NEURAL SPACE MAPPING METHODS

FOR MODELING AND DESIGN

OF MICROWAVE CIRCUITS

To my wife Cristina

NEURAL SPACE MAPPING METHODS

FOR MODELING AND DESIGN

OF MICROWAVE CIRCUITS

By

JOSE ERNESTO RAYAS-SANCHEZ, M.Sc. (Eng.)

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

June 2001

DOCTOR OF PHILOSOPHY (2001) McMASTER UNIVERSITY
(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: **Neural Space Mapping Methods for Modeling and Design of Microwave Circuits**

AUTHOR: José Ernesto Rayas-Sánchez
B.Sc. (Eng) (Instituto Tecnológico y de Estudios Superiores de Occidente, ITESO)
M.Sc. (Eng) (Intituto Tecnológico y de Estudios Superiores de Monterrey, ITESM)

SUPERVISOR: J.W. Bandler, Professor Emeritus, Department of Electrical and Computer Engineering
B.Sc.(Eng), Ph.D., D.Sc.(Eng) (University of London)
D.I.C. (Imperial College)
P.Eng. (Province of Ontario)
C.Eng., FIEE (United Kingdom)
Fellow, IEEE
Fellow, Royal Society of Canada
Fellow, Engineering Institute of Canada

NUMBER OF PAGES: xx, 172

ABSTRACT

This thesis contributes to the development of novel methods and techniques for computer-aided electromagnetics (EM)-based modeling and design of microwave circuits exploiting two previously unrelated technologies: space mapping (SM) and artificial neural networks (ANNs).

The conventional approach to EM-based modeling of microwave circuits is reviewed, as well as other state-of-the-art neuromodeling techniques. The fundamental space mapping concept is also reviewed. Developing neuromodels based on space mapping technology is addressed. Several SM-based neuromodeling techniques are described. Contrast with other neuromodeling approaches is realized.

An algorithmic procedure to design, called Neural Space Mapping (NSM) optimization, is described. NSM enhances an SM-based neuromodel at each iteration. Other techniques for optimization of microwave circuits using artificial neural networks are reviewed.

Efficient EM-based statistical analysis and yield optimization of microwave components using SM-based neuromodels is described. Other yield-driven EM optimization strategies are briefly reviewed. An innovative strategy to avoid extra EM simulations when asymmetric variations in the physical parameters are assumed is described.

Neural Inverse Space Mapping (NISM) optimization for EM-based microwave design is described. A neural network approximates the inverse mapping at each iteration. The NISM step simply consists of evaluating this neural network at the optimal empirical solution. NISM step is proved to be a quasi-Newton step when the amount of nonlinearity in the inverse neuromapping is small. NISM optimization is compared with other SM-based optimization algorithms.

The theoretical developments are implemented using available software on several advanced and industrially relevant microwave circuits.

Suggestions for further research are provided.

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to Dr. J. W. Bandler, director of research in the Simulation Optimization Systems Research Laboratory at McMaster University and President of Bandler Corporation, for his encouragement, expert guidance and keen supervision throughout the course of this work. He also thanks Dr. S.H. Chisholm, Dr. Q.J. Zhang, Dr. A.D. Spence and Dr. N. Georgieva, members of his Supervisory Committee, for their continuing interest and suggestions.

The author has greatly benefited from working with the OSA90/hope™ microwave computer-aided design system formerly developed by Optimization Systems Associates Inc., now part of Agilent EEsof EDA, and with the ***em***™ full wave electromagnetic field simulator developed by Sonnet Software, Inc. The author is grateful to Dr. J.W. Bandler, former President of Optimization Systems Associates Inc., and to Dr. J.C. Rautio, President of Sonnet Software Inc., for making the OSA90/hope™ and ***em***™ packages available for this work.

Special thanks are due to Dr. Q.J. Zhang of Carleton University, for fruitful cooperation and helpful technical discussions. The author has greatly benefited from working with the software system *NeuroModeler* developed by Dr. Zhang's group at Carleton University.

The author offers his gratitude to Dr. R.M. Biernacki and S.H. Chen, former members of McMaster University and Optimization Systems Associates Inc., now with Agilent EEsof EDA, for their support during the initial stages of this work.

It is the author's pleasure to acknowledge fruitful collaboration and stimulating discussions with his colleagues of the Simulation Optimization Systems Research Laboratory at McMaster University: M.A. Ismail, Dr. M.H. Bakr (now with the University of Victoria), Q.S. Cheng, A.S. Mohamed, S. Porter, Dr. T. Günel (Istanbul Technical University, Turkey), and Dr. F. Guo (now with Mitec, Montreal).

The author gratefully acknowledge the financial assistance provided by the Consejo Nacional de Ciencia y Tecnología of Mexico through scholarship number 54693, by the Instituto Tecnológico y de Estudios Superiores de Occidente of Mexico through a leave of absence with financial aid, by the Department of Electrical and Computer Engineering at McMaster University through a Research Assistantship, by the Ministry of Training for Colleges and Universities in Ontario through an Ontario Graduate Scholarship, by the Natural Sciences and Engineering Research Council of Canada through Grants OGP0007239, STP0201832, STR234854-00, and by the Micronet Network of Centres of Excellence of Canada.

Finally, special thanks are due to my family: my wife Cristina and my children Abril, Gilberto and Yazmin, for their understanding, patience and continuous loving support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

For nearly a half a century computer-aided design (CAD) of electronic circuits have evolved from a set of special purpose, rudimentary simulators and techniques to a variety of highly flexible and interactive, general purpose software systems, with amazing visualization capabilities.

The first efforts to incorporate the computer as a design tool were made in the well-established area of filter synthesis (see Director, 1973). By the middle of the 1950s a number of successful filter design techniques had been developed. Aaron (1956) proposed using a least-squares approach for the realization of transfer functions that approximate a given set of design specifications. Historical contributions by Desoer and Mitra (1961), Calahan (1965), and Smith and Temes (1965) followed after Aaron's philosophy, where interactive optimization methods were proposed for specific classes of filters.

At the same time that synthesis procedures were being developed, significant work was being carried out in the area of circuit simulation: a good general-purpose analysis program was required in order for the computer to be an effective design tool. Similar to the field of network synthesis, the initial attempts at computerized circuit simulation were limited to more or less direct implementation of standard analysis methods. Some of the first general-purpose simulation programs appeared: TAP (Branin, 1962), CORNAP (Pottle, 1965), ECAP (IBM, 1965) and AEDNET (Katzenelson, 1966).

With the advances in sparse matrix methods, numerical integration techniques, and sensitivity calculation methods (adjoint network approach), the first computationally efficient, general purpose simulation programs became available: ECAP-II (Branin et. al., 1971) and CANCER (Nagel and Rohrer, 1971). From the latter evolved SPICE during the mid-1970s, which became the most popular general purpose circuit simulator (see Tuinenga, 1992).

In parallel to the development of circuit modeling, simulation and optimization, numerical electromagnetics was also emerging. The most influential methods for computational electromagnetics were proposed in late 1960s and early 1970s: the Finite Difference Time Domain method (Yee, 1966), the Method of Moments (Harrington, 1967), the Finite Element method for electrical engineering (Silvester, 1969), and the Transmission-Line Matrix method (Akhtarzad and Johns, 1973).

The increasing availability of low-cost yet powerful personal computers in the 1980s and 1990s, as well as the continuing progress in numerical techniques and software engineering, made from the CAD systems an every day tool used by most electronic designers. This explosion in software tools evolved in three relatively separated roads: low frequency mixed-mode CAD, high frequency analog CAD, and electromagnetic CAD.

The arena of low frequency mixed-mode CAD typically includes the areas of digital design languages (Verilog, VHDL, etc.), analog/mixed signal design tools, digital IC design, active device modeling, IC/ASIC design, PLD/FPGA design, PC-board design, hardware/software co-design and co-verification, functional and physical design.

These CAD tools have been significantly shaped by the IC industry needs, and they are usually referred to as electronic design automation (EDA) tools. Martin (1999) and Geppert (2000) provide an analysis and forecast in this arena. A more comprehensive review and perspective is provided by Camposano and Pedram (2000).

The field of high frequency analog CAD has focused on the development of circuit-theoretic design tools for flexible and fast interactive operation. They usually provide modeling capabilities based on quasi-static approximations, linear and nonlinear circuit simulation, frequency domain, steady-state time domain and transient analysis. Some of these tools provide powerful optimization algorithms. Some of them also provide interfacing capabilities to electromagnetic field solvers for design verification and optimization. Examples of this family of tools are Touchstone (1985), Super-Compact (1986), OSA90 (1990) and Transim (Christoffersen, Mughal and Steer, 2000).

The arena of EM field solvers evolved in four different classes of simulators: 2-dimensional, 3-dimensional planar (laterally open or closed box), and 3-dimensional arbitrary-geometry. Swanson (1991, 1998) made an excellent review of commercially available CAD tools for electromagnetic simulation. Mirotznik and Prather (1997) provided a useful guide on how to choose EM software.

The needs of the microwave and wireless electronics industry have significantly shaped the evolvement of CAD tools for EM and high frequency circuit design. Microwave structures working at increasingly higher frequencies made classical empirical models less reliable in predicting the actual behavior of manufactured components. Emerging microwave technologies (such us coplanar waveguide (CPW) circuits, multiple-layered circuits, and integrated circuit antennas) pushed towards the development of more EM-based models. Stringent design specifications and the drive for time-to-market in the wireless industry demanded sophisticated optimization-based design algorithms.

The rapid growth in the communications and high-speed electronics industries is demanding more integration between these three CAD arenas. Computer-aided electronic design is now claiming a holistic approach due to the facts that switching speeds in digital hardware are now in the microwave range, and that RF modules are now part of commercial systems-on-a-chip. Additionally, thermal and mechanical considerations are increasingly impacting this design process, enforcing an even more multidisciplinary and interrelated approach for the future automated design.

A recent new trend in microwave and millimeter-wave CAD is the use of artificial neural networks (ANNs) for efficient exploitation of EM simulators (see Gupta, 1998). Similarly, space mapping (SM) emerged as an innovative approach to automated microwave design that combines the accuracy of EM field solvers with the speed of circuit simulators (see Bandler et. al., 1994). The work in this thesis aims at the intersection of these two emerging technologies.

This thesis focuses on the development of novel methods and techniques for computer aided modeling, design and optimization of microwave circuits exploiting two previously unconnected technologies: space mapping and artificial neural networks.

Chapter 2 addresses the development of artificial neural network models based on space mapping technology. We review the fundamental space mapping concept. We also review the conventional approach to EM-based modeling of microwave devices using ANNs, as well as other state-of-the-art techniques for neuromodeling. The SM-based neuromodeling strategy is described. We illustrate how frequency-sensitive neuromappings can effectively expand the range of validity of many empirical models based on quasi-static approximations. Several practical microwave examples illustrate our techniques.

In Chapter 3 we change our focus of interest from modeling to design by optimization. We describe in this chapter an algorithmic procedure to design by enhancing an SM-based neuromodel at each iteration. Neural Space Mapping (NSM) optimization exploits the modeling techniques proposed in Chapter 2. Other techniques for optimization of microwave circuits using ANNs are briefly reviewed. We also review the concept of space mapping with frequency included. NSM optimization requires a number of up-front EM-simulations at the starting point, and employs a novel procedure to avoid troublesome parameter extraction at each iteration. An HTS microstrip filter and a bandstop microstrip filter with open stubs illustrate our algorithm.

Chapter 4 deals with the EM-based statistical analysis and yield optimization of microwave components using SM-based neuromodels. We briefly review other yield-driven EM optimization strategies. We formulate the problem of statistical analysis and design. We describe a creative way to avoid extra EM simulations when asymmetric variations in the physical parameters are considered. The EM-based yield analysis and optimization of an HTS microstrip filter illustrate our strategies.

In Chapter 5 we describe Neural Inverse Space Mapping (NISM): an efficient optimization method for EM-based microwave design. NISM is the first SM-based optimization algorithm that explicitly makes use of the inverse of the mapping from the EM input space to the empirical input space. A neural network approximates this inverse mapping at each iteration. NISM is contrasted with NSM as well as with other SM-based optimization algorithms through several design examples.

We conclude the thesis in Chapter 6, providing some suggestions for further research.

The author’s original contributions presented in this thesis are:

(1) Formulation and development of the Space Mapping-based Neuromodeling techniques, and their implementation using the software system OSA90/hope (1997).

(2) Formulation and development of the Neural Space Mapping (NSM) optimization algorithm, as well as its implementation in the software system OSA90/hope (1997).

(3) Formulation and implementation of the EM-based yield optimization utilizing Space Mapping-based neuromodels.

(4) Formulation and development of the Neural Inverse Space Mapping (NISM) Optimization algorithm, as well as its fully automated implementation in Matlab™ (1998).

(5) Implementation in Matlab™ (1998) of a fully automated algorithm for training artificial neural networks following a network growing strategy.

(6) Formulation and implementation in Matlab™ (1999) of a fully automated algorithm for statistical parameter extraction.

(7) Design of graphical representations for algorithmic strategies typically used in modeling and optimization of microwave circuits.

In collaboration with J.W. Bandler, F. Wang and Q.J. Zhang, the author originally implemented Space Mapping-based neuromodels using the software system *NeuroModeler* (1999).

Together with M.H. Bakr, J.W. Bandler, K. Madsen and J. Søndergaard the author collaborated in the original development of the Trust Region Aggressive Space Mapping optimization algorithm exploiting surrogate models.

Together with M.H. Bakr, J.W. Bandler, M.A. Ismail, Q.S. Cheng and S. Porter the author collaborated in the development of the software system SMX (2001).

Together with J.W. Bandler, N. Georgieva, M.A. Ismail and Q.J. Zhang, the author collaborated in the original development of a generalized Space Mapping tableau approach to device modeling.

Together with J.W. Bandler and M.A. Ismail, the author collaborated in the original development of an expanded Space Mapping design framework exploiting preassigned parameters.

Chapter 2

SPACE MAPPING BASED NEUROMODELING

2.1 INTRODUCTION

A powerful new concept in neuromodeling of microwave circuits based on Space Mapping technology is described in this chapter. The ability of Artificial Neural Networks (ANN) to model high-dimensional and highly nonlinear problems is exploited in the implementation of the Space Mapping concept. By taking advantage of the vast set of empirical models already available, Space Mapping based neuromodels decrease the number of EM simulations for training, improve generalization ability and reduce the complexity of the ANN topology with respect to the classical neuromodeling approach.

Five innovative techniques are proposed to create Space Mapping based neuromodels for microwave circuits: Space Mapped Neuromodeling (SMN), Frequency-Dependent Space Mapped Neuromodeling (FDSMN), Frequency Space Mapped Neuromodeling (FSMN), Frequency Mapped Neuromodeling (FMN) and Frequency Partial-Space Mapped Neuromodeling (FPSM). Excepting SMN, all these approaches establish a frequency-sensitive neuromapping to expand the frequency region of accuracy of the empirical models already available for microwave components that were developed using quasi-static analysis. We contrast our approach with the classical neuromodeling approach employed in the microwave arena, as well as with other state-of-the-art neuromodeling techniques.

Simultaneously with the work of Devabhaktuni, Xi, Wang and Zhang (1999), we used for the first time Huber optimization to efficiently train the ANNs (see Bandler, Ismail, Rayas-Sánchez and Zhang, 1999a).

The Space Mapping based neuromodeling techniques are illustrated by two case studies: a microstrip right angle bend and a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter.

2.2 THE SPACE MAPPING CONCEPT

Space Mapping (SM) is a novel concept for circuit design and optimization that combines the computational efficiency of coarse models with the accuracy of fine models. The coarse models are typically empirical equivalent circuit engineering models, which are computationally very efficient but often have a limited validity range for their parameters, beyond which the simulation results may become inaccurate. On the other hand, detailed or "fine" models can be provided by an electromagnetic (EM) simulator, or even by direct measurements: they are very accurate but CPU intensive. The SM technique establishes a mathematical link between the coarse and the fine models, and directs the bulk of CPU intensive evaluations to the coarse model, while preserving the accuracy and confidence offered by the fine model. The SM technique was originally developed by Bandler, Biernacki, Chen, Grobelny and Hemmers (1994).

Let the vectors $\boldsymbol{x}_c$ and $\boldsymbol{x}_f$ represent the design parameters of the coarse and fine models, respectively, and $\boldsymbol{R}_c(\boldsymbol{x}_c)$ and $\boldsymbol{R}_f(\boldsymbol{x}_f)$ the corresponding model responses. $\boldsymbol{R}_c$ is much faster to calculate but less accurate than $\boldsymbol{R}_f$.

As illustrated in Fig. 2.1 the aim of SM optimization is to find an appropriate mapping $\boldsymbol{P}$ from the fine model parameter space $\boldsymbol{x}_f$ to the coarse model parameter space $\boldsymbol{x}_c$

$$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f) \tag{2-1}$$

such that

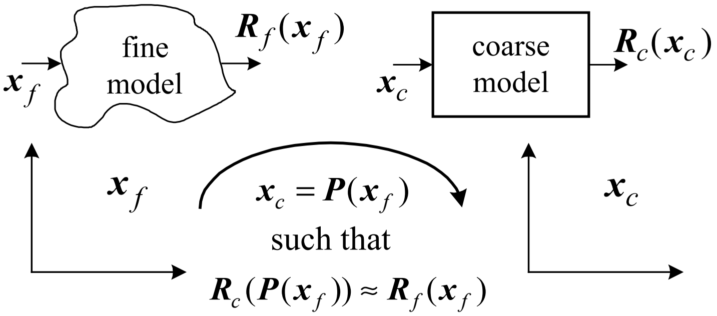
$$\boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f)) \approx \boldsymbol{R}_f(\boldsymbol{x}_f) \tag{2-2}$$

Once a mapping $\boldsymbol{P}$ valid in the region of interest is found, the coarse model can be used for fast and accurate simulations in that region.

2.3 NEUROMODELING MICROWAVE CIRCUITS

Artificial neural networks are particularly suitable in modeling high-dimensional and highly nonlinear devices, as those found in the microwave area, due to their ability to learn and generalize from data, their non-linear processing nature, and their massively parallel structure.

It has been shown by White, Gallant, Hornik, Stinchcombe and Wooldridge (1992) that standard multilayer feedforward networks can approximate any measurable function to any desired level of accuracy, provided a deterministic relationship between



fine
model
$\boldsymbol{x}_f$
$\boldsymbol{R}_f(\boldsymbol{x}_f)$
coarse
model
$\boldsymbol{x}_c$
$\boldsymbol{R}_c(\boldsymbol{x}_c)$
$\boldsymbol{x}_f$
$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f)$
such that
$\boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f)) \approx \boldsymbol{R}_f(\boldsymbol{x}_f)$
$\boldsymbol{x}_c$

Fig. 2.1 Illustration of the aim of Space Mapping (SM).

input and target exists. Following Haykin (1999), ANN that are too small cannot approximate the desired input-output relationship, while those with too many internal parameters perform correctly on the learning set, but give poor generalization ability.

According to Burrascano and Mongiardo (1999), the most widely used ANN paradigm in the microwave arena is the multi-layer perceptron (MLP), which is usually trained by the well established backpropagation algorithm.

ANN models are computationally more efficient than EM or physics-based models and can be more accurate than empirical models. ANNs are suitable models for microwave circuit yield optimization and statistical design, as demonstrated by Zaabab, Zhang and Nakhla (1995) as well as by Burrascano, Dionigi, Fancelli and Mongiardo (1998).

In the conventional neuromodeling approach, an ANN is trained such that it approximates the fine model response $\boldsymbol{R}_f$ in a region of interest for the design parameters $\boldsymbol{x}_f$ and operating frequency ω, as illustrated in Fig. 2.2, where vector $\boldsymbol{w}$ contains the internal parameters of the ANN (weighting factors, bias, etc.). Once the ANN is trained with sufficient learning samples, that is, once the optimal $\boldsymbol{w}$ is found, the ANN can be used as a fast and accurate model within the region of interest. The complexity of the

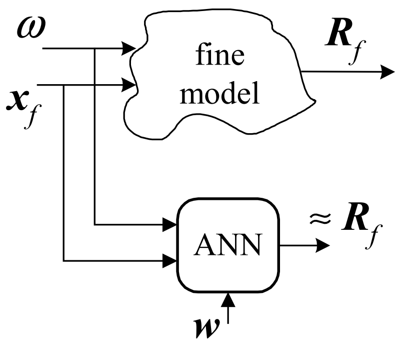


Fig. 2.2 Conventional neuromodeling approach.

ANN must be properly selected: the number of internal free parameters has to be sufficiently large to achieve a small learning error, and sufficiently small to avoid a poor generalization performance.

This training can be seen as an optimization problem where the internal parameters of the neural network are adjusted such that the ANN model best fits the training data.

A large amount of training data is usually needed to ensure model accuracy. For microwave circuits this training data is usually obtained by either EM simulation or by measurement. Generating a large amount of training data can be very expensive for microwave problems because the simulation/measurements must be performed for many combinations of different values of geometrical, material, process and input signal parameters. This is the main drawback of the conventional ANN modeling approach. Without sufficient training data, the neural models developed may not be reliable. Additionally, it has been shown by Stone (1982) that the number of learning samples needed to approximate a function grows exponentially with the ratio between the dimensionality and its degree of smoothness; this well known effect is called "the curse of dimensionality", e.g., Haykin (1999).

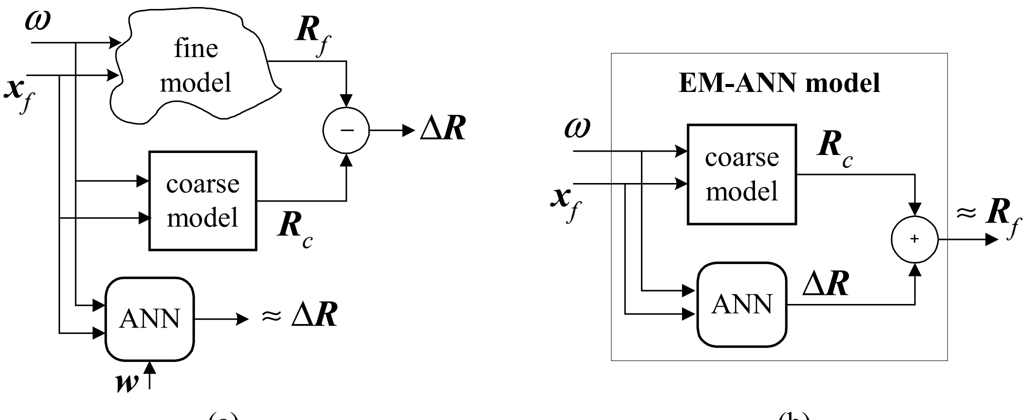
A popular alternative to reduce the dimension of the learning set is to carefully select the learning points using the Design of Experiments (DoE) methodology, to ensure adequate parameter coverage, as in the work by Creech, Paul, Lesniak, Jenkins and Calcatera (1997).

Another way to speed up the learning process is proposed in the work of Burrascano and Mongiardo (1999) by means of preliminary neural clusterization of similar responses using the Self Organizing Feature Map (SOM) approach.

Innovative strategies have been proposed to reduce the learning data needed and to improve the generalization capabilities of an ANN by incorporating empirical models: the hybrid EM-ANN modeling approach, the Prior Knowledge Input (PKI) modeling method, and the Knowledge Based ANN (KBNN) approach.

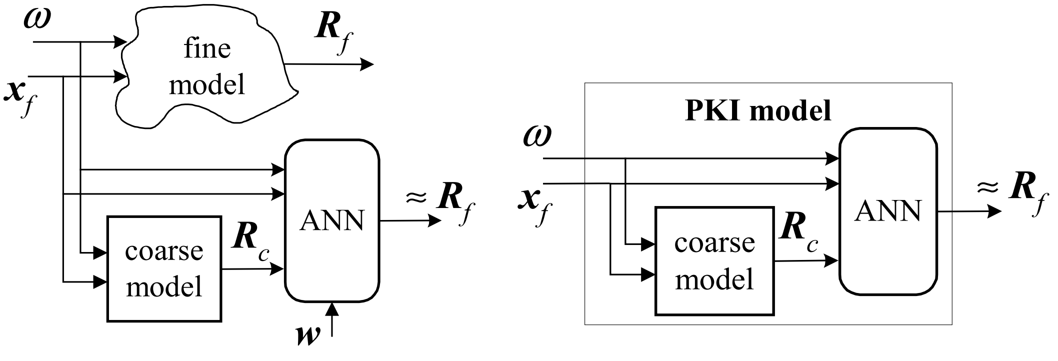
Proposed by Watson and Gupta (1996), the Hybrid EM-ANN modeling approach makes use of the difference in *S*-parameters between the available coarse model and the fine model to train the corresponding neural network, as illustrated in Fig. 2.3. The number of fine model simulations is reduced due to a simpler input-output relationship. The Hybrid EM-ANN method is also called Difference Method.

In the Prior Knowledge Input (PKI) method, developed by Watson, Gupta and Mahajan (1998), the coarse model output is used as input for the ANN in addition to the other inputs (physical parameters and frequency). The neural network is trained such that its response is as close as possible to the fine model response for all the data in the training set, as illustrated in Fig. 2.4. According to Watson, Gupta and Mahajan (1998), the PKI approach exhibits better accuracy than the EM-ANN approach, but it requires a



(a) (b)

Fig. 2.3 EM-ANN neuromodeling concept: (a) EM-ANN neuromodeling, (b) EM-ANN model.



(a) (b)

Fig. 2.4 PKI neuromodeling concept: (a) PKI neuromodeling, (b) PKI model.

more complex ANN.

A detailed description of the PKI method as well as numerous illustrations of the Hybrid EM-ANN method can be found in the work by Zhang and Gupta (2000).

In the knowledge based ANN approach (KBNN), developed by Wang and Zhang (1997), the microwave empirical or semi-analytical information is incorporated into the internal structure of the ANN, as illustrated in Fig. 2.5.

Knowledge Based ANNs are non fully connected networks, with one or several layers assigned to the microwave knowledge in the form of single or multidimensional functions, usually obtained from available analytical models based on quasi-static approximations.

By inserting the microwave empirical formulas into the neural network structure, the empirical formulas can be refined or adjusted as part of the overall neural network training process. Since these empirical functions are used for some neurons instead of standard activation functions, knowledge based neural networks do not follow a regular multilayer perceptron and are trained using other methods than the conventional

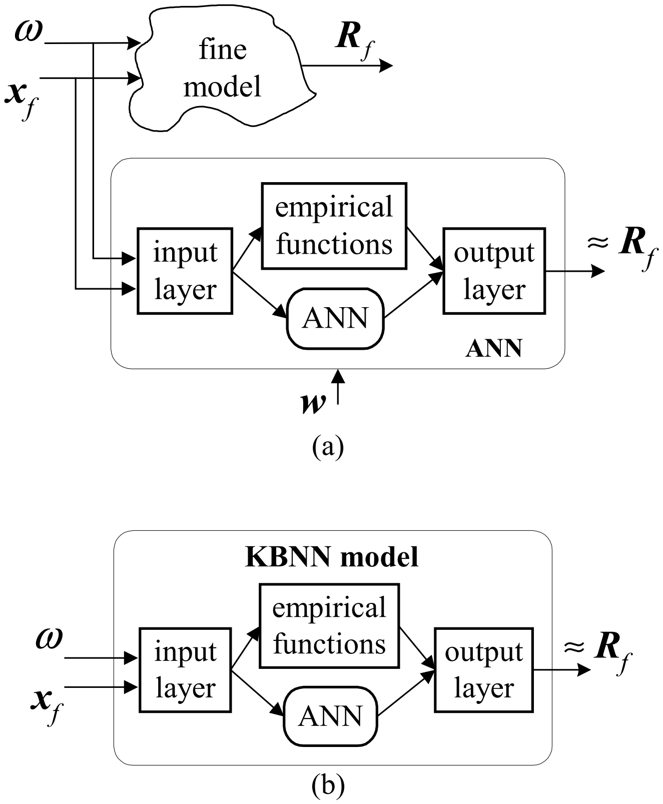


Fig. 2.5 KBNN neuromodeling concept: (a) KBNN neuromodeling, (b) KBNN model.

backpropagation. Two excellent references for KBNNs are provided by Wang (1998) and by Zhang and Gupta (2000).

2.4 SPACE MAPPING BASED NEUROMODELING

We propose innovative schemes to combine SM technology and ANN for the modeling of high frequency components. The fundamental idea is to construct a nonlinear multidimensional vector mapping function $\boldsymbol{P}$ from the fine to the coarse input space using an ANN. This can be done in a variety of ways, to make a better use of the coarse model information for developing the neuromodel. The implicit knowledge in the coarse model, that can be considered as an "expert", not only allows us to decrease the number of learning points needed, but also to reduce the complexity of the ANN and to improve the generalization performance.

In the Space Mapped Neuromodeling (SMN) approach the mapping from the fine to the coarse parameter space is implemented by an ANN. Fig. 2.6 illustrates the SMN concept. We have to find the optimal set of the internal parameters of the ANN, such that the coarse model response is as close as possible to the fine model response for all the learning points.

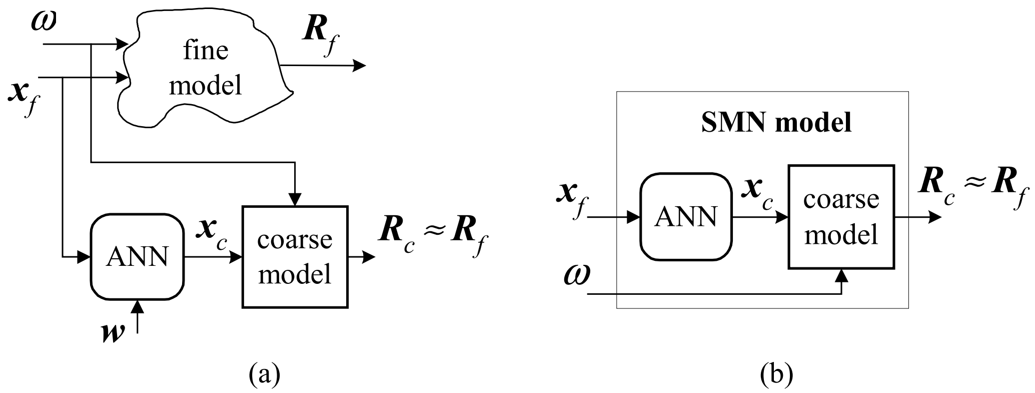
The mapping can be found by solving the optimization problem

$$\min_{\boldsymbol{w}} \left\| [\boldsymbol{e}_1^T \quad \boldsymbol{e}_2^T \quad \cdots \quad \boldsymbol{e}_l^T]^T \right\| \tag{2-3}$$

where vector $\boldsymbol{w}$ contains the internal parameters of the neural network (weights, bias, etc.) selected as optimization variables, l is the total number of learning samples, and $\boldsymbol{e}_k$ is the error vector given by

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_c, \omega_j) \tag{2-4}$$

$$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_{f_i}) \tag{2-5}$$



(a) (b)

Fig. 2.6 Space Mapped neuromodeling concept: (a) SMN neuromodeling, (b) SMN model.

with

$$i = 1, \ldots, B_p \tag{2-6}$$

$$j = 1, \ldots, F_p \tag{2-7}$$

$$k = j + F_p(i-1) \tag{2-8}$$

where B_p is the number of training base points for the input design parameters and F_p is the number of frequency points per frequency sweep. It is seen that the total number of learning samples is $l = B_p\ F_p$. The specific characteristics of $\boldsymbol{P}$ depend on the ANN paradigm chosen, whose internal parameters are in $\boldsymbol{w}$.

In this work, a Huber norm is used in (2-3), exploiting its robust characteristics for data fitting, as shown by Bandler, Chen, Biernacki, Gao, Madsen and Yu (1993).

Once the mapping is found, i.e., once the ANN is trained with an acceptable generalization performance, a space mapped neuromodel for fast, accurate evaluations is immediately available.

2.4.1 Including Frequency in the Neuromapping

Many of the empirical models already available for microwave circuits were developed using methods for quasi-static analysis. For instance, in the case of microstrip circuits, it is often assumed that the mode of wave propagation in the microstrip is pure TEM, as indicated by Gupta, Garg and Bahl (1979). This implies that the effective dielectric constant ε_e and the characteristic impedance Z_o do not vary with frequency. Nevertheless, non-TEM behavior causes ε_e and Z_o to be functions of frequency. Therefore, these empirical models usually yield good accuracy over a limited range of low frequencies.

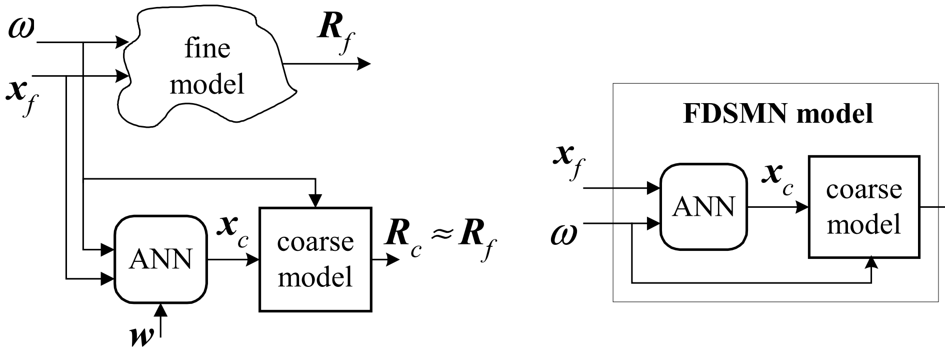
A method to directly overcome this limitation is by establishing a frequency-sensitive mapping from the fine to the coarse input spaces. This is realized by considering frequency as an extra input variable of the ANN that implements the mapping.

In the Frequency-Dependent Space Mapped Neuromodeling (FDSMN) approach, illustrated in Fig. 2.7, both coarse and fine models are simulated at the same frequency, but the mapping from the fine to the coarse parameter space is dependent on the frequency. The mapping is found by solving the same optimization problem stated in (2-3) but substituting (2-4) and (2-5) by

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_c, \omega_j) \tag{2-9}$$

$$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_{f_i}, \omega_j) \tag{2-10}$$

With a more comprehensive scope, the Frequency Space Mapped Neuromodeling (FSMN) technique establishes a mapping not only for the design parameters but also for the frequency variable, such that the coarse model is simulated at a mapped frequency ω_c to match the fine model response.



(a) (b)

Fig. 2.7 Frequency-Dependent Space Mapped Neuromodeling concept: (a) FDSMN neuromodeling, (b) FDSMN model.

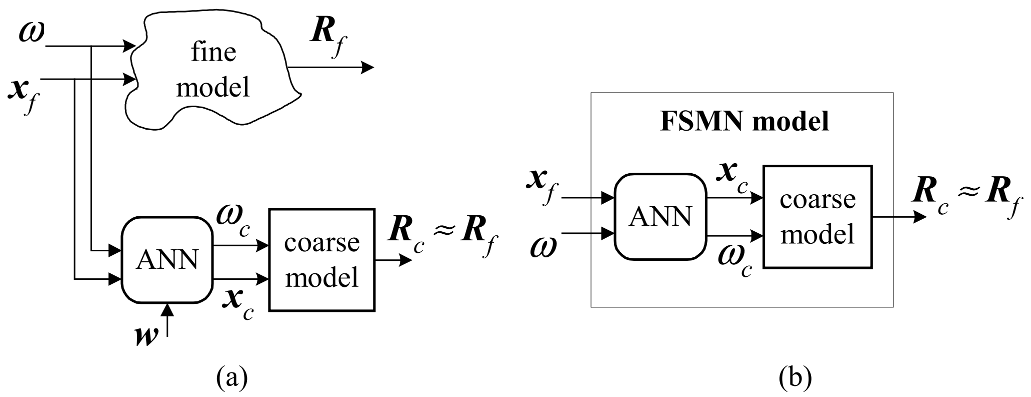


Fig. 2.8 Frequency Space Mapped Neuromodeling concept: (a) FSMN neuromodeling, (b) FSMN model.

FSMN is realized by adding an extra output to the ANN that implements the mapping, as shown in Fig. 2.8. The mapping is found by solving the same optimization problem stated in (2-3) but substituting (2-4) and (2-5) by

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_c, \omega_c) \qquad (2\text{-}11)$$

$$\begin{bmatrix} \boldsymbol{x}_c \\ \omega_c \end{bmatrix} = \boldsymbol{P}(\boldsymbol{x}_{f_i}, \omega_j) \qquad (2\text{-}12)$$

It is common to find microwave problems where the equivalent circuit model behaves almost as the electromagnetic model does but with a shifted frequency response, i.e., the shapes of the responses are nearly identical but shifted. For those cases, a good alignment between both responses can be achieved by simulating the coarse model at a different frequency from the real frequency used by the fine model.

The Frequency Mapped Neuromodeling (FMN) technique implements this strategy, as shown in Fig. 2.9, by simulating the coarse model with the same physical parameters used by the fine model, but at a different frequency ω_c to align both responses.

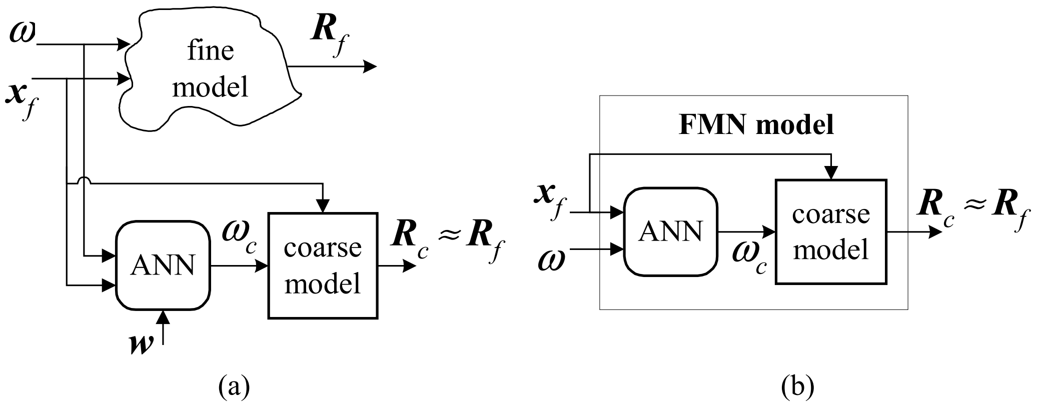


Fig. 2.9 Frequency Mapped Neuromodeling concept: (a) FMN neuromodeling, (b) FMN model.

In the FMN technique the mapping is found by solving the same optimization problem stated in (2-3) but replacing (2-4) and (2-5) by

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_{f_i}, \omega_c) \tag{2-13}$$

$$\omega_c = P(\boldsymbol{x}_{f_i}, \omega_j) \tag{2-14}$$

Mapping the whole set of physical parameters, as in the SMN, FDSMN and FSMN techniques, might lead to singularities in the coarse model response during training. This problem is overcome by establishing a partial mapping for the physical parameters, making an even more efficient use of the implicit knowledge in the coarse model. We have found that mapping only some of the physical parameters can be enough to obtain acceptable accuracy in the neuromodel for many practical microwave problems. This allows us a significant reduction in the ANN complexity w.r.t. the SMN, FDSMN and FSMN techniques and a significant reduction in the time for training, because fewer optimization variables are used. Frequency Partial-Space Mapped Neuromodeling (FPSMN) implements this idea, illustrated in Fig. 2.10.

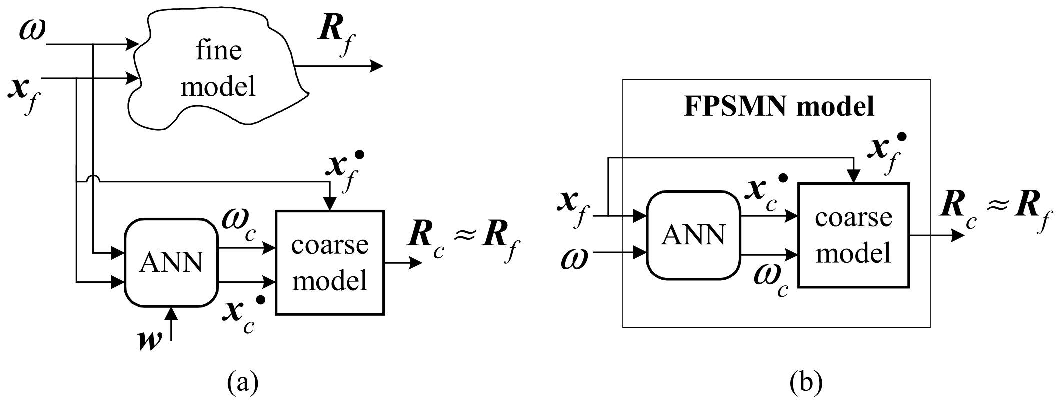


Fig. 2.10 Frequency Partial-Space Mapped Neuromodeling concept: (a) FPSMN neuromodeling, (b) FPSMN model.

The selection of the physical parameters to be mapped in the FPSMN technique can be realized by the user, who usually has a good understanding of the physical structure and is able to detect the most relevant parameters by inspection. When this experience is not available, we can select the parameters to be mapped in FPSMN by using the sensitivity information on the coarse model: the sensitivity of the coarse model response w.r.t. each coarse model parameter can be calculated inexpensively, and this information can be used as a criterion to select the parameters to be mapped.

In the FPSMN technique the mapping is found by solving the same optimization problem stated in (2-3) but replacing (2-4) and (2-5) by

$$e_k = R_f(x_{f_i}, \omega_j) - R_c(x^\bullet_{f_i}, x^\bullet_c, \omega_c) \tag{2-15}$$

$$\begin{bmatrix} x^\bullet_c \\ \omega_c \end{bmatrix} = P(x_{f_i}, \omega_j) \tag{2-16}$$

where $x^\bullet_{f_i}$ vector contains a suitable subset of the design physical parameters x_{f_i} at the ith training base point.

Finally, there can be microwave problems where the complete set of responses contained in $\boldsymbol{R}_f$ is difficult to approximate using a mapped coarse model with a single ANN. In those cases, the learning task can be distributed among a number of ANNs, which in turn divides the output space into a set of subspaces. The corresponding ANNs can then be trained individually, to match each response (or subset of responses) contained in $\boldsymbol{R}_f$. This implies the solution of several independent optimization problems instead of a single one.

2.4.2 Starting Point and Learning Data Samples

The starting point for the optimization problem stated in (2-3) is the initial set of internal parameters of the ANN, denoted by $\boldsymbol{w}^{(0)}$, which is chosen assuming that the coarse model is actually a good model and therefore the mapping is not necessary. In other words, $\boldsymbol{w}^{(0)}$ is chosen such that the ANN implements a unit mapping $\boldsymbol{P}$ ($\boldsymbol{x}_c = \boldsymbol{x}_f$ and/or $\omega_c = \omega$). This is applicable to the five SM-based neuromodeling techniques previously described.

The ANN must be trained to learn the mapping between the fine and the coarse input spaces within the region of interest. In order to maintain a reduced set of learning data samples, an n-dimensional star set for the base learning points (see Fig. 2.11) is considered in this work, as in the work by Biernacki, Bandler, Song and Zhang (1989). It is seen that the number of learning base points for a microwave circuit with n design parameters is $B_p = 2n + 1$.

Since we want to maintain a minimum number of learning points (or fine evaluations), the complexity of the ANN is critical. To avoid a poor generalization

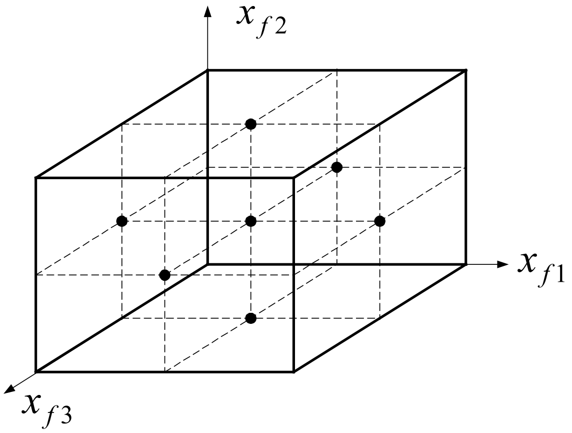


Fig. 2.11 Three-dimensional star set for the learning base points.

performance, we have to use the simplest ANN that gives an acceptable learning error over the region of training and adequate generalization performance in the testing set (see Haykin, 1999).

2.5 SPACE MAPPING BASED NEUROMODELS USING 3-LAYER PERCEPTRONS

We use 3-layer perceptrons (3LP) to implement the mapping in all our SM-based neuromodeling techniques. Fig. 2.12 shows the realization of the SMN approach with a 3-layer perceptron with h hidden neurons. Notice that the FDSMN approach can be implemented by including an additional input for the frequency ω and enabling a v_{n+1} in the 3-layer perceptron in Fig. 2.12. The adaptation of this paradigm to all the other three cases is realized in a similar manner, by considering an additional output for the mapped frequency ω_c and disabling the corresponding inputs and/or outputs.

