In this way we can use a forward model to help determining which inverse sub-model should be selected for a particular value of input. In our method input  $\bar{x}$  is supplied to each inverse sub-model and output from them is fed to the accurately trained forward model respectively, which generate different  $y$ . These outputs are then compared with the input data  $\bar{x}$ . The inverse sub-model that produces least error between  $y$  and  $\bar{x}$  is selected and the output from the corresponding inverse sub-model is chosen as the final output of the overall inverse modeling problem.



**Figure 3.2: Diagram of inverse sub-model combining technique after derivative division for a two sub-model system. Inverse sub-model 1, and inverse sub-model 2 in set (A) are competitively trained version of the inverse sub-models. Inverse sub-model 1 and inverse sub-model 2 in set (B) are trained with the divided data based on derivative criteria (3.15) – (3.16). The input and output of the overall combined model is  $\bar{x}$  and  $\bar{y}$  respectively.**

Let us assume an inverse model is divided into  $N$  different inverse sub-models according to derivative criteria. The error between the input of the  $p^{\text{th}}$  inverse sub-model and output of the forward model (also called error from inverse-forward sub-model pair) is calculated as

$$E_p = \frac{1}{2} \sqrt{\sum_{\substack{i \in I_y \\ j \in I_x}} (y_i^{(p)} - \bar{x}_j)^2 / (\bar{x}_j^{\max} - \bar{x}_j^{\min})^2} \quad (3.18)$$

where,  $p = 1, 2, 3, \dots, N$  and we have assumed  $I_y$  and  $I_x$  contain equal number of indices.  $N$  is the number of sub models. As an example,  $E_1$  would be lower than  $E_2, E_3, \dots, E_N$  if a sample  $\bar{x}$  belongs to the inverse sub-model 1.

We include another constraint to the inverse sub-model selection criteria. This constraint checks for the training range. If an inverse sub-model produces an output that is located outside its training range, then the corresponding output is not selected even though the error ( $E_p$ ) of (3.18) is less than that of other inverse sub-models. If the outputs of other inverse sub-models are also found outside their training range then we compare their magnitude of distances from the boundary of training range. An inverse sub-model producing the shortest distance is selected in this case. For sub-model  $p$  the distance of a particular output outside the training range can be defined as

$$U_i^{(p)} = \begin{cases} \bar{y}_i^{(p)} - \bar{y}_i^{\max}, & \text{for } \bar{y}_i^{(p)} > \bar{y}_i^{\max} \\ \bar{y}_i^{\min} - \bar{y}_i^{(p)}, & \text{for } \bar{y}_i^{(p)} < \bar{y}_i^{\min} \\ 0, & \text{otherwise} \end{cases} \quad (3.19)$$

where  $i \in I_x$ ,  $p = 1, 2, 3, \dots, N$ , and  $\bar{y}_i^{\max}$  and  $\bar{y}_i^{\min}$  are the maximum and minimum values of  $\bar{y}_i$  respectively obtained from the training data. For any output  $\bar{y}$  if the distance is zero, then the output is located inside the training range. The total distance outside the range for all the outputs of an inverse sub-model  $p$  can be calculated as

$$U_p = \sum_{i \in I_x} U_i^{(p)} \quad (3.20)$$

where  $i \in I_x$  and  $p = 1, 2, 3, \dots, N$ .

The calculated  $E_p$  and  $U_p$  are used to determine which inverse sub-model should be selected for a particular set of input. The inverse sub-model selection criteria can be expressed as

$$\bar{y} = \bar{y}^{(p)}, \quad (3.21)$$

if  $(U_p = 0)$  AND  $(U_q = 0)$  AND  $(E_p < E_q)$ , or  $((U_p \neq 0)$  OR  $(U_q \neq 0))$  AND  $(U_p < U_q)$  for all values of  $q$  where  $q = 1, 2, 3, \dots, N$  and  $q \neq p$ . For example, inverse sub-model 1 is selected if outputs from all the inverse sub-models are located inside the training range and the error produced by the inverse-forward sub-pair 1 is less than the error produced by all other pairs, or if the output of any of the inverse sub-model is located outside the training range and the distance of the output of inverse sub-model 1 is the least of that of all other inverse sub-models.

In cases when outputs from multiple inverse sub-models remain inside the training range (i.e.,  $U_p = 0$ ) and at the same time the errors (i.e.,  $E_p$ ) calculated from the

corresponding inverse-forward pairs are all smaller than a threshold value ( $E_T$ ), then the outputs of those inverse sub-models are valid solutions. As an example suppose we have 3 inverse sub-models ( $N = 3$ ). For a particular sample of data if the outputs from inverse sub-model 1 and inverse sub-model 2 both fall within the training range ( $U_1 = U_2 = 0$ ) and the errors  $E_1$  and  $E_2$  are both less than the threshold error  $E_T$ , then solutions from inverse sub-model 1 and inverse sub-model 2 are both accepted.

The purpose of the model combining technique is to reproduce the original multivalued input-output relationship for the user. Our method is an advance over the direct inverse modeling method since the latter produces only an inaccurate result in case there are multivalued solutions (i.e., produces a single solution which may not match any of the original multi-values). Our method can be used to provide a quick model to reproduce multivalued solutions in inverse EM problems. Using the solutions from the proposed inverse model (including reproduced multivalued solutions), the user can proceed to circuit design.

### **3.2.5 Accuracy Enhancement of Sub-Model Combining Method**

Here we describe two ways to further enhance the selection and thus improve the accuracy of the overall inverse model. These enhancement techniques are used only for some sub regions where model selections are inaccurate. In most cases regularly trained inverse sub-models will be accurate with no need of these enhancement techniques. The sub regions which need enhancement can be determined by checking the model selection using the known divisions in training data. The application of the enhancement techniques will incrementally increase model development time.

### 3.2.5.1 Competitively Trained Inverse Sub-Model

To further improve the inverse sub-model selection accuracy an additional set of competitively trained inverse sub-model can be used. These inverse sub-models are trained to learn not only what is correct but also what is wrong. Correct data are the data that belong only to a particular inverse sub-model. Conversely, incorrect data are the data in which  $\bar{x}$  belongs to other inverse sub-models and  $\bar{y}$  is deliberately set to zero, so that the inverse sub-model is forced to learn wrong values of  $\bar{y}$  for  $\bar{x}$  that do not belong to this inverse sub-model. The output values of these inverse sub-models are not very accurate. But they are reliable to identify if an input belongs to the inverse sub-model or not. Therefore, they are used for the inverse sub-model selection purpose only. Once the selection has been made the final output is taken from the regularly trained (i.e., not competitively trained) inverse sub-model. In Figure 3.2, the inverse sub-models in set (A) represent the competitively trained inverse sub-models and set (B) represents regularly trained inverse sub-models.

### 3.2.5.2 Forward Sub-Model

The default forward model used in model combining method is trained with the entire set of training data. The decision of choosing the right inverse sub-model depends on the accuracy of both inverse sub-models and forward models. We can further tighten the accuracy of the forward model by training multiple forward sub-models using the same groups of data used to train inverse sub-models. These forward sub-models capture the same data range as its inverse counterpart and therefore the inverse and forward sub-

model pairs are capable of producing more accurate decision. In Figure 3.2 the forward models are replaced with the forward sub-models.

### **3.3 Overall Inverse Modeling Methodology**

The overall methodology of inverse modeling combines all the aspects described in the previous section. The inverse model of a microwave device may contain unique or non-unique behavior over various regions of interest. In the region with unique solutions direct segmentation can be applied and training error is expected to be low. On the other hand, in the region with non-uniqueness, the model should be divided according to derivative. If the overall problem is simple, the methodology will end with a simple inverse model directly trained with all data. In complicated cases, the methodology uses derivative division and sub-model combining method to increase model accuracy. This approach increases the overall efficiency of modeling. The flow diagram of the overall inverse modeling approach is presented in Figure 3.3. The overall methodology is summarized in the following steps:

Step 1. Define the inputs and outputs of the model. Detailed formulation can be found in

Section 3.2.1. Generate data using EM simulator or measurement. Swap the input and output data to obtain data for training inverse model. Train and test the inverse model. If the model accuracy is satisfied then stop. Results obtained here is the direct inverse model.

Step 2. Segment the training data into smaller sections. If there have been several consecutive iterations between Steps 2 and 5, then go to Step 6.

Step 3. Train and test models individually with segmented data.

Step 4. If the accuracy of all the segmented models in Step 3 is satisfied, stop. Else for the segments that have not reached accuracy requirements, proceed to the next steps.

Step 5. Check for multivalued solutions in model's training data using (3.12) and (3.13). If none are found then perform further segmentation by going to Step 2.

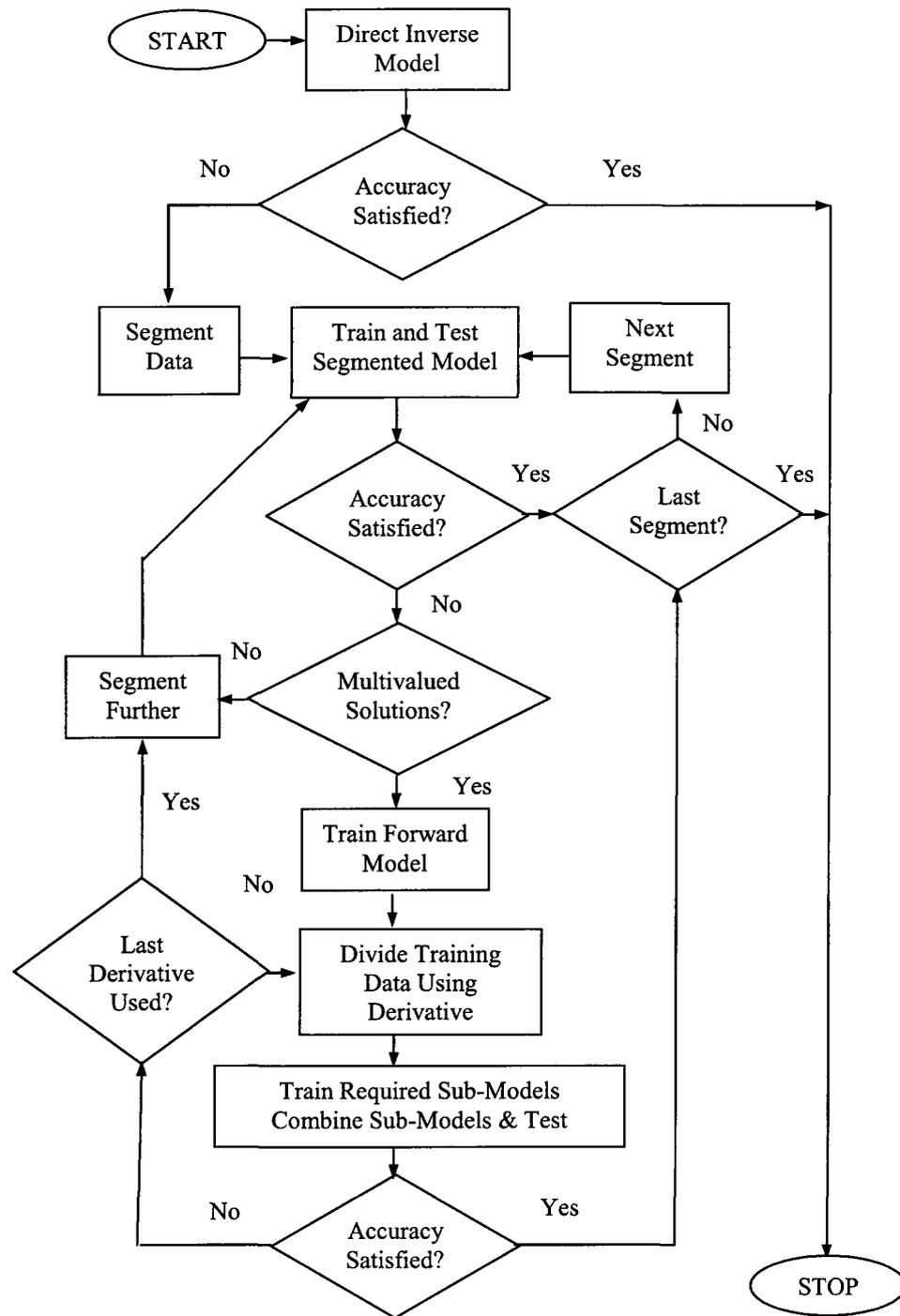
Step 6. Train a neural network forward model.

Step 7. Using the adjoint neural network of the forward model divide the training data according to derivative criteria as described in Section 3.2.3.

Step 8. With the divided data, train necessary sub-models, for example two inverse sub-models. Optionally obtain two competitively trained inverse sub-models and two forward sub-models.

Step 9. Combine all the sub-models that have been trained in Step 8 according to method in Section 3.2.4. Test the combined inverse sub-models. If the test accuracy is achieved then stop. Else go to Step 7 for further division of data according to derivative information in different dimensions, or if all the dimensions are exhausted, go to Step 2.

The algorithm increases efficiency by choosing the right techniques in the right order. For simple problems, the algorithm stops immediately after the direct inverse modeling technique. In this case no data segmentation or other techniques are used, and training time is short. The segmentation and subsequent techniques will be applied only when the directly trained model cannot meet accuracy criteria. In this way, more training time is needed only with more complexity in the model input-output relationship, such as the multivalued relationship.



**Figure 3.3: Flow diagram of overall inverse modeling methodology consisting of direct, segmentation, derivative dividing and model combining techniques.**

## 3.4 Examples and Applications to Filter Design

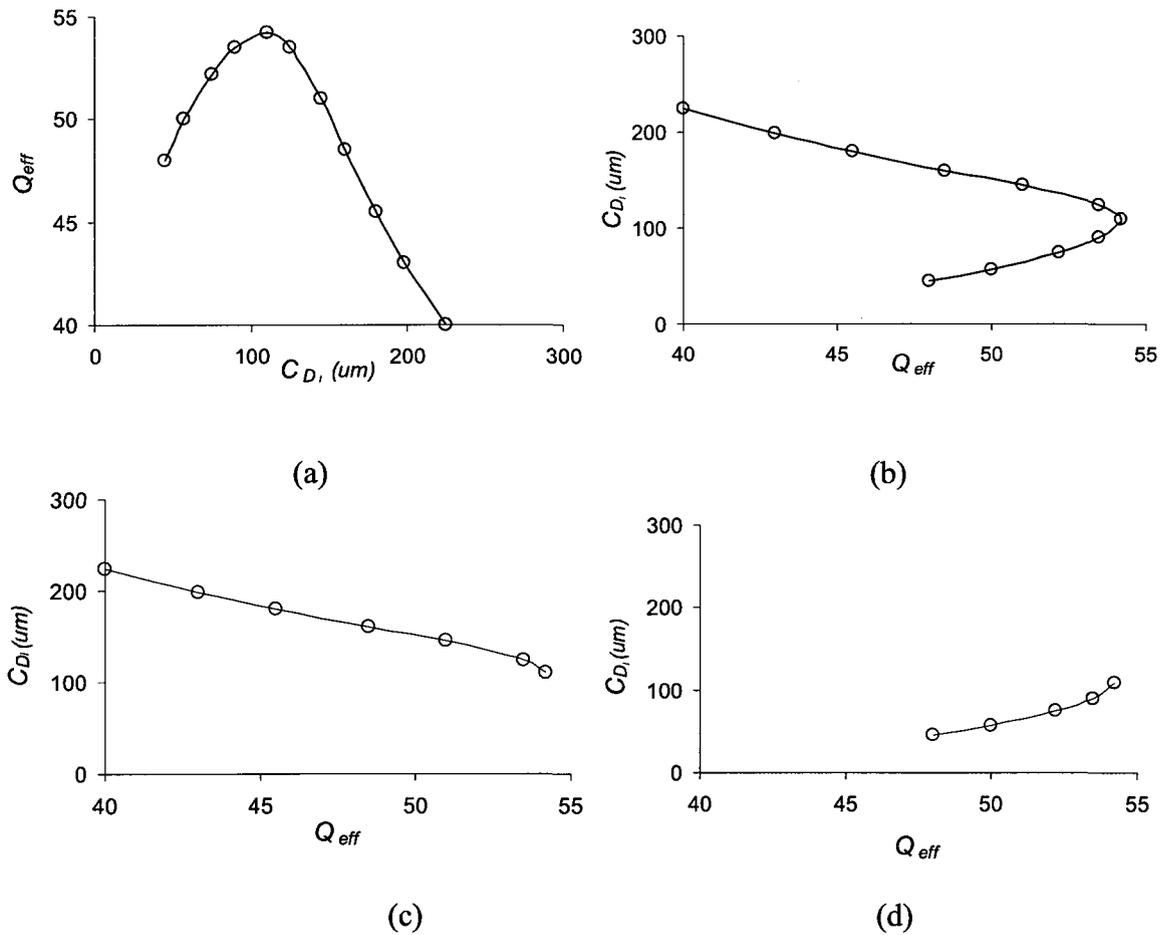
### 3.4.1 Example 1: Inverse Spiral Inductor Model

In this example we illustrate the proposed technique through a spiral inductor modeling problem where the input of the forward model is the inner mean diameter ( $C_{D_i}$ ) of the inductor, and the output is the effective quality factor ( $Q_{eff}$ ). Figure 3.4(a) shows the variation of  $Q_{eff}$  with respect to inner diameter [126]. The inverse model of this problem is shown in Figure 3.4(b), which shows non-unique input-output relationship since in the range from  $Q_{eff} = 47$  to  $Q_{eff} = 55$ , a single  $Q_{eff}$  value will produce two different  $C_{D_i}$  values. We have implemented (3.10), (3.11), (3.12) and (3.13) in *NeuroModelerPlus* [127] to detect the existence of multivalued solutions as described in Section 3.2.2. We supply the training data to *NeuroModelerPlus* and set values of parameters as  $G_M = 80$ ,  $G_R = 80$ , and  $\alpha = 0.01$ . The program detects several contradictions in the data.

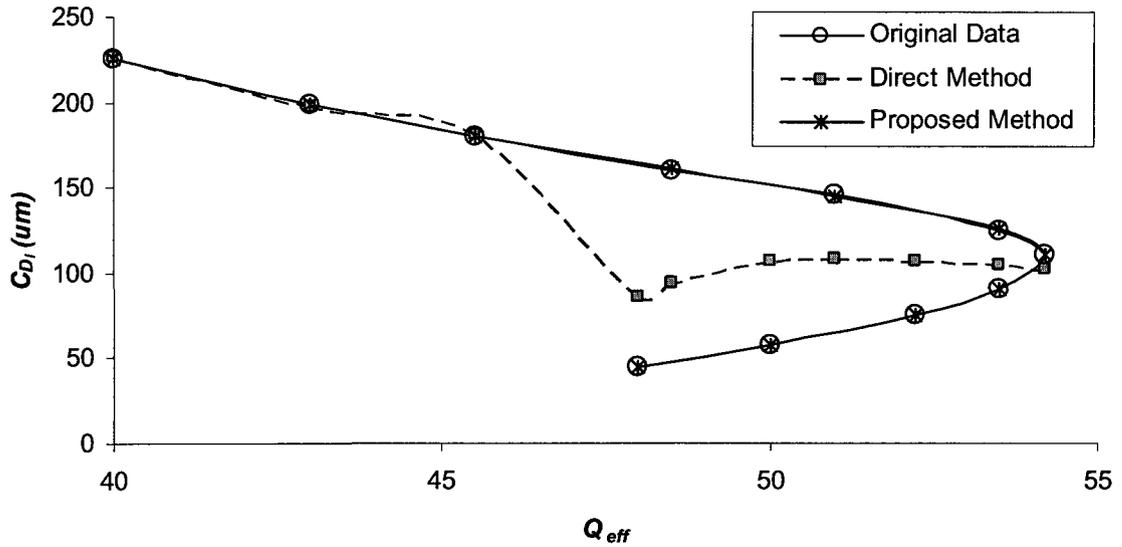
In the next step, we divide the training data according to derivative. We trained a neural network forward model to learn the data in Figure 3.4(a) and used its adjoint neural network to compute the derivatives  $\frac{\partial Q_{eff}}{\partial C_{D_i}}$ . We compared all the values of derivatives and the lowest absolute value was found to be 0.018. The next large absolute value of derivative was 0.07. Therefore we chose the value of  $\beta = 0.02$  which is in between 0.018 and 0.07. The training data are divided such that samples satisfying (3.15) are divided into group I and samples satisfying (3.16) are divided into group II. Figures 3.4(c) and 3.4(d) show the plots of two divided groups, which confirm that the individual

groups become free of multivalued solutions after dividing the data according to the derivative information.

