

Advanced Neural Network Modeling and Electromagnetic Optimization Approaches for Microwave Passive Components

by

Jing Jin, B.Eng

A thesis submitted to
the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Electronic Science and Technology, School of Microelectronics
Tianjin University, Tianjin, China

Ottawa-Carleton Institute for Electrical and Computer Engineering
Carleton University, Ottawa, Ontario, Canada

© 2021

Jing Jin

Abstract

Microwave modeling and optimization techniques play important roles in electromagnetic (EM)-based microwave component design. The purpose of this thesis is to propose advanced neural network modeling and EM optimization approaches for microwave passive components. This thesis first proposes a new deep neural network technique to solve high-dimensional microwave modeling problems which are more challenging than those solved by previous shallow neural networks. A smooth rectified linear unit (ReLU) is proposed for the new deep neural network. The proposed deep neural network employs both the sigmoid function and the smooth ReLU as activation functions. An advanced three-stage deep learning algorithm is proposed to train the new deep neural network model. This algorithm can determine the number of hidden layers with sigmoid functions and those with smooth ReLUs in the training process. It can also overcome the vanishing gradient problem for training the deep neural network. The proposed deep neural network technique can solve microwave modeling problems in higher dimension than the previous neural network method, i.e., shallow neural network method. Compared to the standard deep neural network using only ReLUs, the proposed deep neural network can achieve higher accuracy with fewer number of hidden neurons. The proposed deep neural network technique is a useful surrogate modeling technique for microwave components.

Based on the investigation of the surrogate modeling technique, this thesis further studies the efficient surrogate-based EM optimization method and proposes an advanced cognition-driven EM optimization incorporating transfer function-based feature surrogate for EM geometry optimization of microwave filters. The proposed

optimization technique addresses the situations where the response at the starting point for the design optimization is substantially misaligned with the design specifications. We propose to extract transfer function-based feature parameters for optimization to address the challenge that the features cannot be clearly and explicitly identified from the filter response. Multiple transfer function-based feature parameters are extracted and used to develop the feature surrogate model for the proposed cognition-driven optimization. Furthermore, we derive new objective functions for the cognition-driven optimization directly in the feature space. The proposed cognition-driven optimization incorporating transfer function-based feature surrogate can achieve faster convergence than the existing feature-assisted EM optimization methods.

Moreover, to address the EM topology optimization that our proposed cognition-driven optimization method is not applicable, this thesis further proposes an efficient EM topology optimization technique for microwave component design. In the proposed technique, the finite element method (FEM) is used for EM simulation. We propose a new method to integrate Matrix Padé via Lanczos (MPVL) and Householder formula. The proposed method combines the advantages of MPVL and Householder formula so that the effort of solving the large FEM matrix equation at many frequencies is reduced to the effort of solving only a small matrix problem at a single frequency point, thereby speeding up the topology optimization process. We further propose a new method to reduce the small matrix problem into an even smaller one by exploiting the inheritance pattern of genetic algorithm (GA). New formulations are derived to exploit the common factors between new children EM

structures and the parent EM structures in each generation of GA so that computational expense can be further reduced. The proposed technique incorporating advanced FEM-MPVL and the inheritance pattern of GA can greatly accelerate the topology optimization process. All the proposed approaches in this thesis are illustrated by several microwave examples.

KEY WORDS: Microwave modeling, electromagnetic optimization, deep neural networks, feature parameters, surrogate-based optimization, electromagnetic topology optimization.

To My Parents

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr. Qi-Jun Zhang, for his continued support, professional guidance, constant encouragement, and helpful suggestions during the course of this research work. His innovative insights and rigorous academic attitude have made my research journey a memorable one. It was my great honor to work under his supervision and guidance.

I would also like to express my deep appreciation to my co-supervisor, Dr. Jianguo Ma, for his support and advice on this research work. His invaluable counsel provided great help on my way accomplishing this research work.

I would like to acknowledge Dr. Mohamed Bakr, Dr. Derek McNamara, Dr. Ramachandra Achar, Dr. Qiang Liu, Dr. Yongtao Ma, Dr. Jiangtao Xu, and Dr. Feng Feng for their valuable suggestions and corrections for my thesis.

I would like to thank all my former and present colleagues, especially Dr. Lin Zhu, Dr. Shuxia Yan, Dr. Feng Feng, Dr. Chao Zhang, Dr. Weicong Na, Dr. Wenyuan Liu, Dr. Jianan Zhang, Dr. Wei Zhang, Dr. Zhihao Zhao, Shunlu Zhang, and Jinyuan Cui in our research group for their inspiring discussions and helpful suggestions.

I would also like to thank the staff in School of Microelectronics, Tianjin University and the staff in Department of Electronics, Carleton University for offering helpful and constant support during this research work.

Many thanks to my friends Jixuan Xie, Li Yu, Fan Chen, Aochen Wang, Yuan Zhang, Ruixue Wang, Wen Wen, Zhaopeng Wang, Fu Liu, and others for their friendship and encouragement.

Finally, I wish to thank my parents and my brother for their endless love, trust, support, and encouragement throughout the years of my study. A special thank to my dear friend C. Yi for the constant accompany and support.

Table of Contents

Abstract	i
Acknowledgements	v
List of Figures	xii
List of Tables	xxi
1 Introduction	1
1.1 Introduction and Motivation	1
1.2 List of Contributions	10
1.3 Thesis Organization	12
2 Literature Review	14
2.1 Review of Artificial Neural Networks for Microwave Modeling and Design	14
2.1.1 Neural Network Structures	15
2.1.2 Neural Network Training	19

2.2	Review of Deep Neural Networks	21
2.3	Review of Surrogate-Based EM Optimization Methods	23
2.3.1	EM Optimization Using Neuro-Transfer Function Surrogates	25
2.3.2	EM Optimization Using Feature-Based Surrogates	26
2.3.3	Cognition-Driven EM Optimization in Feature Space	27
2.3.4	Feature-Assisted Neuro-TF Surrogate-Based EM Optimization	28
2.4	Review of EM Topology Optimization	29
2.4.1	EM Fast Frequency Sweep to Speed up EM Topology Opti- mization	32
2.4.2	Matrix Reduction Method for EM Topology Optimization .	35
2.5	Summary	36

3 Deep Neural Network Technique for High-Dimensional Microwave

	Modeling	37
3.1	Introduction	37
3.2	The Proposed Deep Neural Network Technique for High-Dimensional Microwave Modeling	39
3.2.1	Activation Functions for the Proposed Deep Neural Network Technique	39
3.2.2	Structure of the Proposed Deep Neural Network Technique .	44
3.2.3	Proposed Deep Learning Algorithm for Training the New Deep Neural Network	48

3.2.4	Process of Developing the Deep Neural Network for Microwave Component Modeling	54
3.2.5	Discussion	56
3.3	Application Examples	58
3.3.1	Modeling of a Fourth-Order Multicoupled Cavity Filter	58
3.3.2	Modeling of a Sixth-Order Multicoupled Cavity Filter	70
3.4	Conclusion	80
4	Advanced Cognition-Driven EM Optimization Incorporating Transfer Function-Based Feature Surrogate for Microwave Filters	82
4.1	Introduction	82
4.2	Definition of the Optimization Problem	84
4.3	Proposed Cognition-Driven EM Optimization Using Transfer Function-Based Feature Surrogate	84
4.3.1	Transfer Function-Based Feature Parameter Extraction	85
4.3.2	Cognition-Driven Optimization Using Transfer Function-Based Feature Surrogate	95
4.4	Application Examples	108
4.4.1	Optimization of a Three-Pole H-Plane Filter	108
4.4.2	Optimization of a Four-Pole Waveguide Filter	114
4.5	Conclusion	119
5	Efficient EM Topology Optimization Incorporating Advanced Matrix Padé via Lanczos and Genetic Algorithm for Microwave Com-	

ponents	120
5.1 Introduction	120
5.2 The Proposed EM Topology Optimization Incorporating Advanced FEM-MPVL and GA	122
5.2.1 Proposed Method to Incorporate Householder Formula Into FEM-MPVL Equations	124
5.2.2 Efficient Method to Obtain $\mathbf{L}^{(k,m)}$ for (5.15)	132
5.2.3 Proposed Formulations to Update $[\mathbf{M}^{(k,m)}]^{-1}$ Efficiently With- out Resolving the Matrix Inversion During Optimization	136
5.2.4 The Process of the Proposed EM Topology Optimization	146
5.2.5 Discussion of Relations Between the Proposed Technique and Other Optimization Techniques	148
5.3 Application Examples	153
5.3.1 EM Topology Optimization of a Waveguide Filter	153
5.3.2 EM Topology Optimization of a Three-Pole Waveguide Filter	160
5.3.3 EM Topology Optimization of a T-Junction Waveguide Diplexer	165
5.4 Conclusion	172
6 Conclusions and Future Research	173
6.1 Conclusions	173
6.2 Future Research	175
Appendix	179

List of Figures

2.1	Multilayer perceptrons (MLP) neural network structure containing one input layer, one output layer, and one or more hidden layers. [41].	17
2.2	A recurrent neural network with feedback of delayed neural network output [41].	18
2.3	Structure of the neuro-TF surrogate model [72].	26
2.4	Feature frequency parameters \mathbf{f} and ripple height parameters \mathbf{r} illustrated using the response of a four-pole filter [78].	28
2.5	Structure of the multifeature-assisted neuro-TF surrogate model [81].	29
3.1	Two types of activation functions used in the hybrid deep neural network. (a) Sigmoid function. (b) ReLU function.	40
3.2	The comparison of the proposed smooth ReLU and the conventional non-smooth ReLU. Solid line represents the proposed smooth ReLU with $\mu = 0.05$ and dashdot line represents the conventional non-smooth ReLU.	44

3.3 Structure of the proposed hybrid deep neural network model for microwave components. The hidden layers in the proposed deep neural network are divided into two parts. N_s hidden layers close to the input layer employ sigmoid functions as neuron activation functions, while the rest N_r hidden layers utilize the proposed smooth ReLUs as neuron activation functions. 45

3.4 The flow diagram of the process for developing the proposed deep neural network model. 57

3.5 The structure of the deep neural network high-dimensional parameter-extraction model for the fourth-order filter. The model has 35 inputs, where $x_i = dB(S_{11}(f_i)), i = 1, 2, \dots, 35$. We used a 13-layer deep neural network for this example. The hidden layers include eight layers with sigmoid functions and three layers with the proposed smooth ReLUs. Each hidden layer is composed of 100 hidden neurons. All the hidden layers are shown in the box. 62

3.6 The comparison of continuity of the partial derivative $\partial M_{23}/\partial S_{11}(f_{18})$ between the hybrid deep neural network with conventional non-smooth ReLUs (i.e., the model obtained after Stage II) and that with the proposed smooth ReLUs (i.e., the model obtained after Stage III). This is for the fourth-order filter example. The horizontal axis represents S_{11} in dB at f_{18} , and the vertical axis represents the partial derivative of M_{23} with respect to S_{11} in dB at f_{18} . Solid line represents the derivative of the hybrid deep neural network with the proposed smooth ReLUs and dashdot line represents the derivative of the hybrid deep neural network with the conventional non-smooth ReLUs. The conventional non-smooth ReLU leads to discontinuity of the derivative of the model as shown in the figure, where the dashdot line has vertical rises and vertical drops. The proposed smooth ReLU can obviously improve the continuity of the derivatives of the proposed deep neural network model. 64

3.7 The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the slightly detuned fourth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21} 67

3.8 The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the highly detuned fourth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21} 69

3.9	The comparison of continuity of the partial derivative $\partial M_{55}/\partial S_{11}(f_{21})$ between the hybrid deep neural network with conventional non-smooth ReLUs (i.e., the model obtained after Stage II) and that with smooth ReLUs (i.e., the model obtained after Stage III). This is for the sixth-order filter example. The horizontal axis represents S_{11} in dB at f_{21} , and the vertical axis represents the partial derivative of M_{55} with respect to S_{11} in dB at f_{21} . Solid line represents the derivative of the hybrid deep neural network with the proposed smooth ReLUs and dashdot line represents the derivative of the hybrid deep neural network with conventional non-smooth ReLUs. The conventional non-smooth ReLU leads to discontinuity of the derivative of the model as shown in the figure, where the dashdot line has vertical rises and vertical drops. The proposed smooth ReLU can obviously improve the continuity of the derivatives of the proposed deep neural network parameter-extraction model.	73
3.10	The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the slightly detuned sixth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21}	77
3.11	The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the highly detuned sixth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21}	79

4.1	Illustration of a four-pole filter response. $f_1, f_2, f_3,$ and f_4 are the passband feature frequencies; s_1 and s_2 are the stopband feature frequencies; $b_1, b_2,$ and b_3 are the feature bandwidths; $r_1, r_2,$ and r_3 are the feature ripples; h_1 and h_2 are the boundary feature heights. . .	93
4.2	Illustration of the rectification of the feature ripples and the boundary feature heights using equations (4.21) and (4.22) with $h_p = 0.1$ and $\varphi = 0.1$	95
4.3	The mechanism of the proposed parameter extraction for transfer function-based feature parameters. The extracted five sets of transfer function-based feature parameters are passband feature frequencies $\mathbf{f}^{n,(k)}$, stopband feature frequencies $\mathbf{s}^{n,(k)}$, feature bandwidths $\mathbf{b}^{n,(k)}$, rectified feature ripples $\widehat{\mathbf{r}}^{n,(k)}$, and rectified boundary feature heights $\widehat{\mathbf{h}}^{n,(k)}$	96
4.4	The flowchart of the proposed cognition-driven optimization using transfer function-based feature surrogate.	107
4.5	Structure of the three-pole H-plane filter. Four geometrical variables are used for the proposed optimization, i.e., $\boldsymbol{\phi}_F = [L_1 \ L_2 \ W_1 \ W_2]^T$	109

4.6	<p>Comparison of EM responses between the proposed cognition-driven EM optimization and the multifeature-assisted optimization for the three-pole H-plane filter. (a) EM response at the starting point for both techniques. (b) EM responses from the starting point (dashed line) to the 3rd iteration (solid line) of the proposed optimization. (c) EM responses from the 3rd iteration (dashed line) to the final solution (solid line) after 6 iterations of the proposed optimization. (d) EM responses from the starting point (dashed line) to the 7th iteration (solid line) of the multifeature-assisted optimization. (e) EM responses from the 7th iteration (dashed line) to the final solution (solid line) after 9 iterations of the multifeature-assisted optimization. The black lines represent the specification.</p>	112
4.7	<p>Structure of the four-pole waveguide filter. Five geometrical variables are used for the proposed EM design optimization.</p>	115

4.8 Comparison of EM responses between the proposed cognition-driven EM optimization and the multifeature-assisted optimization for the four-pole waveguide filter. (a) EM response at the starting point for both techniques. (b) EM responses from the starting point (dashed line) to the 5th iteration (solid line) of the proposed optimization. (c) EM responses from the 5th iteration (dashed line) to the final solution (solid line) after 8 iterations of the proposed optimization. (d) EM responses from the starting point (dashed line) to the 11th iteration (solid line) of the multifeature-assisted optimization. (e) EM responses from the 11th iteration (dashed line) to the final solution (solid line) after 13 iterations of the multifeature-assisted optimization. The black lines represent the specification. 117

5.1 The concept of EM topology optimization as illustrated through a planar filter configuration. During optimization, the material in each cell inside the design space can independently change from metal to nonmetal (or vice versa). EM topology optimization will adjust the metal/nonmetal pattern so that the filter response is optimized. . . 123

5.2 An illustration of a one-point crossover process in GA. Two children, i.e., two new chromosomes, are generated from the crossover of genes of two parents. The genes in each child chromosome are inherited partly from the parent 1 and partly from the parent 2. 137

5.3	A simple example to illustrate the inheritance pattern of GA. By comparing the chromosome of the child (child 1) with the chromosomes of two parents, we obtain the index sets B_α^i , B_β^i , and B_ψ^i for $i = 1, 2$	140
5.4	The flowchart of the proposed EM topology optimization technique.	149
5.5	The configuration of the waveguide filter in the first example. The design space includes the two cross-sectional planes inside the waveguide. Each plane is divided into 60 cells. Each cell consists of several triangular meshes.	154
5.6	The comparison of S_{11} obtained after EM topology optimization and the initial S_{11} response for the filter in the first example. Note that the initial S_{11} response is the response of the best chromosome in the first generation. The black lines represent the specifications.	156
5.7	The optimal EM structure of the waveguide filter after EM topology optimization in the first example.	156
5.8	The configuration of the three-pole waveguide filter. The design space includes the four cross-sectional planes inside the waveguide. Each plane is divided into 60 cells. Each cell consists of several triangular meshes.	161
5.9	The comparison of S_{11} obtained after EM topology optimization and the initial S_{11} response for the three-pole filter. Note that the initial S_{11} response is the response of the best chromosome in the first generation. The black lines represent the specifications.	162

5.10	The optimal EM structure of the three-pole waveguide filter after EM topology optimization. Blue cells are covered with metals.	162
5.11	The structure of the T-junction diplexer after preliminary optimization. The widths of the coupling windows are $w_1 = 9.11$ mm, $w_2 = 6.06$ mm, $w_3 = 7.79$ mm, and $w_4 = 4.49$ mm.	166
5.12	The comparison of S_{11} after the preliminary optimization and S_{11} after the proposed EM topology optimization for the T-junction waveguide diplexer example.	167
5.13	The configuration of the T-junction diplexer for EM topology optimization. The design space includes the six cross-sectional planes inside the waveguide. Each plane is divided into 60 cells. Each cell consists of several triangular meshes.	167
5.14	The optimal EM structure of the T-junction diplexer after topology optimization. Blue cells are covered with metals.	168

List of Tables

3.1	Results of Different Stages in the Training Process of the Proposed Deep Neural Network Parameter-Extraction Model for the Fourth-Order Filter Example	61
3.2	Comparison of Shallow and Deep Neural Networks with a Similar Number of Training Parameters for Parameter-Extraction Modeling of the Fourth-Order Filter Example	65
3.3	Comparison of the Proposed Hybrid Deep Neural Network and Pure ReLU Deep Neural Network for Parameter-Extraction Modeling of the Fourth-Order Filter Example	66
3.4	Results of Different Stages in the Training Process of the Proposed Deep Neural Network Parameter-Extraction Model for the Sixth-Order Filter Example	72
3.5	Comparison of Shallow and Deep Neural Networks with a Similar Number of Training Parameters for Parameter-Extraction Modeling of the Sixth-Order Filter Example	74

3.6	Comparison of the Proposed Hybrid Deep Neural Network and Pure ReLU Deep Neural Network for Parameter-Extraction Modeling of the Sixth-Order Filter Example	75
4.1	Comparison of CPU Time of Different Optimization Methods for the Three-Pole H-Plane Filter	113
4.2	Comparison of CPU Time of Different Optimization Methods for the Four-Pole Waveguide Filter	118
5.1	Comparison of CPU Time Between the Proposed EM Topology Optimization Technique and Several Other Topology Optimization Methods for the Waveguide Filter in the First Example	157
5.2	Comparison of CPU Time Between the Proposed Topology Optimization Technique and Several Other Topology Optimization Methods for the Three-Pole Waveguide Filter Example	163
5.3	Comparison of CPU Time Between the Proposed Topology Optimization Technique and Several Other Topology Optimization Methods for the T-Junction Diplexer Example	169

List of Symbols

$\mathbf{A}_0^{(k,m)}$ The FEM-MPVL matrix at the frequency expansion point for the m th chromosome in the k th generation

$\mathbf{b}^{n,(k)}$ The vector containing all the feature bandwidths for the n th geometrical sample in the k th optimization iteration

$b_i^{n,(k)}$ The i th feature bandwidth for the n th geometrical sample in the k th optimization iteration

B^λ The index set of active neurons in the λ th layer of neurons with smooth ReLUs

B_s The set including all the frequencies at which the transfer function response $H^{n,(k)}$ equal to h_s

$B^{(k,m)}$ The index set of metal cells for the m th chromosome in the k th GA generation

B_α^i The index set of metal cells in the child EM structure that are inherited from the i th parent EM structure

B_β^i The index set of those remaining metal cells in the i th parent

- B_ψ^i The index set of other metal cells in the child that are not inherited from the i th parent
- c_1, c_2 The indices of two worst chromosomes in the k th generation
- \mathbf{d}_n The desired outputs of the neural network for the inputs \mathbf{x}_n
- d_{jn} The j th element of \mathbf{d}_n
- $\tilde{\mathbf{D}}_{\alpha\alpha}$ The $N_\alpha \times N_\alpha$ sub-matrix in the matrix $\tilde{\mathbf{M}}^{-1}$
- $\tilde{\mathbf{D}}_{\alpha\beta}$ The $N_\alpha \times N_\beta$ sub-matrix in the matrix $\tilde{\mathbf{M}}^{-1}$
- $\tilde{\mathbf{D}}_{\beta\alpha}$ The $N_\beta \times N_\alpha$ sub-matrix in the matrix $\tilde{\mathbf{M}}^{-1}$
- $\tilde{\mathbf{D}}_{\beta\beta}$ The $N_\beta \times N_\beta$ sub-matrix in the matrix $\tilde{\mathbf{M}}^{-1}$
- $\mathbf{D}_{\alpha\alpha}^{(k+1,c_1)}$ The $N_\alpha \times N_\alpha$ sub-matrix in the matrix $[\mathbf{M}^{(k+1,c_1)}]^{-1}$
- $\mathbf{D}_{\alpha\psi}^{(k+1,c_1)}$ The $N_\alpha \times N_\psi$ sub-matrix in the matrix $[\mathbf{M}^{(k+1,c_1)}]^{-1}$
- $\mathbf{D}_{\psi\alpha}^{(k+1,c_1)}$ The $N_\psi \times N_\alpha$ sub-matrix in the matrix $[\mathbf{M}^{(k+1,c_1)}]^{-1}$
- $\mathbf{D}_{\psi\psi}^{(k+1,c_1)}$ The $N_\psi \times N_\psi$ sub-matrix in the matrix $[\mathbf{M}^{(k+1,c_1)}]^{-1}$
- $\mathbf{e}^{(k,m)}$ The solution vector containing the unknown values used in approximating the electrical field for the EM structure whose shape and topology is represented by $\phi^{(k,m)}$
- $\tilde{\mathbf{e}}^{(k,m)}$ The solution of the FEM-MPVL equation

- $E(\mathbf{w})$ The error function for training the neural network model
- E_b The training error of the deep neural network before the recent addition of the sigmoid layer
- E_n The error for the n th training sample
- E_r The error threshold for the deep neural network model
- E_{train} The training error at the end of training of the deep neural network including the recently added layer
- E_{test} The test error of the deep neural network
- f_{p1} The lower frequency boundary of the passband
- f_{p2} The upper frequency boundary of the passband
- f_{s1} The lower frequency boundary of the stopband
- f_{s2} The upper frequency boundary of the stopband
- f_i^{low} The lower frequency boundary for the i th passband feature frequency
- f_i^{upp} The upper frequency boundary for the i th passband feature frequency
- $\mathbf{f}^{n,(k)}$ The vector containing all the passband feature frequencies for the n th geometrical sample in the k th optimization iteration
- $f_i^{n,(k)}$ The i th passband feature frequency for the n th geometrical sample in the k th optimization iteration

- $\mathbf{g}_n^{(k,m)}$ The vector of size $2N \times 1$ for the calculation of $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ in the MPVL iterations
- $G^{n,(k)}$ The gain factor of the transfer function extracted using vector fitting for the n th geometrical sample in the k th optimization iteration
- h_p The user pre-defined passband response specification
- h_s The user pre-defined stopband response specification
- $\mathbf{h}^{n,(k)}$ The vector containing the boundary feature heights for the n th geometrical sample in the k th optimization iteration
- $h_1^{n,(k)}$ The lower boundary feature heights for the n th geometrical sample in the k th optimization iteration
- $h_2^{n,(k)}$ The upper boundary feature heights for the n th geometrical sample in the k th optimization iteration
- $\widehat{\mathbf{h}}^{n,(k)}$ The vector containing the rectified boundary feature heights for the n th geometrical sample in the k th optimization iteration
- $\widehat{h}_i^{n,(k)}$ The i th rectified boundary feature height for the n th geometrical sample in the k th optimization iteration
- $H^{n,(k)}$ The magnitude of the transfer function response for the n th geometrical sample in the k th optimization iteration
- $J\tau$ The index set of unknowns in the τ th cell of the design space

$J^{(k,m)}$ The index set of unknowns in all the metal cells of the design space for the m th chromosome in the k th generation

\tilde{J} The index set of unknowns in all cells in the design space

$\tilde{J}(j)$ The j th element in the index set \tilde{J}

J_α^i The index set of the unknowns in the metal cells that are inherited from the i th parent

J_β^i The index set of the unknowns in those remaining metal cells in the i th parent

J_ψ^i The index set of the unknowns in other metal cells not inherited from the i th parent

\tilde{J}_α The index sets formulated in (5.24) corresponding to the chosen parent $\tilde{\phi}$

\tilde{J}_β The index sets formulated in (5.26) corresponding to the chosen parent $\tilde{\phi}$

\tilde{J}_ψ The index sets formulated in (5.28) corresponding to the chosen parent $\tilde{\phi}$

k The index of the surrogate optimization iteration or the generation counter of GA optimization

$\mathbf{K}_i^{(k,m)}$ The FEM matrices ($i = 0, 1, 2$) of size $N \times N$ for the m th chromosome in the population of k th generation

\mathbf{K}_s The FEM matrix formulated at the frequency expansion point for the base EM structure as $\mathbf{K}_s = \mathbf{K}_0^s + \gamma_0 \mathbf{K}_1^s + \gamma_0^2 \mathbf{K}_2^s$

- \mathbf{K}_i^s The FEM matrices ($i = 0, 1, 2$) for the base EM structure
- $\Delta \mathbf{K}^{(k,m)}$ An $N \times N$ diagonal matrix corresponding to the m th chromosome in the k th generation
- $\Delta K_{i,j}^{(k,m)}$ The (i, j) th element in the matrix $\Delta \mathbf{K}^{(k,m)}$
- $\mathbf{L}^{(k,m)}$ The solution of the matrix equation $\mathbf{K}_s \mathbf{L}^{(k,m)} = \mathbf{Q}^{(k,m)}$
- $\tilde{\mathbf{L}}$ The solution of the matrix equation $\mathbf{K}_s \tilde{\mathbf{L}} = \tilde{\mathbf{Q}}$
- m The index of the chromosome in the population of GA optimization
- m_1, m_2 The indices of two parent chromosomes selected from the k th generation
- $\mathbf{M}^{(k,m)}$ The product of $[\mathbf{Q}^{(k,m)}]^T \mathbf{L}^{(k,m)}$
- $\tilde{\mathbf{M}}$ The matrix defined in (5.16) corresponding to the chosen parent EM structure $\tilde{\phi}$
- $\tilde{\mathbf{M}}_{\alpha\alpha}$ The $N_\alpha \times N_\alpha$ sub-matrix in the matrix $\tilde{\mathbf{M}}$
- $\tilde{\mathbf{M}}_{\alpha\beta}$ The $N_\alpha \times N_\beta$ sub-matrix in the matrix $\tilde{\mathbf{M}}$
- $\tilde{\mathbf{M}}_{\beta\alpha}$ The $N_\beta \times N_\alpha$ sub-matrix in the matrix $\tilde{\mathbf{M}}$
- $\tilde{\mathbf{M}}_{\beta\beta}$ The $N_\beta \times N_\beta$ sub-matrix in the matrix $\tilde{\mathbf{M}}$
- $\mathbf{M}_{\alpha\alpha}^{(k+1,c_1)}$ The $N_\alpha \times N_\alpha$ sub-matrix in the matrix $\mathbf{M}^{(k+1,c_1)}$
- $\mathbf{M}_{\alpha\psi}^{(k+1,c_1)}$ The $N_\alpha \times N_\psi$ sub-matrix in the matrix $\mathbf{M}^{(k+1,c_1)}$

- $\mathbf{M}_{\psi\alpha}^{(k+1,c_1)}$ The $N_\psi \times N_\alpha$ sub-matrix in the matrix $\mathbf{M}^{(k+1,c_1)}$
- $\mathbf{M}_{\psi\psi}^{(k+1,c_1)}$ The $N_\psi \times N_\psi$ sub-matrix in the matrix $\mathbf{M}^{(k+1,c_1)}$
- n The index of the training sample or the geometrical sample
- N The number of elements in the unknown solution vector $\mathbf{e}^{(k,m)}$
- N_e The effective order of the transfer function for all the geometrical samples
- N_f The number of passband feature frequencies
- N_g The total number of geometrical samples for training the surrogate model
- N_λ The number of hidden neurons in the λ th layer, where $\lambda = 1, 2, \dots, \Gamma$
- N_{GA} The maximum number of generations of GA
- N_{or} The level of the orthogonal distribution
- N_p The number of chromosomes in each generation of GA
- N_r The number of hidden layers with smooth ReLUs
- N_s The number of hidden layers with sigmoid functions
- N_{tr} The total number of training samples for the neural network model
- N_v The number of design variables
- N_x The number of the neural network inputs
- N_y The number of the neural network outputs

- $N_J^{(k,m)}$ The number of indices in the index set $J^{(k,m)}$
- $N_{\tilde{J}}$ The number of elements in \tilde{J} .
- N_α^i The number of elements in the index sets J_α^i
- N_β^i The number of elements in the index sets J_β^i
- N_ψ^i The number of elements in the index sets J_ψ^i
- N_α The number of elements in the index sets \tilde{J}_α
- N_β The number of elements in the index sets \tilde{J}_β
- N_ψ The number of elements in the index sets \tilde{J}_ψ
- $\mathbf{p}^{n,(k)}$ The vector of effective poles (i.e., poles with positive imaginary parts) obtained from the n th geometrical sample in the k th optimization iteration
- $p_i^{n,(k)}$ The i th effective pole for the n th geometrical sample in the k th optimization iteration
- $p_i^{n,(k)*}$ The conjugate pair of $p_i^{n,(k)}$
- $\mathbf{P}^{(k)}$ The trust radius which is a vector consisting of the ranges of all the design variables in the k th iteration
- $P_i^{(k)}$ The i th element in the trust radius $\mathbf{P}^{(k)}$

- $\mathbf{Q}^{(k,m)}$ An $N \times N_j^{(k,m)}$ matrix corresponding to indices of unknowns in the metal cells of the design space for the m th chromosome in the population of k th generation
- $\mathbf{q}_j^{(k,m)}$ The j th column in the matrix $\mathbf{Q}^{(k,m)}$
- $\tilde{\mathbf{Q}}$ The matrix containing columns corresponding to indices of unknowns in all cells of the design space
- $\tilde{Q}_{i,j}$ The (i, j) th element in the matrix $\tilde{\mathbf{Q}}$
- $\mathbf{r}^{n,(k)}$ The vector containing all the feature ripples for the n th geometrical sample in the k th optimization iteration
- $r_i^{n,(k)}$ The i th feature ripple for the n th geometrical sample in the k th optimization iteration
- $\hat{\mathbf{r}}^{n,(k)}$ The vector containing all the rectified feature ripples for the n th geometrical sample in the k th optimization iteration
- $\hat{r}_i^{n,(k)}$ The i th rectified feature ripple for the n th geometrical sample in the k th optimization iteration
- \mathbf{R} The vector describing the EM excitation at the input port
- $\mathbf{s}^{n,(k)}$ The vector containing stopband feature frequencies for the n th geometrical sample in the k th optimization iteration

$s_1^{n,(k)}$ The lower stopband feature frequency for the n th geometrical sample in the k th optimization iteration

$s_2^{n,(k)}$ The upper stopband feature frequency for the n th geometrical sample in the k th optimization iteration

\mathbf{S}_F The EM responses of the fine model (e.g., EM simulation)

$\mathbf{S}(\phi)$ The S-parameters of EM responses for the microwave component whose shape and topology is represented by ϕ

$\mathbf{T}^{(k,m)}$ The reduced-order matrix for the m th chromosome in the k th generation

$t_{\xi,\eta}^{(k,m)}$ The (ξ, η) th element in the matrix $\mathbf{T}^{(k,m)}$

$\mathbf{u}_\eta^{(k,m)}$ The vector representing the solution of (5.12) for an EM structure with arbitrary shape and topology

$\tilde{\mathbf{u}}_\eta^{(k,m)}$ The vector representing the solution of (5.12) for the base EM structure. $\tilde{\mathbf{u}}_\eta^{(k,m)}$ is obtained from (5.18) and is used to calculate $\mathbf{u}_\eta^{(k,m)}$

$\tilde{\mathbf{u}}_0$ The vector used to denote $\tilde{\mathbf{u}}_0^{(k,m)}$

$\tilde{\mathbf{z}}_\eta^{(k,m)}$ The vector assembled by selecting $N_J^{(k,m)}$ elements from the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$

U A user-defined objective function

U_b The objective function for \mathbf{Y}^b

U_f The objective function for \mathbf{Y}^f

- U_h The objective function for \mathbf{Y}^h
- U_r The objective function for \mathbf{Y}^r
- U_s The objective function for \mathbf{Y}^s
- \tilde{U} The overall objective function for the transfer function-based feature surrogate model
- $\mathbf{V}^{(k,m)}$ The $2N \times \rho$ matrix containing the orthonormal basis of the Krylov subspaces for the model order reduction
- $\mathbf{v}_\eta^{(k,m)}$ The η th column in the matrix $\mathbf{V}^{(k,m)}$
- $\tilde{\mathbf{v}}_\eta^{(k,m)}$ The first half of the vector $\mathbf{v}_\eta^{(k,m)}$
- \mathbf{w} The vector containing all the weights in the neural network
- w_{ij}^λ The weight between the i th neuron of the λ th layer and the j th neuron of the $(\lambda - 1)$ th layer
- $\mathbf{w}^{(k)}$ The optimal values of weighting parameters in the surrogate model at the k th optimization iteration
- W_i The parameter used to determine which parent should be chosen as $\tilde{\phi}$
- \mathbf{x} The vector containing external inputs to the neural network model
- x_i The i th input to the neural network model
- \mathbf{x}_n The input vector of the n th training data

- X The total input to the neuron in a neural network
- X_i^λ The total input to the i th neuron in the λ th layer
- \mathbf{y} The vector containing outputs of the neural network model
- y_i The i th output of the neural network model
- $\mathbf{Y}_{dat}^{n,(k)}$ The vector containing the data of the five sets of transfer function-based feature parameters for the n th geometrical sample in the k th optimization iteration
- \mathbf{Y} The output vector of the transfer function-based feature surrogate model
- \mathbf{Y}^b The feature bandwidths in the output vector of the transfer-function based feature surrogate model
- Y_i^b The i th feature bandwidth in \mathbf{Y}^b
- \mathbf{Y}^f The passband feature frequencies in the output vector of the transfer-function based feature surrogate model
- Y_i^f The i th passband feature frequency in \mathbf{Y}^f
- \mathbf{Y}^h The rectified boundary feature heights in the output vector of the transfer-function based feature surrogate model
- Y_i^h The i th boundary feature height in \mathbf{Y}^h
- \mathbf{Y}^r The rectified feature ripples in the output vector of the transfer-function based feature surrogate model

- Y_i^r The i th feature ripple in \mathbf{Y}^r
- \mathbf{Y}^s The stopband feature frequencies in the output vector of the transfer-function based feature surrogate model
- Y_1^s The first stopband feature frequency in \mathbf{Y}^s
- Y_2^s The second stopband feature frequency in \mathbf{Y}^s
- $\mathbf{z}^{n,(k)}$ The vector of effective zeros (i.e., zeros with positive imaginary parts) obtained from the n th geometrical sample in the k th optimization iteration
- $z_i^{n,(k)}$ The i th effective zero for the n th geometrical sample in the k th optimization iteration
- $z_i^{n,(k)*}$ The conjugate pair of $z_i^{n,(k)}$
- $\hat{\mathbf{z}}^{n,(k)}$ The vector containing the selected N_f zeros of the transfer function for the n th geometrical sample in the k th optimization iteration
- Z_i^λ The output of the i th hidden neuron in the λ th layer
- γ The complex propagation constant
- γ_0 The complex propagation constant at the frequency expansion point
- Γ The total number of layers of the proposed deep neural network
- δ_i^λ The local gradient of E_n with respect to the total input to the i th neuron of λ th layer X_i^λ

- ε A user-defined threshold in the cognition-driven optimization
- ζ_i The size of the matrix equation solved to extract $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$
- ι A very large number (e.g., 10^6)
- $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_\rho$ A set of orthonormal vectors of size $2N \times 1$
- λ The index of layers in the deep neural network, where $\lambda = 1, 2, \dots, \Gamma$
- μ The parameter to determine the range in which the non-smooth part of the conventional ReLU is replaced with a quadratic function
- ρ The order of the reduced-order model
- ς_e, ς_c The coefficients for the trust radius update
- $\sigma_s(\cdot)$ The sigmoid function
- $\sigma'_s(\cdot)$ The derivative of the sigmoid function
- $\sigma_r(\cdot)$ The rectified linear unit (ReLU)
- $\sigma'_r(\cdot)$ The derivative of ReLU
- $\sigma_q(\cdot)$ The quadratic function used to replace the non-smooth part of the ReLU for X in the range of $[-\mu, \mu]$
- $\sigma_{pr}(\cdot)$ The proposed smooth ReLU
- $\sigma'_{pr}(\cdot)$ The derivative of the proposed smooth ReLU

- τ The index of the small cell in the design space
- Υ_1 The weight for U_b in the overall objective function \tilde{U}
- Υ_2 The weight for U_r in the overall objective function \tilde{U}
- Υ_3 The weight for U_h in the overall objective function \tilde{U}
- φ The weight for $r_i^{n,(k)}$ and $h_i^{n,(k)}$ when performing the rectification
- ϕ_F The vector of geometrical design variables in fine model
- ϕ_F^* The vector representing the optimal geometrical design solution
- $\phi_F^{(k)}$ The vector representing the nominal point of geometrical design variables in the k th iteration
- $\phi_F^{n,(k)}$ The vector representing the n th geometrical sample in the k th optimization iteration
- $\phi_i^{n,(k)}$ The i th element in the vector $\phi_F^{n,(k)}$
- ϕ The vector consisting of the binary design variables for EM topology optimization. It represents the shape and topology of metal in the design space.
- ϕ_τ The binary design variable representing the material in the τ th cell of the design space
- ϕ^* The vector representing the optimal solution of binary design variables in EM topology optimization.

- $\phi^{(k,m)}$ The vector of binary design variables used as the m th chromosome in the population of k th generation
- $\phi_{\tau}^{(k,m)}$ The τ th element of $\phi^{(k,m)}$
- $\phi^{(k,m_1)}$ The vector of binary design variables representing the first parent EM structure (i.e., parent 1) selected from the population of k th generation
- $\phi^{(k,m_2)}$ The vector of binary design variables representing the second parent EM structure (i.e., parent 2) selected from the population of k th generation
- $\phi^{(k+1,c_1)}$ The vector of binary design variables representing the first child EM structure (i.e., child 1) generated by two selected parent EM structures for the $(k+1)$ th generation
- $\phi^{(k+1,c_2)}$ The vector of binary design variables representing the second child EM structure (i.e., child 2) generated by two selected parent EM structures for the $(k+1)$ th generation
- $\tilde{\phi}$ The vector of binary design variables representing the chosen parent EM structure
- χ The adjustment control index parameter used to determine the update of the trust radius
- Ψ A pre-determined constant used to determine the increasing rate of Υ_2 during the optimization process
- ω The angular frequency

$\Omega_s^{(k)}$ The trust region of geometrical parameters of the surrogate model in the k th iteration

$\|\cdot\|$ L2 norm

Chapter 1

Introduction

1.1 Introduction and Motivation

Microwave passive components have been widely used in many modern electronic devices/systems such as laptops, smart phones, mobile communication systems, wireless networks, communication satellites, and so on. Modeling and design of microwave passive components have become more challenging because of the demand for higher functionality, better reliability, and shorter design cycle. As a result, there is a need for more accurate, more cost-effective, and more efficient computer-aided design (CAD) techniques for microwave design and optimization. The success of CAD techniques relies heavily on accurate models and efficient optimization approaches. The objective of this thesis is to propose advanced neural network modeling and electromagnetic optimization approaches for microwave passive components.

Functional surrogate modeling techniques play an important role in surrogate-based microwave modeling or design [1]-[7]. The developed microwave component

models allow fast simulation and optimization and subsequently can be implemented in high-level circuit designs or computer-aided tuning of microwave components. Several parametric modeling techniques, such as neural network technique [2], [3], support vector machine (SVM) [4], [5], Kriging [6], [7], and polynomial-based surrogate modeling [1], have been reported for modeling of microwave components. SVM, Kriging, and polynomial surrogates have good generalization capability when training data are limited [1], while neural network is well suited to the case when the amount of training data is large.

Artificial neural network (ANN) has been a powerful functional surrogate modeling technique in the area of microwave modeling and design [8]-[11]. ANN can be trained to learn the nonlinear input-output relationships from corresponding data. These trained ANN models can then be used to provide fast answers to the tasks they have learned [12]. This makes ANN an efficient alternative to empirical model or electromagnetic (EM) simulation for microwave modeling. Applications have been reported in filter modeling and design [13]-[16], power amplifier modeling [17]-[20], nonlinear microwave device modeling [21]-[24], parametric modeling of microwave components [25], [26], coplanar waveguide (CPW) circuit modeling [27], and microwave component design [28]. The applications of ANNs in the area of microwave modeling and design are achieved mostly using shallow neural networks, i.e., ANNs with one or two hidden layers. In microwave modeling field, there are situations where the model inputs are in high dimension. Microwave modeling in high dimension using neural networks requires training a neural network to learn a nonlinear function with high-dimensional inputs. When the input dimension of

the nonlinear function becomes higher, the relationship between the inputs and the outputs of the microwave model will become more complicated and the modeling problem will become harder. Note that for discrete modeling problems such as image classification and speech recognition, the input dimension can be thousands or even higher. For microwave modeling problems with continuous input-output relationships, dimension beyond 10 or 20 may become challenging. In this case, there are two possible solutions to address this kind of complicated problems using neural networks. One is to add more hidden neurons to the shallow neural network, and the other is to add more hidden layers. It has been proved that the neural network with many hidden layers can perform significantly better than the shallow neural network when both neural networks have the same number of weight parameters [29], [30].

In recent years, there has been growing interest in the neural network community in neural networks with many hidden layers, known as the deep neural network [31]-[35]. It is recognized to be very powerful at modeling intricate relationships in large data sets [31]. Deep neural networks have produced outstanding results in a variety of challenging fields, such as image classification [36], speech recognition [37], language processing [38], machine translation [39], and sentiment analysis [40]. Training of deep neural networks is also called deep learning. It is an intensive topic in the neural network community. The deep learning process is not as easy as the learning process of shallow ANNs. The selection of nonlinear hidden neuron activation functions could affect the quality of training of deep neural networks. The most commonly used activation functions in conventional (shallow) neural networks are

smooth switch functions such as the sigmoid function [41]. Shallow neural networks with sigmoid activation functions usually can be trained successfully by gradient-based backpropagation algorithm from a random starting point. However, training deep neural networks with sigmoid functions using backpropagation algorithm is problematic because of the vanishing gradient problem [42], [43]. The vanishing gradient problem is the phenomenon that the gradient of the error function with respect to neural network internal weight parameters approaches zero as the gradient propagates from output layer backwards through more and more hidden layers [44]. Recently, the rectified linear unit (ReLU) becomes one of the most popular activation functions for deep neural networks because ReLU allows deep neural networks to be trained without the vanishing gradient problem [31], [45]. Deep neural networks with ReLUs have been demonstrated to be effective for speech recognition [46], image classification [33], acoustic models [47], and so on. For neural networks with ReLUs, only a subset of neurons are active for a given input. In order to reach similar accuracy as the neural network with sigmoid activation functions, a neural network with ReLUs therefore would need much more hidden neurons [45]. In addition, conventional ReLU is a hard switch function, whose output is non-smooth. Therefore, the outputs of the standard deep neural network with conventional ReLUs will not be smooth. While the standard deep neural network with non-smooth outputs is suitable for discrete input-output relationships such as speech recognition and image classification, it is not suitable for microwave modeling where the input-output relationships are smooth and continuous. Therefore, how to take advantages of the deep neural network to address the challenges in high-dimensional modeling

of microwave components still remains an open subject in the literature.

Based on the investigation of surrogate modeling techniques such as neural networks, a further research topic is surrogate-based EM optimization techniques. EM optimization plays an important role in the design of microwave components. EM-based optimizations are computationally expensive because repetitive EM simulations with different geometrical parameters as design variables are required in the optimization process. Efficient EM-based design optimization methods need to be developed to perform automated design and increase the optimization speed.

Surrogate-based optimization techniques have been introduced to speed up the EM-based optimization [48]-[51]. Space mapping is one of the popularly used techniques for surrogate-based EM optimization [52], [53]. The space mapping concept combines the computational efficiency of coarse models with the accuracy of fine models. The space mapping techniques establish mathematical links between the coarse and the fine models and direct the bulk of the CPU-intensive computations to the coarse model, while preserving the accuracy offered by the fine model. Researches on space mapping have focused on a variety of areas, such as implicit space mapping [54]-[56], input space mapping [57], [58], output space mapping [59], [60], neural space mapping [61]-[64], and practical examples using space mapping [65]-[71]. Space mapping techniques usually require the availability of coarse models.

To address the situation where the coarse model is unavailable, another popular surrogate-based method that combines the neural network and the transfer function (neuro-TF) has been developed to perform parametric modeling and design optimization for microwave devices [72]-[74]. In the neuro-TF method, EM responses of

passive microwave components versus frequency are expressed by the widely used transfer functions. Sensitivity analysis has also been involved in neuro-TF parametric modeling techniques to increase the modeling accuracy [26]. In [73] and [74], the neuro-TF methods have been introduced into EM design optimization, where gradient-based optimization is performed with neuro-TF models as surrogate models. When the response at the starting point for the design optimization is substantially misaligned with the design specifications, methods in [73] and [74] can be easily stuck in local minima. To address this issue, the global optimization technique is widely used to provide the maximum probability to find the global optimal solution [75], but requires a relatively large amount of EM simulations. An alternative efficient technique is the feature-based EM optimization [76], [77], where the feature parameters are extracted to drive EM optimization to enhance the ability to avoid being trapped in local minima and increase the optimization efficiency simultaneously. In [78], a cognition-driven method using feature parameters to drive the EM optimization is presented for microwave filter design. In [79], an enhanced cognition-driven method is reported to address the situation where the filter response of the starting point has an incorrect number of feature frequencies. To further address a situation where the features cannot be clearly and explicitly identified from the filter response, features extracted from transfer function response (transfer function-based features) are used to assist the surrogate to perform the EM optimization [80], [81]. Considering that the transfer function-based features can address the situation where the features cannot be clearly and explicitly identified from the filter response, it is of great interest to investigate the use of the transfer

function-based features to perform cognition-driven optimizations.

The aforementioned surrogate-based EM optimization method is suitable for the EM geometry optimization, where the performance of the microwave component is optimized by adjusting the values of a given set of geometrical parameters, such as length and width of microwave structures [9], [82]. However, it is not suitable for the EM topology optimization where different EM structures cannot be related by changing the values of a given set of geometrical parameters. EM topology optimization is a numerical optimization method to find the optimal shape and topology of the EM structure by repetitively adding or eliminating geometrical/material elements within the design space [83], [84]. It can search for the optimal EM structure without constraints of shape and topology in the design space.