In Fig. 2.12, $\boldsymbol{x}_f \in \Re^n$ is the vector containing the n input physical parameters to be mapped, $\boldsymbol{v} \in \Re^n$ contains the input signals after scaling, $\boldsymbol{z} \in \Re^h$ is the vector

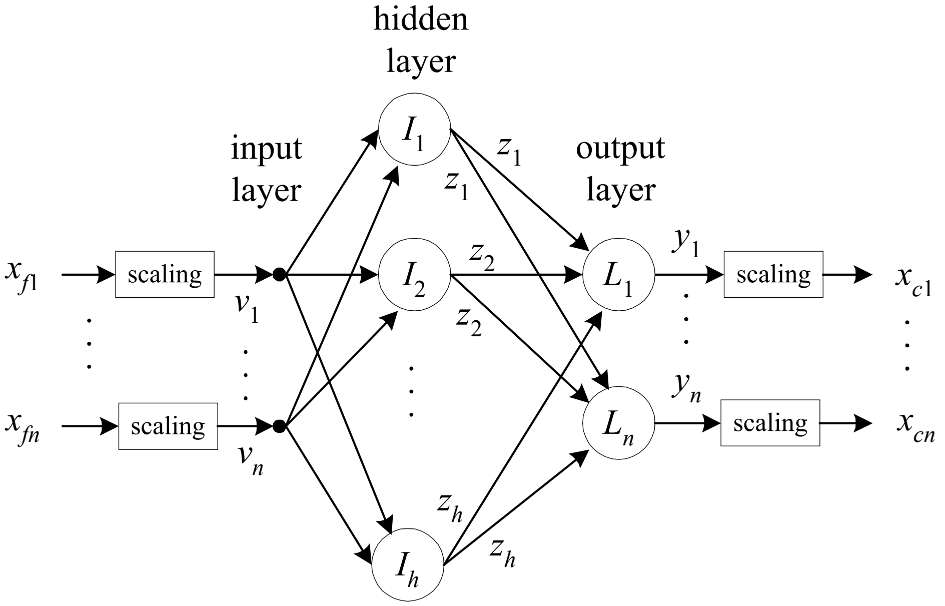


Fig. 2.12 SM neuromapping with 3-layer perceptron.

containing the signals from the h hidden neurons, $\boldsymbol{y} \in \Re^n$ is the vector of output signals before scaling, and vector $\boldsymbol{x}_c \in \Re^n$ has the neuromapping output.

In order to control the relative importance of the different input parameters and to define a suitable dynamical range for the region of interest, an input scaling such that $-1 \leq v_i \leq 1$ is used,

$$v_i = -1 + \frac{2(x_{fi} - x_{fi\,\mathrm{min}})}{(x_{fi\,\mathrm{max}} - x_{fi\,\mathrm{min}})}, \quad i = 1, 2, \cdots, n \tag{2-17}$$

The hidden layer signals are given by

$$z_i = \varphi(b_i^h + \boldsymbol{v}^T \boldsymbol{w}_i^h) \ , \quad i = 1, 2, \cdots, h \tag{2-18}$$

where $\varphi(\cdot)$ is the activation function used for all hidden neurons and $\boldsymbol{w}_i^h$ is the vector of synaptic weights of the i-th hidden neuron,

$$\boldsymbol{w}_i^h = \begin{bmatrix} w_{i1}^h & w_{i2}^h & \cdots & w_{in}^h \end{bmatrix}^T \ , \quad i = 1, 2, \cdots, h \tag{2-19}$$

and $\boldsymbol{b}^h$ is the vector of bias elements for the hidden neurons,

$$\boldsymbol{b}^h = \begin{bmatrix} b_1^h & b_2^h & \cdots & b_h^h \end{bmatrix}^T \tag{2-20}$$

The output layer signals are given by

$$y_i = b_i^o + \boldsymbol{z}^T \boldsymbol{w}_i^o \ , \ \ i = 1, 2, \cdots, n \tag{2-21}$$

where $\boldsymbol{w}_i^o$ is the vector of synaptic weights of the *i*-th output neuron,

$$\boldsymbol{w}_i^o = \begin{bmatrix} w_{i1}^o & w_{i2}^o & \cdots & w_{ih}^o \end{bmatrix}^T , \ \ i = 1, 2, \cdots, n \tag{2-22}$$

and $\boldsymbol{b}^o$ is the vector of bias elements for the output neurons,

$$\boldsymbol{b}^o = \begin{bmatrix} b_1^o & b_2^o & \cdots & b_n^o \end{bmatrix}^T \tag{2-23}$$

To provide an equivalent scaling of the output signals to that one used in the input,

$$x_{ci} = x_{fi\,\min} + \frac{1}{2}(y_i + 1)\,(x_{fi\,\max} - x_{fi\,\min}) \ , \ \ i = 1, 2, \cdots, n \tag{2-24}$$

The vector $\boldsymbol{w}$ containing the total set of internal parameters of the ANN taken as optimization variables for a three layer perceptron is then defined as

$$\boldsymbol{w} = [(\boldsymbol{b}^h)^T \quad (\boldsymbol{b}^o)^T \quad (\boldsymbol{w}_1^h)^T \quad \ldots \quad (\boldsymbol{w}_h^h)^T \quad (\boldsymbol{w}_1^o)^T \quad \ldots \quad (\boldsymbol{w}_n^o)^T\,]^T \tag{2-25}$$

From (2-19), (2-20), (2-22) and (2-23) it is seen that the number of optimization variables involved in solving (2-3) following a SMN technique is $n(2h+1)+h$.

As stated in Section 2.4.2, we use a unit mapping as the starting point for training the ANN during SM-based neuromodeling. Since the same kind of scaling is being applied at the input and at the output of the ANN, as stated in (2-17) and (2-24), then $\boldsymbol{x}_c \approx \boldsymbol{x}_f$ implies that $y_i \approx v_i$ for $i = 1,2,\ldots,n$ in the region of interest of the physical parameters $\boldsymbol{x}_f$. To decouple the hidden neurons we can choose the initial hidden bias and hidden weighting factors as

$$\begin{bmatrix} (\boldsymbol{w}_1^h)^T \\ \vdots \\ (\boldsymbol{w}_h^h)^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{D} \\ \mathbf{0} \end{bmatrix} \begin{matrix} {\scriptstyle n\times n} \\ {\scriptstyle (h-n)\times n} \end{matrix} \quad \text{and } \boldsymbol{b}^h = \mathbf{0} \tag{2-26}$$

where $\boldsymbol{D}$ is an n by n diagonal matrix with elements $w_{ii}{}^h$, hence

$$z_i = \varphi(b_i^h + \boldsymbol{v}^T \boldsymbol{w}_i^h) = \varphi(w_{ii}{}^h\, v_i) \;\text{ for }\; i = 1, 2, \cdots, n \;\text{ and } z_i = 0 \text{ for }\; i = n+1,\; \cdots, h \tag{2-27}$$

To decouple the output neurons we can define the output weighting factors as

$$\begin{bmatrix} (\boldsymbol{w}_1^o)^T \\ \vdots \\ (\boldsymbol{w}_n^o)^T \end{bmatrix} = \begin{bmatrix} w_{11}^o & 0 & \cdots & & 0 & \cdots \\ 0 & w_{22}^o & 0 & \cdots & 0 & \cdots \\ \vdots & \vdots & \ddots & & 0 & \cdots \\ 0 & 0 & 0 & w_{nn}^o & 0 & \cdots \end{bmatrix}_{n\times h} \tag{2-28}$$

so that

$$y_i = b_i^o + \boldsymbol{z}^T \boldsymbol{w}_i^o = b_i^o + w_{ii}^o\, \varphi(w_{ii}{}^h\, v_i) \;,\quad i = 1, 2, \cdots, n \tag{2-29}$$

2.5.1 Sigmoid Activation Function

If a sigmoid or logistic function is used, then the response of *k*th hidden neuron is given by (see Haykin, 1999)

$$z_k = \varphi_k\,(s_k) = \frac{1}{1+e^{-s_k}}\,,\quad k = 1, 2, \cdots, h \tag{2-30}$$

which is approximated by using

$$z_k = \varphi_k\,(s_k) = \frac{1}{1+e^{-s_k}} \approx \frac{1}{2}\left(\frac{1}{2}s_k + 1\right) \;\text{ for }\; |\, s_k \,| << 1 \tag{2-31}$$

Since $-1 \le v_i \le 1$ due to the input scaling, if we take $w_{ii}{}^h = 0.1$ for $i = 1, 2, \cdots, n$ in (2-29) then (2-30) can be approximated using (2-31) by

$$y_i = b_i^o + \frac{w_{ii}^o}{2}\left[\frac{1}{20}\ v_i + 1\right] \ , \quad i = 1, 2, \cdots, n \tag{2-32}$$

It is seen that by taking

$$w_{ii}^o = 40 \ \text{ and } \ b_i^o = -20 \ \text{ for } \ i = 1, 2, \cdots, n \tag{2-33}$$

we achieve the desired unit neuromapping. Therefore, a starting point for the internal parameter of the ANN when sigmoid activation functions are used is given by (2-26) with $w_{ii}{}^{h} = 0.1$, (2-28) and (2-33).

2.5.2 Hyperbolic Tangent Activation Function

If a hyperbolic tangent is used, the response of the *k*th hidden neuron is

$$z_k = \varphi_k\,(s_k) = \tanh\,(s_k) = \frac{e^{s_k} - e^{-s_k}}{e^{s_k} + e^{-s_k}} \tag{2-34}$$

A linear approximation to (2-34) can be obtained from its Taylor series expansion

$$z_k = \varphi_k\,(s_k) = \tanh\,(s_k) \approx s_k - \frac{s_k{}^3}{3} + \frac{2s_k{}^5}{15} - \frac{17s_k{}^7}{315} + \cdots \approx s_k \quad \text{for } |s_k| \text{ small} \tag{2-35}$$

Since $-1 \le v_i \le 1$ due to the input scaling, if we take $w_{ii}{}^{h} = 0.1$ for $i = 1, 2, \cdots, n$ in (2-29) then (2-34) can be approximated using (2-35) by

$$y_i = b_i^o + \frac{w_{ii}^o\ v_i}{10} \ , \quad i = 1, 2, \cdots, n \tag{2-36}$$

It is seen that by taking

$$w_{ii}^o = 10 \ \text{ and } \ b_i^o = 0 \ \text{ for } \ i = 1, 2, \cdots, n \tag{2-37}$$

we achieve the desired unit neuromapping. Therefore, a starting point for the internal parameters of the ANN when hyperbolic tangent activation functions are used is given by (2-26) with $w_{ii}^{h} = 0.1$, (2-28) and (2-37).

In the examples described below, we considered sigmoid functions as well as hyperbolic tangent functions to implement the nonlinear activation functions for the neurons in the hidden layer. SM based neuromodels using 3-layer perceptrons have been realized in a variety of ways by Bandler, Ismail, Rayas-Sánchez and Zhang (1999a-c) as well as by Bandler, Rayas-Sánchez and Zhang (1999a-b).

2.6 CASE STUDIES

2.6.1 Microstrip Right Angle Bend

Consider a microstrip right angle bend, as illustrated in Fig. 2.13, with the following input parameters: conductor width *W*, substrate height *H*, substrate dielectric constant ε_r, and operating frequency ω. Three neuromodels exploiting SM technology are developed for the region of interest shown in Table 2.1.

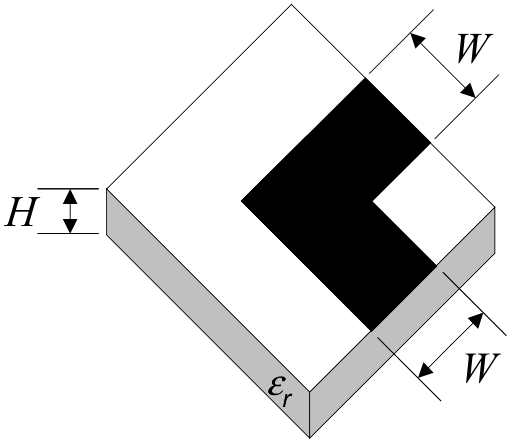


Fig. 2.13 Microstrip right angle bend.

TABLE 2.1
REGION OF INTEREST FOR THE
MICROSTRIP RIGHT ANGLE BEND

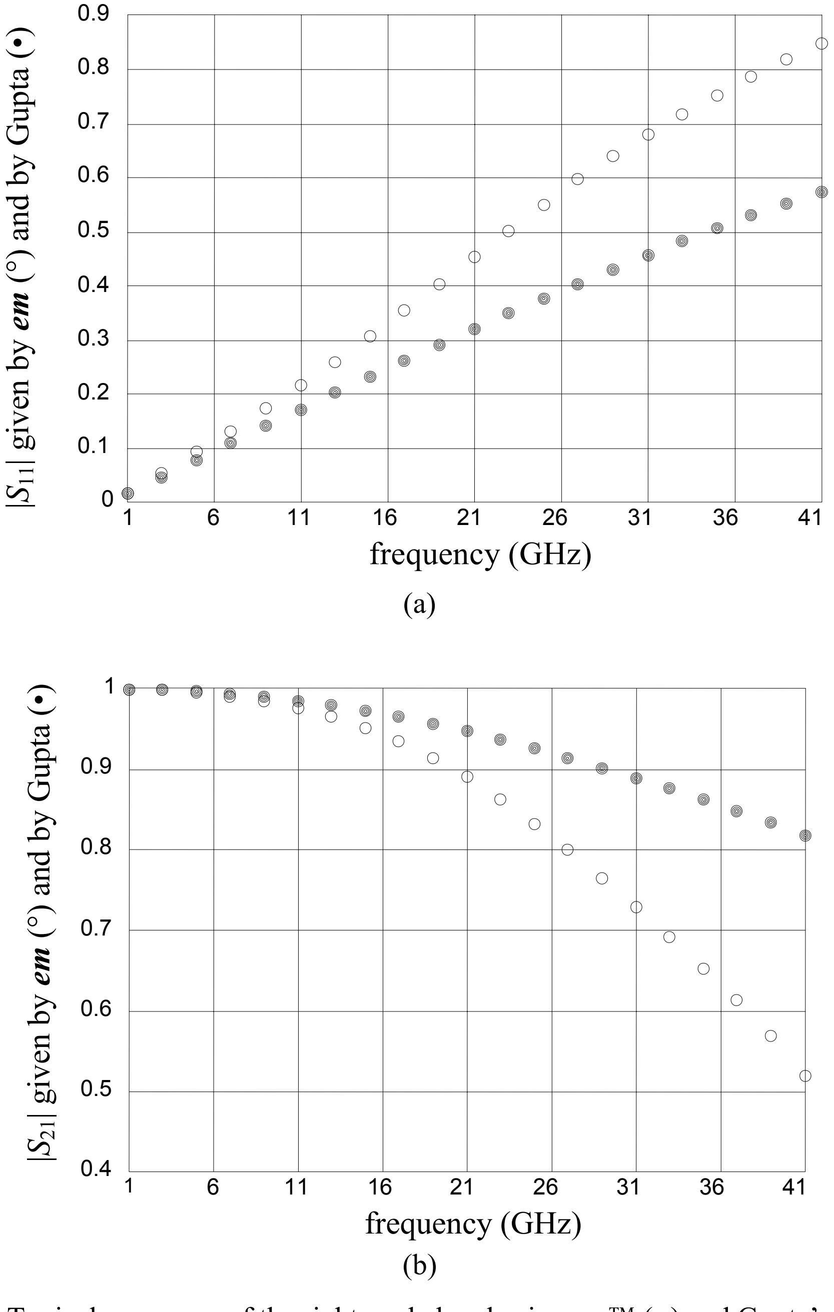
Parameter	Minimum value	Maximum value
W	20 mil	30 mil
H	8 mil	16 mil
ε_r	8	10
ω	1 GHz	41 GHz

The model proposed by Gupta, Garg and Bahl (1979), consisting of a lumped LC circuit whose parameter values are given by analytical functions of the physical quantities W, H and ε_r is taken as the "coarse" model and implemented in OSA90/hope™ (1997). Sonnet's ***em***™ (1997) is used as the fine model. To parameterize the structure, the Geometry Capture technique proposed by Bandler, Biernacki and Chen (1996) available in Empipe™ (1997) is utilized.

Fig. 2.14 shows typical responses of the coarse and fine models before any neuromodeling, using a frequency step of 2 GHz (F_p = 21). The coarse and fine models are compared in Fig. 2.15 using 50 random test base-points with uniform statistical distribution within the region of interest (1050 test samples). Gupta's model, in this region of physical parameters, yields acceptable results for frequencies less than 10 GHz.

With a star set for the learning base points (n = 3, B_p = 7), 147 learning samples (l = 147) are used for three SM based neuromodels, and the corresponding ANNs were implemented and trained within OSA90/hope™. Huber optimization was employed as the training algorithm, exploiting its robust characteristics for data fitting as shown by Bandler, Chen, Biernacki, Gao, Madsen and Yu (1993).

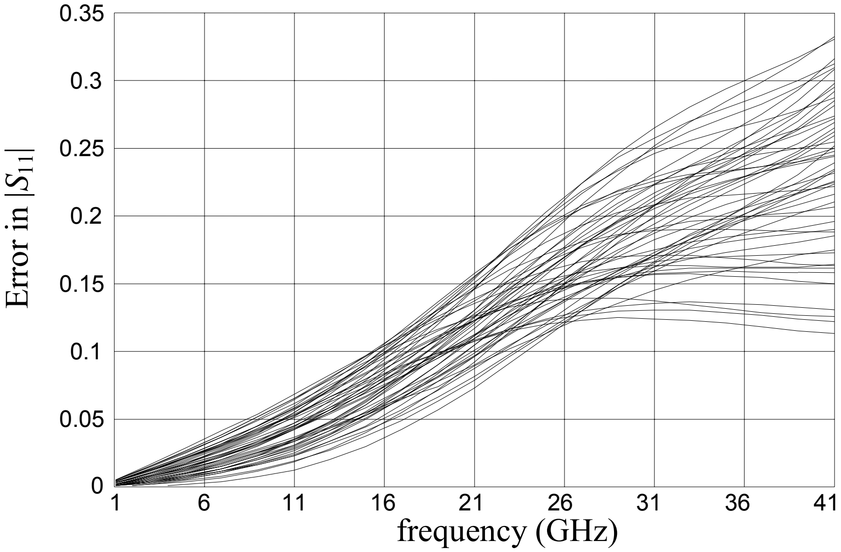
Fig. 2.16 shows the results for the SMN model implemented with a 3-layer perceptron with 3 input, 6 hidden and 3 output neurons (3LP:3-6-3). A FDSMN model is developed using a 3LP:4-7-3, and the improved results are shown in Fig. 2.17.



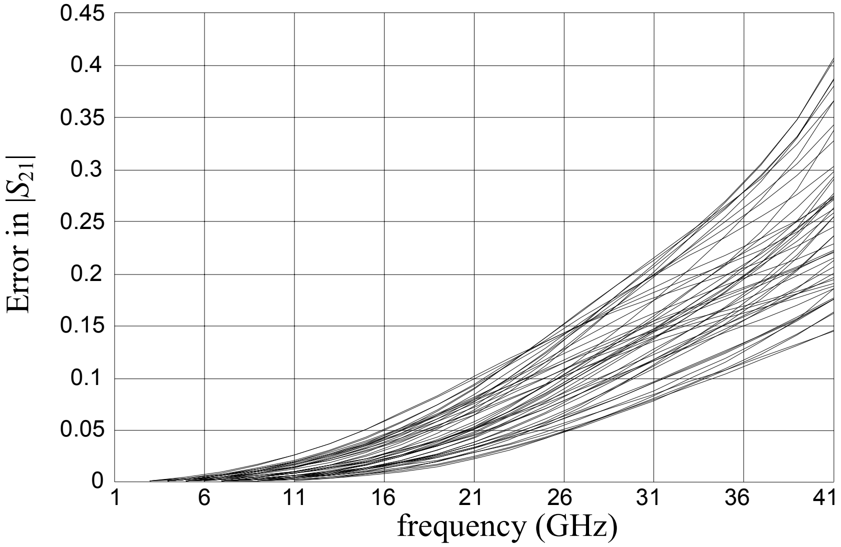
(a)

(b)

Fig. 2.14 Typical responses of the right angle bend using ***em***™ (○) and Gupta's model (•) before any neuromodeling: (a) $|S_{11}|$, (b) $|S_{21}|$.



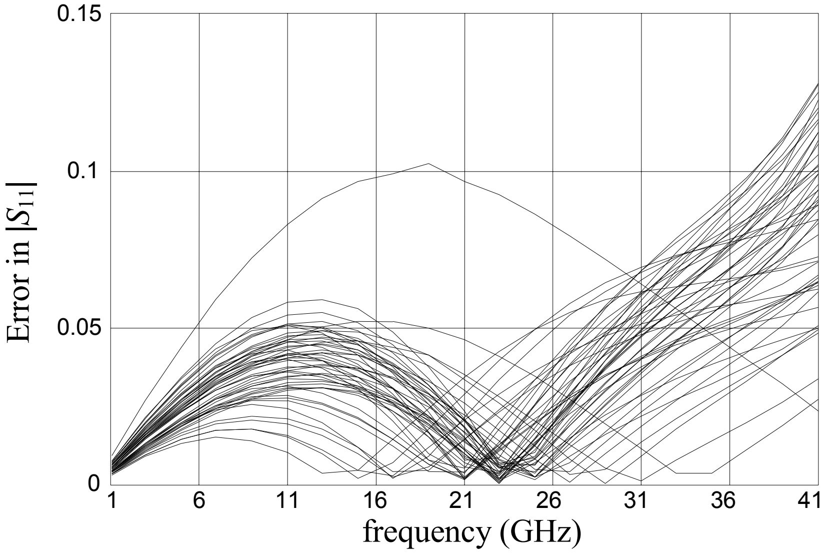
(a)



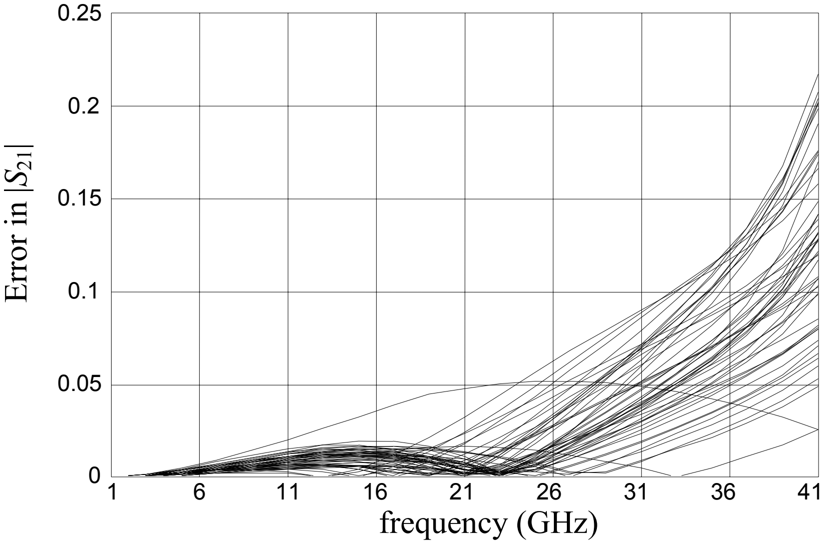
(b)

Fig. 2.15 Comparison between ***em***™ and Gupta model of a right angle bend: (a) error in $|S_{11}|$ with respect to ***em***™, (b) error in $|S_{21}|$ with respect to ***em***™.

In Fig. 2.18 the results for the FSMN model with a 3LP:4-8-4 are shown, which are even better (as expected). To implement the FSMN approach, an OSA90 child



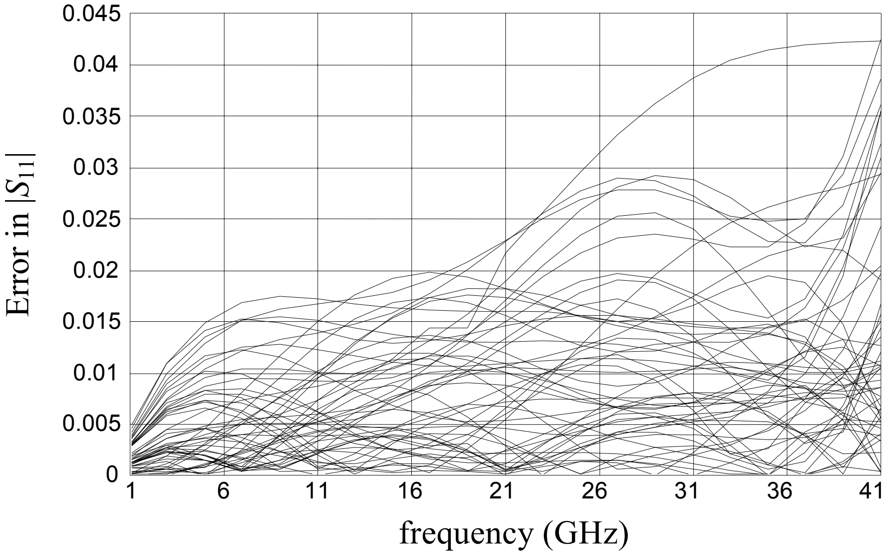
(a)



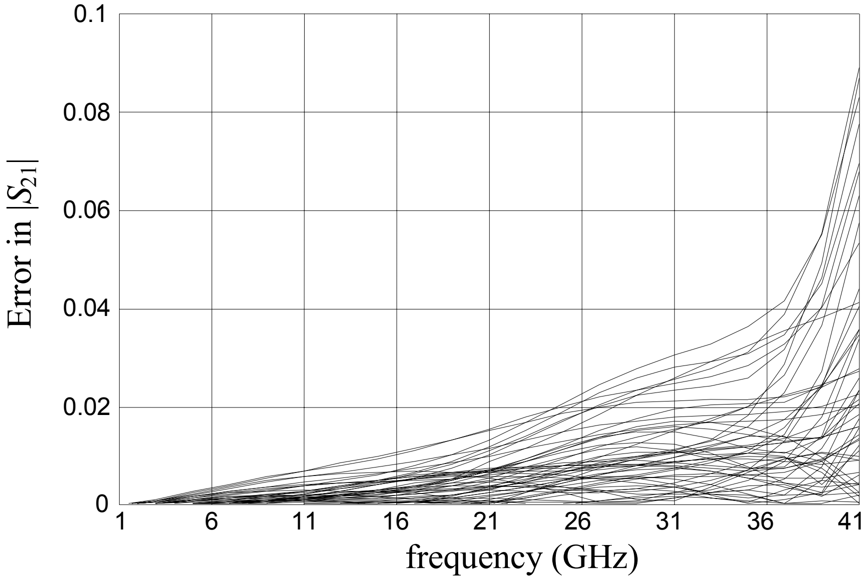
(b)

Fig. 2.16 Comparison between ***em***™ and SMN model of a right angle bend: (a) error in $|S_{11}|$ with respect to ***em***™, (b) error in $|S_{21}|$ with respect to ***em***™.

program is employed to simulate the coarse model with a different frequency variable using Datapipe.

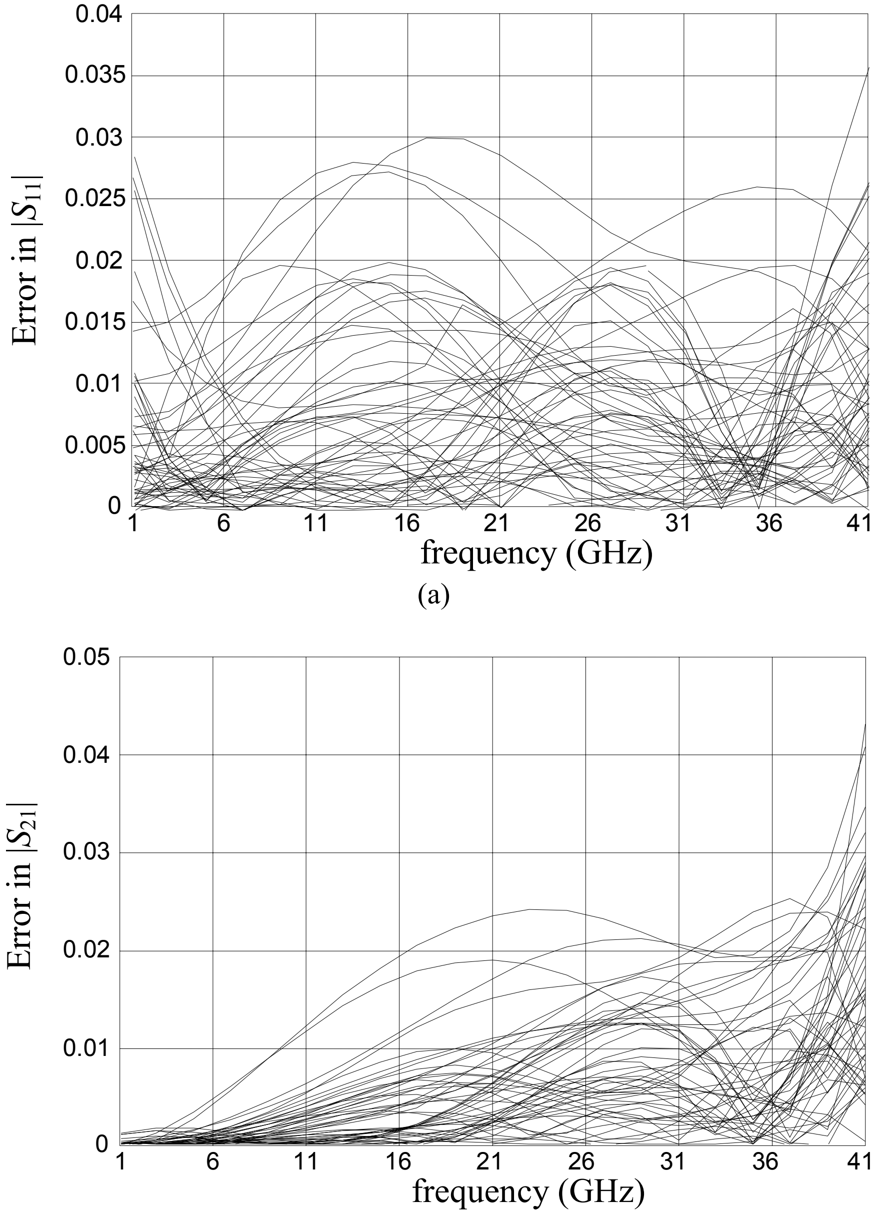


(a)



(b)

Fig. 2.17 Comparison between ***em***™ and FDSMN model of a right angle bend: (a) error in $|S_{11}|$ with respect to ***em***™, (b) error in $|S_{21}|$ with respect to ***em***™.



(a)

(b)

Fig. 2.18 Comparison between ***em***™ and FSMN model of a right angle bend: (a) error in $|S_{11}|$ with respect to ***em***™, (b) error in $|S_{21}|$ with respect to ***em***™.

It is seen in Fig. 2.18 that the FSMN model yields excellent results for the whole frequency range of interest, overcoming the frequency limitations of the empirical model by a factor of four.

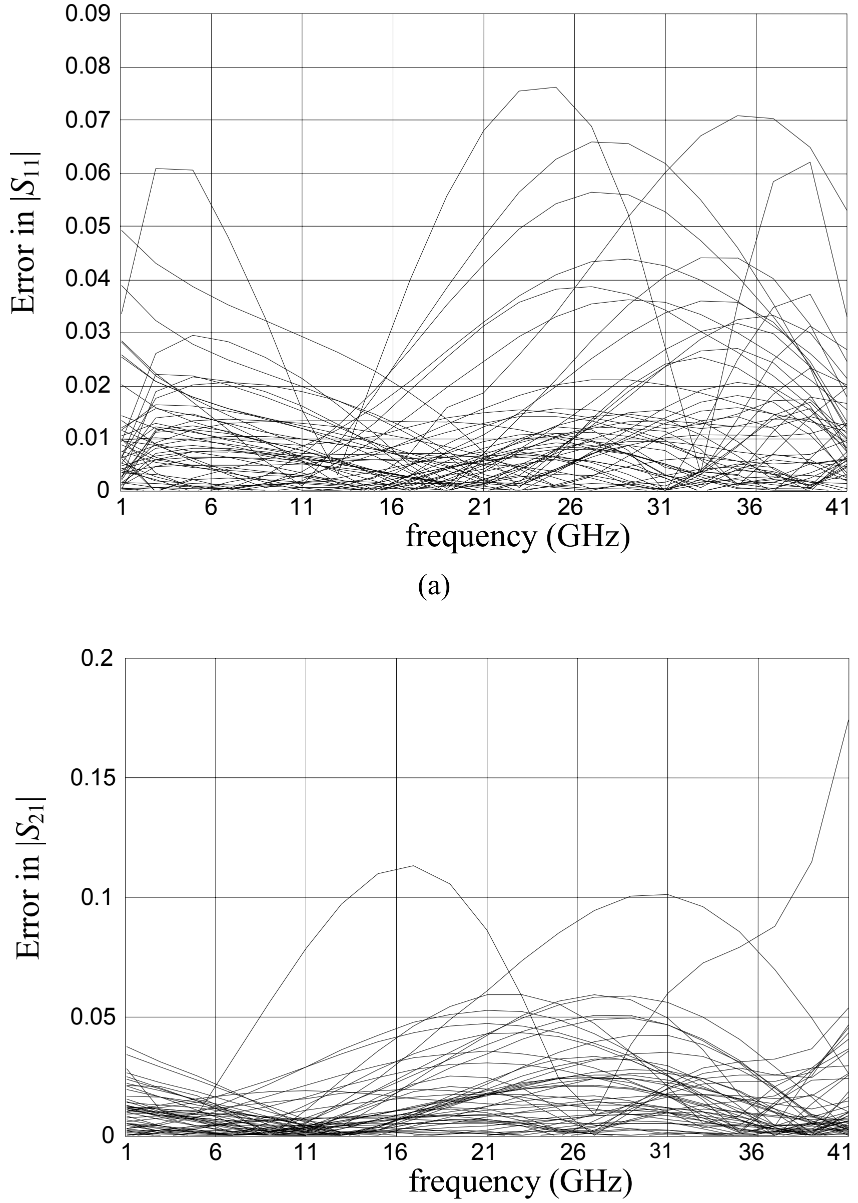
To compare these results with those from a classical neuromodeling approach, an ANN was developed using *NeuroModeler*™ (1998). Training the ANN with the same 147 learning samples, the best results were obtained for a 3LP:4-15-4 trained with the conjugate gradient and quasi-Newton methods. Due to the small number of learning samples, this approach did not provide good generalization capabilities, as illustrated in Fig. 2.19. To produce similar results to those in Fig. 2.18 using the same ANN complexity, the learning samples have to increase from 147 to 315.

Fig. 2.20 summarizes the different neuromodeling approaches applied to this case study.

2.6.2 HTS Quarter-Wave Microstrip Filter

Fig. 2.21 illustrates the physical structure of a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter, to be modeled in the region of interest shown in Table 2.2. L_1, L_2 and L_3 are the lengths of the parallel coupled-line sections and S_1, S_2 and S_3 are the gaps between the sections. The width W is the same for all the sections as well as for the input and output microstrip lines, of length L_0. A lanthanum aluminate substrate with thickness H and dielectric constant ε_r is used. The metalization is considered lossless. Two space mapping based neuromodels are developed in the region of interest defined by Table 2.2, taking as design parameters $\boldsymbol{x}_f = [L_1\ L_2\ L_3\ S_1\ S_2\ S_3]^T$.

It has been already shown in the work by Bandler, Biernacki, Chen, Getsinger, Grobelny, Moskowitz and Talisa (1995) that the responses of this narrow bandwidth filter



(a)

(b)

Fig. 2.19 Comparison between ***em***™ and classical neuromodel of a right angle bend: (a) error in $|S_{11}|$ with respect to ***em***™, (b) error in $|S_{21}|$ with respect to ***em***™.

are very sensitive to dimensional changes.

Sonnet's ***em***™ (1997) driven by Empipe™ (1997) was employed as the fine

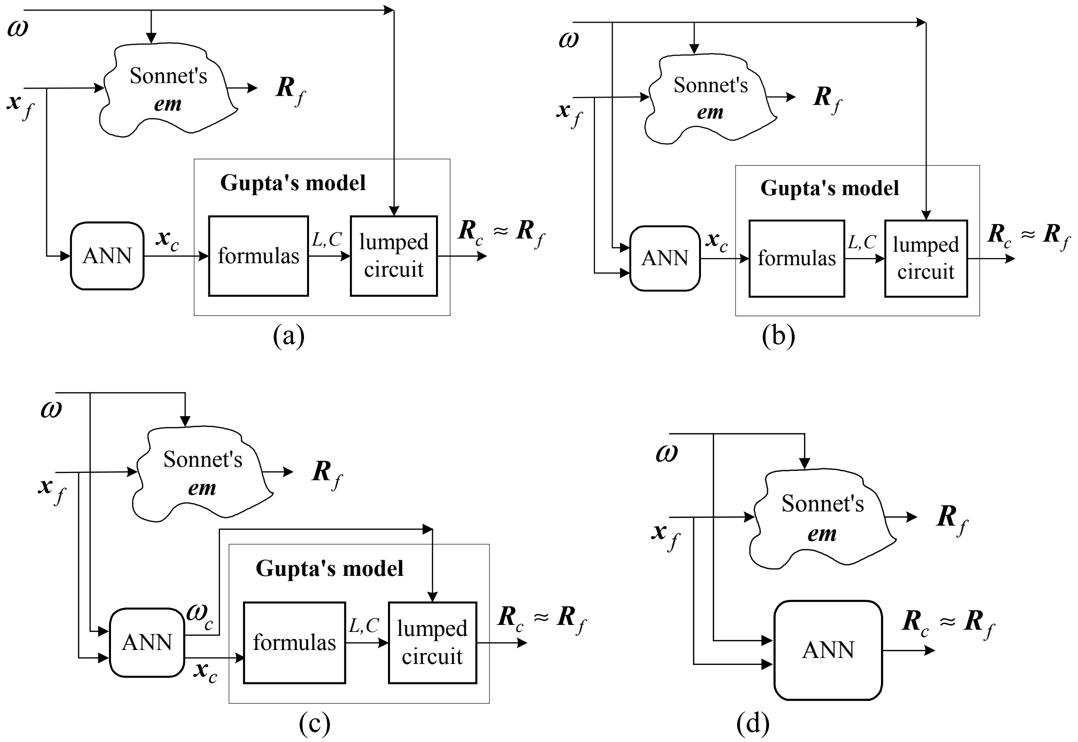


Fig. 2.20 Different neuromodeling approaches for the right angle bend: (a) SMN, (b) FDSMN, (c) FSMN, and (d) classical neuromodeling.

model, using a high-resolution grid with a 1mil×1mil cell size.

Sections of OSA90/hope™ built-in linear elements MSL (microstrip line) and MSCL (two-conductor symmetrical coupled microstrip lines) connected by circuit theory over the same MSUB (microstrip substrate definition) are taken as the “coarse” model.

Typical responses of the coarse and fine models before any neuromodeling are shown in Fig. 2.22, using a frequency step of 0.02 GHz (F_p = 14). About 10 hrs of CPU simulation time was needed for a single frequency sweep on an HP C200-RISC workstation. Following a multidimensional star set (n = 6), 13 learning base points are used (l = 182). To evaluate the generalization performance, 7 testing base points not seen in the learning set are used.

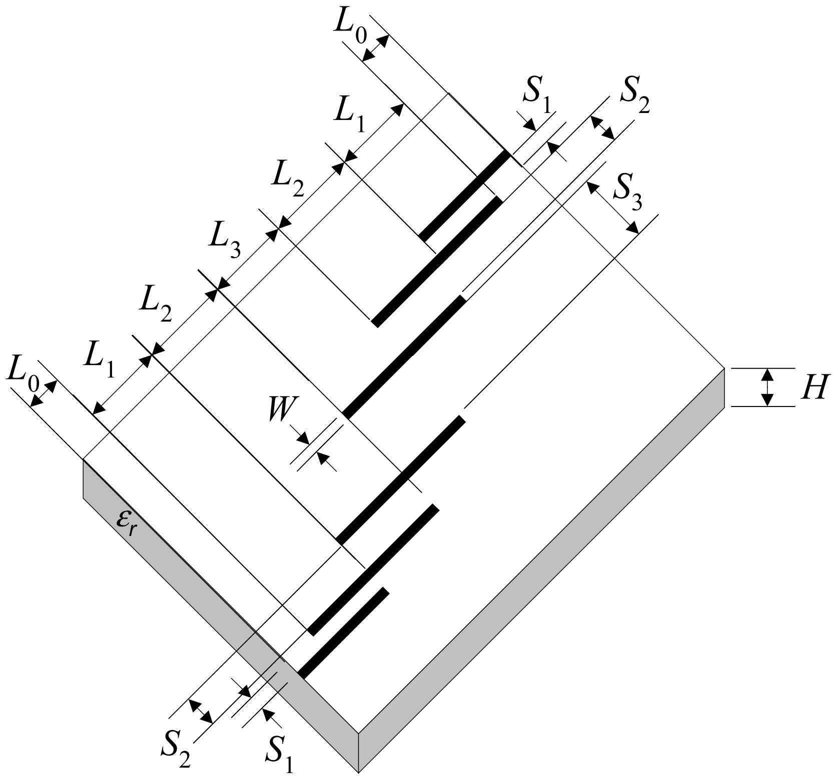


Fig. 2.21 HTS quarter-wave parallel coupled-line microstrip filter.

TABLE 2.2
REGION OF INTEREST FOR THE HTS FILTER

Parameter	Minimum value	Maximum value
W	7 mil	7 mil
H	20 mil	20 mil
ε_r	23.425	23.425
Loss tang	3×10^{-5}	3×10^{-5}
L_0	50 mil	50 mil
L_1	175 mil	185 mil
L_2	190 mil	210 mil
L_3	175 mil	185 mil
S_1	18 mil	22 mil
S_2	75 mil	85 mil
S_3	70 mil	90 mil
ω	3.901 GHz	4.161 GHz

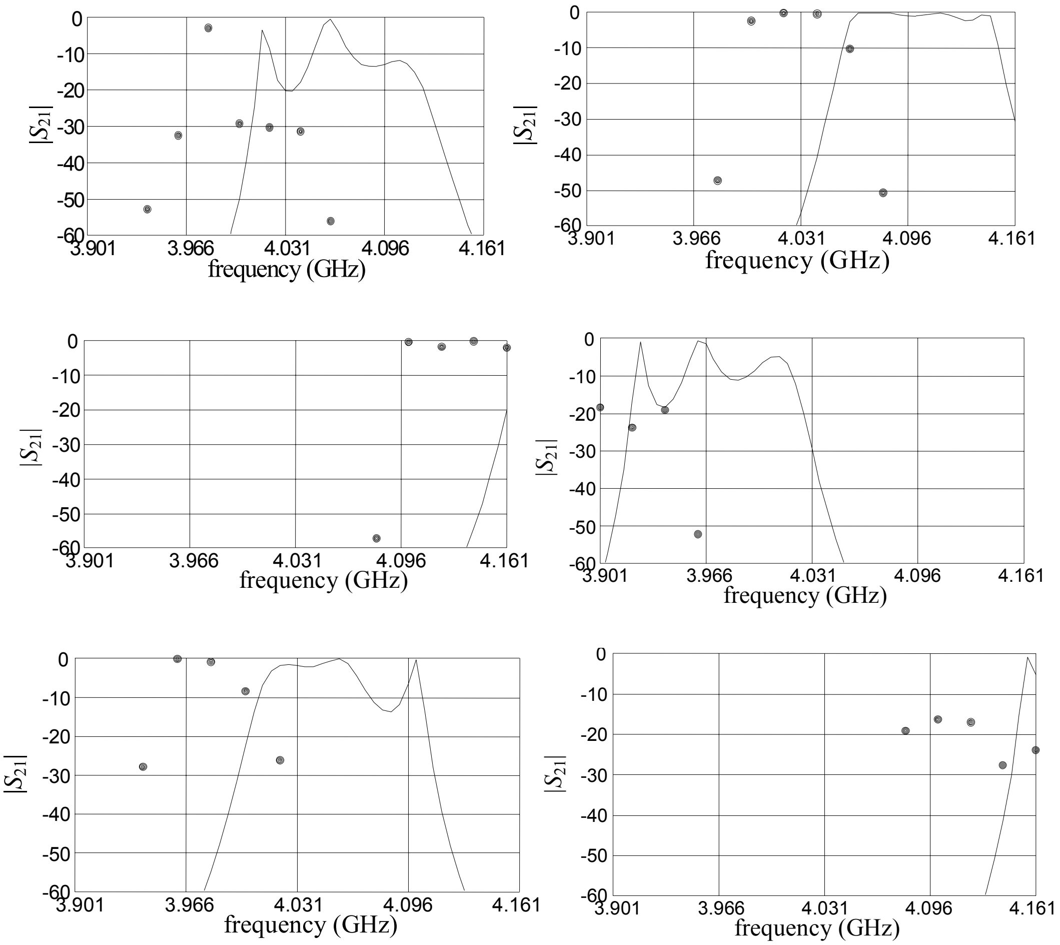


Fig. 2.22 Typical responses of the HTS filter using ***em***™ (•) and OSA90/hope™ model (–) before any neuromodeling at three learning and three testing points.