Two inverse sub-models of the spiral inductor were trained using the divided data of Figure 3.4(c) and 3.4(d). The two individual sub-models became very accurate and they were combined using the model combining technique. For comparison purpose, a separate model was trained using the direct inverse modeling method, which means that all the training samples in Figure 3.4(b) were used without any data division to train a single inverse model. The results are shown in Figure 3.5. It shows that the model obtained from direct inverse modeling method produce inaccurate result because of confusions over training data with multivalued solutions. The model trained using the proposed methodology delivers accurate solutions that match the data for the entire range. Average test error reduced from 13.6% down to 0.05% using proposed techniques over the direct inverse modeling method.



**Figure 3.4: Non-uniqueness of input-output relationship is observed when  $Q_{eff}$  vs.  $C_{D_i}$  data of a forward spiral inductor model is exchanged to formulate an inverse model. (a) Unique relationship between input and output of a forward model, (b) Non-unique relationship of input-output of an inverse model obtained from forward model of (a). Training data containing multivalued solutions of Figure 3.4(b) are divided into groups according to derivative. (c) Group I data with negative derivative, (d) group II data with positive derivative. Within each group, the data are free of multivalued solutions, and consequently the input-output relationship becomes unique.**

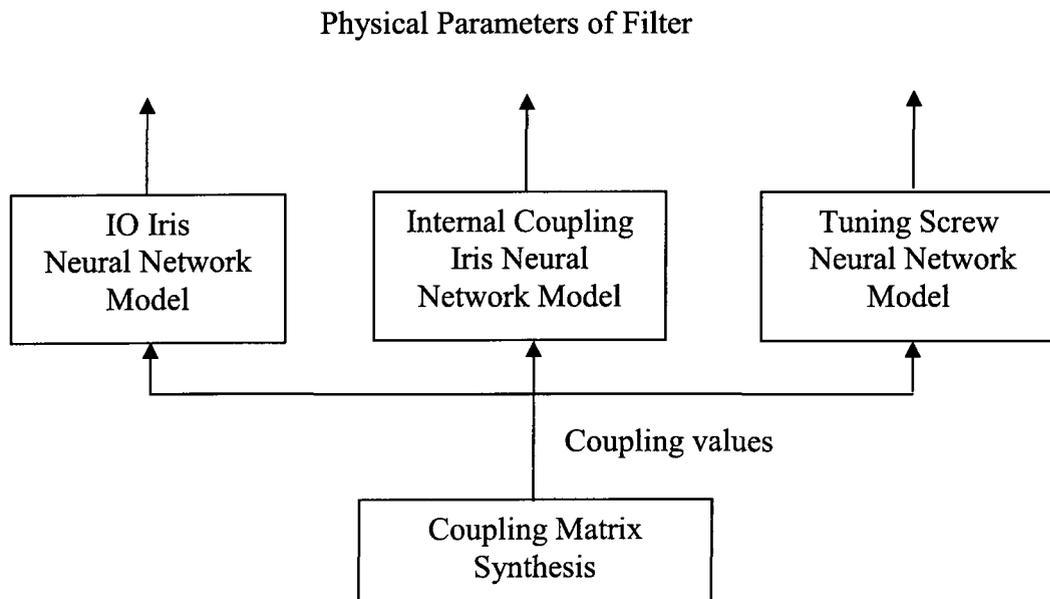


**Figure 3.5: Comparison of inverse model using the proposed methodology and the direct inverse modeling method for the spiral inductor example.**

### 3.4.2 Example 2: Filter Design Approach and Development of Inverse Coupling Iris and IO Iris Models

Neural network modeling techniques are applied to the microwave waveguide filter design. The filter design starts from synthesizing the coupling matrix to satisfy ideal filter specifications. The EM method for finding physical/geometrical parameters to realize the required coupling matrix is an iterative EM optimization procedure. In this procedure performs EM analysis (mode-matching or finite element methods) on each waveguide junction of the filter to get the generalized scattering matrix (GSM). From GSM we extract coupling coefficients. We then modify the design parameters (i.e., the dimensions of filter) and re-perform EM analysis iteratively until the required coupling coefficients are realized. In our proposed approach we avoid this iterative step and use neural network inverse models to directly provide the filter dimensions.

In the present work, the filter is decomposed into three different modules each representing a separate filter junction. Neural network inverse models of these junctions were developed separately using the proposed methodology. The three models are the input-output (IO) iris, the internal coupling iris, and the tuning screws. Training data for neural networks are generated from physical parameters firstly through EM simulation (Mode matching method) producing GSM. Coupling values are then obtained from GSM through analytical equations. Figure 3.6 demonstrates the filter design approach. More detailed information on modeling and design procedure for filter can be found in [17].



**Figure 3.6: Diagram of the filter design approach using the neural network inverse models.**

In this example, we develop two inverse neural network models for the waveguide filter. The first neural network inverse model of the filter structure is developed for the internal coupling iris. The inputs and outputs of the internal coupling iris forward model are

$$\mathbf{x} = [C_D \ \omega_o \ L_v \ L_h]^T \quad (3.22)$$

$$\mathbf{y} = [M_{23} \ M_{14} \ P_v \ P_h]^T \quad (3.23)$$

where  $C_D$  is the circular cavity diameter,  $\omega_o$  is the center frequency,  $M_{23}$  and  $M_{14}$  are coupling values,  $L_v$  and  $L_h$  are the vertical and horizontal coupling slot lengths and  $P_v$  and  $P_h$  are the loading effect of the coupling iris on the two orthogonal modes, respectively.

The inverse model is formulated as

$$\bar{\mathbf{y}} = [x_3 \ x_4 \ y_3 \ y_4]^T = [L_v \ L_h \ P_v \ P_h]^T \quad (3.24)$$

$$\bar{\mathbf{x}} = [x_1 \ x_2 \ y_1 \ y_2]^T = [C_D \ \omega_o \ M_{23} \ M_{14}]^T. \quad (3.25)$$

The second inverse model of the filter is the IO iris model. The input parameters of IO iris inverse model are circular cavity diameter  $C_D$ , center frequency  $\omega_o$ , and the coupling value  $R$ . The output parameters of the model are the iris length  $L_r$ , the loading effect of the coupling iris on the two orthogonal modes  $P_v$  and  $P_h$ , and the phase loading on the input rectangular waveguide  $P_{in}$ . The IO iris forward model is formulated as

$$\mathbf{x} = [C_D \ \omega_o \ L_r]^T \quad (3.26)$$

$$\mathbf{y} = [R \ P_v \ P_h \ P_{in}]^T. \quad (3.27)$$

The inverse model is defined as

$$\bar{\mathbf{y}} = [x_3 \ y_2 \ y_3 \ y_4]^T = [L_r \ P_v \ P_h \ P_{in}]^T \quad (3.28)$$

$$\bar{\mathbf{x}} = [x_1 \ x_2 \ y_1]^T = [C_D \ \omega_o \ R]^T. \quad (3.29)$$

Training data were generated in the forward way (according to forward model) and the data are then reorganized for training inverse model. The entire data was used to train the inverse internal coupling iris model. For IO iris model four different sets of training data were generated according to the width of iris using mode-matching method. The model for each set was trained and tested separately using the direct inverse modeling method. For both the iris models direct training produced good accuracy in terms of average and L2 (least squares [2]) errors. However the worst-case errors were large. Therefore in the next step the data was segmented into smaller sections. Models for these sections were trained separately, which reduced the worst-case error. The final model results of the coupling iris model shows that the average error reduced from 0.24% to 0.17% and worst case error reduced from 14.2% to 7.2%. The average error for IO iris model reduced from 1.2% to 0.4% and worst case error reduced from 54% to 18.4%. The errors for other sets of IO iris model also reduced similarly. We can improve the accuracy further by splitting the data set into more sections and achieve accurate results as

required. In this example, our methodology stops with accurate inverse model at Step 4 without derivative division of data. These models are developed using proposed methodology and provide better accuracy than models developed using direct method.

### 3.4.3 Example 3: Inverse Tuning Screw Model

The last neural network inverse model of the filter is developed for tuning screw model. This model has complicated input-output relationships requiring the full algorithm to be applied. Here we describe this example in detail. The model outputs are the phase shift of the horizontal mode across the tuning screw  $P_h$ , coupling screw length  $L_c$ , and the horizontal tuning screw length  $L_h$ . The input parameters of this model are circular cavity diameter  $C_D$ , center frequency  $\omega_o$ , the coupling between the two orthogonal modes in one cavity  $M_{12}$ , and the difference between the phase shift of the vertical mode and that of the horizontal mode across the tuning screw  $P$ . The forward tuning screw model is defined as

$$\mathbf{x} = [C_D \ \omega_o \ L_h \ L_c]^T \quad (3.30)$$

$$\mathbf{y} = [M_{12} \ P \ P_h]^T. \quad (3.31)$$

The inverse model is formulated as

$$\bar{\mathbf{y}} = [y_3 \ x_3 \ x_4]^T = [P_h \ L_h \ L_c]^T \quad (3.32)$$

$$\bar{\mathbf{x}} = [x_1 \ x_2 \ y_1 \ y_2]^T = [C_D \ \omega_o \ M_{12} \ P]^T. \quad (3.33)$$

In the initial step the inverse model was trained directly using entire training data. The training error was high even with many hidden neurons. Therefore we proceed to segment the data into smaller sections. In this example we used the segmentation, which corresponds to 2 adjacent samples of frequency  $\omega_o$  and 2 adjacent samples of diameter  $C_D$ . Each segment of data was used to train a separate inverse model. Some of the segments produced accurate models with error less than 1% while others were still inaccurate.

The segments that could not reach the desired accuracy were checked for the existence of multivalued solutions individually. The method to check the existence of multivalued solutions using (3.10), (3.11), (3.12) and (3.13) as described in Section 3.2.2 has been implemented in the *NeuroModelerPlus* [127] software. We use this program to detect the existence of multivalued solutions in training data. For this example, neighborhood size  $\alpha = 0.01$ , maximum slope  $G_M = 80$ , and maximum slope change  $G_R = 80$  were chosen. *NeuroModelerPlus* suggests that the data contain multivalued solutions. Therefore, we need to proceed to train a neural network forward model and apply the derivative division technique to divide the data.

To compute the derivative we trained a neural network as forward tuning screw model. Then derivatives were computed using adjoint neural network model through *NeuroModelerPlus*. Considering  $\beta = 0$  and applying the derivative  $\frac{\partial P}{\partial L_h}$  to (3.15) and (3.16), we divided the data into group I and group II respectively. Two inverse sub-models were trained using group I and group II data. As in Step 8 of the methodology we

trained two forward sub-models using data of group I and group II. The equations for error criteria  $E_1$  and  $E_2$ , distance criteria  $U_1$  and  $U_2$ , and model selection can be obtained using (3.18), (3.20), and (3.21) respectively.

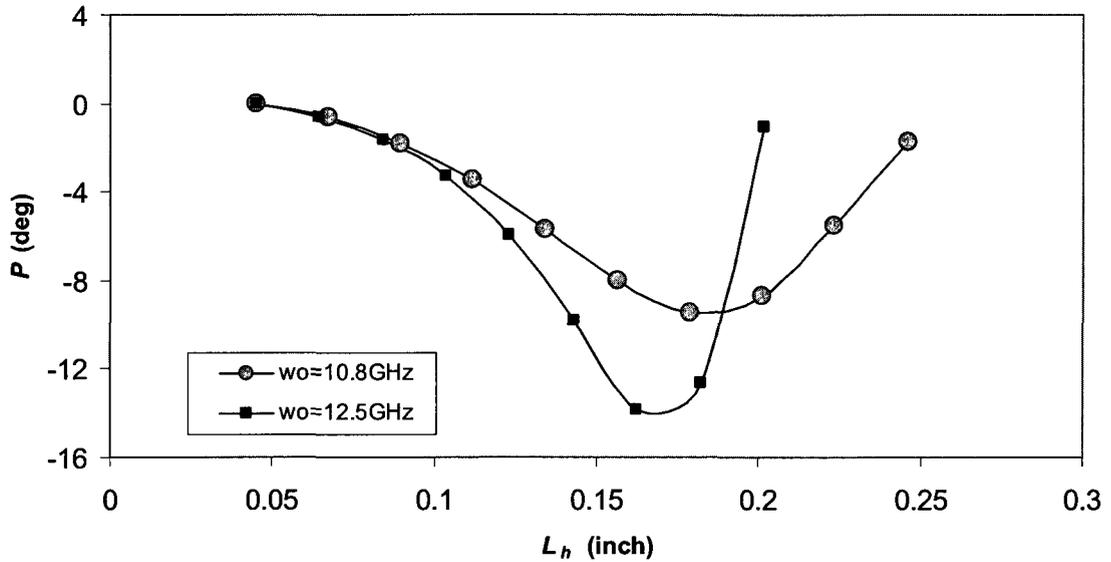
The entire process was done using *NeuroModelerPlus*. The segments that failed to reach good accuracy before became more than 99% accurate after derivative division and model combining technique were applied. The process was continued until all data were captured. A few of the sub-models needed the accuracy enhancement techniques to select the right models and thus reach the desired accuracy. The result of the inverse model using proposed methodology is compared with direct inverse method in Table 3.1, showing the average, L2 and worst-case errors between model and test data. The table demonstrates that the proposed methodology produces significantly better result than the direct method.

Figure 3.7 shows the plot of phase ( $P$ ) for various horizontal screw lengths ( $L_h$ ), which defines the forward model relationship. The two curves in the figure represent forward training data at two different frequencies. The forward relationship is unique which means that there are no multivalued solutions. Figures 3.8(a) and 3.8(b) show the outputs of two inverse models trained using direct and proposed methodology where the output and input are  $L_h$  and  $P$  respectively for two different frequencies. The data of the two plots represent the same data as that in Figure 3.7 except the input and output are swapped. The inverse training data in both plots of Figure 3.8(a) and 3.8(b) contain multivalued solutions and it is clear from the two plots that the inverse model trained using direct method cannot match the data whereas the inverse model using proposed

methodology produce the output  $L_h$  very accurately for the entire range. To demonstrate the variation of multivalued problem at different cavity diameter  $C_D$  we show two more plots in Figure 3.8(c) and 3.8(d). They correspond to two different diameters at the same frequency. Figure 3.8(c) contains multivalued data whereas Figure 3.8(d) does not contain any multivalued data. The plots also compare the outputs of the proposed method and direct method. From Figure 3.8(d) we can see that for single valued case, both methods produce acceptable result whereas in multivalued case (Figure 3.8(c)) only the proposed model can produce accurate result. In reality it is not known beforehand which region contains multivalued data and which region does not. This is why the proposed algorithm is useful to automatically detect the regions that contain multivalued data and apply the appropriate techniques in that region to improve accuracy. In this way, model development can be performed more systematically by computer.

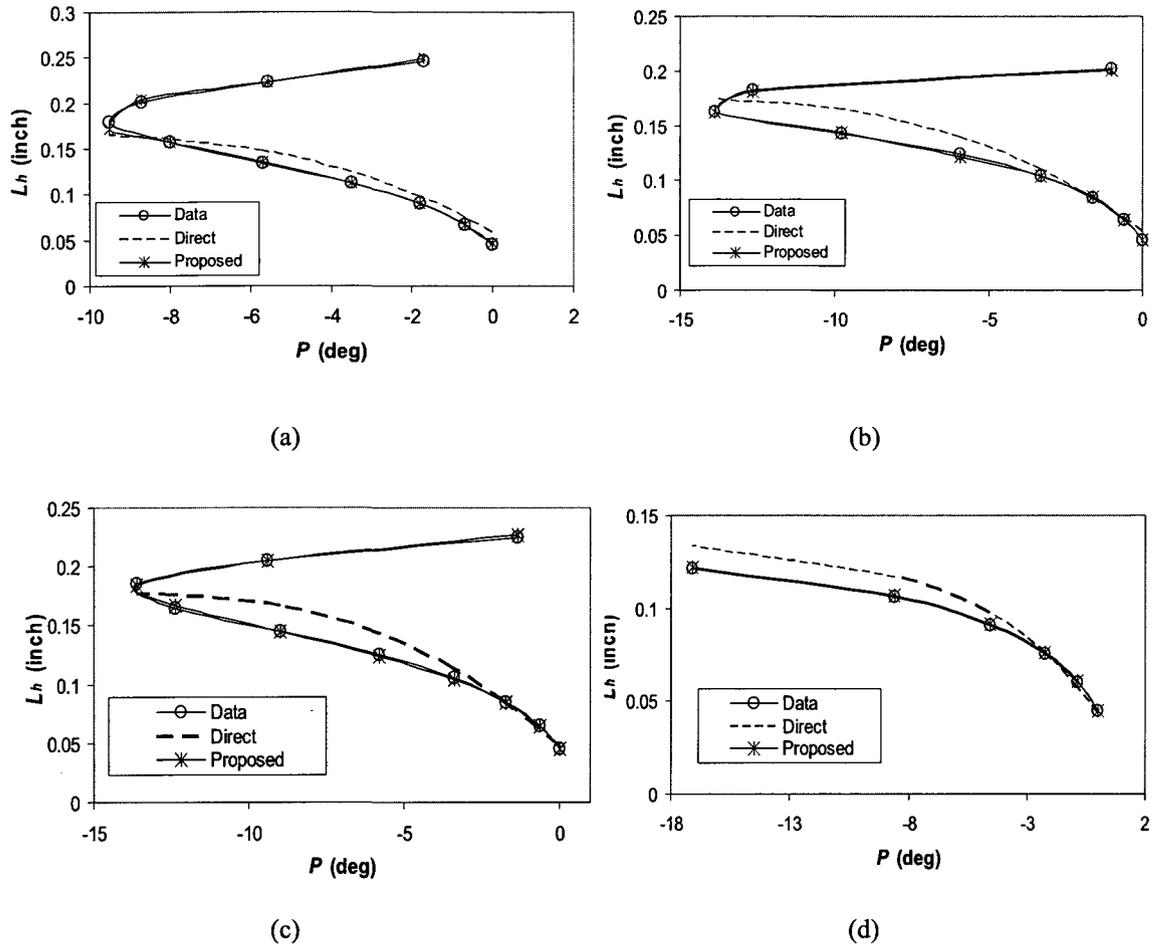
**Table 3.1: Comparison of model test errors between direct and proposed methods for tuning screw model**

Neural network inverse Modeling Method	Model test error (%)		
	Average	L2	Worst case
Direct neural model	3.85	7.51	94.25
Proposed neural model	0.40	0.59	5.10

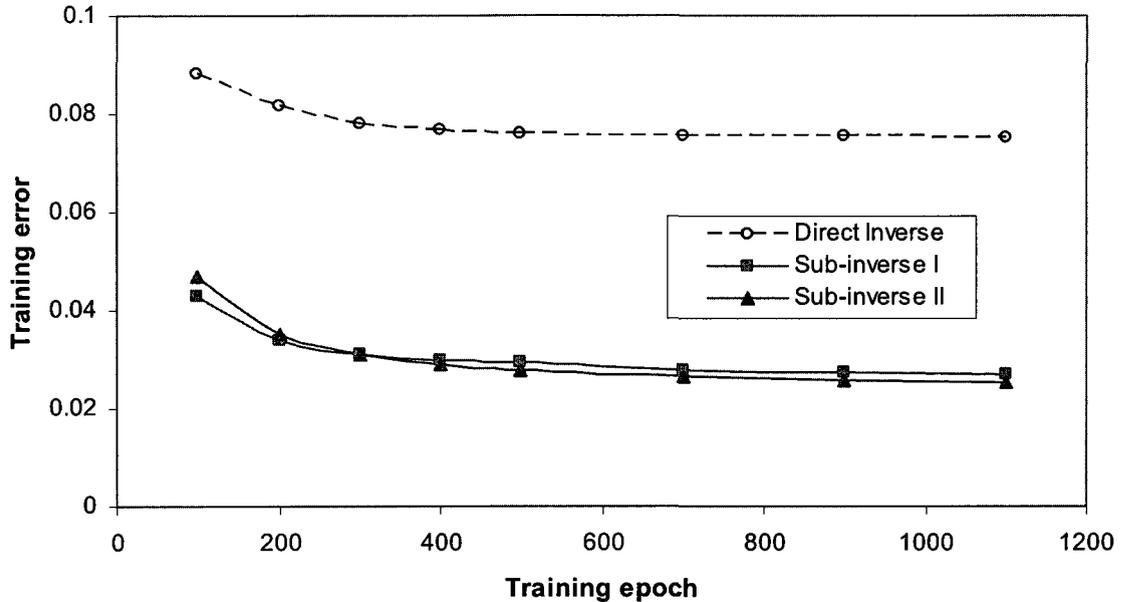


**Figure 3.7: Original data showing variation of phase angle ( $P$ ) with respect to horizontal screw length ( $L_h$ ) describing unique relationship of forward tuning screw model.**

As an additional demonstration of the usefulness of derivative division we applied the same derivative as that described earlier in this section to the entire training data and divided the data into two groups (containing 28000 and 6000 samples respectively) according to (3.15) and (3.16). The training errors of the individual inverse sub-models are compared with that of the direct inverse model in Figure 3.9, which shows that derivative division technique reduces the training error significantly. The test errors are similar as training error in this example. The training epoch in the figure is defined as one iteration of training when all training data have been used to make an update of neural network weights [1].