EM topology optimization has been applied in the design of microwave components to achieve higher performance or more compact configurations [84]-[86]. In EM topology optimization, the entire design space is divided into many small cells and the material composition of each cell is regarded as a design variable. Topology optimization has been formulated into either continuous optimization or binary optimization according to the variation mode of the material properties in each small cell in the design space. In [87]-[90], the material properties are allowed to vary continuously in each small cell within the design space in the topology optimization process. The topology optimizations in [91]-[93] deal with the presence or absence of the material in each small cell within the design space, i.e., the material properties are represented by binary variables. There are generally two types of optimization methods for the topology optimization, i.e., the gradient-based optimization

method [83], [86] and the gradient-free optimization method [94]-[97]. The main advantage of the gradient-based optimization method is to converge rapidly toward the nearest local optimal solution. However, it may suffer from getting stuck in the local optima. The gradient-free optimization, such as the genetic algorithm (GA), can provide the maximum probability to find the global optimal solution at the cost of higher computational expense.

EM topology optimization problem cannot be solved by the widely used surrogate-based optimization because different shapes and topologies cannot be related by changing the values of a given set of geometrical parameters. Therefore, we cannot use conventional surrogate models to represent EM solutions versus topology changes. Subsequently, repetitive EM simulations for different shapes and topologies of EM structures are needed in EM topology optimization process. Typical EM simulation methods such as finite element method (FEM) require solving matrix equations with very large matrices. In the EM topology optimization, large matrix equations (e.g. FEM matrix equations) need to be solved repetitively for different EM structures over a frequency range, which is time-consuming. The matrix partitioning method has been exploited to improve the computational efficiency for solving large matrix equations in EM topology optimization [91]. With the matrix partitioning method, the EM solution of a new EM structure can be obtained by solving a relatively small matrix equation which is associated with the elements in the design space. However, even if the matrix associated with the elements in the design space is smaller than the original large matrix, the time for solving this relatively small matrix equation is not negligible (since the matrix size is still sev-

eral thousand). This small matrix equation needs to be solved at each and every frequency to obtain the EM solution over a frequency range, which also limits the computational efficiency. Besides, in EM topology optimization using the matrix partitioning method, the partitioned matrix equation associated with the constant elements needs to be pre-solved and stored for each frequency, which will cause a huge memory cost. As a consequence, there is a demand for efficient methods to reduce the computational cost of the EM topology optimization process.

In this thesis, we propose advanced neural network modeling and efficient EM optimization approaches for microwave passive components. A new deep neural network technique is proposed for microwave modeling to address the challenges due to high-dimensional inputs. To accelerate the EM optimization process for microwave component design, we propose an advanced cognition-driven optimization incorporating transfer function-based feature surrogate. Note that the transfer function-based feature surrogate is defined as a surrogate model representing the relationship between geometrical variables and feature parameters. The surrogate model inputs are geometrical variables, and outputs are feature parameters extracted based on transfer function response. As a further advancement, an efficient EM topology optimization technique incorporating advanced matrix Padé via Lanczos and genetic algorithm is proposed to speed up the EM topology optimization process of microwave components.

1.2 List of Contributions

The main objective of this thesis is to investigate and develop advanced modeling and optimization approaches for microwave passive components. This thesis first proposes a new deep neural network technique for high-dimensional microwave modeling. The developed deep neural network technique is a useful surrogate modeling technique for microwave components. Based on the investigation of the surrogate modeling technique, this thesis further investigates the efficient surrogate-based EM optimization method and proposes an advanced cognition-driven EM optimization incorporating transfer function-based feature surrogate for microwave filters. As a further advance, in order to address the EM topology optimization that the surrogate-based optimization method is not applicable, this thesis proposes an efficient EM topology optimization method incorporating advanced Matrix Padé via Lanczos and genetic algorithm for microwave components. The major contributions of this thesis are summarized as follows:

- A novel hybrid deep neural network modeling technique is proposed to solve complex high-dimensional modeling problems of microwave components [98]. In the proposed technique, we formulate a hybrid structure utilizing both sigmoid functions and ReLUs as activation functions for hidden neurons. A smooth ReLU is proposed for the hybrid deep neural network model. Compared to the conventional ReLU, the proposed smooth ReLU can improve the smoothness of the model outputs and provide continuity of derivatives. The proposed deep neural network can be trained to learn the training data in

high-dimensional space. An advanced three-stage deep learning algorithm is proposed to train the new deep neural network and to overcome the vanishing gradient problem that would otherwise exist in training the hybrid deep neural networks. The proposed deep neural network technique can solve higher dimensional microwave modeling problems more efficiently as compared to that solved by the shallow neural network. Compared to the pure ReLU deep neural network, the proposed technique can achieve higher accuracy with fewer number of hidden neurons.

- An advanced cognition-driven EM optimization incorporating transfer function-based feature surrogate for EM optimization of microwave filters is proposed [99]. The proposed optimization technique addresses the challenges where the response at the starting point for the design optimization is substantially misaligned with the design specifications and the filter response does not have explicitly identifiable feature information. Multiple feature parameters, including passband feature frequencies, stopband feature frequencies, feature bandwidths, feature ripples, and boundary feature heights, are extracted from the transfer function response in the proposed technique. New objective functions are formulated directly in the feature space for the cognition-driven optimization. The proposed cognition-driven optimization incorporating transfer function-based feature surrogate can achieve faster convergence than the existing feature-assisted EM optimization methods.
- An efficient EM topology optimization technique is proposed for the design

of microwave components [100]. A new method is proposed to integrate Matrix Padé via Lanczos (MPVL) and Householder formula. Using the proposed method, the effort of solving the large FEM matrix equation at many frequencies is reduced to the effort of solving only a small matrix problem at a single frequency point, thereby speeding up the topology optimization process. We further propose a new method to reduce the small matrix problem into an even smaller one by exploiting the inheritance pattern of GA. New formulations are derived to exploit the common factors between new children EM structures and the parent EM structures in each GA generation so that computational expense can be further reduced. The proposed technique incorporating the advanced MPVL and the inheritance pattern of GA can greatly accelerate the EM topology optimization process.

1.3 Thesis Organization

The organization of the rest of the thesis is described as follows:

Chapter 2 provides a literature review of existing microwave modeling and optimization methods that are relevant to this thesis. A review of artificial neural networks for microwave modeling and design is presented. The basic concepts of artificial neural network techniques are described. A review of the popular deep neural networks is also conducted. Several existing surrogate-based EM optimization methods are discussed. Further, a review of the EM topology optimization is provided, and existing numerical methods to accelerate the EM topology optimization process are discussed.

Chapter 3 proposes a deep neural network technique for high-dimensional microwave modeling. The proposed technique addresses the challenges in the high-dimensional microwave modeling, i.e., modeling of smooth and continuous input-output relationship with high-dimensional inputs. Two application examples are utilized to demonstrate this proposed technique.

Based on the investigation of the surrogate modeling technique in Chapter 3, Chapter 4 studies the surrogate-based EM optimization method and proposes an advanced cognition-driven EM optimization incorporating transfer function-based feature surrogate for microwave filters. By utilizing the advantages of the transfer function-based feature surrogate, the proposed optimization can achieve faster convergence than the existing feature-assisted EM optimization methods. This proposed technique is illustrated by two microwave filter examples.

To further address the EM topology optimization that the optimization method proposed in Chapter 4 is not applicable, Chapter 5 presents an efficient EM topology optimization technique incorporating advanced Matrix Padé via Lanczos and genetic algorithm for microwave design. The proposed technique can greatly accelerate the EM topology optimization process. The computational advantage of the proposed technique is illustrated using three application examples.

Finally, Chapter 6 provides conclusions of the thesis and discussions on future research directions.

Chapter 2

Literature Review

2.1 Review of Artificial Neural Networks for Microwave Modeling and Design

The increasing demands for high functionality and short design cycle in the microwave industry have created a need for accurate and efficient CAD techniques for microwave modeling and design. Artificial neural networks (ANNs) have recently gained recognition as a useful tool for microwave modeling and design [8]-[12]. ANNs can be trained from the simulated or measured microwave data to learn the behavior of microwave components/circuits. The trained ANN models can provide fast and accurate answers to the tasks they have learned and subsequently can be used for high-level simulation and design [101]-[105]. Neural networks have been attractive alternatives to traditional methods such as analytical methods, or empirical modeling solutions, or detailed EM/physics models [8].

ANN techniques have been reported in various microwave applications such as filters [13]-[16], power amplifiers [17]-[20], antennas [106], microstrip components and

circuits [107], [108], very-large-scale-integration (VLSI) interconnects [109], [110], field effect transistor (FET) devices [111], [112], coplanar waveguide (CPW) components [27], embedded passives [113], high electron mobility transistor (HEMT) devices [114], [115], spiral inductors [116], microwave reflectometer [28], wireless power transfer system[117], etc. These applications in the area of microwave modeling and design are achieved mostly using shallow neural networks, i.e., ANNs with one or two hidden layers.

Important works have been done by many researchers to investigate ANN-based microwave modeling and design techniques based on different neural network structures, including multilayer perceptron (MLP) neural networks [41], wavelet neural networks [102], recurrent neural networks [118], radial basis function (RBF) neural networks [17], time-delay neural networks [119], [120], dynamic neural networks [21], [121], state-space dynamic neural networks [122], knowledge-based neural networks [123], and the recently reported multivalued neural networks [13]. In the following subsection, we review several commonly used neural network structures in the area of microwave modeling and design.

2.1.1 Neural Network Structures

A neural network is an information processing system that is composed of two types of basic components, namely, neurons and links. The neurons are the processing elements, and the links are the interconnections between neurons [8]. In a neural network, each neuron (except for the neurons at the input layer) receives stimuli (inputs) from the neighboring neurons connected to it, processes the information,

and produces an output. The way in which the information is processed by a neuron is determined by the activation function used for this neuron. There are different activation functions including the sigmoid function, the arc-tangent function, the hyperbolic tangent function, etc [41]. Neurons can process information in different ways, and the connections from the neurons to one another can be different. Different information processing elements and different connection manner between the neurons can construct different neural network structures.

Among various neural network structures, a basic type of neural network is the feedforward neural network. An important class of feedforward neural networks is the MLP neural network which is the most popular type of neural network used in microwave modeling and design [41]. The neurons in the MLP structure are usually grouped into layers. The first layer is known as the input layer, and the last layer is known as the output layer. The other layers between the input and output layers are known as hidden layers. Typically, there are one input layer, one output layer, and one or more hidden layers in an MLP structure, as demonstrated in Fig. 2.1 [41]. MLP neural networks are widely used in microwave modeling and design. Practical applications are commonly achieved using MLP with one or two hidden layers [15], [16], [19], [110], [113]. Recently, there has been growing interest in the neural network community in neural networks with many hidden layers, which can represent functions with higher complexity [31].

In addition to MLP neural networks, the RBF neural network and the wavelet neural network are also belong to feedforward neural networks. They are both neural networks with one hidden layer [41]. The activation functions used in the

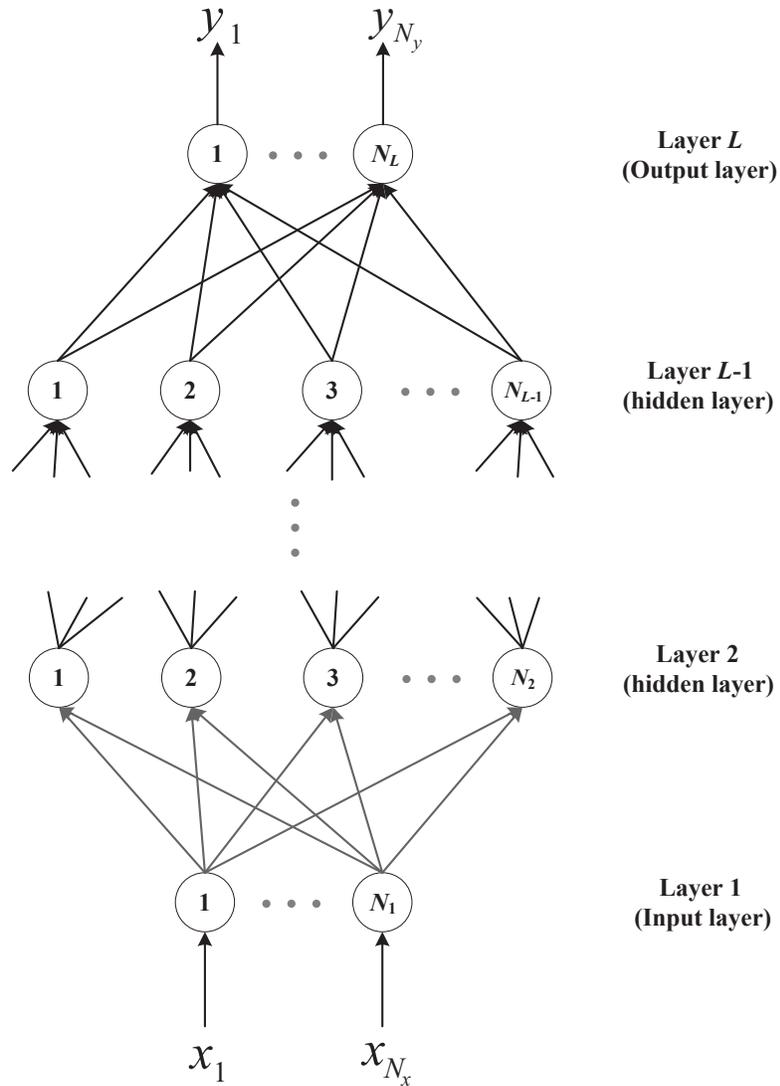


Figure 2.1: Multilayer perceptrons (MLP) neural network structure containing one input layer, one output layer, and one or more hidden layers. [41].

RBF neural networks are radial basis activation functions [17], while the activation functions used in the wavelet networks are wavelet functions [124]. Both RBF neural

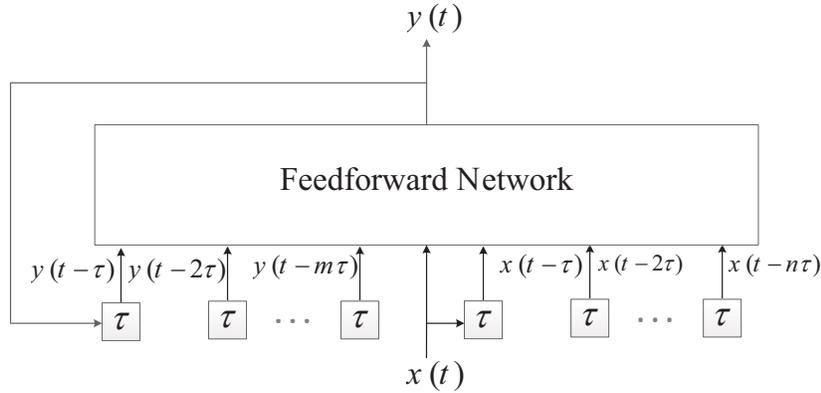


Figure 2.2: A recurrent neural network with feedback of delayed neural network output [41].

networks and wavelet neural networks are useful in the area of microwave modeling and design. Applications have been reported for modeling and design of microwave devices and systems [125]-[127].

Besides the basic feedforward neural networks, there are other types of neural networks. A specific type of neural network capable of learning and then representing dynamic system behavior is the recurrent neural network (RNN) [118]. Fig. 2.2 shows a RNN structure in which delayed neural network outputs are fed back to the inputs [41]. As the time parameter is introduced, the model inputs and outputs of RNN are both functions of time. Therefore, the RNN structure suits time-domain modeling tasks to represent time-domain behaviors. Another specific type of neural network is the dynamic neural network which is presented to describe continuous time domain behavior modeling [121]. Dynamic neural network is suitable for large-signal modeling of nonlinear microwave circuits and systems. Another popular type of microwave-oriented ANN structure is the knowledge-based

neural network (KBNN). KBNN combines neural networks with existing engineering knowledge (e.g, equivalent-circuit models and empirical equations) [25], [123].

Various neural network structures have been developed for different microwave applications. Before using a neural network model in microwave design, we have to train the neural network model with the simulated or measured circuit data. In the following subsection, we describe the general neural network training process.

2.1.2 Neural Network Training

A neural network model representing the input-output relationship of a microwave component is developed through a process called neural network training. Let \mathbf{x} be a vector of size $N_x \times 1$ containing the external inputs to the neural network model (e.g., design parameters of a given microwave component). Let \mathbf{y} be a vector of size $N_y \times 1$ containing the outputs from the neural network model (e.g., EM responses of the given microwave component). The input-output relationship of the neural network model is given by [41]

$$\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{w}). \quad (2.1)$$

where \mathbf{w} is a vector containing all the weight parameters representing various interconnections in the neural network model.

To represent the input-output relationship of a microwave component, the neural network has to be trained through a set of training data generated from original EM/physics simulations or measurements. Let sample pairs $(\mathbf{x}_n, \mathbf{d}_n)$, $n = 1, 2, \dots, N_{tr}$, represent the training data, where \mathbf{d}_n is simulated/measured output

data (i.e., the desired outputs of the neural network model) for the input \mathbf{x}_n , and N_{tr} is the number of training samples. The objective of the training process is to determine the weight vector \mathbf{w} such that the difference between neural network outputs and the desired outputs is minimized. The difference, also known as the training error, is formulated as [41]

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N_{tr}} \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{d}_n\|^2. \quad (2.2)$$

Different training algorithms have been presented to train the neural networks. In microwave applications, the commonly used training algorithms include gradient-based training algorithms [8] such as backpropagation (BP), conjugate gradient, quasi-Newton, and so on. Another type of training algorithm is the global optimization method [8] such as simulated annealing and genetic algorithms. They can be used to search for globally optimal solutions of weight parameters at the cost of longer training time.

To evaluate the quality of a trained neural network model, an independent set of data, i.e., the test data, is used to test the model. An error function, i.e., the test error, is defined similarly to the training error to describe the difference between the model outputs and the desired outputs for the test data. If both training and testing errors are smaller than the user-defined error threshold, the neural network model training process is finished. The well-trained neural network model can be used in high-level microwave simulation and design to provide fast model evaluation replacing the computationally intensive EM/physics simulators.

2.2 Review of Deep Neural Networks

As reviewed in the previous section, applications of neural networks in the area of microwave modeling and design are achieved mostly using shallow neural networks, i.e., ANNs with one or two hidden layers. However, in microwave modeling and design field, there are more challenging problems that are beyond the capability of shallow neural networks. In this case, the neural network with many hidden layers become a suitable choice. It has been proved that the neural network with many hidden layers can perform significantly better than the shallow neural network when both neural networks have the same number of weight parameters [29], [30].

In recent years, there has been growing interest in the neural network community in neural networks with many hidden layers, known as deep neural networks [31]-[35]. Typically, the numbers of network layers in a deep neural network range from five to more than a thousand [128]. Deep neural networks are recognized to be very powerful at modeling intricate relationships in large data sets [31]. It can represent functions with higher complexity than shallow neural networks. Outstanding applications and results of deep neural networks have been produced in a variety of challenging fields, such as image classification [36], speech recognition [37], language processing [38], machine translation [39], sentiment analysis [40], etc. Recently, research and development of deep neural networks for microwave applications have been reported [129]-[134].

As the rapid development of computing technology, researchers have developed various deep neural network structures, including feedforward deep neural networks

[128], convolutional neural networks (CNNs) [135], and recurrent neural networks (RNNs) [136]. A basic type of deep neural network structure is the feedforward deep neural network such as MLP with many hidden layers. The structure of MLP with many hidden layers is the same as the MLP structure shown in Fig. 2.1, where the number of hidden layers should be three or more. The feedforward deep neural network can be used to learn the relationship between a fixed-size input and a fixed-size output [31]. CNN is a particular type of feedforward deep neural network designed to process data in the form of multiple arrays, e.g., images or videos [31]. RNN is a specific type of deep neural network that contains feedback loops involving delay units. It is suitable for problems involving sequential inputs, e.g., language or speech [31]. A variation of RNN as shown in Fig. 2.2 has been used for representing dynamic behavior of microwave components [118].

Training of deep neural network is also known as deep learning. It is an intensive topic in the neural network community. The deep learning process is not as easy as the learning process of shallow ANNs. The quality of training of deep neural networks can be affected by the selection of types of activation functions. In shallow neural networks, the most commonly used activation functions are the smooth nonlinear functions, such as the sigmoid function [41]. Shallow neural networks with sigmoid activation functions usually can be trained successfully by gradient-based BP algorithm from a random starting point. However, training deep neural networks with sigmoid functions using BP algorithm is problematic because of the vanishing gradient problem [42], [43]. The vanishing gradient problem is the phenomenon that the gradient of the error function with respect to neural network

internal weight parameters approaches zero as the gradient propagates from the output layer backward through more and more hidden layers [44].

At present, one of the most popular activation functions for deep neural networks is the rectified linear unit (ReLU). This is because ReLU allows deep neural networks to be trained without the vanishing gradient problem [31], [45]. Deep neural networks with ReLUs have been demonstrated to be effective for speech recognition [46], image classification [33], acoustic models [47], and so on. For neural networks with ReLUs, only a subset of neurons are active for a given input. In order to reach similar accuracy as the neural network with sigmoid activation functions, a neural network with ReLUs therefore would need much more hidden neurons [45].

In this section and the previous section, we reviewed the powerful surrogate modeling techniques, i.e., the neural network techniques. Based on the investigation of the surrogate modeling techniques, a further research topic is the surrogate-based EM optimization methods.

2.3 Review of Surrogate-Based EM Optimization Methods

EM-based optimization plays an important role in the design of microwave components. The EM optimization problem can be formulated as follows. Let ϕ_F represent a vector of geometrical design variables. Let ω represent the angular frequency. Let $\mathbf{S}_F(\phi_F, \omega)$ represent the EM response of the fine model (e.g., EM simulation), which is a function of design variables ϕ_F and frequency ω . The EM

optimization problem to be solved is defined as [48]

$$\boldsymbol{\phi}_F^* = \arg \min_{\boldsymbol{\phi}_F} U(\mathbf{S}_F(\boldsymbol{\phi}_F, \omega)) \quad (2.3)$$

where U represents the given objective function, which can be a minimax or a generalized l_p function; $\boldsymbol{\phi}_F^*$ represents the optimal design solution satisfying the design specifications.

EM-based optimizations are computationally expensive because repetitive EM simulations with different geometrical parameters as design variables are required in the optimization process. The underlying EM simulations can be very time consuming, resulting in the high computational cost of the whole EM optimization process. Surrogate-based optimization methods have been introduced to speed up the EM-based optimization [65]-[70], [73], [74], [76]-[81]. The basic concept of surrogate-based EM optimization is that the direct optimization of the computationally expensive model (e.g., EM simulation) is replaced by iterations involving the generation, optimization, and updating of a computationally efficient surrogate model [137]. Space mapping (SM) is one of the popularly used techniques for surrogate-based EM optimization [52], [53]. The idea of SM-based optimization is to obtain the optimal design solution of the fine model through iterative optimization and updating of the surrogate model, i.e., the coarse model composed with suitable mappings. Space mapping methods can integrate the accuracy of the fine models and the computational efficiency of the coarse models, thereby speeding up the EM optimization process [65]-[70]. Sensitivity analysis has also been involved in the space mapping optimization to accelerate the optimization process [138]. How-

ever, space mapping techniques usually require the availability of coarse models. Recently, research efforts have focused on surrogate-based optimization methods that can be used even if coarse models (equivalent circuits or empirical models) are unavailable [73], [74], [76]-[81]. In the following subsections, we discuss several popular surrogate-based EM optimization methods addressing the situation where coarse models are unavailable.

2.3.1 EM Optimization Using Neuro-Transfer Function Surrogates

Recently, EM optimization method that utilizes surrogates combining neural networks and transfer functions (neuro-TF) has been developed for microwave optimization and design [73], [74]. This method can be used to perform efficient EM optimization even if the coarse model (e.g., equivalent circuit model) is not available. In the neuro-TF surrogate model, the EM responses of microwave passive components versus frequency are expressed by the widely used transfer functions. The relationships between the coefficients of transfer functions and the geometrical variables of the microwave passive components are represented by neural networks [26], [72]. Fig. 2.3 illustrates the typical structure of the neuro-TF surrogate model. The neuro-TF model allows fast and accurate evaluation of EM behavior with respect to changes in geometrical parameters. In EM optimization using neuro-TF surrogates, the optimal design solution is obtained through iterative optimization and updating of the neuro-TF model.

Researches investigating different types of transfer functions, such as the rational

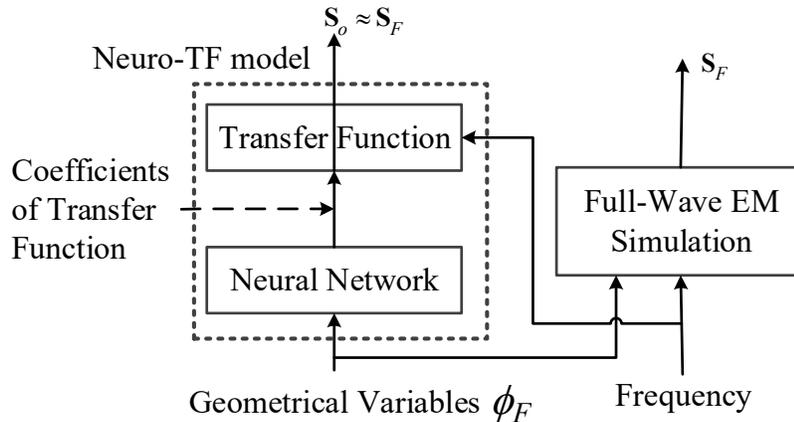


Figure 2.3: Structure of the neuro-TF surrogate model [72].

transfer functions [72], [73], the pole-residue-based transfer functions [2], and the pole-zero-based transfer functions [139], have been reported in recent years. Sensitivity analysis has also been involved to increase the accuracy of the neuro-TF surrogate model [26].

2.3.2 EM Optimization Using Feature-Based Surrogates

Research works have investigated feature parameters of EM responses for microwave modeling and optimization [76], [77]. In [77], a feature-based surrogate is developed based on suitably selected ‘visible’ response features (e.g., 3dB bandwidth, frequency location of -20dB response level, local ‘peaks’, and/or ‘valleys’) rather than the responses (e.g., S-parameters versus frequency) themselves. The feature-based surrogate modeling process includes three steps. Step 1: extracting response features; Step 2: constructing response surface approximation model of the features corresponding to design variables; Step 3: evaluating the surrogate model response by interpolating predicted features. The developed feature-based surrogate models

can be applied to EM optimization to reduce the computational cost. Since the feature-based surrogate model is developed based on features extracted from EM responses, it does not rely on the availability of coarse models.

2.3.3 Cognition-Driven EM Optimization in Feature Space

Cognition-driven optimization is a specific type of surrogate-based optimization method which does not process explicit surrogate models. In contrast, cognition-driven optimization proceeds with a meaningful surrogate model which is implied by the engineer's experience and intuition [78]. The motivation of this method is the cognitive design process of the experienced filter designer. In cognition-driven optimization presented in [78], two sets of intermediate feature space parameters are used to develop two kinds of space mapping. Fig. 2.4 [78] illustrates the two sets of feature space parameters, i.e., feature frequency parameters and ripple height parameters, in the response of a four-pole filter. A space mapping is developed to map the design variables to the feature frequency parameters, and another space mapping is developed to further map the feature frequency parameters to the ripple height parameters. By directly formulating the optimization in the feature space, the cognition-driven optimization can improve the optimization efficiency and the ability to avoid being trapped in local optimal [78].

As an extension of [78], an enhanced cognition-driven optimization method has been reported in [79]. By correcting the number of feature frequencies, the enhanced cognition-driven optimization can work well even if the filter response at the starting point does not have a correct number of feature frequency parameters [79]. The

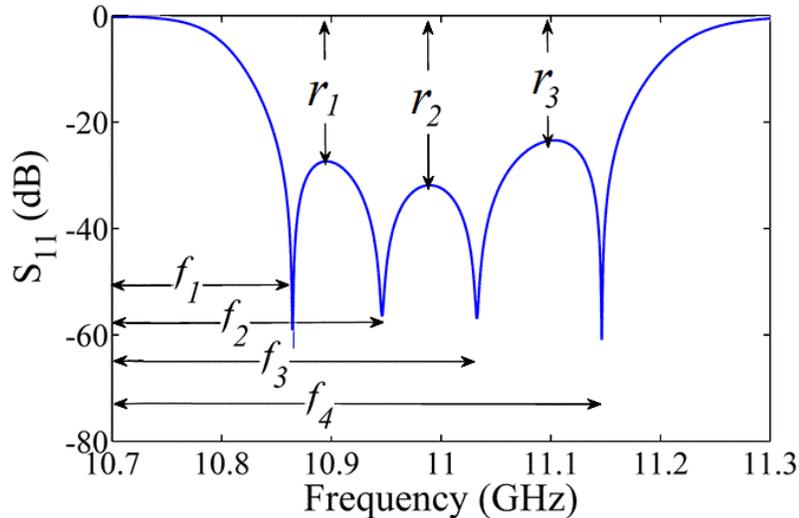


Figure 2.4: Feature frequency parameters f and ripple height parameters r illustrated using the response of a four-pole filter [78].

cognition-driven optimization methods are suitable for design of filters with equal-ripple responses.

2.3.4 Feature-Assisted Neuro-TF Surrogate-Based EM Optimization

In the feature-based optimization methods reviewed in the above two subsections, the feature parameters are extracted directly from the EM response (e.g., S-parameters). In some practical cases, the EM response does not have explicitly identifiable feature information. More recently, researchers have investigated the use of transfer function-based features to assist the surrogate-based EM optimization [80], [81]. Transfer function-based features are defined as the feature parameters that are identified from the transfer function response. They can address the challenge when the feature parameters cannot be effectively extracted from the EM responses.

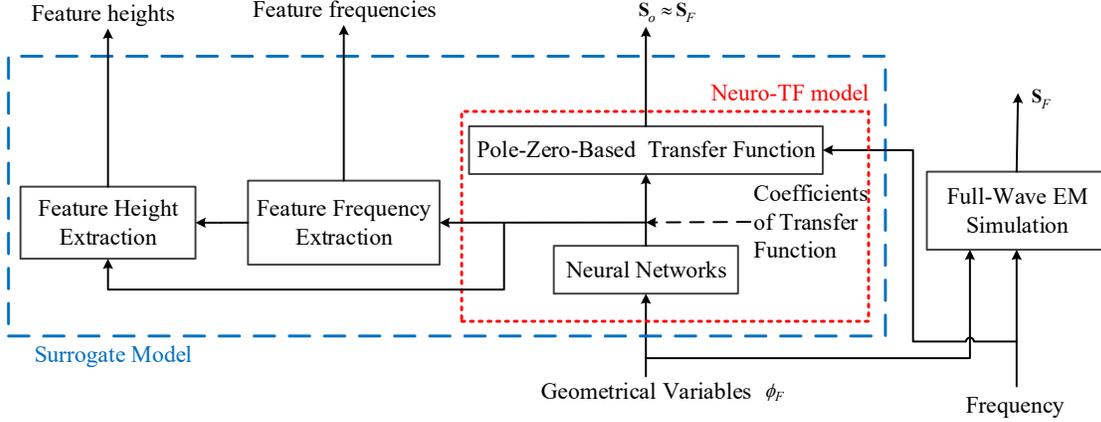


Figure 2.5: Structure of the multifeature-assisted neuro-TF surrogate model [81].

In [80], feature zeros of the neuro-TF are used as feature parameters to assist the neuro-TF surrogate-based optimization. As a further advance over the work of [80], the research work in [81] introduced a multifeature-assisted neuro-TF surrogate-based EM optimization method. Fig. 2.5 shows the structure of the multifeature-assisted neuro-TF surrogate model [81]. In [81], two sets of feature parameters, namely, the feature frequencies and the feature heights, are identified from the pole-zero-based neuro-TF response. With the assistance of transfer function-based features, the surrogate-based EM optimization can obtain the optimal solution in shorter CPU time than that without the assistance of features. It also shows a better ability to avoid the local optimal.

2.4 Review of EM Topology Optimization

The surrogate-based EM optimization methods reviewed in the previous section are suitable only for the EM geometry optimization, where the performance of the mi-

microwave component is optimized by adjusting the values of a given set of geometrical parameters, such as length and width of microwave structures [9], [82]. In EM geometry optimization, the geometrical change of the EM structure is defined completely by the changes in the values of a given set of geometrical parameters. Therefore, surrogate-based optimization methods can be used to speed up EM geometry optimization. In addition to EM geometry optimization, there are another two types of EM optimization methods in the literature, i.e., EM shape optimization and EM topology optimization. In EM shape optimization, the EM structure is optimized by modifying the geometrical boundary of objects in the design space [140], [141]. EM shape optimization allows optimal EM structures with unrestricted shapes in the design space. In EM shape optimization, the topology of the EM structure, e.g., the number of holes in the design space, is fixed in the optimization process. The third type of EM optimization is EM topology optimization. It is a numerical optimization method to find the optimal shape and topology of the EM structure by repetitively adding or eliminating geometrical/material elements within the design space [83], [84]. EM topology optimization can search for the optimal EM structure without constraints of shape and topology in the design space.

EM topology optimization does not require the definition of an initial layout of the structure within the design space. Instead of changing the dimensions of a predefined shape, topology optimization methods search for the optimal structure by changing the distribution of materials within a given design space. In this way, these methods can introduce/remove blocks and holes of material in the design space, change the shape and topology of the EM structure, and explore new possibilities

for optimal design solutions [142]. EM topology optimization has been applied in the design of microwave components to achieve higher performance or more compact configurations [84]-[86].

Various topology optimization techniques, such as homogenization methods [143], density-based methods [87], [88], ON/OFF (discrete/binary) methods [84], [93], and level-set methods [144], have been investigated by many researchers for EM optimization and design. In EM topology optimization, the entire design space is usually divided into many small cells and the material composition of each cell is regarded as a design variable. Topology optimization has been formulated into either continuous optimization or binary optimization according to the variation mode of the material properties in each small cell in the design space. In [87]-[90], the material properties are allowed to vary continuously in each small cell within the design space in the topology optimization process. The topology optimizations in [91]-[93] deal with the presence or absence of the material in each small cell within the design space, i.e., the material properties are represented by binary variables.

Since the EM topology optimization converges towards an optimal design solution based on a cost function related to the EM response (such as S-parameters) of the component under consideration, it requires efficient numerical methods for EM simulations and powerful optimization algorithms. The EM simulations in EM topology optimization often rely on accurate numerical methods that solve Maxwell's equations, such as the finite element method [84], [143] and the method of moments [91]. There are generally two types of optimization methods for the topology optimization, i.e., the gradient-based optimization method [83], [86] and the

gradient-free optimization method [94]-[97]. The main advantage of the gradient-based optimization method is to converge rapidly toward the nearest local optimal solution. However, it may suffer from getting stuck in the local optima. The gradient-free optimization, such as the genetic algorithm (GA), can provide the maximum probability to find the global optimal solution at the cost of higher computational expense.

EM topology optimization problem cannot be solved by the widely used surrogate-based optimization because different shapes and topologies cannot be related by changing the values of a given set of geometrical parameters. Therefore, repetitive EM simulations for different shapes and topologies of EM structures are needed in EM topology optimization process. Typical EM simulation methods such as finite element method (FEM) require solving matrix equations with very large matrices. In the EM topology optimization, large matrix equations (e.g. FEM matrix equations) need to be solved repetitively for different EM structures over a frequency range, which is time-consuming. In the following subsection, we discuss several existing numerical methods to reduce the computational cost of solving different EM structures during optimization, thereby speeding up the EM topology optimization process.

2.4.1 EM Fast Frequency Sweep to Speed up EM Topology Optimization

EM fast frequency sweep using model order reduction (MORe) techniques is an efficient method to improve the computational efficiency of the EM simulations during

the topology optimization. MORE techniques [145], [146] can be used to speed up the EM simulation by avoiding repetitively solving large EM matrix equations for different frequencies. Using MORE techniques, large matrix equations of EM problems need to be solved only at a single frequency point. Note that, in addition to the computational cost at a single frequency point, there is some computational overhead to prepare the matrices needed due to the application of MORE techniques. However, this overhead is negligible if the EM solutions need to be solved at many frequency points. In practical applications, most of the situations involve EM simulation at many frequency points. Therefore, the effort of solving large matrix equations at many frequencies is approximately reduced to the effort of solving the large matrix equation at a single frequency point, due to the application of MORE techniques.

Different MORE techniques, such as the asymptotic waveform evaluation (AWE) [147], Galerkin AWE [148], and the matrix Padé via Lanczos (MPVL) [149], have been explored for EM fast frequency sweep. Here we recall the MPVL method on FEM as present in [149] to demonstrate the idea of using EM fast frequency sweep to speed up the EM simulations during topology optimization. The FEM equations to solve the EM problem can be formulated in a generalized form as [150]

$$(\mathbf{K}_0 + \gamma\mathbf{K}_1 + \gamma^2\mathbf{K}_2)\mathbf{e} = \gamma\mathbf{R} \quad (2.4)$$

where γ is the complex propagation constant, which is a function of the frequency; \mathbf{K}_0 , \mathbf{K}_1 , and \mathbf{K}_2 are FEM matrices, which are dependent on the EM structure but independent on γ ; \mathbf{R} is a vector describing the EM excitation at the input port; \mathbf{e}

represents the solution vector containing the unknown values used in approximating the electrical field. In EM topology optimization, FEM equations need to be solved over a frequency range.

In order to apply MPVL on the FEM matrix equation, the FEM matrix equation (2.4) is cast into an equivalent equation as [149]

$$(\mathbf{A}_1 + \gamma \mathbf{A}_2) \tilde{\mathbf{e}} = \gamma \tilde{\mathbf{R}} \quad (2.5)$$

where

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} \mathbf{0}_N & \mathbf{I}_N \\ \mathbf{K}_0 & \mathbf{K}_1 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} -\mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{K}_2 \end{bmatrix}, \\ \tilde{\mathbf{e}} &= \begin{bmatrix} \mathbf{e} \\ \gamma \mathbf{e} \end{bmatrix}, \tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{R} \end{bmatrix}. \end{aligned} \quad (2.6)$$

where N is the number of elements in the unknown solution vector \mathbf{e} , \mathbf{I}_N and $\mathbf{0}_N$ represent the $N \times N$ identity matrix and the $N \times N$ zero matrix, respectively. The following definitions are made at a specific frequency which is defined as the frequency expansion point [149]:

$$\mathbf{A} = -(\mathbf{A}_1 + \gamma_0 \mathbf{A}_2)^{-1} \mathbf{A}_2, \quad \mathbf{g} = (\mathbf{A}_1 + \gamma_0 \mathbf{A}_2)^{-1} \tilde{\mathbf{R}}, \quad (2.7)$$

where γ_0 represents the complex propagation constant at the frequency expansion point for MPVL.

For an EM structure, given \mathbf{A}_1 , \mathbf{A}_2 , $\tilde{\mathbf{R}}$, \mathbf{A} , and \mathbf{g} , the results returned from the MPVL algorithm are the $\rho \times \rho$ tridiagonal matrix \mathbf{T} , the $\rho \times 1$ vector \mathbf{a} , and

the $2N \times \rho$ matrix \mathbf{V} , where ρ is the order of the reduced-order model. Then the solution to the FEM-MPVL equation (2.5) can be obtained as [149]

$$\tilde{\mathbf{e}} = \gamma \mathbf{V} (\mathbf{I}_\rho - (\gamma - \gamma_0) \mathbf{T})^{-1} \mathbf{a}. \quad (2.8)$$

Applying MPVL on FEM, the large matrix equation needs to be solved only at a single frequency point. EM solutions at other frequencies can be obtained by the information calculated at the single frequency. In this way, the EM simulations of different EM structures can be accelerated, thereby accelerating the EM topology optimization process.

2.4.2 Matrix Reduction Method for EM Topology Optimization

The matrix reduction method, such as the matrix partitioning method, has been exploited to improve the computational efficiency for solving large matrix equations in EM topology optimization [91]. The matrix partitioning method exploits the characteristic that elements outside the design space are held constant during EM topology optimization. Based on this characteristic, the large matrix equation of the EM problem is partitioned. The partitioned matrix equation associated with the constant elements is pre-solved and the corresponding information is stored. With the matrix partitioning method, the EM solution of a new EM structure can be obtained by solving a relatively small matrix equation which is associated with the elements in the design space. This small matrix equation is solved at each and every frequency to obtain the EM solution over a frequency range. In

this way, the computational cost of solving the large matrix equations at different frequencies is reduced to the cost of solving relatively small matrix equations at different frequencies, thereby reducing the computational cost of the EM topology optimization process. In EM topology optimization using the matrix partitioning method, the partitioned matrix equation associated with the constant elements needs to be pre-solved and stored for each frequency, which may cause a huge memory cost. Other matrix reduction methods, such as the Householder formula [151] and the diakoptics approach [152], can be potentially used for EM topology optimization because they have similar concept as the matrix partitioning method.

2.5 Summary

In this chapter, a literature review of existing modeling and optimization methods for microwave passive components, which are relevant to this thesis, has been provided. A review of ANNs for microwave modeling and design has been presented. The basic concepts of ANN techniques, including ANN structures and the training, have been discussed. A review of popular deep neural networks has also been conducted. Several existing surrogate-based EM optimization methods, including EM optimization using neuro-TF surrogates, EM optimization using feature-based surrogates, cognition-driven EM optimization in feature space, and feature-assisted neuro-TF surrogate-based EM optimization, have been reviewed. Further, a review of the EM topology optimization has also been provided. Existing numerical methods to accelerate the EM topology optimization process, such as EM fast frequency sweep method and the matrix reduction method, have also been discussed.

Chapter 3

Deep Neural Network Technique for High-Dimensional Microwave Modeling

3.1 Introduction

This chapter introduces the deep neural network technique into the microwave modeling field to solve microwave modeling problems which are much more challenging than those solved by previous shallow neural networks. In the microwave modeling field, there are situations where the model inputs are in high dimension. Microwave modeling in high dimension using neural networks requires training a neural network to learn a nonlinear function with high-dimensional inputs. When the input dimension of the nonlinear function becomes higher, the relationship between the inputs and the outputs of the microwave model will become more complicated and the modeling problem will become harder. In this case, we consider the use of the deep neural network to overcome the challenges due to high-dimensional inputs. The most commonly used activation function in the existing deep neural network

is the rectified linear unit (ReLU), which is a piecewise hard switch function. However, such a ReLU is not suitable for microwave modeling where the input-output relationships are smooth and continuous. On the other hand, the most commonly used activation functions in the conventional (shallow) neural networks are smooth switch functions, such as the sigmoid function. However, training deep neural networks with sigmoid functions using the backpropagation algorithm is problematic because of the vanishing gradient problem.

In this chapter, we propose a new deep neural network modeling technique [98] specially to address the challenges in high dimensional microwave modeling, i.e., modeling of smooth and continuous input-output relationship with high dimensional inputs. We propose to combine sigmoid functions and ReLUs to form a hybrid deep neural network. Since the sigmoid function has higher nonlinearity than ReLU function, the proposed hybrid deep neural network can reduce the number of hidden neurons compared to the deep neural network using only ReLUs. The combination of two types of activation functions makes it possible to avoid the vanishing gradient problem when training the deep neural network. In order to obtain smooth outputs and continuous derivatives, we further propose a novel smooth ReLU function to be used for all the ReLU neurons in the new sigmoid/ReLU hybrid deep neural network. A systematic three-stage deep learning algorithm is proposed for training the new hybrid deep neural network. This algorithm can determine the number of hidden layers with sigmoid functions and those with smooth ReLUs in the training process. It can also overcome the vanishing gradient problem for training the deep neural network. Our proposed technique can solve microwave modeling problems in

higher dimension than the shallow neural network. Compared to the existing deep neural network using only conventional ReLUs, the proposed technique can achieve higher accuracy with fewer number of hidden neurons.

3.2 The Proposed Deep Neural Network Technique for High-Dimensional Microwave Modeling

3.2.1 Activation Functions for the Proposed Deep Neural Network Technique

A deep neural network is an artificial neural network that has many hidden layers between its inputs and outputs. The number of hidden layers in a deep neural network should be three or more [153]. The selection of types of activation functions can affect the quality of training of deep neural networks. In the proposed deep neural network technique for high-dimensional modeling of microwave components, we propose to use both sigmoid functions and ReLUs as activation functions for the hidden neurons in order to overcome the vanishing gradient problem and reduce the number of hidden neurons. The reason that the sigmoid function and the ReLU are used in the proposed deep neural network is as follows. The sigmoid function is the most commonly used activation function in conventional shallow neural networks, and ReLU is one of the most popular activation functions for standard deep neural networks. If only sigmoid functions are used in the deep neural network, there would be the vanishing gradient problem in the training process. If only ReLUs are used in the deep neural network, the deep neural network would need much more

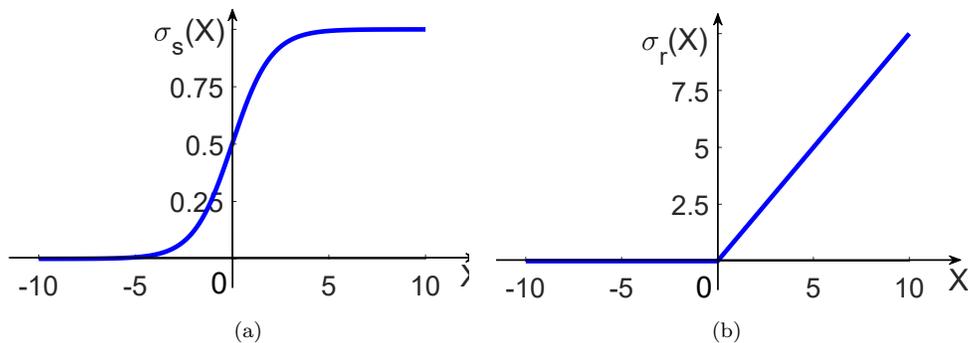


Figure 3.1: Two types of activation functions used in the hybrid deep neural network. (a) Sigmoid function. (b) ReLU function.

hidden neurons compared to that using sigmoid functions. Therefore, we propose to combine sigmoid functions and ReLUs in the hybrid deep neural network to avoid the vanishing gradient problem as well as reduce the number of hidden neurons.

The sigmoid function is shown in Fig. 3.1(a) and expressed as [8]

$$\sigma_s(X) = \frac{1}{(1 + e^{-X})} \quad (3.1)$$

The symbol X represents the total input to the hidden neuron [41]. Shallow neural networks with sigmoid functions usually can be trained successfully with gradient-based learning methods, while deep neural networks with sigmoid functions cannot be trained effectively using gradient-based methods because of the vanishing gradient problem [44], [47]. In such training methods, each of the internal weight parameters in the neural network receives an update proportional to the gradient of the error function with respect to that weight parameter in each iteration of training. The gradient can be computed by the backpropagation (BP) algorithm [154]. According to (3.1), the derivative of the sigmoid function $\sigma'_s(X)$ is always in the

range of

$$0 < \sigma'_s(X) = \sigma_s(X)(1 - \sigma_s(X)) \leq 0.25 \quad (3.2)$$

The gradients of the error function with respect to the weight parameters are computed by the chain rule back-propagated from the output layer downward to the input layer. This computation of the gradients has the effect of multiplying lots of these small numbers ($\sigma'_s(X)$) to compute gradients in deep neural networks [154]. The gradients close to the input layer will decrease exponentially and approach zero gradually with the increase of the number of layers in the deep neural network. The gradients approaching zero would prevent the weights from being updated during training process. Consequently, the deep neural network with sigmoid functions cannot be trained effectively using gradient-based learning methods. This is known as the vanishing gradient problem for training deep neural networks. In fact, when using sigmoid activation functions, the gradient will usually vanish especially at the beginning of learning [44], [155].

To overcome the vanishing gradient problem in the training process of deep neural networks, ReLU has been developed as the activation function instead of the sigmoid function for deep neural networks [31]. The ReLU is shown in Fig. 3.1(b) and expressed as [47]

$$\sigma_r(X) = \max(X, 0) = \begin{cases} X & \text{if } X > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

The gradient of ReLU is

$$\sigma'_r(X) = \begin{cases} 1 & \text{if } X > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

The hidden neuron with ReLU will be active only if the total input X is above zero. The partial derivative of ReLU is always one when it is active. When training the deep neural network with ReLUs, the gradients will be calculated by multiplying lots of ones. Thus vanishing gradients do not exist along the path of active hidden neurons in an arbitrary deep neural network with ReLUs [47]. The gradient information can propagate well along the path of active neurons in the deep neural network with ReLUs.