The coarse and fine models responses before any neuromodeling are compared in Fig. 2.23, at both the learning and the testing sets, showing very large errors in the empirical model with respect ***em***™ due to the shifting in its frequency response, as seen in Fig. 2.22.

To explore the effects of simulating the coarse model at a mapped frequency, a FMN technique (see Fig. 2.9) is implemented with a 3LP:7-5-1 trained with Huber optimization. As shown in Fig. 2.24, the FMN approach yields good frequency alignment between both responses, although a significant error in the amplitudes remains.

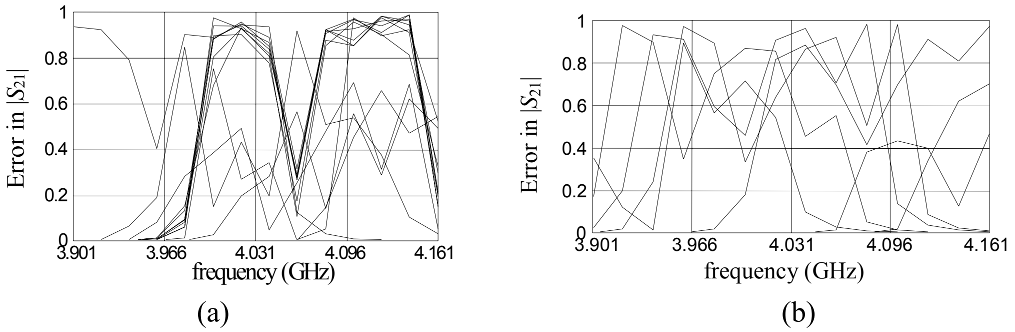


Fig. 2.23 Coarse model error w.r.t. ***em***™ before any neuromodeling: (a) in the learning set, (b) in the testing set.

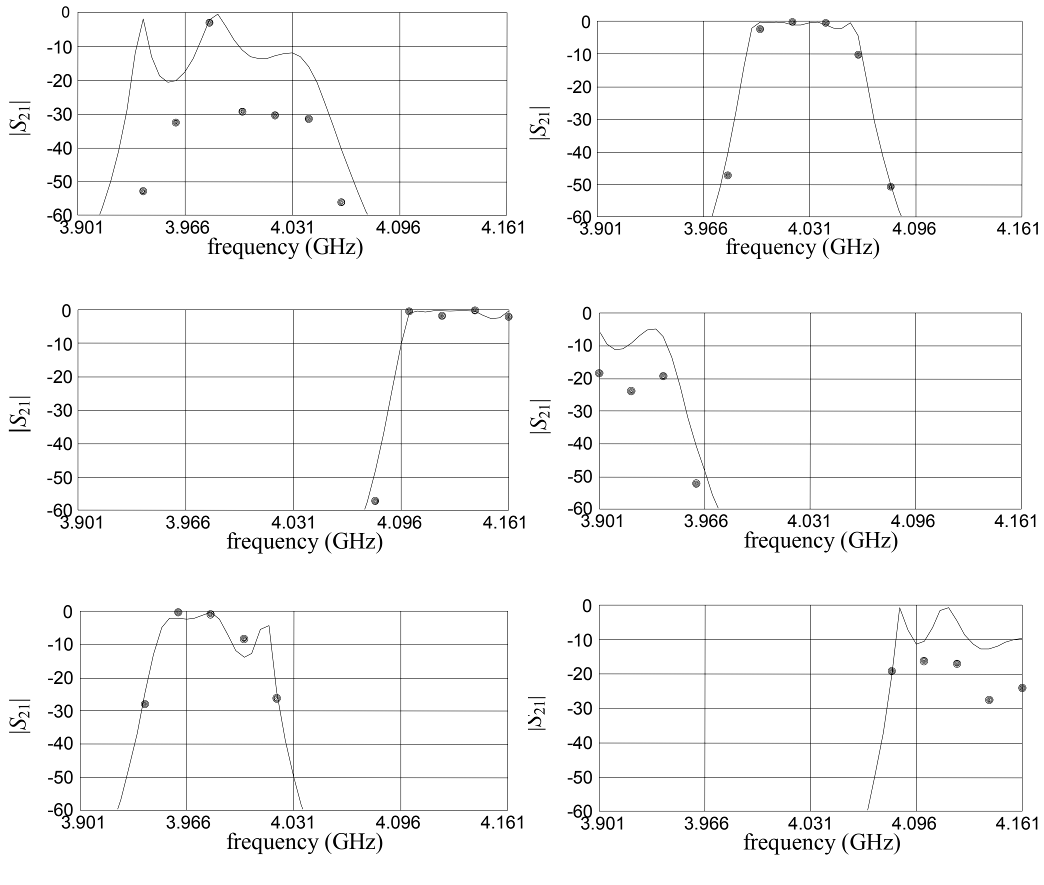


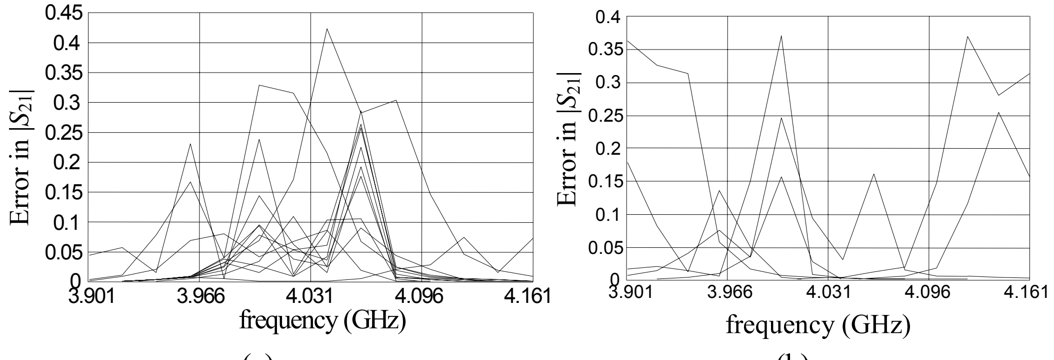
Fig. 2.24 Typical responses of the HTS filter using ***em***™ (•) and FMN model (−) at the same three learning and three testing points as in Fig. 2.22.

The complete training and generalization errors for the FMN model are shown in Fig. 2.25. Comparing Fig. 2.23 with Fig. 2.25 it is seen that a significant reduction in the error is achieved by mapping only the frequency.

Excellent results are obtained for the FPSMN modeling approach (see Fig. 2.10). In this case we mapped only L1 and S1, giving the rest of the design paramaters to the coarse model without any transformation, i.e., we take $\boldsymbol{x}^{\bullet}_{c} = [L_{1\mathrm{c}}\ S_{1\mathrm{c}}]^{T}$ and $\boldsymbol{x}^{\bullet}_{f} = [L_2\ L_3\ S_2\ S_3]^{T}$. Optimal generalization performance was achieved with a 3LP:7-7-3, trained with Huber optimization.

As illustrated in Fig. 2.26, where the same learning and testing points used in Fig. 2.22 and Fig. 2.24 were chosen, an outstanding agreement between the fine model and the FPSMN model is achieved. The complete learning and generalization performance for the FPSMN is shown in Fig. 2.27.

As a final test, both the FPSMN model and the fine model are simulated at three different base points using a very fine frequency sweep, with a frequency step of 0.005GHz. Remarkable matching is obtained, as illustrated in Fig. 2.28.



(a) (b)

Fig. 2.25 FMN model error w.r.t. ***em***™: (a) in the learning set, (b) in the testing set.

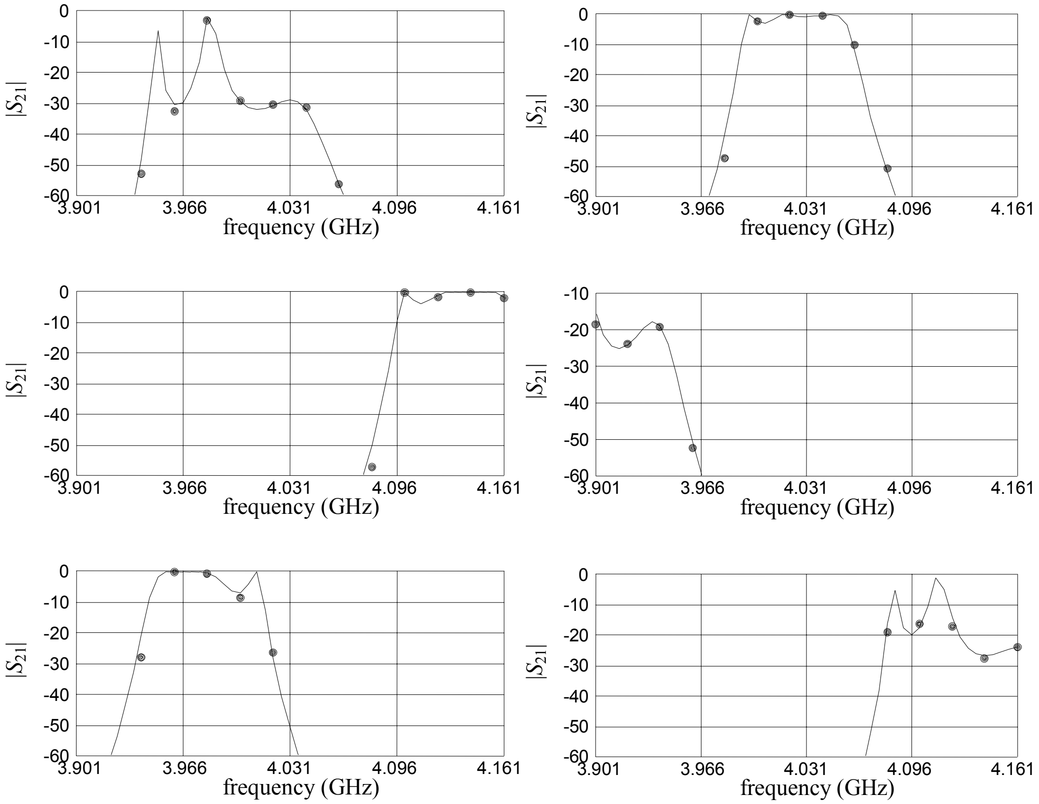
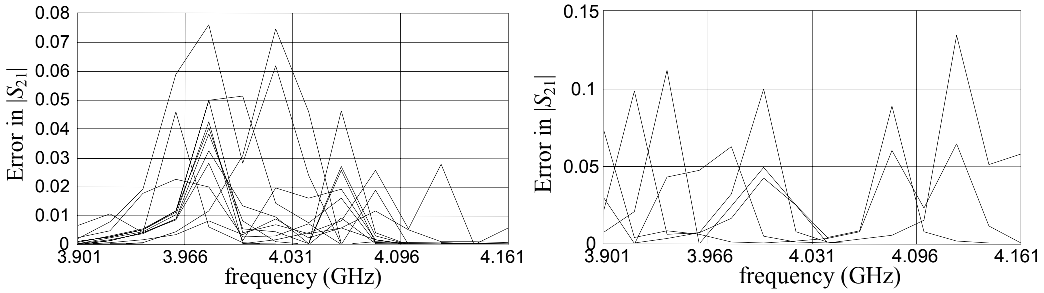


Fig. 2.26 Typical responses of the HTS filter using ***em***™ (•) and FPSMN model (−) at the same three learning and three testing points as in Fig. 2.22.



(a) (b)

Fig. 2.27 FPSMN model error w.r.t. ***em***™: (a) in the learning set, (b) in the testing set.

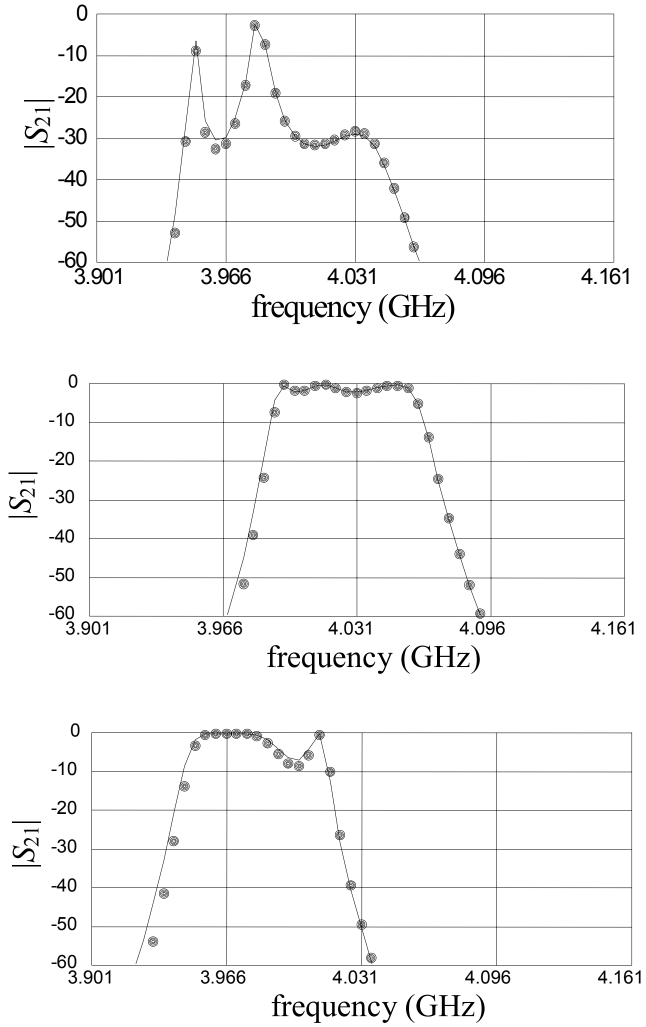


Fig. 2.28 Comparison between the HTS filter response using ***em***™ (•) and FPSMN model (–) at three base points using a fine frequency sweep.

2.7 RELATIONSHIP BETWEEN SM BASED NEUROMODELING AND GSM MODELING

A Generalized Space Mapping (GSM) approach to device modeling was developed by Bandler, Georgieva, Ismail, Rayas-Sánchez and Zhang (1999), in which a comprehensive tableau for a linear mapping applicable for both the design parameters as well as the frequency variable is formulated. GSM modeling is closely related to the SM-based neuromodeling.

As we showed in the previous sections of this chapter, our SM based neuromodeling approach is capable of establishing a nonlinear mapping for the design parameters only (SMN modeling), the design parameters with frequency dependence (FDSMN), the frequency only (FMN), or the design parameters and the frequency simultaneously (FSMN and FPSMN).

Due to the nonlinear nature of the neuromapping, the SM based neuromodeling techniques do not require the frequency range to be segmented in case of severe misalignment between the coarse and fine frequency responses, in contrast with the piecewise linear approach usually needed in these cases when the GSM techniques are applied.

Furthermore, in the FMN, FSMN and FPSMN techniques, a coupling between the transformed frequency ω_c and the design parameters $\boldsymbol{x}_f$ is in principle assumed, which represents the most general case in the GSM approach.

The nonlinear mapping used by the SM-based neuromodeling techniques allows us to cover a much larger region in the input design parameter space than that one typically covered by GSM techniques. When the GSM techniques are used, a multiple space mapping approach is usually needed if the region of interest is large or if the model responses are very sensitive to the design parameters in the region of interest.

The Space Mapping Super Model (SMSM) concept, the Frequency Space Mapping Super Model (SMSM) concept and the Multiple Space Mapping (MSM) concept are variations of the GSM approach to device modeling. These techniques are described in the work by Bandler, Georgieva, Ismail, Rayas-Sánchez and Zhang (1999), as well as in the work by Bandler, Ismail and Rayas-Sánchez (2000).

2.8 CONCLUDING REMARKS

We have described in this chapter novel applications of Space Mapping technology to the modeling of microwave circuits using artificial neural networks. Five powerful techniques to generate SM based neuromodels have been described and illustrated: Space Mapped Neuromodeling (SMN), Frequency-Dependent Space Mapped Neuromodeling (FDSMN), Frequency Space Mapped Neuromodeling (FSMN), Frequency Mapped Neuromodeling (FMN) and Frequency Partial-Space Mapped Neuromodeling (FPSMN).

The SM-based neuromodeling techniques exploit the vast set of empirical models already available, decrease the number of fine model evaluations needed for training, improve generalization performance and reduce the complexity of the ANN topology w.r.t. the classical neuromodeling approach.

Frequency-sensitive neuromapping is demonstrated to be a clever strategy to expand the usefulness of microwave empirical models that were developed using quasi-static analysis.

We have also demonstrated FMN as an effective technique to align severely frequency-shifted responses.

By establishing a partial mapping for the physical parameters, a more efficient use of the implicit knowledge in the coarse model is achieved and the corresponding neuromapping becomes simpler and easier to train.

As an original alternative to the classical backpropagation algorithm, Huber optimization is employed to efficiently train the neural network that implements the mapping, exploiting its robust characteristics for data fitting.

Chapter 3

NEURAL SPACE MAPPING (NSM) OPTIMIZATION

3.1 INTRODUCTION

As we showed in Chapter 2, Artificial Neural Networks (ANNs) are suitable models for microwave structures. Neuromodels are computationally much more efficient than EM or physical models and can be more accurate than empirical, physics-based models. Once they are trained with reliable learning data, obtained by either EM simulation or by measurement, the neuromodels can be used for efficient and accurate optimization within the region of training. This has been the conventional approach to optimization of microwave structures using ANNs; see for example the work of Watson and Gupta (1997).

The principal drawback of this ANN optimization approach is the cost of generating sufficient learning samples, since the simulations/measurements must be performed for many combinations of different values of geometrical, material, process and input signal parameters over a large region. Additionally, it is well known that the extrapolation ability of neuromodels is poor, making unreliable any solution predicted outside the training region. Introducing knowledge, as in the approach of Watson, Creech and Gupta (1999), can alleviate these limitations.

A powerful new method for optimization of microwave circuits based on Space

Mapping (SM) technology and Artificial Neural Networks (ANN) is described in this Chapter. An innovative strategy is proposed to exploit the SM-based neuromodeling techniques in an efficient Neural Space Mapping (NSM) optimization algorithm. SM-based neuromodeling techniques are described in the previous chapter and were developed by Bandler, Ismail, Rayas-Sánchez and Zhang (1999). In this chapter, we change our focus of interest, from modeling to design by optimization.

NSM optimization requires a reduced set of upfront fine model simulations or learning base points. A coarse or empirical model is used as source of knowledge that reduces the required amount of learning data and improves the generalization and extrapolation performance of the SM-based neuromodel. The sensitivity information of the coarse model is also employed as a means to select the initial learning base points.

A novel procedure that does not require troublesome parameter extraction to predict the next point is described.

As before, Huber optimization is used to train the SM-based neuromodels at each iteration.

In order to reduce the amount of fine model simulations, the SM-based neuromodels are developed without using testing points: their generalization performance is controlled by gradually increasing their complexity starting with a 3-layer perceptron with 0 hidden neurons, i.e., starting with a linear mapping.

NSM optimization is illustrated by the optimization of a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter and a bandstop microstrip filter with quarter-wave resonant open stubs. These results are compared in Chapter 5 with those obtained using a more advanced ANN-based space mapping optimization algorithm.

3.2 A BRIEF REVIEW ON OPTIMIZATION OF MICRO-WAVE CIRCUITS USING NEURAL NETWORKS

In Section 2.3 we made a review of the main neural network modeling techniques employed in the microwave arena. Burrascano and Mongiardo (1999) developed a more detailed review on that subject. It is clear that neural networks have been extensively used for modeling in many different variations.

In contrast, the use of neural networks for design by optimization is at an earlier stage. A few variations in the use of neural networks for optimization of microwave circuits have been reported.

The most widely used technique for neural optimization of microwave circuits consists of generating a neuromodel of the microwave circuit within a certain training region of the design parameters, and then applying conventional optimization to the neuromodel to find the optimal solution that yields the desired response. A neuromodel can be developed for the whole microwave circuit to be optimized, or in a decomposed fashion, where small neuromodels are developed for each individual component in the circuit, which are later connected by circuit theory. Full wave EM simulations are typically employed to generate the training data. The generalization ability of the neuromodel(s) is controlled during the training process by using validation data and testing data, also obtained from EM simulations. Examples of this neural optimization approach can be found in the work by Horng, Wang and Alexopoulos (1993), Zaabab, Zhang and Nakhla (1995), Veluswami, Nakhla and Zhang (1997), Watson and Gupta (1997), and Burrascano, Dionigi, Fancelli and Mongiardo (1998).

As stated before, the previous neural optimization approach has two main disadvantages: the time required to generate sufficient training, validation and testing samples, and the unreliability of the "optimal" solution when it lies outside the training region.

One way to decrease the amount of up-front EM simulations is shown in the work by Burrascano and Mongiardo (1999), where the neuromodel to be optimized consists of several neural networks, each of them specialized for a cluster of responses that were previously identified.

Both limitations of the conventional neural optimization approach can be alleviated by incorporating prior knowledge into the neural network structure. In the work by Watson, Creech and Gupta (1999), an EM-ANN approach (see Section 2.3) was used to optimize a CPW patch antenna. Similarly, an end-coupled band-pass filter in a 2-layer configuration was designed by Cho and Gupta (1999) following also an EM-ANN approach.

A fourth variation for the design of microwave circuits with ANNs is by using synthesis neural networks. A synthesis neural network is trained to learn the mapping from the responses to the design parameters of the microwave circuit. In this sense, a conventional neuromodel becomes an analysis neural network. The problem of training a synthesis neural network is known as the inverse modeling problem, since the input and output variables are interchanged.

The analysis problem is characterized by a single-value mapping: given a vector of design parameters we have only one possible vector of responses. However, for inverse problems, the mapping can often be multivalued: a given vector of responses can be generated by several different vectors of design parameters. This leads the synthesis neural network to make poor generalizations. Another complication of the inverse modeling problem is the coverage of the input space by the training data, since the full characterization of the input space (microwave circuit responses) is usually not available.

Watson, Cho and Gupta (1999) successfully developed a dedicated algorithm for the design of multilayer asymmetric coupled transmission structures using a combination of analysis and synthesis neural networks. In this work, the input space of the synthesis neural network is not the set of S parameters, but a set of LC parameters that are later translated into the conventional responses.

The neural optimization technique described in this chapter makes use of the knowledge available in equivalent circuit models following a space mapping approach.

3.3 THE SPACE MAPPING CONCEPT WITH FREQUENCY INCLUDED

As indicated in Section 2.2, Space Mapping (SM) is a powerful concept for circuit design and optimization that combines the computational efficiency of “coarse” models with the accuracy of “fine” models. The SM concept can be extended to consider not only a mapping between the physical design parameters, but also between other independent variables. Frequency Space Mapping (FSM) was originally proposed by Bandler, Biernacki, Chen, Hemmers and Madsen (1995) as an strategy to improve the parameter extraction process when the shapes of two responses are similar but severely misaligned. FSM was originally employed to align those kinds of responses along the frequency axis first.

In the Space Mapping technique with frequency included, the operating frequency ω is also considered in the mapping function. This allows us to simulate the coarse model at a different frequency ω_c.

Let the vectors $\boldsymbol{x}_c$ and $\boldsymbol{x}_f$ represent the design parameters of the coarse and fine models, respectively, and $\boldsymbol{R}_c(\boldsymbol{x}_c, \omega_c)$ and $\boldsymbol{R}_f(\boldsymbol{x}_f, \omega)$ the corresponding model responses (for example, $\boldsymbol{R}_c$ and $\boldsymbol{R}_f$ might contain the real and imaginary parts of S_{21}). As before, $\boldsymbol{R}_c$ is much faster to calculate but less accurate than $\boldsymbol{R}_f$.

The aim of Space Mapping optimization, including frequency, is to find an appropriate mapping $\boldsymbol{P}$ from the fine model input space to the coarse model input space

$$\begin{bmatrix} \boldsymbol{x}_c \\ \omega_c \end{bmatrix} = \boldsymbol{P}(\boldsymbol{x}_f, \omega) \tag{3-1}$$

such that

$$\boldsymbol{R}_c(\boldsymbol{x}_c, \omega_c) \approx \boldsymbol{R}_f(\boldsymbol{x}_f, \omega) \tag{3-2}$$

Once a mapping $\boldsymbol{P}$ valid in the region of interest for the design parameters $\boldsymbol{x}_f$ and operating frequency ω is found, the coarse model can be used for fast and accurate simulations in that region.

3.4 NSM OPTIMIZATION: AN OVERVIEW

Fig. 3.1 shows the flow diagram of NSM optimization. Here we explain the overall operation of NSM optimization; a detailed description of the main blocks is presented in the following sections.

We start by finding the optimal coarse model solution $\boldsymbol{x}_c^*$ that yields the desired response by applying conventional optimization to the coarse model. We then select $2n$ additional points following an n-dimensional star set, as in the work by Biernacki, Bandler, Song and Zhang (1989), centered at $\boldsymbol{x}_c^*$, as illustrated in Fig. 3.2, where n is the number of design parameters ($\boldsymbol{x}_c, \boldsymbol{x}_f \in \Re^n$).

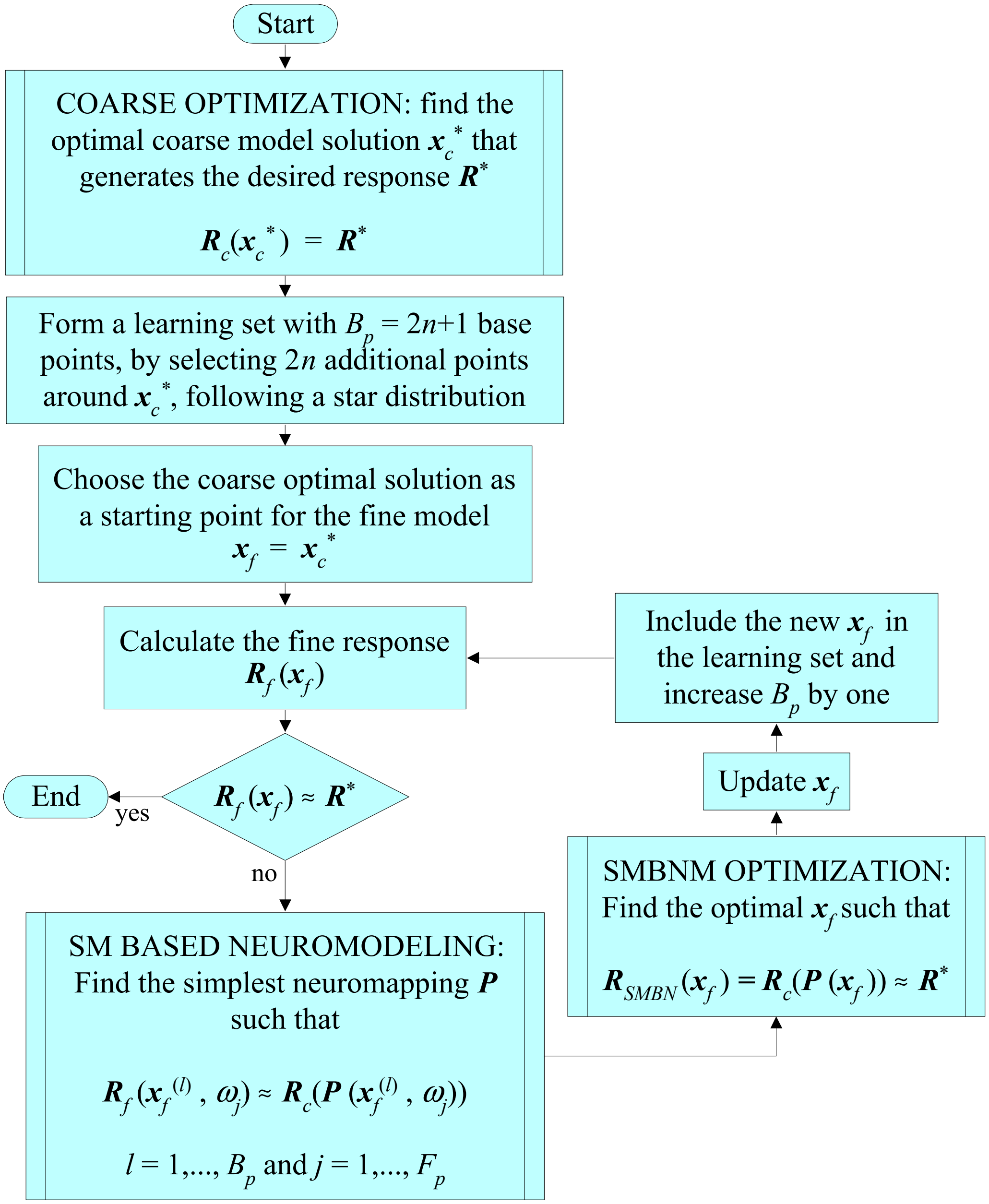

Start
COARSE OPTIMIZATION: find the optimal coarse model solution $\boldsymbol{x}_c^*$ that generates the desired response $\boldsymbol{R}^*$
$\boldsymbol{R}_c(\boldsymbol{x}_c^*) = \boldsymbol{R}^*$
Form a learning set with $B_p = 2n+1$ base points, by selecting $2n$ additional points around $\boldsymbol{x}_c^*$, following a star distribution
Choose the coarse optimal solution as a starting point for the fine model
$\boldsymbol{x}_f = \boldsymbol{x}_c^*$
Calculate the fine response $\boldsymbol{R}_f(\boldsymbol{x}_f)$
Include the new $\boldsymbol{x}_f$ in the learning set and increase B_p by one
$\boldsymbol{R}_f(\boldsymbol{x}_f) \approx \boldsymbol{R}^*$
End
yes
no
Update $\boldsymbol{x}_f$
SMBNM OPTIMIZATION: Find the optimal $\boldsymbol{x}_f$ such that
$\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f) = \boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f)) \approx \boldsymbol{R}^*$
SM BASED NEUROMODELING: Find the simplest neuromapping $\boldsymbol{P}$ such that
$\boldsymbol{R}_f(\boldsymbol{x}_f^{(l)}, \omega_j) \approx \boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f^{(l)}, \omega_j))$
$l = 1,\dots, B_p$ and $j = 1,\dots, F_p$

Fig. 3.1 Neural Space Mapping (NSM) Optimization.

The amount of deviation from $\boldsymbol{x}_c^*$ for each design parameter is determined according to the coarse model sensitivities. The larger the sensitivity of the coarse model

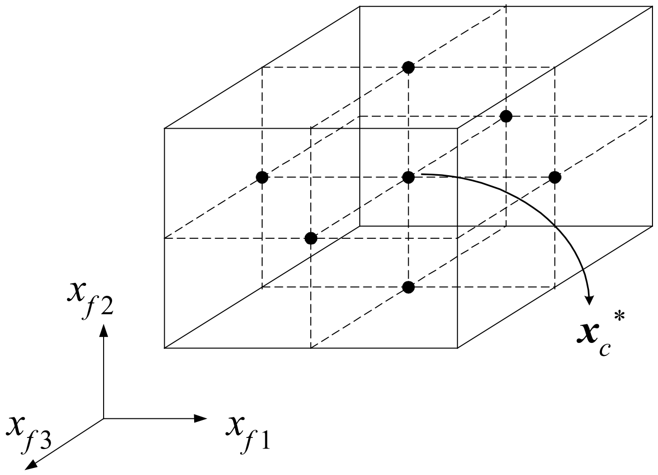


Fig. 3.2 Three-dimensional star set for the initial base points during NSM optimization.

response w.r.t. a certain parameter, the smaller the percentage of variation of that parameter. We assume that the coarse model sensitivities are similar to those of the fine model, which is usually the case in many practical problems since the coarse model represents the same physical system as the fine model.

The fine model response $\boldsymbol{R}_f$ at the optimal coarse model solution $\boldsymbol{x}_c^*$ is then calculated. If $\boldsymbol{R}_f$ is approximately equal to the desired response, the algorithm ends, otherwise we develop an SM-based neuromodel over the $2n+1$ fine model points initially available.

Once an SM-based neuromodel with small learning errors is available, we use it as an improved coarse model, optimizing its parameters to generate the desired response. The solution to this optimization problem becomes the next point in the fine model parameter space, and it is included in the learning set.

We calculate the fine model response at the new point, and compare it with the desired response. If it is still different, we re-train the SM-based neuromodel over the extended set of learning samples and the algorithm continues, otherwise, the algorithm terminates.

3.5 COARSE OPTIMIZATION

During the coarse optimization phase of NSM optimization (see Fig. 3.1), we want to find the optimal coarse model solution $\boldsymbol{x}_c^*$ that generates the desired response over the frequency range of interest. The vector of coarse model responses $\boldsymbol{R}_c$ might contain r different responses of the circuit,

$$\boldsymbol{R}_c(\boldsymbol{x}_c) = [\boldsymbol{R}_c^1(\boldsymbol{x}_c)^T \quad \ldots \quad \boldsymbol{R}_c^r(\boldsymbol{x}_c)^T]^T \tag{3-3}$$

where each individual response has been sampled at F_p frequency points,

$$\boldsymbol{R}_c^k(\boldsymbol{x}_c) = [R_c^k(\boldsymbol{x}_c, \omega_1) \quad \ldots \quad R_c^k(\boldsymbol{x}_c, \omega_{F_p})]^T, \;\; k = 1, \ldots, r \tag{3-4}$$

The desired response $\boldsymbol{R}^*$ is expressed in terms of specifications. Following Bandler and Chen (1988), the problem of circuit design using the coarse model can be formulated as

$$\boldsymbol{x}_c^* = \arg \min_{\boldsymbol{x}_c} U(\boldsymbol{R}_c(\boldsymbol{x}_c)) \tag{3-5}$$

where U is a suitable objective function. Typically, U is a minimax objective function expressed in terms of upper and lower specifications for each response and frequency sample. Bandler and Chen (1988) formulated a rich collection of objective functions, for different design constraints.

3.6 REFINING THE SM-BASED NEUROMODEL DURING NSM OPTIMIZATION

At the ith iteration, we want to find the simplest neuromapping $\boldsymbol{P}^{(i)}$ such that the coarse model using that mapping approximates the fine model at all the accumulated learning base points. We are interested in the simplest neural network in order to avoid poor generalization performance, since no testing samples are being used.

This is realized by solving the optimization problem

$$\boldsymbol{w}^* = \arg \min_{\boldsymbol{w}} \left\| [\cdots \quad \boldsymbol{e}_s^{\,T} \quad \cdots]^T \right\| \tag{3-6}$$

with

$$\boldsymbol{e}_s = \boldsymbol{R}_f(\boldsymbol{x}_f^{\,(l)}, \omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_{c_j}^{\,(l)}, \omega_{c_j}) \ , \ \boldsymbol{e}_s \in \Re^r \tag{3-7}$$

$$\begin{bmatrix} \boldsymbol{x}_{c_j}^{\,(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f^{\,(l)}, \omega_j, \boldsymbol{w}) \tag{3-8}$$

$$j = 1, \ldots, F_p \tag{3-9}$$

$$l = 1, \ldots, 2n + i \tag{3-10}$$

$$s = j + F_p(l-1) \tag{3-11}$$

where $2n + i$ is the number of training base points for the input design parameters and F_p is the number of frequency points per frequency sweep. It is seen that the total number of learning samples at the *i*th iteration is $(2n+i)F_p$, and the length of the total error vector in (3-6) is $(2n+i)rF_p$.

(3-8) is the input-output relationship of the ANN that implements the mapping at the *i*th iteration. Vector $\boldsymbol{w}$ contains the internal parameters (weights, bias, etc.) of the ANN. The paradigm chosen to implement $\boldsymbol{P}$ is a 3-layer perceptron (3LP).

All the SM-based neuromodeling techniques proposed by Bandler, Ismail, Rayas-Sánchez and Zhang (1999), described in the previous chapter, can be exploited to efficiently solve (3-6).

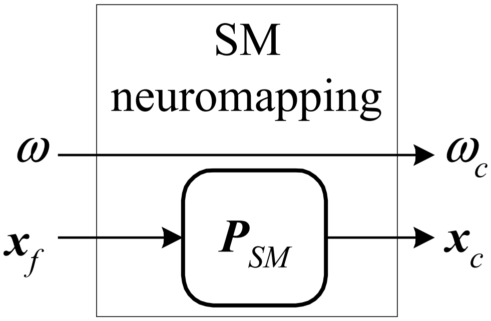


Fig. 3.3 Space Mapped neuromapping.

In the Space Mapped (SM) neuromodeling approach only the design parameters are mapped, and the coarse model is simulated at the same frequency as the fine model. The corresponding neuromapping is illustrated in Fig. 3.3, which is expressed as

$$\begin{bmatrix} \boldsymbol{x}_{c_j}{}^{(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{P}_{SM}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \boldsymbol{w}) \\ \omega_j \end{bmatrix} \tag{3-12}$$

In the Frequency-Dependent Space Mapped (FDSM) neuromodeling approach, both coarse and fine models are simulated at the same frequency, but the mapping from the fine to the coarse parameter space is dependent on the frequency. The corresponding neuromapping is illustrated in Fig. 3.4, and it is expressed as

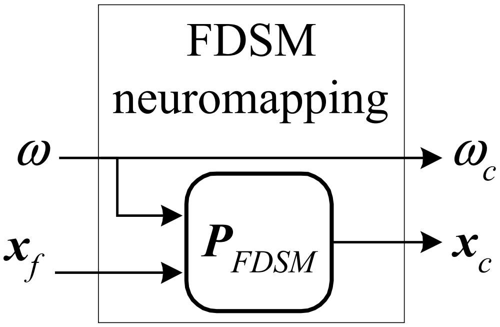


Fig. 3.4 Frequency-Dependent Space Mapped neuromapping.

$$\begin{bmatrix} \boldsymbol{x}_{c_j}{}^{(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{P}^{(i)}_{FDSM}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) \\ \omega_j \end{bmatrix} \tag{3-13}$$

The Frequency Space Mapped (FSM) neuromodeling technique establishes a mapping not only for the design parameters but also for the frequency variable, such that the coarse model is simulated at a different frequency to match the fine model response. The FSM neuromapping is illustrated in Fig. 3.5, and it is expressed as

$$\begin{bmatrix} \boldsymbol{x}_{c_j}{}^{(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) = \left[\boldsymbol{P}^{(i)}_{FSM}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w})\right] \tag{3-14}$$

For those cases where the shapes of the fine and coarse model responses are very similar but shifted in frequency, the Frequency Mapped (FM) neuromodeling technique simulates the coarse model with the same physical parameters used by the fine model, but at a different frequency to align both responses. In this manner, we can compensate the coarse model when it consists of a circuit-theoretic approximation based on quasi-static assumptions. Fig. 3.6 illustrates the FM neuromapping, which is expressed as

$$\begin{bmatrix} \boldsymbol{x}_{c_j}{}^{(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{x}_f{}^{(l)} \\ P^{(i)}_{FM}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) \end{bmatrix} \tag{3-15}$$

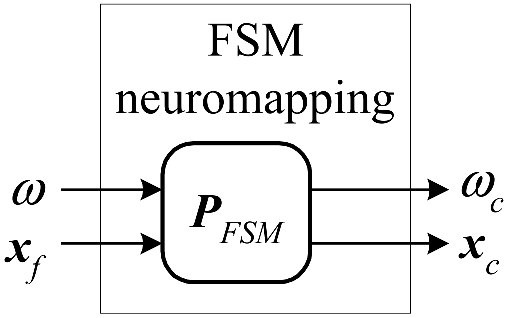


Fig. 3.5 Frequency Space Mapped neuromapping.

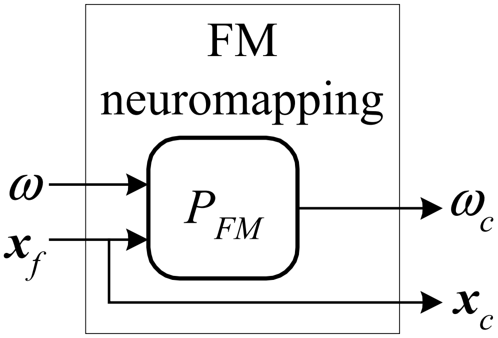


Fig. 3.6 Frequency Mapped neuromapping.

Finally, the Frequency Partial-Space Mapped (FPSM) neuromodeling technique maps only some of the design parameters and the frequency, making an even more efficient use of the implicit knowledge in the coarse model. The FPSM neuromapping is represented in Fig. 3.7, and it is mathematically expressed as

$$\begin{bmatrix} \boldsymbol{x}_{c_j}{}^{(l)} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{x}_f^{\bullet\,(l)} \\ \boldsymbol{x}_{c_j}^{\bullet\,(l)} \\ \omega_{c_j} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_f^{\bullet\,(l)} \\ \boldsymbol{P}_{FPSM}^{(i)}(\boldsymbol{x}_f{}^{(l)}, \omega_j, \boldsymbol{w}) \end{bmatrix} \tag{3-16}$$

Note that the “design” parameters of the coarse model do not change with frequency only in the SMN and FM neuromappings.

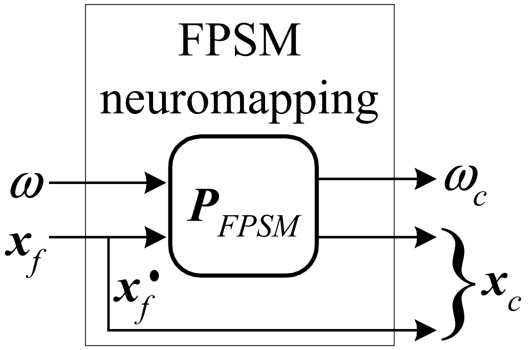


Fig. 3.7 Frequency Partial-Space Mapped neuromapping.

The starting point for the first training process is a unit mapping, i.e., $\boldsymbol{P}^{(0)}(\boldsymbol{x}_f^{(l)}, \omega_j, \boldsymbol{w}_u) = [\boldsymbol{x}_f^{(l)T}\ \omega_j]^T$, for $j = 1,\ldots, F_p$ and $l = 1,\ldots, 2n+1$, where $\boldsymbol{w}_u$ contains the internal parameters of the ANN that give a unit mapping. The SM-based neuromodel is trained in the next iterations using the previous mapping as the starting point.

The complexity of the ANN (the number of hidden neurons and the SM-based neuromodeling technique) is gradually increased according to the learning error ε_L, starting with a linear mapping (3-layer perceptron with 0 hidden neurons). In other words, we use the simplest ANN that yields an acceptable learning error ε_L, defined as

$$\varepsilon_L = \left\| [\cdots \quad \boldsymbol{e}_s{}^T \quad \cdots]^T \right\| \tag{3-17}$$

where $\boldsymbol{e}_s$ is obtained from (3-7) using the current optimal values for the ANN internal parameters $\boldsymbol{w}^*$.

In our implementation, the neuromapping for the first iteration is approximated using the FMN technique, so that any possible severe misalignment in frequency between the coarse and the fine model responses is first alleviated. Then, the physical parameters are gradually mapped, following a FPSM technique.

Linear Adaptive Frequency-Space Mapping (LAFSM) is a special case of NSM optimization, corresponding to the situation when the number of hidden neurons of the ANN is zero at all iterations.

When solving (3-6) at each NSM iteration, the number of unknowns, i.e., the length of $\boldsymbol{w}$, is established by the kind of neuromapping selected (SM, FDSM, FSM, FM or FPSM) and the number of hidden neurons used. On the other hand, the number of equations is fixed and given by $(2n+i)rF_p$. At each NSM iteration we verify that the length of $\boldsymbol{w}$ is not larger than $(2n+i)rF_p$ to avoid solving an under-determined system when training the SM-based neuromodel.

3.7 SM-BASED NEUROMODEL OPTIMIZATION

At the *i*th iteration of NSM optimization, we use the simplest SM-based neuromodel with small learning error over the 2*n*+*i* accumulated points as an improved coarse model, optimizing its design parameters to generate the desired response. The solution to the direct optimization of this mapped, enhanced, coarse model gives us the next iterate.