**Figure 3.8: Comparison of output ( $L_h$ ) of inverse tuning screw model trained using direct and proposed methods at two different frequencies (a)  $\omega_o = 10.8$  GHz,  $C_D = 1.11$  inch (b)  $\omega_o = 12.5$  GHz,  $C_D = 1.11$  inch. It is evident that this inverse model has non-unique outputs. The proposed method produced more accurate inverse model than that of direct inverse method. Inverse data are plotted for two different diameters (c)  $\omega_o = 11.85$ GHz,  $C_D = 1.09$  and (d)  $\omega_o = 11.85$  and  $C_D = 0.95$ . Figure 3.8(c) contains multivalued data whereas 3.8(d) does not contain any multivalued data. This demonstrates the necessity of automatic algorithms to detect and handle multivalued scenarios in different regions of the modeling problem.**



**Figure 3.9: Training error of inverse tuning screw model following direct inverse modeling approach and proposed derivative division approach. The training errors of both the inverse sub-models are lower than that of direct inverse model.**

#### **3.4.4 Example 4: A 4-pole Filter Design for Device Level Verification Using the Three Developed Inverse Models**

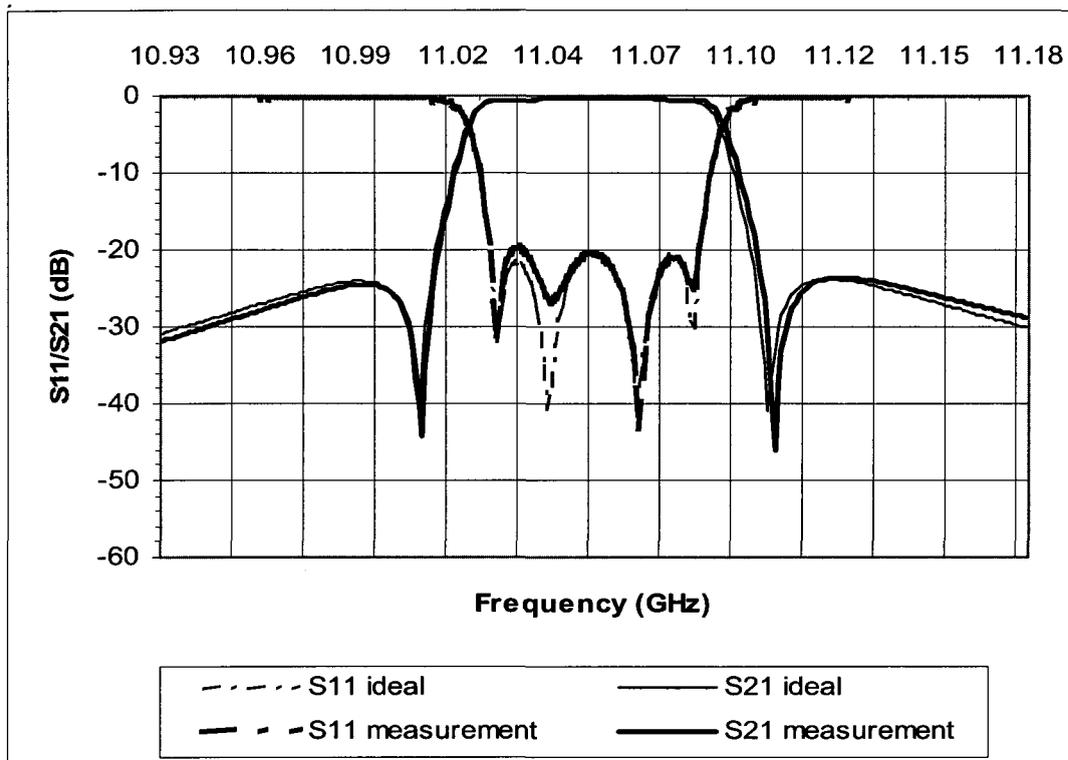
In this example we use the neural network inverse models that were developed in Examples 2 and 3 to design a 4-pole filter with 2 transmission zeros. Compared to the example in [17] which shows the simulation results only, the present example describes new progress, where the filter results are used to fabricate an actual filter and real measurement data are used to validate the neural network solutions.

The layout of a 4-pole filter is similar to that in [17]. The filter center frequency is 11.06 GHz, bandwidth is 58 MHz and cavity diameter is chosen to be 1.17". The normalized ideal coupling values are

$$R_1 = R_2 = 1.07$$

$$M = \begin{bmatrix} 0 & 0.86 & 0 & -0.278 \\ 0.86 & 0 & 0.82 & 0 \\ 0 & 0.82 & 0 & 0.86 \\ -0.278 & 0 & 0.86 & 0 \end{bmatrix} \quad (3.34)$$

The trained neural network inverse models developed in Examples 2 and 3 are used to calculate irises and tuning screw dimensions. The filter is manufactured and then tuned by adjusting irises and tuning screws to match the ideal response. Figure 3.10 compares the measured and the ideal filter response. Dimensions are listed in Table 3.2. Very good correlation can be seen between the initial dimensions provided by the neural network inverse models and the measured final dimensions of the fine tuned filter.



**Figure 3.10: Comparison of the ideal 4-pole filter response with the measured filter response after tuning. The dimensions of the measured filter were obtained from neural network inverse models.**

**Table 3.2: Comparison of dimensions of the 4-pole filter obtained by the neural network inverse model and measurement**

Filter design variables	Neural model (inch)	Measurement (inch)	Difference (inch)
IO irises	0.405	0.405	0
$M_{23}$ iris	0.299	0.297	-0.002
$M_{14}$ iris	0.212	0.216	0.004
$M_{11}/M_{44}$ tuning screws	0.045	0.005	-0.040
$M_{22}/M_{33}$ tuning screws	0.133	0.135	0.002
$M_{12}/M_{34}$ coupling screws	0.111	0.115	0.004
Cavity length	1.865	1.864	-0.001

### 3.4.5 Example 5: A 6-pole Filter Design for Device Level Verification of Proposed Methods

In this example, we design a 6-pole waveguide filter using the proposed methodology. The specification of this 6-pole filter is different from that of Example 4. The filter center frequency is 12.155 GHz, bandwidth is 64 MHz and cavity diameter is chosen to be 1.072". This filter is higher in order and more complex in nature than that of Example 4. This filter uses an additional iris named slot iris. For this reason in addition to the neural

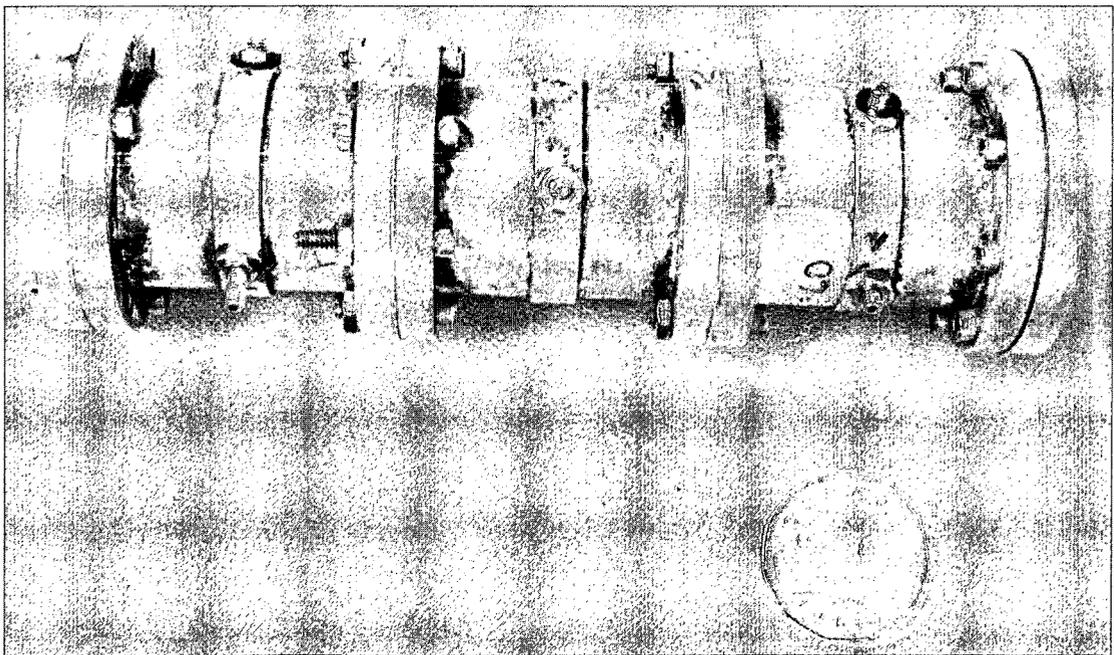
models of Examples 2, and 3, we developed another inverse model for slot iris. The inputs of the slot iris model are cavity diameter  $C_D$ , center frequency  $\omega_0$  and coupling  $M$  and the outputs are iris length  $L$ , vertical phase  $P_v$  and horizontal phase  $P_h$ . This model and the other three neural network inverse models developed in Examples 2 and 3 were used to design a filter. This filter is fabricated and measured for device level verification. The normalized ideal coupling values are

$$R_1 = R_2 = 1.077$$

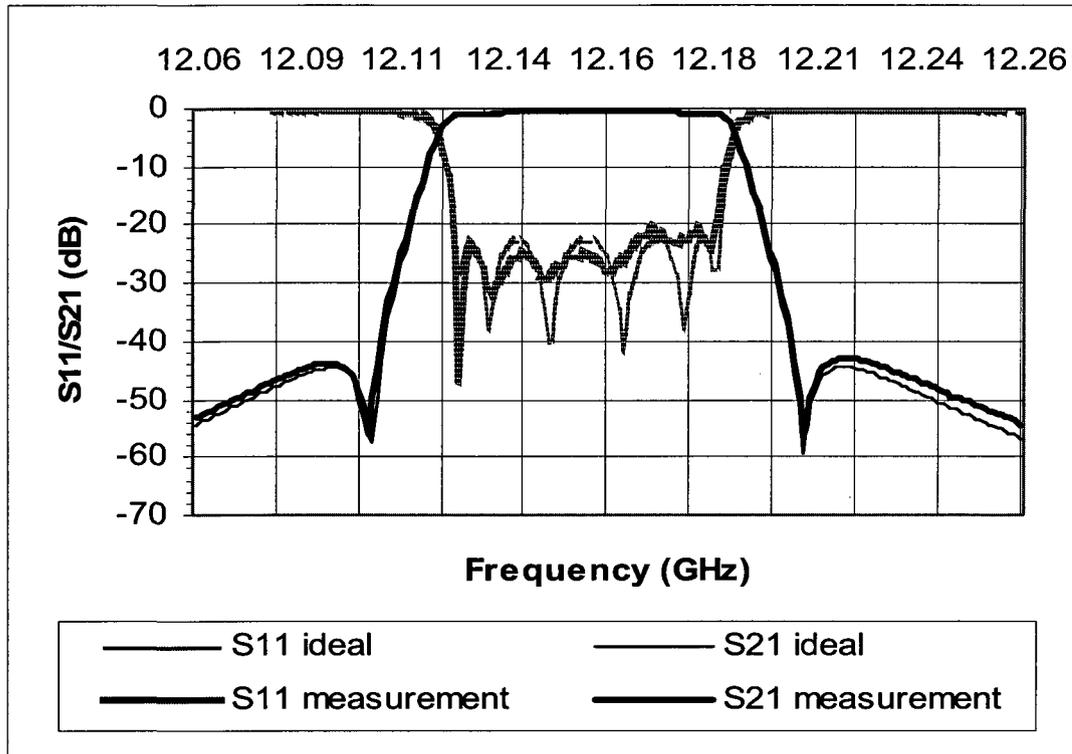
$$M = \begin{bmatrix} 0 & 0.855 & 0 & -0.16 & 0 & 0 \\ 0.855 & 0 & 0.719 & 0 & 0 & 0 \\ 0 & 0.719 & 0 & 0.558 & 0 & 0 \\ -0.16 & 0 & 0.558 & 0 & 0.614 & 0 \\ 0 & 0 & 0 & 0.614 & 0 & 0.87 \\ 0 & 0 & 0 & 0 & 0.87 & 0 \end{bmatrix} \quad (3.35)$$

After obtaining the filter dimensions from the inverse neural network models we manufactured the filter and tuned it by adjusting irises and tuning screws to match the ideal response. The picture of the fabricated filter is shown in Figure 3.11. Figure 3.12 presents the response of the tuned filter and compares with the ideal one showing a perfect match between each other. The dimensions of the tuned filter are measured and compared with the dimensions obtained from the neural network inverse models in Table 3.3, along with EM design results. From Table 3.3 we see that the neural network dimensions match the measurement dimensions very well. The quality of the solutions from the inverse neural networks is similar to that from the EM design, both being

excellent starting points for final tuning of the filter. The biggest error of screw dimensions, common for both the inverse neural network solution and the EM design, is observed in cavity 2, which is caused by the manufacturing error. The cavity length was manufactured short by 0.003" and that error affected the screw dimensions. In other words this error was compensated by tuning.



**Figure 3.11: Picture of the 6-pole waveguide filter designed and fabricated using the proposed neural network method.**



**Figure 3.12: Comparison of the 6-pole filter response with ideal filter response. The filter was designed, fabricated, tuned and then measured to obtain the dimensions.**

**Table 3.3: Comparison of dimensions obtained by the EM model, the neural network inverse models and the measurement of the tuned 6-pole filter**

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

The advantage of using the trained neural network inverse models is also realized in terms of time compared to EM models. An EM simulator can be used for synthesis, which requires typically 10 to 15 iterations to generate inverse model dimensions. Comparisons of time to obtain the dimensions using the EM and the trained neural network models are listed in Table 3.4. It shows that the time required by the neural network inverse models are negligible compared to EM models.

**Table 3.4: Comparison of time to obtain the dimensions by neural network inverse models and EM models**

Filter design approach	Model evaluation time (s)		
	IO iris	Coupling iris	Tuning screw
EM	15	120	240
Neural network	0.14E-3	0.1E-3	1.3E-3

### 3.5 Additional Discussion on Examples

In this Chapter, the three-layer multilayer perceptron neural network structure was used for each neural network model and quasi-Newton training algorithm was used to train the neural network models. Testing data are used after training the model to verify the generalization ability of these models. Automatic model generation algorithm of *NeuroModelerPlus* [127] was used to develop these models, which automatically train the model until model training, and testing accuracy is satisfied. The training error and

test errors are generally similar because sufficient training data was used in the examples.

The coupling value in this work is formulated as coupling bandwidth since they are the product of normalized coupling values and bandwidth. In this way bandwidth is no longer needed as a model input, help reducing training data and increasing model accuracy.

The tuning time is approximately the same for both the EM and the neural network design. Even though the EM method gives the best solution of a filter, physical machining process cannot guarantee 100% accurate dimension. Therefore after manufacturing the filter tuning is required. The amount of time spent on tuning also depends on how accurate the dimensions are. If the dimensions are far different from their perfect values, then tuning time will increase. The neural network method provides approximately the same dimension as the EM method. They both provide excellent starting points for tuning. As a result the tuning time is relatively short and is the same for both the EM and neural network methods. Consequently the tuning time does not alter the comparison between the EM and neural network method.

The training time for the direct inverse tuning screw model is approximately 6 minutes. In the proposed algorithm if we perform segmentation it will add 28.5 seconds, and if multivalued solutions are detected in a segment, it adds another 7.5 seconds for a forward model for a small segment containing 200 samples. The training time for the complete inverse tuning screw model using the proposed methodology is approximately 5.5 hours. For coupling iris the direct inverse model containing 37000 samples takes 26 minutes to train. The proposed method divides the model into 4 smaller segments, each

containing approximately 9000 samples, and takes 10 additional minutes per segment. The time to train a direct IO iris inverse model containing 125000 data requires 2.5 hours. The training time using proposed methodology is 6 hours including the time for training segmented models. The training time for these models were obtained using *NeuroModelerPlus* parallel-automated model generation algorithm [127] on an Intel Quad core processor at 2.4GHz. The training time for proposed inverse models is longer than that of the direct inverse models. However, once the models are trained, the proposed model is very fast for the designer, providing solutions nearly instantly.

The technique is useful for highly repeated design tasks such as designing filters of different orders and different specifications. The technique is not suitable if the inverse model is for purpose of only one or a few particular designs, because the model training time will make the technique cost-ineffective. Therefore the technique should be applied to inverse tasks, which will be re-used frequently. In such case the benefit of using the models far out-weights the cost of training because of four reasons: (a) Training is a one-time investment, and the benefit of the model increases when the model is used over and over again. For example, the two different filters in the Examples use the same set of iris and tuning screw models. (b) Conventional EM-design is part of the design cycle while neural network training is outside design cycle. (c) Circuit design requires much human involvement, while neural network training is a machine-based computational task. (d) Neural network training can be done by a model developer and the trained model can be used by multiple designers. The neural network approach cuts expensive design time by shifting much burden to off-line computer-based neural network training [2]. An even

more significant benefit of the proposed technique is the new feasibility of interactive design and what-if analysis using the instant solutions of inverse neural networks, substantially enhancing design flexibility and efficiency. The use of inverse models to find the design variables is also advantageous over optimization using a fast neural network forward model. Because, optimization may suffer from convergence problems in a complex design optimization task. Also, it still requires many evaluations of the forward model.

### **3.6 Summary**

In this Chapter, two major contributions of the thesis have been presented. Efficient neural network modeling techniques have been presented and applied to microwave filter modeling and design. The inverse modeling technique has been formulated and non-uniqueness of input-output relationship has been addressed. Methods to identify multivalued solutions and divide training data have been proposed for training inverse models. Data of inverse model have been divided based on derivatives of forward model and then been used separately to train more accurate inverse sub-models. A method to correctly combine the inverse sub-models has been presented. The inverse models developed using the proposed techniques have become more accurate than that using the direct method. A design approach using the inverse models has been proposed. The proposed methodology has been applied to waveguide filter modeling and design. Very good correlation has been found between neural network predicted dimensions and that of perfectly tuned filters. This modeling approach has been proven useful for fast solution to inverse problems in microwave design.

## **Chapter 4: High Dimensional Neural Network Techniques and Application to Microwave Filter Modeling**

In this Chapter, another major contribution of the thesis is presented. We propose an effective method for developing high dimensional neural network model for microwave filters. This novel method is suitable for developing neural network models for microwave structures that have many design variables. A structure decomposition approach is proposed to divide the high dimensional problem into several low dimensional subproblems. Formulation of the submodels is proposed in such a way that the neural network models can be trained to learn geometrical parameter to equivalent circuit parameter or S-parameter depending on the complexity of the behaviour of the structure. Neural network submodels for the substructures are then developed conveniently. A method is then proposed to combine the submodels with equivalent circuit model which produces an approximate solution of the overall microwave structure. An additional neural network is formulated to map the approximate solution to the accurate solution. The overall high dimensional model is obtained by combining the neural network submodels, equivalent circuit model, and neural network mapping model. An algorithm to develop the high dimensional modeling efficiently is proposed. The proposed modeling approach is validated through high dimensional filter modeling

example which shows that the proposed modeling approach is advantageous of producing high dimensional model otherwise impractical to obtain using conventional neural network approach.