While the conventional ReLU as formulated in (3.3) is suitable for discrete input-output relationships such as speech recognition and image classification, it is not suitable for microwave modeling directly because the output of the conventional ReLU is not smooth. In the microwave modeling application, the outputs of the model are expected to be smooth and continuous. The first-order derivatives of the deep neural network microwave model need to be continuous if the developed model is used for design optimization. In order to make the deep neural network suitable for microwave modeling, we propose a smooth ReLU for the new deep neural network model.

The conventional ReLU is a two-piece piecewise function as shown in Fig. 3.1(b). The non-smooth part of the conventional ReLU is at the intersection point (i.e., $X = 0$) between the two pieces. We define a small range around zero for X as

$[-\mu, \mu]$. We propose to use a quadratic function to replace the non-smooth part of the conventional ReLU for X in the range of $[-\mu, \mu]$ and keep the function output to be equal to the output of the conventional ReLU for X outside the range. To guarantee that the proposed ReLU function is smooth, the quadratic function should have the same outputs and derivatives as those of the conventional ReLU at the two points where $X = -\mu$ and $X = \mu$. To satisfy these requirements, we derive the quadratic function in the range of $[-\mu, \mu]$ as

$$\sigma_q(X) = \frac{1}{4\mu}X^2 + \frac{1}{2}X + \frac{1}{4}\mu, \quad -\mu \leq X \leq \mu \quad (3.5)$$

The proposed smooth ReLU function can be formulated as

$$\sigma_{pr}(X) = \begin{cases} X & \text{if } X > \mu \\ \frac{1}{4\mu}X^2 + \frac{1}{2}X + \frac{1}{4}\mu & \text{if } -\mu \leq X \leq \mu \\ 0 & \text{if } X < -\mu \end{cases} \quad (3.6)$$

Fig. 3.2 shows the comparison of the proposed smooth ReLU with $\mu = 0.05$ and the conventional ReLU. Based on (3.6), the derivative of the proposed smooth ReLU is

$$\sigma'_{pr}(X) = \begin{cases} 1 & \text{if } X > \mu \\ \frac{1}{2\mu}X + \frac{1}{2} & \text{if } -\mu \leq X \leq \mu \\ 0 & \text{if } X < -\mu \end{cases} \quad (3.7)$$

From (3.6) and (3.7), the proposed smooth ReLU and the conventional ReLU have equal values for $|X| \geq \mu$ and equal derivatives at the two points where $|X| = \mu$. In

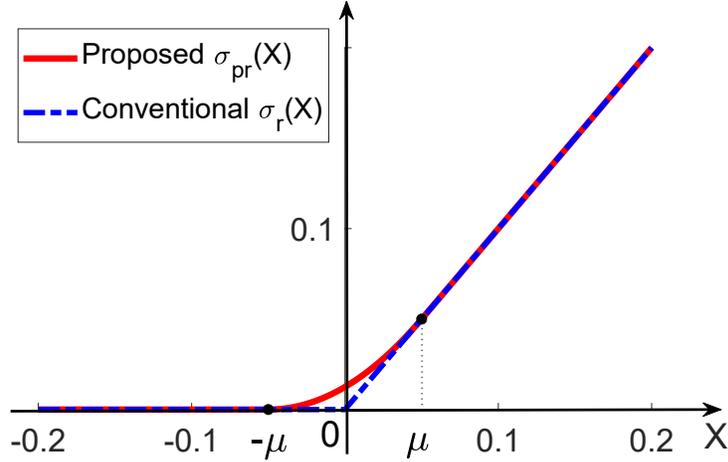


Figure 3.2: The comparison of the proposed smooth ReLU and the conventional non-smooth ReLU. Solid line represents the proposed smooth ReLU with $\mu = 0.05$ and dashdot line represents the conventional non-smooth ReLU.

other words, the proposed smooth ReLU has continuous derivatives for the entire X space.

In the following section, we propose a hybrid deep neural network technique combining sigmoid function and the proposed smooth ReLU. The proposed deep neural network technique can integrate the advantages of these two activation functions and avoid their drawbacks.

3.2.2 Structure of the Proposed Deep Neural Network Technique

We propose a hybrid deep neural network technique to solve complex microwave modeling problems with model inputs in high dimension. The proposed hybrid deep neural network model is shown in Fig. 3.3. It is a fully connected neural network with many hidden layers. We propose to utilize both sigmoid functions and smooth

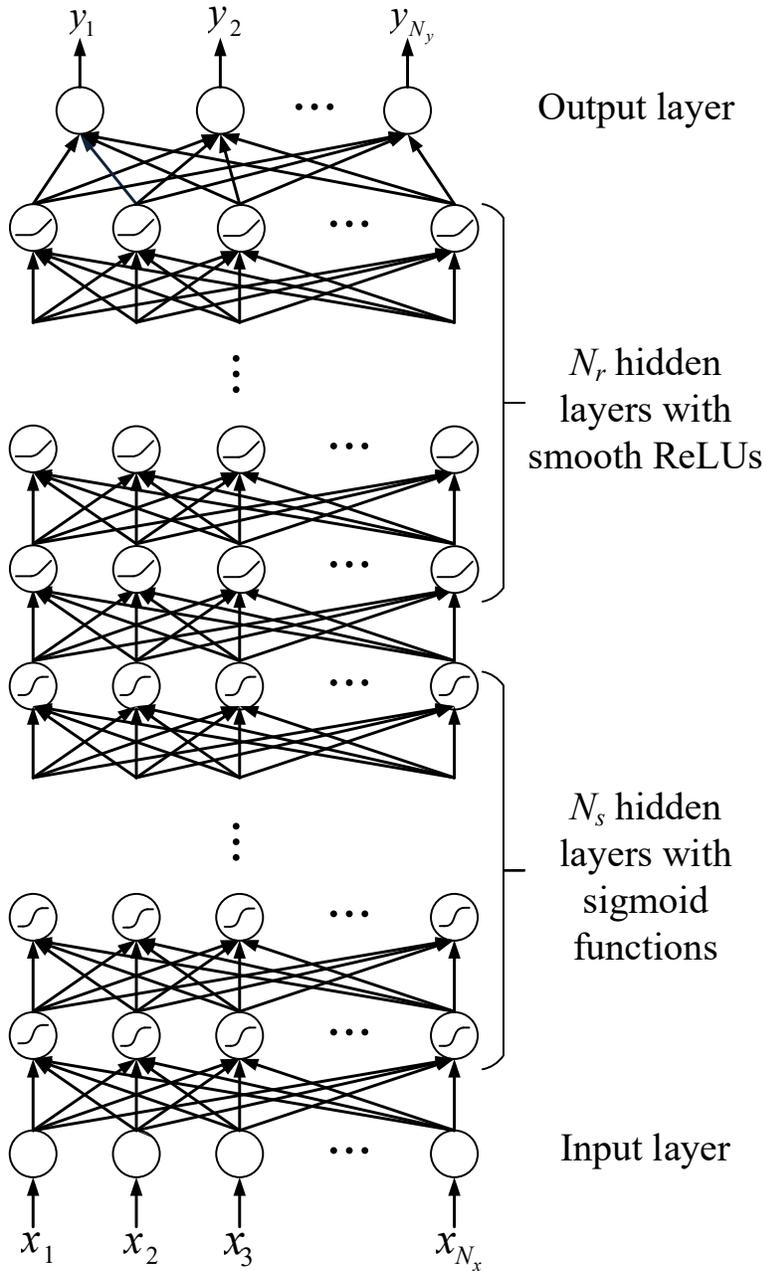


Figure 3.3: Structure of the proposed hybrid deep neural network model for microwave components. The hidden layers in the proposed deep neural network are divided into two parts. N_s hidden layers close to the input layer employ sigmoid functions as neuron activation functions, while the rest N_r hidden layers utilize the proposed smooth ReLUs as neuron activation functions.

ReLU as activation functions for the hybrid deep neural network model. If we use sigmoid functions as activation functions for all hidden neurons, there will be the vanishing gradient problem for training the deep neural network with many hidden layers. If we use smooth ReLUs as activation functions for all hidden neurons, the number of hidden neurons will be much more compared to using sigmoid activation functions. In order to avoid the vanishing gradient problem as well as reduce the number of hidden neurons, we combine the sigmoid functions and smooth ReLUs in the proposed hybrid deep neural network.

Out of all the hidden neurons in the proposed deep neural network, we want to make as many hidden neurons as possible employ sigmoid functions in order to reduce the number of hidden neurons. However, if the number of sigmoid layers is too large, there will be the vanishing gradient problem. Therefore, we need to train the sigmoid hidden layers first to determine the maximum number of sigmoid layers that will not lead to the vanishing gradient problem. After training the sigmoid hidden layers, we begin to add ReLU hidden layers above the trained sigmoid layers. In this way, we can preserve the trained sigmoid hidden layers and make the further training easier. Therefore, the hidden layers close to the input layer employ sigmoid functions as activation functions, while the rest of hidden layers utilize the proposed smooth ReLUs as activation functions.

Let $\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_{N_x}]^T$ represent the external inputs to the proposed deep neural network model, where N_x is the number of the model inputs. Let $\mathbf{y} = [y_1 \ y_2 \ y_3 \ \dots \ y_{N_y}]^T$ represent the outputs of the proposed deep neural network model, where N_y is the number of the model outputs. For example, for the high-

dimensional modeling of microwave filters to identify the coupling parameters from the given S -parameters, \mathbf{x} is composed of S -parameters of the filter at various sampled frequencies, and \mathbf{y} is composed of the coupling parameters of the filter. Suppose the number of hidden layers with sigmoid functions is N_s , and the number of hidden layers with smooth ReLUs is N_r . The total number of layers of the proposed deep neural network will be $\Gamma = N_s + N_r + 2$, including the hidden, input, and output layers. Let the symbol λ be the index of layers in the deep neural network, where $\lambda = 1, 2, \dots, \Gamma$. Let N_λ be defined as the number of hidden neurons in the λ th layer. Let w_{ij}^λ represents the weight between the i th neuron of the λ th layer and the j th neuron of the $(\lambda - 1)$ th layer. There is an additional weight parameter for each neuron (w_{i0}^λ) representing the bias for the i th neuron of the λ th layer. We define X_i^λ as the input to the i th hidden neuron in the λ th layer [8]

$$X_i^\lambda = \sum_{j=0}^{N_{\lambda-1}} w_{ij}^\lambda Z_j^{\lambda-1} \quad (3.8)$$

For the hidden layers with smooth ReLUs as activation functions, only a subset of hidden neurons are active for a given input. We define B^λ as the index set of active hidden neurons in the λ th layer of neurons with smooth ReLUs. Specifically, B^λ is defined as

$$B^\lambda = \left\{ i \mid X_i^\lambda > -\mu \right\}, \lambda = N_s + 2, \dots, \Gamma - 1 \quad (3.9)$$

The outputs of the proposed deep neural network model can be computed by the feedforward process. Let Z_i^λ represent the output of the i th neuron in the λ th layer.

Z_i^λ can be calculated by the feedforward computation as

$$Z_i^\lambda = \begin{cases} x_i & \text{if } \lambda = 1 \\ \sigma_s\left(\sum_{j=0}^{N_{\lambda-1}} w_{ij}^\lambda Z_j^{\lambda-1}\right) & \text{if } \lambda = 2, \dots, N_s + 1 \\ \sigma_{pr}\left(\sum_{j=0}^{N_{\lambda-1}} w_{ij}^\lambda Z_j^{\lambda-1}\right) & \text{if } \lambda = N_s + 2 \\ \sigma_{pr}\left(\sum_{j \in B^{\lambda-1}} w_{ij}^\lambda Z_j^{\lambda-1} + w_{i0}^\lambda\right) & \text{if } \lambda = N_s + 3, \dots, \Gamma - 1 \\ \sum_{j \in B^{\lambda-1}} w_{ij}^\lambda Z_j^{\lambda-1} + w_{i0}^\lambda & \text{if } \lambda = \Gamma \end{cases} \quad (3.10)$$

where $\sigma_s(\cdot)$ and $\sigma_{pr}(\cdot)$ are the sigmoid function and the proposed smooth ReLU, respectively.

The outputs of the deep neural network model can be extracted from the output layer as

$$y_i = Z_i^\Gamma, \quad i = 1, \dots, N_\Gamma \quad (3.11)$$

After the feedforward process of the proposed model is formulated, the next step is to develop an algorithm to adjust values of weight parameters in order to train the deep neural network model with the training data.

3.2.3 Proposed Deep Learning Algorithm for Training the New Deep Neural Network

We define the n th training data as $(\mathbf{x}_n, \mathbf{d}_n)$, $n = 1, 2, \dots, N_{tr}$, where \mathbf{d}_n is defined as the desired outputs of the deep neural network model for inputs \mathbf{x}_n , and N_{tr} is defined as the total number of training samples. For example, for the parameter-

extraction model of microwave filters, \mathbf{x}_n should be a set of S -parameters of the filter and \mathbf{d}_n should be the coupling parameters corresponding to \mathbf{x}_n . For deep learning process, we use the standard error function [41] defined as

$$E(\mathbf{w}) = \sum_{n=1}^{N_{tr}} \left(\frac{1}{2} \sum_{j=1}^{N_y} (y_j(\mathbf{x}_n, \mathbf{w}) - d_{jn})^2 \right) \quad (3.12)$$

where $y_j(\mathbf{x}_n, \mathbf{w})$ is the j th output of the deep neural network model for \mathbf{x}_n , and d_{jn} is the j th element of \mathbf{d}_n .

During the training process, the derivatives of the error function in (3.12) with respect to all the weights in the deep neural network model need to be derived. Here, we propose a derivative calculation method based on the BP concept for multilayer perceptron (MLP) [41], extended to our proposed deep neural network structure. The total error function defined in (3.12) can be considered as the summation of the errors of the model for all training samples. The error for the n th training sample can be computed by [41]

$$E_n = \frac{1}{2} \sum_{j=1}^{N_y} (y_j(\mathbf{x}_n, \mathbf{w}) - d_{jn})^2 \quad (3.13)$$

To compute the derivatives for the proposed deep neural network model based on BP algorithm, we define δ_i^λ to represent the local gradient of E_n with respect to the total input to the i th neuron of λ th layer X_i^λ , i.e., $\delta_i^\lambda = \frac{\partial E_n}{\partial X_i^\lambda}$.

The δ_i^λ can be initialized at the output layer as

$$\delta_i^\Gamma = y_i(\mathbf{x}_n, \mathbf{w}) - d_{in} \quad (3.14)$$

The BP algorithm will start from δ_i^Γ and propagate this local gradient backward from the output layer through the hidden layers to the input layer of the proposed deep neural network model. In our proposed model, the hidden layers close to the output layer use the proposed smooth ReLUs as activation functions. The local gradient δ_i^λ in these hidden layers with the proposed smooth ReLUs can be derived as

$$\delta_i^\lambda = \begin{cases} \sum_{j \in B^{\lambda+1}} \delta_j^{\lambda+1} w_{ji}^{\lambda+1} & \text{if } X_i^\lambda > \mu \\ \left(\sum_{j \in B^{\lambda+1}} \delta_j^{\lambda+1} w_{ji}^{\lambda+1} \right) \left(\frac{1}{2\mu} X_i^\lambda + \frac{1}{2} \right) & \text{if } -\mu \leq X_i^\lambda \leq \mu, \\ 0 & \text{if } X_i^\lambda < -\mu \end{cases}, \quad (3.15)$$

$$\lambda = \Gamma - 1, \Gamma - 2, \dots, N_s + 3, N_s + 2$$

Here, in (3.15), we need to extend the definition of B^λ to the output layer, i.e., $\lambda = \Gamma$. Linear activation functions are used in the output layer, and consequently, all neurons in the output layer are active.

Subsequently, the BP algorithm will propagate this local gradient backward from hidden layers with smooth ReLUs to those with sigmoid functions in the proposed deep neural network model. For the topmost sigmoid hidden layer directly connected to the hidden layer with smooth ReLUs, the local gradient δ_i^λ can be derived as

$$\delta_i^\lambda = \left(\sum_{j \in B^{\lambda+1}} \delta_j^{\lambda+1} w_{ji}^{\lambda+1} \right) Z_i^\lambda (1 - Z_i^\lambda), \quad \lambda = N_s + 1 \quad (3.16)$$

For the other sigmoid hidden layers except for the topmost one, the computation of

local gradient δ_i^λ is similar to the standard BP [41] as

$$\delta_i^\lambda = \left(\sum_{j=1}^{N_{\lambda+1}} \delta_j^{\lambda+1} w_{ji}^{\lambda+1} \right) Z_i^\lambda (1 - Z_i^\lambda), \quad \lambda = N_s, N_s - 1, \dots, 3, 2 \quad (3.17)$$

Equations (3.16) and (3.17) are used to calculate the local gradient in different sigmoid hidden layers. Equation (3.16) is valid only in the sigmoid hidden layer that is directly connected to the hidden layer with smooth ReLUs. It is different from the standard BP because some of the hidden neurons in the hidden layer with smooth ReLUs may not be active.

After finishing the BP process, the derivatives of E_n with respect to the weight parameters in the λ th layer can be computed by [41]

$$\frac{\partial E_n}{\partial w_{ij}^\lambda} = \delta_i^\lambda Z_j^{\lambda-1}, \quad \lambda = \Gamma, \Gamma - 1, \dots, 3, 2 \quad (3.18)$$

where $Z_j^{\lambda-1}$ is calculated using (3.10).

If we directly train the hybrid deep neural network model with both sigmoid functions and the proposed smooth ReLUs, the vanishing gradient problem may exist. This is because the derivative of the smooth ReLU is not always as much as one when the hidden neuron with smooth ReLU is active. Thus we propose a three-stage deep learning algorithm for training the new deep neural network. The proposed three-stage deep learning algorithm can overcome the vanishing gradient problem for training the proposed deep neural network. The above derivative information will be used in the proposed three-stage deep learning algorithm. In order to make the deep learning algorithm simpler, we use the same number of neurons in each hidden layer. The number of neurons per layer is prefixed according to

experience before we start training. We determine the number of hidden neurons according to the complexity of the microwave modeling problem. The input dimension is related to the complexity of the modeling problem. If the input dimensions are relatively high, the number of hidden neurons per layer will be relatively large; otherwise, the number of hidden neurons per layer will be relatively small. In Stage I, we train the deep neural network with only sigmoid hidden layers and determine the value of N_s , i.e., the number of sigmoid layers. The number of hidden layers in a deep neural network should be three or more. Usually, we start from $N_s = 3$. A deep neural network with three sigmoid hidden layers usually has no vanishing gradient problem. We train the deep neural network with N_s sigmoid hidden layers to make the training error as small as possible. If the accuracy requirement of the model is satisfied, the deep neural network training is finished and we do not need Stages II and III. Otherwise, we add a new sigmoid layer again and again until the accuracy requirement of the model is satisfied or until the vanishing gradient problem begins to appear. Every time we add one sigmoid layer, we need to train the deep neural network and calculate the training error. Let E_{train} be defined as the training error at the end of training of the deep neural network including the recently added sigmoid layer. Let E_b represent the training error of the deep neural network before the recent addition of the sigmoid layer. If $E_{train} < E_b$ and at the same time, E_{train} still cannot be reduced to the required error threshold, it means that the deep neural network is in underlearning state. Upon encountering such a state, we need to add a new sigmoid hidden layer again. If $E_{train} \geq E_b$, and E_{train} does not decrease even after many epochs of training, it means that there is the

vanishing gradient problem. In this case, we delete the last added sigmoid hidden layer, fix the total number of sigmoid layers N_s , and proceed to Stage II.

In Stage II, the purpose is to continue adding more hidden layers so that the training error of the deep neural network can be reduced further by training. Hidden layers with proposed smooth ReLUs are expected to be added according to the structure of the proposed deep neural network shown in Fig. 3.3. However, if we add hidden layers with proposed smooth ReLUs directly, the vanishing gradient problem may arise as the total number of hidden layers (including those already added in Stage I) is already large. This is because the derivative of the smooth ReLU is in the range of $[0,1]$ according to (3.7). Therefore, instead of adding hidden layers with smooth ReLUs directly, we propose to temporarily add hidden layers with conventional ReLUs (which will not lead to the vanishing gradient problem) above the trained sigmoid hidden layers. Consequently, within Stage II, the calculation of local gradient δ_i^λ for the BP process in the ReLU hidden layers will become

$$\delta_i^\lambda = \begin{cases} \sum_{j \in B^{\lambda+1}} \delta_j^{\lambda+1} w_{ji}^{\lambda+1} & \text{if } X_i^\lambda > 0 \\ 0 & \text{if } X_i^\lambda \leq 0 \end{cases}, \lambda = \Gamma - 1, \Gamma - 2, \dots, N_s + 3, N_s + 2 \quad (3.19)$$

Instead of (3.15), (3.19) is used to calculate the local gradient in the ReLU hidden layers in the training process of Stage II. The local gradients in other hidden layers are calculated as described in (3.16) and (3.17). A new hidden layer with conventional ReLUs is added again and again until the accuracy requirement of the deep neural network model is satisfied. In this way, we can determine the number

of ReLU hidden layers, i.e., the value of N_r . From here, we proceed to Stage III.

In Stage III, we use the proposed smooth ReLUs to replace all the conventional non-smooth ReLUs in the deep neural network model trained from Stages I and II. The values of the proposed smooth ReLU and the conventional ReLU are mostly close according to Fig. 3.2. Therefore, the replacement of conventional ReLUs by smooth ReLUs may slightly affect the output values of the deep neural network model. We refine the accuracy of the model by further training the deep neural network that has N_s hidden layers with sigmoid functions and N_r hidden layers with smooth ReLUs. The final deep neural network trained from Stage III will provide smooth input-output relationships required for microwave modeling.

3.2.4 Process of Developing the Deep Neural Network for Microwave Component Modeling

The process of developing the deep neural network for high-dimensional microwave component modeling can be summarized as follows.

- Step 1) Define the data range of model inputs, then generate randomly distributed training data and test data in the defined range (for example, using microwave simulators). Determine the number of hidden neurons per layer. Define the required accuracy threshold of the model as E_r . Initialize $N_s = 3$, $N_r = 0$.
- Step 2) Use the training data to train a deep neural network with N_s sigmoid hidden layers. Calculate E_{train} . If $N_s = 3$, go to Step 4).
- Step 3) Check if the vanishing gradient problem exists or not by comparing the

- values of E_{train} and E_b . If $E_{train} \geq E_b$, there exists the vanishing gradient problem, go to Step 7); else if $E_{train} < E_b$, go to Step 4).
- Step 4) Check if the training is in underlearning state. If $E_{train} > E_r$, set $E_b = E_{train}$, add one hidden layer with sigmoid functions, i.e., $N_s = N_s + 1$, and go back to Step 2); else if $E_{train} \leq E_r$, go to Step 5).
- Step 5) Test the deep neural network. Let E_{test} represent the test error. If $E_{test} > E_r$, which means that the training is in overlearning state, go to Step 6); else if $E_{test} \leq E_r$, stop the deep neural network training process.
- Step 6) Add more training data, train the deep neural network, calculate E_{train} and go back to Step 4).
- Step 7) Delete the last added sigmoid hidden layer, thus $N_s = N_s - 1$, and train the deep neural network with N_s sigmoid hidden layers to make the training error as small as possible. Add one hidden layer with conventional non-smooth ReLUs to the trained deep neural network. Set $N_r = 1$.
- Step 8) Train the deep neural network that has N_s hidden layers with sigmoid functions and N_r hidden layers with conventional ReLUs. Calculate E_{train} .
- Step 9) Check if the training is in underlearning state. If $E_{train} > E_r$, add one hidden layer with conventional ReLUs, i.e., $N_r = N_r + 1$, and go back to Step 8); else if $E_{train} \leq E_r$, go to Step 10).
- Step 10) Test the deep neural network model. If $E_{test} > E_r$, which means that the training is in overlearning state, add more training data and go back to Step 8); else if $E_{test} \leq E_r$, go to Step 11).
- Step 11) Use the proposed smooth ReLUs to replace all the conventional non-smooth

ReLU's in the trained deep neural network.

Step 12) Train the deep neural network that has N_s hidden layers with sigmoid functions and N_r hidden layers with proposed smooth ReLU's to refine the model accuracy. Stop the modeling process.

Fig. 3.4 shows the flow diagram of the process for developing the proposed deep neural network model. It includes three stages. Stage I includes Steps 2) to 6). Stage II includes Steps 7) to 10). Stage III includes Steps 11) to 12). The concept of using underlearning and overlearning criterion in the training process is similar to that in the automated model generation (AMG) algorithm presented in [25].

3.2.5 Discussion

For a given modeling problem, there is a choice whether to use the shallow neural network or the deep neural network. We can select either the shallow or the deep neural network according to the complexity of the modeling problem. The input dimension is related to the complexity of the modeling problem. If the input dimension is low, a shallow neural network can be selected for the modeling problem. If the input dimension is high, the deep neural network will be more effective for the modeling problem. For high-dimensional microwave modeling problems, the relationship between model inputs and outputs becomes more complex. Deep neural networks have already been proven to be more powerful than the shallow neural network for modeling complex input-output relationships [29], [30]. The standard deep neural networks with non-smooth outputs are not suitable for microwave modeling problems with smooth input-output relationships, while the proposed deep neu-

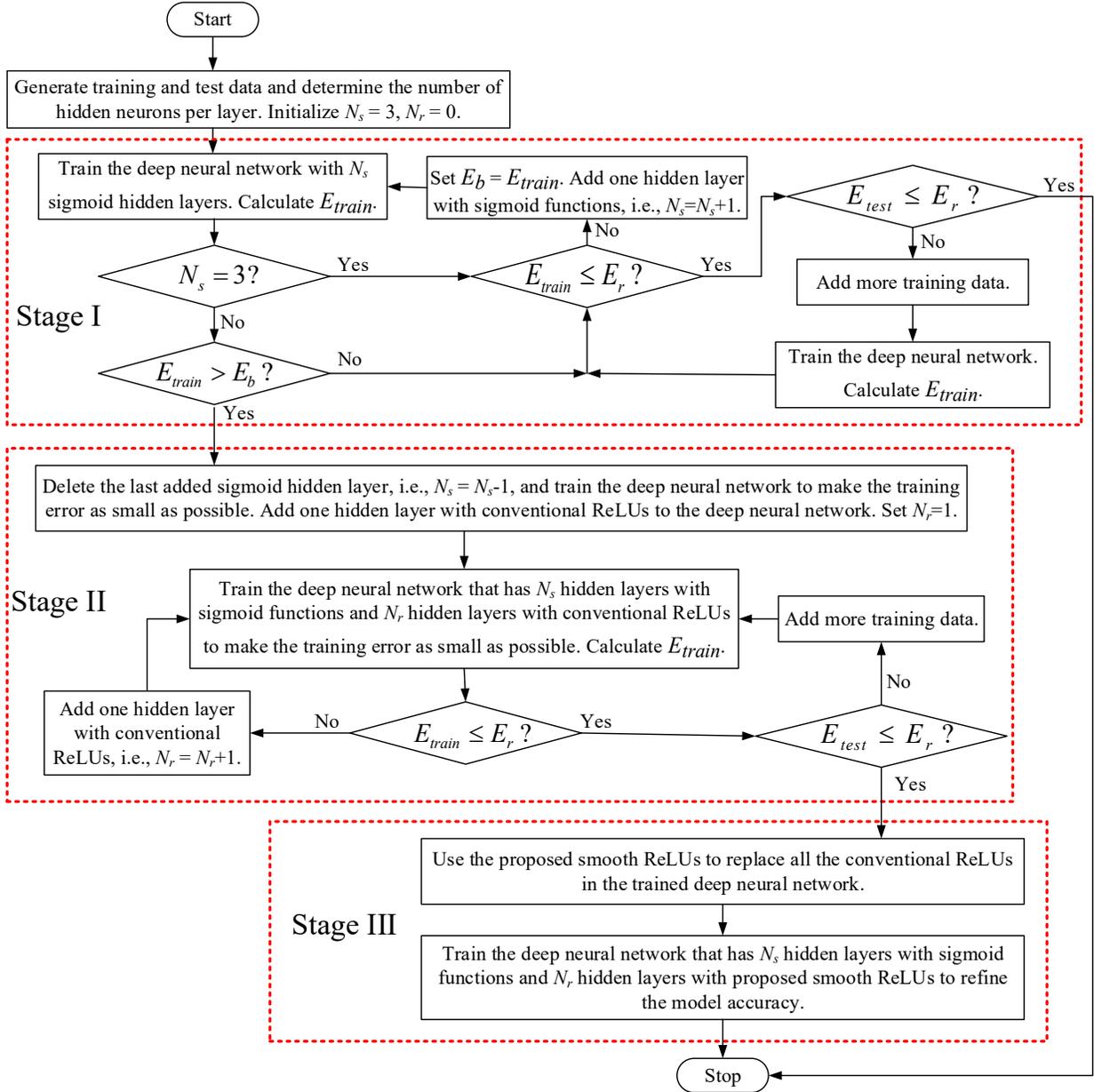


Figure 3.4: The flow diagram of the process for developing the proposed deep neural network model.

ral network can represent the smooth input-output relationship. Therefore, the proposed deep neural network is suitable for high-dimensional microwave modeling problems to learn the smooth and complex input-output relationships.

The value of μ in (3.6) can affect the smoothness of the model outputs and the accuracy of the deep neural network model after replacement of conventional ReLUs by smooth ReLUs. The input-output relationships of microwave modeling problems are smooth. If μ is too small, the model outputs may not be smooth enough. However, if the value of μ is too large, the vanishing gradient problem may be more serious in the training process of Stage III. In other words, large μ will make it hard to refine the model to the required accuracy. We recommend $\mu = 0.05$ to reach a good compromise between these two aspects.

3.3 Application Examples

3.3.1 Modeling of a Fourth-Order Multicoupled Cavity Filter

We apply the proposed deep neural network technique to the high-dimensional parameter-extraction modeling of microwave filters in this example. For the purposes of design or diagnosis and tuning, we often need to develop parameter-extraction models of microwave circuits [156]. This is especially true for microwave filters [158]-[164]. The purpose of the high-dimensional parameter-extraction model in this example is to estimate the coupling parameters from the given S -parameters. The inputs to the model are S -parameters at various frequencies, which describe the electrical behavior of microwave filters, and the outputs of the model are cou-

pling parameters. A classical method to extract the coupling parameters is the optimization-based method [159]-[162], [164]. For a given set of S -parameters over a frequency range, the filter circuit is simulated iteratively to find the optimal coupling parameters such that the filter response matches the given S -parameters.

In this example, we use the proposed deep neural network technique to develop a parameter-extraction model for a fourth-order multicoupled cavity filter with a 4-GHz center frequency and a 40-MHz bandwidth [163]. To develop the proposed deep neural network model, we need to determine the model inputs, the model outputs, the number of hidden neurons per layer, the number of hidden layers using sigmoid functions, the number of hidden layers using ReLUs, and the values for the weight parameters.

The ideal coupling matrix for this filter example is

$$\mathbf{M}_{ideal} = \begin{bmatrix} -0.0157 & 0.8950 & 0 & -0.2346 \\ 0.8950 & -0.010 & 0.8080 & 0 \\ 0 & 0.8080 & -0.010 & 0.8950 \\ -0.2346 & 0 & 0.8950 & -0.0157 \end{bmatrix} \quad (3.20)$$

The outputs of the deep neural network model are the eight nonzero coupling parameters in the \mathbf{M}_{ideal} matrix, i.e., $\mathbf{y} = [M_{11} \ M_{22} \ M_{33} \ M_{44} \ M_{12} \ M_{23} \ M_{34} \ M_{14}]^T$. For this fourth-order filter, the frequency band used for the model is from 3930 MHz to 4070 MHz, which includes the passband plus a portion of stopbands on both sides of the passband. We use S_{11} in dB at 35 frequency samples in the frequency band as inputs to the deep neural network model, i.e., $\mathbf{x} = [dB(S_{11}(f_1)) \ dB(S_{11}(f_2)) \ \dots \ dB(S_{11}(f_{34})) \ dB(S_{11}(f_{35}))]^T$, where $f_1, f_2, \dots, f_{34}, f_{35}$ are 35 frequency samples. The number

of model inputs is determined to be 35 because it was observed as the minimum number needed to achieve the required model accuracy according to numerical experiments. The inputs of the model are in high dimension, i.e., 35 dimensions. The commonly used shallow ANNs with only one or two hidden layers cannot represent this high-dimensional input-output relationship effectively. Our proposed deep neural network modeling technique is suitable for this high-dimensional modeling problem.

In order to generate the training and test data, we assume a tolerance of ± 0.3 for every nonzero coupling parameter. We use randomly distributed samples in this range for coupling parameters. For each and every sample of coupling parameters, we simulate the filter to obtain corresponding S -parameters. By swapping the data of coupling parameters and S -parameters, we can get training and test data for the parameter-extraction model.

We use the proposed deep neural network technique to develop the parameter-extraction model for the fourth-order filter. We set the test error threshold for this modeling example as 2%. The number of hidden neurons in each hidden layer is 100 for this modeling example. The results of three-stage training are summarized in Table 3.1. In Stage I, we train the deep neural network with only sigmoid layers. The number of sigmoid hidden layers can be automatically determined by the proposed deep learning algorithm. From Table 3.1, we can see that deep neural network with eight sigmoid hidden layers can reach the test error of 2.62%. The corresponding error between S -parameters from the test data and those calculated from extracted coupling parameters is 1.66%. The training error of the deep neu-

Table 3.1: Results of Different Stages in the Training Process of the Proposed Deep Neural Network Parameter-Extraction Model for the Fourth-Order Filter Example

Training stage	Hidden layers of the deep neural network	Training error	Test error	Model type
Stage I	8 sigmoid layers	2.04%	2.62%	Smooth
	9 sigmoid layers	17.33%	17.33%	model
Stage II	8 layers with sigmoid functions and 3 layers with conventional ReLUs	1.21%	1.89%	Non-smooth model
Stage III	8 layers with sigmoid functions and 3 layers with proposed smooth ReLUs	1.22%	1.88%	Smooth model

ral network with nine sigmoid hidden layers remains high and cannot be reduced anymore by continued training. In other words, the vanishing gradient problem begins to appear when the number of hidden layers increases to nine. Therefore, the number of sigmoid hidden layers in the deep neural network is fixed to eight. If fewer sigmoid layers are used, the deep neural network would require more hidden neurons with ReLUs to achieve the same model accuracy. If more sigmoid layers are used (e.g., nine sigmoid layers as shown in Table 3.1), there would be the vanishing gradient problem in the training process. In Stage II, we begin to add hidden layers with conventional ReLUs to the deep neural network with eight sigmoid hidden layers trained in Stage I. After the training process of Stage II, the hybrid deep neural network that has eight hidden layers with sigmoid functions and three hidden layers with conventional non-smooth ReLUs can reach the test error

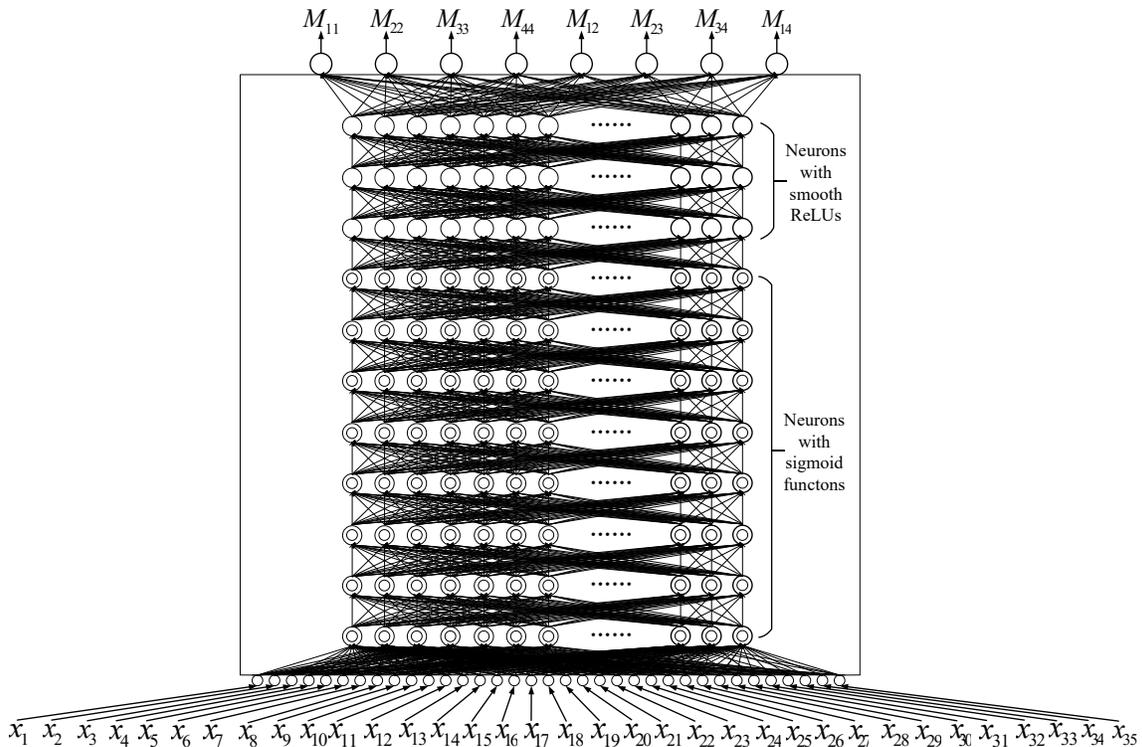


Figure 3.5: The structure of the deep neural network high-dimensional parameter-extraction model for the fourth-order filter. The model has 35 inputs, where $x_i = dB(S_{11}(f_i)), i = 1, 2, \dots, 35$. We used a 13-layer deep neural network for this example. The hidden layers include eight layers with sigmoid functions and three layers with the proposed smooth ReLUs. Each hidden layer is composed of 100 hidden neurons. All the hidden layers are shown in the box.

of 1.89%. In Stage III, we use the proposed smooth ReLU with $\mu = 0.05$ to replace the conventional non-smooth ReLU in the deep neural network model obtained from Stage II. After the training process of Stage III, the proposed deep neural network that has eight hidden layers with sigmoid functions and three hidden layers with smooth ReLUs can reach the test error of 1.88% with smooth outputs and continuous derivatives. The corresponding error between S -parameters from the test

data and those calculated from extracted coupling parameters is 0.87%. The overall training data for this example include 500 000 samples, and the overall testing data for this example include 300 000 samples. The structure of the deep neural network parameter-extraction model for this fourth-order filter example is shown in Fig. 3.5. An example of the derivative is shown in Fig. 3.6. The conventional non-smooth ReLU leads to discontinuity of the derivative of the model as shown in Fig. 3.6. The proposed smooth ReLU can obviously improve the continuity of the derivatives of the proposed deep neural network parameter-extraction model. The continuous derivatives mean that the model outputs are smooth.

For comparison purposes, we also use the shallow neural network to develop a parameter-extraction model for this fourth-order filter. A shallow neural network with two hidden layers and 253 hidden neurons in each hidden layer is trained with the training data. This shallow neural network has the similar number of internal training parameters as that of the deep neural network with eight hidden layers and 100 hidden neurons in each layer from Stage I. Here, training parameters include internal weight parameters and biases in the neural network. The comparison of the shallow and deep neural networks is shown in Table 3.2. The test error of the shallow neural network is 3.85% while that of the deep neural network is 2.62%. We can see that deep neural network is more effective for improving modeling accuracy than shallow neural network when both neural networks have a similar number of training parameters and the same type of activation functions.

In order to illustrate the benefit of combining sigmoid function with ReLU, we try to develop a parameter-extraction model for this filter using deep neural network

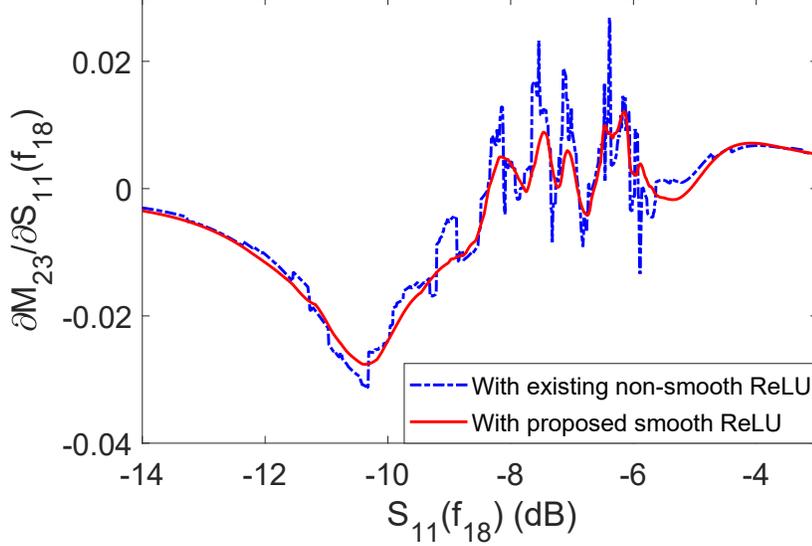


Figure 3.6: The comparison of continuity of the partial derivative $\partial M_{23}/\partial S_{11}(f_{18})$ between the hybrid deep neural network with conventional non-smooth ReLUs (i.e., the model obtained after Stage II) and that with the proposed smooth ReLUs (i.e., the model obtained after Stage III). This is for the fourth-order filter example. The horizontal axis represents S_{11} in dB at f_{18} , and the vertical axis represents the partial derivative of M_{23} with respect to S_{11} in dB at f_{18} . Solid line represents the derivative of the hybrid deep neural network with the proposed smooth ReLUs and dashdot line represents the derivative of the hybrid deep neural network with the conventional non-smooth ReLUs. The conventional non-smooth ReLU leads to discontinuity of the derivative of the model as shown in the figure, where the dashdot line has vertical rises and vertical drops. The proposed smooth ReLU can obviously improve the continuity of the derivatives of the proposed deep neural network model.

with only conventional ReLUs. In this thesis, we define the deep neural network with only conventional ReLUs as pure ReLU deep neural network. In other words, in a pure ReLU deep neural network, conventional ReLU functions are used by all hidden neurons in all hidden layers, while sigmoid functions are not used by any

Table 3.2: Comparison of Shallow and Deep Neural Networks with a Similar Number of Training Parameters for Parameter-Extraction Modeling of the Fourth-Order Filter Example

Neural network structure	Hidden neurons per layer	Number of training parameters	Training error	Test error
shallow neural network (2 sigmoid hidden layers)	253	75k	3.29%	3.85%
deep neural network (8 sigmoid hidden layers)	100	75k	2.04%	2.62%

hidden neuron in any hidden layer. Pure ReLU deep neural networks with the same number of layers as or more number of layers than that of the proposed deep neural network model are utilized to develop the parameter-extraction model for the fourth-order filter. The number of hidden neurons in each hidden layer is 100 for both the proposed deep neural network and the pure ReLU deep neural networks. The comparison of results is summarized in Table 3.3. Our proposed deep neural network can achieve a better test error than pure ReLU deep neural network with the same number of hidden neurons. Adding more hidden neurons in pure ReLU deep neural network can reduce the test error, and therefore, the pure ReLU deep neural network will need much more hidden neurons to achieve a similar accuracy as our proposed deep neural network.

After we obtained the parameter-extraction model using our proposed deep neural network technique for this filter example, we try to use the trained model to extract the coupling parameters for a slightly and a highly detuned filter to examine

Table 3.3: Comparison of the Proposed Hybrid Deep Neural Network and Pure ReLU Deep Neural Network for Parameter-Extraction Modeling of the Fourth-Order Filter Example

Structure of deep neural networks	Training error	Test error
The proposed 13-layer hybrid deep neural network ¹	1.22%	1.88%
13-layer pure ReLU network	2.73%	3.38%
16-layer pure ReLU network	2.23%	3.27%

¹ The proposed 13-layer deep neural network is a hybrid network that has eight hidden layers with sigmoid functions and three hidden layers with proposed smooth ReLUs.

the developed parameter-extraction model.

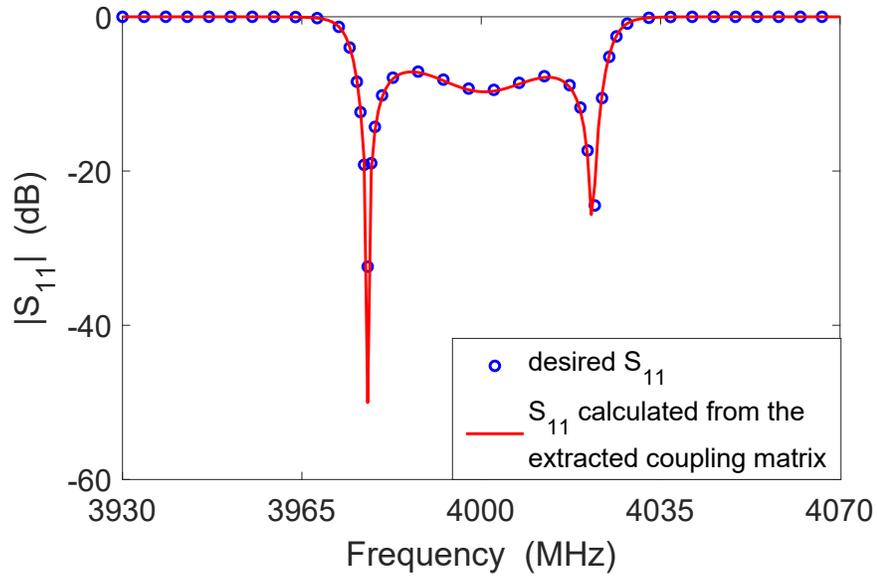
The desired coupling parameters for the slightly detuned filter are

$$\mathbf{M}_{desired} = \begin{bmatrix} -0.0607 & 0.7950 & 0 & -0.1346 \\ 0.7950 & 0.060 & 0.9080 & 0 \\ 0 & 0.9080 & -0.040 & 0.7950 \\ -0.1346 & 0 & 0.7950 & 0.0343 \end{bmatrix} \quad (3.21)$$

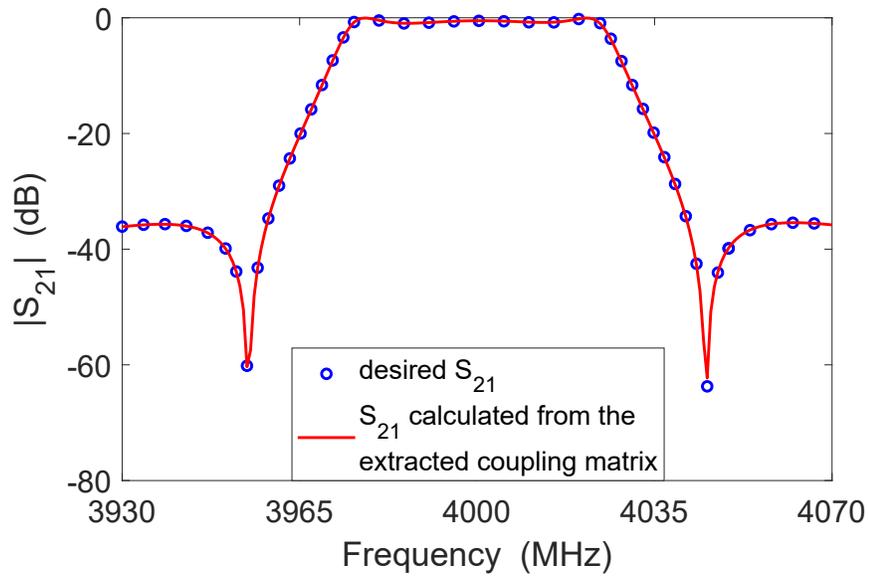
The coupling parameters extracted using the proposed deep neural network parameter-extraction model are

$$\mathbf{M}_{extracted} = \begin{bmatrix} -0.0677 & 0.7912 & 0 & -0.1344 \\ 0.7912 & 0.0588 & 0.9035 & 0 \\ 0 & 0.9035 & -0.0369 & 0.8025 \\ -0.1344 & 0 & 0.8025 & 0.0370 \end{bmatrix} \quad (3.22)$$

Fig. 3.7 shows the S -parameters from the desired and extracted coupling matrix for the slightly detuned filter.