We denote the SM-based neuromodel response as $\boldsymbol{R}_{SMBN}$, defined as

$$\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f) = [\boldsymbol{R}_{SMBN}^1(\boldsymbol{x}_f)^T \quad \dots \quad \boldsymbol{R}_{SMBN}^r(\boldsymbol{x}_f)^T]^T \tag{3-18}$$

where

$$\boldsymbol{R}_{SMBN}^k(\boldsymbol{x}_f) = [R_c^k(\boldsymbol{x}_{c1}, \omega_{c1}) \quad \dots \quad R_c^k(\boldsymbol{x}_{cF_p}, \omega_{cF_p})]^T,\ k = 1, \dots, r \tag{3-19}$$

with

$$\begin{bmatrix} \boldsymbol{x}_{c_j} \\ \omega_{c_j} \end{bmatrix} = \boldsymbol{P}^{(i)}(\boldsymbol{x}_f, \omega_j, \boldsymbol{w}^*) \tag{3-20}$$

$$j = 1, \dots, F_p \tag{3-21}$$

The solution to the following optimization problem becomes the next iterate:

$$\boldsymbol{x}_f^{(2n+i+1)} = \arg \min_{\boldsymbol{x}_f} U(\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f)) \tag{3-22}$$

with *U* defined as in (4-5). If an SMN neuromapping is used to implement $\boldsymbol{P}^{(i)}$ (see Fig. 3.3), the next iterate can be obtained in a simpler manner by solving

$$\boldsymbol{x}_f^{(2n+i+1)} = \arg \min_{\boldsymbol{x}_f} \left\| \boldsymbol{P}_{SM}^{(i)}(\boldsymbol{x}_f, \boldsymbol{w}^*) - \boldsymbol{x}_c^* \right\| \tag{3-23}$$

3.8 NSM ALGORITHM

The proposed algorithm for implementing NSM is as follows

Step 0. Find $\boldsymbol{x}_c^*$ by solving (3-5).

Step 1. Choose $\boldsymbol{x}_f^{(1)},\ldots,\ \boldsymbol{x}_f^{(2n)}$ following a star set around $\boldsymbol{x}_c^*$.

Step 2. Initialize $i=1$, $\boldsymbol{x}_f^{(2n+i)} = \boldsymbol{x}_c^*$.

Step 3. Stop if $\left\| \boldsymbol{R}_f(\boldsymbol{x}_f^{(2n+i)},\omega_j) - \boldsymbol{R}_c(\boldsymbol{x}_c^*,\omega_j) \right\| \le \varepsilon_R$, $j=1,\ldots,F_p$.

Step 4. Initialize $\boldsymbol{P}^{(i)} = \boldsymbol{P}^{(i-1)}$, where

$$\boldsymbol{P}^{(0)}(\boldsymbol{x}_f^{(l)},\omega_j,\boldsymbol{w}_u) = \begin{bmatrix} \boldsymbol{x}_f^{(l)} \\ \omega_j \end{bmatrix},\ j=1,\ldots,F_p;\ \ l=1,\ldots,2n+i\,.$$

Step 5. Find $\boldsymbol{w}^*$ by solving (3-6).

Step 6. Calculate ε_L using (3-17).

Step 7. If $\varepsilon_L > \varepsilon_{\min}$, increase the complexity of $\boldsymbol{P}^{(i)}$ and go to Step 5.

Step 8. If an SM neuromapping is used to implement $\boldsymbol{P}^{(i)}$, solve (3-23), otherwise solve (3-22).

Step 9. Set $i=i+1$; go to Step 3.

3.9 EXAMPLES

3.9.1 HTS Microstrip Filter

We apply NSM optimization to a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter (Bandler, Biernacki, Chen, Getsinger, Grobelny, Moskowitz and Talisa, 1995), whose physical structure is illustrated in Fig. 2.21. L_1, L_2 and L_3 are the lengths of the parallel coupled-line sections and S_1, S_2 and S_3 are the separations between the sections. The width W is the same for all the microstrip sections as well as for the input and output microstrip lines, which have a length L_0. A lanthanum aluminate substrate with thickness H and dielectric constant ε_r is used.

The specifications are $|S_{21}| \geq 0.95$ in the passband and $|S_{21}| \leq 0.05$ in the stopband, where the stopband includes frequencies below 3.967 GHz and above 4.099 GHz, and the passband lies in the range [4.008GHz, 4.058GHz]. The design parameters are $\boldsymbol{x}_f = [L_1\ L_2\ L_3\ S_1\ S_2\ S_3]^T$. We take L_0 = 50 mil, H = 20 mil, W = 7 mil, ε_r = 23.425, loss tangent = 3×10^{-5}; the metalization is considered lossless.

Sonnet's ***em***™ (1997) driven by Empipe™ (1997) is employed as the fine model, using a high-resolution grid with a 1mil×1mil cell size, with interpolation disabled.

The conceptual schematic of the coarse model used for the HTS filter is illustrated in Fig. 3.8. OSA90/hope™ (1997) built-in linear elements MSL (microstrip line), MSCL (two-conductor symmetrical coupled microstrip lines) and OPEN (open circuit) connected by circuit theory over the same MSUB (microstrip substrate definition)

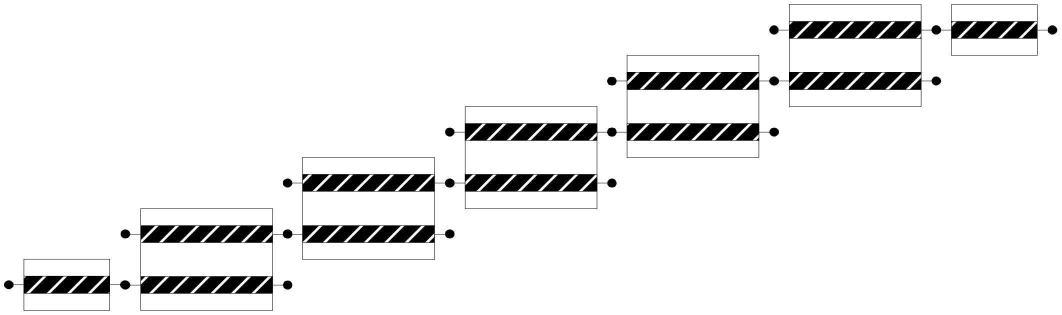


Fig. 3.8 Representation of the coarse model for the HTS microstrip filter.

are taken as the "coarse" model.

The following optimal coarse model solution is found: $\boldsymbol{x}_c^* = [188.33 \quad 197.98 \quad 188.58 \quad 21.97 \quad 99.12 \quad 111.67]^T$ (mils), as in the work by Bakr, Bandler, Biernacki, Chen, and Madsen (1998). The coarse and fine model responses at the optimal coarse solution are shown in Fig. 3.9.

The initial $2n$+1 points are chosen by performing sensitivity analysis on the coarse model: a 3% deviation from $\boldsymbol{x}_c^*$ for L_1, L_2, and L_3 is used, while a 20% is used for S_1, S_2, and S_3. The corresponding fine and coarse model responses at these 13 star-set learning points are shown in Fig. 3.10.

Fig. 3.11 shows the evolution of the learning errors at the $2n$+1 points as we increase the complexity of the neuromapping during the first iteration. It is seen that mapping the frequency has a dramatic effect on the alignment of the responses, and a

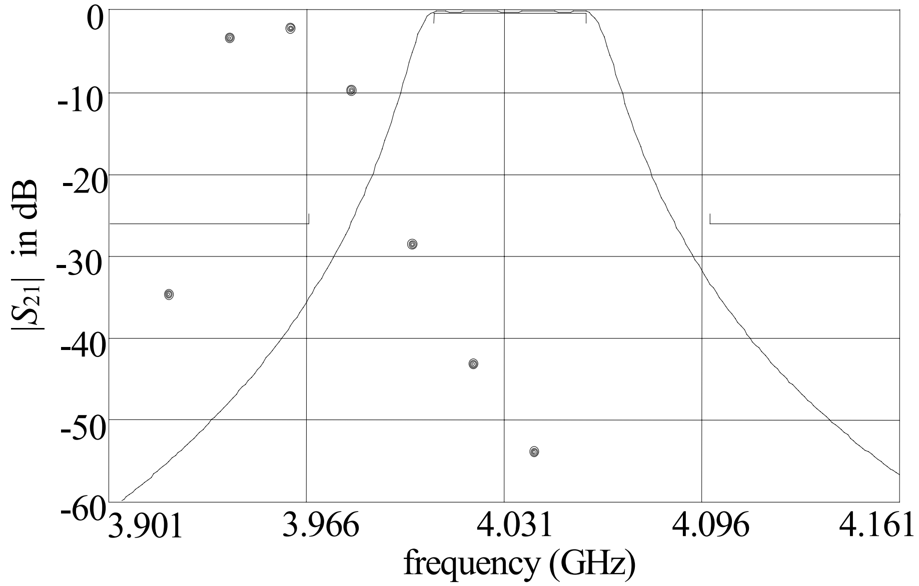
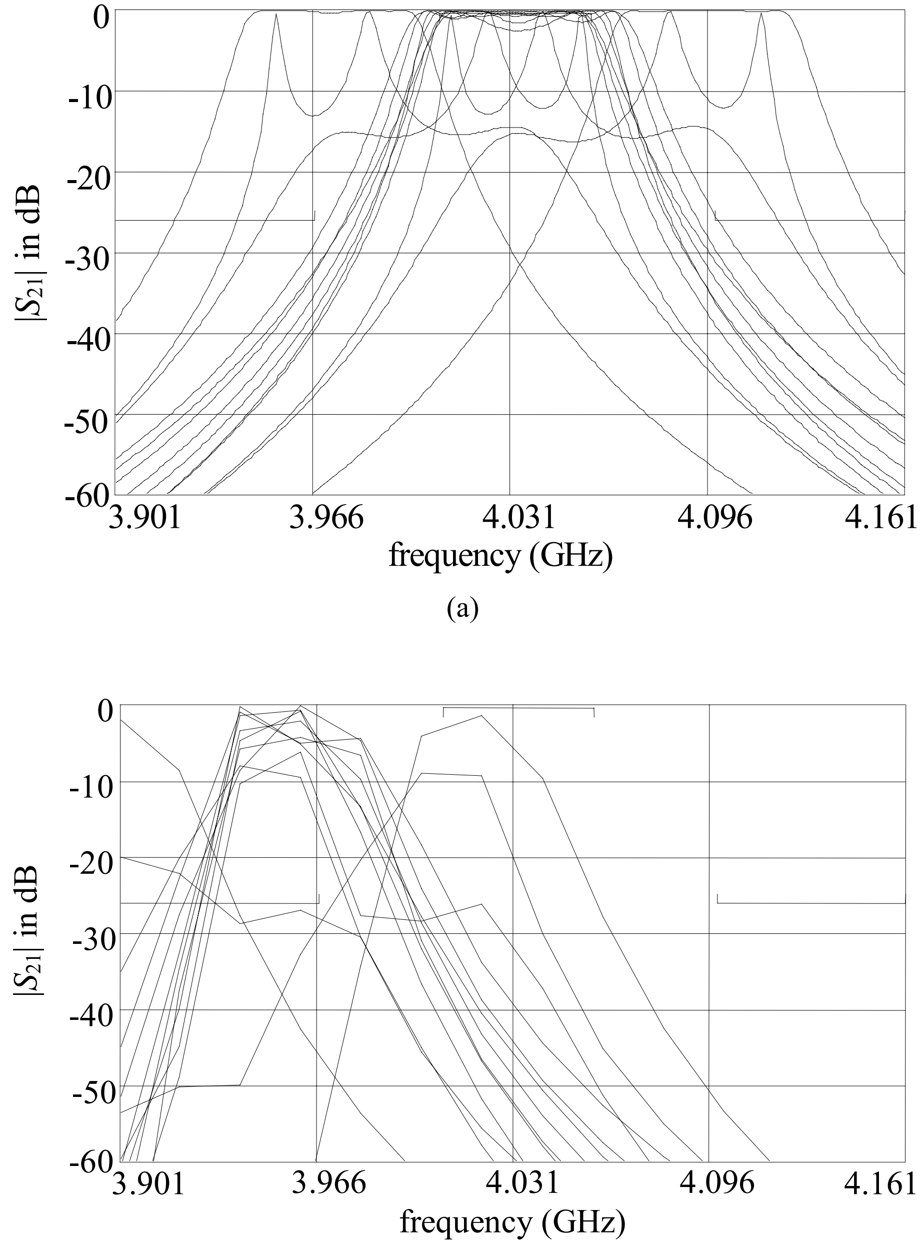


Fig. 3.9 Coarse and fine model responses at the optimal coarse solution: OSA90/hope™ (–) and ***em***™ (●).

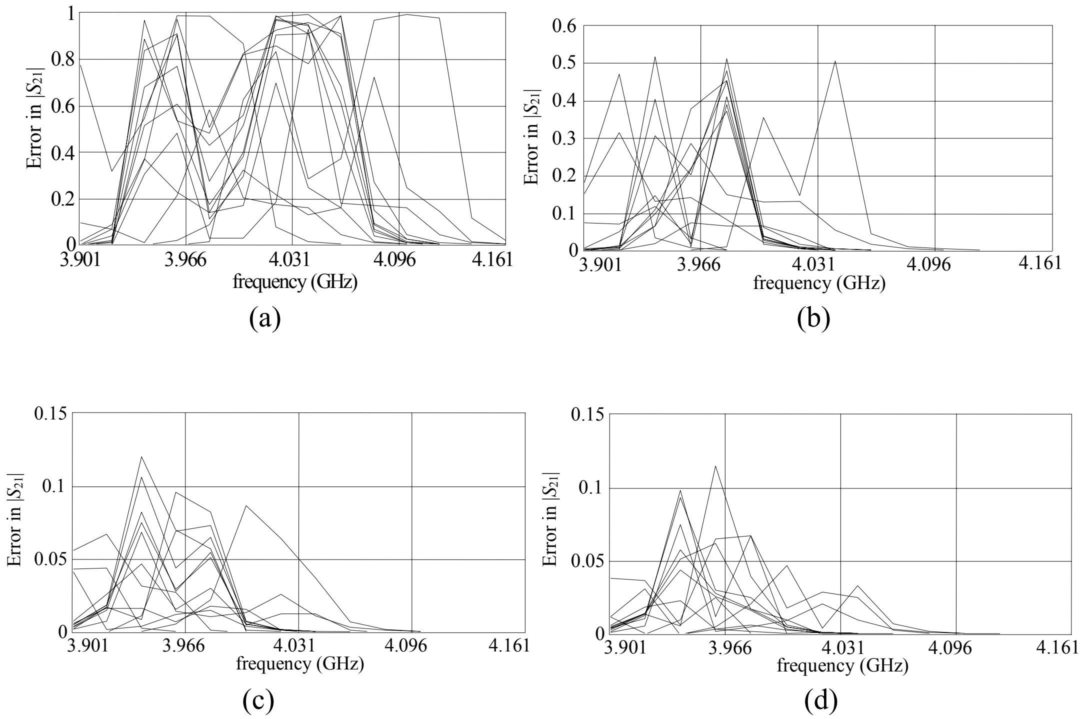
simple FPSM neuromapping is needed. The final mapping is implemented with a 3-layer perceptron with 7 inputs (6 design parameters and the frequency), 5 hidden neurons, and 3 output neurons (ω, L_1, and S_1).



(a)

(b)

Fig. 3.10 Coarse and fine model responses at the initial $2n+1$ base points around the optimal coarse solution: (a) OSA90/hope™, (b) ***em***™.



(a)

(b)

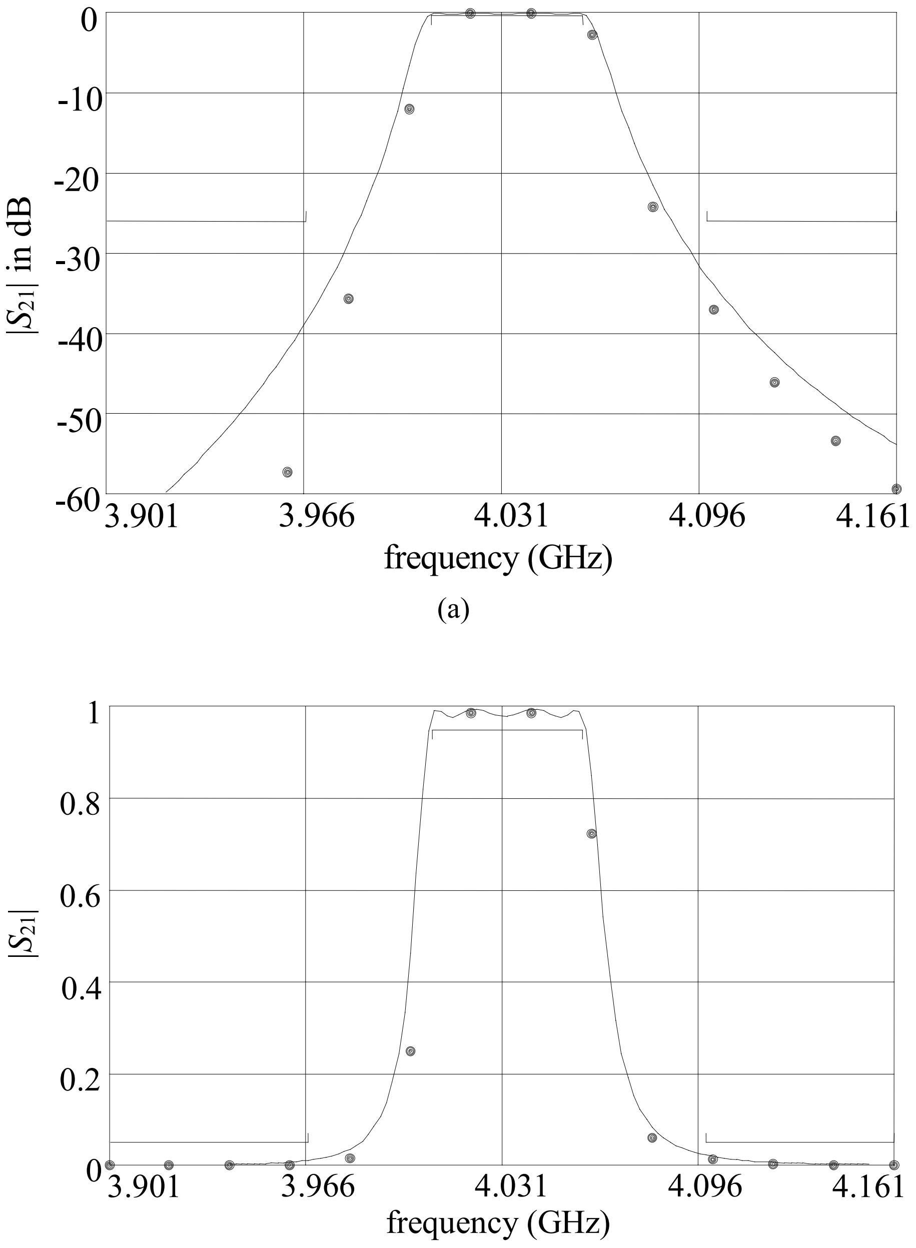
(c)

(d)

Fig. 3.11 Learning errors at initial base points: (a) at the starting point, (b) mapping ω with a 3LP:7-3-1, (c) mapping ω and L1 with a 3LP:7-4-2, and (d) mapping ω, L1 and S1 with a 3LP:7-5-3.

As indicated in Step 8 of the NSM algorithm, we calculate the next point by optimizing the coarse model with the mapping found. The next point predicted is $\boldsymbol{x}_f^{(14)} =$ [185.37 195.01 184.24 21.04 86.36 91.39] ^{T} (mils), which matches the desired response with excellent accuracy, as seen in Fig. 3.12.

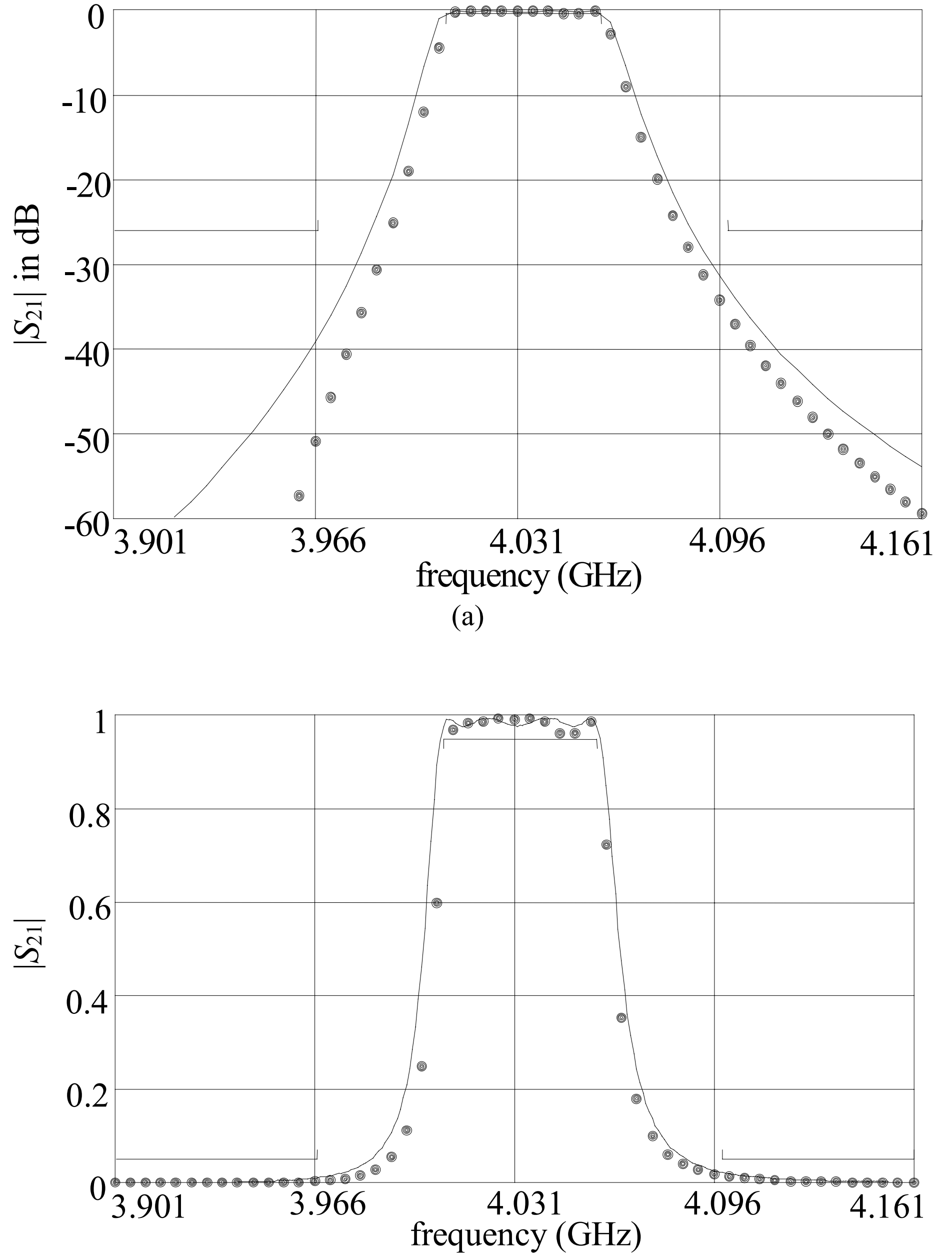
As a final test, both the FPSMN model and the fine model are simulated at the NSM solution $\boldsymbol{x}_f^{\ (14)}$ using a very fine frequency sweep, with a frequency step of 0.005GHz. The NSM solution satisfies the specifications, as shown in Fig. 3.13. A detailed illustration of the passband using an even finer frequency sweep is shown in Fig. 3.14. The HTS filter is optimized in only one NSM iteration.



(a)

(b)

Fig. 3.12 ***em***™ (•) and FPSM 7-5-3 (–) model responses at the next point predicted after the first NSM iteration: (a) $|S_{21}|$ in dB, (b) $|S_{21}|$.



(a)

(b)

Fig. 3.13 ***em***™ (•) and FPSMN 7-5-3 (–) model responses, using a fine frequency sweep, at the next point predicted after the first NSM iteration: (a) |S_{21}| in dB, (b) |S_{21}|.

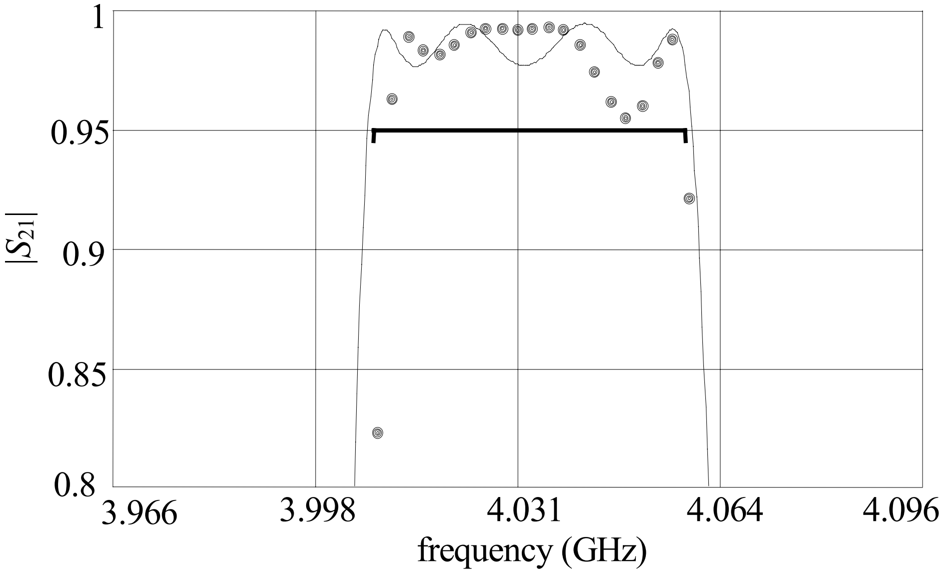


Fig. 3.14 ***em***™ (•) and FPSMN 7-5-3 (–) model responses in the passband, using a fine frequency sweep, at the next point predicted after the first NSM iteration.

3.9.2 Bandstop Microstrip Filter With Open Stubs

NSM optimization is applied to a bandstop microstrip filter with quarter-wave resonant open stubs, illustrated in Fig. 3.15. L_1, L_2 are the open stub lengths and W_1, W_2 the corresponding widths. An alumina substrate with thickness H = 25 mil, width W_0 = 25 mil and dielectric constant ε_r = 9.4 is used for a 50 Ω feeding line.

The specifications are $|S_{21}| \leq 0.05$ in the stopband and $|S_{21}| \geq 0.9$ in the passband, where the stopband lies between 9.3 GHz and 10.7 GHz, and the passband includes frequencies below 8 GHz and above 12 GHz. The design parameters are $\boldsymbol{x}_f = [W_1\ W_2\ L_0\ L_1\ L_2]^T$.

Sonnet's ***em***™ (1997) driven by Empipe™ (1997) was employed as the fine model, using a high-resolution grid with a 1mil×1mil cell size.

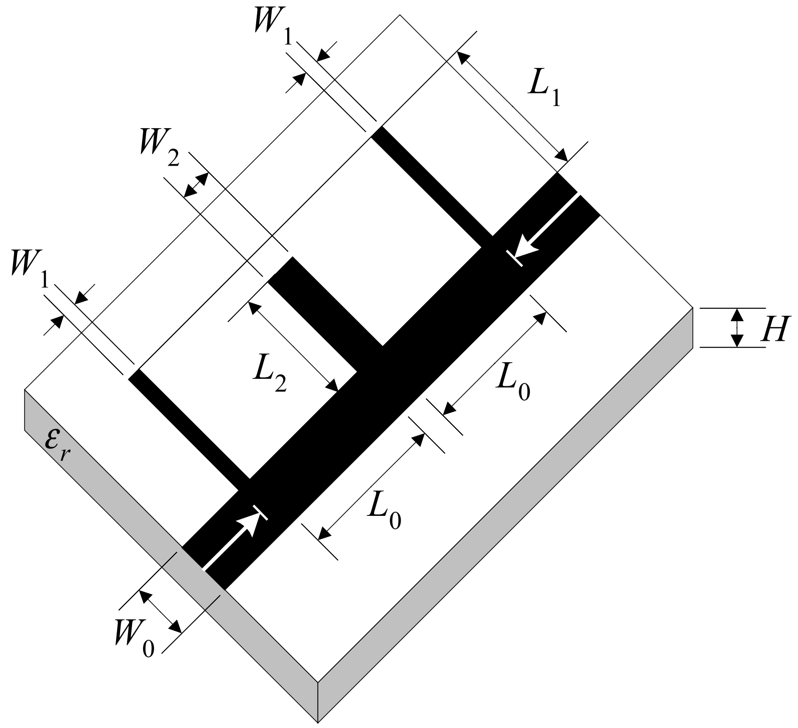


Fig. 3.15 Bandstop microstrip filter with quarter-wave resonant open stubs.

As the coarse model, we use simple transmission lines for modeling each microstrip section (see Fig. 3.16) and classical formulas, see, e.g., Pozar (1998), to calculate the characteristic impedance and the effective dielectric constant of each transmission line. It is seen that $L_{c2} = L_2 + W_0/2$, $L_{c1} = L_1 + W_0/2$, and $L_{c0} = L_0 + W_1/2 + W_2/2$. We use OSA90/hope™ (1997) built-in transmission line elements TRL.

The following optimal coarse model solution is found for L_0, L_1, and L_2 of quarter-wave lengths at 10 GHz: $\boldsymbol{x}_c^* = [6.00 \;\; 9.01 \;\; 106.45 \;\; 110.15 \;\; 108.81]^T$ (mils). The coarse and fine model responses at the optimal coarse solution are shown in Fig. 3.17.

The initial 2n+1 points are chosen by performing sensitivity analysis on the coarse model: a 50% deviation from $\boldsymbol{x}_c^*$ for W_1, W_2, and L_0 is used, while a 15% is used

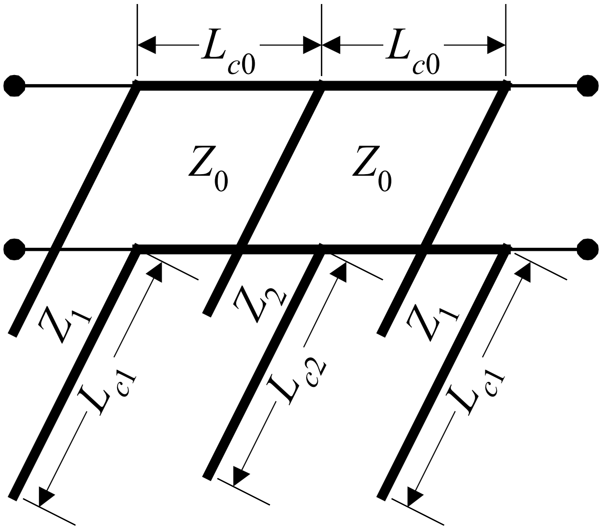


Fig. 3.16 Coarse model of the bandstop microstrip filter with open stubs.

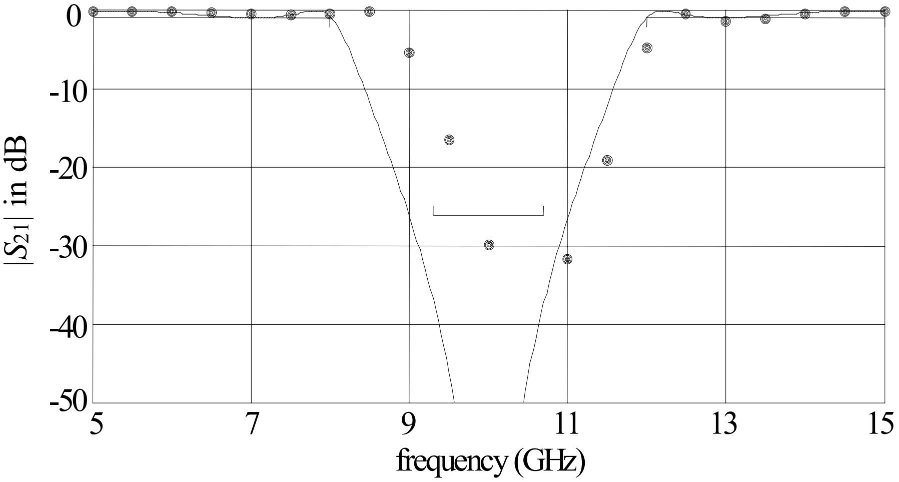


Fig. 3.17 Coarse and fine model responses at the optimal coarse solution: OSA90/hope™ (−) and ***em***™ (•).

for L_1, and L_2.

Initially, a simple FM neuromapping (see Fig. 3.6) with 2 hidden neurons (3LP:6-2-1, ω) was used to match the responses at the learning base points. The FM neuromodel and the fine model responses at the optimal coarse solution are shown in Fig. 3.18.

Optimizing the FM neuromodel to satisfy the specifications (Step 8 of the NSM algorithm), the next iterate is $\boldsymbol{x}_f^{(12)} = [6.54 \;\; 16.95 \;\; 91.26 \;\; 113.30 \;\; 120.72]^T$ (mils). The coarse and fine model responses at this point are shown in Fig. 3.19.

We performed a second NSM iteration. $\boldsymbol{x}_f^{(12)}$ is included in the learning base points. Now a FPSM neuromapping with 3 hidden neurons is needed to match the $2n$+2 points: only ω and W_2 are mapped (3LP:6-3-2, ω, W_2). Fig. 3.20 shows the FPSM neuromodel and the fine model responses at $\boldsymbol{x}_f^{(12)}$. Optimizing the FPSM neuromodel, the next iterate is $\boldsymbol{x}_f^{(13)} = [5.92 \;\; 13.54 \;\; 83.34 \;\; 114.14 \;\; 124.81]^T$ (mils). The coarse and fine model responses at $\boldsymbol{x}_f^{(13)}$ are shown in Fig. 3.21.

As a final test, using a fine frequency sweep, we show in Fig. 3.22 the fine model response at $\boldsymbol{x}_f^{(13)}$ and the optimal coarse response. The bandstop microstrip filter is optimized in two NSM iterations.

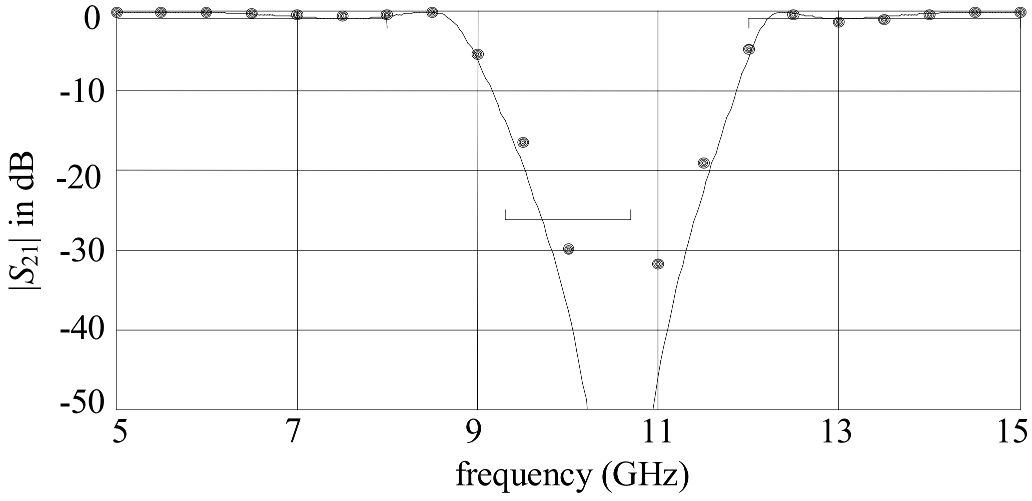


Fig. 3.18 FM (3LP:6-2-1, ω) neuromodel (–) and the fine model (•) responses at the optimal coarse solution.

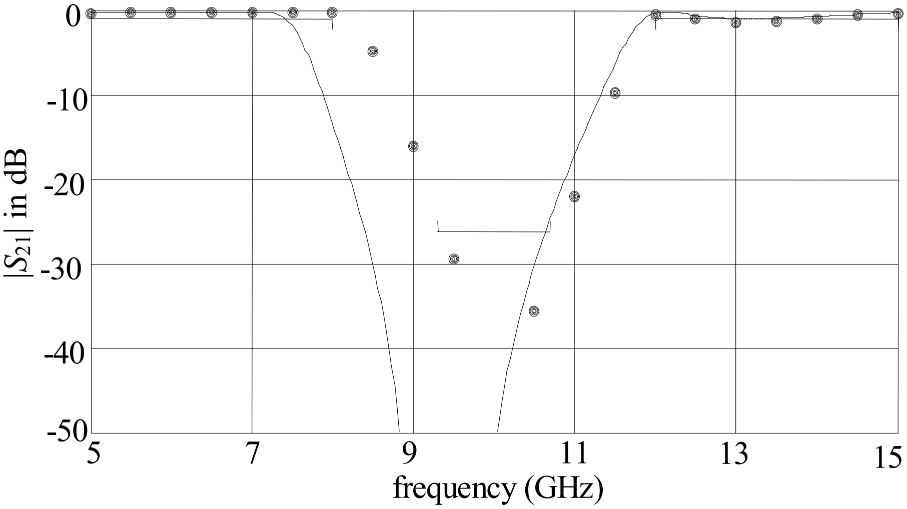


Fig. 3.19 Coarse (−) and fine (•) model responses at the next point predicted by the first NSM iteration.

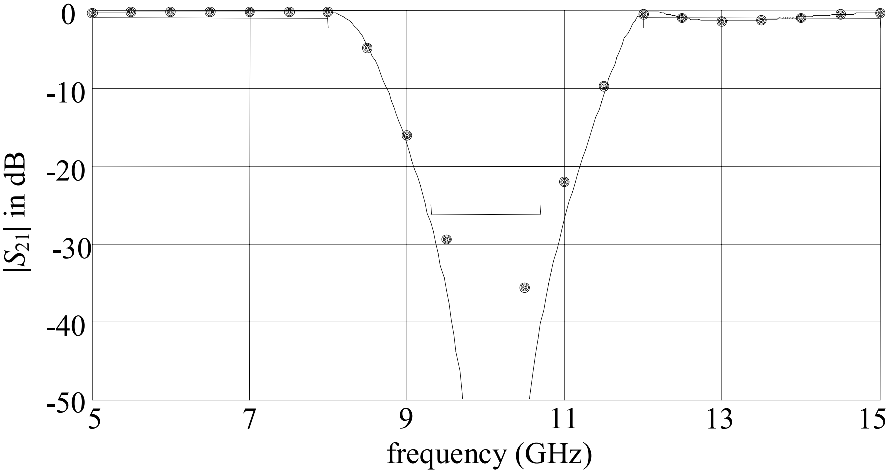


Fig. 3.20 FPSM (3LP:6-3-2, ω, W_2) neuromodel (−) and the fine model (•) responses at the point predicted by the first NSM iteration.

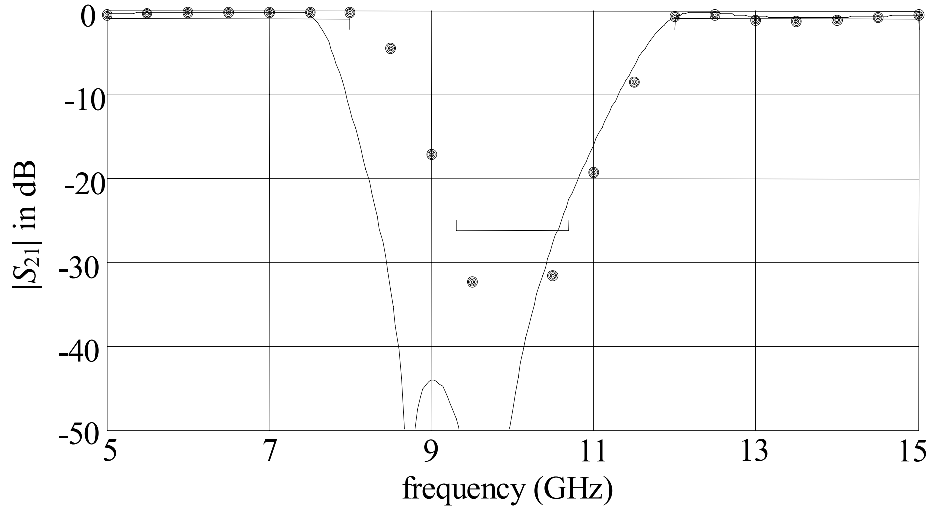


Fig. 3.21 Coarse (−) and fine model (•) responses at the next point predicted by the second NSM iteration.

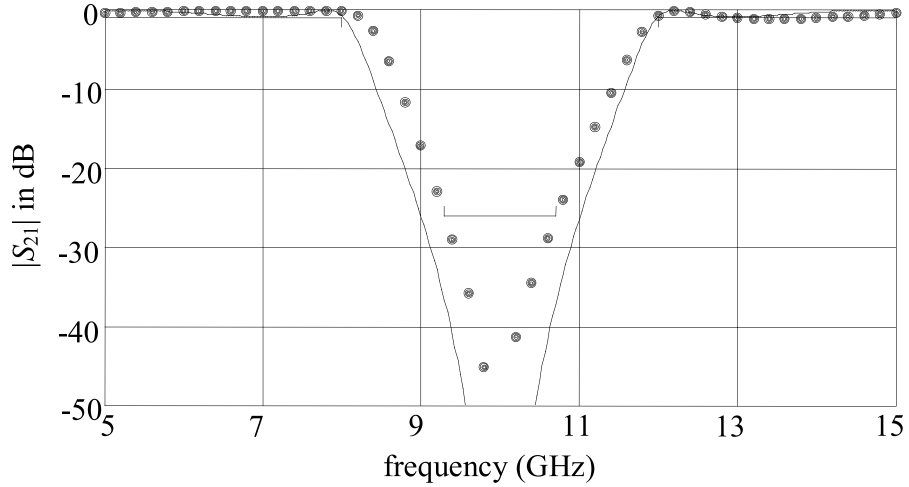


Fig. 3.22 Fine model response (•) at the next point predicted by the second NSM iteration and optimal coarse response (−), using a fine frequency sweep.

3.10 CONCLUDING REMARKS

We have described in this chapter an innovative algorithm for EM optimization based on Space Mapping technology and Artificial Neural Networks. Neural Space Mapping (NSM) optimization exploits our SM-based neuromodeling techniques, described in Chapter 2, to efficiently approximate the mapping from the fine to the coarse input space at each iteration.

NSM does not require parameter extraction to predict the next point. An initial mapping is established by performing upfront fine model analysis at a reduced number of base points. The coarse model sensitivities are exploited to select those base points.

Huber optimization is used to train simple SM-based neuromodels at each iteration. The SM-based neuromodels are developed without using testing points: their generalization performance is controlled by gradually increasing their complexity starting with a 3-layer perceptron with 0 hidden neurons.

A high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter and a bandstop microstrip filter with quarter-wave resonant open stubs illustrate our optimization technique.

NSM optimization was developed by Bakr, Bandler, Ismail, Rayas-Sánchez and Zhang (2000a-c). NSM optimization is compared in Chapter 5 with another SM-based neural network optimization algorithm.

Chapter 4

YIELD EM OPTIMIZATION VIA SM-BASED NEUROMODELS

4.1 INTRODUCTION

Accurate yield optimization and statistical analysis of microwave components are crucial ingredients for manufacturability-driven designs in a time-to-market development environment. Yield optimization requires intensive simulations to cover the entire statistic of possible outcomes of a given manufacturing process.

Electromagnetic (EM) full-wave field solvers are regarded as highly accurate to predict the behavior of microwave structures. With the increasing availability of commercial EM simulators, it is very desirable to include them in the statistical analysis and yield-driven design of microwave circuits. Given the high cost in computational effort imposed by the EM simulators, creative procedures must be searched to efficiently use them for statistical analysis and design.

Yield-driven EM optimization using multidimensional quadratic models that approximate the EM model responses for efficient and accurate evaluations was proposed by Bandler, Biernacki, Chen, Grobelny and Ye (1993). A more integrated CAD environment for statistical analysis and yield-driven circuit design was later proposed in the work by Bandler, Biernacki, Chen and Grobelny (1994), where the quadratic modeling techniques and interpolation techniques (to deal with the discretization of the geometrical parameters of the EM structure) were unified.

We describe in this chapter the use of space mapping-based neuromodels for efficient and accurate EM-based statistical analysis and yield optimization of microwave structures. We mathematically formulate the yield optimization problem using SM-based neuromodels. A general equation to express the relationship between the fine and coarse model sensitivities through a nonlinear, frequency-sensitive neuromapping is presented. This equation represents a generalization of a lemma found in previous work following a different approach.