## 4.1 Introduction

Due to the increased complexity and variety of microwave structures, the number of design variables per structure is on the rise. In order to develop an accurate neural network model that can represent EM behavior of filters over a range of values of geometrical variables, we need to provide EM data at sufficiently sampled points in the space of geometrical variables [1], [2]. The amount of data required increases very fast with the number of input variables of the model. For this reason, developing a neural network model that has many input variables becomes challenging as data generation becomes too expensive. Therefore, we need an effective method to develop accurate neural network high dimensional models without requiring massive data.

Various advanced neural network structures have been investigated for microwave modeling such as knowledge-based neural network [13], [14] for simplifying input-output relationship. It reduces the cost of neural network training for highly nonlinear input-output modeling problems. However, it does not have the mechanism to address the challenge of high dimensional modeling problems directly. Modular neural network is an interesting technique which has the potential to address high dimensional modeling problem because of neural network decomposition. It has been investigated within artificial neural network community for applications such as face detection [128],

[129], voice recognition [130], pattern recognition [131], [132], directional relay algorithm for power transmission line [133], problem simplification [134], etc. This technique decomposes a complex neural network into several simple sub-neural network modules. The modular neural network technique has been used to improve the learning capability of neural networks. However, the existing modular neural network method is not directly suitable for high dimensional neural network modeling of microwave filters, because it has not been formulated to accommodate the knowledge of microwave filter formulas. Another problem with the existing neural network decomposition is the absence of connections between neural network decomposition and microwave filter decomposition.

Recently, microwave filter has been modeled and designed using neural network techniques [17]. The main objective of [17] is to produce neural network inverse model of filter components so that we can avoid repetitive EM model evaluation for fast design. Neural network inverse submodels produce approximate values of the filter dimensions from a given coupling matrix, which is used as a starting point of the filter design. In this Chapter, we propose a new method to obtain a complete model as accurate as EM model for an entire filter with many input variables. We propose a new formulation to integrate neural network decomposition with filter structure decomposition and then incorporate circuit knowledge to obtain a complete filter model. We start with decomposing an overall filter structure into substructures, which reduces the number of input variables per submodel. As a result, data generation for the submodels becomes inexpensive. This allows us to develop neural network submodels conveniently. The developed neural

network submodels are then incorporated with empirical/equivalent circuit model to obtain the response of the overall filter. However, decomposition causes the submodels to lose exact details of the overall filter and when combined with empirical/equivalent circuit model, they only produce an approximate solution of the overall filter. For this reason, an additional neural network model is trained to map the approximate solution to the accurate EM solution.

Data generation of an overall filter is expensive. Conventional neural network method requires many samples of an overall filter during training to achieve good accuracy, whereas the proposed method requires only a few samples of the overall filter to achieve the same. For this reason, the proposed method becomes significantly less expensive than the conventional method. The new method is used to develop complex filter models that hold many input variables. Results show that using the proposed method, we can develop accurate high dimensional neural network models in inexpensive way. The evaluation time of the proposed neural network model is faster than that of the EM model. This makes the proposed method effective and useful for design optimization where many geometrical design variables need to be changed and EM behavior need to be evaluated repetitively.

## **4.2 Proposed High Dimensional Modeling Approach**

### **4.2.1 Problem Statement**

The main objective is to obtain fast parametric models for filters that hold many design

variables which are mainly geometrical parameters. Let us assume  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T$  to be an  $n$ -vector containing all the input variables of a model, e.g., iris length, cavity length, bandwidth etc. for a filter. Let  $\mathbf{y} = [y_1 \ y_2 \ y_3 \ \dots \ y_n]^T$  be an  $m$ -vector containing output parameters such as S-parameter of the filter. A conventional neural network model for the problem is defined as

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{w}) \quad (4.1)$$

where  $\mathbf{f}$  defines the input-output relationship and  $\mathbf{w}$  is a neural network internal weight vector. In this approach, we use a multilayer-perceptron or a radial-basis-function neural network [2] to represent the entire function of (4.1), with  $\mathbf{x}$  represented by input neurons, and  $\mathbf{y}$  represented by output neurons. This conventional approach is suitable for developing simple filter models where the number of input variables is small. On the other hand, when a filter model has many input variables, massive amount of data are required for neural network model training to achieve good accuracy. This massive data generation and model training become too expensive and impractical. To overcome this limitation, we propose to use the decomposition approach to simplify the high dimensional problem into a set of small subproblems. Let  $f_1$  to  $f_{N_{sub}}$  represent  $N_{sub}$  simple subfunctions which define the input-output relationships of a set of simple functions representing various partial information of  $\mathbf{f}(\cdot)$  of (4.1). Each of the subfunctions is defined by small number of input variables and the input-output relationship becomes

simpler than the overall high dimensional function. In this way, cost of data generation and model development is reduced. However, the definition of partial information or the formulation of neural network submodels will not be effective unless we combine filter decomposition concept with neural network decomposition. Furthermore, the question of how to recombine the submodels to form the final overall filter model and recover the missing information between subproblems must be answered for the neural network decomposition.

#### 4.2.2 Neural Network Submodels

We formulate neural network decomposition together with filter decomposition. A filter with many design variables is decomposed into several substructures each representing a specific part of the filter. Neural network submodels are then developed to represent each substructure. Let us assume that a filter is decomposed into  $N_{sub}$  types of substructures. Let  $\tilde{\mathbf{x}}_i$  be a vector containing the design variables of the  $i^{\text{th}}$  substructure and  $\mathbf{z}_i$  be a vector containing the output parameters of the  $i^{\text{th}}$  substructure. As an example, the input vector  $\tilde{\mathbf{x}}_i$  contains geometrical parameters such as length and width of an iris and the output vector  $\mathbf{z}_i$  contains electrical parameters such as coupling coefficients of the iris. A neural network submodel for the substructure is defined as

$$\mathbf{z}_i = \mathbf{f}_i(\tilde{\mathbf{x}}_i, \mathbf{w}_i) \quad (4.2)$$

where  $\mathbf{f}_i$  defines the geometrical to electrical relationship of the  $i^{\text{th}}$  submodel,  $\mathbf{w}_i$  is a

vector containing neural network weight parameters for the  $i^{\text{th}}$  submodel, and  $i = 1, 2, 3, \dots, N_{sub}$ . The vector  $\tilde{\mathbf{x}}_i$  is a subset of the overall input vector  $\mathbf{x}$  and is expressed as

$$\begin{aligned}\tilde{\mathbf{x}}_i &\subset \mathbf{x} \\ \tilde{\mathbf{x}}_i &= \mathbf{Q}_i \mathbf{x}\end{aligned}\tag{4.3}$$

where  $\mathbf{Q}_i$  is a selection matrix containing 1s and 0s in order to pick corresponding inputs of submodel  $i$  from the overall input vector  $\mathbf{x}$ .

In order to formulate meaningful submodels for filter applications, we need to combine filter decomposition concepts with the submodel of (4.2) and (4.3). In microwave waveguide filters, the electrical couplings between various sections of the filter is dominantly determined by the physical/geometrical parameters of the corresponding parts of the filter structure, and slightly affected by geometrical parameters of other sections [17], [135]. Based on this concept, we use the  $\mathbf{Q}_i$  matrix to select the geometrical parameters of the relevant part of the filter ignoring other parts, and use  $z_i$  to represent the electrical coupling between the selected parts of the filter.

Data for each submodel is generated using an EM simulator and neural network submodels are then trained. Let us assume  $N_i^s$  to be the number of training samples required to develop neural network submodel  $i$ . The submodel is developed by optimizing internal weight vector  $\mathbf{w}_i$  to minimize the error between outputs and training data. The training error of submodel  $i$  is expressed as

$$E_i^s = \sum_{k=1}^{N_i^s} \left\| \left( \mathbf{f}_i(\tilde{\mathbf{x}}_i^k, \mathbf{w}_i) - \mathbf{d}_i^k \right) \right\|^2 \quad (4.4)$$

where vector  $\tilde{\mathbf{x}}_i^k$  is the  $k^{\text{th}}$  sample of the training data for input neurons of the  $i^{\text{th}}$  submodel which contains the values of geometrical parameters of the  $i^{\text{th}}$  substructure, and vector  $\mathbf{d}_i^k$  is the  $k^{\text{th}}$  sample of the training data for output neurons which contains the EM solution of the  $i^{\text{th}}$  substructure. Data generation for submodels becomes less expensive than that for the overall filter model. Because, the submodels contain fewer input variables than the overall filter model and the input-output relationships of the submodels become simpler than that of the overall filter model.

#### **4.2.3 Integration of Neural Network Submodels with Empirical/Equivalent Circuit Model**

The neural network submodels should be recombined to form the overall filter model. Here, we formulate an approach where a filter empirical/equivalent circuit model is used to obtain the solution of the overall filter by using the outputs from the neural network submodels. Some of the neural network submodels may be used multiple times as the same junction may appear several times in the overall model. For example, in a four pole H-plane filter there are three internal irises. We can develop one model of internal iris and use it three times. Because, the iris submodel is trained with a range of values of length, different iris submodels can be represented with the same neural network iris submodel with different values of  $\tilde{x}_i$ . Multiple uses of submodels become a big advantage

of the proposed method. In this way, we can obtain all the submodels needed for an overall filter model by training only a few neural network submodels. Let  $N_o$  be the number of neural network submodels needed to form the overall filter model. The equivalent circuit model is expressed in terms of the outputs of the neural network submodels as

$$\mathbf{y}^a = \mathbf{f}_q(z_1, z_2, \dots, z_{N_o}) \quad (4.5)$$

where  $\mathbf{y}^a$  is a vector containing approximate values of the outputs of the overall filter,  $\mathbf{f}_q$  represents the empirical/equivalent circuit function, and  $z_1$  to  $z_{N_o}$  are electrical parameters obtained from  $N_o$  submodels.

The type of operation in (4.5) is simple and insignificant in terms of computational cost. Thus, an approximate solution of the overall filter is obtained by combining neural network submodels and empirical/equivalent circuit model.

#### 4.2.4 Neural Network Mapping Model for Accuracy Improvement

The outputs from the neural network submodels provide values of the electrical parameters (e.g., coupling matrix for a filter), which are approximate since effects of high order modes are lost due to decomposition of the overall filter. Thus, the solution obtained from the empirical/equivalent circuit model is also approximate. Here, we propose an additional neural network model, called neural network mapping model, to map the approximate solution to the accurate EM solution of the overall filter. Samples of the overall filter are generated to obtain the training data for the mapping model. Based

on the concept of prior-knowledge input [14], we formulate the inputs of the mapping model using the approximate solution  $y^a$ , and the input variables of the overall filter,  $x$ . The outputs are the accurate solution of the overall filter  $y$  that corresponds to  $x$ . Thus, the neural network mapping model is defined as

$$y = f_M(x, y^a, w_M) \quad (4.6)$$

where  $f_M$  defines the input-output relationship of the mapping model and  $w_M$  is a vector containing neural network internal weight parameters. Let us assume that we need  $N^M$  samples of the overall filter to train the mapping model accurately. The neural network mapping model is developed by minimizing the error between EM data and neural network output by optimizing neural network internal weight parameters. The training error of the mapping model is expressed as

$$E^M = \sum_{k=1}^{N^M} \left\| \left( f_M(x^k, y^a|_{x=x^k}, w_M) - D^k \right) \right\|^2 \quad (4.7)$$

where  $D^k$  is the  $k^{\text{th}}$  sample of training data for output neurons and which is the EM solution of the overall filter.

The mapping function in (4.6) becomes simple since it is defined with an approximate solution. For this reason, the neural network mapping model can be developed accurately with a few samples of the overall filter. In this way, the number of expensive EM simulation of the overall filter is reduced. As a result, data generation and

model training in the proposed method become feasible. The mapping model can be a single model or a set of models each representing an individual output parameter of the overall model.

#### 4.2.5 Overall Modeling Structure

An accurate high dimensional model representing the overall filter is constructed by combining the neural network submodels, circuit model and neural network mapping model. The diagram of the overall high dimensional modeling structure is presented in Figure 4.1.

The neural network mapping model as defined in (4.6) can be expressed in terms of equivalent circuit model of (4.5) as

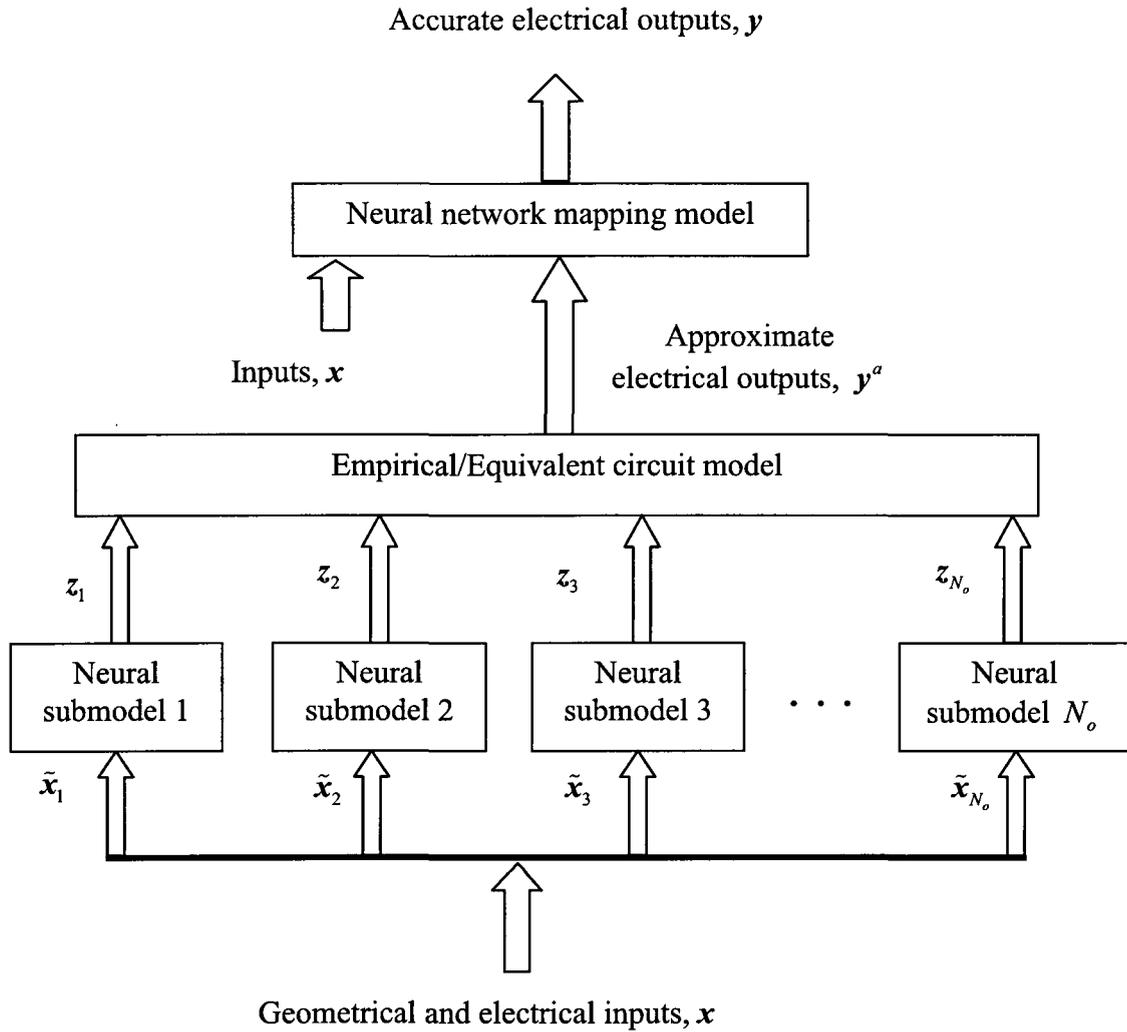
$$\mathbf{y} = \mathbf{f}_M \left( \mathbf{x}, \mathbf{f}_q(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{N_o}), \mathbf{w}_M \right). \quad (4.8)$$

We can further express (4.8) in terms of neural network submodels defined in (4.2) as

$$\mathbf{y} = \mathbf{f}_M \left( \mathbf{x}, \mathbf{f}_q \left\{ \mathbf{f}_1(\tilde{\mathbf{x}}_1, \mathbf{w}_1), \mathbf{f}_2(\tilde{\mathbf{x}}_2, \mathbf{w}_2), \dots, \mathbf{f}_{N_o}(\tilde{\mathbf{x}}_{N_o}, \mathbf{w}_{N_o}) \right\}, \mathbf{w}_M \right). \quad (4.9)$$

Substituting the relationship of (4.3) in (4.9) yields

$$\mathbf{y} = \mathbf{f}_M \left( \mathbf{x}, \mathbf{f}_q \left\{ \mathbf{f}_1(\mathbf{Q}_1 \mathbf{x}, \mathbf{w}_1), \mathbf{f}_2(\mathbf{Q}_2 \mathbf{x}, \mathbf{w}_2), \dots, \mathbf{f}_{N_o}(\mathbf{Q}_{N_o} \mathbf{x}, \mathbf{w}_{N_o}) \right\}, \mathbf{w}_M \right), \quad (4.10)$$



**Figure 4.1: Diagram of the proposed high dimensional modeling structure.**

which is equivalent to

$$y = f_M(x, w_o) \quad (4.11)$$

where  $w_o$  is a vector containing neural network internal weight parameters of the high

dimensional neural network model. In (4.10), the vectors  $w_1$  to  $w_{N_o}$  contain weight parameters of neural network submodels and vector  $w_M$  contains weight parameters of neural network mapping model. These vectors are optimized during neural network training of submodels and mapping model. The vector  $w_M$  is optimized after the optimization of the vectors  $w_1$  to  $w_{N_o}$ . When the overall high dimensional model is constructed combining the trained neural network submodels and mapping model, the vectors  $w_1$  to  $w_{N_o}$  and  $w_M$  all together become equivalent to the vector  $w_o$  of (4.11).

The relationship of (4.11) is equivalent to that of (4.1) except (4.11) is a combination of several simple submodels each with few input variables, whereas (4.1) is a single complicated model with many input variables. The vector  $w$  of (4.1) is equivalent to the vector  $w_o$  of (4.11). The difference is that the vector  $w_o$  is optimized step by step through neural network submodels and mapping model training. Thus, in the proposed method, combination of several low dimensional submodels, circuit model, and neural network mapping model produce the overall high dimensional model.

In the proposed method, a few expensive data of the overall filter are needed for the neural network mapping model as explained before. On the other hand, in the conventional method, many expensive data are required to achieve reasonable accuracy because of two reasons: i) the model is a single function of many input variables as defined in (4.1), and ii) the relationship of (4.1), which relates geometrical to circuit parameters directly, is complicated.

Let  $t_o$  represent data generation time per sample of an overall filter. Let

$N^c$  represent the number of samples of data of the overall filter required for the neural network model in the conventional approach. The cost of data generation in the conventional method is expressed as

$$T^c = t_o \times N^c. \quad (4.12)$$

Let  $t_i$  represent data generation time per sample for submodel  $i$ ,  $i = 1, 2, 3, \dots, N_{sub}$ . As defined before, we assume  $N_i^s$  and  $N^M$  represent the number of samples of data required to develop neural network submodel  $i$  and mapping model, respectively. The cost of data generation in the proposed method is expressed as

$$T^P = t_o \times N^M + \sum_{i=1}^N (t_i \times N_i^s) \quad (4.13)$$

where  $i = 1, 2, \dots, N_{sub}$  and  $N_{sub}$  is the number of types of substructures decomposed from an overall structure as defined before in Section 4.2.2. Data generation time per sample of the overall filter is much more expensive than that of a submodel, i.e.,  $t_o \gg t_i$ . Also as explained before, the mapping function becomes simple since it is defined with an approximate solution. Thus, the proposed method requires much less data of the overall filter, i.e.,  $N^M \ll N^c$ . For these reasons, the data generation cost of the proposed method ( $T^P$ ) becomes less than that of the conventional method ( $T^c$ ), i.e.,

$$T^P \ll T^c. \quad (4.14)$$

Training time increases with the number of model input variables, number of hidden neurons and number of training data. The number of input variables for submodels is low. The input-output function is also simple which translates into low number of hidden neurons. For these reasons, training time for submodel becomes short. The mapping function is simple as explained before and since  $N^M \ll N^c$ , the training time of the mapping model is also short. As a result, the total model training time of the proposed method becomes much less than that of the conventional method, i.e.,

$$\Gamma^P \ll \Gamma^c \quad (4.15)$$

where  $\Gamma^P$  and  $\Gamma^c$  represent model training time of the proposed and the conventional method. Combining (4.14) and (4.15) yields

$$T^P + \Gamma^P \ll T^c + \Gamma^c. \quad (4.16)$$

This describes that the total time for data generation and model training of the proposed method is much less than those of the conventional method.