(a)



(b)

Figure 3.7: The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the slightly detuned fourth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21} .

According to (3.21), (3.22), and Fig. 3.7, a good match between the desired and extracted coupling parameters along with an excellent match between the responses from desired and extracted coupling matrices have been achieved for the slightly detuned filter.

For the highly detuned filter, the desired coupling parameters are

$$\mathbf{M}_{desired} = \begin{bmatrix} -0.2657 & 1.145 & 0 & -0.4846 \\ 1.145 & -0.24 & 1.058 & 0 \\ 0 & 1.058 & 0.26 & 0.645 \\ -0.4846 & 0 & 0.645 & 0.2343 \end{bmatrix} \quad (3.23)$$

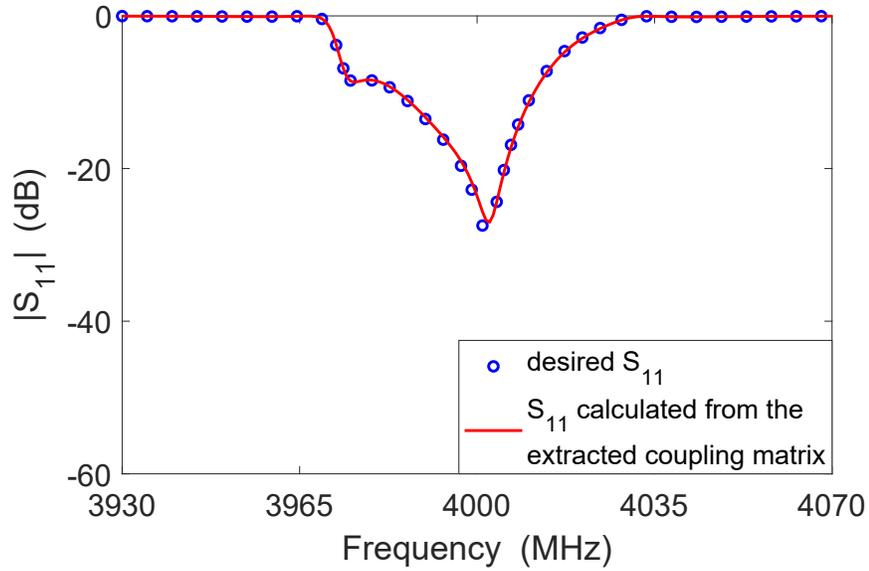
The coupling parameters extracted using the proposed deep neural network parameter-extraction model are

$$\mathbf{M}_{extracted} = \begin{bmatrix} -0.2703 & 1.1348 & 0 & -0.4917 \\ 1.1348 & -0.2443 & 1.0658 & 0 \\ 0 & 1.0658 & 0.2545 & 0.6421 \\ -0.4917 & 0 & 0.6421 & 0.2355 \end{bmatrix} \quad (3.24)$$

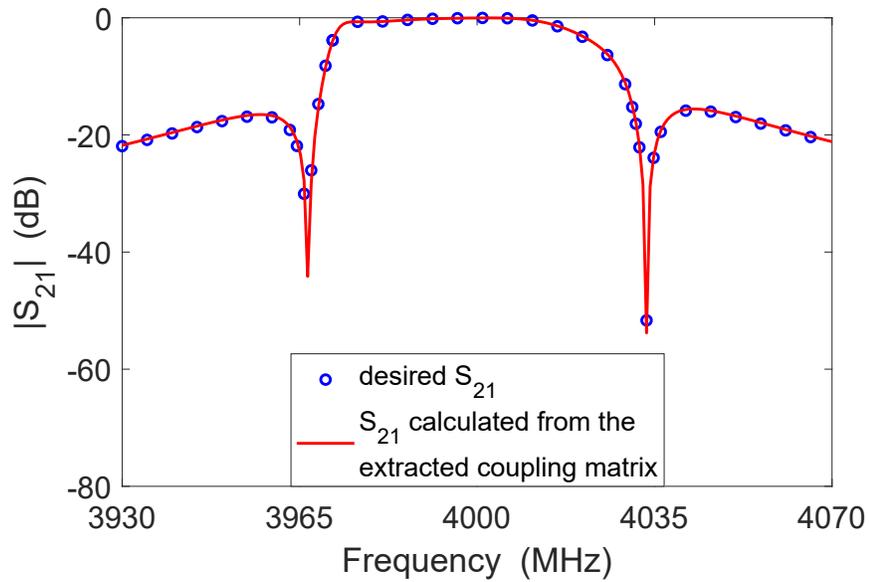
Fig. 3.8 shows the S -parameters from the desired and extracted coupling matrix for the highly detuned filter.

According to (3.23), (3.24), and Fig. 3.8, a good match between the desired and extracted coupling parameters along with an excellent match between the responses from desired and extracted coupling matrices have been achieved for the highly detuned filter.

From the test results, we can see that the proposed deep neural network technique is suitable for performing the parameter extraction of microwave filters. Our



(a)



(b)

Figure 3.8: The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the highly detuned fourth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21} .

proposed deep neural network can provide parameter extraction solutions for both the slightly and highly detuned filters instantly, while the conventional optimization method will take more CPU time to extract the solutions by repetitively simulating the filter circuit during optimization iterations.

3.3.2 Modeling of a Sixth-Order Multicoupled Cavity Filter

The development of the proposed deep neural network technique for parameter-extraction modeling of a sixth-order multicoupled cavity filter [164] is illustrated in this example. The filter is centered at 11785.5 MHz with a 56.2-MHz bandwidth. The ideal coupling matrix for this filter is

$$\mathbf{M}_{ideal} = \begin{bmatrix} -0.0473 & 0.8489 & 0 & 0 & 0 & 0 \\ 0.8489 & -0.0204 & 0.6064 & 0 & 0 & 0 \\ 0 & 0.6064 & -0.0305 & 0.5106 & 0 & -0.2783 \\ 0 & 0 & 0.5106 & 0.0005 & 0.7709 & 0 \\ 0 & 0 & 0 & 0.7709 & -0.0026 & 0.7898 \\ 0 & 0 & -0.2783 & 0 & 0.7898 & 0.0177 \end{bmatrix} \quad (3.25)$$

There are twelve nonzero coupling parameters in the \mathbf{M}_{ideal} matrix, and therefore, the outputs of the model can be represented as $\mathbf{y} = [M_{11} M_{22} M_{33} M_{44} M_{55} M_{66} M_{12} M_{23} M_{34} M_{45} M_{56} M_{36}]^T$. For this sixth-order filter, the frequency band for the model is from 11720 MHz to 11850 MHz, which includes the passband from 11757.4 MHz to 11813.6 MHz plus a portion of stopbands on both sides. We use S_{11} in dB at 41 frequency samples in the frequency band as inputs to the model, i.e., $\mathbf{x} = [dB(S_{11}(f_1)) dB(S_{11}(f_2)) \dots dB(S_{11}(f_{40})) dB(S_{11}(f_{41}))]^T$, where $f_1, f_2, \dots, f_{40}, f_{41}$ are 41 frequency samples. The inputs of the model are in high dimension,

i.e., 41 dimensions. In order to generate the training and test data, we assume a tolerance of ± 0.3 for every nonzero coupling parameter. We use randomly distributed samples in this range for coupling parameters. For each and every sample of coupling parameters, we simulate the filter to obtain corresponding S -parameters. By swapping the coupling parameters and S -parameters, we can get training and test data for the parameter-extraction modeling.

We use the proposed deep neural network technique to develop the parameter-extraction model for the sixth-order filter. We set the test error threshold for this modeling example as 2%. The input-output relationship of the sixth-order filter example is more complex than that of the fourth-order filter in the previous subsection. Therefore, compared to the fourth-order filter example, more hidden neurons per layer are used for the sixth-order filter example. The number of hidden neurons in each hidden layer is 200 for this example. The results of three-stage training are summarized in Table 3.4. In Stage I, we can see that the deep neural network with eight sigmoid hidden layers can reach the test error of 2.40%. The training error of the deep neural network with nine sigmoid hidden layers remains high and cannot be reduced anymore by continued training. Therefore, the vanishing gradient problem begins to appear when the number of hidden layers increases to nine. In Stage II, we begin to add hidden layers with conventional ReLUs to the deep neural network with eight sigmoid hidden layers trained in Stage I. After the training process of Stage II, the hybrid deep neural network that has eight hidden layers with sigmoid functions and four hidden layers with conventional non-smooth ReLUs can reach the test error of 1.85%. In Stage III, we use the proposed smooth ReLU with $\mu = 0.05$

Table 3.4: Results of Different Stages in the Training Process of the Proposed Deep Neural Network Parameter-Extraction Model for the Sixth-Order Filter Example

Training stage	Hidden layers of the deep neural network	Training error	Test error	Model type
Stage I	8 sigmoid layers	1.95%	2.40%	Smooth
	9 sigmoid layers	17.33%	17.34%	model
Stage II	8 layers with sigmoid functions and 4 layers with conventional ReLUs	1.34%	1.85%	Non-smooth model
Stage III	8 layers with sigmoid functions and 4 layers with proposed smooth ReLUs	1.31%	1.79%	Smooth model

to replace the conventional non-smooth ReLU in the deep neural network model obtained from Stage II. After the training process of Stage III, the proposed deep neural network that has eight hidden layers with sigmoid functions and four hidden layers with proposed smooth ReLUs can reach the test error of 1.79% with smooth outputs and continuous derivatives. An example of the derivative is shown in Fig. 3.9. The conventional non-smooth ReLU leads to discontinuity of the derivative of the model as shown in Fig. 3.9. The proposed smooth ReLU can obviously improve the continuity of the derivatives of the proposed deep neural network parameter-extraction model. The continuous derivatives mean that the model outputs are smooth.

For comparison purposes, we use the shallow neural network to develop a parameter-extraction model for this sixth-order filter. A shallow neural network with two hid-

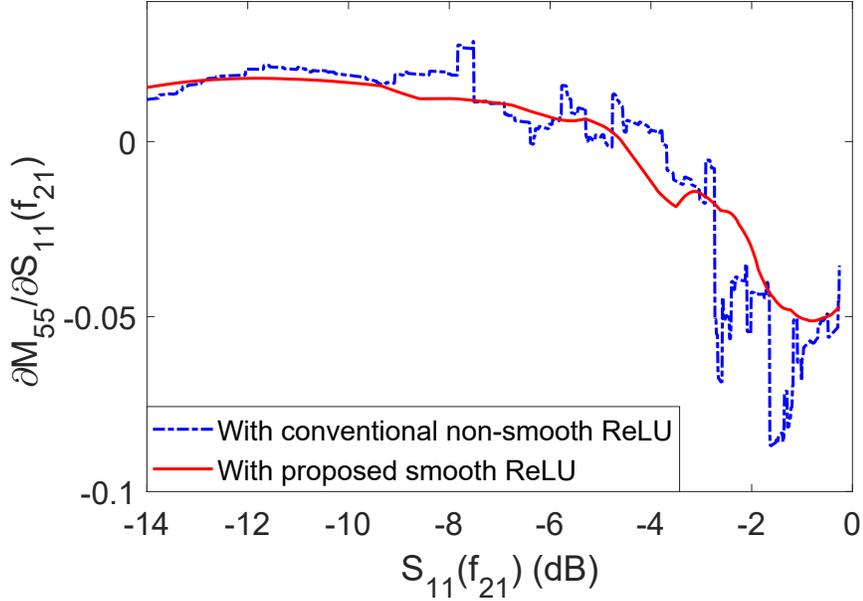


Figure 3.9: The comparison of continuity of the partial derivative $\partial M_{55}/\partial S_{11}(f_{21})$ between the hybrid deep neural network with conventional non-smooth ReLUs (i.e., the model obtained after Stage II) and that with smooth ReLUs (i.e., the model obtained after Stage III). This is for the sixth-order filter example. The horizontal axis represents S_{11} in dB at f_{21} , and the vertical axis represents the partial derivative of M_{55} with respect to S_{11} in dB at f_{21} . Solid line represents the derivative of the hybrid deep neural network with the proposed smooth ReLUs and dashdot line represents the derivative of the hybrid deep neural network with conventional non-smooth ReLUs. The conventional non-smooth ReLU leads to discontinuity of the derivative of the model as shown in the figure, where the dashdot line has vertical rises and vertical drops. The proposed smooth ReLU can obviously improve the continuity of the derivatives of the proposed deep neural network parameter-extraction model.

den layers and 514 hidden neurons in each hidden layer is trained with the training data. This shallow neural network has the similar number of internal training parameters as that of the deep neural network with eight sigmoid hidden layers and 200 hidden neurons in each layer from Stage I. Here, training parameters include

Table 3.5: Comparison of Shallow and Deep Neural Networks with a Similar Number of Training Parameters for Parameter-Extraction Modeling of the Sixth-Order Filter Example

Structure of neural network	Hidden neurons per layer	Number of training parameters	Training error	Test error
shallow neural network (2 sigmoid hidden layers)	514	292k	4.22%	4.89%
deep neural network (8 sigmoid hidden layers)	200	292k	1.95%	2.40%

internal weight parameters and biases in the neural network. The CPU time for training the shallow and deep neural network models are 4.86 hours and 4.64 hours, respectively, which are close. The comparison of the shallow and deep neural networks is shown in Table 3.5. The test error of the shallow neural network is 4.89% while that of the deep neural network is 2.40%. We can see that deep neural network is more effective for improving modeling accuracy than shallow neural network when both neural networks have a similar number of training parameters and the same type of activation functions.

In order to illustrate the benefit of combining sigmoid function with ReLU, we try to develop a parameter-extraction model for this filter using pure ReLU deep neural network with the same number of layers as or more number of layers than that of the proposed deep neural network model. The number of hidden neurons in each hidden layer is 200 for both the proposed deep neural network and the pure ReLU deep neural networks. The comparison of results is summarized in Table 3.6.

Table 3.6: Comparison of the Proposed Hybrid Deep Neural Network and Pure ReLU Deep Neural Network for Parameter-Extraction Modeling of the Sixth-Order Filter Example

Structure of deep neural network	Training error	Test error
The proposed 14-layer hybrid deep neural network ¹	1.31%	1.79%
14-layer pure ReLU network	2.68%	3.16%
16-layer pure ReLU network	2.43%	3.00%
20-layer pure ReLU network	2.20%	2.73%

¹ The proposed 14-layer deep neural network is a hybrid network that has 8 hidden layers with sigmoid functions and 4 hidden layers with proposed smooth ReLUs.

Our proposed deep neural network can achieve a better test error than pure ReLU deep neural network with the same number of hidden neurons. Adding more hidden neurons in pure ReLU deep neural network can reduce the test error. The CPU time for training the proposed deep neural network model is 9.11 hours, while the CPU time for training the three pure ReLU deep neural networks are 4.95 hours, 7.02 hours, and 11.82 hours, respectively. We can see that training of the pure ReLU deep neural network takes less time than that of the proposed deep neural network when both neural networks have the same number of hidden neurons. As the number of hidden layers in the pure ReLU deep neural network increases, the training time used is increased as well. The CPU time for training the 20-layer pure ReLU deep neural network is longer than that for training the proposed deep neural network, but the model accuracy of the pure ReLU deep neural network is still lower than

the proposed deep neural network model. From the comparison results in Table 3.6 and the comparison of CPU time, we can see that pure ReLU deep neural network will need much more hidden neurons to achieve similar accuracy as our proposed deep neural network, which will also lead to longer CPU time.

After we obtained the parameter-extraction model using our proposed deep neural network technique for this filter example, we try to use the trained parameter-extraction model to extract the coupling parameters for a slightly and a highly detuned filter to examine the developed parameter-extraction model.

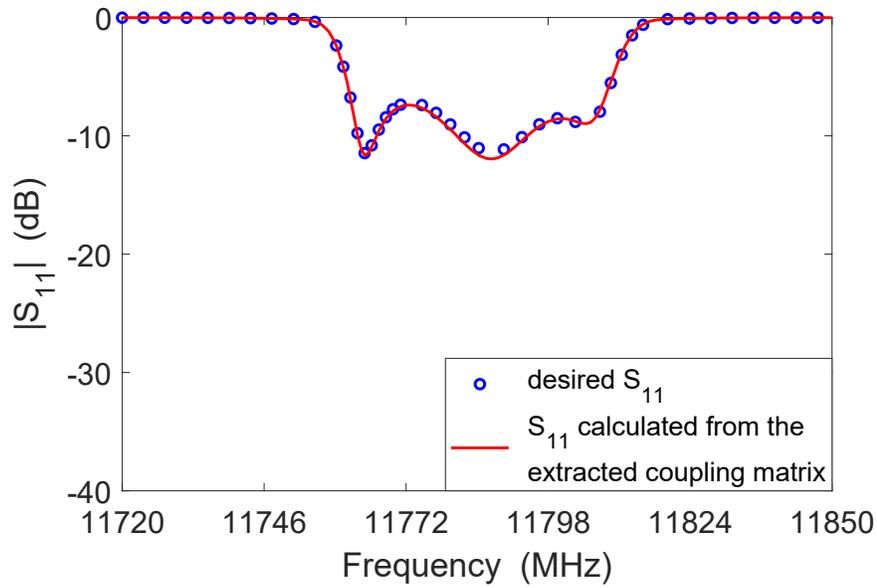
The desired coupling parameters for the slightly detuned filter are

$$\mathbf{M}_{desired} = \begin{bmatrix} 0.0127 & 0.9489 & 0 & 0 & 0 & 0 \\ 0.9489 & 0.0296 & 0.7064 & 0 & 0 & 0 \\ 0 & 0.7064 & -0.0805 & 0.4106 & 0 & -0.1783 \\ 0 & 0 & 0.4106 & 0.0505 & 0.6709 & 0 \\ 0 & 0 & 0 & 0.6709 & -0.0526 & 0.6898 \\ 0 & 0 & -0.1783 & 0 & 0.6898 & -0.0323 \end{bmatrix} \quad (3.26)$$

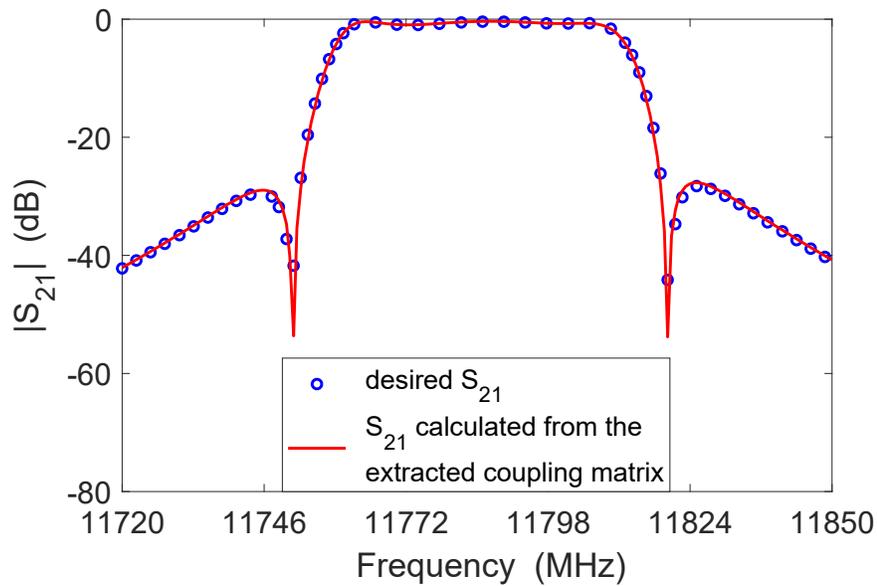
The coupling parameters extracted using the proposed deep neural network parameter-extraction model are

$$\mathbf{M}_{extracted} = \begin{bmatrix} 0.0173 & 0.9483 & 0 & 0 & 0 & 0 \\ 0.9483 & 0.0215 & 0.6993 & 0 & 0 & 0 \\ 0 & 0.6993 & -0.0753 & 0.4080 & 0 & -0.1828 \\ 0 & 0 & 0.4080 & 0.0500 & 0.6715 & 0 \\ 0 & 0 & 0 & 0.6715 & -0.0520 & 0.6918 \\ 0 & 0 & -0.1828 & 0 & 0.6918 & -0.0412 \end{bmatrix} \quad (3.27)$$

Fig. 3.10 illustrates the S -parameters from the desired and extracted coupling



(a)



(b)

Figure 3.10: The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the slightly detuned sixth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21} .

matrix for the slightly detuned filter.

According to (3.26), (3.27), and Fig. 3.10, a good match between the desired and extracted coupling parameters along with an excellent match between the responses from desired and extracted coupling matrices have been achieved for the slightly detuned filter.

For the highly detuned filter, the desired coupling parameters are

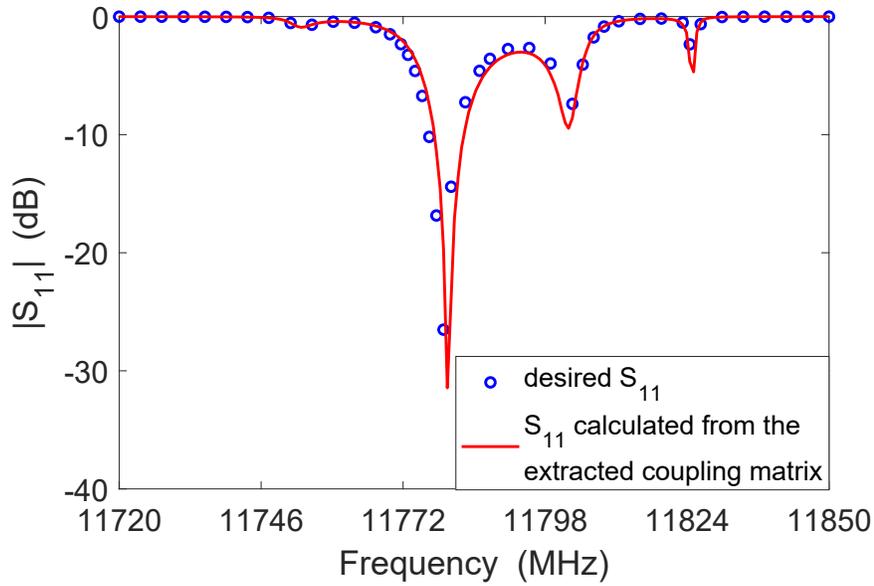
$$\mathbf{M}_{desired} = \begin{bmatrix} 0.2027 & 0.5989 & 0 & 0 & 0 & 0 \\ 0.5989 & 0.2296 & 0.8564 & 0 & 0 & 0 \\ 0 & 0.8564 & -0.2805 & 0.7606 & 0 & -0.0283 \\ 0 & 0 & 0.7606 & -0.2495 & 0.5209 & 0 \\ 0 & 0 & 0 & 0.5209 & -0.2526 & 0.5398 \\ 0 & 0 & -0.0283 & 0 & 0.5398 & 0.2677 \end{bmatrix} \quad (3.28)$$

The coupling parameters extracted using the proposed deep neural network parameter-extraction model are

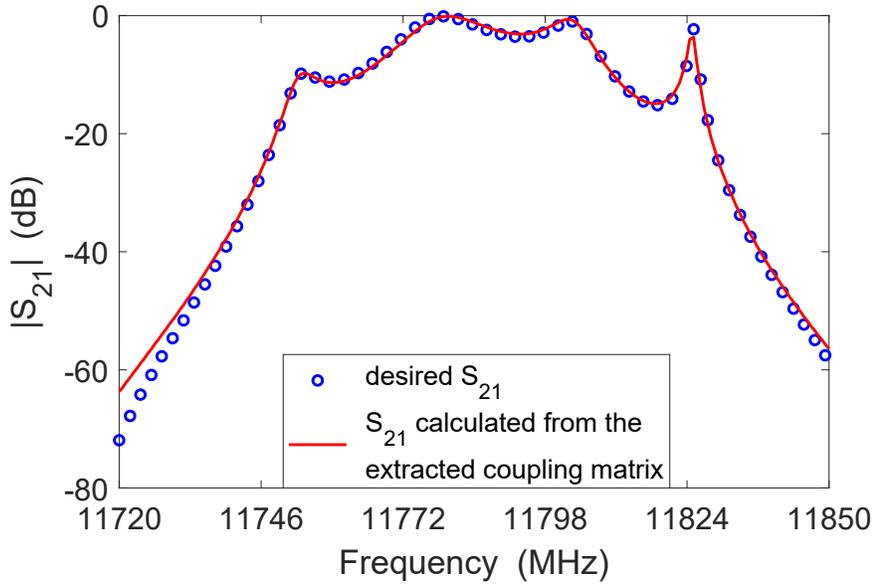
$$\mathbf{M}_{extracted} = \begin{bmatrix} 0.1940 & 0.6153 & 0 & 0 & 0 & 0 \\ 0.6153 & 0.1811 & 0.8747 & 0 & 0 & 0 \\ 0 & 0.8747 & -0.2501 & 0.7481 & 0 & -0.0183 \\ 0 & 0 & 0.7481 & -0.2589 & 0.4957 & 0 \\ 0 & 0 & 0 & 0.4957 & -0.2312 & 0.5299 \\ 0 & 0 & -0.0183 & 0 & 0.5299 & 0.2659 \end{bmatrix} \quad (3.29)$$

Fig. 3.11 illustrates the S -parameters from the desired and extracted coupling matrix for the highly detuned filter.

According to (3.28), (3.29), and Fig. 3.11, a good match between the desired and extracted coupling parameters along with an excellent match between the responses



(a)



(b)

Figure 3.11: The comparison between the desired S -parameters (o) and the S -parameters calculated from the extracted coupling matrix (-) for the highly detuned sixth-order filter. (a) Return loss S_{11} . (b) Insertion loss S_{21} .

from desired and extracted coupling matrices have been achieved for the highly detuned filter.

The test results show that the proposed deep neural network method can be used reliably to perform the parameter extraction for microwave filters. Unlike the conventional optimization method, our proposed deep neural network model does not need to simulate the filter circuit iteratively for each detuned filter. Once the deep neural network model is developed, it can be used to quickly extract the coupling parameters for both the slightly and the highly detuned filters as long as the detuned filters are in the range of the training data.

3.4 Conclusion

In this chapter, a hybrid deep neural network modeling technique has been proposed to solve complex high-dimensional modeling problems of microwave components. In the proposed technique, we have formulated a hybrid structure utilizing both sigmoid functions and ReLUs as activation functions for hidden neurons. A smooth ReLU has been proposed for the hybrid deep neural network model. Compared to the conventional ReLU, the proposed smooth ReLU can improve the smoothness of the model outputs and provide continuity of derivatives. The proposed deep neural network can be trained to learn the training data in high-dimensional space. An advanced three-stage deep learning algorithm has been proposed to train the new deep neural network and to overcome the vanishing gradient problem that would otherwise exist in training our hybrid deep neural networks. As illustrated in the examples, the proposed technique can solve higher dimensional microwave

modeling problems more efficiently over those solved by the shallow neural network. Compared to the pure ReLU deep neural network, the proposed technique can achieve higher accuracy with fewer number of hidden neurons.

Chapter 4

Advanced Cognition-Driven EM Optimization Incorporating Transfer Function-Based Feature Surrogate for Microwave Filters

4.1 Introduction

The deep neural network technique presented in the previous chapter is a useful surrogate modeling technique for microwave components. A further research topic is the surrogate-based electromagnetic (EM) optimization technique for microwave design. EM optimization plays an important role in the design of microwave components. EM-based optimizations are computationally expensive because repetitive EM simulations with different geometrical parameters as design variables are required in the optimization process. Surrogate-based optimization methods have been widely used to speed up the EM-based optimization. Among them, the cognition-driven EM optimization is an effective method to increase the optimization efficiency and the ability to avoid being trapped in local minima. How-

ever, there is a challenge that the existing cognition-driven EM optimization cannot proceed well when the filter response does not have explicitly identifiable feature information at the starting point or during the optimization.

In this chapter, we propose an advanced cognition-driven EM optimization method [99] incorporating transfer function-based feature surrogate for optimization of microwave filters. The proposed optimization method addresses the situations where the response at the starting point for the EM optimization is substantially misaligned with the design specifications and the filter response does not have explicitly identifiable feature information. We propose to extract multiple feature parameters, including passband feature frequencies, stopband feature frequencies, feature bandwidths, feature ripples, and boundary feature heights, from the transfer function response which is vector fitted based on the EM response (S-parameters). The proposed technique can extract these transfer function-based feature parameters effectively even for the cases where the features cannot be clearly and explicitly identified from the filter response. These extracted feature parameters are used to develop the feature surrogate model for the proposed cognition-driven optimization. Furthermore, we derive new objective functions for the cognition-driven optimization directly in the feature space. The proposed cognition-driven optimization incorporating transfer function-based feature surrogate can achieve faster convergence than the existing feature-assisted EM optimization methods.

4.2 Definition of the Optimization Problem

Let ϕ_F represent a vector of geometrical design variables. Let ω represent the angular frequency. Let \mathbf{S}_F represent the filter responses of the fine model (e.g., EM simulation), which is a function of design variables ϕ_F and frequency ω . The original EM optimization problem is defined as [48]

$$\phi_F^* = \arg \min_{\phi_F} U(\mathbf{S}_F(\phi_F, \omega)) \quad (4.1)$$

where U represents the given objective function, which can be a minimax or a generalized l_p function; ϕ_F^* represents the optimal design solution satisfying the design specifications.

4.3 Proposed Cognition-Driven EM Optimization Using Transfer Function-Based Feature Surrogate

The motivation of the cognition-driven technique is the cognitive design process used by experienced filter designers. The experienced designers would move the frequency locations of reflection zeros of the filter response to the passband, rather than trying to push the entire filter response directly toward the design specification. The original cognition-driven EM optimization [78] can effectively move the filter response to satisfy the specification when the filter response at the starting point for the design optimization is substantially misaligned with the design specifications. However, there is a challenge that the original cognition-driven EM optimization

cannot proceed well when the filter response does not have explicitly identifiable feature information at the starting point or during the optimization. This challenge limits the selection of the starting point and the effectiveness of the optimization process for the original cognition-driven optimization.

This chapter proposes an advanced cognition-driven EM optimization technique using transfer function-based feature surrogate. The proposed technique can extract the feature parameters even if the filter response does not have explicitly identifiable feature information. In the proposed technique, we obtain the transfer function response to represent the filter EM response. The feature parameters are extracted from the transfer function response instead of directly extracted from the filter response. These feature parameters are called the transfer function-based feature parameters. The transfer function-based feature parameters can be extracted effectively even if the feature information can not be explicitly identified from the filter response. The surrogate model developed using the transfer function-based feature parameters is used to perform the optimization. Here, we describe the proposed cognition-driven optimization technique in detail.

4.3.1 Transfer Function-Based Feature Parameter Extraction

Let k represent the index of the proposed surrogate optimization iteration. At the beginning of the k th optimization iteration, data generation for training the surrogate model is firstly performed. In the proposed technique, orthogonal distribution, i.e., a specific type of design of experiment (DOE) sampling distribution, is used for

generating multiple geometrical samples around the nominal point (i.e., the optimal solution of the surrogate model from the previous optimization iteration) [73]. Multiple geometrical samples of fine model EM responses are generated in parallel using multiple EM simulations simultaneously. As the orthogonal distribution is a very efficient sampling method, the number of EM simulations increases squarely as the number of design variables increases. Let N_g represent the total number of geometrical samples.

The vector fitting technique [165] is used to extract the poles and zeros of the transfer function from the filter EM response (e.g., S-parameters) of each geometrical sample. In this chapter, S_{11} parameter is used to represent the filter response. The poles and zeros for all the geometrical samples are obtained in complex values by using proper scaling and shifting for the frequency range during vector fitting process. Let N_e be the effective order of the transfer function for all the geometrical samples. Let $\mathbf{p}^{n,(k)}$ and $\mathbf{z}^{n,(k)}$ represent the vector of effective poles and zeros (i.e., poles and zeros with positive imaginary parts) obtained from the n th geometrical sample in the k th optimization iteration, respectively, where $\mathbf{p}^{n,(k)} = [p_1^{n,(k)} \ p_2^{n,(k)} \ \dots \ p_{N_e}^{n,(k)}]^T$, $\mathbf{z}^{n,(k)} = [z_1^{n,(k)} \ z_2^{n,(k)} \ \dots \ z_{N_e}^{n,(k)}]^T$, and $n = 1, 2, \dots, N_g$. The number of effective poles/zeros is equal to N_e .

In the proposed optimization method, multiple transfer function-based feature parameters are extracted from the transfer function response which is vector fitted based on the filter EM responses (S-parameters). We first introduce the passband feature frequencies to guide the move of the filter response to the desired passband. The passband feature frequencies are defined as the frequency locations of reflection

zeros inside the passband of the filter response. Let N_f represent the number of passband feature frequencies, which is equal to the order of the filter. To extract N_f passband feature frequencies, we select the N_f zeros which have relatively small real part values among all the effective zeros of the transfer function. The feature frequencies are closely related to the imaginary parts of the selected N_f zeros of the transfer function. Let $\hat{z}_i^{n,(k)}$ represent the i th selected zero of the transfer function for the n th geometrical sample in the k th optimization iteration. The selection of $\hat{z}_i^{n,(k)}$ is formulated as

$$\hat{z}_i^{n,(k)} = z_{\eta_i}^{n,(k)} \quad i = 1, 2, \dots, N_f \quad (4.2)$$

where the index η_i is defined by

$$\eta_i = \begin{cases} \arg \min_{\xi \in \{1, \dots, N_e\}} \left\| \operatorname{Re}(z_{\xi}^{n,(k)}) \right\|, & i = 1 \\ \arg \min_{\substack{\xi \in \{1, \dots, N_e\} \\ \xi \notin \{\eta_1, \dots, \eta_{i-1}\}}} \left\| \operatorname{Re}(z_{\xi}^{n,(k)}) \right\|, & i = 2, \dots, N_f \end{cases} \quad (4.3)$$

where $z_{\xi}^{n,(k)}$ represents the ξ th zero of the transfer function obtained using vector fitting technique for the n th geometrical sample in the k th optimization iteration. In order to establish the feature surrogate, the passband feature frequencies for each geometrical sample need to be in an ascending sequence based on the imaginary parts of the selected zeros of the transfer function. Let $f_i^{n,(k)}$ represent the i th passband feature frequency for the n th geometrical sample in the k th optimization

iteration. The formulation for extracting $f_i^{n,(k)}$ is derived as

$$f_i^{n,(k)} = \begin{cases} \min_{\xi \in \{1, \dots, N_f\}} \left\{ \hat{f}_\xi^{n,(k)} \right\}, & i = 1 \\ \min_{\substack{\xi \in \{1, \dots, N_f\} \\ \hat{f}_\xi^{n,(k)} \notin \{f_1^{n,(k)}, \dots, f_{i-1}^{n,(k)}\}}} \left\{ \hat{f}_\xi^{n,(k)} \right\}, & i = 2, \dots, N_f \end{cases} \quad (4.4)$$

where $\hat{f}_\xi^{n,(k)}$ is the imaginary part of the selected zero $\hat{z}_\xi^{n,(k)}$, i.e.,

$$\hat{f}_\xi^{n,(k)} = \text{Im}(\hat{z}_\xi^{n,(k)}), \xi = 1, \dots, N_f \quad (4.5)$$

Let $\mathbf{f}^{n,(k)}$ represent the vector containing all the passband feature frequencies for the n th geometrical sample in the k th optimization iteration, formulated as

$$\mathbf{f}^{n,(k)} = \left[f_1^{n,(k)} \quad f_2^{n,(k)} \quad \dots \quad f_{N_f}^{n,(k)} \right]^T \quad (4.6)$$

where $n = 1, 2, \dots, N_g$. The extracted passband feature frequencies are considered as the first set of transfer function-based feature parameters used to train the proposed feature surrogate for the optimization.

After the passband feature frequencies are identified, we extract the stopband feature frequencies. In order to extract the stopband feature frequencies, we first define the stopband specification as $\|\mathbf{S}_F(\phi_F, \omega)\| \geq h_s, \omega \leq f_{s1}$ and $\omega \geq f_{s2}$, where h_s represents the user pre-defined stopband response specification; f_{s1} and f_{s2} represent the lower and upper frequency boundaries of the stopband, respectively. Let $H^{n,(k)}$ represent the magnitude of the transfer function response for the n th geometrical sample in the k th optimization iteration. The stopband feature frequencies are

defined as the two frequencies at which the transfer function response $H^{n,(k)}$ first equal to h_s beyond the passband. The stopband feature frequencies can help optimize the filter response to satisfy stopband specifications. Since the effective poles $\mathbf{p}^{n,(k)}$ and zeros $\mathbf{z}^{n,(k)}$ of the transfer function for the n th geometrical sample in the k th optimization iteration have been obtained, the magnitude of the transfer function response $H^{n,(k)}$ is formulated as [80]

$$H^{n,(k)}(\omega) = G^{n,(k)} \left\| \frac{\prod_{i=1}^{N_e} (j\omega - z_i^{n,(k)})(j\omega - z_i^{n,(k)*})}{\prod_{i=1}^{N_e} (j\omega - p_i^{n,(k)})(j\omega - p_i^{n,(k)*})} \right\| \quad (4.7)$$

where ω represents the angular frequency; $G^{n,(k)}$ represents the gain factor of the transfer function extracted using vector fitting for the n th geometrical sample in the k th optimization iteration; $z_i^{n,(k)}$ and $p_i^{n,(k)}$ represent the i th effective zeros and poles for the n th geometrical sample in the k th optimization iteration, respectively; $z_i^{n,(k)*}$ and $p_i^{n,(k)*}$ are the conjugate pairs of $z_i^{n,(k)}$ and $p_i^{n,(k)}$, respectively.

Let $\mathbf{s}^{n,(k)}$ represent the vector containing stopband feature frequencies for the n th geometrical sample in the k th optimization iteration. The calculation of stopband feature frequencies $\mathbf{s}^{n,(k)}$ is derived as

$$\mathbf{s}^{n,(k)} = \begin{bmatrix} s_1^{n,(k)} & s_2^{n,(k)} \end{bmatrix}^T \quad (4.8)$$

where $s_1^{n,(k)}$ and $s_2^{n,(k)}$ represent the lower stopband feature frequency and the upper stopband feature frequency, respectively, defined by

$$s_1^{n,(k)} = \arg \min_{\substack{\omega \in B_s \\ \omega \leq f_1^{n,(k)}}} \left\| \omega - f_1^{n,(k)} \right\| \quad (4.9)$$

$$s_2^{n,(k)} = \arg \min_{\substack{\omega \in B_s \\ \omega \geq f_{N_f}^{n,(k)}}} \left\| \omega - f_{N_f}^{n,(k)} \right\| \quad (4.10)$$

and B_s represents a set containing all the frequencies at which the transfer function response equal to the stopband response specification h_s , i.e.,

$$B_s = \{ \omega \mid H^{n,(k)}(\omega) - h_s = 0 \} \quad (4.11)$$

The obtained stopband feature frequencies are considered as the second set of transfer function-based feature parameters used to train the proposed feature surrogate.

Besides the passband and stopband feature frequencies, we propose to extract the third set of transfer function-based feature parameters, which are named as feature bandwidths. The feature bandwidths are defined as the distances between two successive passband feature frequencies. Let $b_i^{n,(k)}$ represent the i th feature bandwidth for the n th geometrical sample in the k th optimization iteration. The calculation of $b_i^{n,(k)}$ is derived as

$$b_i^{n,(k)} = f_{i+1}^{n,(k)} - f_i^{n,(k)}, \quad i = 1, 2, \dots, N_f - 1 \quad (4.12)$$

Let $\mathbf{b}^{n,(k)}$ represent the vector containing all the feature bandwidths, formulated as

$$\mathbf{b}^{n,(k)} = \left[b_1^{n,(k)} \quad b_2^{n,(k)} \quad \dots \quad b_{N_f-1}^{n,(k)} \right]^T \quad (4.13)$$

The sum of all the feature bandwidths $b_i^{n,(k)}$ approximately equals the entire passband bandwidth for the n th geometrical sample in the k th optimization iteration. The feature bandwidths are introduced to limit the distances between two successive passband feature frequencies during the optimization. This limitation can

enforce all the passband feature frequencies moving toward the design specification at similar speed, increasing the robustness of the overall optimization.

The three sets of the feature parameters defined above focus on moving the feature frequencies into the design specification. Simultaneously with the movement of the feature frequencies, the filter response also needs to approach the design specification. Therefore, the fourth set of feature parameters for the transfer function-based feature optimization, named as feature ripples, are introduced. The feature ripples are defined as the maximum magnitude of the transfer function responses located between two successive passband feature frequencies. Let $r_i^{n,(k)}$ represent the i th feature ripple for the n th geometrical sample in the k th optimization iteration. The number of feature ripples is one less than the number of the passband feature frequencies, i.e., $N_f - 1$. The calculation for $r_i^{n,(k)}$ is derived as

$$r_i^{n,(k)} = \max_{f_i^{n,(k)} \leq \omega \leq f_{i+1}^{n,(k)}} H^{n,(k)}(\omega), \quad i = 1, 2, \dots, N_f - 1 \quad (4.14)$$

where $H^{n,(k)}(\omega)$ is calculated using the transfer function formulated in (4.7). Let $\mathbf{r}^{n,(k)}$ represent the vector containing all the feature ripples for the n th geometrical sample in the k th optimization iteration. The vector $\mathbf{r}^{n,(k)}$ is formulated as

$$\mathbf{r}^{n,(k)} = \left[r_1^{n,(k)} \quad r_2^{n,(k)} \quad \dots \quad r_{N_f-1}^{n,(k)} \right]^T \quad (4.15)$$

The feature ripples are used to guide the filter response to satisfy the passband response specification.

Along with the feature ripples, another set of feature parameters are also needed to guarantee the filter response to satisfy the passband response specification. This

set of feature parameters are named as boundary feature heights. To define the boundary feature heights, we first define the passband specification as $\|\mathbf{S}_F(\boldsymbol{\phi}_F, \omega)\| \leq h_p, f_{p1} \leq \omega \leq f_{p2}$, where h_p represents the passband response specification; f_{p1} and f_{p2} represent the lower and upper frequency boundaries of the passband, respectively. Let $h_1^{n,(k)}$ and $h_2^{n,(k)}$ represent the lower and upper boundary feature heights, respectively, which are defined as the magnitude of the transfer function response $H^{n,(k)}(\omega)$ at the lower and upper frequency boundaries of the passband, i.e.,

$$h_1^{n,(k)} = H^{n,(k)}(f_{p1}) \quad (4.16)$$

$$h_2^{n,(k)} = H^{n,(k)}(f_{p2}) \quad (4.17)$$

Let $\mathbf{h}^{n,(k)}$ represent the vector containing the boundary feature heights for the n th geometrical sample in the k th optimization iteration, formulated as

$$\mathbf{h}^{n,(k)} = \begin{bmatrix} h_1^{n,(k)} & h_2^{n,(k)} \end{bmatrix}^T \quad (4.18)$$

The boundary feature heights are used as the fifth set of transfer function-based feature parameters to train the feature surrogate for the proposed optimization.

For visually illustrating the five sets of feature parameters, we use an example of a four-pole filter response, as shown in Fig. 4.1. In Fig. 4.1, f_1, f_2, f_3 , and f_4 are the passband feature frequencies; s_1 and s_2 are the stopband feature frequencies; b_1, b_2 , and b_3 are the feature bandwidths; r_1, r_2 , and r_3 are the feature ripples; h_1 and h_2 are the boundary feature heights. Feature frequencies f_1 and f_2 are very close to each other because the imaginary parts of the first and second selected zeros are very close to each other. Note that, the filter passband and the passband

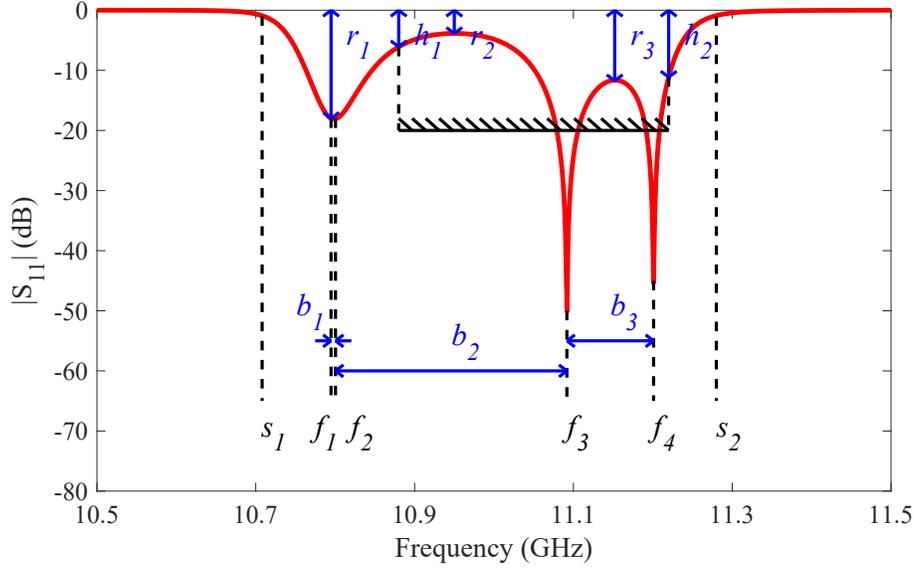


Figure 4.1: Illustration of a four-pole filter response. f_1 , f_2 , f_3 , and f_4 are the passband feature frequencies; s_1 and s_2 are the stopband feature frequencies; b_1 , b_2 , and b_3 are the feature bandwidths; r_1 , r_2 , and r_3 are the feature ripples; h_1 and h_2 are the boundary feature heights.

specification have different definitions. As shown in Fig. 4.1, the filter passband is the passband covering the feature passband frequencies (i.e., f_1 , f_2 , f_3 , and f_4) of the present filter response, while the passband specification (i.e., the shadowed line) is the design specification which is pre-defined by the user. From Fig. 4.1, we can see that four passband feature frequencies are all inside the filter passband, two of which (f_1 and f_2) are outside the passband specification, while the other two (f_3 and f_4) are inside the passband specification.