We illustrate the use of space mapping based neuromodels for EM statistical analysis and yield optimization by a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter.

4.2 STATISTICAL CIRCUIT ANALYSIS AND DESIGN: PROBLEM FORMULATION

In practice, random variations in the manufacturing process of a microwave device may result in a significant percentage of the produced devices not meeting the specifications. When designing, it is essential to account for these inevitable uncertainties. Many recent significant contributions have been made to the statistical analysis and design of microwave circuits, see for example the work by Bandler, Biernacki, Chen, Grobelny and Ye (1993), Bandler, Biernacki, Chen and Grobelny (1994), Carroll and Chang (1996), etc. An excellent review of different approaches to statistical design can be found in the work by Song (1991).

Let $\boldsymbol{x} \in \Re^n$ represent the vector of n design parameters of the microwave device whose r responses at frequency ω are contained in vector $\boldsymbol{R}(\boldsymbol{x}, \omega) \in \Re^r$ (for example, $\boldsymbol{R}(\boldsymbol{x}$

,ω) might contain the real and imaginary parts of S_{11} at 10 GHz for a given physical structure).

The design goals are defined by a vector $\boldsymbol{S}_u(\omega) \in \Re^r$ of upper specifications and a vector $\boldsymbol{S}_l(\omega) \in \Re^r$ of lower specifications imposed on the responses $\boldsymbol{R}(\boldsymbol{x}\ ,\omega)$ at each frequency of interests. A lower specification on the *k*th response at frequency ω requires $R_k(\boldsymbol{x}, \omega) \geq S_{lk}(\omega)$ while an upper specification requires $R_k(\boldsymbol{x}, \omega) \leq S_{uk}(\omega)$. It is possible to impose both a lower and an upper specification on a single response.

Two error vectors $\boldsymbol{e}_u$, $\boldsymbol{e}_l \in \Re^r$ can be used to measure the degree to which a response satisfies or violates the specifications,

$$\boldsymbol{e}_l(\boldsymbol{x},\omega) = \boldsymbol{S}_l(\omega) - \boldsymbol{R}(\boldsymbol{x},\omega) \tag{4-1}$$

$$\boldsymbol{e}_u(\boldsymbol{x},\omega) = \boldsymbol{R}(\boldsymbol{x},\omega) - \boldsymbol{S}_u(\omega) \tag{4-2}$$

Nonnegative weighting factors can be included in (4-1) and (4-2) for scaling purposes. In practice, these two error vectors are sampled at a finite set of frequency points of interest, not necessarily overlapping. The corresponding two sets of vectors can be combined in a single error vector

$$\boldsymbol{e}(\boldsymbol{x}) = [\boldsymbol{e}_{l1}{}^T \quad \boldsymbol{e}_{l2}{}^T \quad \cdots \quad \boldsymbol{e}_{u1}{}^T \quad \boldsymbol{e}_{u2}{}^T \quad \cdots]^T \tag{4-3}$$

whose dimensionality is denoted by M. Clearly, negative components in $\boldsymbol{e}$ indicate satisfaction of the corresponding specifications.

In the nominal design, we are interested in finding a single vector of design parameters $\boldsymbol{x}^*$, called optimal nominal solution, for which the responses $\boldsymbol{R}(\boldsymbol{x}^*)$ optimally satisfy the design specifications $\boldsymbol{S}_u$ and $\boldsymbol{S}_l$ at all frequency points of interest. Following Bandler and Chen (1988), this task can be formulated as a minimax optimization problem

$$\boldsymbol{x}^* = \arg \min_{\boldsymbol{x}} U(\boldsymbol{x}) \tag{4-4}$$

$$U(\boldsymbol{x}) = \max_{j} e_j(\boldsymbol{x}) \tag{4-5}$$

where $e_j(\boldsymbol{x})$ is the jth element in the error vector (4-3), with $j = 1 \ldots M$.

In the statistical approach to circuit design, we take into account that the design parameters of the manufactured device outcomes $\boldsymbol{x}^k$ are actually spread around the nominal point $\boldsymbol{x}$ according to their statistical distributions and tolerances. These parameters can be represented as

$$\boldsymbol{x}^k = \boldsymbol{x} + \boldsymbol{\Delta x}^k, \quad k = 1, 2, \ldots, N \tag{4-6}$$

where N is the number of such outcomes. We associate with each outcome an acceptance index I_a defined by

$$I_a(\boldsymbol{x}^k) = \begin{cases} 1, & \text{if } U(\boldsymbol{x}^k) \le 0 \\ 0, & \text{if } U(\boldsymbol{x}^k) > 0 \end{cases} \tag{4-7}$$

If N is sufficiently large for statistical significance, we can approximate the yield Y at the nominal point $\boldsymbol{x}$ by using

$$Y(\boldsymbol{x}) \approx \frac{1}{N} \sum_{k=1}^{N} I_a(\boldsymbol{x}^k) \tag{4-8}$$

An error vector $\boldsymbol{e}(\boldsymbol{x}^k) \in \Re^M$ is associated to each circuit outcome $\boldsymbol{x}^k$ according to (4-1)-(4-3). Following Grobelny (1991), the optimal yield solution $\boldsymbol{x}^{Y*}$ can be found by solving

$$\boldsymbol{x}^{Y*} = \arg \min_{\boldsymbol{x}} \sum_{k \in K} \alpha_k H_1(\boldsymbol{x}^k) \tag{4-9}$$

$$K = \left\{ k \,\middle|\, H_1(\boldsymbol{x}^k) > 0 \right\} \tag{4-10}$$

where $H_1(\boldsymbol{x}^k)$ is the generalized l_1 function

$$H_1(\boldsymbol{x}^k) = \begin{cases} \sum_{j\in J} e_j(\boldsymbol{x}^k) & \text{if } J(\boldsymbol{x}^k) \text{ is not empty} \\ \left[\sum_{j=1}^{M} [e_j(\boldsymbol{x}^k)]^{-1} \right]^{-1} & \text{if } J(\boldsymbol{x}^k) \text{ is empty} \end{cases} \tag{4-11}$$

$$J = \left\{ j \,\middle|\, e_j(\boldsymbol{x}^k) \geq 0 \right\} \tag{4-12}$$

and α_k are positive multipliers calculated from

$$\alpha_k = \frac{1}{\left| H_1(\boldsymbol{x}^{(0)} + \Delta\boldsymbol{x}^k) \right|} \;, \; k = 1, 2, \ldots, N \tag{4-13}$$

where $\boldsymbol{x}^{(0)}$ is the starting point, for which a good candidate is the optimal nominal solution $\boldsymbol{x}^*$. It is seen that the optimal yield objective function in (4-9) equals the number of failed circuits N_{fail} at the starting point, and provides a continuous approximation to N_{fail} during optimization. If necessary, yield optimization can be restarted with α_k updated with the current solution. We use in this work the highly efficient implementation of yield analysis and optimization available in OSA90/hope™ (1997).

4.3 YIELD ANALYSIS AND OPTIMIZATION USING SPACE MAPPING BASED NEUROMODELS

Bandler, Rayas-Sánchez and Zhang (2001) proposed the use of SM-based neuromodels to perform accurate and efficient yield analysis and optimization of microwave devices. The aim is to combine the computational efficiency of coarse models (typically equivalent circuit models) with the accuracy of fine models (typically EM simulators). We assume that the SM-based neuromodel is already available, obtained either from a modeling process (see Chapter 2) or from an optimization process (see Chapter 3).

Let the vectors $\boldsymbol{x}_c$, $\boldsymbol{x}_f \in \Re^n$ represent the design parameters of the coarse and fine models, respectively. In general, the operating frequency ω, used by the fine model, can be different to that one used by the coarse model, denoted as ω_c. Let $\boldsymbol{R}_c(\boldsymbol{x}_c,\omega_c)$, $\boldsymbol{R}_f(\boldsymbol{x}_f,\omega)$ $\in \Re^r$ represent the coarse and fine model responses at the frequencies ω_c and ω, respectively. We denote the corresponding SM-based neuromodel responses at frequency ω as $\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f,\omega)$, given by

$$\boldsymbol{R}_{SMBN}(\boldsymbol{x}_f,\omega) = \boldsymbol{R}_c(\boldsymbol{x}_c,\omega_c) \tag{4-14}$$

with

$$\begin{bmatrix} \boldsymbol{x}_c \\ \omega_c \end{bmatrix} = \boldsymbol{P}(\boldsymbol{x}_f,\omega) \tag{4-15}$$

where the mapping function $\boldsymbol{P}$ is implemented by a neural network following any of the 5 neuromapping variations (SM, FDSM, FSM, FM or FPSM) described in Chapters 2 and 3. As stated before, we assume that a suitable mapping function $\boldsymbol{P}$ has already been found (i.e., a neural network with suitable complexity has already been trained).

If the SM-based neuromodel is properly developed,

$$\boldsymbol{R}_f(\boldsymbol{x}_f,\omega) \approx \boldsymbol{R}_{SMBN}(\boldsymbol{x}_f,\omega) \tag{4-16}$$

for all $\boldsymbol{x}_f$ and ω in the training region. The Jacobian of the fine model responses w.r.t. the fine model parameters, $\boldsymbol{J}_f \in \Re^{r\times n}$, is defined as

$$\boldsymbol{J}_f = \begin{bmatrix} \dfrac{\partial R_f^1}{\partial x_{f1}} & \cdots & \dfrac{\partial R_f^1}{\partial x_{fn}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial R_f^r}{\partial x_{f1}} & \cdots & \dfrac{\partial R_f^r}{\partial x_{fn}} \end{bmatrix} \tag{4-17}$$

On the other hand, the Jacobian of the coarse model responses w.r.t. the coarse model parameters and mapped frequency, denoted by $\boldsymbol{J}_c \in \Re^{r\times(n+1)}$, is given by

$$\boldsymbol{J}_c = \begin{bmatrix} \dfrac{\partial R_c^1}{\partial x_{c1}} & \cdots & \dfrac{\partial R_c^1}{\partial x_{cn}} & \dfrac{\partial R_c^1}{\partial \omega_c} \\ \vdots & \ddots & \vdots & \vdots \\ \dfrac{\partial R_c^r}{\partial x_{c1}} & \cdots & \dfrac{\partial R_c^r}{\partial x_{cn}} & \dfrac{\partial R_c^r}{\partial \omega_c} \end{bmatrix} \tag{4-18}$$

while the Jacobian of the mapping w.r.t. the fine model parameters, denoted by $\boldsymbol{J}_P \in \Re^{(n+1)\times n}$, is given by

$$\boldsymbol{J}_P = \begin{bmatrix} \dfrac{\partial x_{c1}}{\partial x_{f1}} & \cdots & \dfrac{\partial x_{c1}}{\partial x_{fn}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial x_{cn}}{\partial x_{f1}} & \cdots & \dfrac{\partial x_{cn}}{\partial x_{fn}} \\ \dfrac{\partial \omega_c}{\partial x_{f1}} & \cdots & \dfrac{\partial \omega_c}{\partial x_{fn}} \end{bmatrix} \tag{4-19}$$

From (4-17)-(4-19), the sensitivities of the fine model responses can be approximated using

$$\boldsymbol{J}_f \approx \boldsymbol{J}_c\, \boldsymbol{J}_P \tag{4-20}$$

The accuracy of the approximation of $\boldsymbol{J}_f$ using (4-20) will depend on how well the SM-based neuromodel reproduces the behavior of the fine model in the training region, i.e., it will depend on the accuracy of the approximation (4-16).

(4-20) represents a generalization of the lemma found by Bakr, Bandler, Georgieva and Madsen (1999), where a linear, frequency-insensitive mapping function was assumed. Naturally, (4-20) will be accurate over a larger region since the mapping is nonlinear and frequency-sensitive, which has proved to be a very significant advantage when dealing with coarse models based on quasi-static approximations.

If the mapping is implemented with a 3-layer perceptron with h hidden neurons

(4-15) is given by

$$\boldsymbol{P}(\boldsymbol{x}_f,\omega)=\boldsymbol{W}^o\boldsymbol{\Phi}(\boldsymbol{x}_f,\omega)+\boldsymbol{b}^o \tag{4-21}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}_f,\omega)=[\varphi(s_1) \quad \varphi(s_2) \quad \ldots \quad \varphi(s_h)]^T \tag{4-22}$$

$$\boldsymbol{s}=\boldsymbol{W}^h\begin{bmatrix}\boldsymbol{x}_f\\ \omega\end{bmatrix}+\boldsymbol{b}^h \tag{4-23}$$

where $\boldsymbol{W}^o \in \Re^{(n+1)\times h}$ is the matrix of output weighting factors, $\boldsymbol{b}^o\in \Re^{n+1}$ is the vector of output bias elements, $\boldsymbol{\Phi} \in \Re^h$ is the vector of hidden signals, $\boldsymbol{s} \in \Re^h$ is the vector of activation potentials, $\boldsymbol{W}^h \in \Re^{h\times(n+1)}$ is the matrix of hidden weighting factors, $\boldsymbol{b}^h\in \Re^h$ is the vector of hidden bias elements and h is the number of hidden neurons. A typical choice for the nonlinear activation functions is hyperbolic tangents, i.e., $\varphi(\cdot)$ = tanh(·). All the internal parameters of the neural network, $\boldsymbol{b}^o$, $\boldsymbol{b}^h$, $\boldsymbol{W}^o$ and $\boldsymbol{W}^h$ are constant since the SM-based neuromodel has been already developed.

The Jacobian $\boldsymbol{J}_P$ is obtained from (4-21)-(4-23) as

$$\boldsymbol{J}_P=\boldsymbol{W}^o\boldsymbol{J}_{\Phi}\boldsymbol{W}^h \tag{4-24}$$

where $\boldsymbol{J}_{\Phi}\in \Re^{h\times h}$ is a diagonal matrix given by $\boldsymbol{J}_{\Phi}$ = diag($\varphi'(s_j)$), with j = 1… h.

If the SM-based neuromodel uses a 2-layer perceptron, the Jacobian $\boldsymbol{J}_P$ is simply

$$\boldsymbol{J}_P=\boldsymbol{W}^o \tag{4-25}$$

which corresponds to the case of a frequency-sensitive linear mapping. Notice that by substituting (4-25) in (4-20) and assuming a frequency-insensitive neuromapping we obtain the lemma found in the work by Bakr, Bandler, Georgieva and Madsen (1999), since in the case of a 2-layer perceptron with no frequency dependence, $\boldsymbol{W}^o \in \Re^{n\times n}$.

4.4 EXAMPLE

Consider a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter, whose physical structure is illustrated in Fig. 2.21. The description of its physical structure, the design specifications, as well as the fine and coarse model descriptions can be found in Section 3.9.1.

4.4.1 Yield Analysis and Optimization Assuming Symmetry

The SM-based neuromodel of the HTS filter obtained in Chapter 2 is used to perform yield analysis and optimization. This model was obtained assuming that the design parameters are $\boldsymbol{x}_f = [L_1\ L_2\ L_3\ S_1\ S_2\ S_3]^T$, and taking L_0 = 50 mil, H = 20 mil, W = 7 mil, ε_r = 23.425, loss tangent = 3×10^{-5}; the metalization was considered lossless. The corresponding SM-based neuromodel is illustrated in Fig. 4.1, which implements a frequency partial-space mapped neuromapping with 7 hidden neurons, mapping only L_1, S_1 and the frequency (3LP:7-7-3). L_{1c} and S_{1c} in Fig. 4.1 denote the corresponding two physical dimensions as used by the coarse model, i.e., after being transformed by the mapping. Notice from Fig. 2.21 that it is assumed that the physical structure of the HTS filter posses vertical and horizontal geometrical symmetry.

Applying direct minimax optimization to the coarse model, we obtain the optimal coarse solution $\boldsymbol{x}_c^* = [188.33\ \ 197.98\ \ 188.58\ 21.97\ \ 99.12\ \ 111.67]^T$ (mils). The coarse model response at $\boldsymbol{x}_c^*$ is shown in Fig. 4.2. The fine model response at the optimal coarse solution is shown in Fig. 4.3 using a fine frequency sweep.

We apply direct minimax optimization to the SM-based neuromodel, taking $\boldsymbol{x}_c^*$ as the starting point, to obtain the optimal SM-based neuromodel nominal solution $\boldsymbol{x}_{SMBN}^*$

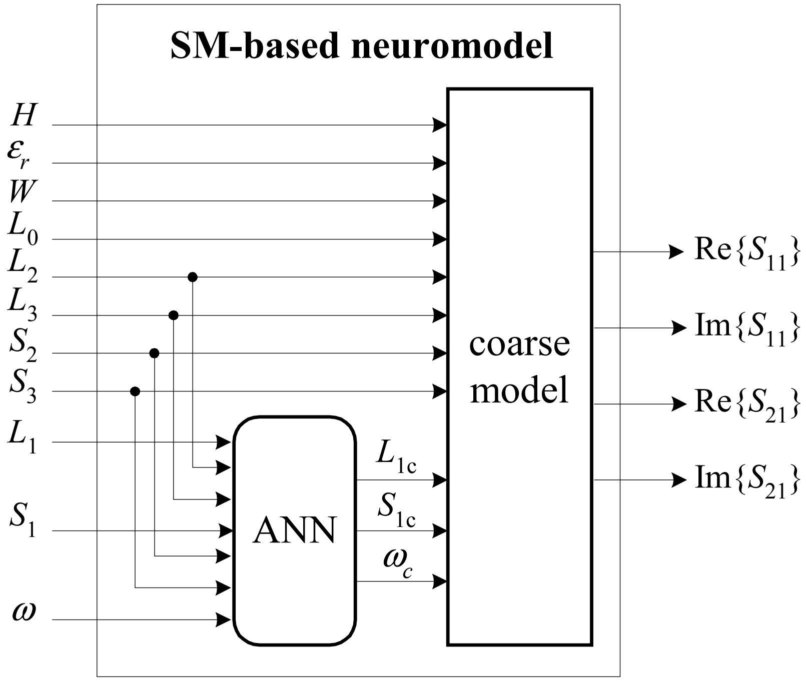


Fig. 4.1 SM-based neuromodel of the HTS filter for yield analysis assuming symmetry (L_{1c} and S_{1c} correspond to L_1 and S_1 as used by the coarse model).

= [185.79 194.23 184.91 21.05 82.31 89.32]^T (mils). Fig. 4.4 shows excellent agreement between the SM-based neuromodel response and the fine model response at $\boldsymbol{x}_{SMBN}^*$.

To realize yield analysis, we consider 0.2% of variation for the dielectric constant and for the loss tangent, as well as 75 micron of variation for the physical dimensions, as suggested by Mansour (2000), with uniform statistical distributions. These tolerances are larger than other typical manufacturing tolerances reported in the literature (e.g., see Burrascano and Mongiardo, 1999).

We perform Monte Carlo yield analysis of the SM-based neuromodel around $\boldsymbol{x}_{SMBN}^*$ with 500 outcomes using OSA90/hope™ (1997). The responses for 50 of those

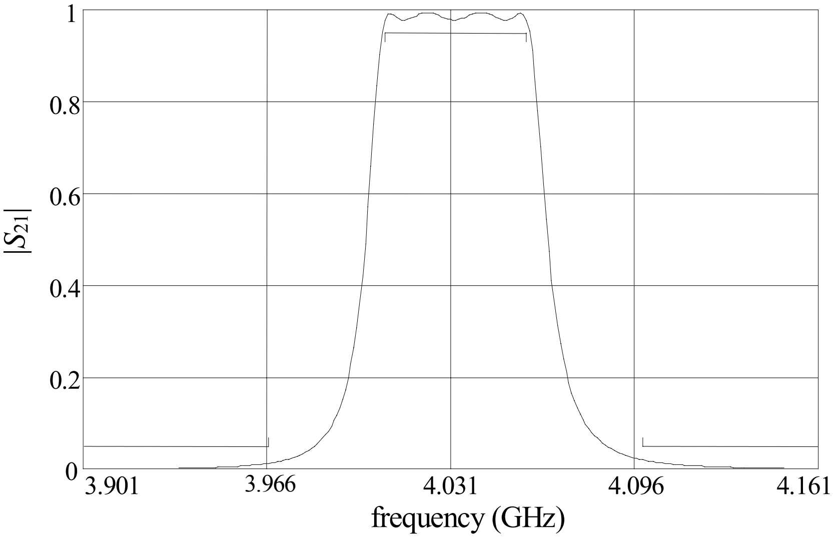


Fig. 4.2 Optimal coarse model response for the HTS filter.

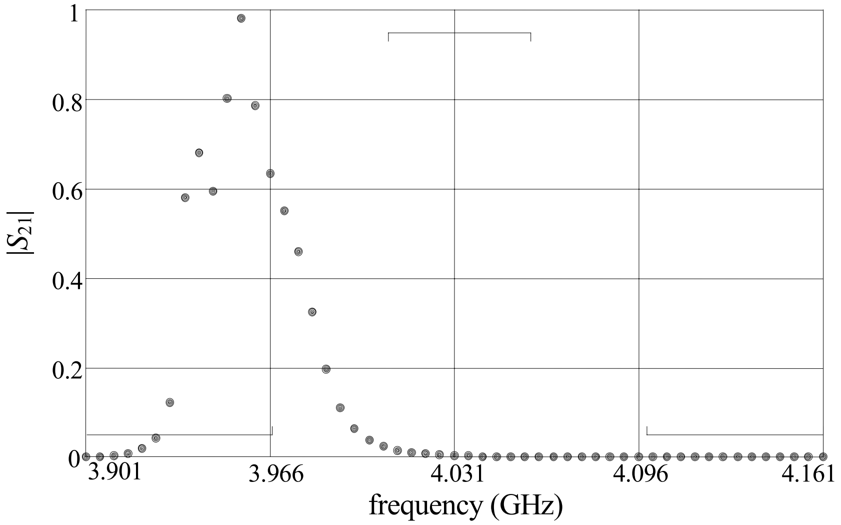


Fig. 4.3 HTS filter fine model response at the optimal coarse solution.

outcomes are shown in Fig. 4.5. The yield calculation is shown in Fig. 4.6. A yield of only 18.4% is obtained at $\boldsymbol{x}_{SMBN}^{*}$, which is reasonable considering the well-known high sensitivity of this microstrip circuit.

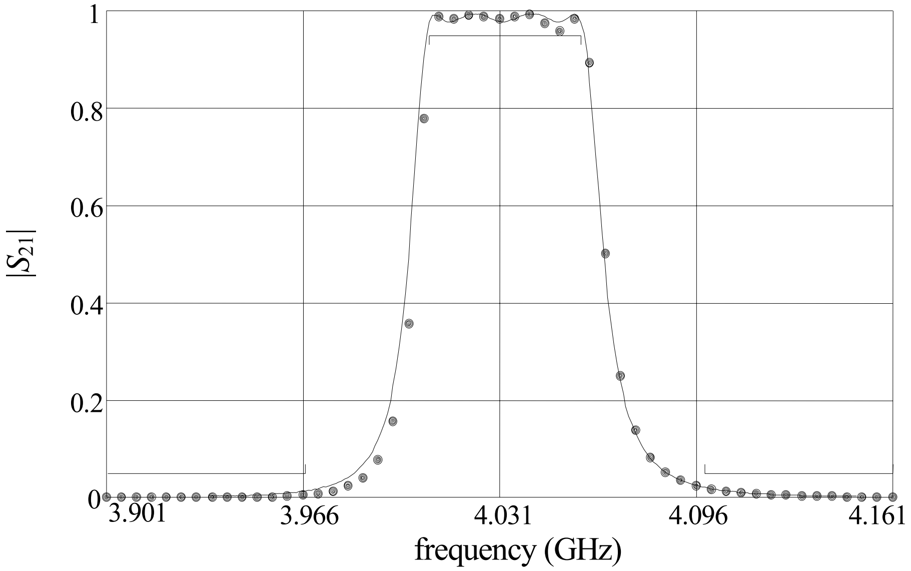


Fig. 4.4 HTS filter fine model response and SM-based neuromodel response at the optimal nominal solution $\boldsymbol{x}_{SMBN}{}^{*}$.

Performing yield analysis using 500 outcomes with the SM-based neuromodel of the HTS filter takes a few tens of seconds on a conventional computer (PC AMD 640MHz, 256M RAM, on Windows NT 4.0), while a single outcome calculation for the same circuit using an EM simulation takes around 5 hours on the same computer. The SM-based neuromodel makes feasible the EM-based yield analysis of this complex microwave structure.

We then apply yield optimization to the SM-based neuromodel with 500 outcomes using the Yield-Huber optimizer available in OSA90/hope™ (1997), obtaining the following optimal yield solution: $\boldsymbol{x}_{SMBN}{}^{Y*}$ = [183.04 196.91 182.22 20.04 77.67 83.09]^{T} (mils). The corresponding responses for 50 of those outcomes are shown in Fig. 4.7. The yield is increased from 18.4% to 66%, as shown in Fig. 4.8. Once again, an excellent agreement is observed between the fine model response and the SM-based

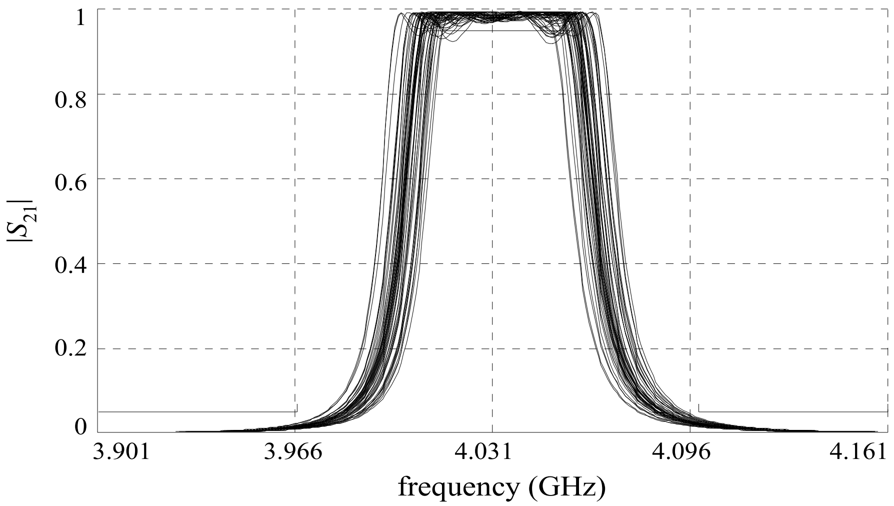


Fig. 4.5 Monte Carlo yield analysis of the HTS SM-based neuromodel responses around the optimal nominal solution $\boldsymbol{x}_{SMBN}^*$ with 50 outcomes.

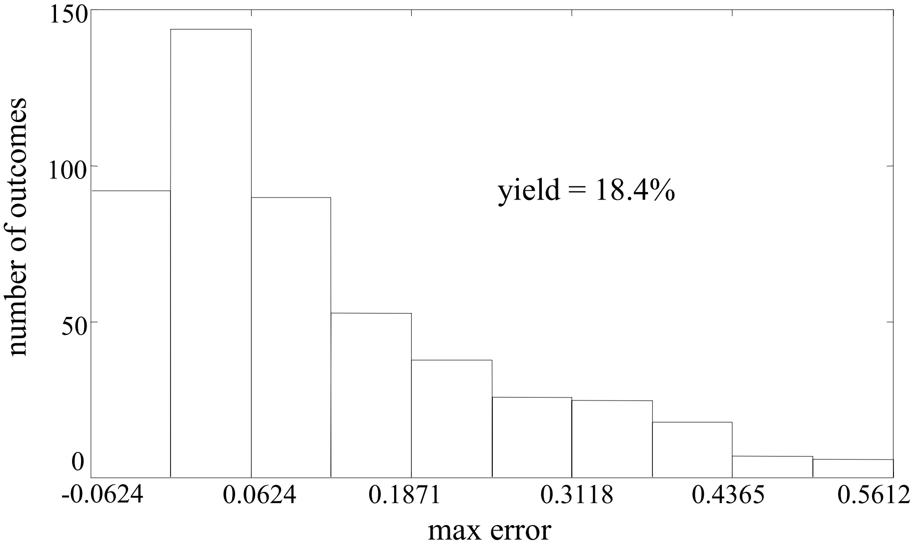


Fig. 4.6 Histogram of the yield analysis of the SM-based neuromodel around the optimal nominal solution $\boldsymbol{x}_{SMBN}^*$ with 500 outcomes.

neuromodel response at the optimal yield solution $\boldsymbol{x}_{SMBN}^{Y*}$ (see Fig. 4.9).

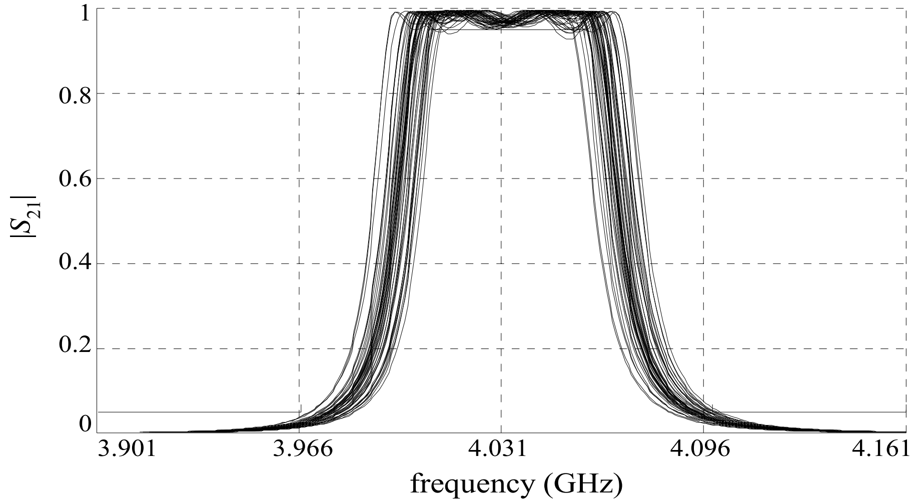


Fig. 4.7 Monte Carlo yield analysis of the SM-based neuromodel responses around the optimal yield solution $\boldsymbol{x}_{SMBN}{}^{Y*}$ with 50 outcomes.

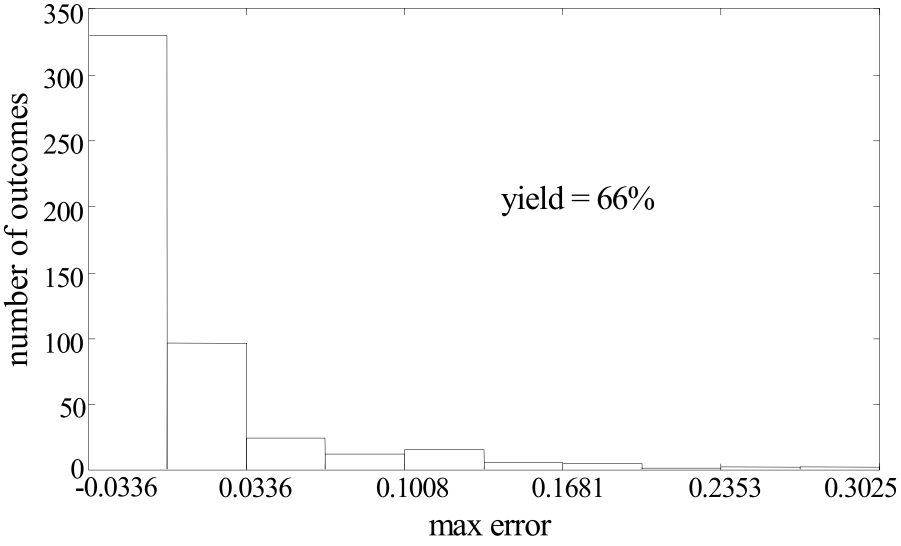


Fig. 4.8 Histogram of the yield analysis of the SM-based neuromodel around the optimal yield solution $\boldsymbol{x}_{SMBN}{}^{Y*}$ with 500 outcomes (considering symmetry).

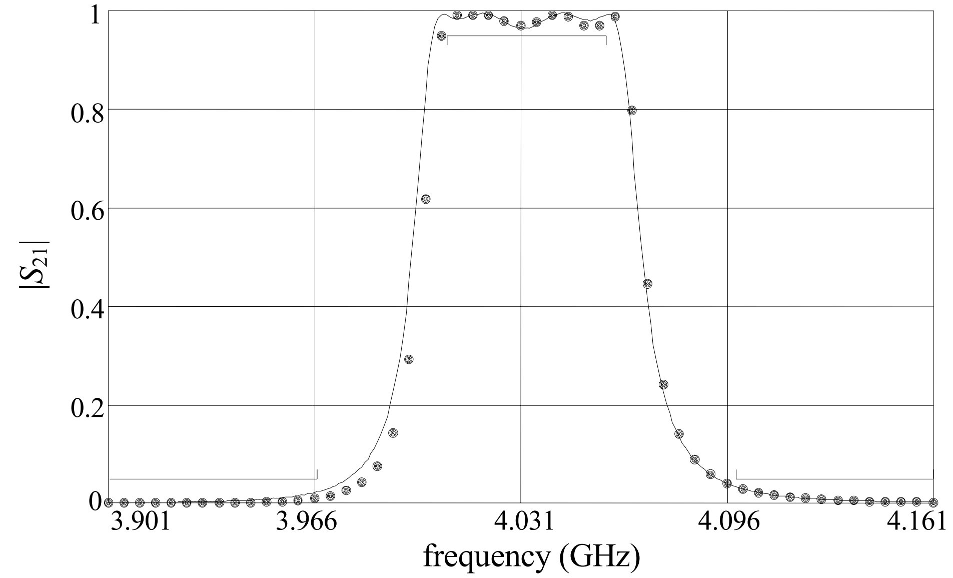


Fig. 4.9 Fine model response and SM-based neuromodel response for the HTS filter at the optimal yield solution $\boldsymbol{x}_{SMBN}^{Y*}$.

4.4.2 Considering Asymmetric Variations due to Tolerances

It is clear that our SM-based neuromodel assumes that the random variations in the physical design parameters due to the tolerances are symmetric (See Fig. 2.21 and Fig. 4.1).

In order to make a more realistic statistical analysis of the HTS filter, we consider that all the lengths and separations in the structure are asymmetric, as illustrated in Fig. 4.10.

Developing a new SM-based neuromodel for this asymmetric structure would be very time consuming, since the dimensionality of the problem becomes very large, and many fine model training points would be needed. We propose the strategy illustrated in

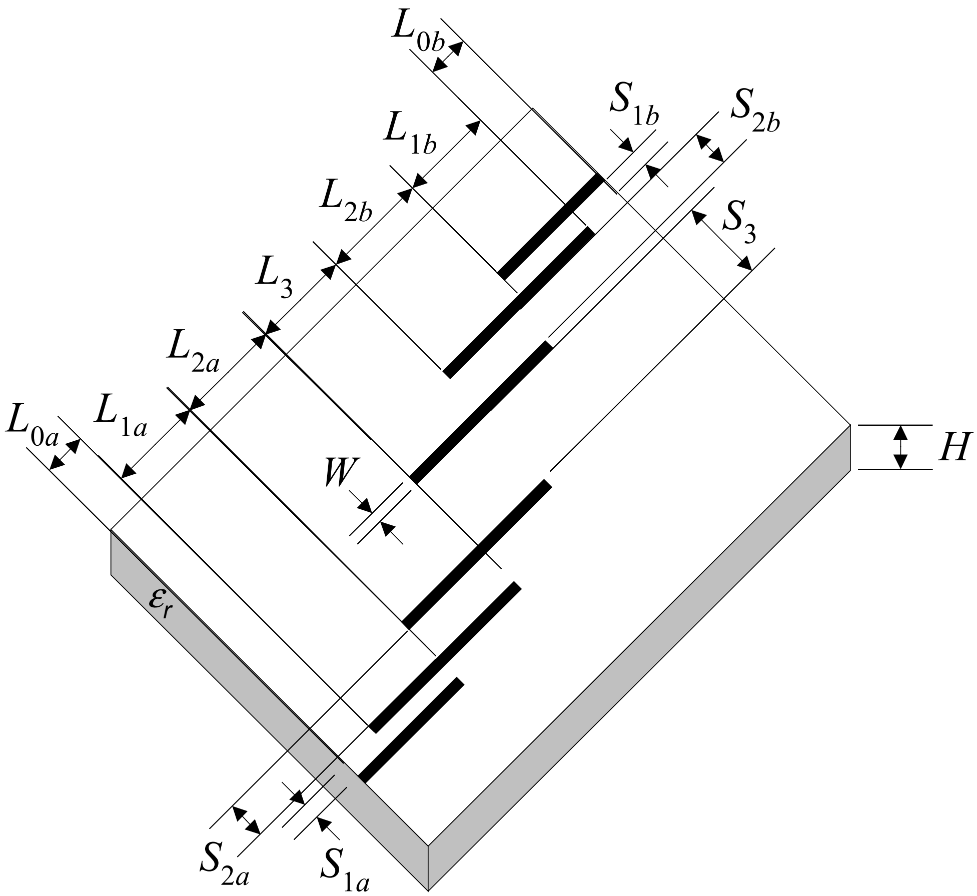


Fig. 4.10 Physical structure of the HTS filter considering asymmetry.

Fig. 4.11 instead. In this approach, we re-use the available neuromapping to take into account asymmetric random variations in the physical parameters due to their tolerances, taking advantage of the asymmetric nature of the coarse model (compare Fig. 4.1 and Fig. 4.11).

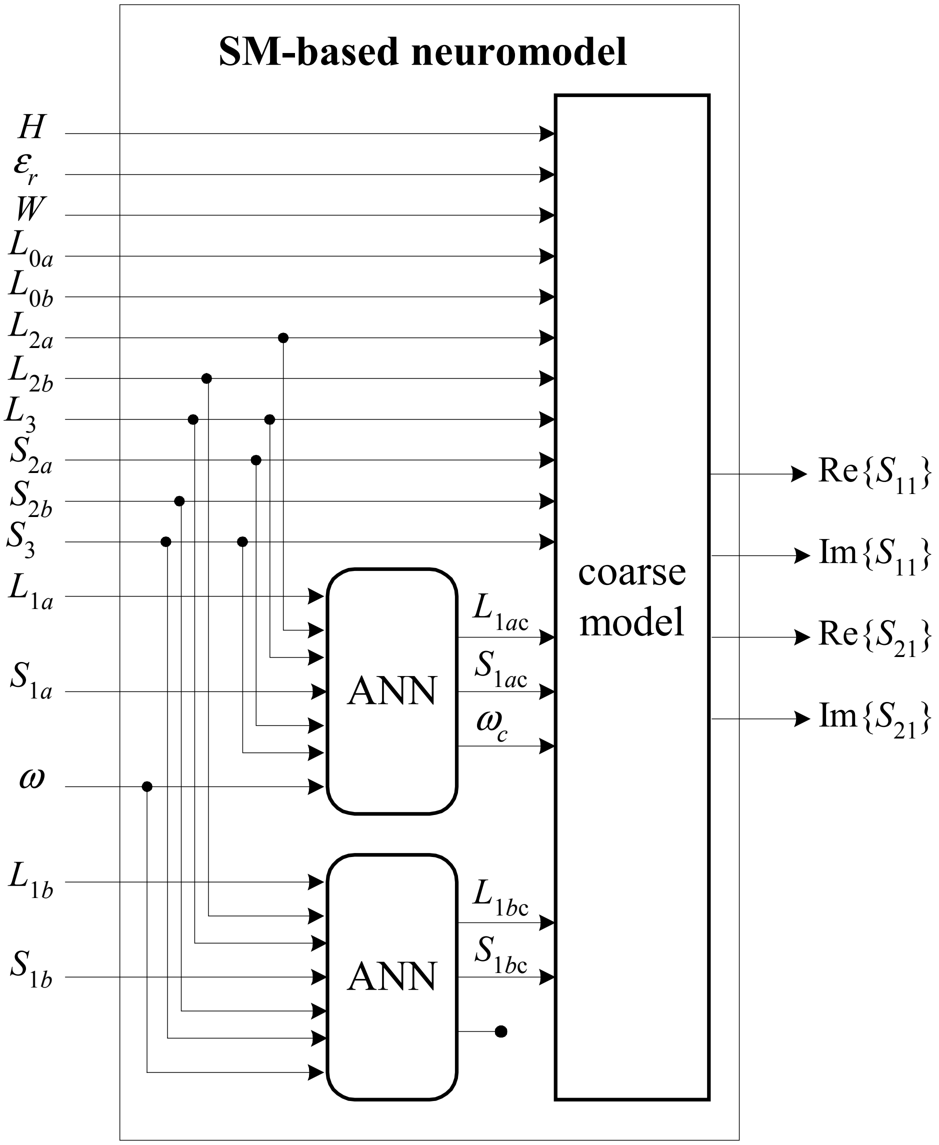


Fig. 4.11 SM-based neuromodel of the HTS filter with asymmetric tolerances in the physical parameters (L_{1ac} and S_{1ac} represent the corresponding length and separation for the coarse model components in the lower-left side of the structure -see Fig. 4.10- while L_{1bc} and S_{1bc} represent the corresponding dimensions for the upper-right section).

L_{1ac} and S_{1ac} in Fig. 4.11 now represent the corresponding length and separation for the coarse model components in the lower-left side of the structure, while L_{1bc} and S_{1bc}

represent the corresponding dimensions for the upper-right section (see Fig. 3.8). Notice also that assigning a separate neuromapping to each of these sections makes physical sense, since the electromagnetic interaction between the microstrip lines in either the lower-left or upper-right sections of the structure is much larger than that one between the left-right or lower-upper microstrip lines.

Re-using the available neuromapping as described avoids the need for extra fine model evaluations. Taking into account the excellent generalization performance of our SM-based neuromodel, this approach should provide a good approximation to the yield considering that the tolerances are small.

We perform Monte Carlo yield analysis of the asymmetric SM-based neuromodel around the optimal nominal solution $\boldsymbol{x}_{SMBN}{}^{*}$ with 500 outcomes. The corresponding responses for 50 of those outcomes are shown in Fig. 4.12. The histogram of the yield at the optimal nominal solution $\boldsymbol{x}_{SMBN}{}^{*}$ with 500 outcomes is illustrated in Fig. 4.13. A yield of only 14% was obtained for the asymmetric structure.

We then perform Monte Carlo yield analysis of the asymmetric SM-based neuromodel around the optimal yield solution $\boldsymbol{x}_{SMBN}{}^{Y*}$ with 500 outcomes; 50 of those outcomes are illustrated in Fig. 4.14. Notice that the optimal yield design is kept symmetric all the time. The yield obtained for the asymmetric structure around the optimal yield solution $\boldsymbol{x}_{SMBN}{}^{Y*}$ is 68.8%, as illustrated in Fig. 4.15.

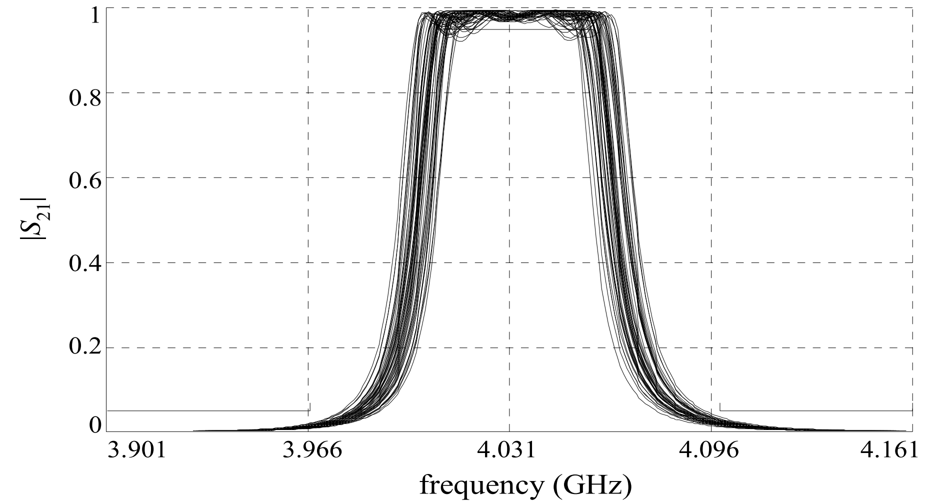


Fig. 4.12 Monte Carlo yield analysis of the SM-based neuromodel responses, considering asymmetry, around the optimal nominal solution $\boldsymbol{x}_{SMBN}{}^{*}$ with 50 outcomes.