### 4.3 Algorithm for Proposed High Dimensional Model Development

We describe an overall high dimensional modeling algorithm. The flow diagram of the algorithm is presented in Figure 4.2.

The steps are described as follows:

Step 1. Identify the parts of an overall filter that can be used as substructures. For a waveguide filter, discontinuities can be decomposed into substructures. Decompose the overall filter into substructures.

Step 2. Generate training data of the decomposed substructures using EM simulations. Standard sampling approach can be employed for this purpose.

Step 3. Train and test neural network submodels for all the decomposed substructures.

Step 4. If the submodels are accurate, go to the next Step. Else, generate some more data of the substructures by sampling intermediate points using EM simulation, add those to the existing data, and go to the Step 3.

Step 5. Generate a few data of the overall filter using EM simulation. Sweep the input variables ( $x$ ) and obtain corresponding output solutions ( $y$ ) of the overall filter.

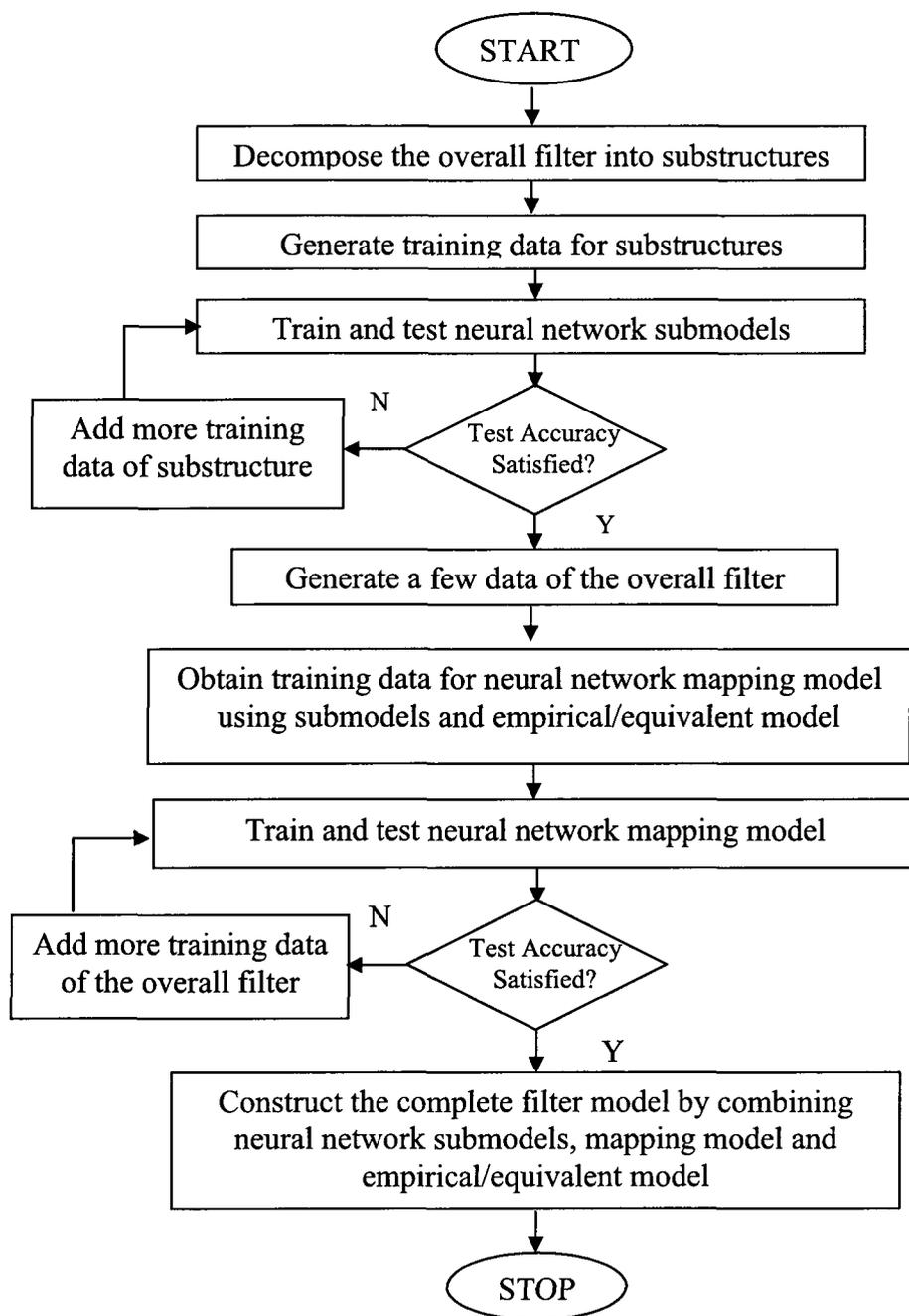
Step 6. Combine neural network submodels and empirical/equivalent circuit model.

Step 7. Supply the samples of the input variables ( $x$ ) to the combined neural network submodels and empirical/equivalent circuit model to obtain samples of approximate solution ( $y^a$ ) of the overall filter.

Step 8. Using the concept of prior knowledge input [14], assemble training data for the mapping model. Use the samples of  $x$  and  $y^a$  of Step 7 as the data for the input neurons. Use the samples of  $y$  that corresponds to the samples of  $x$  as the

data for the output neurons. Train neural network mapping model using some of the assembled data. Test the mapping model with the rest of the data. If accuracy is satisfied, go to the next Step. Else, generate a few more data of the overall filter, add those to the existing data of the overall filter and go to Step 7.

Step 9. Combine neural network submodels, empirical/equivalent circuit model, and neural network mapping model as described in Section 4.2.5 to obtain the overall model of the filter.



**Figure 4.2: Flow diagram of the proposed high dimensional neural network modeling approach.**

## 4.4 Modeling Examples

### 4.4.1 Illustration of the Proposed Modeling Techniques for an H-Plane Filter

We illustrate the proposed modeling method through a 4-pole H-plane filter model development. The diagram of the filter is shown in Figure 4.3. The filter model has eight variables as inputs which include five geometrical variables: iris widths  $W_1$ ,  $W_2$ , and  $W_3$ , cavity lengths  $L_{b1}$  and  $L_{b2}$ , and three electrical variables: bandwidth  $B$ , center frequency  $\omega_o$ , and frequency  $\omega$ . The filter outputs are S-parameters  $S_{11}$  and  $S_{12}$ . Thus, the input and output vector of the filter model is

$$\mathbf{x} = [W_1 \ W_2 \ W_3 \ L_{b1} \ L_{b2} \ B \ \omega_o \ \omega]^T \quad (4.17)$$

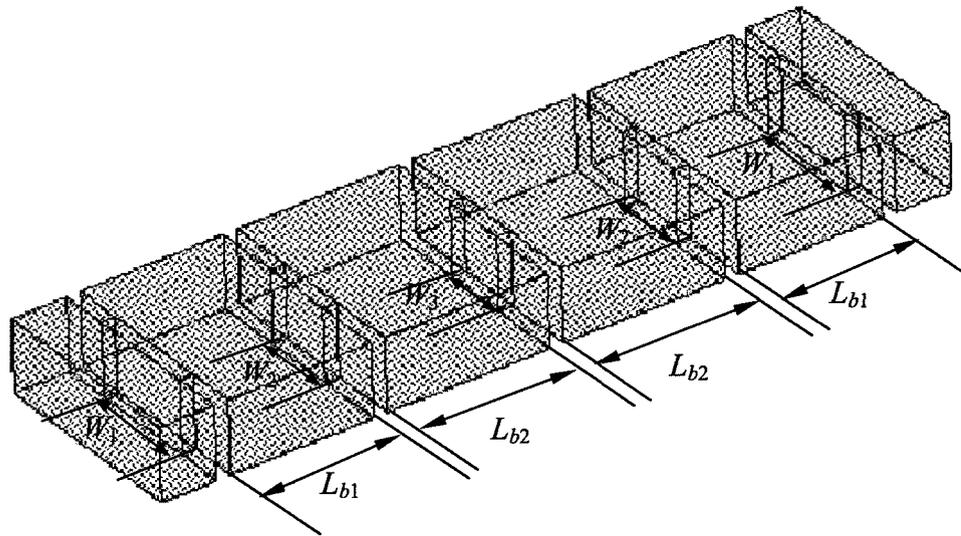
$$\mathbf{y} = [S_{11} \ S_{12}]^T \quad (4.18)$$

We first decompose the waveguide filter into two types ( $N_{sub} = 2$ ) of substructures: input-output iris and internal coupling iris. We will develop two neural network submodels of the two substructures in the next Step. Each submodel contains two input variables: width of iris  $W$  and center frequency  $\omega_o$ . We use coupling and phase length as the output parameters of the submodels [17]. Thus, the input and output vectors of the submodels are

$$\tilde{\mathbf{x}}_i = [W_i \ \omega_o]^T \quad (4.19)$$

$$\mathbf{z}_i = [M_i^a \ P_i^a]^T \quad (4.20)$$

where  $i = \{1, 2\}$ ,  $M_i^a$  and  $P_i^a$  represent approximate values of coupling parameter and phase length of the  $i^{\text{th}}$  submodel. Notice that the number of input variables of each submodel as expressed in (4.19) is less than that of the overall model as expressed in (4.17).



**Figure 4.3: Diagram of a 4-pole H-plane filter. The filter model holds eight input variables including five geometrical dimensions, bandwidth, center frequency, and sweeping frequency.**

In this Step, we develop two neural network submodels for the two types of irises. We generate training data by simulating the substructures using EM simulator based on mode-matching method. The S-parameters are then used to calculate the coupling values and phase lengths following the same steps and equations presented in [17]. We generate

35751 samples which cover a large range of iris width and center frequency for each submodel. Data generation time per sample for each of the submodels is 0.6 s which is inexpensive as the input-output relationships are simple and the submodels hold only two input variables each. Training time for each submodel is less than 1 minute. The average errors of the submodels are less than 1%. Automatic model generation module of *NeuroModelerPlus* [127] is used to develop the two neural network submodels.

Following the Step 5 of the modeling algorithm in Section 4.3, we generate data of the overall filter using EM simulator. EM data are generated simulating 46 different filters. In the next Step, we combine the neural network submodels and filter equivalent circuit model as shown in Figure 4.4 to obtain the approximate S-parameter of the filter. Note that the input-output iris model is used twice and the internal coupling iris model is used three times to represent the overall 4-pole filter, i.e.,  $N_o = 5$ . In other words, 5 submodels required in the filter are obtained by training only 2 submodels. The neural network submodels produce approximate coupling matrix and subsequently, circuit model generates approximate S-parameters of the 4-pole filter using the following equation [135]:

$$\begin{aligned} S_{11}^a &= 1 + 2jR_1^a [\lambda \mathbf{I} - j\mathbf{R}^a + \mathbf{M}^a]_{11}^{-1} \\ S_{12}^a &= -2j\sqrt{R_1^a R_2^a} [\lambda \mathbf{I} - j\mathbf{R}^a + \mathbf{M}^a]_{g1}^{-1} \end{aligned} \quad (4.21)$$

in which  $\lambda = \frac{\omega_o}{B} \left( \frac{\omega}{\omega_o} - \frac{\omega_o}{\omega} \right)$ ,  $g$  is the filter order and  $g = 4$  in this case,  $\mathbf{I}$  is a  $g \times g$  identity matrix,  $\mathbf{M}^a$  is the  $g \times g$  approximate coupling matrix,  $\mathbf{R}^a$  is a  $g \times g$  matrix with all entries

zero except  $[\mathbf{R}^a]_{11} = R_1^a$  and  $[\mathbf{R}^a]_{gg} = R_2^a$ ,  $R_1^a$  and  $R_2^a$  are approximate values of the filter's input and output coupling parameters, respectively.

In Step 7, we supply the geometrical values of 46 filters used in Step 5 to the combined neural network submodels and filter empirical/equivalent model and obtain approximate S-parameter by sweeping frequency from 10.95 GHz to 13.05 GHz with 1 MHz step. The center frequency is held constant at 12 GHz and bandwidth is swept from 50 MHz to 500 MHz with 10 MHz step. The model outputs at this stage are

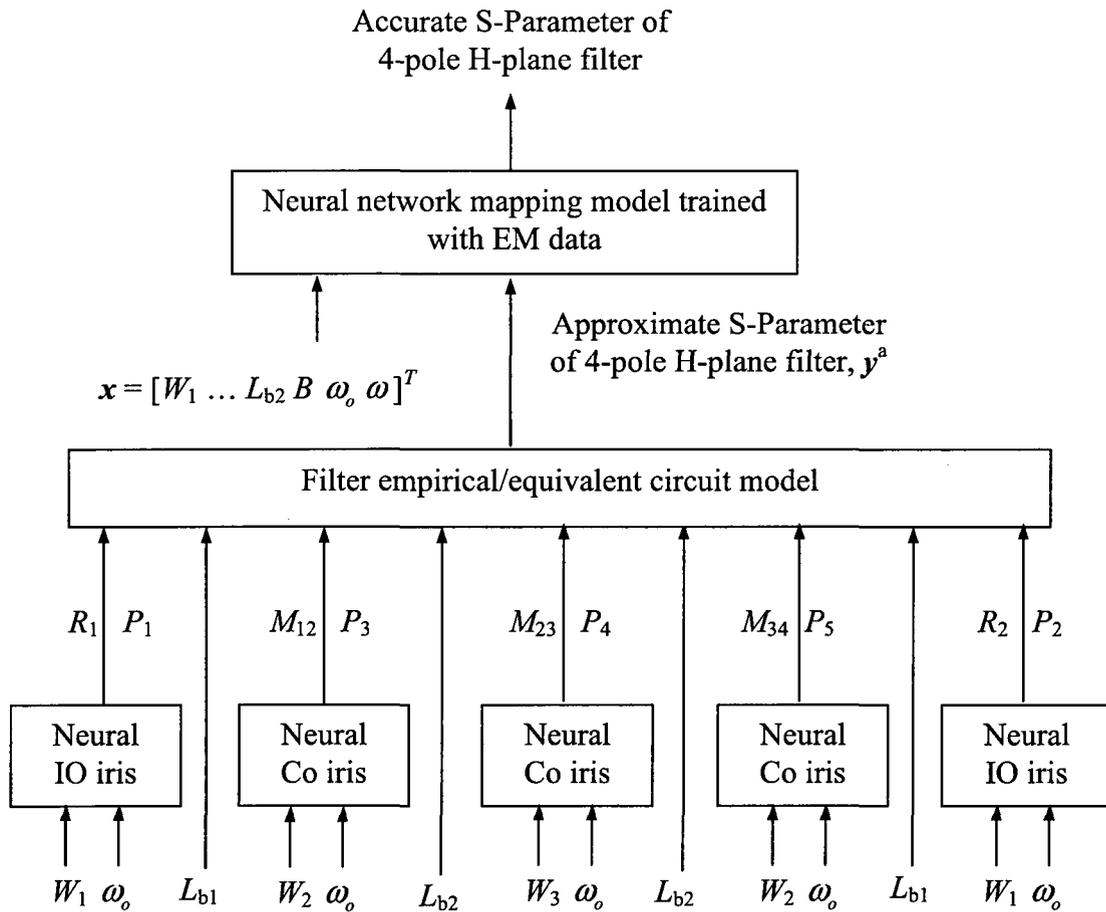
$$\mathbf{y}^a = [S_{11}^a \ S_{12}^a]^T \quad (4.22)$$

where the superscript "a" denotes that the values are approximate.

As described in Step 8 of the modeling algorithm in Section 4.3, we assemble training data for the input neurons of the mapping model using the input samples  $\mathbf{x}$  of Step 5 and approximate output samples,  $\mathbf{y}^a$  obtained in Step 7 of the 46 filters. The training data for the output neurons are the accurate S-parameter of the 46 filters generated using EM simulation in Step 5. These data are then used to train and test neural network mapping model, which maps the approximate S-parameter to the accurate S-parameter. Four different sets of training and testing data as shown in Table 4.1 are used to develop four mapping models. In Set 1, we use data of 23 filters for training and data of 23 other filters for testing. Training samples are reduced and testing samples are increased in the subsequent sets. The training error of the mapping models are less than

0.5%.

After the mapping model is trained, we construct the complete model of a 4-pole filter using neural network submodels, circuit model and mapping model in *NeuroModelerPlus* as shown in Figure 4.4. The model is then used for testing purpose. For comparison, we develop four neural network models following conventional method and using the same four sets of data used in the proposed method. In conventional method neural network model is trained to learn the complicated relationship between geometrical variables and S-parameter directly. The result is summarized in Table 4.1, which shows that the proposed method produces more accurate result than the conventional method. The amount of data is not enough to produce the eight dimensional parametric model of H-plane filter in the conventional method. On the other hand, the proposed method converts the overall function into a set of simple subfunctions and thus is able to produce the accurate model with those limited training data.

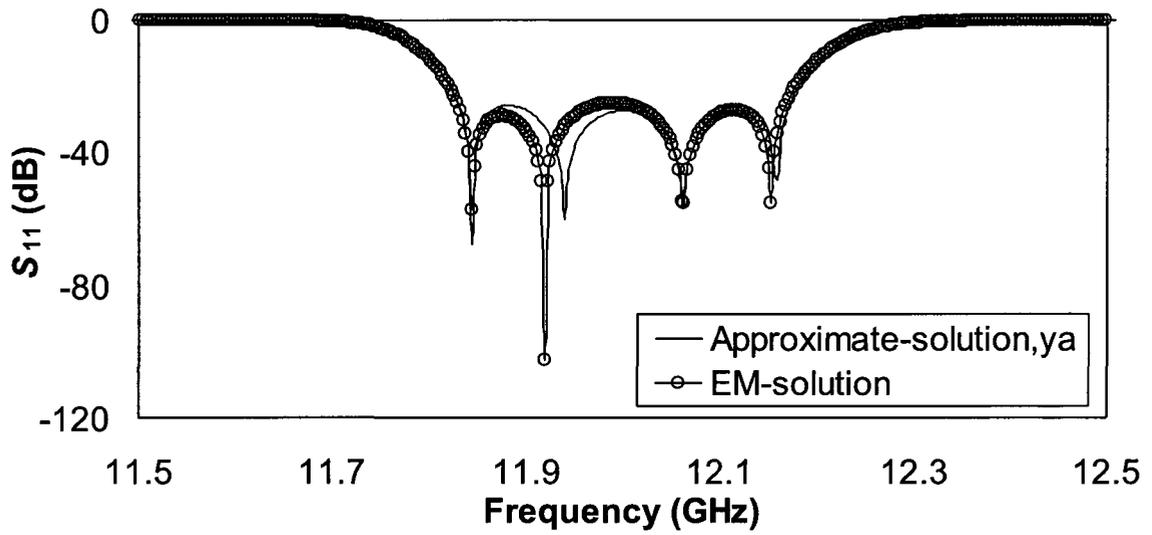


**Figure 4.4: High dimensional modeling structure for the 4-Pole H-plane filter. Two neural network submodels: input-output iris model (IO iris) and coupling iris model (Co iris) are developed decomposing the filter. Five submodels required by the overall filter as shown in this figure are obtained by training only 2 neural network submodels. Equivalent circuit model of a filter are used to obtain the approximate S-parameter. A neural network mapping model is then used to obtain the accurate S-parameter of the 4-pole H-plane filter.**