Since the feature ripples and boundary feature heights vary much as the geometrical parameters change, it is hard to train the surrogate directly using the data of the feature ripples and boundary feature heights. We propose to rectify these

two sets of feature parameters using sigmoid functions. Let $\widehat{\mathbf{r}}^{n,(k)}$ and $\widehat{\mathbf{h}}^{n,(k)}$ represent the vectors containing the rectified feature ripples and the rectified boundary feature heights for the n th geometrical sample in the k th optimization iteration, respectively, formulated as

$$\widehat{\mathbf{r}}^{n,(k)} = \left[\widehat{r}_1^{n,(k)} \quad \widehat{r}_2^{n,(k)} \quad \dots \quad \widehat{r}_{N_f-1}^{n,(k)} \right]^T \quad (4.19)$$

$$\widehat{\mathbf{h}}^{n,(k)} = \left[\widehat{h}_1^{n,(k)} \quad \widehat{h}_2^{n,(k)} \right]^T \quad (4.20)$$

where

$$\widehat{r}_i^{n,(k)} = \frac{2}{1 + e^{\frac{r_i^{n,(k)} - h_p}{\varphi}}}, \quad i = 1, 2, \dots, N_f - 1 \quad (4.21)$$

$$\widehat{h}_i^{n,(k)} = \frac{2}{1 + e^{\frac{h_i^{n,(k)} - h_p}{\varphi}}}, \quad i = 1, 2 \quad (4.22)$$

where φ represents the weight for $r_i^{n,(k)}$ and $h_i^{n,(k)}$ when performing the rectification. Fig. 4.2 shows an illustration of the rectification of the feature ripples and the boundary feature heights using (4.21) and (4.22) with $h_p = 0.1$ and $\varphi = 0.1$. As shown in Fig. 4.2, the rectification magnifies the resolution of the two sets of feature parameters close to the passband response specification $h_p = 0.1$ and reduces the resolution of those feature parameters far beyond $h_p = 0.1$. Smaller value for the weight φ makes $\widehat{\mathbf{r}}^{n,(k)}$ and $\widehat{\mathbf{h}}^{n,(k)}$ have higher resolution close to h_p and lower resolution far beyond h_p . In this way, we can significantly increase the influence of the feature parameters close to the passband response specification, resulting in the improvement of the surrogate accuracy close to the passband response specification. A more accurate surrogate close to the specification can result in a better optimiza-

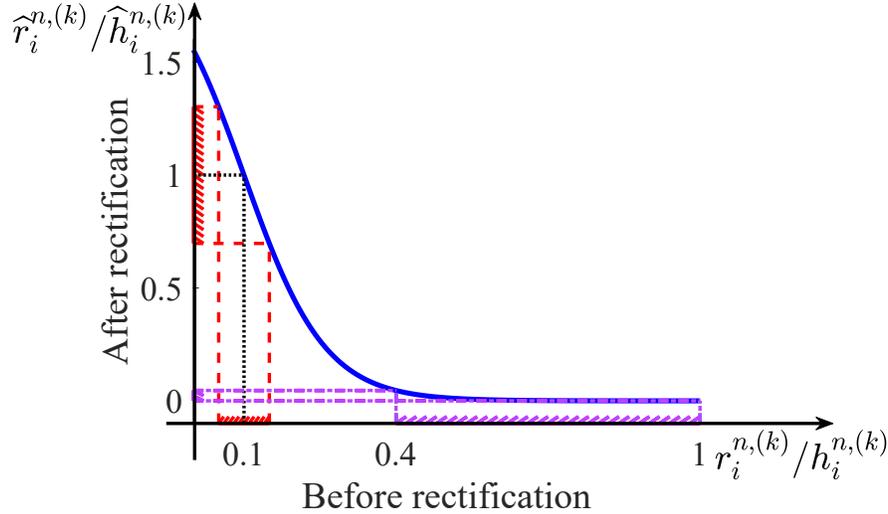


Figure 4.2: Illustration of the rectification of the feature ripples and the boundary feature heights using equations (4.21) and (4.22) with $h_p = 0.1$ and $\varphi = 0.1$.

tion convergence. The mechanism of the proposed parameter extraction for transfer function-based feature parameters is illustrated in Fig. 4.3.

4.3.2 Cognition-Driven Optimization Using Transfer Function-Based Feature Surrogate

In this subsection, we describe how to perform the proposed cognition-driven optimization using transfer function-based feature surrogate. Firstly, we define $\Omega_s^{(k)}$ to represent the trust region of geometrical parameters of the surrogate model in the k th iteration [73],

$$\Omega_s^{(k)} = \left\{ \phi_F \mid (\mathbf{I} - \text{diag}(\mathbf{P}^{(k)}))\phi_F^{(k)} \leq \phi_F \leq (\mathbf{I} + \text{diag}(\mathbf{P}^{(k)}))\phi_F^{(k)} \right\} \quad (4.23)$$

where \mathbf{I} represents the identity matrix. $\mathbf{P}^{(k)}$ is defined as the trust radius which is a vector consisting of the ranges of all the design variables in the k th iteration.

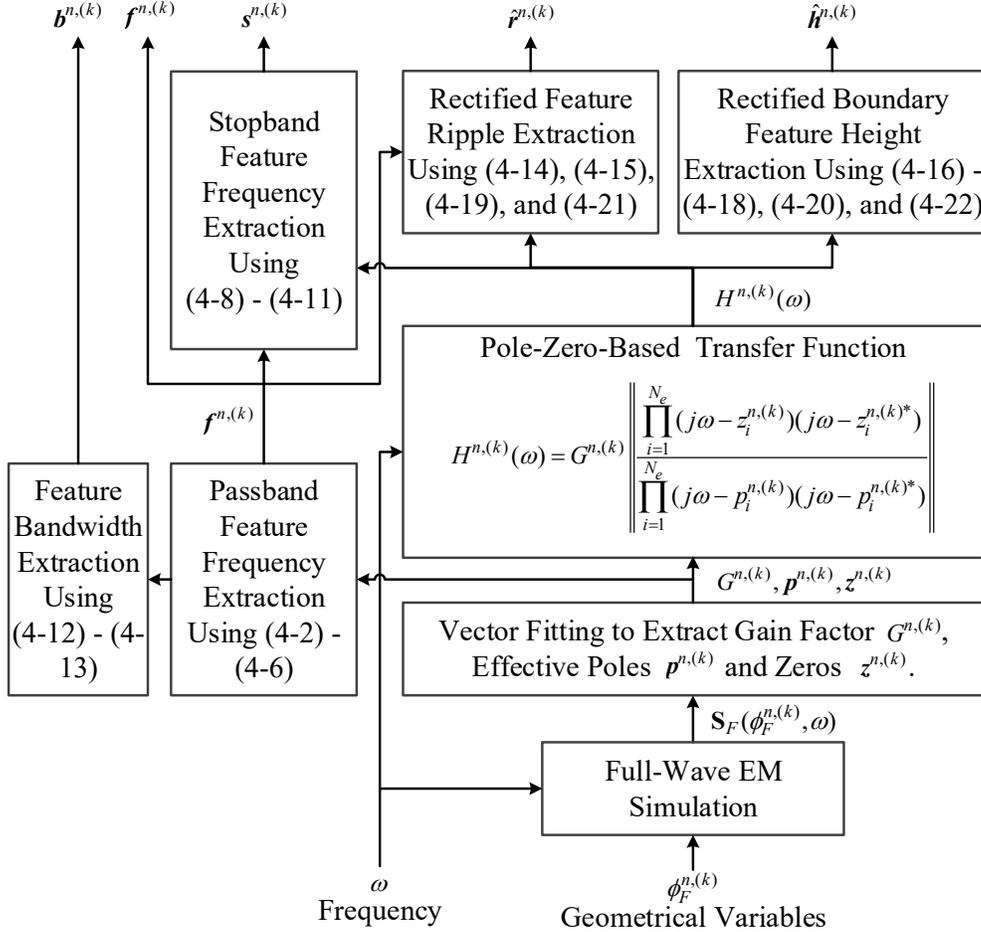


Figure 4.3: The mechanism of the proposed parameter extraction for transfer function-based feature parameters. The extracted five sets of transfer function-based feature parameters are passband feature frequencies $f^{n,(k)}$, stopband feature frequencies $s^{n,(k)}$, feature bandwidths $b^{n,(k)}$, rectified feature ripples $\hat{r}^{n,(k)}$, and rectified boundary feature heights $\hat{h}^{n,(k)}$.

$\mathbf{P}^{(k)}$ contains percentage values, representing the percentage of the design variable variations over the nominal point $\phi_F^{(k)}$. The surrogate model is developed within the region of $\Omega_s^{(k)}$. We generate multiple EM data $S_F(\phi_F^{n,(k)}, \omega)$, $n = 1, 2, \dots, N_g$,

simultaneously at multiple geometrical samples in parallel [73],

$$\begin{aligned} & \left\{ \mathbf{S}_F(\boldsymbol{\phi}_F^{n,(k)}, \omega) \mid n = 1, 2, \dots, N_g \right\} \\ & = \left\{ \mathbf{S}_F(\boldsymbol{\phi}_F^{1,(k)}, \omega), \mathbf{S}_F(\boldsymbol{\phi}_F^{2,(k)}, \omega), \dots, \mathbf{S}_F(\boldsymbol{\phi}_F^{N_g,(k)}, \omega) \right\} \end{aligned} \quad (4.24)$$

where the first geometrical sample $\boldsymbol{\phi}_F^{1,(k)}$ is equal to the nominal point $\boldsymbol{\phi}_F^{(k)}$, i.e., $\boldsymbol{\phi}_F^{1,(k)} = \boldsymbol{\phi}_F^{(k)}$, and the remaining geometrical samples $\boldsymbol{\phi}_F^{n,(k)}$, $n = 2, \dots, N_g$, are distributed around the nominal point $\boldsymbol{\phi}_F^{(k)}$ following orthogonal distribution. Note that $\boldsymbol{\phi}_F^{n,(k)} = [\phi_1^{n,(k)}, \phi_2^{n,(k)}, \dots, \phi_{N_v}^{n,(k)}]^T$, where N_v is the number of design variables. After data generation, we establish the proposed transfer function-based feature surrogate model. We perform the parameter extraction using (4.2)-(4.22) to extract the five sets of transfer function-based feature parameters $\mathbf{f}^{n,(k)}$, $\mathbf{s}^{n,(k)}$, $\mathbf{b}^{n,(k)}$, $\widehat{\mathbf{r}}^{n,(k)}$, and $\widehat{\mathbf{h}}^{n,(k)}$ from $\mathbf{S}_F(\boldsymbol{\phi}_F^{n,(k)}, \omega)$ for all the geometrical samples, $n = 1, 2, \dots, N_g$.

Let \mathbf{Y} represent the output vector of the transfer function-based feature surrogate model. The surrogate output \mathbf{Y} consists of \mathbf{Y}^f (passband feature frequencies), \mathbf{Y}^s (stopband feature frequencies), \mathbf{Y}^b (feature bandwidths), \mathbf{Y}^r (rectified feature ripples), and \mathbf{Y}^h (rectified boundary feature heights), i.e.,

$$\mathbf{Y} = \left[\mathbf{Y}^{fT} \quad \mathbf{Y}^{sT} \quad \mathbf{Y}^{bT} \quad \mathbf{Y}^{rT} \quad \mathbf{Y}^{hT} \right]^T \quad (4.25)$$

The data of $(\boldsymbol{\phi}_F^{n,(k)}, \mathbf{f}^{n,(k)})$, $(\boldsymbol{\phi}_F^{n,(k)}, \mathbf{s}^{n,(k)})$, $(\boldsymbol{\phi}_F^{n,(k)}, \mathbf{b}^{n,(k)})$, $(\boldsymbol{\phi}_F^{n,(k)}, \widehat{\mathbf{r}}^{n,(k)})$, and $(\boldsymbol{\phi}_F^{n,(k)}, \widehat{\mathbf{h}}^{n,(k)})$ are used to train \mathbf{Y}^f , \mathbf{Y}^s , \mathbf{Y}^b , \mathbf{Y}^r , and \mathbf{Y}^h , respectively. Since the relationship between the transfer function-based feature parameters and the geometrical parameters has relatively low nonlinearity, we propose to use the quadratic function as the function of the surrogate model to learn this relationship. Let $\mathbf{w}^{(k)}$ represent

the optimal values of weighting parameters in the surrogate model at the k th optimization iteration. The number of weighting parameters for the quadratic function is dependent on the number of design variables. The number of design variables is denoted as N_v . The number of weighting parameters is then calculated to be $(N_v + 1)(N_v + 2)/2$. Let $\mathbf{Y}_{dat}^{n,(k)}$ represent the vector containing the data of the five sets of transfer function-based feature parameters for the n th geometrical sample in the k th optimization iteration, formulated as

$$\mathbf{Y}_{dat}^{n,(k)} = \left[\mathbf{f}^{n,(k)T} \quad \mathbf{s}^{n,(k)T} \quad \mathbf{b}^{n,(k)T} \quad \hat{\mathbf{r}}^{n,(k)T} \quad \hat{\mathbf{h}}^{n,(k)T} \right]^T \quad (4.26)$$

We derive the calculation of $\mathbf{w}^{(k)}$ by solving linear equations in (4.27).

$$\begin{bmatrix} \mathbf{Y}_{dat}^{1,(k)T} \\ \mathbf{Y}_{dat}^{2,(k)T} \\ \vdots \\ \mathbf{Y}_{dat}^{n,(k)T} \\ \vdots \\ \mathbf{Y}_{dat}^{N_g,(k)T} \end{bmatrix} = \begin{bmatrix} 1 & \phi_1^{1,(k)} & \phi_2^{1,(k)} & \dots & \phi_{N_v}^{1,(k)} & \left(\phi_1^{1,(k)}\right)^2 & \phi_1^{1,(k)}\phi_2^{1,(k)} & \dots & \left(\phi_{N_v}^{1,(k)}\right)^2 \\ 1 & \phi_1^{2,(k)} & \phi_2^{2,(k)} & \dots & \phi_{N_v}^{2,(k)} & \left(\phi_1^{2,(k)}\right)^2 & \phi_1^{2,(k)}\phi_2^{2,(k)} & \dots & \left(\phi_{N_v}^{2,(k)}\right)^2 \\ \vdots & \vdots \\ 1 & \phi_1^{n,(k)} & \phi_2^{n,(k)} & \dots & \phi_{N_v}^{n,(k)} & \left(\phi_1^{n,(k)}\right)^2 & \phi_1^{n,(k)}\phi_2^{n,(k)} & \dots & \left(\phi_{N_v}^{n,(k)}\right)^2 \\ \vdots & \vdots \\ 1 & \phi_1^{N_g,(k)} & \phi_2^{N_g,(k)} & \dots & \phi_{N_v}^{N_g,(k)} & \left(\phi_1^{N_g,(k)}\right)^2 & \phi_1^{N_g,(k)}\phi_2^{N_g,(k)} & \dots & \left(\phi_{N_v}^{N_g,(k)}\right)^2 \end{bmatrix} \mathbf{w}^{(k)T} \quad (4.27)$$

Note that, the geometrical samples for generating EM data include one nominal point and N_{or}^2 orthogonal geometrical samples, where N_{or} represents the level of the orthogonal distribution. The number of orthogonal geometrical samples should

satisfy two criteria. One is that the number of orthogonal geometrical samples should be equal to or greater than the number of weighting parameters, i.e., $N_{or}^2 \geq (N_v + 1)(N_v + 2)/2$, in order to accurately solve (4.27) to obtain the optimal $\mathbf{w}^{(k)}$. The other is that the level N_{or} should be equal to or greater than $N_v - 1$, i.e., $N_{or} \geq N_v - 1$, in order to obtain enough orthogonal geometrical samples. Besides, N_{or} is better to be an odd number to make the samples orthogonal to each other [166]. Therefore, N_{or} is determined to be the minimum odd number that satisfies $N_{or} \geq \max\{\sqrt{(N_v + 1)(N_v + 2)/2}, N_v - 1\}$. Once N_{or} is determined, the number of geometrical samples can be subsequently calculated using $N_g = N_{or}^2 + 1$.

After the surrogate model development, we perform the optimization. We propose to derive new objective functions for different sets of surrogate outputs. Let U_f represent the objective function for \mathbf{Y}^f . To optimize \mathbf{Y}^f , we first divide the passband specification into N_f sub-bands. The objective function U_f is to move the i th passband feature frequency Y_i^f of the surrogate model into the i th sub-band. The objective function U_f is formulated as

$$U_f(\mathbf{Y}^f(\phi_F, \mathbf{w}^{(k)})) = \sum_{i=1}^{N_f} \left\| e_i^f(\phi_F, \mathbf{w}^{(k)}) \right\|^2 \quad (4.28)$$

where

$$e_i^f(\phi_F, \mathbf{w}^{(k)}) = \begin{cases} f_i^{low} - Y_i^f(\phi_F, \mathbf{w}^{(k)}), & Y_i^f \leq f_i^{low} \\ 0, & f_i^{low} < Y_i^f \leq f_i^{upp} \\ Y_i^f(\phi_F, \mathbf{w}^{(k)}) - f_i^{upp}, & Y_i^f > f_i^{upp} \end{cases} \quad (4.29)$$

and

$$f_i^{low} = \frac{(i-1)(f_{p2} - f_{p1})}{N_f} + f_{p1} \quad (4.30)$$

$$f_i^{upp} = \frac{i \cdot (f_{p2} - f_{p1})}{N_f} + f_{p1} \quad (4.31)$$

Let U_s be defined as the objective function for stopband feature frequencies. The objective for the stopband is to make the magnitude of filter response larger than h_s when the frequencies are inside the stopband. This objective can be converted to make the stopband feature frequencies locate beyond the stopband boundaries. Since the stopband feature frequencies should also locate beyond the passband boundaries, the objective function U_s is formulated as

$$U_s(\mathbf{Y}^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})) = \|e_1^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})\|^2 + \|e_2^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})\|^2 \quad (4.32)$$

where

$$e_1^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) = \begin{cases} f_{s1} - Y_1^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}), & Y_1^s \leq f_{s1} \\ 0, & f_{s1} < Y_1^s \leq f_{p1} \\ Y_1^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) - f_{p1}, & Y_1^s > f_{p1} \end{cases} \quad (4.33)$$

$$e_2^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) = \begin{cases} f_{p2} - Y_2^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}), & Y_2^s \leq f_{p2} \\ 0, & f_{p2} < Y_2^s \leq f_{s2} \\ Y_2^s(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) - f_{s2}, & Y_2^s > f_{s2} \end{cases} \quad (4.34)$$

Let U_b be defined as the objective function for feature bandwidths. The objective

function U_b for the feature bandwidth uses a relax specification, which limits the value of feature bandwidth to be larger than the total passband bandwidth divided by N_f (i.e., the number of passband feature frequencies) and smaller than the total passband bandwidth divided by $N_f - 1$. The objective function U_b is formulated as

$$U_b(\mathbf{Y}^b(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})) = \sum_{i=1}^{N_f-1} \|e_i^b(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})\|^2 \quad (4.35)$$

where

$$e_i^b(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) = \begin{cases} \frac{f_{p2}-f_{p1}}{N_f} - Y_i^b(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}), & Y_i^b \leq \frac{f_{p2}-f_{p1}}{N_f} \\ Y_i^b(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) - \frac{f_{p2}-f_{p1}}{N_f-1}, & Y_i^b > \frac{f_{p2}-f_{p1}}{N_f-1} \\ 0, & \text{otherwise} \end{cases} \quad (4.36)$$

Let U_r represent the objective function for feature ripples. Since $\mathbf{Y}^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})$ is trained using the rectified data $(\boldsymbol{\phi}_F^{n,(k)}, \hat{\mathbf{r}}^{n,(k)})$, the passband response specification $\|\mathbf{S}_F(\boldsymbol{\phi}_F, \omega)\| \leq h_p$ is converted to $Y_i^r \geq 1, i = 1, 2, \dots, N_f - 1$, based on (4.21). The objective function U_r is formulated as

$$U_r(\mathbf{Y}^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})) = \begin{cases} U_r^+(\mathbf{Y}^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})), & U_r^+ > 0 \\ U_r^-(\mathbf{Y}^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})), & U_r^+ \leq 0 \end{cases} \quad (4.37)$$

where

$$U_r^+(\mathbf{Y}^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})) = \sum_{i=1}^{N_f-1} (\max\{1 - Y_i^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}), 0\})^2 \quad (4.38)$$

$$U_r^-(\mathbf{Y}^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})) = - \sum_{i=1}^{N_f-1} (Y_i^r(\boldsymbol{\phi}_F, \mathbf{w}^{(k)}) - 1)^{-2} \quad (4.39)$$

Let U_h be defined as the objective function for boundary feature heights. Based on (4.22), the specification for the boundary feature heights is $Y_i^h \geq 1$, $i = 1, 2$. The objective function U_h is derived similarly to U_r , formulated as

$$U_h(\mathbf{Y}^h(\phi_F, \mathbf{w}^{(k)})) = \begin{cases} U_h^+(\mathbf{Y}^h(\phi_F, \mathbf{w}^{(k)})), & U_h^+ > 0 \\ U_h^-(\mathbf{Y}^h(\phi_F, \mathbf{w}^{(k)})), & U_h^+ \leq 0 \end{cases} \quad (4.40)$$

where

$$U_h^+(\mathbf{Y}^h(\phi_F, \mathbf{w}^{(k)})) = \sum_{i=1}^2 (\max\{1 - Y_i^h(\phi_F, \mathbf{w}^{(k)}), 0\})^2 \quad (4.41)$$

$$U_h^-(\mathbf{Y}^h(\phi_F, \mathbf{w}^{(k)})) = - \sum_{i=1}^2 (Y_i^h(\phi_F, \mathbf{w}^{(k)}) - 1)^{-2} \quad (4.42)$$

Let \tilde{U} be defined as the overall objective function for the transfer function-based feature surrogate model. The overall objective function \tilde{U} is formulated as the weighted sum of all the objective functions for different sets of feature parameters, i.e.,

$$\begin{aligned} \tilde{U}(\mathbf{Y}(\phi_F, \mathbf{w}^{(k)})) = & U_f(\mathbf{Y}^f(\phi_F, \mathbf{w}^{(k)})) + U_s(\mathbf{Y}^s(\phi_F, \mathbf{w}^{(k)})) + \Upsilon_1 U_b(\mathbf{Y}^b(\phi_F, \mathbf{w}^{(k)})) \\ & + \Upsilon_2 U_r(\mathbf{Y}^r(\phi_F, \mathbf{w}^{(k)})) + \Upsilon_3 U_h(\mathbf{Y}^h(\phi_F, \mathbf{w}^{(k)})) \end{aligned} \quad (4.43)$$

where

$$\Upsilon_1 = \frac{U_f(\mathbf{Y}^f(\phi_F^{(k)}, \mathbf{w}^{(k)}))}{U_f(\mathbf{Y}^f(\phi_F^{(1)}, \mathbf{w}^{(1)}))} \quad (4.44)$$

$$\Upsilon_2 = \frac{\Psi U_f(\mathbf{Y}^f(\phi_F^{(1)}, \mathbf{w}^{(1)}))}{(\Psi^2 - 1) U_f(\mathbf{Y}^f(\phi_F^{(k)}, \mathbf{w}^{(k)})) + U_f(\mathbf{Y}^f(\phi_F^{(1)}, \mathbf{w}^{(1)}))} \quad (4.45)$$

$$\Upsilon_3 = \Upsilon_2 \prod_{i=1}^{N_f} \frac{1}{2} \left(\frac{Y_i^f(\boldsymbol{\phi}_F^{(k)}, \mathbf{w}^{(k)}) - f_{p1}}{|Y_i^f(\boldsymbol{\phi}_F^{(k)}, \mathbf{w}^{(k)}) - f_{p1}|} \cdot \frac{f_{p2} - Y_i^f(\boldsymbol{\phi}_F^{(k)}, \mathbf{w}^{(k)})}{|f_{p2} - Y_i^f(\boldsymbol{\phi}_F^{(k)}, \mathbf{w}^{(k)})|} + 1 \right) \quad (4.46)$$

where Ψ is a constant (normally should be larger than 1), which is predetermined before the optimization (e.g., $\Psi = 10$). The value of Ψ determines the increasing rate of Υ_2 during the proposed optimization. The weighting parameter Υ_1 equals one at the beginning of the proposed optimization. This activates U_b to help enforce all the passband feature frequencies moving toward the specification at a similar speed when the passband feature frequencies are far beyond the specification. The weighting parameter Υ_1 for U_b decreases as the passband feature frequencies approach the design specification. When all the passband feature frequencies satisfy the specification, the weighting parameter Υ_1 automatically becomes zero. After that, the objective function U_b no longer has a contribution to the overall objective function. The restriction of the feature bandwidths due to U_b is automatically removed after all the passband feature frequencies are inside the passband specification. The weighting parameter Υ_2 equals to $1/\Psi$ at the beginning of the proposed optimization. As the passband feature frequencies move approaching the specifications, the weighting parameter Υ_2 automatically increases, increasing the focus on the optimization of U_r . The weighting parameter Υ_3 equals to zero at the beginning of the optimization when the passband feature frequencies are beyond the passband. When all the passband feature frequencies have already located inside the passband, the weighting parameter Υ_3 automatically equals to Υ_2 , allowing the same focus on both U_r and U_h to fine optimizing the filter response to satisfy the response specification.

The proposed cognition-driven optimization using the transfer function-based feature surrogate is formulated as,

$$\boldsymbol{\phi}_F^{(k+1)} = \arg \min_{\boldsymbol{\phi}_F \in \boldsymbol{\Omega}_s^{(k)}} \tilde{U}(\mathbf{Y}(\boldsymbol{\phi}_F, \mathbf{w}^{(k)})) \quad (4.47)$$

where k is defined as the index of optimization iterations, initialized to one at the beginning; $\boldsymbol{\phi}_F^{(k+1)}$ is the optimal surrogate solution in the k th optimization iteration.

Here we propose to develop a trust region algorithm specifically for the advanced cognition-driven optimization to control the surrogate updates, thus ensuring the convergence of the optimization [78]. The initial value of each element $P_i^{(1)}$ in the trust radius $\mathbf{P}^{(1)}$ is determined based on the sensitivity of EM response with respect to the corresponding design variable. Then we derive the trust region algorithm to update the trust radius $\mathbf{P}^{(k)}$ during the optimization. Since the first geometrical sample $\boldsymbol{\phi}_F^{1,(k)}$ is equal to the nominal point $\boldsymbol{\phi}_F^{(k)}$ (i.e., the optimal solution from the previous iteration), the passband feature frequencies, stopband feature frequencies, feature bandwidths, rectified feature ripples, and rectified boundary feature heights extracted from the EM response at $\boldsymbol{\phi}_F^{(k)}$ are $\mathbf{f}^{1,(k)}$, $\mathbf{s}^{1,(k)}$, $\mathbf{b}^{1,(k)}$, $\hat{\mathbf{r}}^{1,(k)}$, and $\hat{\mathbf{h}}^{1,(k)}$, respectively. The overall objective function for the fine model EM response at $\boldsymbol{\phi}_F^{(k)}$ can be subsequently denoted as $\tilde{U}(\mathbf{Y}_{dat}^{1,(k)})$, which is formulated as the weighted sum of all the objective functions for different sets of feature parameters extracted from the EM response, i.e.,

$$\tilde{U}(\mathbf{Y}_{dat}^{1,(k)}) = U_f(\mathbf{f}^{1,(k)}) + U_s(\mathbf{s}^{1,(k)}) + \Upsilon_1 U_b(\mathbf{b}^{1,(k)}) + \Upsilon_2 U_r(\hat{\mathbf{r}}^{1,(k)}) + \Upsilon_3 U_h(\hat{\mathbf{h}}^{1,(k)}) \quad (4.48)$$

Similarly, $\tilde{U}(\mathbf{Y}_{dat}^{1,(k+1)})$ denotes the overall objective function for fine model EM responses at $\phi_F^{(k+1)}$. An adjustment control index parameter χ is calculated to represent the ratio of the actual reduction and the predicted reduction in the value of the proposed objective functions. The formulation of χ is derived as [73]

$$\chi = \frac{\tilde{U}(\mathbf{Y}_{dat}^{1,(k)}) - \tilde{U}(\mathbf{Y}_{dat}^{1,(k+1)})}{\tilde{U}(\mathbf{Y}(\phi_F^{(k)}, \mathbf{w}^{(k)})) - \tilde{U}(\mathbf{Y}(\phi_F^{(k+1)}, \mathbf{w}^{(k)}))} \quad (4.49)$$

The trust radius $\mathbf{P}^{(k+1)}$ is then updated based on the control index parameter χ , calculated as [81]

$$\mathbf{P}^{(k+1)} = \begin{cases} \varsigma_e \mathbf{P}^{(k)}, & \chi > 0.8, \\ \mathbf{P}^{(k)}, & 0.1 \leq \chi \leq 0.8, \\ \varsigma_c \mathbf{P}^{(k)}, & \chi < 0.1, \end{cases} \quad (4.50)$$

where ς_e and ς_c represent the coefficients for the trust radius update. Expansion and contraction of the trust radius depends on the values of ς_e and ς_c in (4.50), respectively. In this thesis, we use $\varsigma_e = 1.3$ and $\varsigma_c = 0.7$.

We iteratively perform EM data generation, feature parameter extraction, surrogate model training, surrogate design optimization, and trust region update. The proposed optimization terminates if the normalized absolute difference between the optimal surrogate solutions in successive iterations is sufficiently small, i.e.,

$$\frac{\|\phi_F^{(k+1)} - \phi_F^{(k)}\|}{\|\phi_F^{(k)}\|} \leq \varepsilon \quad (4.51)$$

where ε is a user-defined threshold (e.g., 10^{-4}). To perform the proposed optimiza-

tion, we first set initial starting point $\phi_F^{(k)}$ and initial trust region at $k = 1$. Using the proposed technique, we start with $\Upsilon_1 = 1$, $\Upsilon_2 = 1/\Psi$, and $\Upsilon_3 = 0$ which allows the optimization to focus on moving the feature frequencies and optimizing the feature ripples simultaneously by minimizing U_f , U_s , U_b , and U_r simultaneously within first several iterations. During the optimization, Υ_1 automatically decreases and Υ_2 automatically increases. This decreases the limitation on the bandwidth and increases the focus on the filter response as the feature frequencies approach to the design specification. When all the passband feature frequencies locate inside the specification, Υ_3 automatically equals to Υ_2 , activating the boundary feature heights to fine optimizing the filter response. The proposed cognition-driven optimization technique, which uses transfer function-based feature surrogate to drive EM optimization, can reach the optimal EM solution faster than the multifeature-assisted surrogate-based EM optimization [81], when the filter response at the starting point for the design optimization is substantially misaligned with the design specifications.

Fig. 4.4 shows the flowchart of the proposed cognition-driven optimization using transfer function-based feature surrogate. The proposed optimization algorithm is summarized as follows.

Step 1) Initialize $k = 1$. Set initial starting point $\phi_F^{(k)}$ and initial trust radius $\mathbf{P}^{(k)}$.

Initialize Υ_1 , Υ_2 , and Υ_3 .

Step 2) Generate multiple EM geometrical samples in parallel around $\phi_F^{(k)}$ in $\Omega_s^{(k)}$

using (4.23) and (4.24) following orthogonal distribution.

Step 3) Extract the five sets of feature parameters $\mathbf{f}^{n,(k)}$, $\mathbf{s}^{n,(k)}$, $\mathbf{b}^{n,(k)}$, $\hat{\mathbf{r}}^{n,(k)}$, and

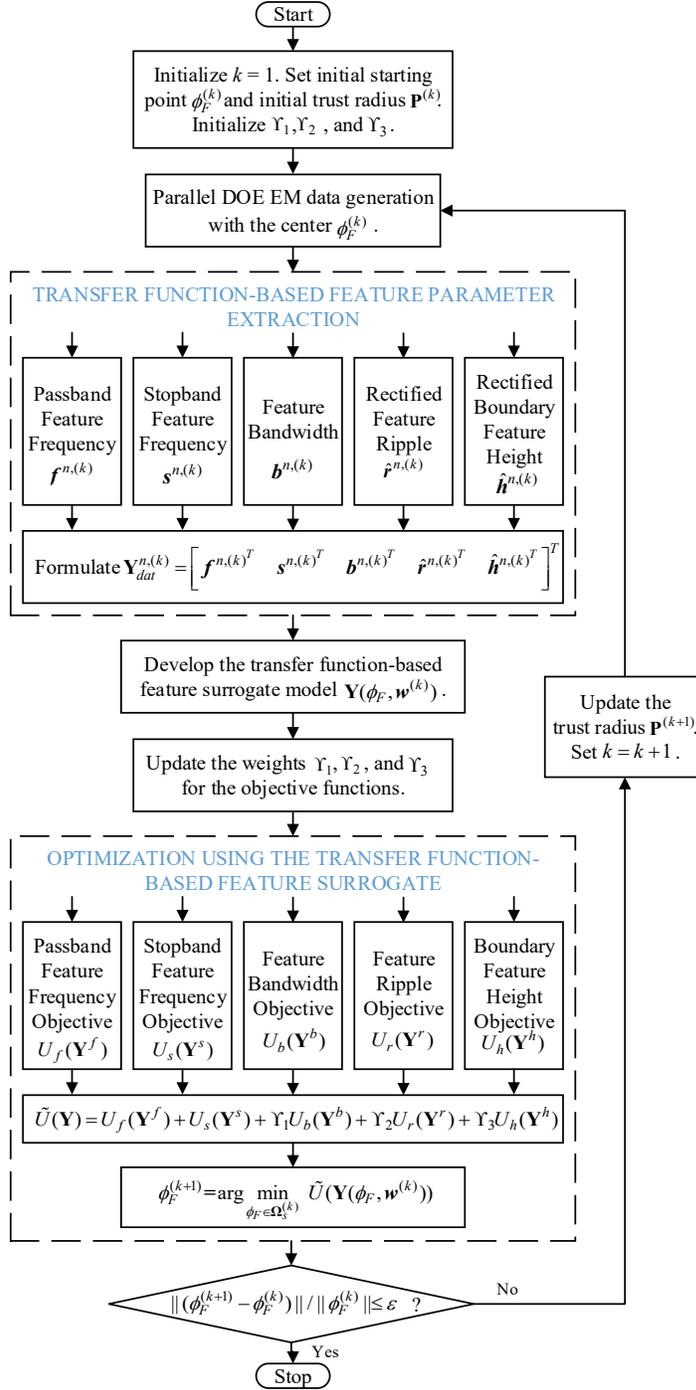


Figure 4.4: The flowchart of the proposed cognition-driven optimization using transfer function-based feature surrogate.

$\hat{\mathbf{h}}^{n,(k)}$ for all the geometrical samples, $n = 1, 2, \dots, N_g$, using (4.2) - (4.22).

Step 4) Develop the transfer function-based feature surrogate model using (4.25) - (4.27).

Step 5) Update the weights Υ_1 , Υ_2 , and Υ_3 for the objective functions using (4.44) - (4.46).

Step 6) Optimize the transfer function-based feature surrogate model to obtain the optimal surrogate solution $\phi_F^{(k+1)}$ using (4.28) - (4.43) and (4.47).

Step 7) If the termination condition (4.51) is satisfied, go to Step 9); otherwise, go to Step 8).

Step 8) Update the trust radius $\mathbf{P}^{(k+1)}$ using (4.48) - (4.50) and increase the iteration index $k = k + 1$, then go to Step 2).

Step 9) Stop the optimization process.

4.4 Application Examples

4.4.1 Optimization of a Three-Pole H-Plane Filter

In the first example, we consider the optimization of a three-pole H-plane filter [167], as shown in Fig. 4.5, to illustrate the proposed optimization technique. The cross section of the waveguide is 19.05 mm \times 9.525 mm (WR-75). The lengths of the resonant cavities are defined as L_1 and L_2 . The widths of the coupling windows are

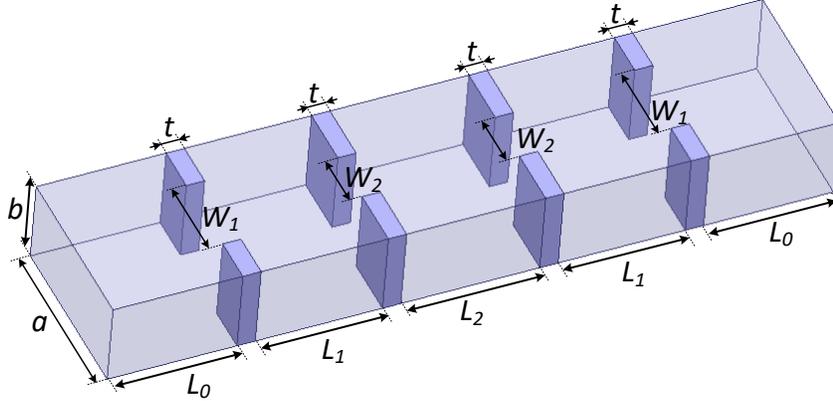


Figure 4.5: Structure of the three-pole H-plane filter. Four geometrical variables are used for the proposed optimization, i.e., $\phi_F = [L_1 \ L_2 \ W_1 \ W_2]^T$.

defined as W_1 and W_2 . The design variables for this filter are $\phi_F = [L_1 \ L_2 \ W_1 \ W_2]^T$. For this three-pole example, the transfer function-based feature parameters consist of three passband feature frequencies (i.e., $\mathbf{Y}^f = [Y_1^f \ Y_2^f \ Y_3^f]^T$, $N_f = 3$), two stopband feature frequencies (i.e., $\mathbf{Y}^s = [Y_1^s \ Y_2^s]^T$), two feature bandwidths (i.e., $\mathbf{Y}^b = [Y_1^b \ Y_2^b]^T$), two feature ripples (i.e., $\mathbf{Y}^r = [Y_1^r \ Y_2^r]^T$), and two boundary feature heights (i.e., $\mathbf{Y}^h = [Y_1^h \ Y_2^h]^T$).

The original design specifications for this example are defined as $|S_{11}| \leq -20$ dB in the frequency range of 11.85 GHz - 12.15 GHz and $|S_{11}| \geq -1$ dB in the frequency range of 11.5 GHz - 11.7 GHz and 12.3 GHz - 12.5 GHz. From the original design specifications, we obtain that $f_{p1} = 11.85$ GHz, $f_{p2} = 12.15$ GHz, $f_{s1} = 11.7$ GHz, and $f_{s2} = 12.3$ GHz in the proposed optimization. According to (4.30) and (4.31), the specifications for the three passband feature frequencies are

$$11.85 \text{ GHz} \leq Y_1^f \leq 11.95 \text{ GHz}$$

$$11.95 \text{ GHz} \leq Y_2^f \leq 12.05 \text{ GHz}$$

$$12.05 \text{ GHz} \leq Y_3^f \leq 12.15 \text{ GHz}$$

The specifications for the two stopband feature frequencies are determined according to (4.33) and (4.34) as

$$11.7 \text{ GHz} \leq Y_1^s \leq 11.85 \text{ GHz}$$

$$12.15 \text{ GHz} \leq Y_2^s \leq 12.3 \text{ GHz}$$

The specifications for the two feature bandwidths are $0.1 \text{ GHz} \leq Y_i^b \leq 0.15 \text{ GHz}$, $i = 1, 2$, which are determined from (4.36). The specifications for the two feature ripples are $Y_i^r \geq 1$, $i = 1, 2$, based on (4.21). The specifications for the two boundary feature heights are $Y_i^h \geq 1$, $i = 1, 2$, based on (4.22).

The starting point is $\phi_F^{(1)} = [15 \ 17 \ 11 \ 8]^T$ (mm). The EM response at the starting point is substantially misaligned with the design specifications, see Fig. 4.6(a). EM evaluations are performed by *HFSS* to obtain fine model EM responses. In this example, the number of design variables is $N_v = 4$. The level of the orthogonal distribution N_{or} should satisfy $N_{or} \geq \max\{\sqrt{15}, 3\}$. Since N_{or} is better to be an odd number to make the samples orthogonal to each other, N_{or} is determined to be five. Therefore, in each iteration, the surrogate model is constructed using EM data of 26 geometrical samples ($5^2 + 1$), including the center point and 25 orthogonal samples around the center point, generated simultaneously in parallel. The proposed technique is suitable for EM-based design optimization of Chebyshev- and elliptic-type responses. The filter responses of this example are standard three-pole Chebyshev curves.

To perform the proposed optimization, we set the initial range of the surrogate model to be $\mathbf{P}^{(1)} = [5\% \ 5\% \ 10\% \ 10\%]^T$ around the center point $\phi_F^{(1)}$. The number of

effective zeros N_e of the transfer function is five for this example. The filter response at the starting point, the filter response after the 3rd iteration, and the final solution using the proposed technique are shown in Fig. 4.6. As shown in Fig. 4.6(b), all the passband feature frequencies are already inside the passband after three iterations of the proposed cognition-driven optimization. The proposed optimization reaches the final EM optimal solution $\phi_F^{(7)} = [13.6075 \ 14.958 \ 9.4719 \ 6.3185]^T$ (mm) after six iterations, as shown in Fig. 4.6(c). As shown in Fig. 4.6(a)-(c), the proposed technique can work well even with the starting point whose response does not have explicitly identifiable feature information. In this way, the proposed optimization can be used for optimizations with different initial points (with/without explicitly identifiable feature information) to search for the optimal solutions.

For comparison purposes, we have also performed the multifeature-assisted optimization in [81] which uses the feature frequencies and feature heights to assist the optimization. The starting point of the multifeature-assisted optimization is the same as that of the proposed technique. Comparison of EM responses between the proposed cognition-driven EM optimization and the multifeature-assisted optimization is shown in Fig. 4.6. As shown in Fig. 4.6(d) and Fig. 4.6(e), the multifeature-assisted optimization moves all the feature frequencies inside the passband after seven iterations and achieves the final solution after nine iterations. In the multifeature-assisted optimization, the neuro-TF surrogate model developed with highly nonlinear filter frequency responses is used for optimization. In the proposed method, filter feature parameters are used to develop the feature surrogate model for optimization. Since the relationship between filter feature parameters

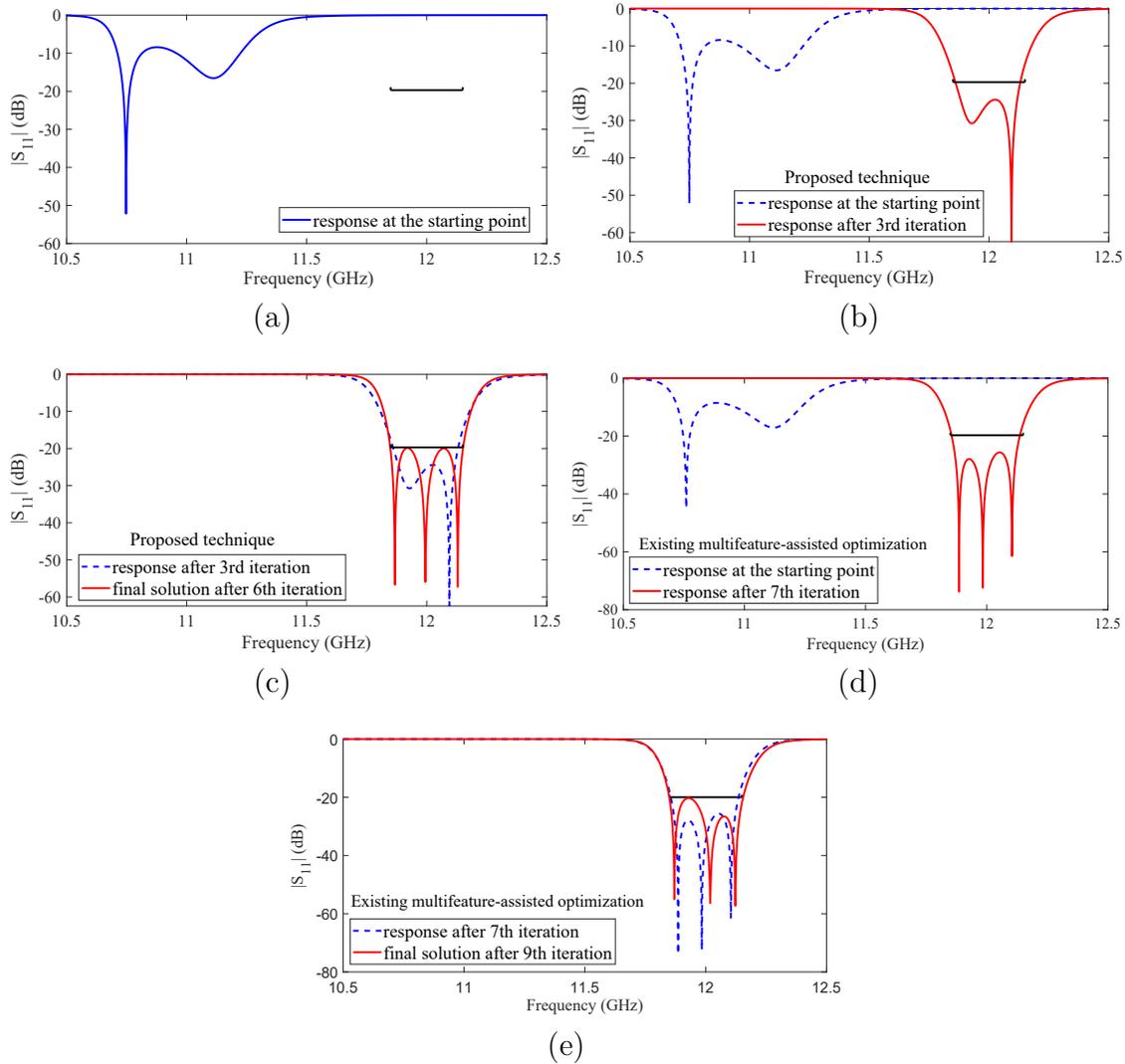


Figure 4.6: Comparison of EM responses between the proposed cognition-driven EM optimization and the multifeature-assisted optimization for the three-pole H-plane filter. (a) EM response at the starting point for both techniques. (b) EM responses from the starting point (dashed line) to the 3rd iteration (solid line) of the proposed optimization. (c) EM responses from the 3rd iteration (dashed line) to the final solution (solid line) after 6 iterations of the proposed optimization. (d) EM responses from the starting point (dashed line) to the 7th iteration (solid line) of the multifeature-assisted optimization. (e) EM responses from the 7th iteration (dashed line) to the final solution (solid line) after 9 iterations of the multifeature-assisted optimization. The black lines represent the specification.

Table 4.1: Comparison of CPU Time of Different Optimization Methods for the Three-Pole H-Plane Filter

Optimization methods	Direct EM optimization	Single-feature assisted optimization	Multi-feature assisted optimization	Proposed optimization
No. of iterations	149	15	9	6
Fine model simulation time in each iteration	2.4 min	3.2 min	3.2 min	3.2 min
No. of fine model evaluations in each iteration	1	26	26	26
Total fine model simulation time	6 h	3.2 min \times 16	3.2 min \times 10	3.2 min \times 7
Time of surrogate model training and optimization	-	65 s \times 15	65 s \times 9	1 s \times 6
Total time	6 h	67.5 min	41.8 min	22.5 min

and geometrical design variables has much lower nonlinearity than the relationship between filter frequency responses and geometrical design variables, the proposed feature surrogate model is much less nonlinear than the neuro-TF surrogate model in [81]. Therefore, the proposed feature surrogate model can have larger model range and better extrapolation compared to the neuro-TF surrogate model, resulting in a faster convergence of the proposed cognition-driven optimization over the multifeature-assisted optimization in [81].

Table 4.1 shows the comparison of the proposed optimization, the multifeature-assisted optimization [81], the single-feature assisted optimization [80], and the direct EM optimization. Because of the usage of parallel computation, the fine model simulation time of the proposed optimization is similar to that of the direct EM optimization in each optimization iteration. Since there is some overhead time when performing parallel computation, the fine model simulation time of the proposed optimization is slightly longer than that of the direct EM optimization in each optimization iteration. The proposed optimization takes 22.5 min to satisfy the design specifications, which is faster than the other optimization methods in the comparison.

4.4.2 Optimization of a Four-Pole Waveguide Filter

In this example, we illustrate the proposed optimization technique using a four-pole waveguide filter with tuning elements [168], as shown in Fig. 4.7. The heights of the posts between the coupling windows are defined as h_1 , h_2 , and h_3 . The heights of the posts in the resonant cavities are defined as h_{c1} and h_{c2} . The design variables are $\boldsymbol{\phi}_F = [h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2}]^T$. The total number of feature parameters for this four-pole filter is three more than that of the three-pole filter example in the previous subsection. There are 14 transfer function-based feature parameters for this four-pole filter, including four passband feature frequencies (i.e., $\mathbf{Y}^f = [Y_1^f \ Y_2^f \ Y_3^f \ Y_4^f]^T$, $N_f = 4$), two stopband feature frequencies (i.e., $\mathbf{Y}^s = [Y_1^s \ Y_2^s]^T$), three feature bandwidths (i.e., $\mathbf{Y}^b = [Y_1^b \ Y_2^b \ Y_3^b]^T$), three feature ripples (i.e., $\mathbf{Y}^r = [Y_1^r \ Y_2^r \ Y_3^r]^T$), and two boundary feature heights (i.e., $\mathbf{Y}^h = [Y_1^h \ Y_2^h]^T$).