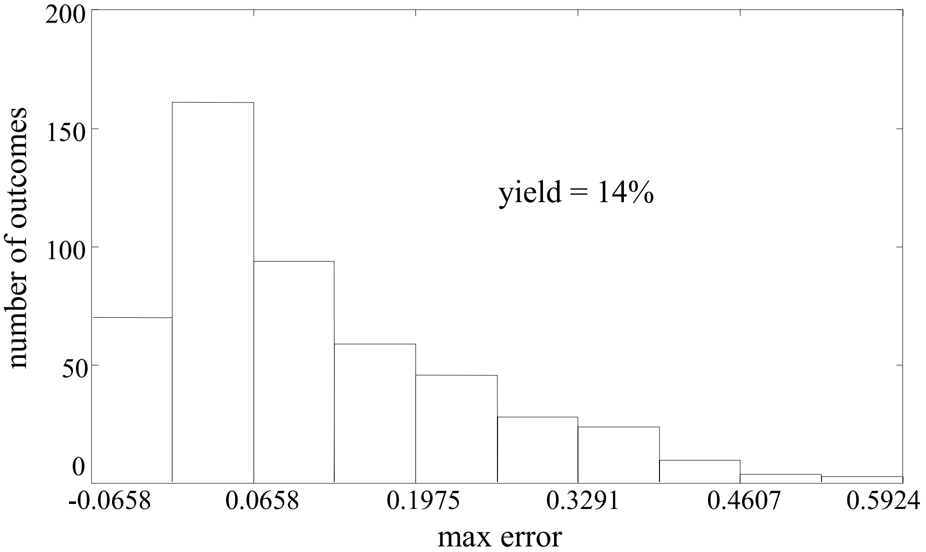


Fig. 4.13 Histogram of the yield analysis of the SM-based neuromodel around the optimal yield solution $\boldsymbol{x}_{SMBN}{}^{Y*}$ with 500 outcomes (considering symmetry).

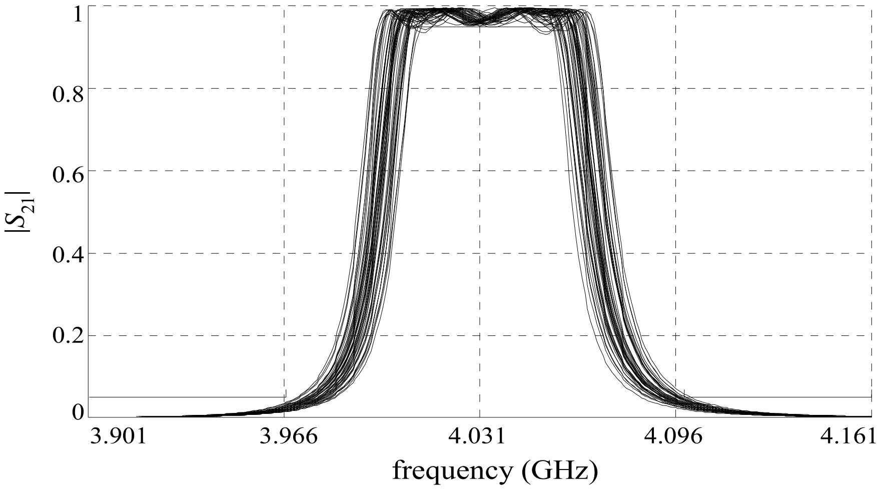


Fig. 4.14 Monte Carlo yield analysis of the SM-based neuromodel responses, considering asymmetry, around the optimal nominal solution $\boldsymbol{x}_{SMBN}^*$ with 50 outcomes.

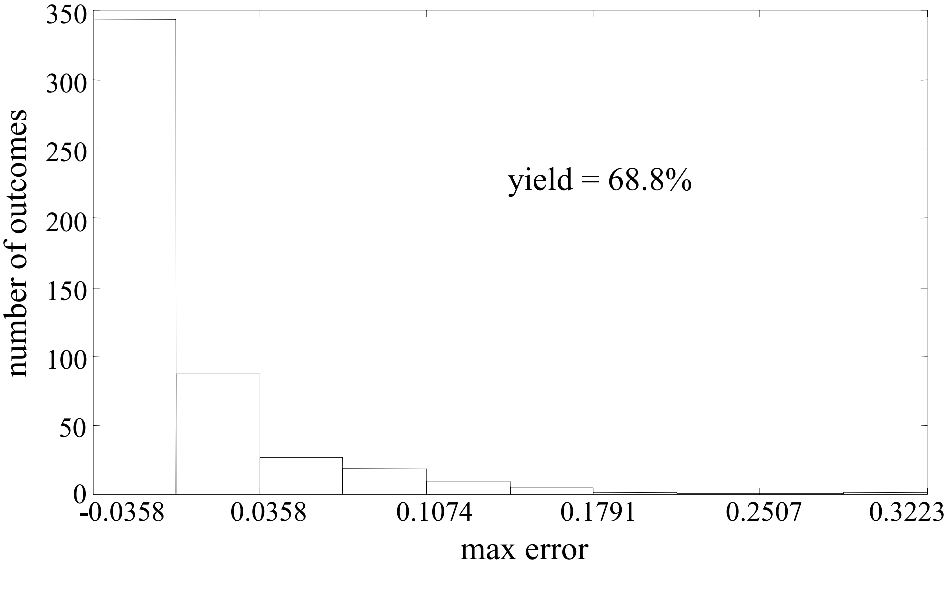


Fig. 4.15 Histogram of the yield analysis of the asymmetric SM-based neuromodel around the optimal yield solution $\boldsymbol{x}_{SMBN}^{Y*}$ with 500 outcomes.

4.5 CONCLUDING REMARKS

We have described in this chapter an efficient procedure to realize electromagnetics-based statistical analysis and yield optimization of microwave structures using space mapping-based neuromodels. This follows the work by Bandler, Rayas-Sánchez and Zhang (2001a,b)

We mathematically formulate the problem of statistical analysis and yield optimization using SM-based neuromodels.

A formulation for the relationship between the fine and coarse model sensitivities through a nonlinear, frequency-sensitive neuromapping is found. This formulation represents a generalization of the lemma found in the work by Bakr, Bandler, Georgieva and Madsen (1999).

We describe a creative way to avoid the need of extra EM simulations to take into account asymmetric variations in the physical parameters due to tolerances by re-using the available neuromappings and exploiting the asymmetric nature of the coarse models in a given SM-based neuromodel

We illustrate our techniques by the yield analysis and optimization of a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter. The yield is increased from 14% to 68.8% for this complex structure. Excellent agreement between the EM responses and the SM-based neuromodel responses is found at both, the optimal nominal solution and the optimal yield solution.

Chapter 5

NEURAL INVERSE SPACE MAPPING (NISM) OPTIMIZATION

5.1 INTRODUCTION

An elegant new algorithm for EM-based design of microwave circuits is described in this chapter: Neural Inverse Space Mapping (NISM) optimization. This is the first Space Mapping (SM) algorithm that explicitly makes use of the inverse of the mapping from the fine to the coarse model parameter spaces.

NISM optimization follows an aggressive formulation by not requiring a number of up-front fine model evaluations to start approximating the mapping.

An innovative yet simple procedure for statistical parameter extraction avoids the need for multipoint matching and frequency mappings.

A neural network whose generalization performance is controlled through a network growing strategy approximates the inverse of the mapping at each iteration. The NISM step consists simply of evaluating the current neural network at the optimal coarse solution. We prove that this step is equivalent to a quasi-Newton step while the inverse mapping remains essentially linear, and gradually departs from a quasi-Newton step as the amount of nonlinearity in the inverse mapping increases.

We contrast our new algorithm with Neural Space Mapping (NSM) optimization, described in Chapter 3, as well as with the Trust Region Aggressive Space Mapping exploiting Surrogates, developed by Bakr, Bandler, Madsen, Rayas-Sánchez and Søndergaard (2000).

NISM optimization was proposed for the first time by Bandler, Ismail, Rayas-Sánchez and Zhang (2001).

5.2 AN OVERVIEW ON NISM OPTIMIZATION

5.2.1 Notation

Let the vectors $\boldsymbol{x}_c$ and $\boldsymbol{x}_f$ represent the design parameters of the coarse and fine models, respectively ($\boldsymbol{x}_c$, $\boldsymbol{x}_f \in \Re^n$). We denote the optimizable fine model responses at point $\boldsymbol{x}_f$ and frequency ω by $\boldsymbol{R}_f(\boldsymbol{x}_f, \omega) \in \Re^r$ where r is the number of responses to be optimized. For example, if the responses to be optimized are $|S_{11}|$ and $|S_{21}|$, then $r = 2$. The vector $\boldsymbol{R}_f(\boldsymbol{x}_f) \in \Re^m$ denotes the fine model responses at the F_p sample frequency points, where $m = rF_p$. Similarly, $\boldsymbol{R}_c(\boldsymbol{x}_c, \omega) \in \Re^r$ contains the r coarse model responses at point $\boldsymbol{x}_c$ and frequency ω, while $\boldsymbol{R}_c(\boldsymbol{x}_c) \in \Re^m$ denotes the coarse model responses at the F_p frequency points, to be optimized.

Additionally, we denote the characterizing fine model responses at point $\boldsymbol{x}_f \in \Re^n$ and frequency ω by $\boldsymbol{R}_{fs}(\boldsymbol{x}_f, \omega) \in \Re^R$, which includes the real and imaginary parts of all the available characterizing responses in the model (considering symmetry). For example, for a 2-port reciprocal network they include Re$\{S_{11}\}$, Im$\{S_{11}\}$, Re$\{S_{21}\}$ and Im$\{S_{21}\}$, and therefore $R = 4$. The vector $\boldsymbol{R}_{fs}(\boldsymbol{x}_f) \in \Re^M$ denotes the characterizing fine model responses at all the F_p frequency points, where $M = RF_p$. Similarly, $\boldsymbol{R}_{cs}(\boldsymbol{x}_c) \in \Re^M$ denotes the corresponding characterizing coarse model responses at all the F_p frequency points.

5.2.2 Flow Diagram

A flow diagram for NISM optimization is shown in Fig. 5.1. We start by performing regular minimax optimization on the coarse model to find the optimal coarse solution $\boldsymbol{x}_c^*$ that yields the desired response. The characterizing fine model responses $\boldsymbol{R}_{fs}$ at the optimal coarse solution $\boldsymbol{x}_c^*$ are then calculated.

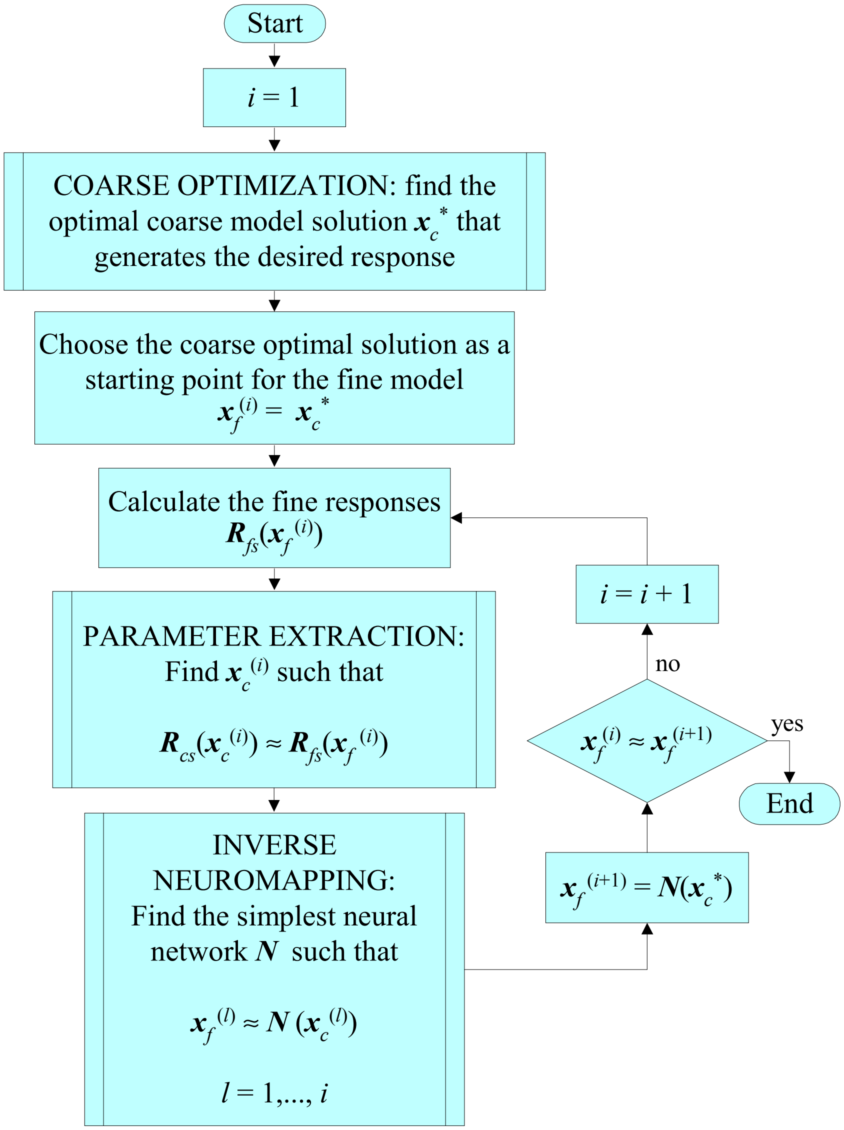

Start
$i = 1$
COARSE OPTIMIZATION: find the optimal coarse model solution $\boldsymbol{x}_c^*$ that generates the desired response
Choose the coarse optimal solution as a starting point for the fine model $\boldsymbol{x}_f^{(i)} = \boldsymbol{x}_c^*$
Calculate the fine responses $\boldsymbol{R}_{fs}(\boldsymbol{x}_f^{(i)})$
PARAMETER EXTRACTION: Find $\boldsymbol{x}_c^{(i)}$ such that $\boldsymbol{R}_{cs}(\boldsymbol{x}_c^{(i)}) \approx \boldsymbol{R}_{fs}(\boldsymbol{x}_f^{(i)})$
INVERSE NEUROMAPPING: Find the simplest neural network $\boldsymbol{N}$ such that $\boldsymbol{x}_f^{(l)} \approx \boldsymbol{N}(\boldsymbol{x}_c^{(l)})$ $l = 1,\ldots, i$
$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{N}(\boldsymbol{x}_c^*)$
$\boldsymbol{x}_f^{(i)} \approx \boldsymbol{x}_f^{(i+1)}$
no
$i = i + 1$
yes
End

Fig. 5.1 Flow diagram for Neural Inverse Space Mapping (NISM) optimization.

We realize parameter extraction, which consists of finding the coarse model parameters that makes the characterizing coarse responses $\boldsymbol{R}_{cs}$ as close as possible to the previously calculated $\boldsymbol{R}_{fs}$.

We continue by training the simplest neural network $\boldsymbol{N}$ that approximates the inverse of the mapping from the fine to the coarse parameter space at the available points.

The new point in the fine model parameter space is then calculated by simply evaluating the neural network at the optimal coarse solution. If the maximum relative change in the fine model parameters is smaller than a previously defined amount we finish, otherwise we calculate the characterizing fine model responses at the new point and continue with the algorithm.

The main operational blocks of the flow diagram in Fig. 5.1 (parameter extraction and inverse neuromapping) are described in detail in the following sections.

5.3 PARAMETER EXTRACTION

The parameter extraction procedure at the *i*th NISM iteration is formulated as the following optimization problem

$$\boldsymbol{x}_c^{(i)} = \arg \min_{\boldsymbol{x}_c} \; U_{PE}(\boldsymbol{x}_c) \tag{5-1}$$

$$U_{PE}(\boldsymbol{x}_c) = \| \boldsymbol{e}(\boldsymbol{x}_c) \|_2^2 \tag{5-2}$$

$$\boldsymbol{e}(\boldsymbol{x}_c) = \boldsymbol{R}_{fs}(\boldsymbol{x}_f^{(i)}) - \boldsymbol{R}_{cs}(\boldsymbol{x}_c) \tag{5-3}$$

We solve (5-1) using the Levenberg-Marquardt algorithm for nonlinear curve fitting available in the Matlab™ Optimization Toolbox (1999).

We normally use $\boldsymbol{x}_c^*$ as the starting point for solving (5-1). This might not be a good starting point when an extremely severe matching problem is being solved, one that has some poor local minimum around $\boldsymbol{x}_c^*$. If the algorithm is trapped in a poor local minimum, we change the starting point for (5-1) by taking a small random perturbation $\boldsymbol{\Delta x}$ around $\boldsymbol{x}_c^*$ until we find an acceptable local minimum, i.e., until we obtain a good matching between both fine and coarse models.

The maximum perturbation $\Delta_{\max}$ is obtained from the maximum absolute sensitivity of the parameter extraction objective function at $\boldsymbol{x}_c^*$ as follows

$$\Delta_{\max} = \frac{\delta_{PE}}{\left\| \nabla U_{PE}(\boldsymbol{x}_c^*) \right\|_\infty} \tag{5-4}$$

Let $\boldsymbol{rand} \in \Re^n$ be a vector whose elements take random values between 0 and +1 every time it is evaluated. The values of the elements of $\boldsymbol{\Delta x}$ are calculated as

$$\Delta x_k = \Delta_{\max}(2rand_k - 1)\,, \quad k = 1, \ldots , n \tag{5-5}$$

A value of $\delta_{PE} = 0.03$ is used in our implementation. Many other values of δ_{PE} could be used in (5-4), since we use it only to escape from a poor local minimum.

A similar strategy for statistical parameter extraction was proposed by Bandler, Biernacki, Chen and Omeragić (1997), where an exploration region is first created by predefining a fixed number of starting points around $\boldsymbol{x}_c^*$.

The proposed algorithm for realizing parameter extraction is stated as follows

```
Algorithm: Parameter Extraction
Begin
    solve (5-1) using $\boldsymbol{x}_c^*$ as starting point
    while $\|\boldsymbol{e}(\boldsymbol{x}_c^{(i)})\|_\infty > \varepsilon_{PE}$
        calculate $\boldsymbol{\Delta x}$ using (5-4) and (5-5)
        solve (5-1) using $\boldsymbol{x}_c^*+\boldsymbol{\Delta x}$ as starting point
end
```

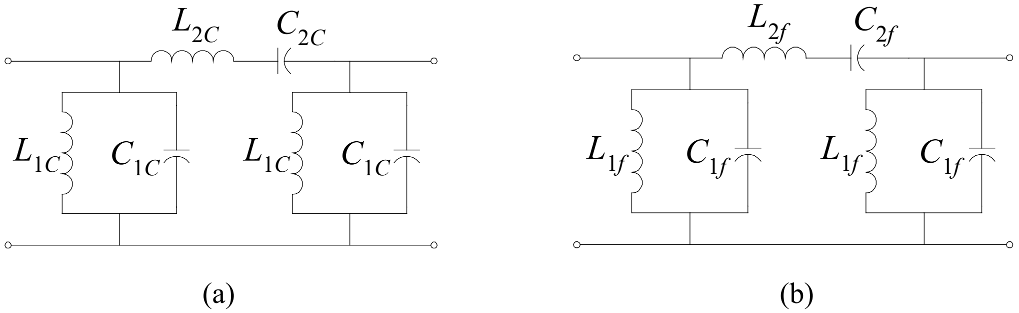
A value of $\varepsilon_{PE} = 0.15$ is used in our implementation, assuming that all the response values are normalized.

5.3.1 Illustration of the Statistical Parameter Extraction Procedure

To illustrate the benefits of using the parameter extraction algorithm described in the previous section, consider the problem of matching the responses of two simple bandpass lumped filters, illustrated in Fig. 5.2. To make the argument clearer, we will assume that both filters have only one optimization variable or design parameter.

Both the coarse and "fine" models consist of canonical sixth-order band pass filters. The coarse model has L_{1c} = 0.0997 nH, L_{2c} = 17.455 nH, $C_{1c} = x_c$ and C_{2c} = 0.058048 pF, being x_c its design parameter. The "fine" model has $L_{1f} = L_{1c}$ + 0.0001 nH, $L_{2f} = L_{2c}$ + 0.017 pH, $C_{1f} = x_f + C_p$ and $C_{2f} = C_{2c}$ + 0.00006 pF, where x_f is its design parameter and C_p is a shifting parameter that will be used to control the degree of deviation between both models.

We take x_c^* = 10.1624 pF. In parameter extraction we want to find x_c such that $\boldsymbol{R}_c(x_c) = \boldsymbol{R}_f(x_c^*)$, where $\boldsymbol{R}_c$ and $\boldsymbol{R}_f$ contain the magnitude of S_{21} for each model, at all the frequency points of interest. The characterizing responses for this problem include the



(a) (b)

Fig. 5.2 Sixth-order band pass lumped filters to illustrate the proposed parameter extraction procedure: (a) coarse model and (b) "fine" model.

real and the imaginary parts of S_{11} and S_{21}.

Fig. 5.3 illustrates the coarse and "fine" model responses at x_c^* when a value of C_p = 6 pF is used in the fine model. Fig. 5.4a shows the objective function (5-2) as a function of x_c, the starting point x_c^*, and the solution found if the statistical procedure is not implemented (i.e., without perturbing the starting point). Clearly, the conventional parameter extraction procedure is trapped in a poor local minimum, and the matching between both models is completely erroneous (see Fig. Fig. 5.4b).

Fig. 5.5 shows the results when the proposed algorithm for parameter extraction is used. The poor local minima are avoided by randomly perturbing the starting point, and excellent match is achieved.

We repeated the experiment with an even more severe misalignment, taking C_p = 10 pF. The corresponding results are shown in Fig. 5.6 and Fig. 5.7. Once again, the

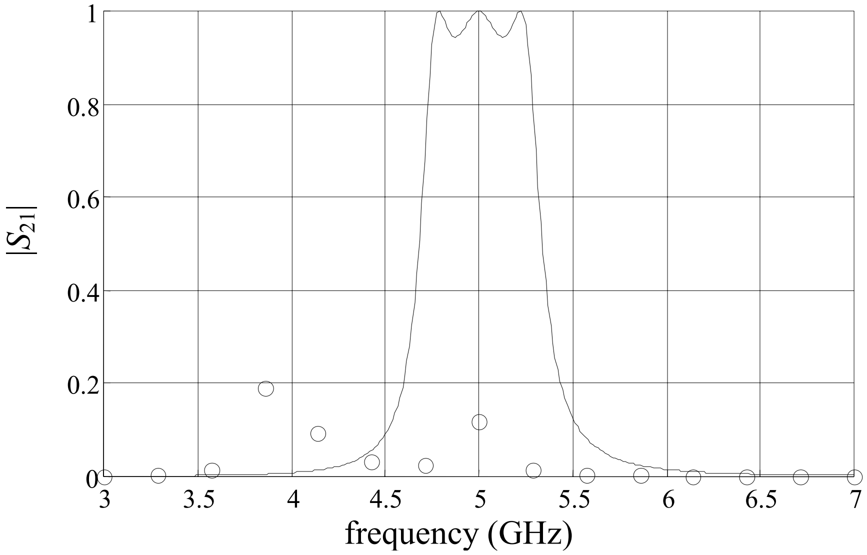
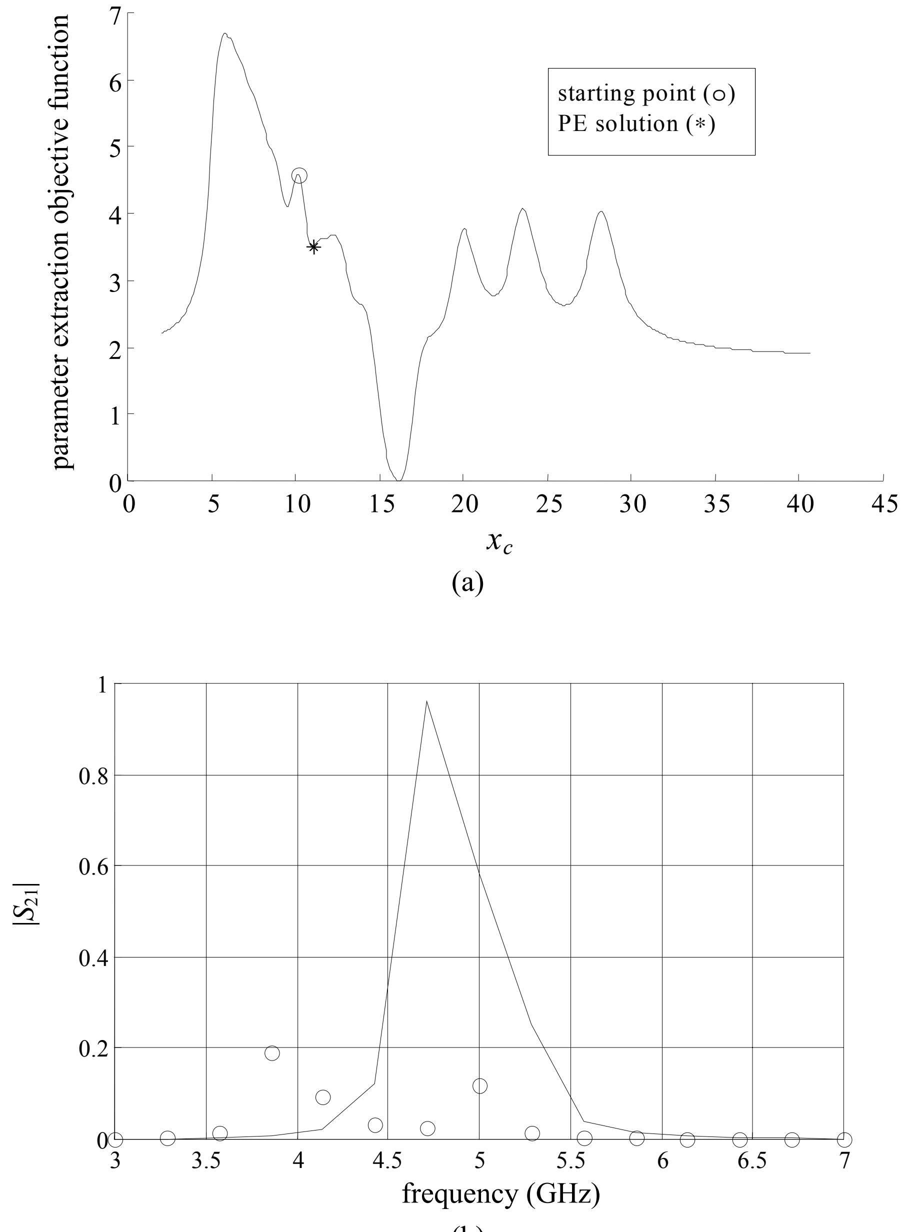


Fig. 5.3 Coarse (–) and fine (○) model responses of the band pass lumped filters at the optimal coarse solution x_c^*.

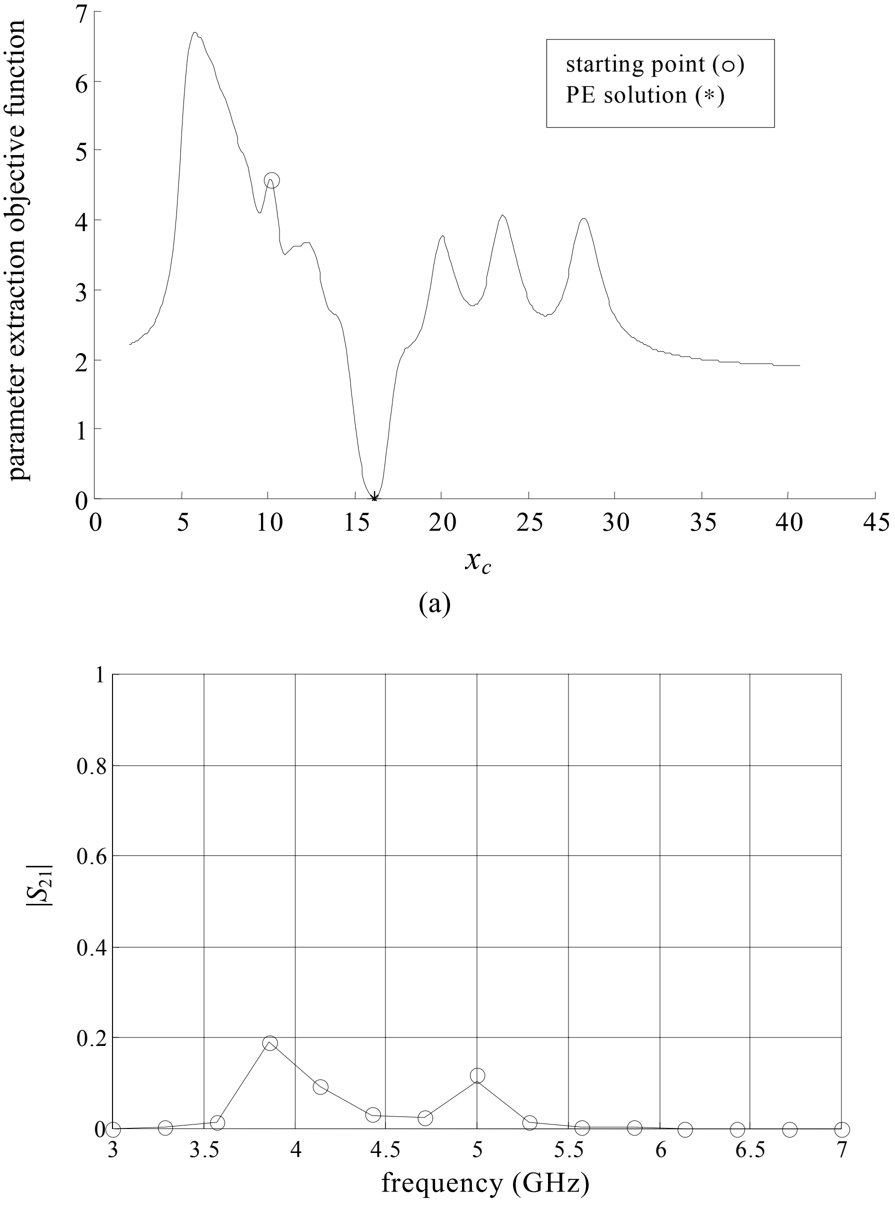


(a)

(b)

Fig. 5.4 Conventional parameter extraction process: (a) objective function, (b) coarse (−) and fine (○) model responses after parameter extraction (C_p = 6 pF).

proposed parameter extraction algorithm avoids poor matching.

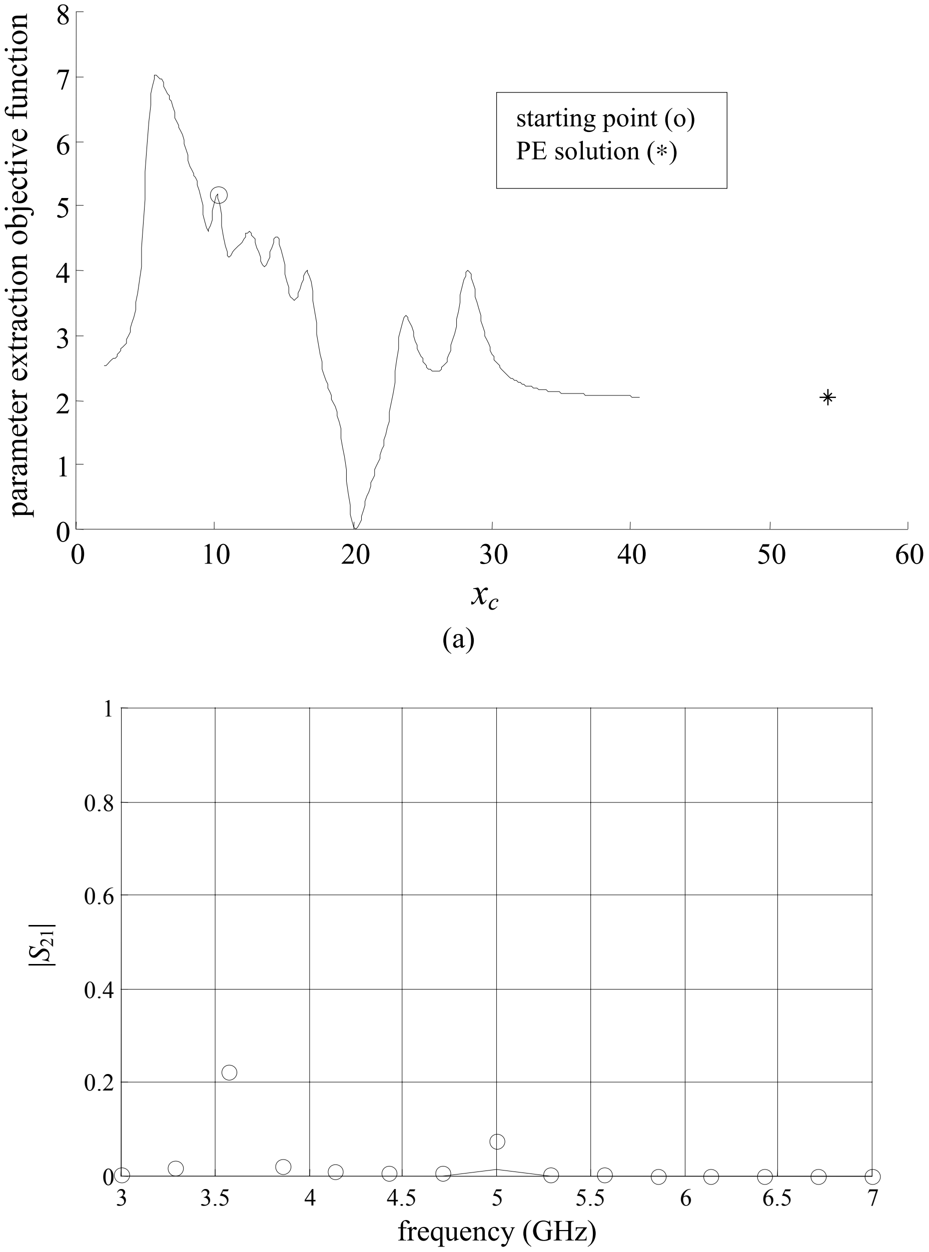


(a)

(b)

Fig. 5.5 Proposed parameter extraction process: (a) objective function, (b) coarse (–) and fine (○) model responses after parameter extraction (C_p = 6 pF).

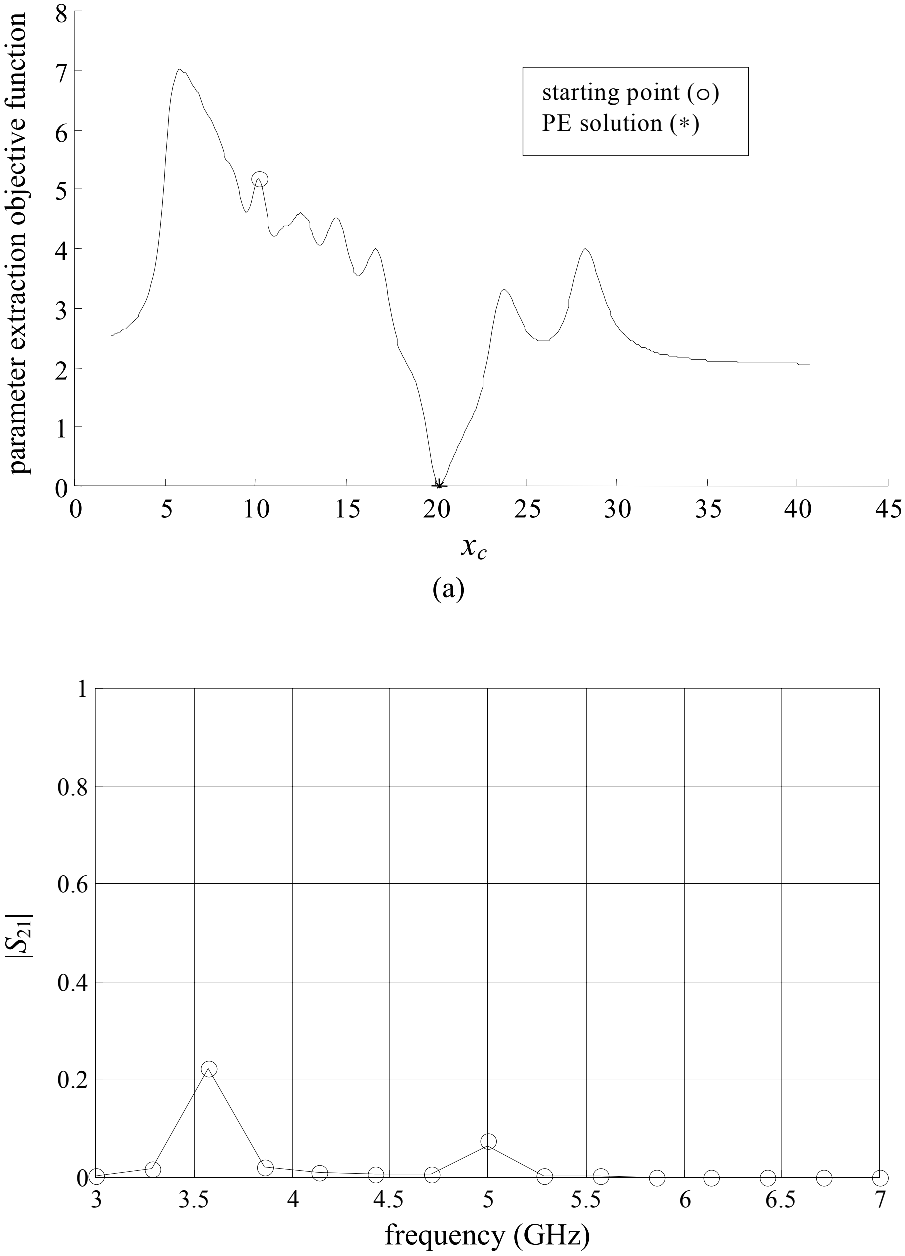
Similar experiments were realized for different values of C_p, repeating the parameter extraction procedure 10 times for each case in order to test the variation in the number of attempts needed for successful parameter extraction. Table 5.1 shows some of the results.



(a)

(b)

Fig. 5.6 Conventional parameter extraction process: (a) objective function, (b) coarse (−) and fine (○) model responses after parameter extraction (C_p = 10 pF).



(a)

(b)

Fig. 5.7 Proposed parameter extraction process: (a) objective function, (b) coarse (−) and fine (○) model responses after parameter extraction (C_p = 10 pF).

TABLE 5.1
RESULTS FOR 10 STATISTICAL PARAMETER EXTRACTIONS FOR THE LUMPED BANDPASS FILTER

PE #	number of attempts needed for successful PE		
	C_p = 5 pF	C_p = 6 pF	C_p = 10 pF
1	2	4	7
2	3	3	6
3	2	3	6
4	4	2	3
5	3	4	2
6	2	6	3
7	2	6	3
8	2	4	7
9	6	2	2
10	4	2	3

5.4 INVERSE NEUROMAPPING

When training the neural network $\boldsymbol{N}$ that implements the inverse mapping we solve the following optimization problem

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \; U_N(\boldsymbol{w}) \tag{5-6}$$

$$U_N(\boldsymbol{w}) = \left\| [\cdots \quad \boldsymbol{e}_l^T \quad \cdots]^T \right\|_2^2 \tag{5-7}$$

$$\boldsymbol{e}_l = \boldsymbol{x}_f^{(l)} - \boldsymbol{N}(\boldsymbol{x}_c^{(l)}, \boldsymbol{w}), \quad l = 1, \ldots, i \tag{5-8}$$

where i is the current NISM iteration and vector $\boldsymbol{w}$ contains the internal parameters (weights, bias, etc.) of the neural network $\boldsymbol{N}$.

The starting point $\boldsymbol{w}^{(0)}$ for solving (5-6) is a unit mapping, i.e. $\boldsymbol{N}(\boldsymbol{x}_c^{(l)}, \boldsymbol{w}^{(0)}) = \boldsymbol{x}_c^{(l)}$, for l = 1,…, i. Closed form expressions were derived in Section 5.2 to implements unit mappings for different nonlinear activation functions.

We use the Scaled Conjugate Gradient (SCG) algorithm available in the

Matlab™ Neural Network Toolbox (1998) for solving (5-6). Notice that the time consumed in solving (5-6) is almost neglectable since no coarse or fine model simulations are needed.

To control the generalization performance of the neural network ***N***, we follow a network growing strategy (see Haykin, 1999), in which case we start with a small perceptron to match the initial points and then add more neurons only when we are unable to meet a small error.

We initially assume a 2-layer perceptron given by

$$\boldsymbol{N}(\boldsymbol{x}_c, \boldsymbol{w}) = \boldsymbol{x}_f = \boldsymbol{W}^o \boldsymbol{x}_c + \boldsymbol{b}^o \tag{5-9}$$

where $\boldsymbol{W}^o \in \Re^{n\times n}$ is the matrix of output weighting factors, $\boldsymbol{b}^o \in \Re^n$ is the vector of output bias elements, and vector $\boldsymbol{w}$ contains $\boldsymbol{b}^o$ and the columns of $\boldsymbol{W}^o$. The starting point is obtained by making $\boldsymbol{W}^o = \boldsymbol{I}$ and $\boldsymbol{b}^o = \boldsymbol{0}$.

If a 2-layer perceptron is not sufficient to make the learning error $U_N(\boldsymbol{w}^*)$ small enough, then we use a 3-layer perceptron with h hidden neurons given by

$$\boldsymbol{N}(\boldsymbol{x}_c, \boldsymbol{w}) = \boldsymbol{W}^o \boldsymbol{\Phi}(\boldsymbol{x}_c) + \boldsymbol{b}^o \tag{5-10}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}_c) = [\varphi(s_1) \quad \varphi(s_2) \quad \dots \quad \varphi(s_h)]^T \tag{5-11}$$

$$\boldsymbol{s} = \boldsymbol{W}^h \boldsymbol{x}_c + \boldsymbol{b}^h \tag{5-12}$$

where $\boldsymbol{W}^o \in \Re^{n\times h}$, $\boldsymbol{b}^o \in \Re^n$, $\boldsymbol{\Phi}(\boldsymbol{x}_c) \in \Re^h$ is the vector of hidden signals, $\boldsymbol{s} \in \Re^h$ is the vector of activation potentials, $\boldsymbol{W}^h \in \Re^{h\times n}$ is the matrix of hidden weighting factors, $\boldsymbol{b}^h \in \Re^h$ is the vector of hidden bias elements and h is the number of hidden neurons. In our implementation of NISM optimization we use hyperbolic tangents as nonlinear activation functions, i.e., $\varphi(\cdot) = \tanh(\cdot)$. Vector $\boldsymbol{w}$ contains vectors $\boldsymbol{b}^o$, $\boldsymbol{b}^h$, the columns of $\boldsymbol{W}^o$ and the columns of $\boldsymbol{W}^h$.

Our starting point for solving (5-6) using (5-10) is also a unit mapping, which is obtained by making $\boldsymbol{b}^o = \boldsymbol{0}$, $\boldsymbol{b}^h = \boldsymbol{0}$, $\boldsymbol{W}^h = 0.1[\boldsymbol{I}\ \boldsymbol{0}]^T$ and $\boldsymbol{W}^o = 10[\boldsymbol{I}\ \boldsymbol{0}]$, assuming that the training data has been scaled between −1 and +1 (see Section 2.5.2). Notice that we consider $h \geq n$ in order to achieve the unit mapping.

The algorithm for finding the simplest inverse neuromapping is stated as follows

```
Algorithm: Inverse Neuromapping
begin
        solve (5-6) using (5-9)
        $h = n$
        while $U_N(\boldsymbol{w}^*) > \varepsilon_L$
                solve (5-6) using (5-10)
                $h = h+1$
end
```

In our implementation we use $\varepsilon_L = 1\times10^{-4}$. Notice that the algorithm for finding the inverse neuromapping uses a 2-layer perceptron during at least the first n+1 NISM iterations, since the points $(\boldsymbol{x}_{\mathrm{c}}^{(i)}, \boldsymbol{x}_f^{(i)})$ can be mapped with a linear mapping for $i = 1 \ldots n+1$. A 3-layer perceptron is needed only when we exceed n+1 NISM iterations and the mapping is significantly nonlinear.

5.5 NATURE OF THE NISM STEP

In this section we prove that the NISM step, $\boldsymbol{x}_f^{(i+1)} = \boldsymbol{N}(\boldsymbol{x}_c^*)$, is equivalent to a quasi-Newton step while the inverse mapping built during NISM optimization remains linear, i.e., while a 2-layer perceptron is enough to approximate the inverse mapping. We also prove that the NISM step gradually departs from a quasi-Newton step as the amount of nonlinearity needed in the inverse mapping increases.