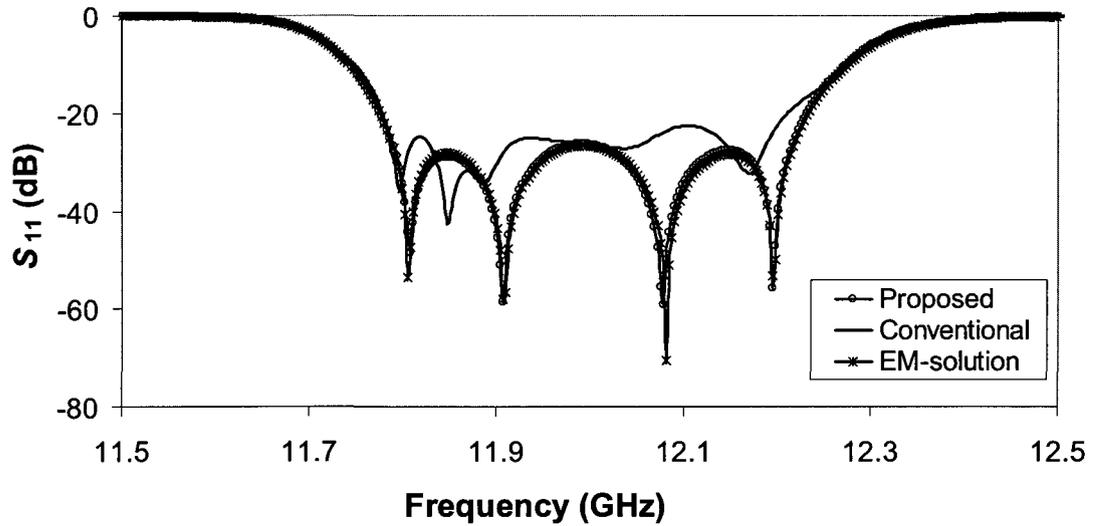
**Table 4.1: Comparison of test errors of 4-pole H-plane filter models developed using conventional and proposed high dimensional modeling approach**

Data Set no.	No. of filter geometries used		Modeling method	Model testing error (%)	
	Train	Test		Least square error	Worst case error
1	23	23	Conventional	2.60	41
			Proposed	0.48	8.6
2	13	33	Conventional	2.26	45
			Proposed	0.55	9.2
3	6	40	Conventional	2.40	45
			Proposed	2.10	25
4	3	43	Conventional	25	212
			Proposed	18	55

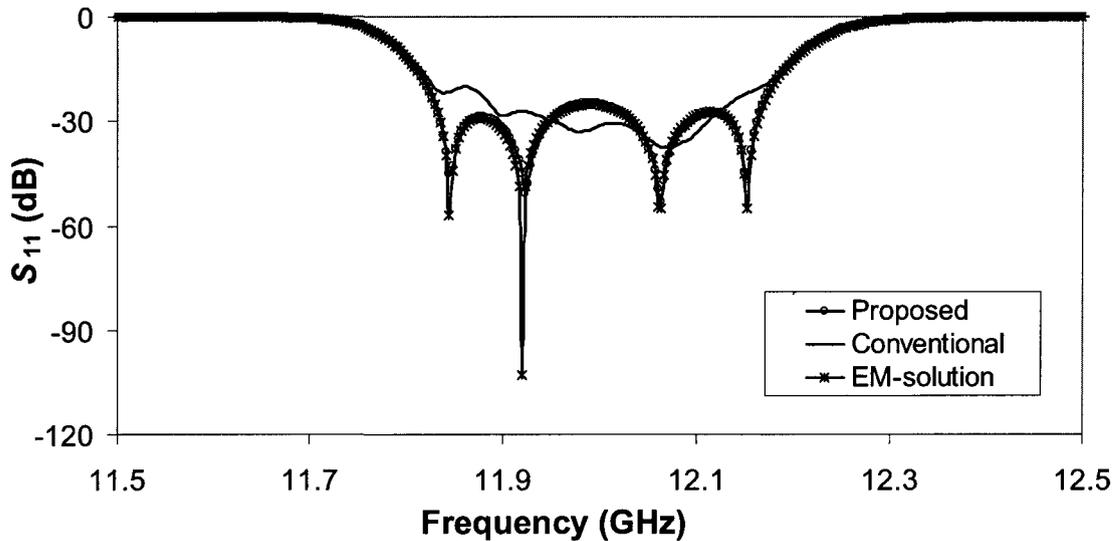
In Figure 4.5, we compare approximate S-parameter of an H-plane filter with its accurate S-parameter. The approximate solution obtained from the neural network submodels and empirical/equivalent circuit model combined is fairly close to the accurate EM solution. For this reason, the input-output relationship of the mapping model becomes simpler than the original modeling relationship between geometrical variables and S-parameters. Figure 4.6 shows 4-pole filter responses from conventional neural network model, proposed model and EM simulation of two different geometrical configurations. In both cases the proposed method produces more accurate result than conventional method.



**Figure 4.5: Comparison of approximate solution with EM- solution of a 4-pole H-plane filter. The approximate solution is obtained without using the mapping model of the proposed method. The similarity between the solutions confirms that a simple mapping using a few training data of overall filter can map the  $y^a$  to accurate EM solution. Filter geometry:  $L_{b1} = 0.54''$ ,  $L_{b2} = 0.60''$ ,  $W_1 = 0.37''$ ,  $W_2 = 0.23''$ ,  $W_3 = 0.21''$ , and  $\omega_o = 12\text{GHz}$ .**



(a)

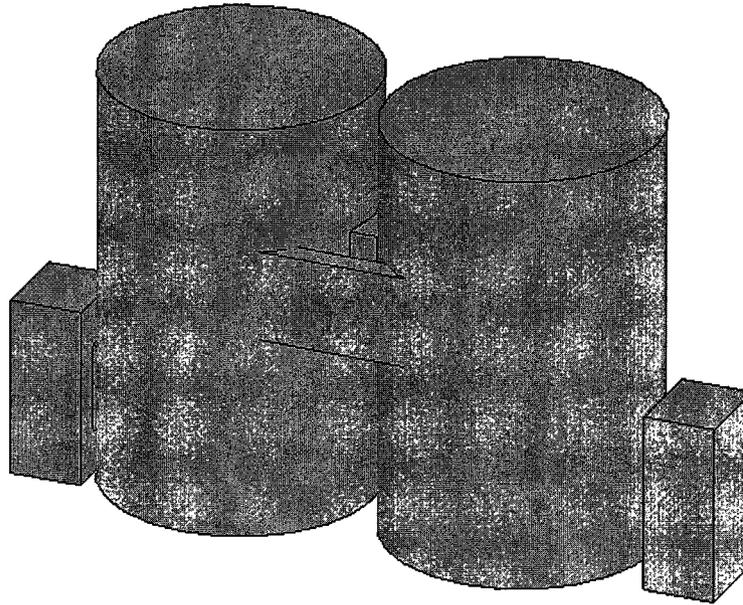


(b)

**Figure 4.6: Comparison of S-parameter of conventional neural network and proposed model of a 4-pole H-plane filter. (a) Filter geometry 1:  $L_{b1} = 0.52''$ ,  $L_{b2} = 0.58''$ ,  $W_1 = 0.38''$ ,  $W_2 = 0.25''$ ,  $W_3 = 0.22''$ ,  $\omega_o = 11.8$  GHz, (b) Filter geometry 2:  $L_{b1} = 0.54''$ ,  $L_{b2} = 0.60''$ ,  $W_1 = 0.37''$ ,  $W_2 = 0.23''$ ,  $W_3 = 0.21''$ ,  $\omega_o = 12$ GHz. Output of the conventional model is not accurate, because the amount of data used for training is not enough for the conventional method. However, the same data is enough for the proposed method.**

#### **4.4.2 Development of a Side-Coupled Circular Waveguide Dual-Mode Filter Model with the Proposed High Dimensional Modeling Technique**

We apply the proposed high dimensional modeling method to develop a neural network model of a complex filter known as side-coupled circular waveguide dual-mode filter [136], [137]. Figure 4.7 shows a physical diagram of the filter. Unlike the conventional longitudinal end-coupled configuration, the filter input-output coupling and coupling between the circular cavities are realized at the sides of the circular cavities. This type of filter offers significant performance improvement and finds its application in the satellite multiplexers with extremely stringent mass, size, and thermal requirements. However, the design and simulation becomes more difficult due to the structural complexity [137].



**Figure 4.7: Diagram of a side-coupled circular waveguide dual-mode filter.**

The filter contains 15 design variables including 12 geometrical parameters, bandwidth, center frequency, and frequency. By using conventional neural network approach to represent this 15-dimensional problem, i.e., 15 input neurons, data generation and neural network training would be prohibitive. Here we apply the proposed neural network decomposition method to simplify the high-dimensional modeling problem into a set of low-dimensional modeling problems. As will be shown in the following, for such complex filters, responses based on submodels alone are not satisfactory. Instead of direct mapping of S-parameters of EM simulator and the neural network model, circuit model based on coupling matrix is adopted as the modeling objective. In doing so, the difficulty in the alignment or mapping of full EM and neural network model responses is significantly reduced, enabling the accurate modeling of complex filters with minimum number of full EM simulations. Once the accurate coupling matrix is achieved, a circuit simulator can be used to obtain the accurate S-parameter for any frequency range.

Thus, the input and output vectors of the model are

$$\mathbf{x} = [L_{r1} \ L_{r2} \ L_{11b1} \ L_{22b1} \ L_{12b1} \ L_{23} \ L_{14} \ L_{11b2} \ L_{22b2} \ L_{12b2} \ L_{b1} \ L_{b2} \ B \ \omega_o \ \omega]^T \quad (4.23)$$

and

$$\mathbf{y} = [R_1 \ R_2 \ M_{11} \ M_{22} \ M_{33} \ M_{44} \ M_{12} \ M_{23} \ M_{34} \ M_{14}]^T, \quad (4.24)$$

respectively. In (4.23),  $L_{r1}$  and  $L_{r2}$  represent lengths of input iris and output iris, respectively,  $L_{11b1}$ ,  $L_{22b1}$ , and  $L_{12b1}$  represent lengths of three screws of cavity 1,  $L_{11b2}$ ,

$L_{22b2}$ , and  $L_{12b2}$  represent three screws of cavity 2,  $L_{23}$  represents length of the sequential coupling iris,  $L_{14}$  represents length of cross coupling iris,  $L_{b1}$  and  $L_{b2}$  represent lengths of cavity 1 and cavity 2, respectively,  $B$  represents bandwidth,  $\omega_o$  represents center frequency, and  $\omega$  represents frequency. In (4.24),  $R_1$  and  $R_2$  represent input and output coupling bandwidth,  $M_{11}$  to  $M_{44}$  are self coupling bandwidths, and  $M_{12}$ ,  $M_{23}$ ,  $M_{34}$ , and  $M_{14}$  represent sequential and cross-coupling bandwidths.

In the first step, we decompose the filter into three types of substructures ( $N_{sub} = 3$ ), named as input-output iris, internal coupling iris, and coupling and tuning screw [17], for which three neural network submodels will be developed. The inputs of the input-output iris model are iris length  $L_r$  and  $\omega_o$ . The outputs are coupling bandwidth  $R$  and phase  $P_v$  representing loading effect of internal coupling iris. The inputs of the internal coupling iris model are lengths of the sequential coupling iris  $L_{23}$ , cross coupling iris  $L_{14}$ , and  $\omega_o$ , and outputs are sequential coupling  $M_{23}$ , cross coupling  $M_{14}$ , and phases  $P_h$  and  $P_v$ . Phases  $P_h$  and  $P_v$  are the loading effect of the internal coupling irises on the two orthogonal modes, respectively. The inputs of coupling and tuning screw model are screw lengths  $L_{11}$ ,  $L_{22}$ , and  $L_{12}$  and  $\omega_o$ . The outputs are coupling bandwidth  $M_{ij}$  for  $i \neq j$ ,  $P_v$ , and  $P_h$ . Note that the number of input variables of each substructure is much less than that of the overall filter.

Next, we combine neural network decomposition with the side-coupled filter decomposition scheme. Following the Step 2 of the modeling algorithm, we generate training data to develop neural network submodels for each of the substructures. Since

each of the substructures has few design variables, for example input-output iris has only two variables; we can generate many data in a short time. This allows us to develop very accurate submodel. Each substructure is simulated using EM simulator based on mode-matching method as described in [136]. The filter input-output couplings are obtained using the group delay method and the inter-resonator couplings are calculated using eigenvalue calculation [135]. We generate 423 samples of data for input-output iris model and the model testing error is 0.5%. We also generate 4930 data samples to develop the internal coupling iris model and less than 0.2% average testing error is achieved for this model. For coupling and tuning screw model, we generate 36015 samples of data and average model testing error is 0.51%. Training times of the three submodels are less than 1 minute, approximately 3 minutes, and 2 hours, respectively. The submodels are trained using automatic model generation module of *NeuroModelerPlus* [127].

In Step 5, full EM data are generated by simulating the entire side-coupled filter with 64 different combinations of geometrical values. The bandwidth and center frequency are varied from 27 MHz to 54 MHz and 11GHz to 11.7 GHz, respectively. As mentioned earlier, instead of using the S-parameters generated using EM simulator directly, coupling parameters are used as the modeling objectives. We extract 64 coupling values using the S-parameter extraction technique as presented in [138].

In the next Step, we combine the neural network submodels to represent the filter structure. Both input-output iris model and coupling and tuning screw model are used twice and internal coupling iris model is used once to represent the filter, i.e.,  $N_o = 5$ . The

neural network submodels are used to produce cross-couplings and empirical models are used to compute self-couplings.

As described in Step 7 of the modeling algorithm in Section 4.3, we produce approximate coupling values using the same samples of geometrical parameters of Step 5. Following the procedure as described in Step 8, we assemble training data of mapping model. Since individual coupling parameters are a function of specific geometrical dimensions rather than a function of all the dimensions, we produce a separate mapping model for each of them. Thus, 10 mapping models for the 10 coupling parameters as described in (4.24) are developed. The mapping models are defined as

$$M_j = f_M(\tilde{\mathbf{x}}_j, M_j^a, \mathbf{w}_M) \quad (4.25)$$

where  $M_j$  represents the  $j^{\text{th}}$  coupling parameter,  $M_j^a$  represents the  $j^{\text{th}}$  approximate coupling parameter obtained from neural network submodel,  $\tilde{\mathbf{x}}_j$  is a subset of  $\mathbf{x}$ ,  $j = 1, 2, 3, \dots, 10$ . Four different sets of EM data of the overall filter as listed in Table 4.2 are used to develop four sets of mapping models. In Set 1, data from 44 filter geometries are used for training and data from 20 other filter geometries are used for testing. The number of filter geometries is reduced for training in the subsequent three sets and listed in Table 4.2. Training time of the 10 neural network mapping models are less than 5 minutes.

We construct an accurate model of the side coupled filter by connecting the 10

neural network mapping models with the submodels and empirical models used to produce approximate coupling matrix. The overall model is then tested using the test data as listed in Table 4.2. For comparison, four neural network models are also trained using the same four data sets in the conventional method, which relates geometrical variables to the coupling matrix directly.

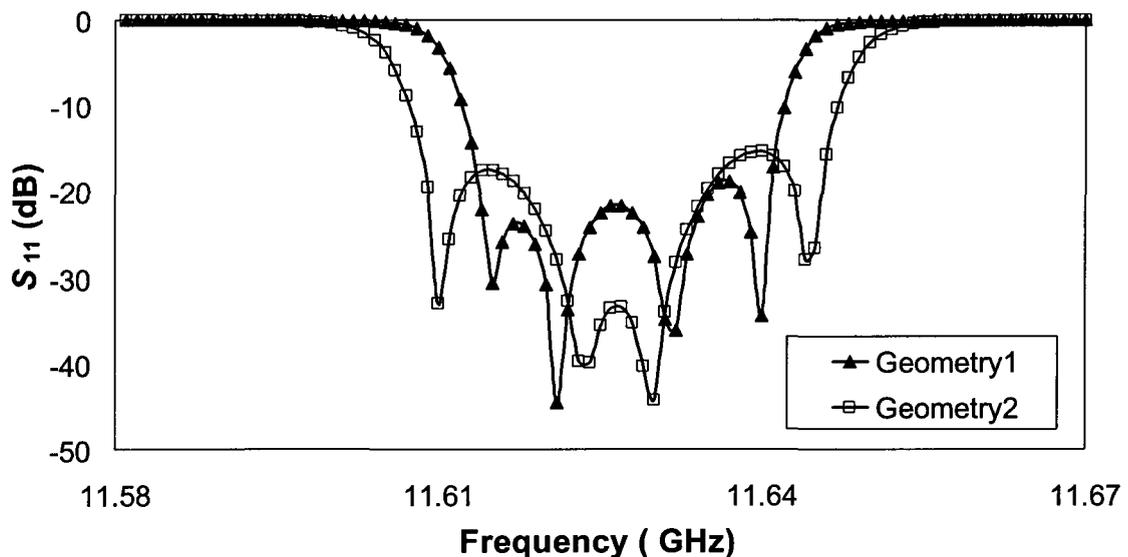
**Table 4.2: Comparison of test error of side-coupled circular waveguide dual-mode filter models developed with conventional and proposed high dimensional modeling approach**

Data Set no.	No. of filter geometries used		Modeling method	Model testing error (%)	
	Train	Test		Average error	Worst case error
1	44	20	Conventional	18.3	227
			Proposed	1.60	8.5
2	32	32	Conventional	23.5	184
			Proposed	2.40	15.8
3	16	32	Conventional	18.7	49.5
			Proposed	5.3	28.9
4	8	32	Conventional	22.1	45.2
			Proposed	5.7	33.5

Table 4.2 compares the model error between the two methods, which shows that the proposed method is much more accurate than the conventional method for all data sets. By using the proposed method, we can produce good accuracy with limited amount of data, because mapping function becomes simple after obtaining approximate couplings

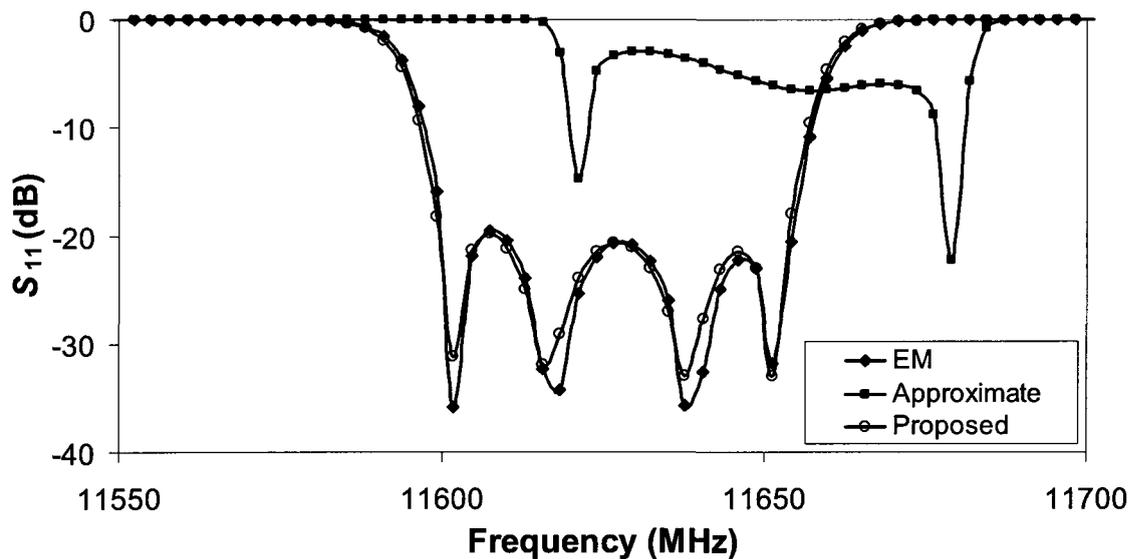
from submodels (trained with inexpensive data) and empirical circuit model. On the other hand, the conventional method is inaccurate, because the amount of training data is insufficient to produce a 15-dimensional side-coupled filter model. If we were to improve the accuracy of the conventional method, we would have to use lot more data which would be expensive and difficult to generate.

In Figure 4.8, we plot responses of two different filter configurations obtained from the proposed model. It shows that the model can be used to obtain responses for various filter geometries.