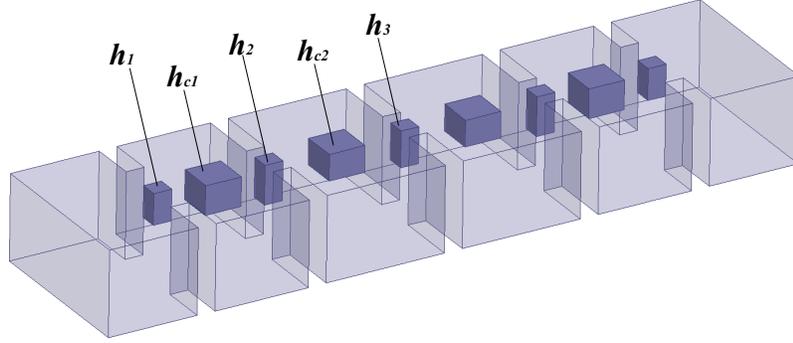


Figure 4.7: Structure of the four-pole waveguide filter. Five geometrical variables are used for the proposed EM design optimization.

For this example, the original design specification is $|S_{11}| \leq -20$ dB in the frequency range of 10.85 GHz - 11.15 GHz and $|S_{11}| \geq -1$ dB in the frequency range of 10.5 GHz - 10.7 GHz and 11.3 GHz - 11.5 GHz. According to the original specification, we obtain the specifications for the four passband feature frequencies using (4.30) and (4.31) as

$$10.85 \text{ GHz} \leq Y_1^f \leq 10.925 \text{ GHz}$$

$$10.925 \text{ GHz} \leq Y_2^f \leq 11.0 \text{ GHz}$$

$$11.0 \text{ GHz} \leq Y_3^f \leq 11.075 \text{ GHz}$$

$$11.075 \text{ GHz} \leq Y_4^f \leq 11.15 \text{ GHz}$$

The specifications for the two stopband feature frequencies are obtained based on (4.33) and (4.34) as

$$10.7 \text{ GHz} \leq Y_1^s \leq 10.85 \text{ GHz}$$

$$11.15 \text{ GHz} \leq Y_2^s \leq 11.3 \text{ GHz}$$

The specifications for the three feature bandwidths based on (4.36) are 0.075 GHz

$\leq Y_i^b \leq 0.1$ GHz, $i = 1, 2, 3$. The specifications for the three feature ripples are $Y_i^r \geq 1$, $i = 1, 2, 3$, based on (4.21). The specifications for the two boundary feature heights are $Y_i^h \geq 1$, $i = 1, 2$, based on (4.22).

The starting point for this example is $\boldsymbol{\phi}_F^{(1)} = [4.0 \ 5.0 \ 4.8 \ 4.0 \ 3.7]^T$ (mm), whose EM response is substantially misaligned with the design specifications, see Fig. 4.8(a). EM evaluations are performed by *HFSS* to obtain fine model EM responses using fast simulation feature. In this example, the number of design variables is $N_v = 5$. The level of the orthogonal distribution N_{or} should satisfy $N_{or} \geq \max\{\sqrt{21}, 4\}$. Since N_{or} is better to be an odd number, N_{or} is determined to be five. Therefore, in each iteration, the surrogate model is constructed using EM data of 26 geometrical samples ($5^2 + 1$), including the center point and 25 orthogonal samples around the center point, generated simultaneously in parallel.

We set the initial range of the surrogate model to be $\boldsymbol{P}^{(1)} = [8\% \ 8\% \ 8\% \ 4\% \ 4\%]^T$ around the center point $\boldsymbol{\phi}_F^{(1)}$. The number of effective zeros N_e of the transfer function is six. Fig. 4.8 shows the filter response at the starting point, the filter response after the 5th iteration, and the final solution using the proposed method. As shown in Fig. 4.8(b), the proposed cognition-driven optimization moves all the feature frequencies into the passband after five iterations. The final EM optimal solution $\boldsymbol{\phi}_F^{(9)} = [3.2572 \ 3.8693 \ 3.3811 \ 3.3369 \ 2.9924]^T$ (mm) is achieved after eight iterations using the proposed optimization, as shown in Fig. 4.8(c).

For comparison purposes, we have also performed the multifeature-assisted optimization in [81] which uses the feature frequencies and feature heights to assist the optimization. The starting point of the multifeature-assisted optimization is the

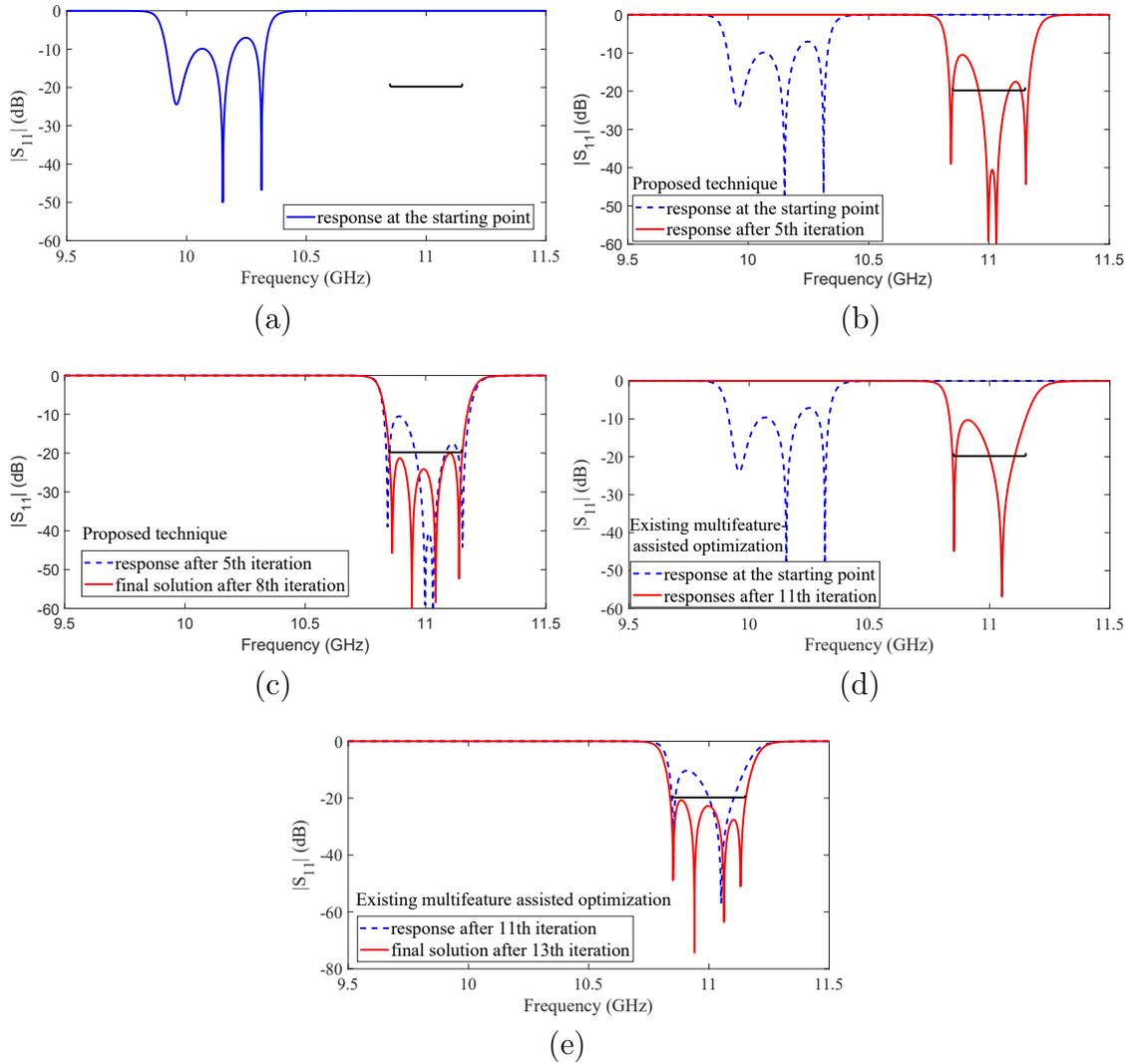


Figure 4.8: Comparison of EM responses between the proposed cognition-driven EM optimization and the multifeature-assisted optimization for the four-pole waveguide filter. (a) EM response at the starting point for both techniques. (b) EM responses from the starting point (dashed line) to the 5th iteration (solid line) of the proposed optimization. (c) EM responses from the 5th iteration (dashed line) to the final solution (solid line) after 8 iterations of the proposed optimization. (d) EM responses from the starting point (dashed line) to the 11th iteration (solid line) of the multifeature-assisted optimization. (e) EM responses from the 11th iteration (dashed line) to the final solution (solid line) after 13 iterations of the multifeature-assisted optimization. The black lines represent the specification.

Table 4.2: Comparison of CPU Time of Different Optimization Methods for the Four-Pole Waveguide Filter

Optimization methods	Direct EM optimization	Single-feature assisted optimization	Multi-feature assisted optimization	Proposed optimization
No. of iterations	300	25	13	8
Fine model simulation time in each iteration	3.8 min	5 min	5 min	5 min
No. of fine model evaluations in each iteration	1	26	26	26
Total fine model simulation time	19 h	5 min \times 26	5 min \times 14	5 min \times 9
Time of surrogate model training and optimization	-	69 s \times 25	69 s \times 13	1 s \times 8
Total time	19 h*	158.8 min	85.0 min	45.1 min

*Design specifications are not satisfied.

same as that of the proposed technique. Fig. 4.8 shows the comparison of EM responses between the proposed optimization and the multifeature-assisted optimization for this example. From Fig. 4.8(d) and Fig. 4.8(e), the multifeature-assisted optimization moves all the feature frequencies into the passband after 11 iterations and achieves the final solution after 13 iterations. Compared to the multifeature-assisted optimization in [81], the proposed optimization converges faster to satisfy

the design specifications. Since the starting point in this example is much farther away from the design specification than those in [78] and [81], this example also demonstrates that the proposed algorithm works well for initial designs that are substantially misaligned with the optimal design.

The comparisons of the proposed optimization, the multifeature-assisted optimization [81], the single-feature assisted optimization [80], and direct EM optimization are shown in Table 4.2. The proposed optimization takes 45.1 min to satisfy the design specifications, which is faster than the other optimization methods.

4.5 Conclusion

An advanced cognition-driven EM optimization incorporating transfer function-based feature surrogate for EM optimization of microwave filters has been proposed in this chapter. The proposed optimization technique has addressed the challenges where the response at the starting point for the design optimization is substantially misaligned with the design specifications and the filter response does not have explicitly identifiable feature information. Multiple feature parameters, including passband feature frequencies, stopband feature frequencies, feature bandwidths, feature ripples, and boundary feature heights, have been extracted from the transfer function response in the proposed technique. New objective functions have been formulated directly in the feature space for the cognition-driven optimization. The proposed cognition-driven optimization incorporating transfer function-based feature surrogate has achieved faster convergence than the existing feature-assisted EM optimization methods.

Chapter 5

Efficient EM Topology Optimization Incorporating Advanced Matrix Padé via Lanczos and Genetic Algorithm for Microwave Components

5.1 Introduction

The cognition-driven optimization method presented in the previous chapter is suitable and efficient for the EM geometry optimization, where the performance of the microwave component is optimized by adjusting the values of a given set of geometrical parameters, such as length and width of microwave structures. However, it is not suitable for the EM topology optimization where different EM structures cannot be related by changing the values of a given set of geometrical parameters. Therefore, repetitive EM simulations for different shapes and topologies of EM structures are needed in EM topology optimization process. Typical EM simulation methods

such as finite element method (FEM) require solving matrix equations with very large matrices. In the EM topology optimization, large matrix equations (e.g. FEM matrix equations) need to be solved repetitively for different EM structures over a frequency range, which is time-consuming.

In this chapter, we propose an efficient EM topology optimization technique [100] for the design of microwave components. In the proposed technique, the three-dimensional FEM is used for EM simulation of EM structures with different shapes and topologies [169]. We propose a new method to integrate Matrix Padé via Lanczos (MPVL) and Householder formula so that the FEM equations can be solved efficiently during the EM topology optimization process. Such an integration cannot be achieved by routine derivation. It is because Householder format of FEM is inconsistent with the requirement of MPVL, and the complex MPVL iterative formulas do not match well with the required submatrix format of Householder formula. One of the major innovations of this chapter is a set of new formulas and new formulations that are derived to effectively integrate the MPVL iterative formulations and the Householder formula with each other. The proposed method combines the advantages of MPVL and Householder formula, where the advantage of the former is to obtain EM solutions over a frequency range by the information calculated at one single frequency, and the advantage of the latter is to reduce the large FEM matrix problem (associated with all the FEM elements) into a small matrix problem (only associated with the FEM elements in the design space). Using the proposed method, the effort of solving the large FEM matrix equation at many frequencies is reduced to the effort of solving only a small matrix problem at a

single frequency point, thereby speeding up the topology optimization process. A further innovation of this chapter is a new method that is proposed to exploit the inheritance pattern of GA so that the small matrix problem can be reduced into an even smaller one. New formulations are derived to exploit the common factors between the new EM structure generated in the optimization process and the old EM structure in the previous iteration. Taking advantage of the extracted common factors, the EM solutions of new EM structures can be computed by solving even smaller matrix equations. In this way, the computational expense during the optimization process can be further reduced. The proposed method, which integrates both MPVL and Householder formula and explores the pattern between the old and new EM structures, can greatly accelerate the topology optimization process.

5.2 The Proposed EM Topology Optimization Incorporating Advanced FEM-MPVL and GA

To illustrate the idea of EM topology optimization [84], we use a planer filter example as shown in Fig. 5.1. The design space is the area where the material can change between metal and nonmetal in the optimization process. The design space is divided into a number of small rectangular cells. The material in each cell can change from metal to nonmetal (or vice versa) during the EM topology optimization. The material in each cell is defined as a variable. Let $\phi_\tau, \tau = 1, 2, 3, \dots, N_v$, be the variable representing the material in the τ th cell of the design space, where N_v is the number of small cells in the design space. As shown in Fig. 5.1, $\phi_\tau = 1$

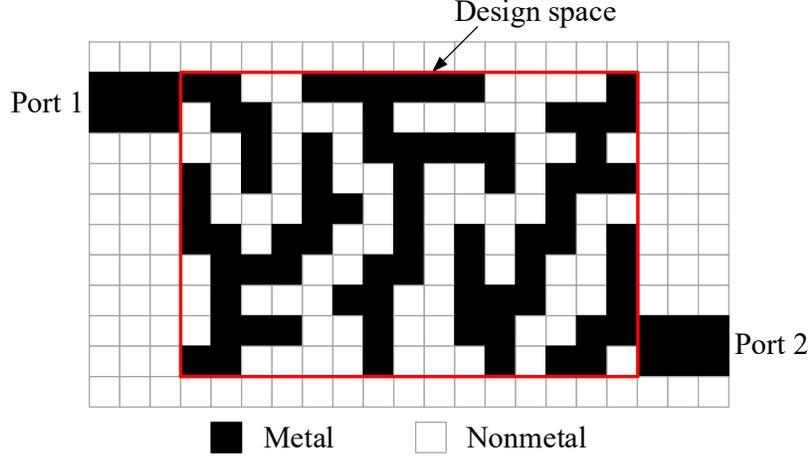


Figure 5.1: The concept of EM topology optimization as illustrated through a planar filter configuration. During optimization, the material in each cell inside the design space can independently change from metal to nonmetal (or vice versa). EM topology optimization will adjust the metal/nonmetal pattern so that the filter response is optimized.

represents that the τ th cell is covered with metal, and $\phi_\tau = 0$ represents that the τ th cell is not covered with metal. The shape and topology of metal in the design space is represented by a binary vector defined as $\boldsymbol{\phi} = [\phi_1 \ \phi_2 \ \phi_3 \ \dots \ \phi_{N_v}]^T$. Let $\mathcal{S}(\boldsymbol{\phi})$ represent the S-parameters of EM responses for the microwave component whose shape and topology is represented by $\boldsymbol{\phi}$. The EM topology optimization is formulated as

$$\boldsymbol{\phi}^* = \arg \min_{\boldsymbol{\phi}} U(\mathcal{S}(\boldsymbol{\phi})) \quad (5.1)$$

where U represents a user-defined objective function, which is usually a minimax or a generalized l_p function, and $\boldsymbol{\phi}^*$ represents the optimal shape and topology of metal in the design space at which the filter response satisfies the design specification.

In the EM topology optimization process using GA, the binary vector represent-

ing the shape and topology of metal in the design space is used as the chromosome in GA. GA is not working on a single chromosome at a time, but on a whole population of chromosomes [170]. New chromosomes are created and optimized by GA until the optimal shape and topology is found.

5.2.1 Proposed Method to Incorporate Householder Formula Into FEM-MPVL Equations

Let k represent the generation counter in GA optimization. Let $\phi^{(k,m)}$ represent the m th chromosome in the population of k th generation. Each chromosome means an EM structure. Different chromosomes mean EM structures with different shapes and topologies. To compute the S-parameters for each shape and topology (represented by $\phi^{(k,m)}$) in the optimization process, FEM is used to solve the EM problem and calculate the field solution. The FEM equation can be formulated in a generalized form as [149]

$$(\mathbf{K}_0^{(k,m)} + \gamma \mathbf{K}_1^{(k,m)} + \gamma^2 \mathbf{K}_2^{(k,m)}) \mathbf{e}^{(k,m)} = \gamma \mathbf{R} \quad (5.2)$$

where γ is the complex propagation constant, which is a function of the frequency; $\mathbf{K}_0^{(k,m)}$, $\mathbf{K}_1^{(k,m)}$, and $\mathbf{K}_2^{(k,m)}$ are FEM matrices, which are dependent on $\phi^{(k,m)}$ but independent on γ ; \mathbf{R} is a vector describing the EM excitation at the input port; $\mathbf{e}^{(k,m)}$ represents the solution vector containing the unknown values used in approximating the electrical field for the EM structure whose shape and topology is represented by $\phi^{(k,m)}$. Let N represent the number of elements in the unknown solution vector $\mathbf{e}^{(k,m)}$.

In EM topology optimization, FEM equations need to be solved over a frequency

range. Here, we propose a new method to integrate MPVL and Householder formula so that the effort of solving the large FEM matrix equation at many frequencies is reduced to the effort of solving only a small matrix problem at one single frequency. This single frequency is defined as the frequency expansion point. Let γ_0 represent the complex propagation constant at the frequency expansion point for MPVL. In the proposed method, we define the EM structure without metal in the design space, i.e., $\phi_\tau = 0, \tau = 1, 2, \dots, N_v$, as the base EM structure. It is used to prepare the basic information for accelerating the calculation of EM responses during the optimization process. Let $\mathbf{K}_0^s, \mathbf{K}_1^s$, and \mathbf{K}_2^s be the FEM matrices for the base EM structure. Let \mathbf{K}_s represent the FEM matrix formulated at the frequency expansion point for the base EM structure as

$$\mathbf{K}_s = \mathbf{K}_0^s + \gamma_0 \mathbf{K}_1^s + \gamma_0^2 \mathbf{K}_2^s \quad (5.3)$$

The matrix \mathbf{K}_s is constant during the EM topology optimization process. The LU factors of \mathbf{K}_s are calculated and stored at the beginning of optimization, and will be used repeatedly when calculating EM responses of different EM structures during the optimization.

For the m th chromosome in the k th GA generation, let $B^{(k,m)}$ represent the index set of metal cells (i.e., cells represented by ones in the chromosome $\phi^{(k,m)}$), defined as

$$B^{(k,m)} = \{\tau | \phi_\tau^{(k,m)} = 1, \tau = 1, 2, \dots, N_v\} \quad (5.4)$$

Let J_τ be defined as the index set of unknowns in the τ th cell of the design space. The value of the index in J_τ is between 1 and N , indicating the position of the

unknown in the vector $\mathbf{e}^{(k,m)}$. For the arbitrary shape and topology represented by $\phi^{(k,m)}$, the index set of unknowns in all the metal cells is expressed as

$$J^{(k,m)} = \bigcup_{\tau \in B^{(k,m)}} J_{\tau} \quad (5.5)$$

The indices in the index set $J^{(k,m)}$ are in ascending order. In our formulation, each cell contains multiple FEM meshes. Therefore, each cell has several unknowns. In other words, each cell index in $B^{(k,m)}$ is related to several indices of unknowns in $J^{(k,m)}$.

In the proposed EM topology optimization, an arbitrary EM structure with metal in the design space is the result of changing the material in some cells of the base EM structure from nonmetal to metal. Changing the material in one cell from nonmetal to metal will affect a few elements in the FEM matrix of the base EM structure (i.e., $\mathbf{K}_0^s + \gamma \mathbf{K}_1^s + \gamma^2 \mathbf{K}_2^s$). If the τ th cell in the design space is covered with metal, i.e., $\phi_{\tau}^{(k,m)} = 1$, the electric field in the τ th cell is zero according to the boundary condition at the perfectly conducting surface [169]. The boundary condition where some of the elements in the solution vector $\mathbf{e}^{(k,m)}$ become zero can be achieved by adding a very large number (e.g., 10^6) to the corresponding diagonal elements in the FEM matrix [169]. These large numbers will force the unknown values (the electric field intensity) in the metal cells equal to zero. By imposing this boundary condition, the FEM matrix equations for an arbitrary EM structure whose shape and topology is represented by $\phi^{(k,m)}$ become

$$(\mathbf{K}_0^s + \gamma \mathbf{K}_1^s + \gamma^2 \mathbf{K}_2^s + \Delta \mathbf{K}^{(k,m)}) \mathbf{e}^{(k,m)} = \gamma \mathbf{R} \quad (5.6)$$

where $\Delta \mathbf{K}^{(k,m)}$ is an $N \times N$ diagonal matrix. The elements in the matrix $\Delta \mathbf{K}^{(k,m)}$ are

$$\Delta K_{i,j}^{(k,m)} = \begin{cases} \iota, & \text{if } i = j \text{ \& } i \in J^{(k,m)} \\ 0, & \text{otherwise} \end{cases} \quad (5.7)$$

where ι is a very large number (e.g., 10^6).

In order to integrate MPVL and Householder formula so that the FEM equation in (5.6) can be solved efficiently in GA optimization iterations, we formulate the matrix $\mathbf{A}_0^{(k,m)}$ at the frequency expansion point as

$$\mathbf{A}_0^{(k,m)} = \begin{bmatrix} -\gamma_0 \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{K}_0^s & \mathbf{K}_1^s + \gamma_0 \mathbf{K}_2^s + \frac{1}{\gamma_0} \Delta \mathbf{K}^{(k,m)} \end{bmatrix} \quad (5.8)$$

The formulation of the matrix in (5.8) is based on the matrix $\mathbf{A}_0^{(k,m)}$ in the Appendix section. Let ρ be the order of the reduced-order model using MPVL [149]. Let $\mathbf{T}^{(k,m)}$ represent the reduced-order matrix with the elements $t_{\xi,\eta}^{(k,m)}$, i.e., $\mathbf{T}^{(k,m)} = [t_{\xi,\eta}^{(k,m)}]_{\rho \times \rho}$, where $\xi = 1, 2, \dots, \eta$, and $\eta = 1, 2, \dots, \rho$. Let $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_\rho$ be a set of orthonormal vectors, where the size of each vector is $2N \times 1$. Let $\mathbf{V}^{(k,m)}$ represent the $2N \times \rho$ matrix containing the orthonormal basis of the Krylov subspaces for the model order reduction,

$$\mathbf{V}^{(k,m)} = [\mathbf{v}_1^{(k,m)} \ \mathbf{v}_2^{(k,m)} \ \dots \ \mathbf{v}_\rho^{(k,m)}] \quad (5.9)$$

Let $\mathbf{g}_\eta^{(k,m)}$ be a vector of size $2N \times 1$ for the calculation of $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ in the MPVL iterations [149] (see the Appendix section where the definition of the vector $\mathbf{g}_\eta^{(k,m)}$ is provided). Here, we derive new formulations to compute $\mathbf{g}_\eta^{(k,m)}$ by deriving the inversion of the matrix $\mathbf{A}_0^{(k,m)}$ in (5.8) using the block matrix inversion method

[171]. After the derivation, $\mathbf{g}_\eta^{(k,m)}$ is expressed as

$$\mathbf{g}_0^{(k,m)} = \begin{bmatrix} \mathbf{u}_0^{(k,m)} \\ \gamma_0 \tilde{\mathbf{u}}_0^{(k,m)} \end{bmatrix} \quad (5.10)$$

and

$$\mathbf{g}_\eta^{(k,m)} = \begin{bmatrix} \frac{1}{\gamma_0} (\mathbf{u}_\eta^{(k,m)} - \tilde{\mathbf{v}}_\eta^{(k,m)}) \\ \mathbf{u}_\eta^{(k,m)} \end{bmatrix}, \eta = 1, 2, \dots, \rho \quad (5.11)$$

where $\tilde{\mathbf{v}}_\eta^{(k,m)}$ is a vector which consists of elements from the first half of the vector $\mathbf{v}_\eta^{(k,m)}$ defined in (5.9); $\mathbf{u}_\eta^{(k,m)}$ is defined by

$$(\mathbf{K}_s + \Delta \mathbf{K}^{(k,m)}) \mathbf{u}_\eta^{(k,m)} = \begin{cases} \mathbf{R}, & \eta = 0, \\ \begin{bmatrix} \mathbf{K}_0^s \\ -\gamma_0 \mathbf{K}_2^s \end{bmatrix}^T \mathbf{v}_\eta^{(k,m)}, & \eta = 1, \dots, \rho. \end{cases} \quad (5.12)$$

where \mathbf{K}_s is the FEM matrix for the base EM structure formulated in (5.3). The matrices \mathbf{K}_s , \mathbf{K}_0^s , and \mathbf{K}_2^s do not change as the shape and topology change.

In the optimization process, (5.12) needs to be solved repetitively for different shapes and topologies. Every time the EM structure changes, the matrix $\Delta \mathbf{K}^{(k,m)}$ changes accordingly while the matrix \mathbf{K}_s remains unchanged. We propose an efficient method to solve (5.12) by using Householder formula [151]. In order to apply Householder formula to (5.12), the matrix $\Delta \mathbf{K}^{(k,m)}$ is expressed as

$$\Delta \mathbf{K}^{(k,m)} = \mathbf{Q}^{(k,m)} (t \mathbf{I}_{N_j^{(k,m)}}) [\mathbf{Q}^{(k,m)}]^T \quad (5.13)$$

where the matrix $\mathbf{Q}^{(k,m)}$ is an $N \times N_j^{(k,m)}$ matrix defined as $\mathbf{Q}^{(k,m)} = [\mathbf{q}_1^{(k,m)}, \mathbf{q}_2^{(k,m)}, \dots,$

$\mathbf{q}_{N_J^{(k,m)}}^{(k,m)}$], where $N_J^{(k,m)}$ is the number of indices in the index set $J^{(k,m)}$. The value of $N_J^{(k,m)}$ depends on the number of meshes in the metal cells in the design space. It is much smaller than the value of N , which depends on the total number of meshes in the whole EM structure. $\mathbf{I}_{N_J^{(k,m)}}$ represents the $N_J^{(k,m)} \times N_J^{(k,m)}$ identity matrix. The column $\mathbf{q}_j^{(k,m)}$, $j = 1, 2, 3, \dots, N_J^{(k,m)}$, has exactly one element equal to one and other elements equal to zero. The i th element in the vector $\mathbf{q}_j^{(k,m)}$, $j = 1, 2, 3, \dots, N_J^{(k,m)}$, is defined as

$$q_{i,j}^{(k,m)} = \begin{cases} 1, & \text{if } i = J^{(k,m)}(j) \\ 0, & \text{otherwise} \end{cases} \quad (5.14)$$

where $J^{(k,m)}(j)$ represents the j th element in the index set $J^{(k,m)}$.

Let $\mathbf{L}^{(k,m)}$ represent the solution of the matrix equation

$$\mathbf{K}_s \mathbf{L}^{(k,m)} = \mathbf{Q}^{(k,m)} \quad (5.15)$$

We define a matrix $\mathbf{M}^{(k,m)}$ formulated as

$$\mathbf{M}^{(k,m)} = [\mathbf{Q}^{(k,m)}]^T \mathbf{L}^{(k,m)} \quad (5.16)$$

According to (5.14), all elements in each column (say j th column) of the matrix $\mathbf{Q}^{(k,m)}$ are zero except for one element (the element whose index is the j th index in set $J^{(k,m)}$) which is one. Therefore, $[\mathbf{Q}^{(k,m)}]^T \mathbf{L}^{(k,m)}$ has the effect of selecting corresponding rows from the matrix $\mathbf{L}^{(k,m)}$. Specifically, the j th row in the matrix $\mathbf{M}^{(k,m)}$ is the i th row in the matrix $\mathbf{L}^{(k,m)}$, where i equals the j th index in the set $J^{(k,m)}$. In the actual algorithm, the matrix $\mathbf{M}^{(k,m)}$ is obtained by simply selecting suitable rows from the matrix $\mathbf{L}^{(k,m)}$ without performing the matrix multiplication

in (5.16).

By incorporating Householder formula [151], the solution of (5.12) is derived as

$$\mathbf{u}_\eta^{(k,m)} = \tilde{\mathbf{u}}_\eta^{(k,m)} - \mathbf{L}^{(k,m)}[\mathbf{M}^{(k,m)}]^{-1}\tilde{\mathbf{u}}_\eta^{(k,m)}, \eta = 0, 1, \dots, \rho. \quad (5.17)$$

where $\tilde{\mathbf{u}}_\eta^{(k,m)}$ is obtained by performing forward/backward (F/B) substitution in

$$\mathbf{K}_s \tilde{\mathbf{u}}_\eta^{(k,m)} = \begin{cases} \mathbf{R}, & \eta = 0, \\ \begin{bmatrix} \mathbf{K}_0^s \\ -\gamma_0 \mathbf{K}_2^s \end{bmatrix}^T \mathbf{v}_\eta^{(k,m)}, & \eta = 1, \dots, \rho. \end{cases} \quad (5.18)$$

and the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$ in (5.17) is assembled by selecting $N_J^{(k,m)}$ elements from the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$. Specifically, the j th element in the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$ equals the i th element in the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$, where i is the j th index in the set $J^{(k,m)}$.

Since the LU factors of the matrix \mathbf{K}_s are calculated and stored, solving (5.18) does not need to perform LU decompositions of new FEM matrices. Note that $\tilde{\mathbf{u}}_0^{(k,m)}$ is the same for different k and m . We use $\tilde{\mathbf{u}}_0$ to denote $\tilde{\mathbf{u}}_0^{(k,m)}$ in the following sections. By incorporating Householder formula, the FEM-MPVL equation has been simplified into solving a small matrix (i.e., the $N_J^{(k,m)} \times N_J^{(k,m)}$ matrix $\mathbf{M}^{(k,m)}$) equation as formulated in (5.17). Subsequently, we can obtain the vector $\mathbf{g}_\eta^{(k,m)}$, $\eta = 0, 1, 2, \dots, \rho$, by substituting (5.17) into (5.10) and (5.11).

The computed vector $\mathbf{g}_\eta^{(k,m)}$ is used for the calculation of $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ in the MPVL iterations [149]. Subsequently, the solution of the FEM-MPVL equation (i.e., $\tilde{\mathbf{e}}^{(k,m)}$, see the Appendix section where the specific definition is provided) is

calculated as [149]

$$\tilde{\mathbf{e}}^{(k,m)} = \gamma \|\mathbf{g}_0^{(k,m)}\| \mathbf{V}^{(k,m)} (\mathbf{I}_\rho - (\gamma - \gamma_0) \mathbf{T}^{(k,m)})^{-1} \mathbf{a}_1 \quad (5.19)$$

where $\mathbf{a}_1 = [1 \ 0 \ \dots \ 0]^T \in \mathbb{R}^\rho$; \mathbf{I}_ρ is an identity matrix in $\mathbb{R}^{\rho \times \rho}$. The EM solution of the FEM matrix equation $\mathbf{e}^{(k,m)}$ is the first half of the vector $\tilde{\mathbf{e}}^{(k,m)}$. Using MPVL, the matrices $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ are calculated only at the frequency expansion point. In other words, $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ are not required to be calculated at every frequency except the frequency expansion point. By changing the value of γ , we can obtain the FEM solution $\mathbf{e}^{(k,m)}$ at different frequencies using the information calculated from (5.19). In this way, for each frequency (each value of γ), we only need to solve a $\rho \times \rho$ matrix equation, where the value of ρ is very small (e.g., 8), even though the number of unknowns in the original FEM equation is hundreds of thousands or millions.

The proposed method combines the advantages of MPVL and Householder formula. If only MPVL method is used, the matrix equation to be solved is still very large even it needs to be solved at the frequency expansion point only. If only Householder formula is used, which has the same concept as the matrix partitioning method, the small matrix equation with an $N_J^{(k,m)} \times N_J^{(k,m)}$ matrix $\mathbf{M}^{(k,m)}$ has to be solved at every frequency in the required frequency range even though the matrix size has been reduced. Using the proposed method which incorporates Householder formula into the MPVL equations, the effort of solving the large FEM matrix equation at many frequencies is reduced to the effort of solving only a small matrix equation with the matrix $\mathbf{M}^{(k,m)}$ at a single frequency point (i.e., the fre-

quency expansion point). In this way, the proposed method integrating MPVL and Householder formula is much more efficient than the standard MPVL method or the existing matrix partitioning method.

In the derived equation (5.17), the matrix $\mathbf{L}^{(k,m)}$ has to be calculated for different chromosomes in the population of GA generations by solving (5.15). To reduce the expense to compute $\mathbf{L}^{(k,m)}$, in the next section, we propose an efficient method to obtain $\mathbf{L}^{(k,m)}$ for different chromosomes (i.e., different m values) in different generations (i.e., different k values) without solving (5.15) repetitively.

5.2.2 Efficient Method to Obtain $\mathbf{L}^{(k,m)}$ for (5.15)

In this section, we propose an efficient method to obtain $\mathbf{L}^{(k,m)}$ for (5.15). The proposed method is based on the fact that each column of the matrix $\mathbf{Q}^{(k,m)}$ in (5.15) corresponds to the index of one unknown in the metal cells of the design space, as defined in (5.14). For a given configuration of the EM structure, we need to generate FEM meshes only once at the beginning of the EM topology optimization process. Once generated, the FEM meshes are fixed for different $\phi^{(k,m)}$ (i.e., different values of k and m). The change of $\phi^{(k,m)}$ for different k and different m is only the change of the material (i.e., metal or nonmetal) within given cells in the design space. Therefore, the indices of unknowns in the design space are fixed during the optimization. Consider the EM structure with all cells in the design space covered with metal, i.e., $\phi_\tau = 1$ for $\tau = 1, 2, 3, \dots, N_v$, let $\tilde{\mathbf{Q}}$ represent the matrix containing columns corresponding to indices of unknowns in all the metal cells. To define the elements in the matrix $\tilde{\mathbf{Q}}$, we rewrite (5.5) and (5.14) specifically for the EM

structure with all cells covered with metal as

$$\tilde{J} = \bigcup_{\tau=1}^{N_v} J_\tau \quad (5.20)$$

$$\tilde{Q}_{i,j} = \begin{cases} 1, & \text{if } i = \tilde{J}(j) \\ 0, & \text{otherwise} \end{cases} \quad (5.21)$$

where \tilde{J} is the index set of unknowns in all cells in the design space, $\tilde{Q}_{i,j}$ is the (i, j) th element in the matrix \tilde{Q} , and $\tilde{J}(j)$ is the j th element in the index set \tilde{J} . Note that $j = 1, 2, \dots, N_{\tilde{J}}$ where $N_{\tilde{J}}$ represents the number of elements in \tilde{J} . For an arbitrary shape and topology represented by $\phi^{(k,m)}$, some of the cells in the design space are covered with metal, and some are not covered with metal. The indices of unknowns in those metal cells correspond to the columns in the matrix $\mathbf{Q}^{(k,m)}$. Therefore, we can select corresponding columns from \tilde{Q} to obtain the matrix $\mathbf{Q}^{(k,m)}$.

Let $\tilde{\mathbf{L}}$ be defined as the solution of the following matrix equation

$$\mathbf{K}_s \tilde{\mathbf{L}} = \tilde{\mathbf{Q}} \quad (5.22)$$

In the proposed method, we first calculate $\tilde{\mathbf{L}}$ for the EM structure with all cells covered with metal by performing F/B substitution to solve the matrix equation (5.22). For an arbitrary shape and topology represented by $\phi^{(k,m)}$, since the matrix $\mathbf{Q}^{(k,m)}$ for (5.15) can be obtained by selecting columns corresponding to the metal cells from \tilde{Q} , the matrix $\mathbf{L}^{(k,m)}$ can also be obtained by selecting corresponding columns from $\tilde{\mathbf{L}}$. In this way, we only need to solve (5.22) once to obtain $\tilde{\mathbf{L}}$ at the beginning of the EM topology optimization. Subsequently, the matrix $\mathbf{L}^{(k,m)}$ for

different shapes and topologies can be obtained quickly by selecting columns from the matrix $\tilde{\mathbf{L}}$ without solving (5.15) repetitively.

Algorithm 1 The proposed algorithm integrating MPVL and Householder formula for EM topology optimization

- 1) Obtain \mathbf{K}_0^s , \mathbf{K}_2^s , $\tilde{\mathbf{Q}}$, and LU factors of \mathbf{K}_s . Calculate $\tilde{\mathbf{u}}_0$ and $\tilde{\mathbf{L}}$. Define γ_0 and an orthonormal set of vectors $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_\rho$. Note that this step is performed only once at the beginning of the optimization.
- 2) For the m th chromosome in the k th GA generation, obtain $\mathbf{Q}^{(k,m)}$ and $\mathbf{L}^{(k,m)}$ by selecting corresponding columns from $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{L}}$, respectively.
- 3) Obtain the matrix $\mathbf{M}^{(k,m)}$ by selecting rows from the matrix $\mathbf{L}^{(k,m)}$. Obtain the vector $\tilde{\mathbf{u}}_0^{(k,m)}$ by selecting elements from the vector $\tilde{\mathbf{u}}_0$.
- 4) Calculate

$$\mathbf{u}_0^{(k,m)} = \tilde{\mathbf{u}}_0 - \mathbf{L}^{(k,m)}[\mathbf{M}^{(k,m)}]^{-1}\tilde{\mathbf{u}}_0^{(k,m)}$$

- 5) Set

$$\mathbf{g}_0^{(k,m)} = \begin{bmatrix} \mathbf{u}_0^{(k,m)} \\ \gamma \mathbf{u}_0^{(k,m)} \end{bmatrix}$$

Set

$$\mathbf{g} = \mathbf{g}_0^{(k,m)}.$$

For $\eta = 1, 2, \dots, \rho$, do Steps 6) to 12):

- 6) Set

$$\mathbf{v}_\eta^{(k,m)} = \frac{\mathbf{g}}{\|\mathbf{g}\|}.$$

- 7) If $\eta > 1$, set

$$t_{\eta,\eta-1}^{(k,m)} = \|\mathbf{g}\|.$$

- 8) Set

$$\lambda_\eta^{(k,m)} = \boldsymbol{\theta}_\eta^T \mathbf{v}_\eta^{(k,m)}.$$

- 9) Perform F/B substitutions to obtain $\tilde{\mathbf{u}}_\eta^{(k,m)}$ from

$$\mathbf{K}_s \tilde{\mathbf{u}}_\eta^{(k,m)} = \begin{bmatrix} \mathbf{K}_0^s \\ -\gamma_0 \mathbf{K}_2^s \end{bmatrix}^T \mathbf{v}_\eta^{(k,m)}$$

- 10) Obtain the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$ by selecting elements from the vector $\tilde{\mathbf{u}}_\eta^{(k,m)}$. Calculate

$$\mathbf{u}_\eta^{(k,m)} = \tilde{\mathbf{u}}_\eta^{(k,m)} - \mathbf{L}^{(k,m)}[\mathbf{M}^{(k,m)}]^{-1}\tilde{\mathbf{u}}_\eta^{(k,m)}$$

-
-
- 11) Assemble the vector $\tilde{\mathbf{v}}_\eta^{(k,m)}$ using the elements from the first half of the vector $\mathbf{v}_\eta^{(k,m)}$. Set

$$\mathbf{g}_\eta^{(k,m)} = \begin{bmatrix} \frac{1}{\gamma_0}(\mathbf{u}_\eta^{(k,m)} - \tilde{\mathbf{v}}_\eta^{(k,m)}) \\ \mathbf{u}_\eta^{(k,m)} \end{bmatrix}.$$

Set

$$\mathbf{g} = \mathbf{g}_\eta^{(k,m)}.$$

- 12) For $\xi = 1, 2, \dots, \eta$ do: Set

$$t_{\xi,\eta}^{(k,m)} = \frac{\boldsymbol{\theta}_\xi^T \mathbf{g}}{\lambda_\xi^{(k,m)}} \quad \text{and} \quad \mathbf{g} = \mathbf{g} - t_{\xi,\eta}^{(k,m)} \mathbf{v}_\xi^{(k,m)}.$$

- 13) Set

$$\mathbf{V}^{(k,m)} = \begin{bmatrix} \mathbf{v}_1^{(k,m)} & \mathbf{v}_2^{(k,m)} & \dots & \mathbf{v}_\rho^{(k,m)} \end{bmatrix},$$

$$\mathbf{T}^{(k,m)} = [t_{\xi,\eta}^{(k,m)}]_{\rho \times \rho}.$$

- 14) Calculate

$$\tilde{\mathbf{e}}^{(k,m)} = \gamma \|\mathbf{g}_0^{(k,m)}\| \mathbf{V}^{(k,m)} (\mathbf{I}_\rho - (\gamma - \gamma_0) \mathbf{T}^{(k,m)})^{-1} \mathbf{a}_1.$$

- 15) Assemble the EM solution vector $\mathbf{e}^{(k,m)}$ using the elements from the first half of the vector $\tilde{\mathbf{e}}^{(k,m)}$.
-

The overall algorithm combining the proposed method to incorporate Householder formula into MPVL and the proposed method to obtain $\mathbf{L}^{(k,m)}$ is summarized in Algorithm 1. With the proposed algorithm, the LU decomposition of the large FEM matrix (i.e., \mathbf{K}_s) needs to be performed only once in the entire optimization process. Subsequently, for an arbitrary shape and topology of the EM structure, the effort of solving the large FEM matrix equation at many frequencies is reduced

to the effort of solving the small matrix equation formulated in (5.17) only at the frequency expansion point. Therefore, the EM topology optimization process can be greatly accelerated.

As a further advance, we investigate the possibility of further reductions of the computational effort for solving the small matrix equation at the single frequency point in the next section.

5.2.3 Proposed Formulations to Update $[\mathbf{M}^{(k,m)}]^{-1}$ Efficiently Without Resolving the Matrix Inversion During Optimization

In order to further reduce the computational effort for solving the small matrix equation involving $\mathbf{M}^{(k,m)}$ as formulated in (5.17), we propose a method to update $[\mathbf{M}^{(k,m)}]^{-1}$ efficiently without resolving the matrix inversion of $\mathbf{M}^{(k,m)}$ during the optimization. The proposed method is based on the inheritance pattern of GA. In each generation of GA, by analyzing shape and topology patterns of new children EM structures versus the parent EM structures of the previous generation, we formulate a new method to exploit the common factors between parent EM structures and children EM structures so that the $[\mathbf{M}^{(k,m)}]^{-1}$ for the children EM structures can be updated by solving even smaller matrix equations. In this way, the EM topology optimization process can be further accelerated.

At the beginning of the GA process, N_p chromosomes, i.e., $\phi^{(1,m)}$, $m = 1, 2, \dots, N_p$ are generated as the first generation. N_p represents the number of chromosomes in each generation of GA. We find $[\mathbf{M}^{(1,m)}]^{-1}$, $m = 1, 2, \dots, N_p$, for chromosomes in the first generation. In the proposed EM topology optimization, we use steady-state

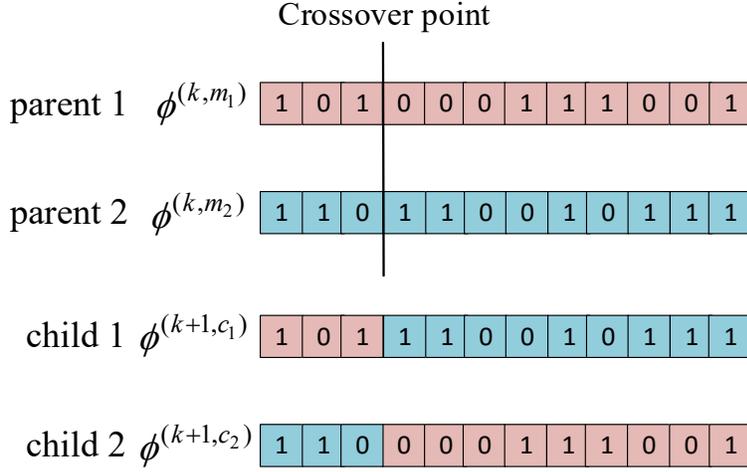


Figure 5.2: An illustration of a one-point crossover process in GA. Two children, i.e., two new chromosomes, are generated from the crossover of genes of two parents. The genes in each child chromosome are inherited partly from the parent 1 and partly from the parent 2.

genetic algorithm to update the population [172]. In each generation, two chromosomes in the current population are selected as parents. Two children, i.e., two new chromosomes, are then generated from the crossover of genes of two parents. Fig. 5.2 shows an illustration of a one-point crossover process. The genes in each child chromosome are inherited partly from the parent 1 and partly from the parent 2. To obtain the new generation, we identify the two worst chromosomes in the current population. The two worst chromosomes are the two chromosomes with the largest and second largest values of the objective function $U(\mathcal{S}(\phi^{(k,m)}))$, where the objective function U is used in the optimization process as shown in (5.1). The population of the new generation is obtained by replacing the two worst chromosomes of the current population with two new children chromosomes.

Let $\phi^{(k,m_1)}$ and $\phi^{(k,m_2)}$ represent two parent EM structures selected from the

population of k th generation. Let $\phi^{(k+1,c_1)}$ and $\phi^{(k+1,c_2)}$ represent two children EM structures generated by two selected parent EM structures for the $(k+1)$ th generation, where c_1 and c_2 represent the indices of two worst chromosomes in the k th generation. Suppose the inverse matrices of both parent EM structures, i.e., $[\mathbf{M}^{(k,m_1)}]^{-1}$ and $[\mathbf{M}^{(k,m_2)}]^{-1}$, are already known, we propose a method to derive the inverse matrix of $\mathbf{M}^{(k+1,c_1)}$ and $\mathbf{M}^{(k+1,c_2)}$ for children EM structures efficiently by using the information extracted from the known inverse matrices of parent EM structures. Here we demonstrate the proposed method by computing $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ of the first child EM structure from the known inverse matrices of parent EM structures.