5.5.1 Jacobian of the Inverse Mapping

From (5-9), the Jacobian $\boldsymbol{J}_N$ of the inverse mapping $\boldsymbol{N}(\boldsymbol{x}_c)$ when a 2-layer perceptron is employed is given by

$$\boldsymbol{J}_N = \boldsymbol{W}^o \tag{5-13}$$

When a 3-layer perceptron is used, the Jacobian $\boldsymbol{J}_N$ is obtained from (5-10) to (5-12) as

$$\boldsymbol{J}_N = \boldsymbol{W}^o \boldsymbol{J}_{\boldsymbol{\Phi}} \boldsymbol{W}^h \tag{5-14}$$

where $\boldsymbol{J}_{\boldsymbol{\Phi}} \in \Re^{h \times h}$ is a diagonal matrix given by $\boldsymbol{J}_{\boldsymbol{\Phi}} = \text{diag}(\varphi'(s_j))$, with $j = 1\ldots h$. We use (5-13) and (5-14) to demonstrate the nature of the NISM step $\boldsymbol{x}_f^{(i+1)} = \boldsymbol{N}(\boldsymbol{x}_c^{*})$.

5.5.2 NISM Step vs. Quasi-Newton Step

A general space mapping optimization problem can be formulated as solving the system of nonlinear equations

$$\boldsymbol{f}(\boldsymbol{x}_f) = \boldsymbol{P}(\boldsymbol{x}_f) - \boldsymbol{x}_c^{*} = \boldsymbol{0} \tag{5-15}$$

where $\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f)$ is the mapping function that makes the coarse model behave as the fine model, i.e., $\boldsymbol{R}_c(\boldsymbol{P}(\boldsymbol{x}_f)) \approx \boldsymbol{R}_f(\boldsymbol{x}_f)$. A Newton step for solving (5-15) is given by

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{J}_P^{-1} \boldsymbol{f} \tag{5-16}$$

where $\boldsymbol{J}_P \in \Re^{n \times n}$ is the Jacobian of the mapping function $\boldsymbol{P}(\boldsymbol{x}_f)$. This can be stated in an equivalent manner by using the Jacobian $\boldsymbol{J}_N \in \Re^{n \times n}$ of the inverse of the mapping $\boldsymbol{x}_f = \boldsymbol{N}(\boldsymbol{x}_c)$ (see appendix B)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{J}_N \boldsymbol{f} \tag{5-17}$$

Approximating $\boldsymbol{J}_N$ directly involves the same computational effort as approximating $\boldsymbol{J}_P$, but calculating the next step using (5-17) is computationally much more efficient than using (5-16), where a system of linear equations, possibly ill-conditioned, must be solved.

If a 2-layer perceptron is being used, we substitute (5-13) in (5-17) to obtain

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{W}^o(\boldsymbol{x}_c^{(i)} - \boldsymbol{x}_c^{*}) \tag{5-18}$$

which can be express using (5-9) as

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{x}_c^{*} - (\boldsymbol{x}_f^{(i)} - \boldsymbol{b}^o) + \boldsymbol{x}_f^{(i)} = \boldsymbol{N}(\boldsymbol{x}_c^{*}) \tag{5-19}$$

From (5-17) and (5-19) we conclude that while the inverse mapping built during NISM optimization remains linear, the NISM step is equivalent to a quasi-Newton step. Notice that we do not use any of the classical updating formulae to calculate an approximation of the inverse of the Jacobian; this is done by simply evaluating the current neural network at the optimal coarse solution.

If a 3-layer perceptron is being used, we substitute (5-14) in (5-17) to obtain

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(i)} - \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{W}^h(\boldsymbol{x}_c^{(i)} - \boldsymbol{x}_c^{*}) \tag{5-20}$$

Adding and subtracting $\boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{b}^h$ to (5-20)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}(\boldsymbol{W}^h\boldsymbol{x}_c^{*} + \boldsymbol{b}^h) - \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}(\boldsymbol{W}^h\boldsymbol{x}_c^{(i)} + \boldsymbol{b}^h) + \boldsymbol{x}_f^{(i)} \tag{5-21}$$

Substituting (5-12) in (5-21)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{s}(\boldsymbol{x}_c^{*}) - \boldsymbol{W}^o\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{s}(\boldsymbol{x}_c^{(i)}) + \boldsymbol{x}_f^{(i)} \tag{5-22}$$

Expanding the term $\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{s}(\boldsymbol{x}_c)$ we obtain

$$\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{s}(\boldsymbol{x}_c) = [\varphi'(s_1)s_1 \quad \ldots \quad \varphi'(s_h)s_h]^T\,. \tag{5-23}$$

Since we are using hyperbolic tangents as nonlinear activation functions, when a small amount of nonlinearity is present (e.g., $s_j < 0.1$), $\varphi(s_j) = s_j$, and $\varphi'(s_j)s_j = s_j = \varphi(s_j)$, for *j* = 1,…, *h*, and using (5-11) we express (5-23) as

$$\boldsymbol{J}_{\boldsymbol{\Phi}}\boldsymbol{s}(\boldsymbol{x}_c) = \boldsymbol{\Phi}(\boldsymbol{x}_c) \tag{5-24}$$

Substituting (5-24) in (5-22)

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{\Phi}(\boldsymbol{x}_c^*) - \boldsymbol{W}^o\boldsymbol{\Phi}(\boldsymbol{x}_c^{(i)}) + \boldsymbol{x}_f^{(i)} \tag{5-25}$$

Adding and subtracting $\boldsymbol{b}^o$ to (5-25) and using (5-10) we express (5-25) as

$$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{W}^o\boldsymbol{\Phi}(\boldsymbol{x}_c^*) + \boldsymbol{b}^o - \boldsymbol{W}^o\boldsymbol{\Phi}(\boldsymbol{x}_c^{(i)}) - \boldsymbol{b}^o + \boldsymbol{x}_f^{(i)} = \boldsymbol{N}(\boldsymbol{x}_c^*) \tag{5-26}$$

In conclusion, the NISM step gradually departs from a quasi-Newton step as the amount of nonlinearity needed in the inverse mapping increases.

5.6 TERMINATION CRITERION

As illustrated in the flow diagram of Fig. 5.1, we stop NISM optimization when the new iterate is close enough to the current point. We do this by testing the relative change in the fine model parameters. If the expression

$$\left\|\boldsymbol{x}_f^{(i+1)} - \boldsymbol{x}_f^{(i)}\right\|_2 \le \varepsilon_{end}(\varepsilon_{end} + \left\|\boldsymbol{x}_f^{(i)}\right\|_2) \tag{5-27}$$

is true, we end NISM optimization taking $\boldsymbol{x}_f^{(i)}$ as the solution, otherwise we continue. We use $\varepsilon_{end} = 5\times10^{-3}$ in our implementation. Notice that the fine model is not evaluated at the point $\boldsymbol{x}_f^{(i+1)}$.

5.7 EXAMPLES

5.7.1 Two-Section Impedance Transformer

As an illustrative case, consider the classical test problem of designing a

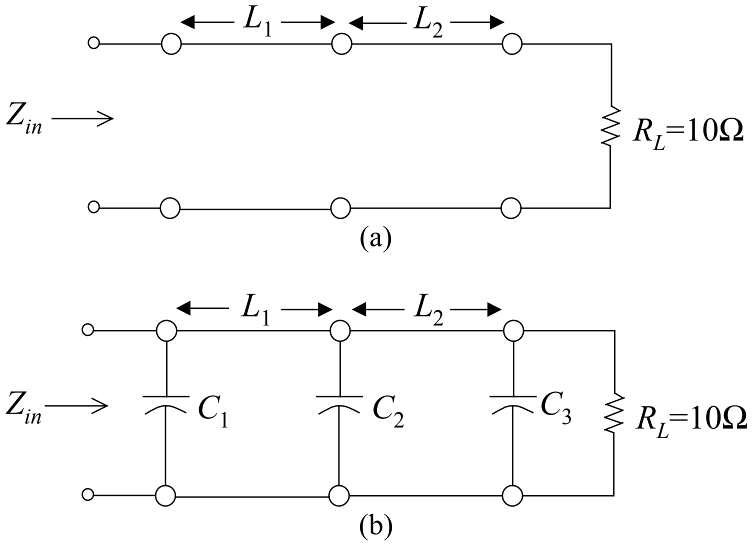


Fig. 5.8 Two-section impedance transformer: (a) coarse model, (b) "fine" model.

capacitively-loaded 10:1 two-section impedance transformer, proposed for the first time by Bandler (1969). The proposed coarse and "fine" models are shown in Fig. 5.8. The coarse model consists of ideal transmission lines, while the "fine" model consists of capacitively-loaded ideal transmission lines, with C_1 = C_2 = C_3 = 10pF. The design specifications are $|S_{11}| \leq 0.50$ for frequencies between 0.5 GHz and 1.5 GHz.

The electrical lengths of the two transmission lines at 1.0 GHz are selected as design parameters. The characteristic impedances are kept fixed at the following values: Z_1 = 2.23615 Ω, Z_2 = 4.47230 Ω. Both models were implemented in OSA90/hope (1997).

The optimal coarse model solution is $\boldsymbol{x}_c^* = [90 \;\; 90]^T$ (degrees). The coarse and fine model responses at $\boldsymbol{x}_c^*$ are shown in Fig. 5.9. We use only 10 frequency points from 0.2 to 1.8 GHz for the "fine" model.

NISM optimization requires only 3 "fine" model evaluations to solve this

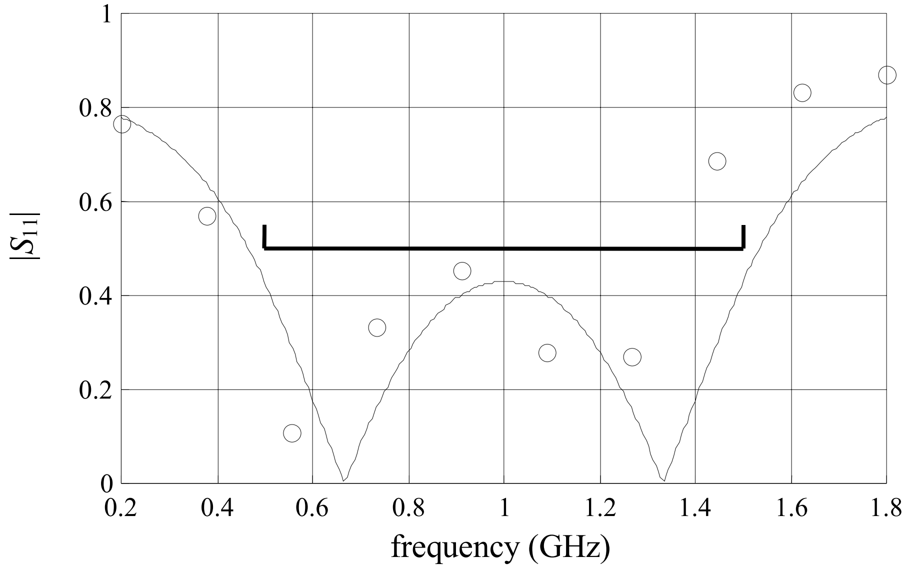


Fig. 5.9 Coarse (–) and fine (○) model responses at the optimal coarse solution $\boldsymbol{x}_c^*$ for the two-section impedance transformer.

problem. The values of the fine model parameters at each iteration are shown in Table 5.2. A 2-layer perceptron was enough to approximate the inverse mapping at all NISM iterations. The "fine" model response at the NISM solution is compared with the optimal coarse model response in Fig. 5.10. The fine model minimax objective function values at each NISM iteration are shown in Fig. 5.11.

Since both the coarse and "fine" models are actually very fast to evaluate, we applied direct minimax optimization to the "fine" model, obtaining $\boldsymbol{x}_f^*$ = [79.2651 74.2322]^T after 64 "fine" model evaluations. In Fig. 5.12 we compare the fine model response at this solution with the optimal NISM response; an excellent match is observed.

The same problem was solved by Bakr, Bandler, Madsen, Rayas-Sánchez and Søndergaard (2000) using Trust Region Aggressive Space Mapping exploiting Surrogates. It is noticed that this algorithm required 7 "fine" model evaluations.

TABLE 5.2
FINE MODEL PARAMETERS FOR THE
TWO-SECTION IMPEDANCE TRANSFORMER
AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[90 90]
2	[84.1990 83.0317]
3	[79.3993 73.7446]

5.7.2 Bandstop Microstrip Filter with Open Stubs

We apply NISM optimization to a bandstop microstrip filter with quarter-wave resonant open stubs, whose physical structure is illustrated in Fig. 3.15. The results obtained are compared with those described in Chapter 3. L_1, L_2 are the open stub lengths and W_1, W_2 the corresponding widths. An alumina substrate with thickness H = 25 mil,

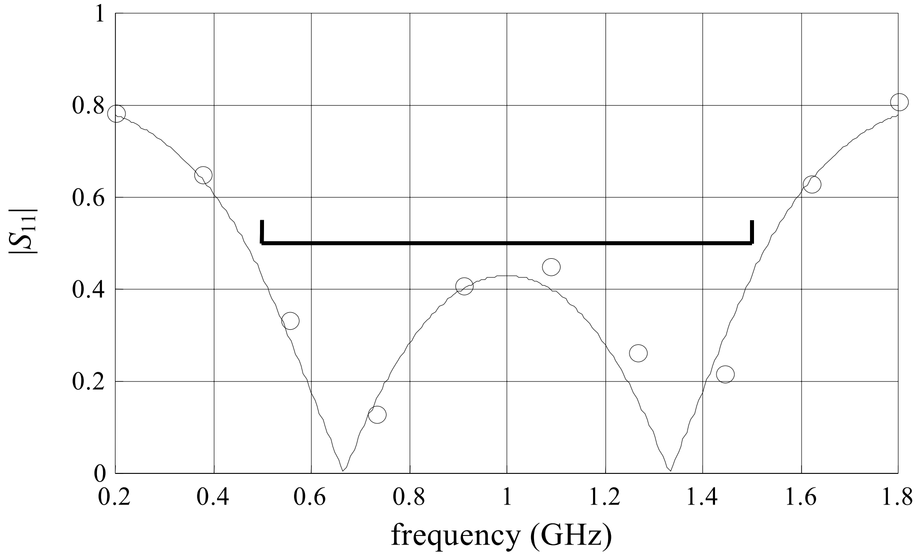


Fig. 5.10 Optimal coarse model response (−) and fine model response at NISM solution (○) for the two-section impedance transformer.

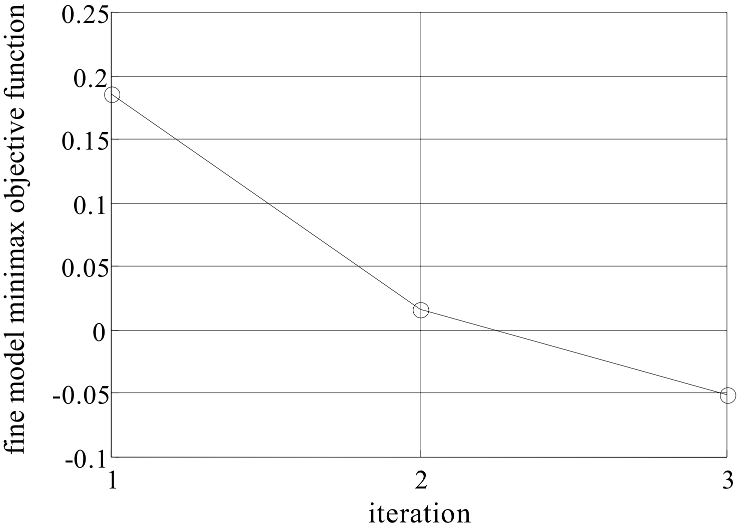


Fig. 5.11 Fine model minimax objective function values for the two-section impedance transformer at each NISM iteration.

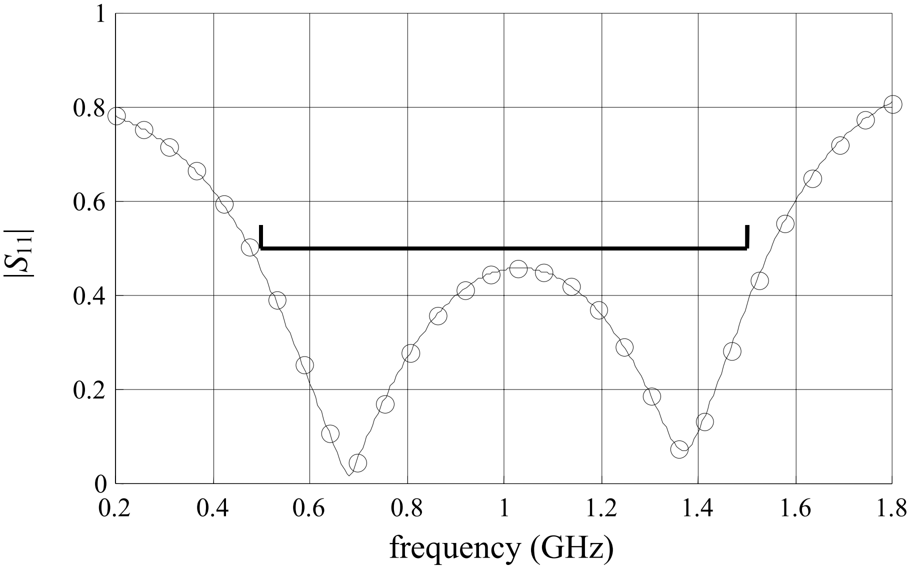


Fig. 5.12 Fine model response at NISM solution (○) and at direct minimax solution (−) for the two-section impedance transformer.

width W_0 = 25 mil and dielectric constant ε_r = 9.4 is used for a 50 Ω feeding line.

The specifications are the same as in Chapter 3: $|S_{21}| \leq 0.01$ in the stopband and $|S_{21}| \geq 0.9$ in the passband, where the stopband lies between 9.3 GHz and 10.7 GHz, and the passband includes frequencies below 8 GHz and above 12 GHz. The design parameters are $\boldsymbol{x}_f = [W_1\ W_2\ L_0\ L_1\ L_2]^T$.

Sonnet's ***em***™ (1997) driven by Empipe™ (1997) was again employed as the fine model, using a high-resolution grid with a 1mil×1mil cell size.

We use exactly the same coarse model described in Chapter 3, illustrated in Fig. 3.16. We also use the same optimal coarse model solution used for NSM optimization. The coarse and fine model responses at the optimal coarse solution are shown in Fig. 5.13, which are equivalent to those shown in Fig. 3.17.

NISM optimization requires only 4 fine model evaluations to solve this problem. The sequence of iterates is shown in Table 5.3 (all the points are on the grid, to avoid

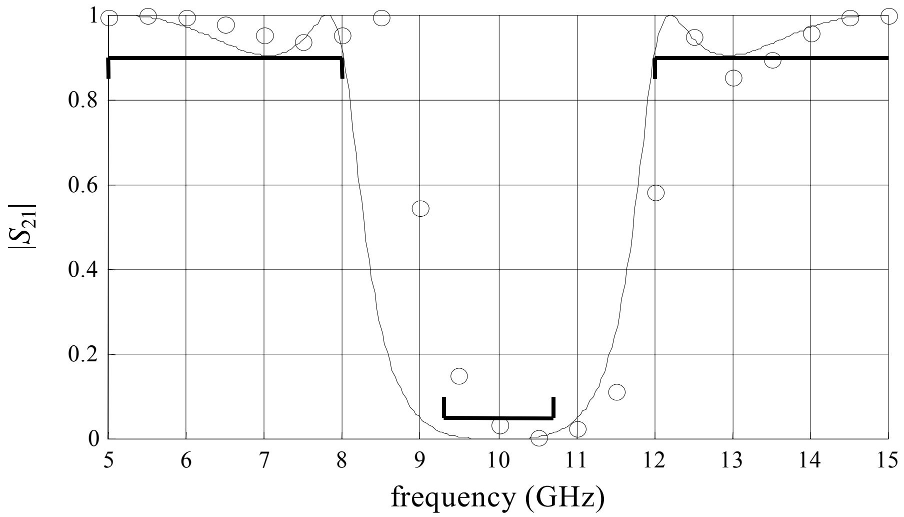


Fig. 5.13 Coarse and fine model responses at the optimal coarse solution for the bandstop filter with open stubs: OSA90/hope™ (–) and ***em***™ (○).

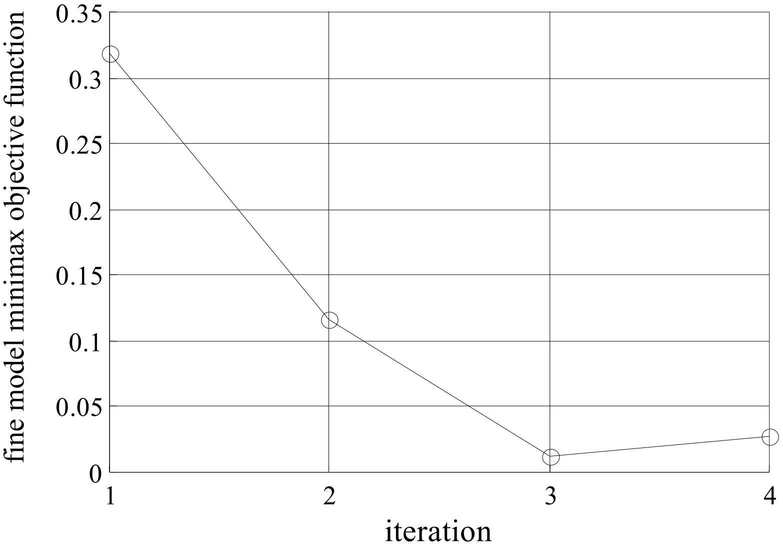


Fig. 5.14 Fine model minimax objective function values for the bandstop microstrip filter at each NISM iteration.

interpolation). A 2-layer perceptron was enough to approximate the inverse mapping at all NISM iterations. The fine model minimax objective function values at each NISM iteration are shown in Fig. 5.14. The fine model response at the NISM solution is compared with the optimal coarse model response in Fig. 5.15.

TABLE 5.3
FINE MODEL PARAMETERS FOR THE BANDSTOP FILTER WITH OPEN STUBS AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[6 9 106 110 109]
2	[7 11 103 112 111]
3	[9 20 95 115 115]
4	[9 19 95 115 114]

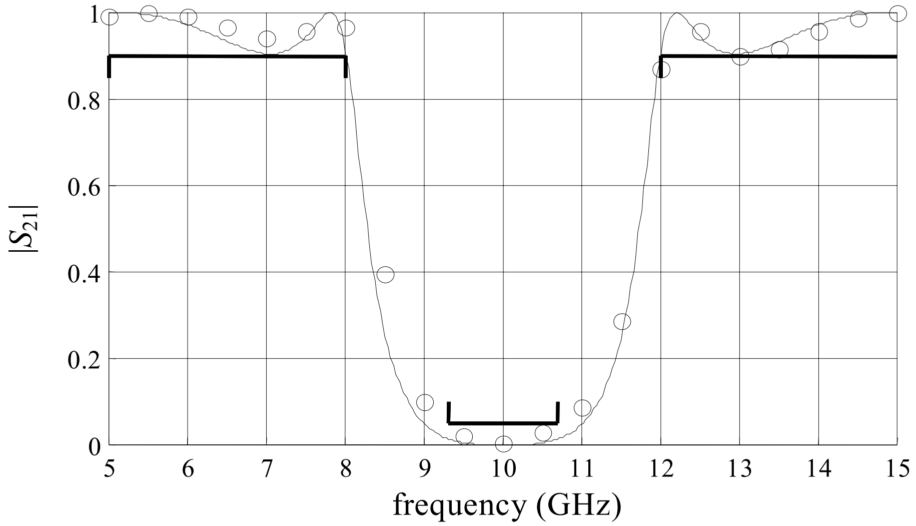


Fig. 5.15 Coarse model response (–) at $\boldsymbol{x}_c^*$ and fine model response (○) at NISM solution for the bandstop microstrip filter with open stubs.

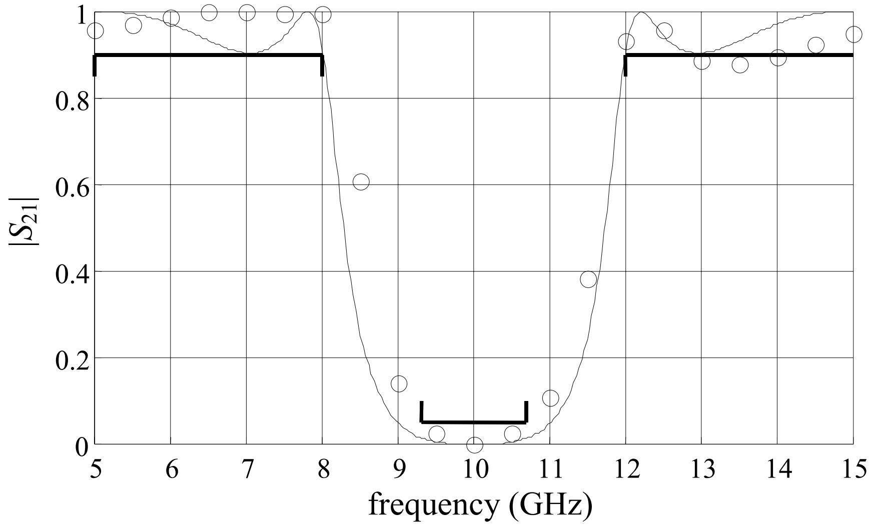


Fig. 5.16 Coarse model response (–) at $\boldsymbol{x}_c^*$ and fine model response (○) at NSM solution, obtained in Chapter 3, for the bandstop microstrip filter with open stubs.

As described in Chapter 3, NSM optimization required 13 fine model evaluations to find the solution to this problem, whose response is shown in Fig. 5.16 (this figure is equivalent to Fig. 3.22: we now use a linear scale for the responses to emphasize the quality of the solutions).

It is remarkable that NISM optimization not only requires fewer fine model evaluations, but also arrives at a solution closer to the solution of the original optimization problem (compare Fig. 5.15 with Fig. 5.16).

5.7.3 High Temperature Superconducting Microstrip Filter

We apply NISM optimization to a high-temperature superconducting (HTS) quarter-wave parallel coupled-line microstrip filter, and contrast our results with those obtained in Chapter 3 for the same problem. The physical structure of the HTS filter is illustrated in Fig. 2.21.

L_1, L_2 and L_3 are the lengths of the parallel coupled-line sections and S_1, S_2 and S_3 are the gaps between the sections. The width W is the same for all the sections as well as for the input and output lines, of length L_0. A lanthanum aluminate substrate with thickness H and dielectric constant ε_r is used.

We use the same specifications: $|S_{21}| \geq 0.95$ in the passband and $|S_{21}| \leq 0.05$ in the stopband, where the stopband includes frequencies below 3.967 GHz and above 4.099 GHz, and the passband lies in the range [4.008GHz, 4.058GHz]. The design parameters are $\boldsymbol{x}_f = [L_1\ L_2\ L_3\ S_1\ S_2\ S_3]^T$.

We use exactly the same fine and coarse models as described in Chapter 3. The schematic representation of the coarse model is illustrated in Fig. 3.8.

The same optimal coarse model solution is used as in Chapter 3. The coarse and fine model responses at the optimal coarse solution are shown in Fig. 5.17. These

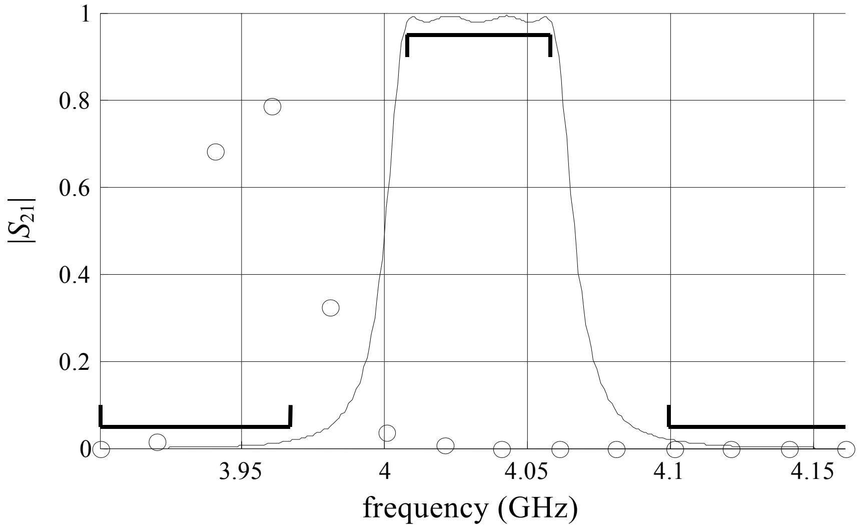


Fig. 5.17 Coarse and fine model responses at the optimal coarse solution for the HTS filter: OSA90/hope™ (–) and ***em***™ (○).

responses are equivalent to those shown in Fig. 3.9, but now they are plotted using linear scaling. Only 14 frequency points per frequency sweep are used for the fine model, as before.

After only 3 fine model simulations the optimal NISM solution was found. The sequence of fine model parameters at each NISM iteration is shown in Table 5.4 (all the points are on the grid, to avoid interpolation). A 2-layer perceptron was enough to approximate the inverse mapping at all NISM iterations.

Fig. 5.18 compares the optimal coarse response with the fine model response at the NISM solution $\boldsymbol{x}_f^{NISM}$ using a fine frequency sweep. An excellent match is achieved by the NISM solution.

A more detailed comparison in the passband is shown in Fig. 5.19, using a very fine frequency sweep.

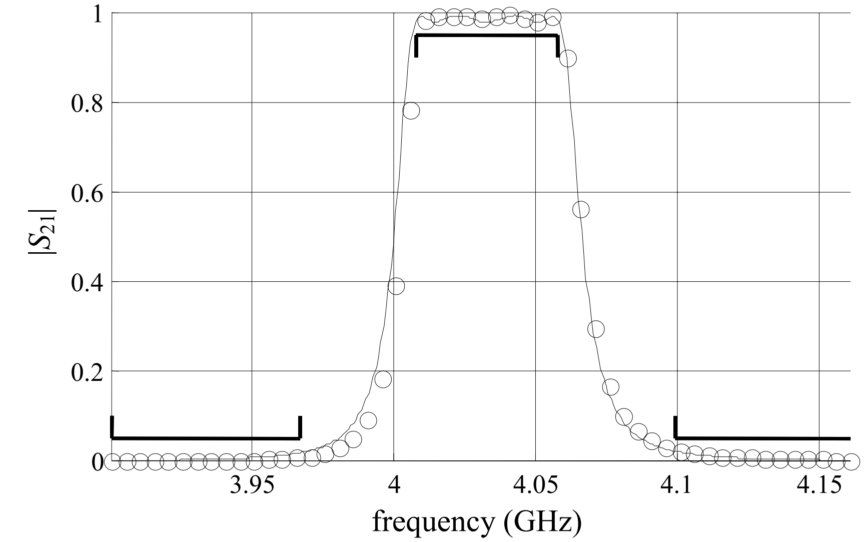


Fig. 5.18 Coarse model response at $\boldsymbol{x}_c^*$ (−), and fine model response at $\boldsymbol{x}_f^{NISM}$ (o), for the HTS filter using a fine frequency sweep.

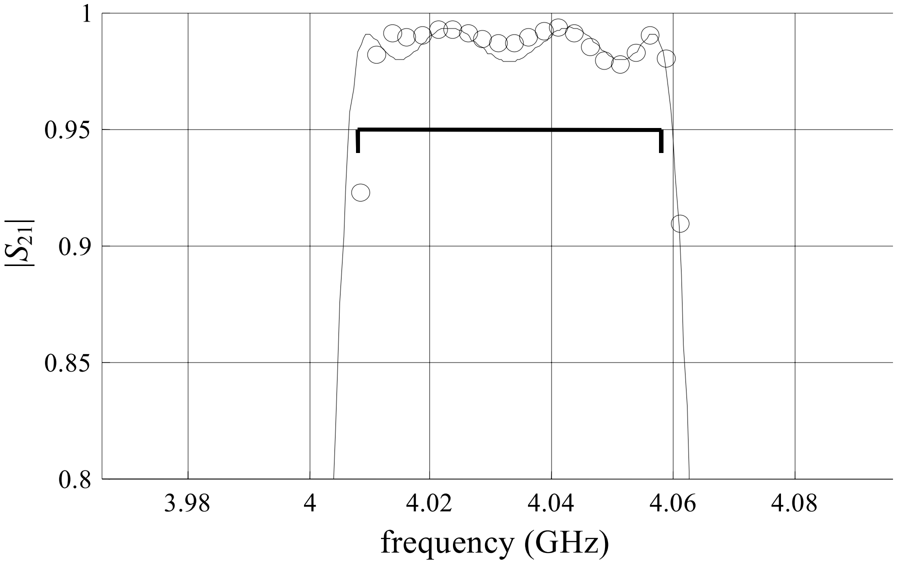


Fig. 5.19 Coarse model response at $\boldsymbol{x}_c^*$ (−) and fine model response at $\boldsymbol{x}_f^{NISM}$ (o) for the HTS filter, in the passband, using a very fine frequency sweep.

TABLE 5.4
FINE MODEL PARAMETERS FOR THE
HTS MICROSTRIP FILTER
AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[188 198 189 22 99 112]
2	[187 196 187 21 84 92]
3	[186 194 185 20 80 89]

The fine model minimax objective function values at each NISM iteration for this problem are shown in Fig. 17

Fig. 5.21 reproduce the results shown in Fig. 3.14, obtained by applying NSM optimization to the same problem, where the optimal NSM solution was found after 14 fine model evaluations, as described in Section 3.9.1.

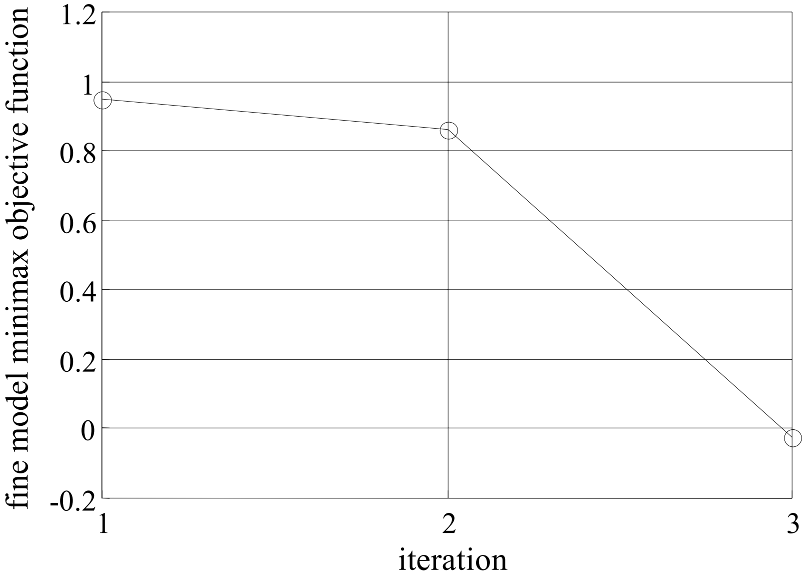


Fig. 5.20 Fine model minimax objective function values for the HTS microstrip filter at each NISM iteration.

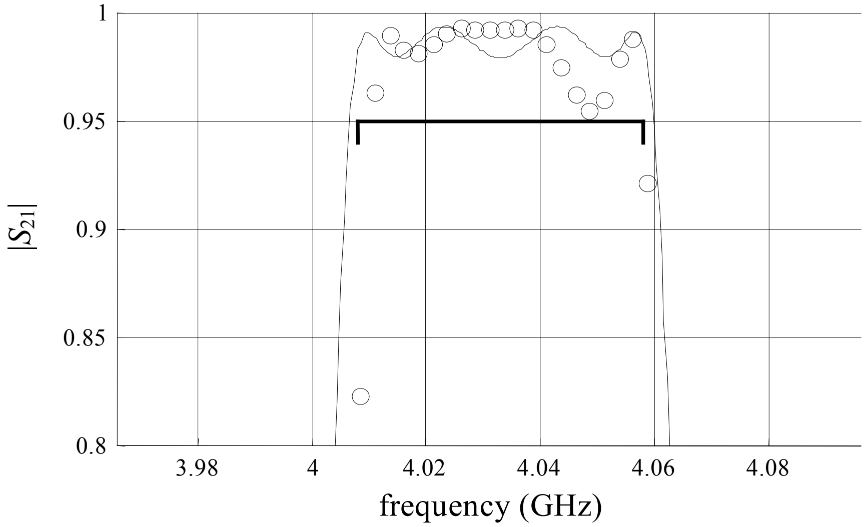


Fig. 5.21 Coarse model response at $\boldsymbol{x}_c^*$ (−) and fine model response at $\boldsymbol{x}_f^{NSM}$ (o) for the HTS filter, in the passband, using a very fine frequency sweep.

This problem was also solved by Bakr, Bandler, Madsen, Rayas-Sánchez and Søndergaard (2000) using Trust Region Aggressive Space Mapping exploiting Surrogates. It is noticed that this algorithm required 8 fine model evaluations; the corresponding fine model minimax objective function values are shown in Fig. 5.22.

Once again, it is seen that NISM optimization is not only more efficient in terms of the required fine model evaluations, but also yields a solution closer to the optimal solution of the original optimization problem (compare Fig. 5.19 with Fig. 5.21, as well as Fig. 5.20 with Fig. 5.22).

For the two previous examples of NISM optimization, parameter extraction was successfully performed in just one attempt at every NISM iteration. That was not the case for the HTS filter, where the parameter extraction objective function has many poor local minima around $\boldsymbol{x}_c^*$. Our proposed algorithm for statistical parameter extraction overcame this problem.

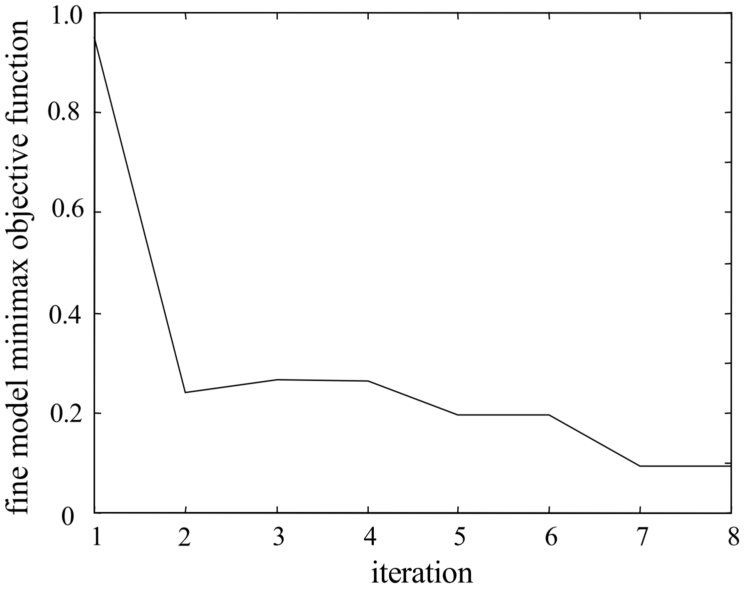


Fig. 5.22 Fine model minimax objective function values for the HTS microstrip filter at each iteration using Trust Region Aggressive Space Mapping exploiting Surrogates, as obtained by Bakr, Bandler, Madsen, Rayas-Sánchez and Søndergaard (2000).

We applied NISM optimization to the HTS filter 5 times in order to test the statistical parameter extraction results. In Table 5.5 we show the number of attempts needed for successful parameter extraction at each NISM iteration for the 5 optimizations. Exactly the same sequence of points illustrated in Table 5.4 was predicted by each of the 5 optimizations.

Table 5.5 also confirms that the most challenging parameter extraction problems in a space mapping-based algorithm appears at the first SM iterations, when the fine model response is far from the optimal coarse model response. As the space mapping algorithm progresses, the fine model response gets closer to the optimal coarse model response, making each time the optimal coarse solution $\boldsymbol{x}_c^*$ a better starting point for the parameter extraction.

TABLE 5.5
PARAMETER EXTRACTION RESULTS FOR 5
NISM OPTIMIZATIONS FOR THE HTS FILTER

i	number of attempts needed for successfull PE				
1	12	9	3	10	8
2	3	3	6	7	3
3	1	1	1	1	1

5.7.4 Lumped Parallel Resonator

In the three examples of NISM optimization described so far, the inverse neuromapping was always approximated by a 2-layer perceptron. Even in the case of the HTS filter, which is far from trivial and computationally very intensive, a simple linear inverse mapping was enough to drive the fine model toward the optimal solution in a few iterations. In order to demonstrate the behavior of NISM optimization when a nonlinear inverse mapping is actually needed, consider the following synthetic problem. Both the coarse and the "fine" models are illustrated in Fig. 5.23.

The coarse model consists of a canonical parallel lumped resonator (see Fig.

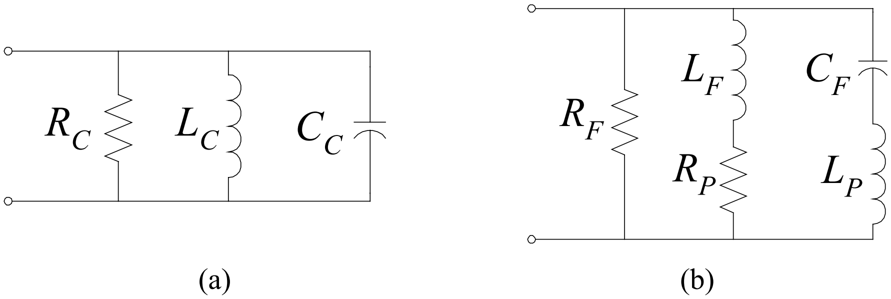


Fig. 5.23 Models for the parallel lumped resonator used to illustrate a nonlinear inverse mapping: (a) coarse model, (b) fine model.

5.23a), whose design parameters are $\boldsymbol{x}_c = [R_C \ \ L_C \ \ C_C]^T$. The "fine" model can be seen as the same parallel lumped resonator with a parasitic series resistor R_P associated to the inductance L_F, and a parasitic series inductance L_P associated to the capacitor C_F (see Fig. 5.23b). The fine model design parameters are $\boldsymbol{x}_f = [R_F \ \ L_F \ \ C_F]^T$. We take $R_P = 0.5\ \Omega$, $L_P = 0.1$ nH. The numerical values of R_C and R_F are expressed in ohms, those of L_C and L_F in nH and those of C_C and C_F in pF.

The design specifications are (assuming a reference impedance of 50 Ω): $|S_{11}| > 0.8$ from 1 GHz to 2.5 GHz and from 3.5 GHz to 5 GHz, $|S_{11}| < 0.2$ from 2.95 GHz to 3.05 GHz.

Performing direct minimax optimization on the coarse model we find the optimal coarse solution $\boldsymbol{x}_c^* = [50 \ \ 0.2683 \ \ 10.4938]^T$. The optimal coarse model response and the fine model response at the optimal coarse solution are illustrated in Fig. 5.24.

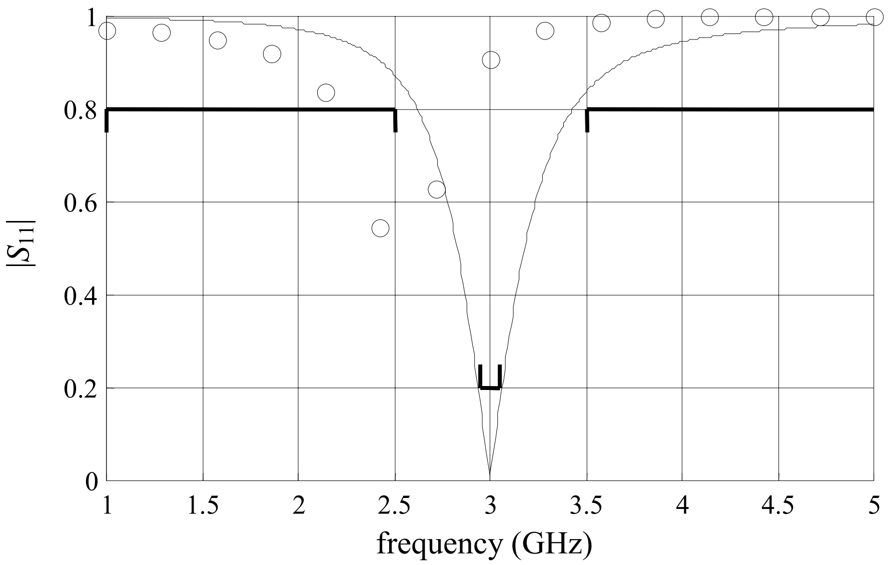


Fig. 5.24 Coarse model response (–) and "fine" model response (o) at the optimal coarse solution $\boldsymbol{x}_c^*$ for the parallel lumped resonator.