**Figure 4.8: Reflection coefficients of two different side-coupled circular waveguide dual-mode filters obtained using the proposed model. Geometry 1:  $B = 27$  MHz,  $\omega_o = 11.627$  GHz; Geometry 2:  $B = 35$  MHz,  $\omega_o = 11.627$  GHz.**

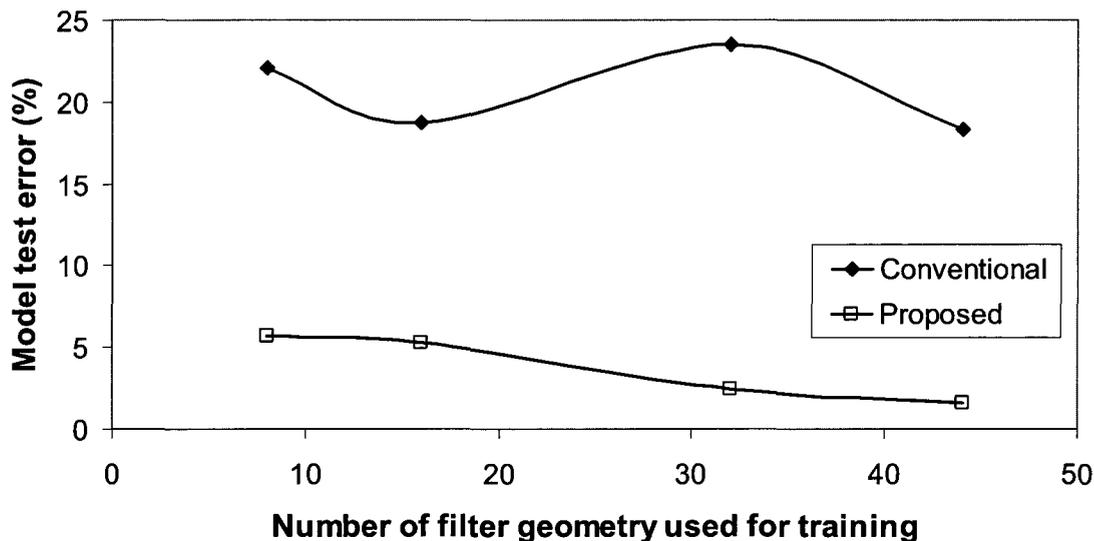
Figure 4.9 shows the effectiveness of mapping model. The approximate filter response, which is generated from the approximate coupling matrix without using proposed mapping models, is not satisfactory. The mapping models then provide accurate couplings which leads to the response very close to the accurate EM response.



**Figure 4.9: Reflection coefficient of a side-coupled circular waveguide dual-mode filter with  $B = 54$  MHz,  $\omega_o = 11.627$  GHz showing the effectiveness of the neural network mapping in the coupling parameter space.**

Figure 4.10 shows a plot of average model test error vs. the number of filter geometry used for model training. The plot shows that the model test error of the proposed method is low and decreases consistently with the number of filter used for training. On the other hand, the error of the conventional method stays high at

approximately 20%. To reduce the error of the conventional method, we need to use massive training data.



**Figure 4.10: Comparison of average model test error vs. the number of filter geometry used for model training in conventional and proposed method of the side-coupled circular waveguide dual-mode filter.**

In Table 4.3, we list model evaluation time of two commonly used EM modeling methods and compare with the evaluation time of the proposed high dimensional neural network modeling method. Full EM simulation of the entire filter needs approximately 6 minutes using mode-matching based EM simulator [136] and 45 minutes using finite-element based EM simulator such as HFSS [139]. The comparison clearly shows that the proposed method is significantly faster than the EM methods enabling fast design and optimization.

**Table 4.3: Comparison of CPU time of EM and neural network model of a side-coupled circular waveguide dual-mode filter**

Modeling Method	Time/model evaluation
Finite element method	45 min
Mode matching method	6 min
Proposed method	0.006 s

In order to develop an accurate model, e.g., less than 2% of model testing error, using the conventional method, we need to sample sufficiently the specified range for all input variables. For example, if we sample 3 values for each of the 10 geometrical variables, 7 values for  $\omega_o$  and 4 values for  $B$ , we need total  $N^c = 3^{10} \times 7 \times 4 = 1.65 \times 10^6$  samples of the overall filter. Using the fastest EM simulation method, i.e., mode matching method, the data generation time per sample of the overall filter is,  $t_o = 6$  min. The total data generation time for this 15-dimensional neural network model of the conventional method using (4.12) is estimated to be,

$$T^c = 6 \text{ min} \times 1.65 \times 10^6 \cong 19 \text{ years}, \quad (4.26)$$

which is too expensive. The model training time ( $\Gamma^c$ ) using the massive training data would also be too expensive.

We now calculate data generation time of the proposed method. Data generation

time per sample of input-output iris, internal coupling iris, and coupling and tuning screw substructures are  $t_1 = 0.6$  s,  $t_2 = 5.6$  s, and  $t_3 = 6.9$  s, respectively. To cover the same range of the input geometrical space that is used in this example, we need  $N_1^s = 48$  samples of input-output iris,  $N_2^s = 192$  samples of internal coupling iris, and  $N_3^s = 288$  samples of coupling and tuning screw substructures. In order to achieve less than 2% of model testing error using the proposed method, we also need approximately 40 samples of the overall filter for the training of mapping model, i.e.,  $N^M = 40$ . The total data generation time of the proposed method using (4.13) is calculated to be,

$$\begin{aligned}
 T^P &= t_o \times N^M + \sum_{i=1}^3 (t_i \times N_i^s) \\
 T^P &= 6 \times 40 \text{ min} + (0.6 \times 48 + 5.0 \times 192 + 6.9 \times 288) \text{ s} \\
 T^P &\cong 4.8 \text{ hours.}
 \end{aligned} \tag{4.27}$$

The model training time ( $\Gamma^P$ ) of three submodels and 10 mapping models all together is less than 10 minutes and is therefore, insignificant. Thus, an accurate neural network model of the side-coupled filter which is very expensive to develop using the conventional neural network method becomes feasible using the proposed method.

## 4.5 Summary

In this Chapter, another major contribution of the thesis has been presented. We have proposed an effective neural network modeling technique for filters that hold many design variables. It is impractical to develop a neural network model for such structures in conventional neural network approach. We have proposed a new formulation to integrate neural network decomposition with filter structure decomposition and then incorporate circuit knowledge to obtain a complete filter model. The filter structure has been decomposed into substructures to reduce the number of variables per submodel. Neural network submodels have then been developed for each of the substructures. Empirical/equivalent circuit models have been combined with neural network submodels to produce an approximate solution of the filter. Another neural network model has then been trained to map approximate solution to the accurate solution of the filter. The result has shown that the proposed method can be used to produce high dimensional models with few full EM training data, which are usually expensive to generate, compared to conventional neural network technique. The method has been very useful for developing neural network models of microwave filters that have many design variables. The developed neural network models have become very useful for fast design optimization of those filters.

## **Chapter 5: Conclusion and Future Work**

### **5.1 Conclusion**

This thesis has presented advanced techniques for neural network based modeling and design of RF/microwave circuits. These techniques have been developed aiming to enhance the present state-of the art microwave computer aided design to a new high level. First contribution towards the goal has been made by proposing neural network inverse modeling techniques to solve inverse EM problems. The inverse approach is an unconventional modeling approach which is useful for simplifying repetitive design procedure. The formulation of the neural network inverse modeling technique has been presented. Non-uniqueness of model input-output relationship has been addressed with methods to identify multivalued solutions and divide training data for training inverse models. Data of inverse model have been proposed to divide based on derivatives of forward model and then have been used separately to train more accurate inverse submodels. Once the neural submodels are trained, they need to be combined to form the complete model. We propose a method to correctly combine the inverse sub-models based on error verification of inverse forward pair as well as the distance of outputs outside the training range. Additional techniques to enhance the accuracy of the model combine are presented. A comprehensive modeling algorithm utilizing various techniques

has been presented. This algorithm has been found useful for efficient model development. The inverse models developed using the proposed techniques are more accurate than that using the direct method.

Furthermore, an inverse design approach using the proposed inverse models has been developed. This approach avoids the need for repetitive model evaluation. The inverse models replace the repetitive loop from the design cycle. In order to validate the techniques, the proposed methodology has been applied to waveguide filter modeling. Results and comparisons show that the proposed method produces much accurate inverse model than conventionally developed neural network. We have also presented design of 4-pole and 6-pole waveguide filters using the developed inverse approach to verify device level simulation and validate the proposed approach. The 6-pole filter has been fabricated, and dimensions of the tuned filter have been measured for verifications. Very good correlation has been found between neural network predicted dimensions and that of the perfectly tuned filters.

Another major contribution has been made by proposing a method to solve high dimensional modeling problem. We have proposed an effective neural network modeling technique for filters that hold many design variables. It was impractical to develop a high dimensional neural network model in conventional neural network approach because of massive cost of data generation and model development. We have proposed a new formulation to integrate neural network decomposition with filter structure decomposition and then incorporate circuit knowledge to obtain a complete filter model. The filter structure has been decomposed into substructures, which reduced the number

of variables per submodel. Neural network submodels have been developed for each of the substructures. Empirical/equivalent circuit models have been combined with neural network submodels to produce an approximate solution of the filter. In order to improve the accuracy, we have proposed to use another neural network model to map approximate solution to the accurate solution of the filter. The neural network sub-models, empirical/circuit model and neural mapping model have been combined to form the overall accurate high dimensional model. The proposed modeling approach has been used to develop complex filter models. The result has shown that the proposed method can be used to produce high dimensional models with few full EM training data, which are usually expensive to generate, compared to conventional neural network technique. The proposed method has allowed us to develop high dimensional neural network models conveniently unlike the conventional method which became too expensive.

The proposed techniques have advanced the computer aided design of microwave structures facilitating engineers to explore design and development conveniently. The developed models have become accurate representation of RF/microwave structures. The techniques have been useful for design and optimization of high dimensional models relaxing the computational cost of EM based models.

## **5.2 Future Work**

Neural network has been established as a powerful alternative for RF/microwave modeling and it has been increasingly becoming more popular for solving complex EM problems. The CAD techniques using neural networks reduce the cost of design and

optimization by achieving EM design accuracy without computation expense of EM based model. As the complexity of problems in the RF/microwave area will continue to increase, we will need further improvement and enhancements in neural network modeling techniques.

This thesis has presented advanced neural network based modeling techniques. These techniques have the potential to be extended to more complex and broader areas of applications. An important direction for future research would be to extend the inverse modeling approach to other filter modeling applications. As an example, microstrip filters will be considered to design in the inverse approach. Circuit parameters of a microstrip filters will be extracted. Inverse neural network models will then be developed where the circuit parameters will be considered as inputs of the neural network and lengths and gaps and other dimensions as outputs of the neural network inverse model. The inverse model will then provide the design parameters which are the geometrical parameters of the filter for a given electrical parameters. This will avoid the repetitive model evaluations and will generate a filter configuration quickly.

Another important direction for future research of the proposed techniques will be to apply the high dimensional approach to solve high dimensional inverse modeling problems. The high dimensional inverse model would significantly improve the design and development time. Finding the optimum values of design parameters might be challenging for a high dimensional model. Developing an inverse model would reduce the complexity and provide the solution quickly. The techniques can be applied to a complete structure or a part of the structure to enhance the design cycle time. The

decomposition of structure might not be possible for all structure. In that case equivalent circuit parameter can be extracted from the structure. Combining an empirical formula will produce a close result. The neural network mapping model then can developed for accuracy improvement.

Another important direction for future research would be to extend the proposed filter modeling techniques to develop model for multiplexers in the microwave and millimeter frequency range. The purpose of such model would be to use the multiplexer model for system level simulation where several waveguide structures are connected. The multiplexer model would provide significant speed for simulation, optimization of a complete system. The input output of the model has to be formulated in accordance with the requirements of the system. The waveguide filter with higher dimension and higher frequency range is also would be an important direction to apply the high dimensional modeling with the inverse approach.

In conclusion, the complexity of microwave design will continue to increase and better modeling solutions will be required for fast and efficient computer aided design. The neural network modeling techniques proposed in this thesis have contributed to the effort to make the microwave computer aided design more efficient. More research in the outlined directions would make the effort stronger and thus would make the microwave CAD more attractive and powerful for solving challenging design and optimization task in the future.

## Bibliography

- [1] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design—from theory to practice," *IEEE Trans. Microwave Theory Tech.*, vol. 51, no. 4, pp. 1339-1350, April 2003.
- [2] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*, Boston, MA: Artech House, 2000.
- [3] A. Patnaik and R. K. Mishra, "ANN techniques in microwave engineering," *IEEE Microwave Mag.*, vol. 1, pp. 55-60, Mar. 2000.
- [4] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microwave Theory Tech.*, vol. 45, no. 12, pp. 2515-2523, December 1997.
- [5] F. Nunez and A. K. Skrivervik, "Filter approximation by RBF-NN and segmentation method," *IEEE MTT-S Int. Microwave Symp. Digest*, vol. 3, June 2004, 1561- 1564.
- [6] J. E. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 1, pp. 420-435, January 2004.
- [7] J. M. Cid and J. Zapata, "CAD of rectangular waveguide H-plane circuits by

- segmentation, finite elements and artificial neural networks,” *IEE Electronic Letters*, vol. 37, 98-99, January 2001.
- [8] A. Mediavilla, A. Tazon, J. A. Pereda, M. Lazaro, I. Santamaria and C. Pantaleon, “Neuronal architecture for waveguide inductive iris bandpass filter optimization,” *in Proceedings of the IEEE-INNS-ENNS Int. Joint Conf. on Neural Networks*, vol. 4, pp. 395-399, July 2000.
- [9] P. Burrascano, M. Dionigi, C. Fancelli, and M. Mongiardo, “A neural network model for CAD and optimization of microwave filters,” *1998 IEEE MTT-S Int. Microwave Symp. Dig.*, vol. 1, pp. 13–16, June 1998.
- [10] A. S. Ciminski, “Artificial neural networks modeling for computer aided design of microwave filter,” *Int. Conf. on Microwaves, Radar and Wireless Communications*, vol. 1, May 2002, 95-99.
- [11] K. C. Gupta, “Emerging trends in Millimeter-Wave CAD,” *IEEE Trans. Microwave Theory Tech.*, vol. 46, no. 6, pp. 747-755, June 1998.
- [12] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, “Applications of artificial neural network techniques in microwave filter modeling, optimization and design,” *Progress In Electromagnetic Research Symposium*, pp. 1972-1976, Beijing, China, Mar. 2007.
- [13] F. Wang and Q. J. Zhang, “Knowledge-based neural models for microwave design,” *IEEE Trans. Microwave Theory Tech.*, vol. 45, no. 12, pp. 2333–2343, Dec. 1997.

- [14] P. M. Watson, K. C. Gupta, and R. L. Mahajan, "Applications of knowledge-based artificial neural network modeling to microwave components," *Int. J. RF and Microwave Comput.-Aided Eng.*, vol. 9, no. 3, pp. 254–260, May 1999.
- [15] Y. Li, K. Wang, and T. Li, "Modular neural network structure with fast training/recognition algorithm for pattern recognition," *IEEE Int. Conf. on Granular Computing*, Aug. 26–28, 2008, Hangzhou, China, pp. 401–406.
- [16] H. Kabir, Yi. Cao, L. Zhang, and Q. J. Zhang, "Neural based EM Modeling," *URSI Int. Symp. on Signals, Systems, and Electronics*, pp. 169-172, 2007.
- [17] 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.*, San Francisco, USA, June 2006.
- [18] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Trans. Microwave Theory Tech.*, vol. 56, no. 4, pp. 867–879, Apr. 2008.
- [19] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "Neural network inverse modeling methods and accurate solution of electromagnetic devices," *Applied Computational Electromagnetic Society Dig.*, pp. 132–137. Monterey, CA, Mar. 2009.
- [20] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "State-of-The-Art microwave filter modeling and design using neural network technique," *Progress In Electromagnetic Research Symposium*, Hangzhou China, Mar. 2008.

- [21] H. Kabir, Y. Cao, and Q. J. Zhang, "Advances of neural network modeling methods for RF/microwave applications," Under review, *Journal of Applied Computational Electromagnetic Society*.
- [22] H. Kabir, Q. J. Zhang, and Ming Yu, "Neural network techniques in electromagnetic applications," Under review, *IEEE Asia-Pacific Microwave Conf.*, Singapore, Dec. 2009.
- [23] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "High Dimensional neural network techniques and applications to microwave filter modeling," Under review, *IEEE Trans. Microwave Theory Tech.*
- [24] R. -S, Guh, "Optimizing feedforward neural networks for control chart pattern recognition through genetic algorithm," *Int. Journal Pattern Recognition and Artificial Intelligence*, vol.18, no. 2, pp. 75-99, March 2004.
- [25] J. Wang and G. Wu, "Multilayer recurrent neural network for real-time synthesis of linear-quadratic optimal control systems," *Proc. IEEE Int. Conf. Neural Networks*, vol. 4, pp. 2506-2511, 1994.
- [26] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr, "Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection," *IEEE Trans. Biomedical Engineering*, vol. 55, no. 2, pp. 512-518, Feb. 2008.
- [27] J. G. Hincapie and R. F. Kirsch, "Feasibility of EMG-based neural network controller for an upper extremity neuroprosthesis," *IEEE Trans. Neural Systems and Rehabilitation Eng.*, vol. 17, no. 1, pp. 80-90, Feb. 2009.

- [28] W. J. Blackwell and F. W. Chen, "Neural network applications in high-resolution atmospheric remote sensing," *J. Lincoln Laboratory*, vol. 15, no. 2, pp. 299-322, 2005.
- [29] S. Chitroub, "Neural network model for standard PCA and its variants applied to remote sensing," *Int. J. of Remote Sensing*, v. 26, no. 10, pp. 2197-2218, May 2005.
- [30] C. Cho and K. C. Gupta, "EM-ANN modeling of overlapping open-ends in multilayer microstrip lines for design of bandpass filters," *IEEE Int. Symp. Antennas and Propagation*, vol. 4, pp. 2592-2595, July 1999.
- [31] X. Ding, V. K. Devabhaktuni, B. Chattaraj, M. C. E. Yagoub, M. Deo, 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. Microwave Theory Tech.*, vol. 52, no. 1, pp. 436-449, January 2004.
- [32] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349-1358, June 1995.
- [33] P. M. Watson, K. C. Gupta, and R. L. Mahajan, "Development of knowledge based artificial neural network models for microwave components," *1998 IEEE MTT-S Int. Microwave Symp. Dig.*, vol. 1, pp. 9-12, June 1998.

- [34] X. Zhang, Y. Cao, and Q. J. Zhang, "A combined transfer function and neural network method for modeling via in multilayer circuits," *2008 Midwest Symp. On Circuits and Systems*, pp. 73-76, August 2008.
- [35] P. M. Watson, G. L. Creech, and K. C. Gupta, "Knowledge based EM-ANN models for the design of wide bandwidth CPW patch/slot antennas," *1999 IEEE MTT-S Int. Microwave Symp. Dig.*, vol. 4, pp. 2588-2591, July 1999.
- [36] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sanchez, and Q. J. Zhang, "Neuromodeling of microwave circuits exploiting space-mapping technology," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 12, pp. 2417-2427, Dec. 1999.
- [37] Q. J. Zhang, L. Ton, and Y. Cao, "Microwave modeling using artificial neural networks and applications to embedded passive modeling," *Int. Conf. on Microwave and Millimeter Wave Technology*, vol. 34, no. 3, pp. 1954-1963, April 2008.
- [38] F. Wang, V. K. Devabhaktuni, and Q. J. Zhang, "A hierarchical neural network approach to the development of a library of neural models for microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. 46, no. 12, pp. 2391-2403, Dec. 1998.
- [39] V. K. Devabhaktuni, M. C. E. Yagoub, and Q. J. Zhang, "A robust algorithm for automatic development of neural-network models for microwave applications," *IEEE Trans. Microwave Theory Tech.*, vol. 49, no. 12, pp. 2282-2291, Dec. 2001.

- [40] C. Ydiz, M. Turkmen, "Very accurate and simple CAD models based on neural networks for coplanar waveguide synthesis," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 15, no. 2, pp. 218-224, March 2005.
- [41] B. Davis, C. White, M. A. Reece, M. E. Jr. Bayne, W. L. Thompson, II, N. L. Richardson, and L. Jr. Walker, "Dynamically configurable pHEMT model using neural networks for CAD," *IEEE MTT-S Int. Microwave Symp. Dig.*, Philadelphia, PA, vol. 1, June 2003, pp. 177-180.
- [42] J. Wood, P. H. Aaen, D. Bridges, D. Lamey, M. Guyonnet, D. S. Chan, N. Monsauret, "A nonlinear electro-thermal scalable model for high-power RF LDMOS transistors," *IEEE Trans. Microwave Theory Tech.*, vol. 57, no. 2, pp. 282-292, Feb. 2009.
- [43] F. Gianni, P. Colantonio, G. Orenco, and A. Serino, "Neural network modeling of microwave FETs based on third-order distortion characterization," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 16, no. 2, pp. 192-200, March 2006.
- [44] X. Li, J. Gao, and Q. J. Zhang, "Microwave noise modeling for PHEMT using artificial neural network techniques," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 19, no. 2, pp. 187-196, Mar. 2009.
- [45] J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, "Exact adjoint sensitivity analysis for neural-based microwave modeling and design," *IEEE Trans. Microwave Theory and Tech.*, vol. 51, pp. 226-237, Jan. 2003.
- [46] V. Markovic, Z. Marinkovic, and N. Males-Ilic, "Application of neural networks in microwave FET transistor noise modeling," *Proceedings of Neural Network*

*Applications in Electrical Engineering*, Belgrade, Yugoslavia, Sept. 2000, pp. 146-151.