For the first child EM structure generated by two selected parent EM structures, we compare the chromosome $\phi^{(k+1,c_1)}$ with the chromosomes of two parents, i.e., $\phi^{(k,m_1)}$ and $\phi^{(k,m_2)}$, respectively. Let B_α^i represent the index set of metal cells in the child EM structure that are inherited from the i th parent EM structure. Let J_α^i represent the index set of the unknowns in the metal cells that are inherited from the i th parent. B_α^i and J_α^i are represented by

$$B_\alpha^i = \{\tau | \phi_\tau^{(k,m_i)} = \phi_\tau^{(k+1,c_1)} = 1, \tau = 1, 2, \dots, N_v\}, i = 1, 2 \quad (5.23)$$

$$J_\alpha^i = \bigcup_{\tau \in B_\alpha^i} J_\tau, i = 1, 2 \quad (5.24)$$

Among all the metal cells of the i th parent, apart from the metal cells passed onto the child, there are some remaining metal cells. Let B_β^i represent the index set of these remaining metal cells in the i th parent. Let J_β^i represent the index set

of the unknowns in these remaining metal cells in the i th parent. B_β^i and J_β^i are represented by

$$B_\beta^i = \{\tau | \tau \in B^{(k,m_i)} \text{ and } \tau \notin B_\alpha^i\}, i = 1, 2 \quad (5.25)$$

$$J_\beta^i = \bigcup_{\tau \in B_\beta^i} J_\tau, i = 1, 2 \quad (5.26)$$

where $B^{(k,m_i)}$ includes the indices of all metal cells in the i th parent, which is obtained from (5.4). Among all the metal cells of the child, apart from the metal cells inherited from the i th parent, there are some other metal cells. Let B_ψ^i represent the index set of such other metal cells in the child that are not inherited from the i th parent. Let J_ψ^i represent the index set of the unknowns in such other metal cells not inherited from the i th parent. B_ψ^i and J_ψ^i are represented by

$$B_\psi^i = \{\tau | \tau \in B^{(k+1,c_1)} \text{ and } \tau \notin B_\alpha^i\}, i = 1, 2 \quad (5.27)$$

$$J_\psi^i = \bigcup_{\tau \in B_\psi^i} J_\tau, i = 1, 2 \quad (5.28)$$

where $B^{(k+1,c_1)}$ includes the indices of all metal cells in the child, which is obtained from (5.4). Let N_α^i , N_β^i , and N_ψ^i represent the number of elements in the index sets J_α^i , J_β^i , and J_ψ^i , respectively.

Fig. 5.3 shows a simple example to illustrate the inheritance pattern of GA. As shown in Fig. 5.3, the chromosome $\phi^{(k+1,c_1)}$ of the child (child 1) is compared with the chromosomes of two parents, i.e., $\phi^{(k,m_1)}$ and $\phi^{(k,m_2)}$, respectively. Comparing child 1 and parent 1, there are four common metal cells that are passed from parent 1 onto child 1. According to (5.23) and (5.24), B_α^1 includes the indices of these

	n	1	2	3	4	5	6	7	8	9	10	11	12
parent 1	$\phi^{(k,m_1)}$	1	0	1	0	0	0	1	1	1	0	0	1
child 1	$\phi^{(k+1,c_1)}$	1	0	1	1	1	0	0	1	0	1	1	1
		$B_\alpha^1 = \{1,3,8,12\}$				$B_\beta^1 = \{7,9\}$							
		$B_\psi^1 = \{4,5,10,11\}$											

	n	1	2	3	4	5	6	7	8	9	10	11	12
parent 2	$\phi^{(k,m_2)}$	1	1	0	1	1	0	0	1	0	1	1	1
child 1	$\phi^{(k+1,c_1)}$	1	0	1	1	1	0	0	1	0	1	1	1
		$B_\alpha^2 = \{1,4,5,8,10,11,12\}$						$B_\beta^2 = \{2\}$					
		$B_\psi^2 = \{3\}$											

Figure 5.3: A simple example to illustrate the inheritance pattern of GA. By comparing the chromosome of the child (child 1) with the chromosomes of two parents, we obtain the index sets B_α^i , B_β^i , and B_ψ^i for $i = 1, 2$.

four common metal cells, i.e., $B_\alpha^1 = \{1, 3, 8, 12\}$, and J_α^1 includes the indices of unknowns in these four common metal cells. Among the six metal cells in parent 1, apart from the four metal cells passed onto child 1, there are two remaining metal cells. According to (5.25) and (5.26), B_β^1 includes the indices of those two remaining metal cells, i.e., $B_\beta^1 = \{7, 9\}$, and J_β^1 includes the indices of unknowns in those two remaining metal cells. Among the eight metal cells in child 1, apart from the four metal cells inherited from parent 1, there are four other metal cells. According to (5.27) and (5.28), B_ψ^1 includes the indices of such other four metal cells, i.e., $B_\psi^1 = \{4, 5, 10, 11\}$, and J_ψ^1 includes the indices of unknowns in such other four metal cells.

The $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ of the child EM structure can be derived efficiently using the information from either $[\mathbf{M}^{(k,m_1)}]^{-1}$ of parent 1 or $[\mathbf{M}^{(k,m_2)}]^{-1}$ of parent 2. Suppose we choose parent 1 to provide the information for deriving the inverse matrix of the child EM structure. For the convenience of description, let $\tilde{\phi}$ represent the chosen parent (parent 1), i.e., $\tilde{\phi} = \phi^{(k,m_1)}$. Let \tilde{J}_α , \tilde{J}_β , and \tilde{J}_ψ represent index sets formulated in (5.24), (5.26), and (5.28) corresponding to the chosen parent $\tilde{\phi}$. Since $\tilde{\phi} = \phi^{(k,m_1)}$, \tilde{J}_α , \tilde{J}_β , and \tilde{J}_ψ equal J_α^1 , J_β^1 , and J_ψ^1 , respectively. Let $\tilde{\mathbf{M}}$ denote the matrix defined in (5.16) corresponding to the chosen parent EM structure $\tilde{\phi}$. Here $\tilde{\mathbf{M}} = \mathbf{M}^{(k,m_1)}$ because $\tilde{\phi} = \phi^{(k,m_1)}$. We divide the matrix $\tilde{\mathbf{M}}$ into block matrices according to the inheritance pattern. The rows in $\tilde{\mathbf{M}}$ are divided into ‘ α ’ rows (i.e., rows corresponding to \tilde{J}_α) and ‘ β ’ rows (i.e., rows corresponding to \tilde{J}_β). The columns in $\tilde{\mathbf{M}}$ are also divided into ‘ α ’ columns (i.e., columns corresponding to \tilde{J}_α) and ‘ β ’ columns (i.e., columns corresponding to \tilde{J}_β). Then the $\tilde{\mathbf{M}}$ for the parent $\tilde{\phi}$ is reformed as

$$\tilde{\mathbf{M}} = \begin{bmatrix} \tilde{\mathbf{M}}_{\alpha\alpha} & \tilde{\mathbf{M}}_{\alpha\beta} \\ \tilde{\mathbf{M}}_{\beta\alpha} & \tilde{\mathbf{M}}_{\beta\beta} \end{bmatrix} \quad (5.29)$$

where the size of $\tilde{\mathbf{M}}_{\alpha\alpha}$ is $N_\alpha \times N_\alpha$, the size of $\tilde{\mathbf{M}}_{\alpha\beta}$ is $N_\alpha \times N_\beta$, the size of $\tilde{\mathbf{M}}_{\beta\alpha}$ is $N_\beta \times N_\alpha$, and the size of $\tilde{\mathbf{M}}_{\beta\beta}$ is $N_\beta \times N_\beta$. Since $\tilde{\phi} = \phi^{(k,m_1)}$, N_α and N_β are N_α^1 and N_β^1 , respectively.

Accordingly, the inverse matrix of $\tilde{\mathbf{M}}$ for the chosen parent can be expressed as

$$\tilde{\mathbf{M}}^{-1} = \begin{bmatrix} \tilde{\mathbf{M}}_{\alpha\alpha} & \tilde{\mathbf{M}}_{\alpha\beta} \\ \tilde{\mathbf{M}}_{\beta\alpha} & \tilde{\mathbf{M}}_{\beta\beta} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbf{D}}_{\alpha\alpha} & \tilde{\mathbf{D}}_{\alpha\beta} \\ \tilde{\mathbf{D}}_{\beta\alpha} & \tilde{\mathbf{D}}_{\beta\beta} \end{bmatrix} \quad (5.30)$$

where $\tilde{\mathbf{D}}_{\alpha\alpha}$, $\tilde{\mathbf{D}}_{\alpha\beta}$, $\tilde{\mathbf{D}}_{\beta\alpha}$, and $\tilde{\mathbf{D}}_{\beta\beta}$ represent the sub-matrices in $\tilde{\mathbf{M}}^{-1}$. Since $\tilde{\mathbf{M}}^{-1}$ is already known, the sub-matrices $\tilde{\mathbf{D}}_{\alpha\alpha}$, $\tilde{\mathbf{D}}_{\alpha\beta}$, $\tilde{\mathbf{D}}_{\beta\alpha}$, and $\tilde{\mathbf{D}}_{\beta\beta}$ are known. The sub-matrix $\tilde{\mathbf{M}}_{\alpha\alpha}$ corresponds to the index set $\tilde{\mathcal{J}}_{\alpha}$. In other words, $\tilde{\mathbf{M}}_{\alpha\alpha}$ is associated with the metal cells inherited by the child. Therefore, $\tilde{\mathbf{M}}_{\alpha\alpha}$ is also a sub-matrix in the $\mathbf{M}^{(k+1,c_1)}$ of the child. Here we propose to extract $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$ from the known $\tilde{\mathbf{M}}^{-1}$ of the chosen parent so that the $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ of the child can be derived efficiently. Based on (5.30), if $N_{\alpha} \leq N_{\beta}$, we compute the $N_{\alpha} \times N_{\alpha}$ $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$ directly. If $N_{\alpha} > N_{\beta}$, we extract $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$ through solving an $N_{\beta} \times N_{\beta}$ size matrix equation as [173]

$$\tilde{\mathbf{M}}_{\alpha\alpha}^{-1} = \tilde{\mathbf{D}}_{\alpha\alpha} + \tilde{\mathbf{D}}_{\alpha\beta}(\mathbf{I}_{\beta\beta} - \tilde{\mathbf{M}}_{\beta\alpha}\tilde{\mathbf{D}}_{\alpha\beta})^{-1}\tilde{\mathbf{M}}_{\beta\alpha}\tilde{\mathbf{D}}_{\alpha\alpha} \quad (5.31)$$

where $\mathbf{I}_{\beta\beta}$ represents the $N_{\beta} \times N_{\beta}$ identity matrix. Let ζ represent the size of the matrix equation that needs to be solved in order to obtain $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$. ζ is calculated by

$$\zeta = \min(N_{\alpha}, N_{\beta}). \quad (5.32)$$

The matrix $\mathbf{M}^{(k+1,c_1)}$ for the child is also divided into block matrices according to the inheritance pattern. The rows in $\mathbf{M}^{(k+1,c_1)}$ are divided into ‘ α ’ rows and ‘ ψ ’ rows (i.e., rows corresponding to $\tilde{\mathcal{J}}_{\psi}$). The columns in $\mathbf{M}^{(k+1,c_1)}$ are also divided into ‘ α ’ columns and ‘ ψ ’ columns (i.e., columns corresponding to $\tilde{\mathcal{J}}_{\psi}$). Then the matrix $\mathbf{M}^{(k+1,c_1)}$ for the child is reformed as

$$\mathbf{M}^{(k+1,c_1)} = \begin{bmatrix} \mathbf{M}_{\alpha\alpha}^{(k+1,c_1)} & \mathbf{M}_{\alpha\psi}^{(k+1,c_1)} \\ \mathbf{M}_{\psi\alpha}^{(k+1,c_1)} & \mathbf{M}_{\psi\psi}^{(k+1,c_1)} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{M}}_{\alpha\alpha} & \mathbf{M}_{\alpha\psi}^{(k+1,c_1)} \\ \mathbf{M}_{\psi\alpha}^{(k+1,c_1)} & \mathbf{M}_{\psi\psi}^{(k+1,c_1)} \end{bmatrix} \quad (5.33)$$

where the size of $\mathbf{M}_{\alpha\alpha}^{(k+1,c_1)}$ is $N_{\alpha} \times N_{\alpha}$, the size of $\mathbf{M}_{\alpha\psi}^{(k+1,c_1)}$ is $N_{\alpha} \times N_{\psi}$, the size of

$\mathbf{M}_{\psi\alpha}^{(k+1,c_1)}$ is $N_\psi \times N_\alpha$, and the size of $\mathbf{M}_{\psi\psi}^{(k+1,c_1)}$ is $N_\psi \times N_\psi$. N_ψ equals N_ψ^1 because $\tilde{\phi} = \phi^{(k,m_1)}$.

The sub-matrix $\mathbf{M}_{\alpha\alpha}^{(k+1,c_1)}$ is inherited from $\tilde{\mathbf{M}}$ of the chosen parent, i.e., $\mathbf{M}_{\alpha\alpha}^{(k+1,c_1)} = \tilde{\mathbf{M}}_{\alpha\alpha}$. The $[\tilde{\mathbf{M}}_{\alpha\alpha}]^{-1}$ is already extracted from the chosen parent. Then the $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ can be derived efficiently using the block matrix inversion method [171] as

$$[\mathbf{M}^{(k+1,c_1)}]^{-1} = \begin{bmatrix} \mathbf{D}_{\alpha\alpha}^{(k+1,c_1)} & \mathbf{D}_{\alpha\psi}^{(k+1,c_1)} \\ \mathbf{D}_{\psi\alpha}^{(k+1,c_1)} & \mathbf{D}_{\psi\psi}^{(k+1,c_1)} \end{bmatrix} \quad (5.34)$$

where $\mathbf{D}_{\alpha\alpha}^{(k+1,c_1)}$, $\mathbf{D}_{\alpha\psi}^{(k+1,c_1)}$, $\mathbf{D}_{\psi\alpha}^{(k+1,c_1)}$, and $\mathbf{D}_{\psi\psi}^{(k+1,c_1)}$ are calculated in the following sequence

$$\mathbf{D}_{tmp} = \tilde{\mathbf{M}}_{\alpha\alpha}^{-1} \mathbf{M}_{\alpha\psi}^{(k+1,c_1)}, \quad (5.35)$$

$$\mathbf{D}_{\psi\psi}^{(k+1,c_1)} = (\mathbf{M}_{\psi\psi}^{(k+1,c_1)} - \mathbf{M}_{\psi\alpha}^{(k+1,c_1)} \mathbf{D}_{tmp})^{-1}, \quad (5.36)$$

$$\mathbf{D}_{\alpha\psi}^{(k+1,c_1)} = -\mathbf{D}_{tmp} \mathbf{D}_{\psi\psi}^{(k+1,c_1)}, \quad (5.37)$$

$$\mathbf{D}_{\psi\alpha}^{(k+1,c_1)} = [\mathbf{D}_{\alpha\psi}^{(k+1,c_1)}]^T, \quad (5.38)$$

$$\mathbf{D}_{\alpha\alpha}^{(k+1,c_1)} = \tilde{\mathbf{M}}_{\alpha\alpha}^{-1} - \mathbf{D}_{\alpha\psi}^{(k+1,c_1)} \mathbf{D}_{tmp}^T. \quad (5.39)$$

With the proposed formulations, we can obtain $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ for the new child EM structure by solving only two smaller matrix equations. One is to extract $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$ by solving a $\zeta \times \zeta$ matrix equation according to (5.29)-(5.32). The other one is to solve an $N_\psi \times N_\psi$ matrix equation as formulated in (5.36) so that $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ can be derived using (5.34)-(5.39). Because both small matrices that need to be solved are much smaller than the $N_J^{(k+1,c_1)} \times N_J^{(k+1,c_1)}$ matrix $\mathbf{M}^{(k+1,c_1)}$, i.e., $\zeta < N_J^{(k+1,c_1)}$ and $N_\psi < N_J^{(k+1,c_1)}$, and even the sum of the sizes of these two small matrices

is not greater than the size of the matrix $\mathbf{M}^{(k+1,c_1)}$, i.e., $\zeta + N_\psi \leq N_J^{(k+1,c_1)}$, the proposed formulations are less time-consuming compared to directly inverting the $N_J^{(k+1,c_1)} \times N_J^{(k+1,c_1)}$ matrix $\mathbf{M}^{(k+1,c_1)}$.

The above derivation from (5.29) to (5.39) was performed under the assumption of $\tilde{\phi} = \phi^{(k,m_1)}$. In general, this derivation can be performed using either $\tilde{\phi} = \phi^{(k,m_1)}$ or $\tilde{\phi} = \phi^{(k,m_2)}$. In other words, the $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ can be derived using the information from either $[\mathbf{M}^{(k,m_1)}]^{-1}$ of parent 1 or $[\mathbf{M}^{(k,m_2)}]^{-1}$ of parent 2. In order to determine which parent should be chosen as $\tilde{\phi}$ in the proposed method, we define W_i as

$$W_i = \max(\zeta_i, N_\psi^i), i = 1, 2. \quad (5.40)$$

where N_ψ^i is the number of elements in J_ψ^i formulated in (5.28), and ζ_i is derived similarly as (5.32), i.e.,

$$\zeta_i = \min(N_\alpha^i, N_\beta^i), i = 1, 2. \quad (5.41)$$

Note that N_ψ^i is the size of the matrix equation formulated in (5.36), and ζ_i is the size of the matrix equation solved to extract $\tilde{\mathbf{M}}_{\alpha\alpha}^{-1}$ as formulated in (5.29)-(5.32). Then the decision whether to choose parent 1 or parent 2 to perform the computations of (5.29) - (5.39) is

$$\tilde{\phi} = \begin{cases} \phi^{(k,m_1)}, & \text{if } W_1 \leq W_2 \\ \phi^{(k,m_2)}, & \text{if } W_1 > W_2 \end{cases} \quad (5.42)$$

Using (5.42), one of the two parent EM structures is automatically selected such that the computational cost of deriving $[\mathbf{M}^{(k+1,c_1)}]^{-1}$ for the child EM structure is less. The selected parent $\tilde{\phi}$ is used to provide information in the computations of

(5.29) - (5.39).

So far, the proposed method has been demonstrated using the computation of $[\mathbf{M}^{(k+1,c_1)}]^{-1}$, i.e., the inverse matrix of the first child EM structure. The process of calculating the inverse matrices of both children EM structures using proposed method is similar. To compute $[\mathbf{M}^{(k+1,c_2)}]^{-1}$, i.e., the inverse matrix of the second child EM structure, we repeat the calculations of (5.23) - (5.42) for child 2. With the proposed formulations, we can efficiently derive the $[\mathbf{M}^{(k+1,c_i)}]^{-1}$, $i = 1, 2$ for the new EM structures in the $(k + 1)$ th generation by using the information extracted from the parent EM structures of the k th generation. By substituting the $[\mathbf{M}^{(k+1,c_i)}]^{-1}$ matrix into (5.17), we can obtain $\mathbf{u}_\eta^{(k+1,c_i)}$ for each child EM structure in the new generation efficiently. Subsequently, the field solutions of each new EM structure in the new generation can be computed efficiently and the EM topology optimization process can be accelerated.

Here, we summarize the contributions of the proposed EM topology optimization technique as described in Sections 5.2.1-5.2.3. The major contribution is the new formulas proposed to integrate the MPVL iterative formulations and Householder formula. Due to the proposed integration, the effort of solving the large FEM problem at many frequencies is reduced to the effort of solving only a small matrix problem at a single frequency point. Therefore, the proposed method integrating MPVL and Householder formula can speed up the EM topology optimization process beyond that achievable by the existing MPVL or matrix partitioning methods. A further contribution is the new formulations derived to exploit the common factors between the generated children EM structures and the parent EM structures in each

GA generation. Using these new formulations, the small matrix problem is reduced into an even smaller one, resulting in an additional reduction of the computational expense of the EM topology optimization process.

5.2.4 The Process of the Proposed EM Topology Optimization

The process of the proposed EM topology optimization technique incorporating the advanced FEM-MPVL algorithm and GA is summarized as follows.

Step 1) Compute the vector \mathbf{R} and matrices \mathbf{K}_0^s , \mathbf{K}_1^s , and \mathbf{K}_2^s for the base EM structure without metal in the design space, i.e., $\phi_\tau = 0, \tau = 1, 2, \dots, N_v$. Select a suitable frequency expansion point where $\gamma = \gamma_0$.

Step 2) Calculate \mathbf{K}_s using (5.3). Perform LU decomposition of \mathbf{K}_s to obtain its LU factors.

Step 3) Solve the matrix equation formulated in (5.18) to obtain $\tilde{\mathbf{u}}_0$ by performing F/B substitution.

Step 4) Formulate $\tilde{\mathbf{Q}}$ for the EM structure with all cells in the design space covered with metal, i.e., $\phi_\tau = 1, \tau = 1, 2, \dots, N_v$, according to (5.21). Solve the matrix equation formulated in (5.22) to obtain $\tilde{\mathbf{L}}$.

Step 5) Initialize the generation counter of GA as $k = 1$. Initialize the GA by generating N_p chromosomes, i.e., $\phi^{(k,m)}, m = 1, 2, \dots, N_p$, as the first generation. Set the maximum number of generations of GA as N_{GA} .

Step 6) For the m th chromosome in the k th generation, i.e., $\phi^{(k,m)}$, we obtain the matrices $\mathbf{Q}^{(k,m)}$ and $\mathbf{L}^{(k,m)}$ by selecting corresponding columns from $\tilde{\mathbf{Q}}$ and

- $\tilde{\mathbf{L}}$, respectively. The matrix $\mathbf{M}^{(k,m)}$ is then obtained by selecting rows from $\mathbf{L}^{(k,m)}$. Note that this step (Step 6) should be performed for $m = 1, 2, \dots, N_p$ if $k = 1$; else if $k \neq 1$, this step should be performed for $m = c_1, c_2$, where c_1 and c_2 are the indices of two worst chromosomes in the previous generation.
- Step 7) If $k = 1$, compute $[\mathbf{M}^{(k,m)}]^{-1}, m = 1, 2, \dots, N_p$, directly; else, update $[\mathbf{M}^{(k,m)}]^{-1}, m = c_1, c_2$, using the information of the parent chromosomes from the previous generation according to the proposed formulations in Section 5.2.3.
- Step 8) Substitute $[\mathbf{M}^{(k,m)}]^{-1}$ into the proposed algorithm integrating MPVL and Householder formula (i.e., Algorithm 1) in Section 5.2.2 to compute $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ at the frequency expansion point. Then obtain the field solutions $\mathbf{e}^{(k,m)}$ at different frequencies using the information from (5.19) for different values of γ . This step (Step 8) should be performed for $m = 1, 2, \dots, N_p$ if $k = 1$; else $m = c_1, c_2$.
- Step 9) Calculate S-parameters (i.e., $\mathbf{S}(\boldsymbol{\phi}^{(k,m)})$) and the corresponding objective function $U(\mathbf{S}(\boldsymbol{\phi}^{(k,m)}))$ based on the calculated field solutions $\mathbf{e}^{(k,m)}$. This step (Step 9) should be performed for $m = 1, 2, \dots, N_p$ if $k = 1$; else $m = c_1, c_2$.
- Step 10) If no chromosomes' responses in the k th generation can satisfy the design specification and $k < N_{GA}$, go to Step 11). Otherwise, stop the optimization process.
- Step 11) In the population of k th generation, using the GA process to select two parents (i.e., m_1 and m_2), identify two worst chromosomes (i.e., c_1 and c_2), and

perform GA operations such as crossover to generate two new children chromosomes. Set $\phi^{(k+1,m)} = \phi^{(k,m)}$, $m = 1, 2, \dots, N_p$. Update the chromosomes $\phi^{(k+1,m)}$, $m = c_1, c_2$, by replacing them with the new children chromosomes. Update the GA generation counter $k = k + 1$. Go to Step 6).

Fig. 5.4 shows the flowchart of the proposed EM topology optimization technique.

5.2.5 Discussion of Relations Between the Proposed Technique and Other Optimization Techniques

In this chapter, the primary goal is to speed up the repetitive evaluations of EM responses for a large number of new EM structures generated during optimization, and therefore to speed up the EM topology optimization process. Compared to existing numerical methods for such a purpose, including the MPVL and the matrix partitioning method, which can be used to accelerate solving the new EM structures during the EM topology optimization, the proposed method integrating MPVL and Householder formula can achieve a higher speedup. It is the first time that MPVL and Householder formula (similar concept as the matrix partitioning method) are integrated simultaneously for EM topology optimization to achieve a multiplicative speedup. Compared to the existing MPVL, the proposed integration of MPVL and Householder formula can achieve an additional speedup by further reducing the large FEM matrix at one frequency point into a small matrix problem at one frequency point for each new EM structure. Compared to the existing matrix partitioning method, the proposed integration can achieve an additional speedup by further

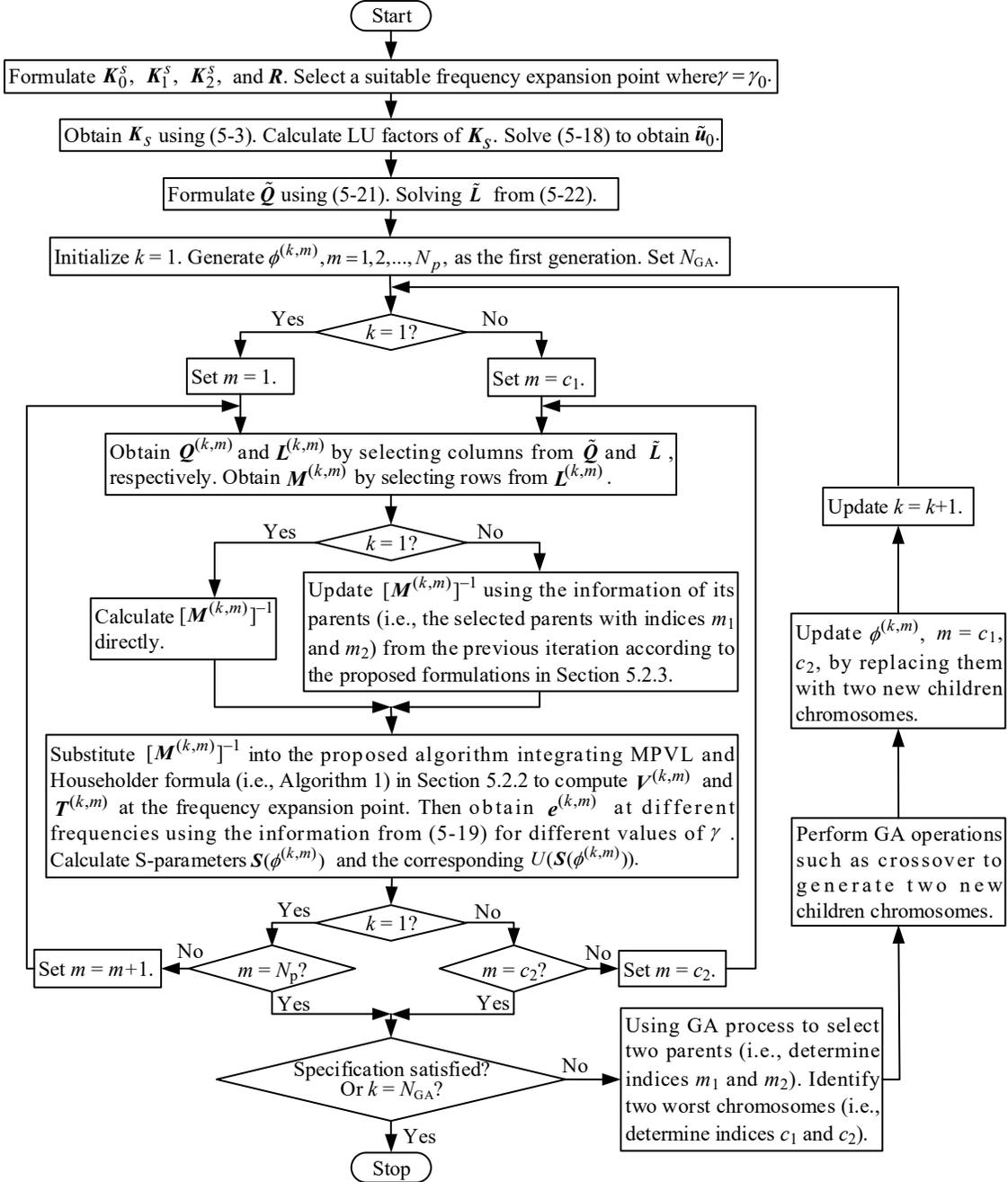


Figure 5.4: The flowchart of the proposed EM topology optimization technique.

reducing the effort of solving the small matrix problem at many frequencies to the effort of solving the small matrix problem at only a single frequency point.

The proposed method is used with GA because GA is one of the major optimization algorithms used in the literature for EM topology optimization. This chapter is not intended to improve the search ability (i.e., find the optimal solution in fewer generations) or the convergence property (including the ability to avoid the stagnation problem) of GA. The GA-based EM topology optimization using our proposed formulas finds the same optimal solutions using the same number of generations as that of the GA without our proposed formulas. The difference is that the proposed method can obtain the same optimization solution in shorter CPU time, because of the proposed speedup algorithm.

There are other optimization algorithms used in the literature for EM topology optimization, such as particle swarm optimization (PSO) [84], [97] and the gradient-based optimization [83]. The comparisons between GA and PSO have been investigated by many researchers in the existing literature [176]-[180]. These comparisons reveal that both GA and PSO have pros and cons, but both are useful. The comparison between GA and the gradient-based optimization has also been investigated in the existing literature [92]. The comparison reveals that the gradient-based optimization converges fast but it is a local approach and usually obtains local optimum, while GA attempts to obtain global optimum through more iterations. The purpose of this chapter is not to advocate GA over other optimization algorithms, but to propose a fast evaluation method to speed up the EM topology optimization under the GA formulation. The existing optimization algo-

rithms such as PSO and the gradient-based algorithms aim to search for a better new EM structure through an iterative updating process, while the proposed integration of MPVL and Householder formula aims to speed up solving the new EM structures produced during the searching process. Since the existing optimization algorithms (e.g., PSO, gradient-based algorithms) and the proposed speedup algorithm are dedicated to two different aspects of EM topology optimization, they are not exclusive to each other (i.e., they will not conflict with each other). Therefore, there are possibilities that the proposed method can be potentially applied under the formulation of other optimization algorithms to speed up the EM topology optimization. In the following discussion, we explore the relationship of the proposed method with the formulation of other optimization algorithms.

We first discuss the relationship of the proposed method with other gradient-free optimization algorithms such as PSO [181] and the branch and bound (BB) algorithm [182]. When using gradient-free optimization algorithms other than GA for EM topology optimization, we still need to solve the EM responses of different EM structures during the optimization process. The proposed method can be used to speed up solving the EM structures during the optimization with other gradient-free optimization algorithms and therefore speed up the optimization process. The reasons are as follows. The proposed method integrating MPVL and Householder formula can be used to efficiently solve the EM structures through reducing the effort of solving the large FEM matrix equation at many frequencies to the effort of solving only a small matrix problem at a single frequency point. This will lead to a major computational speedup because of the proposed method. How to explore

additional speedup by efficiently exploiting the patterns between old and new EM structures in consecutive swarms (PSO) or subsequent branches (BB) will be an interesting direction of future research.

In our preliminary investigation, we apply the proposed method to PSO [181] for an EM topology optimization problem. In the EM topology optimization using PSO, the proposed method that integrates the MPVL and Householder formula is used to speed up the PSO optimization process. The PSO using the proposed method shows a speedup of more than twice over the PSO with standard EM fast frequency sweep in our sample of preliminary investigation. This reveals the potential that the proposed method can be possibly applied in other gradient-free optimization algorithms such as PSO to provide speedup of EM topology optimization. In this sample of preliminary investigation, the search efficiency of PSO and GA is compared. In the observation of this experiment, PSO shows a better search efficiency in the earlier search stages, while GA shows better solutions in the later search stages. In general, as revealed by [94]-[97], both GA and PSO should be useful for EM topology optimization.

Second, we discuss the factors for the potential relationship of the proposed method with gradient-based optimization method. The standard gradient-based optimization method is an efficient method for optimization problems with continuous design variables, such as the classical EM geometry optimization. However, it is not directly applicable to the EM topology optimization of this chapter where the topology design variables are discrete. Some researchers have investigated a modified gradient approach, called topology-gradient based approach for

EM topology optimization with discrete variables, e.g., [83]. However, the powerful gradient-based optimization methods such as quasi-Newton optimization, and Levenberg-Marquardt optimization cannot be used in such a gradient-based topology optimization. In our preliminary investigation using the gradient-based method following [83], the value of the objective function decreases rapidly but it gets stuck in a local minima after several iterations of discrete optimization. During the same number of iterations, the gradient-based method can reduce the objective function more than the GA method. However, the gradient-based optimization is stuck in the local optimal solution obtained, while the GA method can continue to find a better solution after many more generations. For each new EM structure generated during the gradient-based optimization, the proposed method integrating both Householder formula and MPVL can be used to speed up the calculation of EM responses. How to apply the proposed method to accelerate the calculation of the gradients and thus accelerate the gradient-based topology optimization is a possible direction for future research.

5.3 Application Examples

5.3.1 EM Topology Optimization of a Waveguide Filter

In this example, we use the proposed EM topology optimization technique to optimize a waveguide filter. The configuration of the waveguide filter in this example is shown in Fig. 5.5. It is a standard WR-75 waveguide. The width and the height of the cross section of the waveguide are $a = 19.05$ mm and $b = 9.525$ mm, respectively. The design space, i.e., the area where the material can change between metal and

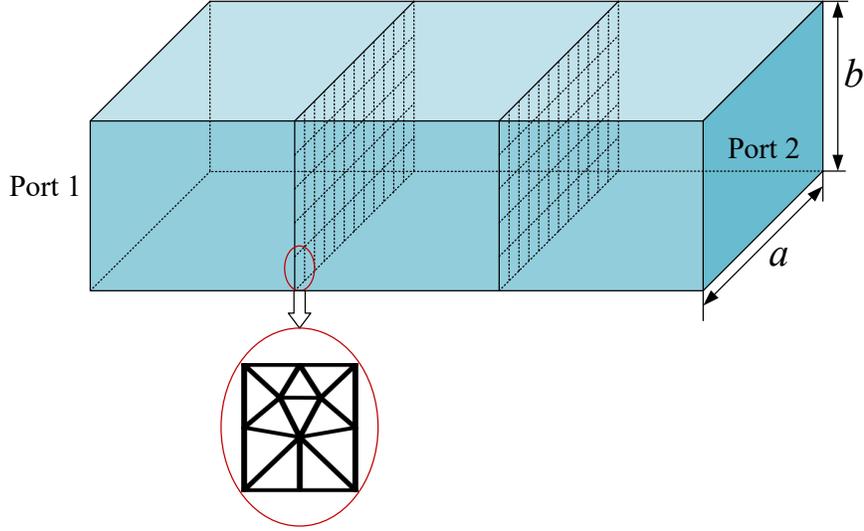


Figure 5.5: The configuration of the waveguide filter in the first example. The design space includes the two cross-sectional planes inside the waveguide. Each plane is divided into 60 cells. Each cell consists of several triangular meshes.

nonmetal during the optimization process, includes two cross-sectional planes inside the waveguide for this filter. Each plane is divided into 60 cells as shown in Fig. 5.5. Two planes give 120 cells in total. The material in each cell is defined as a variable. There are 120 variables in the vector ϕ , i.e., $N_v = 120$, for this example. Then we use Gmsh to generate tetrahedron meshes for the filter structure [174]. After mesh generation, each cell in the design space consists of several triangular meshes as shown in Fig. 5.5. The design specifications of this example are $|S_{11}| \leq -20$ dB in the frequency range of 8.07 GHz - 8.12 GHz and $|S_{11}| \geq -3$ dB in the frequency range of 7.42 GHz - 7.76 GHz and 8.43 GHz - 8.76 GHz.

We use the proposed EM topology optimization technique to optimize the waveguide filter. The purpose is to find the optimal shape and topology of metal in the

design space so that the EM response of the filter can satisfy the given specifications. For this example, the population size in GA is set as $N_p = 100$. At the beginning of GA optimization, we generate 100 binary vectors $\phi^{(1,m)}$, $m = 1, 2, \dots, 100$, with different values as the initial values for chromosomes in the first generation. The response of the best chromosome in the first generation is considered as the initial response of the optimization, which is shown in Fig. 5.6. After optimization using the proposed technique, the optimal shape and topology of metal in the design space are shown in Fig. 5.7. Blue cells in the design space are covered with metal. The S_{11} of the filter after topology optimization is also shown in Fig. 5.6. From Fig. 5.6, we can see that the design specifications of this example are satisfied.

To illustrate the benefit of the proposed technique, we compare the CPU time of the proposed topology optimization and several other methods, including the topology optimization using EM fast frequency sweep (i.e., with MPVL but without Householder formula) [149], the topology optimization using matrix reduction method for EM evaluation such as matrix partitioning method [91], and the topology optimization using EM discrete frequency sweep (i.e., with neither MPVL nor Householder formula). The comparison results are summarized in Table 5.1.

For this filter example, the size of the FEM matrix is $N = 101228$; the number of unknowns in the design space is $N_{\tilde{H}} = 7336$. The frequency samples are 201 frequency points in the range from 7.42 GHz to 8.76 GHz; the frequency expansion point is chosen as 8.09 GHz, which is the center frequency in the frequency range of interest. Note that the frequency expansion point is a user-defined parameter. A valid frequency expansion point should satisfy the condition that the original

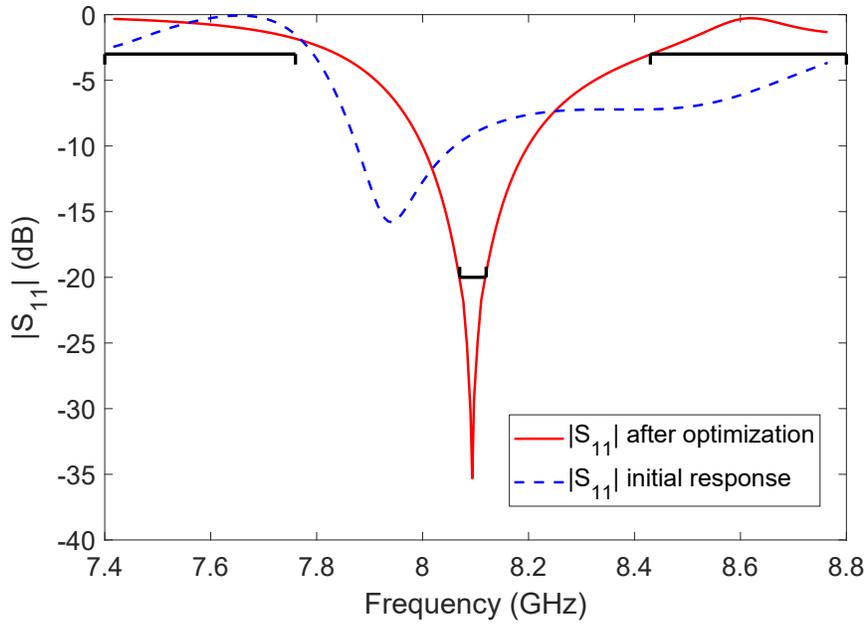


Figure 5.6: The comparison of S_{11} obtained after EM topology optimization and the initial S_{11} response for the filter in the first example. Note that the initial S_{11} response is the response of the best chromosome in the first generation. The black lines represent the specifications.

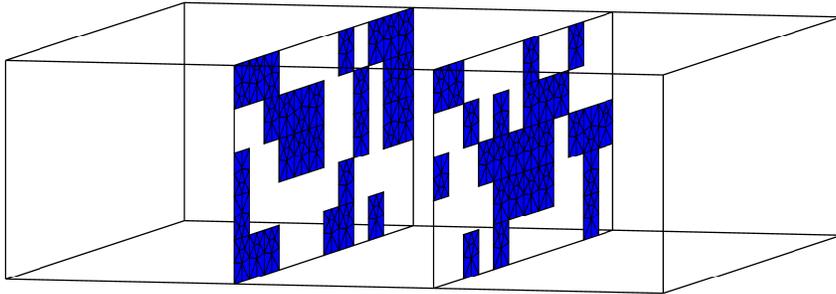


Figure 5.7: The optimal EM structure of the waveguide filter after EM topology optimization in the first example.

matrix in the MPVL algorithm is nonsingular at the frequency expansion point [175]. This condition is normally satisfied in the FEM for EM problems. The order

Table 5.1: Comparison of CPU Time Between the Proposed EM Topology Optimization Technique and Several Other Topology Optimization Methods for the Waveguide Filter in the First Example

Topology optimization methods		Major computational aspects		Average CPU time	Total No. of children EM structures	Total CPU time of the entire optimization
		Matrix size in the matrix equation to be solved	No. of matrix equations to be solved			
The proposed EM topology optimization technique	Initialization	101228×101228	1	65.45 min	-	110.6 h
		$N_J^{(1,m)} \times N_J^{(1,m)}$ ($N_J^{(1,m)} \leq 7336$)	100			
	FEM per child	$\zeta \times \zeta$ ($\zeta \leq 3668$)	1	12.21 s	32300	
	EM structure	$N_\psi \times N_\psi$ ($N_\psi \leq 3668$)	1			
Optimization using EM fast frequency sweep	Initialization	-	0	0 s	-	347.9 h ‡
	FEM per child EM structure	101228×101228	1	38.78 s	32300	
Optimization using EM matrix partitioning method	Initialization	93892×93892	201	118.3 h	-	17792 h †
	FEM per child EM structure	$N_J^{(k,m)} \times N_J^{(k,m)}$ ($N_J^{(k,m)} \leq 7336$)	201	1969.8 s	32300	
Optimization using EM discrete frequency sweep	Initialization	-	0	0 s	-	28727 h ‡
	FEM per child EM structure	101228×101228	201	3201.8 s	32300	

† When using the matrix partitioning method, the matrix size in the initialization is equal to the size of the FEM matrix ($N = 101228$) subtracting the number of unknowns in the design space (7336), i.e., $93892 = 101228 - 7336$. The EM solution for each new EM structure is obtained by solving the $N_J^{(k,m)} \times N_J^{(k,m)}$ matrix equation at different frequency points. Note that $N_J^{(k,m)} \leq 7336$, and on average, $N_J^{(k,m)} = 3668$. The total CPU time for the entire optimization is the estimated CPU time based on the information of actual CPU time from 100 generations of the GA optimization.

‡ For EM topology optimization using EM fast frequency sweep and that using EM discrete frequency sweep, the total CPU time for the entire optimization is the projected CPU time based on the information of actual CPU time from 100 generations of the GA optimization.

ρ of the reduced-order model is determined by performing a convergence analysis of the EM responses. The convergence analysis proceeds this way: we start from a small ρ and increase the value of ρ one by one until the error of the responses between the ρ th-order model and the $(\rho + 1)$ th-order model is less than a user-

defined threshold, i.e., the EM responses obtained from the reduced-order model converge. After the convergence analysis, the order of the reduced-order model for this example is determined as $\rho = 8$, because the responses of 8th-order model and the 9th-order model are almost the same.

As shown in Table 5.1, the CPU time for initialization of the proposed method is 65.45 minutes. The initialization includes one LU decomposition of a 101228×101228 matrix (\mathbf{K}_s), one F/B substitution to calculate $\tilde{\mathbf{u}}_0$, 7336 F/B substitutions to calculate the 101228×7336 matrix $\tilde{\mathbf{L}}$, and calculations of the $N_J^{(1,m)} \times N_J^{(1,m)}$ matrix $[\mathbf{M}^{(1,m)}]^{-1}$, $m = 1, 2, \dots, 100$, for EM structures in the first generation. Note that $N_J^{(1,m)}$ is the number of unknowns in the metal cells of the m th chromosome in the first generation. The value of $N_J^{(1,m)}$ is not greater than the value of N_J , i.e., $N_J^{(1,m)} \leq 7336$, and on average $N_J^{(1,m)} = 3668$. Using the proposed method integrating MPVL and Householder formula, the computational cost of solving the 101228×101228 FEM matrix equation at 201 frequencies is reduced to the cost of solving (on average) a 3668×3668 matrix equation with $[\mathbf{M}^{(1,m)}]$ at a single frequency point of 8.09 GHz (the frequency expansion point), in order to obtain the FEM solution for each chromosome in the first generation. The initialization is performed only once in the entire optimization process. After initialization, in each generation, we solve FEM equations of two new children EM structures using the proposed method to obtain the EM solutions. For each new child EM structure, the 101228×101228 FEM matrix problem at 201 frequencies is reduced into (on average) a 3668×3668 matrix problem with $\mathbf{M}^{(k,m)}$ at a single frequency point of 8.09 GHz, due to the proposed integration of MPVL and Householder formula.

By extracting the inheritance pattern between the new child chromosome (120-bit binary vector) and its parent chromosomes (two 120-bit binary vectors selected from the previous generation), we further reduce the (on average) 3668×3668 matrix problem into solving a $\zeta \times \zeta$ matrix equation and an $N_\psi \times N_\psi$ matrix equation. The matrix sizes ζ and N_ψ vary according to different inheritance patterns of different parents. The values of ζ and N_ψ are not greater than half of N_j , i.e., $\zeta \leq 3668$ and $N_\psi \leq 3668$, and on average $\zeta = 1834$ and $N_\psi = 1834$. Therefore, using the proposed method to calculate the FEM solutions of new children EM structures, the computational cost of solving the 101228×101228 FEM matrix equations at 201 frequency points is reduced to the cost of solving matrix equations whose average matrix size is 1834×1834 at a single frequency point of 8.09 GHz. The total number of generations in GA optimization for this example is 16150. In the EM topology optimization process, we only need to solve the 101228×101228 FEM matrix once. The EM solutions of the 32300 (i.e., 2×16150) new EM structures generated by GA during optimization are obtained by solving small matrix equations at 8.09 GHz for 32300 times. The CPU time for calculating EM solution of each new EM structure using the proposed method is 12.21 seconds. This time is an average time because the matrix sizes ζ and N_ψ vary for different EM structures based on the inheritance pattern. The results of several methods shown in Table 5.1 demonstrate that the total CPU time of the proposed technique is less than the CPU time of other methods. The proposed method provides more than three times speedup over the topology optimization using EM fast frequency sweep. Compared to other methods in Table 5.1, the speedup ratio of the proposed EM topology optimization

method is even larger. In addition, the speedup ratio will increase as the increase of the number of generations.

In this example, we fill the waveguide with dielectric material to support the positions of metal pieces. FEM equations used to compute the EM solutions during the optimization were formulated including the dielectric material. The examples in this chapter are mainly used to demonstrate the computational advantages of the proposed numerical technique. The practical implementation of the device is left as a separate topic of future research.