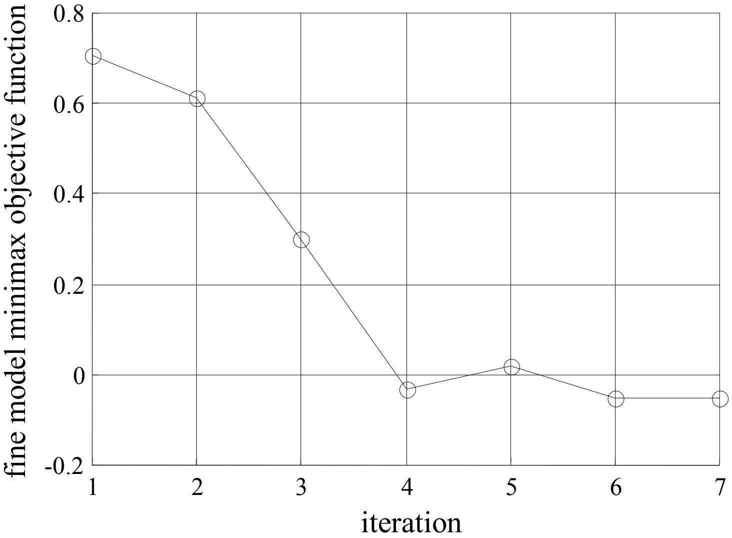


Fig. 5.25 Fine model minimax objective function values for the parallel lumped resonator filter at each NISM iteration.

Applying NISM optimization, we find the optimal space mapped solution after 7 iterations. The fine model minimax objective function values at each iteration are shown in Fig. 5.25. The fine model points at each NISM iteration are illustrated in Table 5.6.

TABLE 5.6
FINE MODEL PARAMETERS FOR THE
PARALLEL LUMPED RESONATOR
AT EACH NISM ITERATION

i	$\boldsymbol{x}_f^{(i)\,T}$
1	[50 0.2683 10.4938]
2	[64.9696 0.3147 4.4611]
3	[72.9922 0.3378 5.5555]
4	[90.0973 0.3623 6.1337]
5	[105.5360 0.3516 5.9095]
6	[110.2669 0.3591 6.0519]
7	[111.0306 0.3594 6.0518]

In this problem, NISM optimization uses a 2-layer perceptron only during the first 4 iterations (*n*+1, as expected), and it uses a 3-layer perceptron in the last 3 iterations. The final inverse mapping is approximated using a 3-layer perceptron with 3 hidden neurons only. It is interesting to notice that the linear mapping is able to obtain a response that satisfies the specifications, since the minimax objective function at the fourth iteration is already negative (see Fig. 5.25).

We compare in Fig. 5.26 the coarse model response at the optimal coarse solution and the fine model response at the NISM solution $\boldsymbol{x}_f^{NISM}$ = [111.0306 0.3594 6.0518]^T.

From this example we can see that even for an extremely simple microwave optimization problem, the complexity of the relationship between the coarse and the fine models can demand a nonlinear inverse mapping to align both models during NISM optimization.

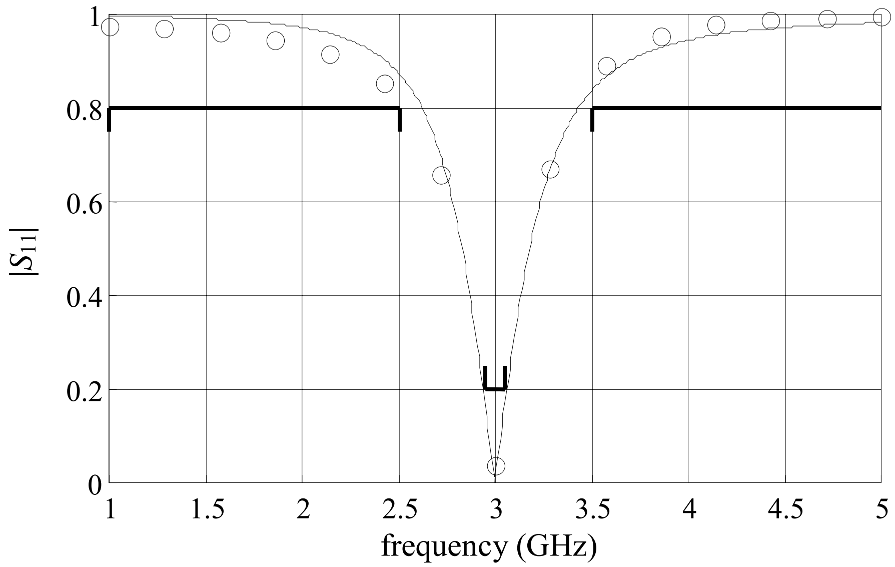


Fig. 5.26 Coarse model response at optimal coarse solution (–) and “fine” model response at the NISM solution (o) for the parallel lumped resonator.

We can also confirm from this example and the HTS filter example that the degree of misalignment between the coarse and fine model responses at the starting point is not an indication of the the degree of nonlinearity in the inverse mapping between both models (compare Fig. 5.24 and Fig. 5.17).

5.8 CONCLUSIONS

We have described in this chapter Neural Inverse Space Mapping (NISM) optimization for EM-based design of microwave structures, where the inverse of the mapping is exploited for the first time in a space mapping algorithm.

NISM optimization follows an aggressive approach in the sense that it does not require up-front EM simulations to start building the mapping.

A simple statistical procedure overcomes the existence of poor local minima during parameter extraction, avoiding the need of multipoint parameter extraction or frequency mapping.

A neural network whose generalization performance is controlled through a network growing strategy approximates the inverse of the mapping at each iteration. We have found that for many practical microwave problems, a simple linear inverse mapping, i.e., a 2-layer perceptron, is sufficient to reach a practically optimal fine model response.

The NISM step simply consists of evaluating the current neural network at the optimal coarse solution. We prove that this step is equivalent to a quasi-Newton step while the inverse mapping remains essentially linear, and gradually departs from a quasi-Newton step as the amount of nonlinearity in the inverse mapping increases.

We also found that our new algorithm exhibits superior performance over the

Neural Space Mapping (NSM) optimization algorithm, described in Chapter 3, as well as over the Trust Region Aggressive Space Mapping exploiting Surrogates, developed by Bakr, Bandler, Madsen, Rayas-Sánchez and Søndergaard (2000).

Chapter 6

CONCLUSIONS

This thesis has presented innovative methods for electromagnetics-based computer-aided modeling and design of microwave circuits exploiting artificial neural networks (ANNs) and space mapping (SM) technology. We have illustrated these methods by modeling and optimizing several practical microstrip structures.

Five powerful techniques to generate SM-based neuromodels have been described and illustrated: Space Mapped Neuromodeling (SMN), Frequency-Dependent Space Mapped Neuromodeling (FDSMN), Frequency Space Mapped Neuromodeling (FSMN), Frequency Mapped Neuromodeling (FMN) and Frequency Partial-Space Mapped Neuromodeling (FPSMN).

The SM-based neuromodeling techniques make use of the vast set of empirical and circuit-equivalent models already available. They need a much smaller number of fine model evaluations for training, improve the generalization performance and reduce the complexity of the ANN topology w.r.t. the conventional neuromodeling approach.

Using frequency-sensitive neuromappings significantly expand the usefulness of microwave empirical models that were developed under quasi-static assumptions. We have also demonstrated that neuromapping the frequency can be an effective technique to align severely shifted responses.

For many practical microwave problems it is not necessary to map the complete set of physical parameters. By establishing a partial mapping, a more efficient use of the implicit knowledge in the empirical model is achieved and the corresponding neuromapping becomes simpler and easier to train.

We have also described an innovative algorithm for EM optimization that exploits our SM-based neuromodeling techniques: Neural Space Mapping (NSM) optimization.

In NSM optimization, an initial mapping is established by performing a reduced number of upfront EM simulations. The coarse model sensitivities are exploited to select those initial points.

NSM does not require parameter extraction to predict the next point. Instead, we use Huber optimization to train simple SM-based neuromodels at each iteration. These SM-based neuromodels are developed without using testing points: their generalization performance is controlled by gradually increasing their complexity starting with a 3-layer perceptron with 0 hidden neurons. The next point is predicted by optimizing the current SM-based neuromodel at each iteration.

We have proposed an efficient strategy to realize electromagnetic-based statistical analysis and yield optimization of microwave structures using SM-based neuromodels.

We mathematically formulate the problem of statistical analysis and yield optimization using SM-based neuromodels. A formulation for the relationship between the fine and coarse model sensitivities through a nonlinear, frequency-sensitive neuromapping has been found, which is a generalization of the lemma found in the work by Bakr, Bandler, Georgieva and Madsen (1999).

When asymmetric variations in the physical parameters due to tolerances are considered, the need of extra EM simulations is avoided by re-using the available neuromappings and exploiting the typical asymmetric nature of the coarse models.

We have also described Neural Inverse Space Mapping (NISM) optimization for EM-based design of microwave structures, where the inverse of the mapping is explicitly used for the first time in a space mapping algorithm.

NISM optimization does not require up-front EM simulations to start building the mapping. A simple statistical procedure overcomes the existence of poor local minima during parameter extraction, avoiding the need of multipoint parameter extraction or frequency mapping.

The inverse of the mapping at each NISM iteration is approximated by a neural network whose generalization performance is controlled through a network growing strategy. We have found that for many practical microwave problems, a simple linear inverse mapping, i.e., a 2-layer perceptron, is sufficient to reach a practically optimal fine model response.

NISM step is calculated by simply evaluating the current neural network at the optimal coarse solution. We proof that this step is equivalent to a quasi-Newton step while the inverse mapping remains essentially linear, and gradually departs from a quasi-Newton step as the amount of nonlinearity in the inverse mapping increases.

In the examples considered we found that NISM optimization exhibits superior performance than NSM optimization, as well as than the Trust Region Aggressive Space Mapping exploiting Surrogates, developed by Bakr, Bandler, Madsen, Rayas-Sánchez and Søndergaard (2000).

From the experience and knowledge gained in the course of this work the author is convinced that the following research topics should be addressed for further development:

(1) The neural space mapping methods for modeling and design considered in this thesis are formulated in the frequency domain. When frequency-independent neuromappings are sufficient, either for SM-based neuromodeling or for NSM optimization, these methods can be in principle applied to the time domain. That is also the case for NISM optimization. Further work is needed to demonstrate this with specific examples. Nevertheless, if a frequency-sensitive neuromapping is needed, further research has to be done for the expansion of these methods to the time domain, especially for transient responses. Feedforward 3-layer perceptrons are not likely to be suitable under these circunstances: a different neural network paradigm might be needed for the neuromapping. The use of recurrent neural networks is suggested (see the work by Fang, Yagoub, Wang and Zhang, 2000).

(2) The SM-based neuromodels were tested by comparing their responses with the corresponding EM responses in the frequency domain. Excellent agreement was observed. Nevertheless, further research is needed to analyze the passivity preservation of the SM-based neuromodels, especially if they are to be developed for transient simulation. The loss of passivity can be a serious problem because passive components inserted in a more complex network may cause the overall network to be unstable (see the work by Dounavis, Nakhla and Achar, 1999).

(3) All the microwave examples considered in this thesis, for both modeling and optimization, are electrically linear. Further work should be realized to apply these neural space mapping methods to nonlinear microwave circuits. A good

candidate to start with is the NISM optimization of a microwave power amplifier, for example.

(4) The practical microwave examples designed in this thesis using neural inverse space mapping (NISM) optimization showed that a simple linear inverse mapping can be enough to arrive at an optimal EM response. This suggests a simplification of NISM optimization by keeping the inverse mapping linear at all the iterations, and removing from the learning set those points that are far from the SM solution when more than n+1 iterations are accumulated.

(5) Related to the previous point, an interesting comparison can be realized between linear inverse space mapping and aggressive space mapping (ASM) using Broyden's update. In order to make a fair comparison, both algorithms should use the same parameter extraction procedure. A good candidate would be the statistical parameter extraction algorithm described in Chapter 5. From this comparison new light will emerge to increase understanding of the SM-based optimization algorithms.

Appendix A

Implementation of SM-based Neuromodels using *NeuroModeler*

Alternative realizations of SM based neuromodels of practical passive components using commercial software are described in this appendix. An SM-based neuromodel of a microstrip right angle bend is developed using *NeuroModeler* (1999), and entered into HP ADS (1999) as a library component through an ADS plug-in module.

The physical structure of this microwave component, the characteristics of the coarse and fine models used, the region of interest for modeling, and the training and testing sets, are described in Section 2.6.1.

Fig. A.1 illustrates the frequency space-mapped neuromodeling (FSMN) strategy for the microstrip right angle bend, which was implemented using *NeuroModeler* as shown in Fig. A.2.

The FSMN model of the microstrip bend as implemented in *NeuroModeler* consists of a total of 6 layers (see Fig. A.2). The first layer, from bottom to top, has the input parameters of the neuromapping (W, H, ε_r, and *freq*), which are scaled to ±1 to improve the numerical behavior during training.

The second layer from bottom to top corresponds to the hidden layer of the ANN implementing the mapping (see Fig. 4b): optimal generalization performance is achieved

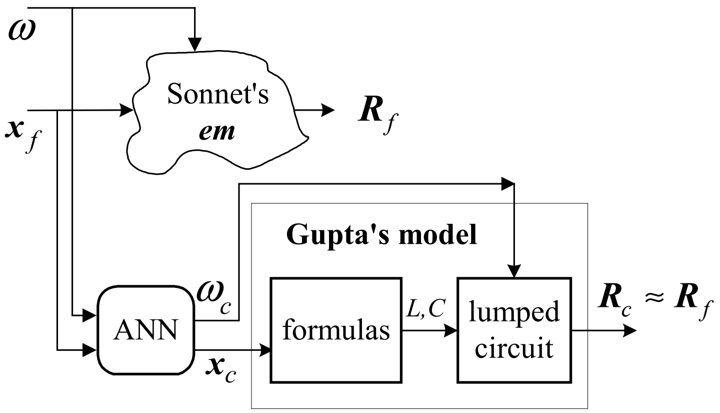
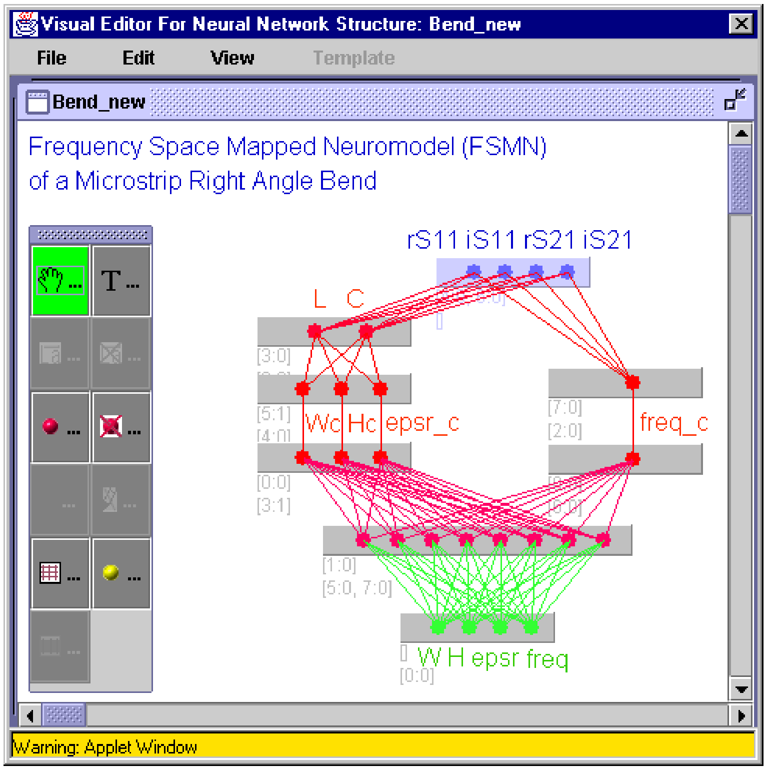


Fig. A.1 Strategy for the frequency space-mapped neuromodel (FSMN) of the microstrip right angle bend.



Frequency Space Mapped Neuromodel (FSMN) of a Microstrip Right Angle Bend

Warning: Applet Window

Fig. A.2 FSMN of the microstrip right angle bend as implemented in *Neuromodeler*.

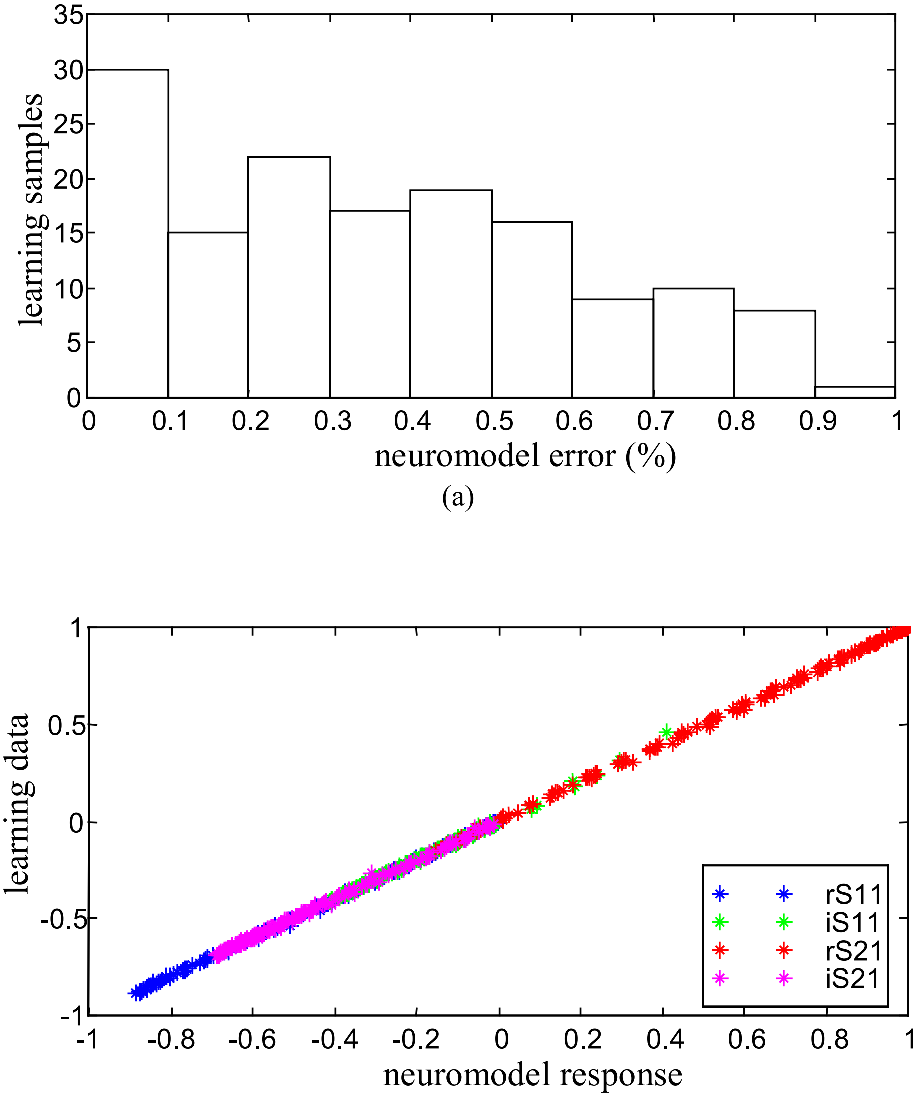
with 8 hidden neurons with sigmoid non-linearities.

The third layer is linear and contains the coarse design parameters $\boldsymbol{x}_c$ and the mapped frequency ω_c before de-scaling. The fourth layer is added to simply de-scale the parameters.

Gupta's formulas to calculate L and C are programmed as the internal analytical functions of the fifth hidden layer, using the built-in `MultiSymbolicFixed` function. Finally, the output layer, at the top, contains a simple internal circuit simulator that computes the real and imaginary parts of S_{11} and S_{21} for the lumped LC equivalent circuit. This layer uses the built-in `CktSimulatorPS` function available in *NeuroModeler*.

Fig. A.3 shows the learning errors and Fig. A.4 the testing errors of the FSMN bend model after training using *NeuroModeler*. Conjugate Gradient and Quasi Newton built-in training methods are used. The average and worst case learning errors are 0.43% and 1.00%, while the average and worst-case testing errors are 1.04% and 10.94%. Excellent generalization performance is achieved. Plots in Fig. A.3 and Fig. A.4 were produced using the export-to-*MatLab*™ capability available in *Neuromodeler*.

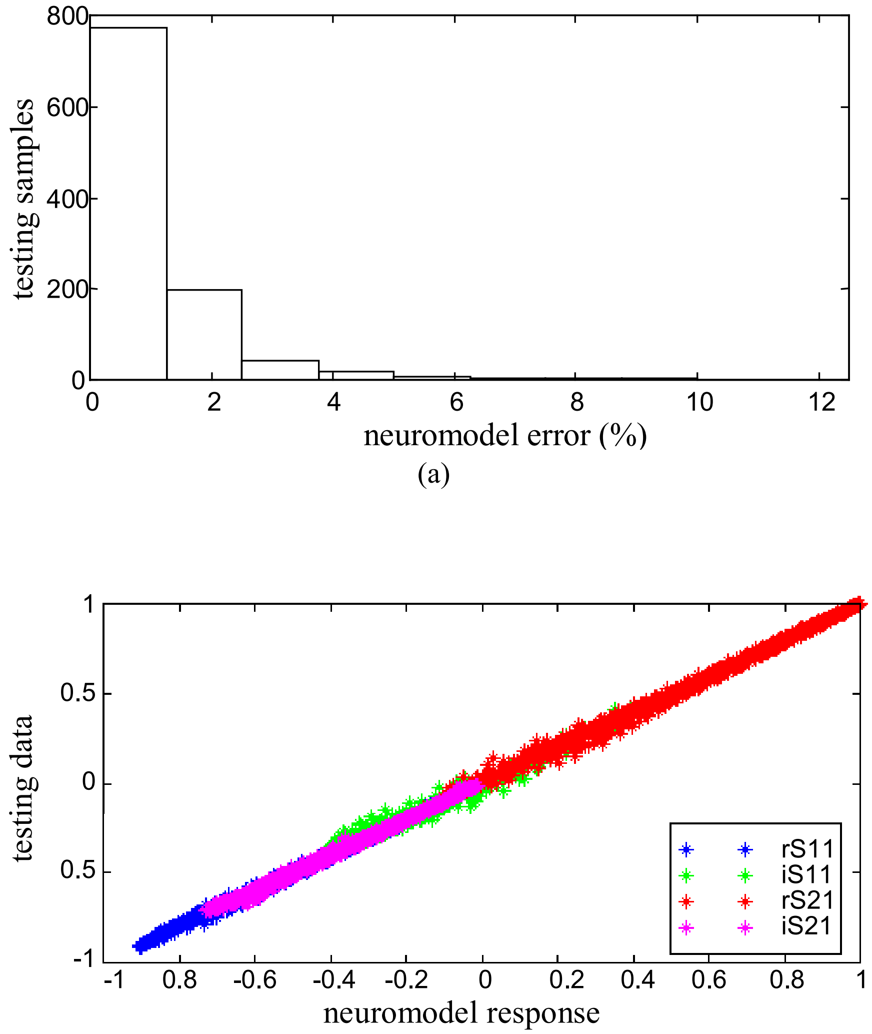
The FSMN model of the right angle bend can now be used in HP ADS for fast and accurate simulations within the region of operation shown in Table 2.1: it can be entered as a user-defined model through the plug-in module *NeuroADS* available in *Neuromodeler*. The use of *Neuromodeler* for implementing SM-based neuromodels was first proposed by Bandler, Rayas-Sánchez, Wang and Zhang (2000).



(a)

(b)

Fig. A.3 Errors in the learning set of the FSMN model after training: (a) histogram of learning errors, (b) correlation to learning data.



(a)

(b)

Fig. A.4 Errors in the testing set of the FSMN model after training: (a) histogram of testing errors, (b) correlation to testing data.

Appendix B

Jacobian of the Inverse Mapping

In this appendix we prove that $\boldsymbol{J}_N = \boldsymbol{J}_P^{-1}$. Let $\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_f)$, with $\boldsymbol{P}: \Re^n \rightarrow \Re^n$, and $\boldsymbol{x}_f = \boldsymbol{N}(\boldsymbol{x}_c)$ its inverse function. Using $\boldsymbol{P}(\boldsymbol{x}_f)$, we can write the system of equations

$$\begin{aligned} dx_{c1} &= \frac{\partial x_{c1}}{\partial x_{f1}} dx_{f1} + \frac{\partial x_{c1}}{\partial x_{f2}} dx_{f2} + \ldots + \frac{\partial x_{c1}}{\partial x_{fn}} dx_{fn} \\ dx_{c2} &= \frac{\partial x_{c2}}{\partial x_{f1}} dx_{f1} + \frac{\partial x_{c2}}{\partial x_{f2}} dx_{f2} + \ldots + \frac{\partial x_{c2}}{\partial x_{fn}} dx_{fn} \\ &\vdots \\ dx_{cn} &= \frac{\partial x_{cn}}{\partial x_{f1}} dx_{f1} + \frac{\partial x_{cn}}{\partial x_{f2}} dx_{f2} + \ldots + \frac{\partial x_{cn}}{\partial x_{fn}} dx_{fn} \end{aligned} \tag{B-1}$$

Let us define

$$\boldsymbol{dx}_c = [dx_{c1} \quad \ldots \quad dx_{cn}]^T \tag{B-2}$$

$$\boldsymbol{dx}_f = [dx_{f1} \quad \ldots \quad dx_{fn}]^T \tag{B-3}$$

$$\boldsymbol{J}_P = \begin{bmatrix} \frac{\partial x_{c1}}{\partial x_{f1}} & \frac{\partial x_{c1}}{\partial x_{f2}} & \cdots & \frac{\partial x_{c1}}{\partial x_{fn}} \\ \frac{\partial x_{c2}}{\partial x_{f1}} & \frac{\partial x_{c2}}{\partial x_{f2}} & \cdots & \frac{\partial x_{c2}}{\partial x_{fn}} \\ \vdots & & & \\ \frac{\partial x_{cn}}{\partial x_{f1}} & \frac{\partial x_{cn}}{\partial x_{f2}} & \cdots & \frac{\partial x_{cn}}{\partial x_{fn}} \end{bmatrix} \tag{B-4}$$

Substituting (B-2)-(B-4) in (B-1)

$$\boldsymbol{dx}_c = \boldsymbol{J}_P\ \boldsymbol{dx}_f \tag{B-5}$$

Similarly, using $\boldsymbol{N}(\boldsymbol{x}_c)$ we obtain

$$\boldsymbol{dx}_f = \boldsymbol{J}_N\ \boldsymbol{dx}_c \tag{B-6}$$

where

$$\boldsymbol{J}_N = \begin{bmatrix} \frac{\partial x_{f1}}{\partial x_{c1}} & \frac{\partial x_{f1}}{\partial x_{c2}} & \cdots & \frac{\partial x_{f1}}{\partial x_{cn}} \\ \frac{\partial x_{f2}}{\partial x_{c1}} & \frac{\partial x_{f2}}{\partial x_{c2}} & \cdots & \frac{\partial x_{f2}}{\partial x_{cn}} \\ \vdots & & & \\ \frac{\partial x_{fn}}{\partial x_{c1}} & \frac{\partial x_{fn}}{\partial x_{c2}} & \cdots & \frac{\partial x_{fn}}{\partial x_{cn}} \end{bmatrix} \tag{B-7}$$

Comparing (B-5) and (B-6) we conclude that $\boldsymbol{J}_N = \boldsymbol{J}_P^{-1}$. Notice that when $\boldsymbol{x}_f$ and $\boldsymbol{x}_c$ have different dimensionality, $\boldsymbol{J}_N$ is the pseudoinverse of $\boldsymbol{J}_P$.

BIBLIOGRAPHY

M.R. Aaron (1956), “The use of least squares in system design,” *IRE Trans. Circuit Theory*, vol. 3, pp. 224-231.

S. Akhtarzad and P.B. Johns (1973), “Transmission-line matrix solution of waveguides with wall losses,” Elec. Lett., vol. 9, pp. 335-336.

M.H. Bakr, J.W. Bandler, R.M. Biernacki, S.H. Chen, and K. Madsen (1998), “A trust region aggressive space mapping algorithm for EM optimization,” *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 2412-2425.

M.H. Bakr, J.W. Bandler, N. Georgieva and K. Madsen (1999), “A hybrid aggressive space-mapping algorithm for EM optimization,” *IEEE Trans. Microwave Theory Tech.*, vol. 47, pp. 2440-2449.

M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez and J. Søndergaard (2000), “Space mapping optimization of microwave circuits exploiting surrogate models,” *IEEE MTT-S Int. Microwave Symp. Digest* (Boston, MA), pp. 1785-1788.

M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (2000a), “Neural space mapping optimization of EM microwave structures,” *IEEE MTT-S Int. Microwave Symp. Digest* (Boston, MA), pp. 879-882.

M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (2000b), “Neural space mapping optimization for EM-based design of RF and microwave circuits,” *First Int. Workshop on Surrogate Modeling and Space Mapping for Engineering Optimization* (Lyngby, Denmark).

M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (2000c), “Neural space mapping optimization for EM-based design,” *IEEE Trans. Microwave Theory Tech.*, vol. 48, 2000, pp. 2307-2315.

M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez and J. Søndergaard (2000), “Space mapping optimization of microwave circuits exploiting surrogate models,” *IEEE Trans. Microwave Theory Tech.*, vol. 48, 2000, pp. 2297-2306.

M.H. Bakr, J.W. Bandler, Q.S. Cheng, M.A. Ismail and J.E. Rayas-Sánchez (2001), “SMX—A novel object-oriented optimization system,” *IEEE MTT-S Int. Microwave Symp. Digest* (Phoenix, AZ), pp. 2083-2086.

J.W. Bandler (1969), “Optimization methods for computer-aided design,” *IEEE Trans. Microwave Theory Tech.*, vol. 17, pp. 533-552.

J.W. Bandler and S.H. Chen (1988), “Circuit optimization: the state of the art,” *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 424-443.

J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and S. Ye (1993), “Yield-driven electromagnetic optimization via multilevel multidimensional models,” *IEEE Trans. Microwave Theory Tech.*, vol. 41, pp. 2269-2278.

J.W. Bandler, S.H. Chen, R.M. Biernacki, L. Gao, K. Madsen and H. Yu (1993), “Huber optimization of circuits: a robust approach,” *IEEE Trans. Microwave Theory Tech.*, vol. 41, pp. 2279-2287.

J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers (1994), “Space mapping technique for electromagnetic optimization,” *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 2536-2544.

J.W. Bandler, R.M. Biernacki, S.H. Chen and P.A. Grobelny (1994), “A CAD environment for performance and yield driven circuit design employing electromagnetic field simulators,” *Proc. IEEE Int. Symp. Circuits Syst.* (London, England), vol. 1, pp. 145-148.

J.W. Bandler, R.M. Biernacki, S.H. Chen, W.J. Getsinger, P.A. Grobelny, C. Moskowitz and S.H. Talisa (1995), “Electromagnetic design of high-temperature superconducting microwave filters,” *Int. J. Microwave and Millimeter-Wave CAE*, vol. 5, pp. 331-343.

J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers and K. Madsen (1995), “Electromagnetic optimization exploiting aggressive space mapping,” *IEEE Trans. Microwave Theory Tech.*, vol. 41, pp. 2874-2882.

J.W. Bandler, R.M. Biernacki and S.H. Chen (1996), “Parameterization of arbitrary geometrical structures for automated electromagnetic optimization,” *IEEE MTT-S Int. Microwave Symp. Digest* (San Francisco, CA), pp. 1059-1062.

J.W. Bandler, R.M. Biernacki, S.H. Chen and D. Omeragić (1999), “Space mapping optimization of waveguide filters using finite element and mode-matching electromagnetic simulators,” *Int. J. RF and Microwave CAE*, vol. 9, pp. 54-70.

J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (1999a), “Neuromodeling of microwave circuits exploiting space mapping technology,” *IEEE MTT-S Int. Microwave Symp. Digest* (Anaheim, CA), pp. 149-152.

J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (1999b), “New directions in model development for RF/microwave components utilizing artificial neural networks and space mapping,” *IEEE AP-S Int. Symp. Digest* (Orlando, FL), pp. 2572-2575.

J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (1999c), "Neuromodeling of microwave circuits exploiting space mapping technology," *IEEE Trans. Microwave Theory Tech.*, vol. 47, pp. 2417-2427.

J.W. Bandler, J.E. Rayas-Sánchez and Q.J. Zhang (1999a), "Space mapping based neuromodeling of high frequency circuits," *Micronet Annual Workshop* (Ottawa, ON), pp. 122-123.

J.W. Bandler, J.E. Rayas-Sánchez and Q.J. Zhang (1999b), "Neural modeling and space mapping: two approaches to circuit design," *XXVI URSI General Assembly* (Toronto, ON), pp. 246.

J.W. Bandler, N. Georgieva, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (1999), "A generalized space mapping tableau approach to device modeling," *European Microwave Conf.* (Munich, Germany), vol. 3, pp. 231-234.

J.W. Bandler, N. Georgieva, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (2001), "A generalized space mapping tableau approach to device modeling," *IEEE Trans. Microwave Theory Tech*., vol. 49, pp. 67-79.

J.W. Bandler, J.E. Rayas-Sánchez, F. Wang and Q.J. Zhang (2000), "Realizations of Space Mapping based neuromodels of microwave components," *AP2000 Millennium Conf. on Antennas & Propagation* (Davos, Switzerland), vol. 1, pp. 460.

J.W. Bandler, J.E. Rayas-Sánchez and Q.J. Zhang (2000), "Software implementation of space mapping based neuromodels of microwave components," *Micronet Annual Workshop* (Ottawa, ON), pp. 67-68.

J.W. Bandler, M.A. Ismail and J.E. Rayas-Sánchez (2000a), "Broadband physics-based modeling of microwave passive devices through frequency mapping," *IEEE MTT-S Int. Microwave Symp. Digest* (Boston, MA), pp. 969-972.

J.W. Bandler, M.A. Ismail and J.E. Rayas-Sánchez (2000b), "Microwave device modeling exploiting generalized space mapping," *First Int. Workshop on Surrogate Modeling and Space Mapping for Engineering Optimization* (Lyngby, Denmark).

J.W. Bandler, M.A. Ismail and J.E. Rayas-Sánchez (2001a), "Expanded space mapping design framework exploiting preassigned parameters," *IEEE MTT-S Int. Microwave Symp. Digest* (Phoenix, AZ), pp. 1151-1154.

J.W. Bandler, M.A. Ismail and J.E. Rayas-Sánchez (2001b), "Broadband physics-based modeling of microwave passive devices through frequency mapping," *Int. J. RF and Microwave CAE*, vol. 11, pp. 156-170.

J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang (2001), "Neural inverse space mapping EM-optimization," *IEEE MTT-S Int. Microwave Symp. Digest* (Phoenix, AZ), pp. 1007-1010.

J.W. Bandler, J.E. Rayas-Sánchez and Q.J. Zhang (2001a), "Space mapping based neuromodeling of high frequency circuits," *Micronet Annual Workshop* (Ottawa, ON), pp. 69-70.

J.W. Bandler, J.E. Rayas-Sánchez and Q.J. Zhang (2001b), "Yield-driven electromagnetic optimization via space mapping-based neuromodels," *Int. J. RF and Microwave CAE*.

R.M. Biernacki, J.W. Bandler, J. Song and Q.J. Zhang (1989), "Efficient quadratic approximation for statistical design," *IEEE Trans. Circuit Syst.*, vol. 36, pp. 1449-1454.

F.H. Branin (1962), "DC and transient analysis of networks using a digital computer," *IRE Intern. Conv. Rec.*, pp. 236-256.

F.H. Branin, G.R. Hogsett, R.L. Lunde and L.E. Kugel (1971), "ECAP II – a new electronic circuit analysis program," *IEEE J. Solid-Sate Circuits*, Vol. 6, pp. 146-166.

P. Burrascano, M. Dionigi, C. Fancelli and M. Mongiardo (1998), "A neural network model for CAD and optimization of microwave filters," *IEEE MTT-S Int. Microwave Symp. Digest* (Baltimore, MD), pp. 13-16.

P. Burrascano and M. Mongiardo (1999), "A review of artificial neural networks applications in microwave CAD," *Int. J. RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, pp. 158-174.

D.A. Calahan (1965), "Computer design of linear frequency selective networks," *Proc. IEEE*, vol. 53, pp. 1701-1706.

R. Camposano and M. Pedram (2000), "Electronic design automation at the turn of the century: accomplishments and vision of the future," *IEEE Trans. Computer-Aided Design of I.C. Syst.*, vol. 19, pp. 1401-1403.

J. Carroll and K. Chang (1996), "Statistical computer-aided design for microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 17, pp. 24-32.

C.E. Christoffersen, U.A. Mughal and M.B. Steer (2000), "Object oriented microwave circuit simulation," *Int. J. RF and Microwave CAE*, vol. 10, pp. 164-182.

C. Cho and K.C. Gupta (1999), "EM-ANN modeling of overlapping open-ends in multilayer lines for design of bandpass filters," *IEEE AP-S Int. Symp. Digest* (Orlando, FL), pp. 2592-2595.

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

C.A. Desoer and S.K. Mitra (1961), “Design of lossy ladder filters by digital computer,” *IRE Trans. Circuit Theory*, vol. 8, pp. 192-201.

V.K. Devabhaktuni, C. Xi, F. Wang and Q.J. Zhang (1999), “Robust training of microwave neural models,” *IEEE MTT-S Int. Microwave Symp. Digest* (Anaheim, CA), pp. 145-148.

S.W. Director (1973), *Computer-Aided Circuit Design, Simulation and Optimization: Benchmark Papers in Electrical Engineering and Computer Science,* Stroudsburg, PA: Dowden, Hutchinson & Ross.

A. Dounavis, E. Gad, R. Achar and M.S. Nakhla (2000), “Passive model reduction of multiport distribuited interconnects,” *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2325-2334.

em™ (1997) Version 4.0b*,* Sonnet Software, Inc., 1020 Seventh North Street, Suite 210, Liverpool, NY 13088.

Empipe™ (1997) Version 4.0, formerly Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7, now Agilent EEsof EDA, 1400 Fountaingrove Parkway Santa Rosa, CA 95403-1799.

Y. Fang, M.C.E. Yagoub, F. Wang and Q.J. Zhang (2000), “A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks,” *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2335-2344.

L. Geppert (2000), “Electronic design automation,” *IEEE Spectrum*, vol. 37, pp. 70-74.

P.A. Grobelny (1995), *Integrated Numerical Modeling Techniques for Nominal and Statistical Circuit Design*, Ph.D. Thesis (Supervisor: J.W. Bandler), Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada.

K.C. Gupta, R. Garg and I.J. Bahl (1979), *Microstrip Lines and Slotlines*. Dedham, MA: Artech House.

R.F. Harrington (1967), “Matrix methods for field problems,” *Proc. IEEE,* vol. 55, pp. 136-149.

S. Haykin (1999), *Neural Networks: A Comprehensive Foundation*. New Jersey, MA: Prentice Hall.

T.S. Horng, C.C. Wang and N.G. Alexopoulos (1993) “Microstrip circuit design using neural networks,” *IEEE MTT-S Int. Microwave Symp. Digest* (Atlanta, GA), pp. 413-416.

HP ADS (1999) Version 1.1, Agilent EEsof EDA, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799.

IBM (1965), “1620 electronic circuit analysis program [ECAP] [1620-EE-02X] user’s manual,” *IBM Application Program File* H20-0170-1.

J. Katzenelson (1966), “AEDNET: a simulator for nonlinear networks,” *Proc. IEEE*, vol. 54, pp. 1536-1552.

R.R. Mansour (2000), “An engineering perspective of microwave CAD design tools,” workshop notes on Automated Circuit Optimization Using Electromagnetic Simulators, *IEEE MTT-S Int. Microwave Symp*. (Boston, MA).

B. Martin (1999), “Electronic design automation: analysis and forecast,” *IEEE Spectrum*, vol. 36, pp. 57-61.

Matlab™ (1998), Version 5.2, The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-9889.

Matlab™ Optimization Toolbox (1999), Version 2, The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098.

Matlab™ Neural Network Toolbox (1998), Version 3, The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098.

M.S. Mirotznik and D. Prather (1997), “How to choose EM software.” *IEEE Spectrum*, vol. 34, pp. 53-58.

L. Nagel and R. Rohrer (1971), “Computer analysis of nonlinear circuits, excluding radiation (CANCER),” *IEEE J. Solid-Sate Circuits*, vol. 6, pp. 166-182.

NeuroModeler (1998) Version 1.0, Prof. Q.J. Zhang, Dept. of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S 5B6.

NeuroModeler (1999) Version 1.2b, Prof. Q.J. Zhang, Dept. of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S 5B6.

OSA90 (1990) Version 1.0, Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7.

OSA90/hope™ (1997) Version 4.0, formerly Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7, now Agilent EEsof EDA, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799.

C. Pottle (1965), “Comprehensive active network analysis by digital computer: a state-space approach,” *Proc. Third Ann. Allerton Conf. Circuits and System Theory*, pp. 659-668.

M. Pozar (1998), *Microwave Engineering*. Amherst, MA: John Wiley and Sons.

J.C. Rautio and R.F. Harrington (1987), “An electromagnetic time-harmonic analysis of shielded microstrip circuits,” *IEEE Trans. Microwave Theory Tech*., vol. 35, pp. 726-730.

P. Silvester (1969), “A general high-order finite-element waveguide analysis program,” *IEEE Trans. Microwave Theory Tech*., vol. 17, pp. 204-210.

B.R. Smith and G.C. Temes (1965), “An interactive approximation procedure for automatic filter synthesis,” *IEEE Trans. Circuit Theory*, vol. 12, pp. 107-112.

J. Song (1991), *Advances in Yield-Driven Design of Microwave Circuits*, Ph.D. Thesis (Supervisor: J.W. Bandler), Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada.

C.J. Stone, (1982) “Optimal global rates of convergence for nonparametric regression,” *Ann. Stat.*, vol. 10, pp. 1040-1053.

Super-Compact (1986), formerly Compact Software, Communications Consulting Corp., Upper Saddle River, NJ, now Ansoft, 201 McLean Blvd, Paterson, NJ 07504.

D.G. Swanson (1991), “Simulating EM fields,” *IEEE Spectrum*, vol. 28, pp. 34-37.

D.G. Swanson (1998), *EM Field Simulators Made Practical*. Lowel, MA: Corporate R&D Group, M/A-COM Division of AMP.

Touchstone (1985), EEsof Inc., Westlake Village, CA, now Agilent EEsof EDA, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799.

P.W. Tuinenga, (1992), *SPICE: A Guide to Circuit Simulation and Analysis Using Pspice*. Englewood Cliffs, NJ: Prentice Hall.

A. Veluswami, M.S. Nakhla and Q.J. Zhang (1997), “The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits,” *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 712-723.

F. Wang and Q.J. Zhang (1997), “Knowledge based neuromodels for microwave design,” *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2333-2343.

F. Wang (1998), *Knowledge Based Neural Networks for Microwave Modeling and Design*, Ph.D. Thesis (Supervisor: Q.J. Zhang), Department of Electronics, Carleton University, Ottawa, Canada.

P.M. Watson and K.C. Gupta (1996), “EM-ANN models for microstrip vias and interconnects in multilayer circuits,” *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2495-2503.

P.M. Watson and K.C. Gupta (1997), “Design and optimization of CPW circuits using EM-ANN models for CPW components,” *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2515-2523.

P.M. Watson, K.C. Gupta and R.L. Mahajan (1998), “Development of knowledge based artificial neural networks models for microwave components,” *IEEE MTT-S Int. Microwave Symp. Digest* (Baltimore, MD), pp. 9-12.

P.M. Watson, G.L. Creech and K.C. Gupta (1999), “Knowledge based EM-ANN models for the design of wide bandwidth CPW patch/slot antennas,” *IEEE AP-S Int. Symp. Digest* (Orlando, FL), pp. 2588-2591.

P.M. Watson, C. Cho and K.C. Gupta (1999), “Electromagnetic-artificial neural network model for synthesis of physical dimensions for multilayer asymmetric coupled transmission structures,” *Int. J. RF and Microwave CAE*, vol. 9, pp. 175-186.

H. White, A.R. Gallant, K. Hornik, M. Stinchcombe and J. Wooldridge (1992), *Artificial Neural Networks: Approximation and Learning Theory*. Oxford, UK: Blackwell.

K.S. Yee (1966), “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media”, *IEEE Trans. on Antennas and Propagation,* vol. 13, pp. 302-307.

A.H. Zaabab, Q.J. Zhang and M.S. Nakhla (1995), “A neural network modeling approach to circuit optimization and statistical design,” *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349-1358.

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

AUTHOR INDEX

SUBJECT INDEX