- [47] P. Burrascano, S. Fiori, and M. Mongiardo, "A review of artificial neural networks applications in microwave computer-aided design," *Int. J. RF Microwave Comput.-Aided Eng.*, vol. 9, pp. 158–174, May 1999.
- [48] V. 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 MTT-S Int. Microwave Symposium Dig.*, vol. 2, pp. 1097-1100, 2002.
- [49] D. Nihad, A. Jehad, O. Amjad, "CAD modeling of coplanar waveguide interdigital capacitor," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 15, no. 6, pp. 551-558, Nov. 2005.
- [50] D. McPhee, M. C. E. Yagoub, "Novel approach for efficient electromagnetic coupling computation in RF/Microwave integrated circuits," *WSEAS Transaction on Communications*, vol. 4, no. 6, pp. 247-255. June 2005.
- [51] Y. J. Lee, Y. -H. Park, F. Niu, and D. Filipovic, "Design and optimization of RF ICs with embedded linear macromodels of multiport MEMS devices," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 17, no. 2, pp. 196-209, March 2007.
- [52] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microwave Theory and Tech.*, vol. 53, No. 11, pp. 3422-3428, Nov. 2005.

- [53] F. Gianni, P. Colantonio, G. Orenco, A. Serino, G. Stegmayer, M. Pirola, and G. Ghione, "Neural networks and volterra series for time-domain power amplifier behavioral models," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 17, no. 2, pp. 160-168, March 2007.
- [54] A. Luchetta, S. Manetti, L. Pellegrini, G. Pelosi, S. Seleri, "Design of waveguide microwave filters by means of artificial neural networks," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 16, no. 6, pp. 554-560, Nov. 2006.
- [55] V. Rizzoli, A. Costanzo, D. Masotti, A. Lipparini, and F. Mastri, "Computer-aided optimization of nonlinear microwave circuits with the aid of electromagnetic simulation," *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 1, pp. 362-377, January 2004.
- [56] S. Bila, D. Baillargeat, M. Aubourg, S. Verdeyme and P. Guillon, "A full electromagnetic CAD tool for microwave devices using a finite element method and neural networks," *Int. J. of Numerical Modeling*, pp. 167-180, 2000.
- [57] Z. Z. Stankovic, B. Milovanovic, N. Doncov, "Neural model of microwave cylindrical cavity loaded with arbitrary-raised dielectric slab," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 19, no. 3, pp. 317-327, May. 2009.
- [58] J. P. Garcia, F. Q. Pereira, D. C. Rebenaque, J. L. G. Tornero, and A. A. Melcon, "A neural-network method for the analysis of multilayered shielded microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 1, pp. 309-320, Jan. 2006.

- [59] P.H.F. Silva, M. G. Passos, and A. G. d'Assunção, "Fast and accurate analysis for the directivity of circular-shape antennas using optimal neural networks," *Microwave and Optical Technology Letters*, vol. 49, no. 11, pp. 2721-2726, Nov. 2007.
- [60] Q. J. Zhang and V. K. Devabhaktuni, "Neural network structures for EM/microwave modeling," *IEEE AP-S Int. Symp. Dig.*, Orlando, FL, July 1999, pp. 2576-2579.
- [61] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York, NY: IEEE Press, 1994.
- [62] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Math Control Signals Syst.*, vol. 2, pp. 303-314, 1989.
- [63] B. Gao and Y. Xu, "Univariant approximation by superposition of a sigmoid function," *J. Mathematical Analysis and Applications*, vol. 178, pp. 221-226, 1993.
- [64] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.
- [65] T. Y. Kwok and D. Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Trans. Neural Networks*, vol. 8, pp. 630-645, May 1997.
- [66] J. de Villiers and E. Barnard, "Backpropagation neural nets with one and two hidden layers," *IEEE Trans. Neural Networks*, vol. 4, pp. 136-141, Jan. 1992.

- [67] F. Wang, V. K. Devabhaktuni, C. Xi, and Q. J. Zhang, "Neural network structures and training algorithms for microwave applications," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 9, pp. 216-240, 1999.
- [68] Y. Fang, M. Yagoub, F. Wang, and Q. J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2335-2344, Dec. 2000.
- [69] J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications and Programming Techniques*, Reading, Mass: Addison-Wesley, 1992.
- [70] J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, "Neural-based dynamic modeling of nonlinear microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 50, no. 12, pp. 2769-2780, Dec. 2002.
- [71] F. Girosi, *Regularization theory, radial basis functions and networks, from statistics to neural networks: theory and pattern recognition applications*, 1992.
- [72] M. J. D. Powell, *Radial basis functions for multivariate interpolation: a review, Algorithms for approximation*, Oxford Univ. Press, 1987.
- [73] Q. H. Zhang and A. Benvensite, "Wavelet networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 889-898, Nov. 1992.
- [74] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, no. 12, pp. 2495-2503, December 1996.
- [75] G. Thimm and E. Fiesler, "High-order and multilayer perceptron initialization," *IEEE Trans. Neural Networks*, vol. 8, pp. 349-359, Mar. 1997.

- [76] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, vol. 1, pp. 318-362, MIT Press, Cambridge, MA, 1986.
- [77] R. L. Watrous, "Learning algorithms for connectionist networks: applied gradient methods for nonlinear optimization," *Proc. IEEE 1<sup>st</sup> Int. Conf. Neural Networks*, vol. 2, pp. 619-627, San Diego, CA, 1987.
- [78] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [79] A. J. F. Van Rooji, L. C. Jain, R. P. Johnson, *Neural networks training using genetic algorithms*, World Scientific, 1996.
- [80] H. Ninomiya, S. Wan, H. Kabir, X. Zhang, and Q. J. Zhang, "Robust training of microwave neural network models using combined global/local optimization techniques," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 995-998, Atlanta, Georgia, Jun. 2008.
- [81] P. E. An, M. brown, C. J. Harris, and S. Chen, "Comparative aspects of neural network algorithms for on-line modeling of dynamic processes," *Proc. Inst. Mech. Eng.*, vol. 207, pp. 223-241, 1993.
- [82] S. McLoone and G. W. Irwin, *Fast gradient-based off-line training of multilayer perceptrons*, *Neural Network Engineering in Dynamic Control Systems*, K. J. Hunt, G. W. Irwin and K. Warwick, Eds. New York: Springer, 1995, ch. 9, pp. 279-200.

- [83] L. Zhang, Y. Cao, S. Wan, H. Kabir, and Q. J. Zhang "Parallel automatic model generation techniques for microwave modeling," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 103-106, Honolulu, Hawaii, Jun. 2007.
- [84] A. Veluswami, M. S. Nakhla, and Q. J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, no. 5, pp. 712-723, May 1997.
- [85] A. Gati, M. F. Wong, G. Alquie, and V. F. Hanna, "Neural network modeling and parameterization applied to coplanar waveguide components," *Int. J. RF and Microwave Comput.-Aided Eng.*, vol. 10, no. 5, pp. 296-307, September 2000.
- [86] Z. Yang, T. Yang, and Y. Liu, "Design of microstrip Lange coupler based on EM-ANN model," *Int. J. Infrared and Millimeter Waves*, vol. 27, no. 10, pp. 1381-1389, October 2006.
- [87] P. Sharma, F. A. Mohammadi, and M. C. E. Yagoub, "Neural design and optimization of RF/Microwave EM-based multichip modules," *2004 RF and Microwave Conf.*, pp. 67-71, October 2004.
- [88] S. Bila, Y. Harkouss, M. Ibrahim, J. Rousset, E. N'Goya, D. Baillargeat, S. Verdeyme, M. Aubourg, and P. Guillon, "An accurate wavelet neural-network-based model for electromagnetic optimization of microwave circuits," *Int. J. RF and Microwave Comput.-Aided Eng.*, vol. 9, pp. 297-306, Dec. 1999.
- [89] P. Sen, W. H. Woods, S. Sarkar, R. J. Pratap, B. M. Dufrene, R. Mukhopadhyay, C. Lee, E. F. Mina, and J. Laskar, "Neural-network-based parasitic modeling and

- extraction verification for RF/millimeter-wave integrated circuit design,” *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 6, pp. 2604-2614, June 2006.
- [90] S. K. Mandal, S. Sural, and A. Patra, “ANN- and PSO-based synthesis of on-chip spiral inductors for RF ICs,” *IEEE Trans. Comput.-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, January 2008.
- [91] Y. Lee and D. S. Filipovic, “ANN based electromagnetic models for the design of RF MEMS switches,” *IEEE Microwave and Wireless Components Letters*, vol. 15, no. 11, pp. 823-825, November 2005.
- [92] R. K. Mishra and A. Patnaik, “Neural network-based CAD model for the design of square-patch antennas,” *IEEE Trans. Antennas Propagat.*, vol. 46, pp. 1890-1891, Dec. 1998.
- [93] N. P. Somasiri, X. Chen, I. D. Robertson, and A. A. Rezazadeh, “Neural network modeler for design optimization of multilayer patch antennas,” *IEE Proceedings of Microwaves, Antennas Propagat.*, vol. 151, no. 6, pp. 514-518, December 2004.
- [94] Y. Kim, S. Keely, J. Ghosh, and H. Ling, “Application of artificial neural networks to broadband antenna design based on a parametric frequency model,” *IEEE Trans. Antennas Propagat.*, vol. 55, no. 3, pp. 669-674, March 2007.
- [95] I. Ratner, H. Ali, and E. M. Petriu, “Neural network simulation of a dielectric ring resonator antenna,” *J. of System Architecture*, pp. 569-581, April 1998.

- [96] H. J. Delgado, M. H. Thursby, and F. M. Ham, "A novel neural network for the synthesis of antennas and microwave devices," *IEEE Trans. Neural Networks*, vol. 16, no. 6, pp. 1590-1600, November 2005.
- [97] H. Sharma and Q. J. Zhang, "Transient electromagnetic modeling using recurrent neural networks," *IEEE MTT-S Int. Microwave Symp.*, pp. 1597-1600, June 2005.
- [98] 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 Applied Computational Electromagnetics Society*, vol. 18, no. 2, pp. 89-97, July 2003.
- [99] A. Zhang, H. Zhang, H. Li, and D. Chen, "A recurrent neural networks based modeling approach for internal circuits of electronic devices," *20<sup>th</sup> Int. Symposium on EMC*, pp. 293-296, 2009.
- [100] Y. Cao, X. Chen, and G. Wang, "Dynamic behavioural modeling of nonlinear microwave devices using real-time recurrent neural network," *IEEE Trans. On electron Devices*, vol. 56, no. 5, pp. 1020-1026, May 2009.
- [101] Y. Cao, R. Ding, Q. J. Zhang, "State space dynamic neural network technique for high-speed IC applications: modeling and stability analysis," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 6, pp. 2398-2409, June 2006
- [102] E. A. Soliman, M. H. Bakr, and N. K. Nikolova, "Neural networks-method of moments (NN-MoM) for the efficient filling of the coupling matrix," *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 6, pp. 1521-1529, June 2004.

- [103] E. K. Murphy, V. V. Yakovlev, "RBF network optimization of complex microwave systems represented by small FDTD modeling data sets," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 7, pp. 3069-3083, July 2006.
- [104] J. Corcoles, M. A. Gonjalez, J. Zapata, "CAD of stacked patch antennas through multipurpose admittance matrices from FEM and neural networks," *Microwave and Optical Technology Letters*, vol. 50, no. 9, pp. 2411-2416, Sep. 2006.
- [105] L. Zhang, J. Xu, M. C. E. Yagoub, and Q. J. Zhang, "Efficient analytical formulation and sensitivity analysis of neuro-space mapping for nonlinear microwave device modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 53, no. 9, pp. 2752-2767, Sept. 2005.
- [106] S. Liao, H. Kabir, and Q. J. Zhang, "Neural network EM-Field based modeling for 3D substructures in finite element method," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 517-520, June 2009.
- [107] P. Burrascano, M. Dionigi, C. Fancelli, and M. Mongiardo, "A neural network model for CAD and optimization of microwave filters," *IEEE MTT-S Int. Microwave Symp. Digest*, vol. 1, 13- 16, June 1998.
- [108] G. Fedi, A. Gaggelli, S. Manetti and G. Pelosi, "Direct-coupled cavity filters design using a hybrid feedforward neural network- finite elements procedure," *Int. J. of RF and Microwave CAE*, vol. 9, 287-296, May 1999.
- [109] G. Fedi, A. Gaggelli, S. Manetti, and G. Pelosi, "A finite-element neural-network approach to microwave filter design," *Microwave and Optical technology lett.*, vol. 19, 36-38, Sept. 1998.

- [110] G. Fedi, S. Manetti, G. Pelosi and S. Selleri, "Design of cylindrical posts in rectangular waveguide by neural network approach," *IEEE Int. Symp. Antenna and Propaga.*, vol. 2, July 2000, 1054-1057.
- [111] V. Miraftab and M. Yu, "Innovative combine RF/Microwave filter EM synthesis and design using neural networks," *URSI, Int. Symp. on Signals, Systems, and Electronics*, pp. 1-4, 2007.
- [112] C. Kudsia, R. Cameron, and W. C. Tang, "Innovations in microwave filters and multiplexing networks for communications satellite systems", *IEEE Trans. Microwave Theory Tech.*, vol. 40, No. 6, 1133-1149, June 1992.
- [113] X. Li, J. Gao, J. Yook and X. Chen, "Bandpass filter design by artificial neural network modeling," in *Proceedings of Asia Pacific Microwave Conf.*, 2005.
- [114] P. M. Watson, C. Cho and K.C. Gupta, "Electromagnetic-Artificial neural network model for synthesis of physical dimensions for multilayer asymmetric coupled transmission structures," *Int. J. of RF and Microwave CAE*, vol. 9, 175-186, May 1999.
- [115] E. N. R. Q. Fernandes, P. H. F. Silva, M. A. B. Melo and A. G. d'Assuncao, "A new neural network model for accurate analysis of microstrip filters on PBG structure," in *Proc. of European Microwave Conf.*, Italy, Sept. 2002.
- [116] F. Gunes and N. Turker, "Artificial neural networks in their simplest forms for analysis and synthesis of RF/Microwave planar transmission lines," *Int. J. of RF and Microwave CAE*, vol. 15, No. 6, 587-600, Nov. 2005.
- [117] M. G. Banciu, E. Ambikairajah and R. Ramer, "Microstrip filter design using

- FDTD and neural networks,” *Microwave and Optical Technology Letters*, vol. 34, No. 3, August 2002, 219-224.
- [118] R. J. Pratap, J. H. Lee, S. Pinel, G.S. May, J. Laskar and E.M. Tentzeris, “Millimeter wave RF front end design using neuro-genetic algorithms,” in *Proc. of Electronic Component and Technology Conf.*, vol. 2, May-June 2005, 1802-1806.
- [119] S. F. Peik, G. Coutts and R. R. Mansour, “Application of neural networks in microwave circuit modeling,” in *Proc. of IEEE Canadian Conf. on Electrical and Computer Eng.*, May 1998, pp. 928-931.
- [120] M. Li, X. Li, X. Liao and J. Yu, “Modeling and optimization of microwave circuits and devices using wavelet neural networks,” *IEEE International Conference on Communications, Circuits and Systems*, June 2004, 1471-1475.
- [121] A. S. Ciminski, “Artificial neural networks modeling for computer-aided design for planar band-rejection filter,” in *Int. Conf. on Microwave, Radar and Wireless Communications*, vol. 2, 2004, 551-554.
- [122] A. R. Harish, “Neural network based yield prediction of microwave filters,” in *Proc. of IEEE APACE*, Shah Alam, Malaysia, 2003, 30-33.
- [123] M. M. Vai, S. Wu, B. Li, and S. Prasad, “Reverse modeling of microwave circuits with bidirectional neural network models,” *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 1492-1494, Oct. 1998.
- [124] S. Selleri, S. Manetti, and G. Pelosi, “Neural network applications in microwave device design,” *Int. J. RF and Microwave CAE.*, vol. 12, pp. 90-97, Jan. 2002.

- [125] J. W. Bandler, M. A. Ismail, J. E. Rayas-sanchez, and Q. J. Zhang, "Neural inverse space mapping (NISM) optimization for EM-base microwave design," *Int. J. RF and Microwave CAE.*, vol. 13, pp. 136-147, Mar. 2003.
- [126] I. Bahl, *Lumped Elements for RF and Microwave Circuits*, Boston: Artech House, 2003.
- [127] *NeuroModelerPlus*, Q.J. Zhang, Department of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada.
- [128] J. Urias, D. Hidalgo, P. Melin, and O. Castillo, "A new method for response integration in modular neural networks using type-2 fuzzy logic for biometric systems," in *Proc. of Int. Joint Conf. on Neural Networks*, Aug. 12–17, 2007, Orlando, Florida, pp. 311–315.
- [129] L. Wang, S. A. Rizvi, and N. M. Nasrabadi, "A predictive residual VQ using modular neural network vector predictor," *IEEE Int. Joint Conf. on Neural Networks*, Jul. 31–Aug. 4, 2005, Montreal, Canada, pp. 2953–2956.
- [130] G. Martinez, P. Melin, and O. Castillo, "Optimization of Modular neural networks using hierarchical genetic algorithms applied to speech recognition," in *Proc. of Int. Joint Conf. on Neural Networks*, Jul. 31–Aug. 4, 2005, Montreal, Canada, pp. 1400–1405.
- [131] P. Melin, A. Mancilla, M. Lopez, and O. Castillo, "Pattern recognition for industrial monitoring and security using the fuzzy sugeno integral and modular neural networks," in *Proc. of Int. Joint Conf. on Neural Networks*, Orlando, Florida, Aug. 12–17, 2007, pp. 2977–2981.

- [132] H. K. Kwan and Y. Cai, "A fuzzy neural network and its application to pattern recognition," *IEEE Trans. Fuzzy Systems*, vol. 2, no. 3, pp. 185-193, Aug. 1994.
- [133] U. Lahiri, A. K. Pradhan, and S. Mukhopadhyaya, "Modular neural network-based directional relay for transmission line protection," *IEEE Trans. on Power Systems*, vol. 20, no. 4, pp. 2154-2155, Nov. 2005.
- [134] R. Anand, K. Mehrotra, C. K. Mohan, and S. Ranka, "Efficient classification for multiclass problems using modular neural networks," *IEEE Trans. on Neural Networks*, vol. 6, no. 1, pp. 117-124, Jan. 1995.
- [135] R. J. Cameron, C. M. Kudsia and R. R. Mansour, *Microwave Filters for Communication Systems: Fundamentals, Design and Applications*: John Wiley & Sons, Inc., 2007.
- [136] J. Zheng and M. Yu, "Rigorous mode-matching method of circular to off-center rectangular side-coupled waveguide junctions for filter applications," *IEEE Trans. Microwave Theory Tech.*, vol. 55, no. 11, pp. 2365-2373, Nov. 2007.
- [137] M. Yu, D. J. Smith, A. Sivadas, and W. Fitzpatrick, "A dual mode filter with trifurcated iris and reduced footprint," *IEEE MTT-S Int. Microwave Symp. Dig.*, Seattle, WA, Jun. 2002, vol. 3, pp. 1457-1460.
- [138] M. A. Ismail, D. Smith, A. Panariello, Y. Wang, and M. Yu, "EM-based design of large-scale dielectric-resonator filters and multiplexers by space mapping," *IEEE Trans. Microwave Theory Tech.*, vol. 52, Issue 1, Part 2, pp. 386-392, Jan. 2004.
- [139] Ansoft HFSS, ver. 11, Ansoft Corporation, Pittsburgh, PA, 2007.