5.3.2 EM Topology Optimization of a Three-Pole Waveguide Filter

The proposed EM topology optimization technique is applied to the optimization of a three-pole waveguide filter in this example. The configuration of the waveguide filter is shown in Fig. 5.8. The width and the height of the cross section of the waveguide are $a = 19.05$ mm and $b = 9.525$ mm, respectively (WR-75). Four cross-sectional planes inside the waveguide structure are the design space, i.e., the area where the material can change between metal and nonmetal during the optimization process. Each plane includes 60 cells as shown in Fig. 5.8. Four planes give 240 cells in total. The material in each cell is defined as a variable. The total number of variables in the vector ϕ for this filter is $N_v = 240$. After generating tetrahedron meshes for the filter structure by Gmsh [174], each cell in the design space consists of several triangular meshes as shown in Fig. 5.8. The design specifications of the three-pole waveguide filter are given by $|S_{11}| \leq -20$ dB in 7.82 GHz - 8.21 GHz and

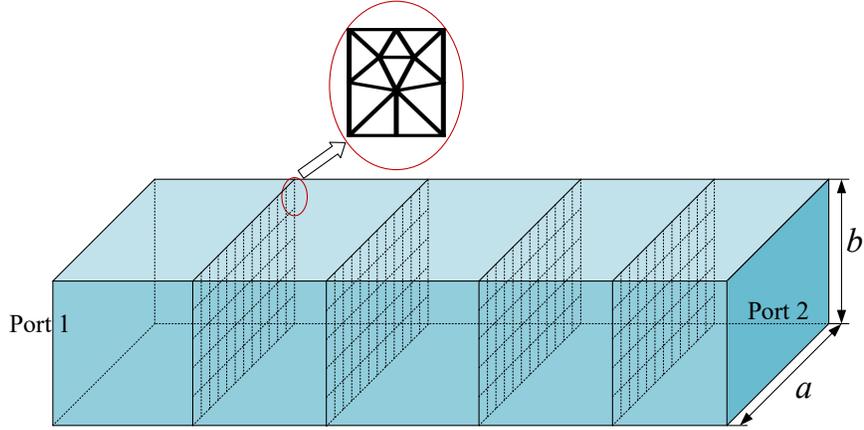


Figure 5.8: The configuration of the three-pole waveguide filter. The design space includes the four cross-sectional planes inside the waveguide. Each plane is divided into 60 cells. Each cell consists of several triangular meshes.

$$|S_{11}| \geq -3 \text{ dB in } 7.42 \text{ GHz} - 7.73 \text{ GHz and } 8.34 \text{ GHz} - 8.76 \text{ GHz.}$$

The proposed EM topology optimization technique is applied to the optimization of the three-pole waveguide filter. The purpose is to find the optimal shape and topology of metal in the design space so that the EM response of the filter can satisfy the given specifications. The population size in GA is set as $N_p = 120$. At the beginning of GA optimization, 120 binary vectors $\phi^{(1,m)}$, $m = 1, 2, \dots, 120$, with different values are generated as the initial values for chromosomes in the first generation. The response of the best chromosome in the first generation is considered as the initial response of the optimization, which is shown in Fig. 5.9. After EM topology optimization, the optimal shape and topology of metal in the design space is illustrated in Fig. 5.10. Blue cells in the design space are covered with metals. The S_{11} of the filter after topology optimization is also shown in Fig. 5.9, which demonstrates that the design specifications of the three-pole filter

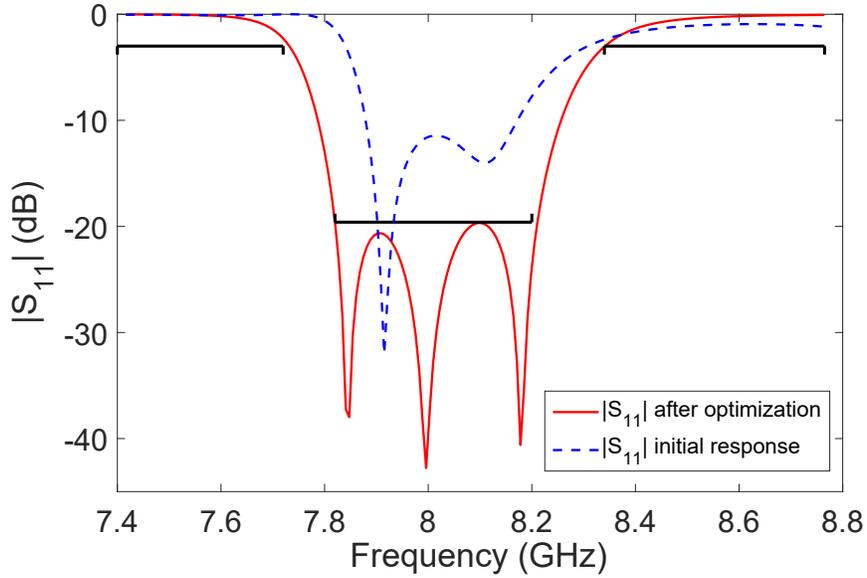


Figure 5.9: The comparison of S_{11} obtained after EM topology optimization and the initial S_{11} response for the three-pole filter. Note that the initial S_{11} response is the response of the best chromosome in the first generation. The black lines represent the specifications.

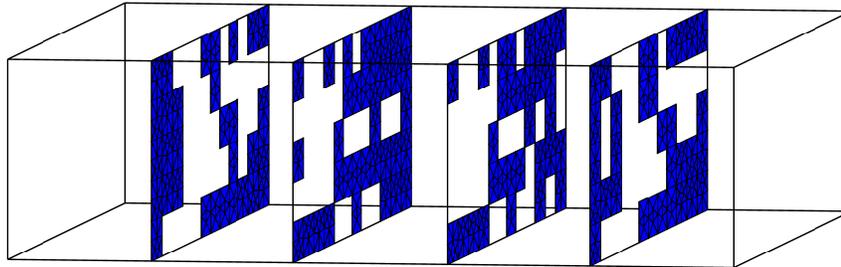


Figure 5.10: The optimal EM structure of the three-pole waveguide filter after EM topology optimization. Blue cells are covered with metals.

are satisfied. Note that the waveguide in this example is also filled with dielectric material to support the positions of metal pieces.

The comparison of the proposed EM topology optimization and several other methods is summarized in Table 5.2 to demonstrate the benefit of the proposed

Table 5.2: Comparison of CPU Time Between the Proposed Topology Optimization Technique and Several Other Topology Optimization Methods for the Three-Pole Waveguide Filter Example

Topology optimization methods		Major computational aspects		Average CPU time	Total No. of children EM structures	Total CPU time of the entire optimization
		Matrix size in the matrix equation to be solved	No. of matrix equations to be solved			
The proposed EM topology optimization technique	Initialization	241376×241376	1	4.04 h	-	247.4 h
		$N_J^{(1,m)} \times N_J^{(1,m)}$ ($N_J^{(1,m)} \leq 14672$)	120			
	FEM per child	$\zeta \times \zeta$ ($\zeta \leq 7336$)	1	49.56 s	17678	
	EM structure	$N_\psi \times N_\psi$ ($N_\psi \leq 7336$)	1			
Optimization using EM fast frequency sweep	Initialization	-	0	0 s	-	503.4 h [‡]
	FEM per child EM structure	241376×241376	1	102.51 s	17678	
Optimization using EM matrix partitioning method	Initialization	226704×226704	201	369.5 h	-	46154 h [†]
	FEM per child	$N_J^{(k,m)} \times N_J^{(k,m)}$	201	9323.7 s	17678	
	EM structure	($N_J^{(k,m)} \leq 14672$)				
Optimization using EM discrete frequency sweep	Initialization	-	0	0 s	-	55209 h [‡]
	FEM per child EM structure	241376×241376	201	11243 s	17678	

[†] When using the matrix partitioning method, the matrix size in the initialization is equal to the size of the FEM matrix ($N = 241376$) subtracting the number of unknowns in the design space (14672), i.e., $226704 = 241376 - 14672$. The EM solution for each new EM structure is obtained by solving the $N_J^{(k,m)} \times N_J^{(k,m)}$ matrix equation at different frequency points. Note that $N_J^{(k,m)} \leq 14672$, and on average $N_J^{(k,m)} = 7336$.

^{† ‡} The total CPU time of the entire optimization for other methods besides the proposed topology optimization technique is obtained using the same method as described in the footnote of Table 5.1.

topology optimization technique. For this example, the size of the FEM matrix is $N = 241376$; the number of unknowns in the design space is $N_{\tilde{H}} = 14672$. The frequency samples are 201 frequency points in the range from 7.42 GHz to 8.76 GHz; the frequency expansion point is chosen as 8.09 GHz, which is the center frequency in the frequency range of interest. The order of the reduced-order model

using MPVL is $\rho = 12$, which is determined by performing a convergence analysis of the EM responses. As shown in Table 5.2, the CPU time for initialization of the proposed method is 4.04 hours. The initialization includes one LU decomposition of a 241376×241376 matrix (\mathbf{K}_s), one F/B substitution to calculate $\tilde{\mathbf{u}}_0$, 14672 F/B substitutions to calculate the 241376×14672 matrix $\tilde{\mathbf{L}}$, and calculations of the $N_J^{(1,m)} \times N_J^{(1,m)}$ matrix $[\mathbf{M}^{(1,m)}]^{-1}$, $m = 1, 2, \dots, 120$, for EM structures in the first generation. Note that $N_J^{(1,m)}$ is the number of unknowns in the metal cells of the m th chromosome in the first generation. The value of $N_J^{(1,m)}$ is not greater than the value of N_J , i.e., $N_J^{(1,m)} \leq 14672$, and on average $N_J^{(1,m)} = 7336$. Using the proposed method integrating MPVL and Householder formula, the computational cost of solving the 241376×241376 FEM matrix equation at 201 frequencies is reduced to the cost of solving (on average) a 7336×7336 matrix equation with $[\mathbf{M}^{(1,m)}]$ at a single frequency point of 8.09 GHz (the frequency expansion point), in order to obtain the FEM solution for each chromosome in the first generation. The initialization is performed only once in the entire optimization process. After initialization, in each generation, we solve FEM equations of two new children EM structures using the proposed method to obtain the EM solutions. For each new child EM structure, the 241376×241376 FEM matrix problem at 201 frequencies is reduced into (on average) a 7336×7336 matrix problem with $\mathbf{M}^{(k,m)}$ at a single frequency point of 8.09 GHz, due to the proposed integration of MPVL and Householder formula. By extracting the inheritance pattern between the new child chromosome (240-bit binary vector) and its parent chromosomes (two 240-bit binary vectors selected from the previous generation), we further reduce the (on average) 7336×7336 matrix

problem into solving a $\zeta \times \zeta$ matrix equation and an $N_\psi \times N_\psi$ matrix equation. The matrix sizes ζ and N_ψ vary according to different inheritance patterns of different parents. The values of ζ and N_ψ are not greater than half of N_j , i.e., $\zeta \leq 7336$ and $N_\psi \leq 7336$, and on average $\zeta = 3668$ and $N_\psi = 3668$. Therefore, using the proposed method to calculate the FEM solutions of new children EM structures, the computational cost of solving the 241376×241376 FEM matrix equations at 201 frequency points is reduced to the cost of solving matrix equations whose average matrix size is 3668×3668 at a single frequency point of 8.09 GHz. The total number of generations in GA optimization for this example is 8839. In the EM topology optimization process, we only need to solve the 241376×241376 FEM matrix once. The EM solutions of the 17678 (i.e., 2×8839) new EM structures generated by GA during optimization are obtained by solving small matrix equations at 8.09 GHz for 17678 times. The CPU time for calculating EM solution of each new EM structure using the proposed method is 49.56 seconds. As shown in Table 5.2, the proposed topology optimization takes the shortest time to find the optimal solution among the several methods compared.

5.3.3 EM Topology Optimization of a T-Junction Waveguide Diplexer

In this example, we consider the topology optimization of a T-junction waveguide diplexer using the proposed technique. We first perform the preliminary optimization to obtain a relatively good starting point for the subsequent EM topology optimization. The preliminary optimization is performed through EM geometry

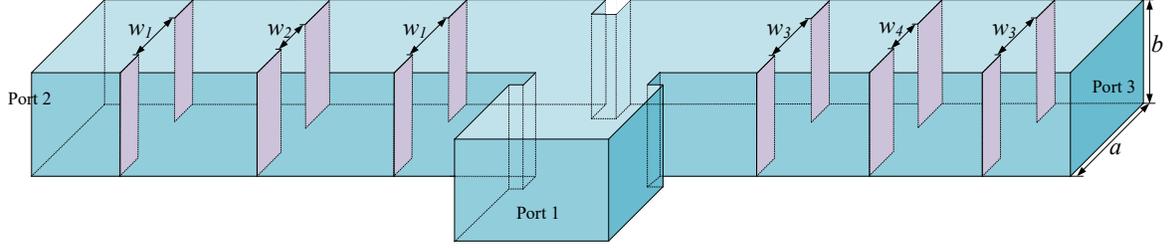


Figure 5.11: The structure of the T-junction diplexer after preliminary optimization. The widths of the coupling windows are $w_1 = 9.11$ mm, $w_2 = 6.06$ mm, $w_3 = 7.79$ mm, and $w_4 = 4.49$ mm.

optimization using HFSS. The design specification of the diplexer is defined as $|S_{11}| \leq -15$ dB in 6.685 GHz - 6.795 GHz and 7.365 GHz - 7.475 GHz. The structure obtained after the preliminary optimization is shown in Fig. 5.11. The response of the diplexer after preliminary optimization is shown as the blue dashed curve in Fig. 5.12.

We then apply the proposed EM topology optimization technique to this diplexer example. The design space for this example includes six cross-sectional planes, i.e., three on the left of the T-junction and three on the right, as shown in Fig. 5.13. Each plane is divided into 60 cells. Six planes give 360 cells in total. The material in each cell is defined as a variable. There are 360 variables in the vector ϕ for this diplexer. The planes are divided properly so that the structure obtained from the preliminary optimization can be represented using a binary vector in the topology optimization. The population size in GA is set as $N_p = 160$. At the beginning of GA optimization, we generate 160 binary vectors $\phi^{(1,m)}$, $m = 1, 2, \dots, 160$, with different values as the initial values for chromosomes in the first generation. Note that the first chromosome in the first generation (i.e., $\phi^{(1,1)}$) represents the structure obtained

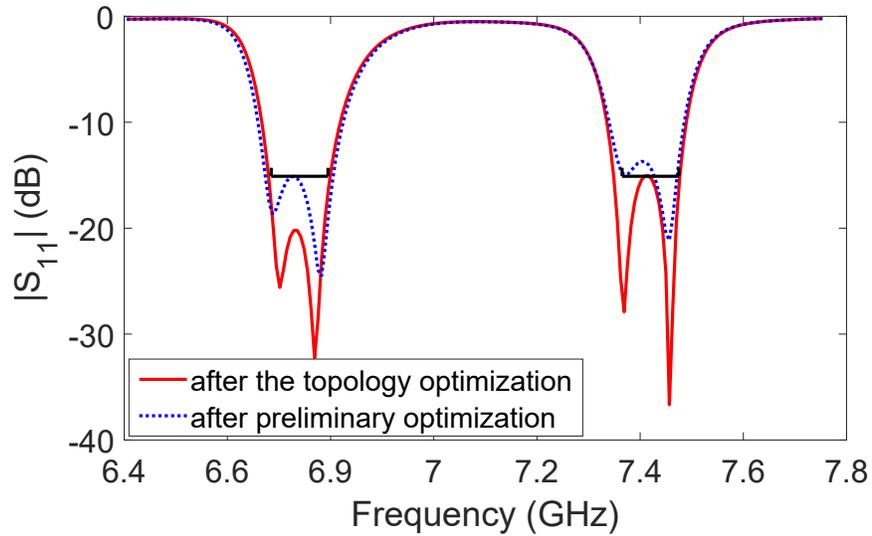


Figure 5.12: The comparison of S_{11} after the preliminary optimization and S_{11} after the proposed EM topology optimization for the T-junction waveguide diplexer example.

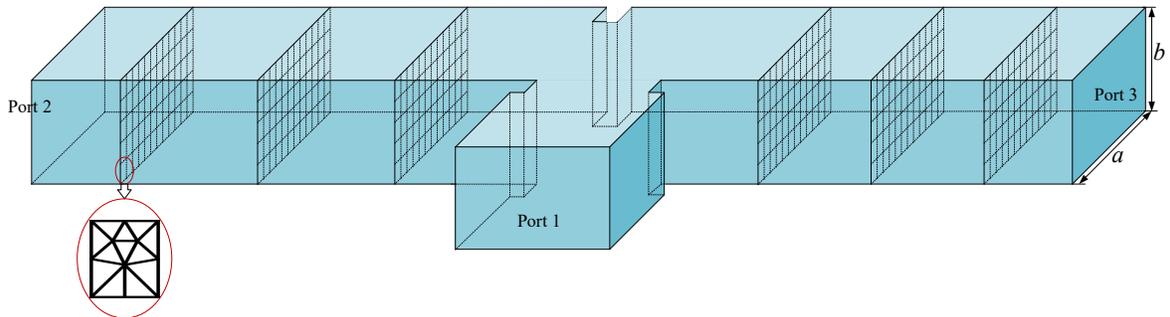


Figure 5.13: The configuration of the T-junction diplexer for EM topology optimization. The design space includes the six cross-sectional planes inside the waveguide. Each plane is divided into 60 cells. Each cell consists of several triangular meshes.

from the preliminary optimization. After the proposed topology optimization, the optimal EM structure of the diplexer is shown in Fig. 5.14. Blue cells in the design space are covered with metals. The S_{11} of the diplexer after the proposed topology optimization is shown as the red solid curve in Fig. 5.12. From Fig. 5.12, we can see

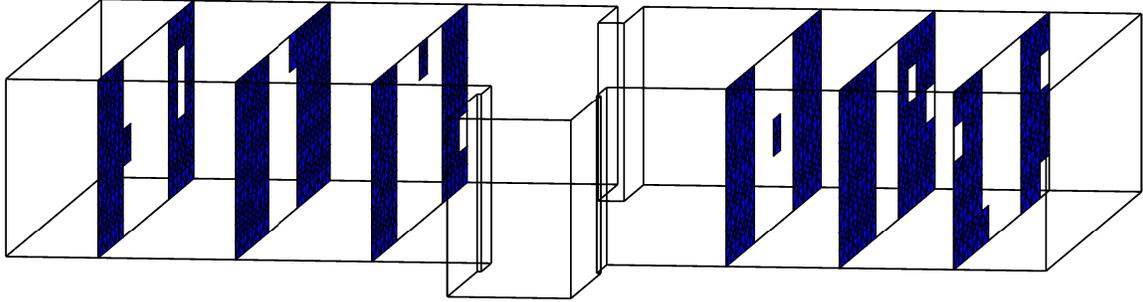


Figure 5.14: The optimal EM structure of the T-junction diplexer after topology optimization. Blue cells are covered with metals.

that the response obtained from the proposed EM topology optimization satisfies the design specification. Note that the waveguide in this example is also filled with dielectric material to support the positions of metal pieces.

The comparison of the CPU time for different methods are summarized in Table 5.3. For this diplexer example, the size of the FEM matrix is $N = 401510$; the number of unknowns in the design space is $N_{\bar{H}} = 20376$. The frequency samples are 201 frequency points in the range from 6.40 GHz to 7.75 GHz. For this diplexer example, because there are two operating frequency bands in the whole frequency range, the selection of the frequency expansion point can be further refined through a refinement process. In the refinement process, we first perform model order reduction using different frequencies as the frequency expansion point. These frequencies are around the center frequency (and include the center frequency) of the whole frequency range of interest. Then, we compare the orders of the reduced-order models derived from different frequency expansion points, and choose the frequency expansion point where the reduced-order model has the lowest order. After the refinement

Table 5.3: Comparison of CPU Time Between the Proposed Topology Optimization Technique and Several Other Topology Optimization Methods for the T-Junction Diplexer Example

Topology optimization methods		Major computational aspects		Average CPU time	Total No. of children EM structures	Total CPU time of the entire optimization
		Matrix size in the matrix equation to be solved	No. of matrix equations to be solved			
The proposed EM topology optimization technique	Initialization	401510×401510	1	8.4 h	-	484.8 h
		$N_J^{(1,m)} \times N_J^{(1,m)}$ ($N_J^{(1,m)} \leq 20376$)	160			
	FEM per child	$\zeta \times \zeta$ ($\zeta \leq 10188$)	1	142.32 s	12050	
	EM structure	$N_{\psi} \times N_{\psi}$ ($N_{\psi} \leq 10188$)	1			
Optimization using EM fast frequency sweep	Initialization	-	0	0 s	-	629.8 h [‡]
FEM per child EM structure	401510×401510	1	188.15 s	12050		
Optimization using EM matrix partitioning method	Initialization	381134×381134	201	442.2 h	-	75458 h [†]
	FEM per child	$N_J^{(k,m)} \times N_J^{(k,m)}$	201	22411.5 s	12050	
	EM structure	($N_J^{(k,m)} \leq 20376$)				
Optimization using EM discrete frequency sweep	Initialization	-	0	0 s	-	85781 h [‡]
FEM per child EM structure	401510×401510	201	25627.5 s	12050		

[†] When using the matrix partitioning method, the matrix size in the initialization is equal to the size of the FEM matrix ($N = 401510$) subtracting the number of unknowns in the design space (20376), i.e., $381134 = 401510 - 20376$. The EM solution for each new EM structure is obtained by solving the $N_J^{(k,m)} \times N_J^{(k,m)}$ matrix equation at different frequency points. Note that $N_J^{(k,m)} \leq 20376$, and on average $N_J^{(k,m)} = 10188$.

^{† ‡} The total CPU time of the entire optimization for other methods besides the proposed topology optimization technique is obtained using the same method as described in the footnote of Table 5.1.

process, the frequency expansion point for this example is chosen as 7.01 GHz, and the order of the reduced-order model is $\rho = 16$.

As shown in Table 5.3, the CPU time for initialization of the proposed method is 8.4 hours. The initialization includes one LU decomposition of a 401510×401510 matrix (\mathbf{K}_s), one F/B substitution to calculate $\tilde{\mathbf{u}}_0$, 20376 F/B substitutions to

calculate the 401510×20376 matrix $\tilde{\mathbf{L}}$, and calculations of the $N_J^{(1,m)} \times N_J^{(1,m)}$ matrix $[\mathbf{M}^{(1,m)}]^{-1}$, $m = 1, 2, \dots, 160$, for EM structures in the first generation. Note that $N_J^{(1,m)}$ is the number of unknowns in the metal cells of the m th chromosome in the first generation. The value of $N_J^{(1,m)}$ is not greater than the value of N_J , i.e., $N_J^{(1,m)} \leq 20376$, and on average $N_J^{(1,m)} = 10188$. Using the proposed method integrating MPVL and Householder formula, the computational cost of solving the 401510×401510 FEM matrix equation at 201 frequencies is reduced to the cost of solving (on average) a 10188×10188 matrix equation with $[\mathbf{M}^{(1,m)}]$ at a single frequency point of 7.01 GHz (the frequency expansion point), in order to obtain the FEM solution for each chromosome in the first generation. The initialization is performed only once in the entire optimization process. After initialization, in each generation, we solve FEM equations of two new children EM structures using the proposed method to obtain the EM solutions. For each new child EM structure, the 401510×401510 FEM matrix problem at 201 frequencies is reduced into (on average) a 10188×10188 matrix problem with $\mathbf{M}^{(k,m)}$ at a single frequency point of 7.01 GHz, due to the proposed integration of MPVL and Householder formula. By extracting the inheritance pattern between the new child chromosome (360-bit binary vector) and its parent chromosomes (two 360-bit binary vectors selected from the previous generation), we further reduce the (on average) 10188×10188 matrix problem into solving a $\zeta \times \zeta$ matrix equation and an $N_\psi \times N_\psi$ matrix equation. The matrix sizes ζ and N_ψ vary according to different inheritance patterns of different parents. The values of ζ and N_ψ are not greater than half of N_J , i.e., $\zeta \leq 10188$ and $N_\psi \leq 10188$, and on average $\zeta = 5094$ and $N_\psi = 5094$. Therefore, using the

proposed method to calculate the FEM solutions of new children EM structures, the computational cost of solving the 401510×401510 FEM matrix equations at 201 frequency points is reduced to the cost of solving matrix equations whose average matrix size is 5094×5094 at a single frequency point of 7.01 GHz. The total number of generations in GA optimization for this example is 6625. In the EM topology optimization process, we only need to solve the 401510×401510 FEM matrix once. The EM solutions of the 12050 (i.e., 2×6625) new EM structures generated by GA during optimization are obtained by solving small matrix equations at 7.01 GHz for 12050 times. The CPU time for calculating EM solution of each new EM structure using the proposed method is 142.32 seconds. As demonstrated in Table 5.3, the proposed topology optimization is faster than several other methods compared.

The three application examples in this section illustrate that the proposed optimization method is a general method that can be applied to different situations with or without a known good starting point. In the situation where a good practical starting point is not available, we generate a population of binary vectors with different values (i.e., different metal/nonmetal patterns) as the initial values for chromosomes in the first generation (such as the first and the second example). In the situation where a good practical starting point is available (such as the third example), we use the good starting point as the first chromosome in the first generation. The other chromosomes in the first generation are initialized by generating a population of binary vectors with different values (i.e., different metal/nonmetal patterns).

5.4 Conclusion

In this chapter, an efficient EM topology optimization technique has been proposed for the design of microwave components. A new method has been proposed to integrate MPVL and Householder formula. Using the proposed method, the effort of solving the large FEM matrix equation at many frequencies has been reduced to the effort of solving only a small matrix problem at a single frequency point, thereby speeding up the topology optimization process. We have further proposed a new method to reduce the small matrix problem into an even smaller one by exploiting the inheritance pattern of GA. New formulations have been derived to exploit the common factors between new children EM structures and the parent EM structures in each GA generation so that computational expense can be further reduced. The proposed technique incorporating the integration of MPVL and Householder formula and the inheritance pattern of GA helps greatly accelerate the EM topology optimization process under GA optimization.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

In this thesis, we have investigated advanced computer-aided-design (CAD) techniques for modeling and optimization of microwave passive components. This thesis has first proposed a hybrid deep neural network modeling technique to solve complex high-dimensional modeling problems of microwave components. In the proposed technique, we have formulated a hybrid structure utilizing both sigmoid functions and ReLUs as activation functions for hidden neurons. A smooth ReLU has been proposed for the hybrid deep neural network model. Compared to the conventional ReLU, the proposed smooth ReLU can improve the smoothness of the model outputs and provide continuity of derivatives. The proposed deep neural network can be trained to learn the training data in high-dimensional space. An advanced three-stage deep learning algorithm has been proposed to train the new deep neural network and to overcome the vanishing gradient problem that would otherwise exist in training the hybrid deep neural networks. As illustrated in the

examples, the proposed technique can solve higher dimensional microwave modeling problems more efficiently over that solved by the shallow neural network. Compared to the pure ReLU deep neural network, the proposed technique can achieve higher accuracy with fewer number of hidden neurons. The developed deep neural network technique is a useful surrogate modeling technique for microwave components.

Based on the investigation of the surrogate modeling technique, this thesis has further studied the efficient surrogate-based EM optimization method and proposed an advanced cognition-driven EM optimization incorporating transfer function-based feature surrogate for microwave filters. The proposed optimization technique has addressed the challenges where the response of the starting point for the design optimization is substantially misaligned with the design specifications and the filter response does not have explicitly identifiable feature information. Multiple feature parameters, including passband feature frequencies, stopband feature frequencies, feature bandwidths, feature ripples, and boundary feature heights, have been extracted from the transfer function response in the proposed technique. New objective functions have been formulated directly in the feature space for the cognition-driven optimization. The proposed cognition-driven optimization incorporating transfer function-based feature surrogate has achieved faster convergence than the existing feature-assisted EM optimization methods.

Furthermore, in order to address the EM topology optimization that our proposed cognition-driven optimization method is not applicable, this thesis has also proposed an efficient EM topology optimization technique for the design of microwave components. A new method has been proposed to integrate MPVL and

Householder formula. Using the proposed method, the effort of solving the large FEM matrix equation at many frequencies has been reduced to the effort of solving only a small matrix problem at a single frequency point, thereby speeding up the topology optimization process. We have further proposed a new method to reduce the small matrix problem into an even smaller one by exploiting the inheritance pattern of GA. New formulations have been derived to exploit the common factors between new children EM structures and the parent EM structures in each GA generation so that computational expense can be further reduced. The proposed technique incorporating the advanced MPVL and the inheritance pattern of GA helps greatly accelerate the EM topology optimization process.

6.2 Future Research

Based on the extension of the modeling and optimization approaches proposed in this thesis, some possible directions for future research are stated as follows

- The deep neural network technique has been studied to solve the high dimensional microwave modeling problem in this thesis. One future research direction is to expand the deep neural network technique to perform microwave component design. Taking advantage of the powerful learning ability of the deep neural network, it is possible to develop a deep neural network-based CAD tool that can automatically produce a set of suitable geometric dimensions when the design specifications are fed in. New formulations need to be studied to obtain suitable parameters from the given design specifications as the deep neural network inputs. The design variables can be used as the

outputs of the deep neural network. New deep learning algorithms need to be developed to effectively train this neural network and address the possible non-uniqueness (multivalued) problem. In this way, the efficiency of microwave component design can be improved.

- The cognition-driven optimization incorporating transfer function-based feature surrogate has been studied for EM-based optimization in this thesis. One future research direction is to expand the proposed cognition-driven optimization method to multiphysics optimization. Multiphysics optimization is required in order to achieve high-performance microwave components. Compared with EM-based optimization, multiphysics optimization is a more challenging task because it involves multiple physics domains such as EM, thermal, and structural mechanics. By introducing the proposed cognition-driven optimization into multiphysics optimization, we can extract feature parameters based on the multiphysics responses. New feature surrogates need to be developed to represent the relationships between the multiphysics design parameters and the extracted feature parameters. Subsequently, the developed feature surrogate can be used to perform multiphysics optimization. Therefore, the multiphysics optimization process can be accelerated.
- The proposed speedup method for EM topology optimization has been developed under the gradient-free GA formulation in this thesis. A future research direction is to expand the proposed speedup method to accelerate the gradient-based EM topology optimization. In the gradient-based EM topol-

ogy optimization, the proposed method integrating both Householder formula and MPVL can be used to speed the calculation of EM responses, but cannot be directly used to speed up the calculation of the gradients. Based on the idea of integrating both Householder formula and MPVL, we can further develop new formulations to speed up the calculation of gradients during the gradient-based EM topology optimization. In this way, the gradient-based EM topology optimization process can be accelerated.

- The computational advantages of the proposed EM topology optimization technique have been demonstrated in this thesis. As a further research, we can expand the proposed EM topology optimization to exploit parallel computations. Currently, the proposed EM topology optimization technique is based on sequential computation, i.e., the EM response of each child EM structure generated by GA is computed one-by-one using the proposed method. To further accelerate the optimization process, we can incorporate parallel computation into the proposed EM topology optimization. For the newly generated chromosomes in each generation, the EM responses can be computed using the proposed method integrating MPVL and Householder formula simultaneously in parallel. Therefore, we can further improve the efficiency of the EM topology optimization process.
- The examples in this thesis are mainly used to demonstrate the computational advantages of the proposed numerical techniques. The practical implementation of the devices is left as a separate topic of future research. The measure-

ment data can be further used to validate the proposed techniques. Another interesting future research topic is to investigate applications of the proposed modeling and optimization techniques for practical design and implementation of microwave components.

Appendix

In this appendix, we describe the formulations of matrix $\mathbf{A}_0^{(k,m)}$ and vector $\mathbf{g}_\eta^{(k,m)}$ in the standard FEM-MPVL. These formulations are to be used in the proposed EM topology optimization technique in Chapter 5. In order to apply MPVL on the FEM matrix equation, the FEM matrix equation (5.2) is cast into an equivalent equation as [149]

$$(\mathbf{A}_1^{(k,m)} + \gamma \mathbf{A}_2^{(k,m)}) \tilde{\mathbf{e}}^{(k,m)} = \gamma \tilde{\mathbf{R}} \quad (6.1)$$

where

$$\begin{aligned} \mathbf{A}_1^{(k,m)} &= \begin{bmatrix} \mathbf{0}_N & \mathbf{I}_N \\ \mathbf{K}_0^{(k,m)} & \mathbf{K}_1^{(k,m)} \end{bmatrix}, \\ \mathbf{A}_2^{(k,m)} &= \begin{bmatrix} -\mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{K}_2^{(k,m)} \end{bmatrix}, \\ \tilde{\mathbf{e}}^{(k,m)} &= \begin{bmatrix} \mathbf{e}^{(k,m)} \\ \gamma \mathbf{e}^{(k,m)} \end{bmatrix}, \\ \tilde{\mathbf{R}} &= \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{R} \end{bmatrix}. \end{aligned} \quad (6.2)$$

where N is the number of elements in the unknown solution vector $\mathbf{e}^{(k,m)}$, \mathbf{I}_N and $\mathbf{0}_N$ represent the $N \times N$ identity matrix and the $N \times N$ zero matrix, respectively. The original equations in [149] do not have superscripts (k, m) . In this appendix, we add the superscripts (k, m) for purpose of consistency with Genetic Algorithm EM optimization used in this thesis.

When applying MPVL to the FEM equation for a fast frequency sweep, one specific frequency is selected as the frequency expansion point. Let γ_0 represent the complex propagation constant at the frequency expansion point for MPVL. Let $\mathbf{A}_0^{(k,m)}$ represent the matrix calculated at the frequency expansion point [149]

$$\mathbf{A}_0^{(k,m)} = \mathbf{A}_1^{(k,m)} + \gamma_0 \mathbf{A}_2^{(k,m)} = \begin{bmatrix} -\gamma_0 \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{K}_0^{(k,m)} & \mathbf{K}_1^{(k,m)} + \gamma_0 \mathbf{K}_2^{(k,m)} \end{bmatrix}. \quad (6.3)$$

The vector $\mathbf{g}_\eta^{(k,m)}$ for the calculation of $\mathbf{V}^{(k,m)}$ and $\mathbf{T}^{(k,m)}$ in the MPVL iterations is obtained by solving the matrix equation as [149]

$$\mathbf{A}_0^{(k,m)} \mathbf{g}_\eta^{(k,m)} = \begin{cases} \tilde{\mathbf{R}}, & \eta = 0, \\ -\mathbf{A}_2^{(k,m)} \mathbf{v}_\eta^{(k,m)}, & \eta = 1, 2, \dots, \rho. \end{cases} \quad (6.4)$$

References

- [1] J. L. Chávez-Hurtado and J. E. Rayas-Sánchez, “Polynomial-based surrogate modeling of RF and microwave circuits in frequency domain exploiting the multinomial theorem,” *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 12, pp. 4371-4381, Dec. 2016.
- [2] F. Feng, C. Zhang, J. Ma, and Q. J. Zhang, “Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions,” *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60-77, Jan. 2016.
- [3] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, “Neural network inverse modeling and applications to microwave filter design,” *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 4, pp. 867-879, Apr. 2008.
- [4] G. Angiulli, M. Cacciola, and M. Versaci, “Microwave devices and antennas modelling by support vector regression machines,” *IEEE Trans. Magnetics*, vol. 43, no. 4, pp. 1589-1592, Apr. 2007.

- [5] J. P. Jacobs, S. Koziel, and S. Ogurtsov, "Computationally efficient multi-fidelity Bayesian support vector regression modeling of planar antenna input characteristics," *IEEE Trans. Antennas. Propag.*, vol. 61, no. 2, pp. 980-984, Feb. 2013.
- [6] S. Koziel and J. W. Bandler, "Accurate modeling of microwave devices using Kriging-corrected space mapping surrogates," *Int. J. Numer. Model. Electron. Netw., Device Fields*, vol. 25, no. 1, pp. 1-14, Jan./Feb. 2012.
- [7] I. Couckuyt, S. Koziel, and T. Dhaene, "Surrogate modeling of microwave structures using Kriging, co-Kriging, and space mapping," *Int. J. Numer. Model. Electron. Netw., Device Fields*, vol. 26, no. 1, pp. 64-73, Jan./Feb. 2013.
- [8] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-From theory to practice," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1339-1350, Apr. 2003.
- [9] J. E. Rayas-Sánchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 420-435, Jan. 2004.
- [10] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q. J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105-118, May 2010.
- [11] J. Jin, F. Feng, W. Na, S. Yan, W. Liu, L. Zhu, and Q. J. Zhang, "Recent advances in neural network-based inverse modeling techniques for microwave

- applications,” *Int. J. Numer. Model. Electron. Networks Devices Fields* (2020): e2732, Feb. 2020.
- [12] V. K. Devabhaktuni, M. Yagoub, Y. Fang, J. Xu, and Q. J. Zhang, “Neural networks for microwave modeling: Model development issues and nonlinear modeling techniques,” *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 11, pp. 4-21, 2001.
- [13] C. Zhang, J. Jin, W. Na, Q. J. Zhang, and M. Yu, “Multivalued neural network inverse modeling and applications to microwave filters,” *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 8, pp. 3781-3797, Jun. 2018.
- [14] S. Li, Y. Wang, M. Yu, and A. Panariello, “Efficient modeling of Ku-band high power dielectric resonator filter with applications of neural networks,” *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 8, pp. 3427-3435, Aug. 2019.
- [15] P. Zhao and K. Wu, “Homotopy optimization of microwave and millimeter-wave filters based on neural network model,” *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 4, pp. 1390-1400, Jan. 2020.
- [16] W. Zhang, F. Feng, J. Zhang, Z. Zhao, J. Ma, and Q. J. Zhang, “Parallel decomposition approach to wide-range parametric modeling with applications to microwave filters,” *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 12, pp. 5288-5306, Dec. 2020.

- [17] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422-3428, Nov. 2005.
- [18] D. Schreurs, M. O'Droma, A. A. Goacher, and M. Gadringer, *RF Power Amplifier Behavioral Modeling*. New York, NY, USA: Cambridge University Press, 2008.
- [19] F. Mkadem and S. Boumaiza, "Physically inspired neural network model for RF power amplifier behavioral modeling and digital predistortion," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 4, pp. 913-923, Apr. 2011.
- [20] S. Wang, M. Roger, J. Sarrazin, and C. Lelandais-Perrault, "Hyperparameter optimization of two-hidden-layer neural networks for power amplifiers behavioral modeling using genetic algorithms," *IEEE Microw. Wireless Compon. Lett.*, vol. 29, no. 12, pp. 802-805, Dec. 2019.
- [21] W. Liu, W. Na, L. Zhu, J. Ma, and Q. J. Zhang, "A Wiener-type dynamic neural network approach to the modeling of nonlinear microwave devices," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 2043-2062, Jun. 2017.
- [22] L. Zhu, Q.-J. Zhang, K. Liu, Y. Ma, B. Peng, and S. Yan, "A novel dynamic neuro-space mapping approach for nonlinear microwave device modeling," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 2, pp. 131-133, Feb. 2016.

- [23] Y. Long, Z. Zhong, and Y.-X. Guo, "A novel 4-D artificial-neural-network-based hybrid large-signal model of GaAs pHEMTs," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 6, pp. 1752-1762, Jun. 2016.
- [24] A.-D. Huang, Z. Zhong, W. Wu, and Y. -X. Guo, "An artificial neural network-based electrothermal model for GaN HEMTs with dynamic trapping effects consideration," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 8, pp. 2519-2528, Aug. 2016.
- [25] W. Na, F. Feng, C. Zhang, and Q. J. Zhang, "A unified automated parametric modeling algorithm using knowledge-based neural network and l_1 optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 3, pp. 729-745, Mar. 2017.
- [26] F. Feng, V. Gongal-Reddy, C. Zhang, J. Ma, and Q. J. Zhang, "Parametric modeling of microwave components using adjoint neural networks and pole-residue transfer functions with EM sensitivity analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1955-1975, Jun. 2017.
- [27] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 12, pp. 2515-2523, Dec. 1997.
- [28] J. Monzó-Cabrera, J. L. Pedreno-Molina, A. Lozano-Gurrero, and A. Toledo-Moreo, "A novel design of a robust ten-port microwave reflectometer with autonomous calibration by using neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 12, pp. 2972-2978, Dec. 2008.

- [29] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” *Proc. Conf. Int. Speech Comm. Assoc.*, Florence, Italy, Aug. 2011, pp. 437-440.
- [30] M. Telgarsky, “Benefits of depth in neural networks,” *Proc. Conf. Learn. Theory*, New York, NY, USA, Jun. 2016, pp. 1517-1539.
- [31] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436-444, May 2015.
- [32] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computat.*, vol. 18, no. 7, pp. 1527-1554, Jul. 2006.
- [33] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances Neural Inf. Process. Syst.*, pp. 1097-1105, 2012.
- [34] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” *Proc. Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645-6649.
- [35] G. W. Cottrell, “New life for neural networks,” *Science*, vol. 313, no. 5786, pp. 454-455, Jul. 2006.
- [36] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, “Learning hierarchical features for scene labeling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915-1929, Aug. 2013.

- [37] D. Chen and B. K.-W. Mak, "Multitask learning of deep neural networks for low-resource speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 23, no. 7, pp. 1172-1183, Jul. 2015.
- [38] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," *Proc. Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 160-167.
- [39] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Proc. Conf. Empirical Methods Natural Language Process.*, Doha, Qatar, Oct. 2014, pp. 1724-1734.
- [40] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach." *Proc. Int. Conf. Mach. Learn.*, Bellevue, Washington, USA, Jul. 2011, pp 513-520.
- [41] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA, USA: Artech House, 2000.
- [42] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient decent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157-166, Mar. 1994.
- [43] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field*

- Guide to Dynamical Recurrent Neural Networks*, S.C. Kremer and J.F. Kolen, Eds. Piscataway, NJ, USA: IEEE Press, 2001.
- [44] Michael A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.
- [45] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” *Proc. Int. Conf. Artificial Intell. Statist.*, Fort Lauderdale, FL, USA, Apr. 2011, pp. 315-323.
- [46] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” *Proc. Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 8609-8613.
- [47] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 2013, pp. 1-6.
- [48] J. W. Bandler *et al.*, “Space mapping: The state of the art,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 337-361, Jan. 2004.
- [49] A. Pietrenko-Dabrowska, and S. Koziel, “Accelerated multiobjective design of miniaturized microwave components by means of nested kriging surrogate,” *Int. J. RF Microw. CAE*, vol. 30, no. 4, pp. 1-11, Jan. 2020.
- [50] F. E. Rangel-Patino, J. L. Chavez-Hurtado, A. Viveros-Wacher, J. E. Rayas-Sanchez, and N. Hakim, “System margining surrogate-based optimization in

- post-silicon validation,” *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 9, pp. 3109-3115, Sep. 2017.
- [51] S. Jakobsson, B. Andersson, and F. Edelvik, “Rational radial basis function interpolation with applications to antenna design,” in *J. Comput. Appl. Math.*, vol. 233, no. 4, pp. 889-904, Dec. 2009.
- [52] M. H. Bakr, J. W. Bandler, R. M. Biernacki, S. H. Chen, and K. Madsen, “A trust region aggressive space mapping algorithm for EM optimization,” *IEEE Trans. Microw. Theory Techn.*, vol. 46, no. 12, pp. 2412-2425, Dec. 1998, doi: 10.1109/22.739229.
- [53] S. Koziel, Q. S. Cheng, and J. W. Bandler, “Space mapping,” *IEEE Microw. Mag.*, vol. 9, no. 6, pp. 105-122, Dec. 2008.
- [54] J. W. Bandler, Q. S. Cheng, N. K. Nikolova, and M. A. Ismail, “Implicit space mapping optimization exploiting preassigned parameters,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 378-385, Jan. 2004.
- [55] S. Koziel, J. W. Bandler, and Q. S. Cheng, “Constrained parameter extraction for microwave design optimisation using implicit space mapping,” *IET Microw. Antennas Propag.*, vol. 5, no. 10, pp. 1156-1163, Jul. 2011.
- [56] Q. S. Cheng, J. W. Bandler, and J. E. Rayas-Sanchez, “Tuning-aided implicit space mapping,” *Int. J. RF Microw. CAE*, vol. 18, no. 5, pp. 445-453, Sep. 2008.

- [57] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 42, no. 12, pp. 2536-2544, Dec. 1994.
- [58] F. Feng, C. Zhang, V. M. R. Gongal-Reddy, Q. J. Zhang, and J. Ma, "Parallel space-mapping approach to EM optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 5, pp. 1135-1148, Apr. 2014.
- [59] S. Koziel, J. W. Bandler, and K. Madsen, "Space mapping with adaptive response correction for microwave design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 2, pp. 478-486, Feb. 2009.
- [60] R. B. Ayed, J. Gong, S. Brisset, F. Gillon, and P. Brochet. "Three-level output space mapping strategy for electromagnetic design optimization," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 671-674, Feb. 2012.
- [61] M. H. Bakr, J. W. Bandler, M. A. Ismail, J. E. Rayas-Sanchez and Q. J. Zhang, "Neural space mapping EM optimization of microwave structures," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Boston, MA, USA, June 2000, pp. 879-882.
- [62] J. E. Rayas-Sanchez, F. Lara-Rojo, and E. Martinez-Guerrero, "A linear inverse space-mapping (LISM) algorithm to design linear and nonlinear RF and microwave circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 3, pp. 960-968, Mar. 2005.

- [63] L. Zhang, J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, "Efficient analytical formulation and sensitivity analysis of neuro-space mapping for nonlinear microwave device modeling," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 9, pp. 2752-2767, Sep. 2005.
- [64] D. Gorissen, L. Zhang, Q. J. Zhang, and T. Dhaene, "Evolutionary neuro-space mapping technique for modeling of nonlinear microwave devices," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 2, pp. 213-229, Feb. 2011.
- [65] F. Feng, J. Zhang, W. Zhang, Z. Zhao, J. Jin, and Q. J. Zhang, "Coarse- and fine-mesh space mapping for EM optimization incorporating mesh deformation," *IEEE Microw. Wireless Comp. Lett.*, vol. 29, no. 8, pp. 510-512, Jul. 2019.
- [66] M. A. Ismail, D. Smith, A. Panariello, Y. Wang, and M. Yu, "EM-based design of large-scale dielectric-resonator filters and multiplexers by space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 386-392, Jan. 2004.
- [67] K. L. Wu, Y. J. Zhao, J. Wang, and M. K. K. Cheng, "An effective dynamic coarse model for optimization design of LTCC RF circuits with aggressive space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 393-402, Jan. 2004.
- [68] S. Amari, C. LeDrew, and W. Menzel, "Space-mapping optimization of planar coupled-resonator microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 5, pp. 2153-2159, May 2006.

- [69] M. Dorica and D. D. Giannacopoulos, "Response surface space mapping for electromagnetic optimization," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1123-1126, Apr. 2006.
- [70] S. Tu, Q. S. Cheng, Y. Zhang, J. W. Bandler, and N. K. Nikolova, "Space mapping optimization of handset antennas exploiting thin-wire models," *IEEE Trans. Antennas. Propag.*, vol. 61, no. 7, pp. 3797-3807, July 2013.
- [71] W. Zhang, F. Feng, V. M. R. Gongal-Reddy, J. Zhang, S. Yan , J. Ma, and Q. J. Zhang, "Space mapping approach to electromagnetic centric multiphysics parametric modeling of microwave components," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 7, pp. 3169-3185, Jul. 2018.
- [72] Y. Cao, G. Wang, and Q. J. Zhang, "A new training approach for parametric modeling of microwave passive components using combined neural networks and transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 11, pp. 2727-2742, Nov. 2009.
- [73] V. M. R. Gongal-Reddy, S. Zhang, C. Zhang, and Q. J. Zhang, "Parallel computational approach to gradient based EM optimization of passive microwave circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 44-59, Jan. 2016.
- [74] V. M. R. Gongal-Reddy, F. Feng, C. Zhang, S. Zhang, and Q. J. Zhang, "Parallel decomposition approach to gradient based EM optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 11, pp. 3380-3399, Nov. 2016.

- [75] B. Liu, H. Yang, and M. Lancaster, "Global optimization of microwave filters based on a surrogate model-assisted evolutionary algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1976-1985, Jun. 2017.
- [76] S. Koziel, "Shape-preserving response prediction for microwave design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 11, pp. 2829-2837, Nov. 2010.
- [77] S. Koziel, Q. S. Cheng, and J. W. Bandler, "Feature-based surrogates for low-cost microwave modelling and optimisation," *IET Microw. Antennas Propag.*, vol. 9, no. 15, pp. 1706-1712, Dec. 2015.
- [78] C. Zhang, F. Feng, V.-M.-R. Gongal-Reddy, Q. J. Zhang, and J. W. Bandler, "Cognition-driven formulation of space mapping for equal-ripple optimization of microwave filters," *IEEE Trans. Microw. Theory Tech.*, vol. 63, no. 7, pp. 2154-2165, Jul. 2015.
- [79] C. Zhang, F. Feng, Q. J. Zhang, and J. W. Bandler, "An enhanced cognition-driven formulation of space mapping for equal-ripple optimization of microwave filters," *IET Microw. Antennas Propag.*, vol. 12, no. 1, pp. 82-91, Jan. 2018.
- [80] F. Feng, C. Zhang, S. Zhang, V. M. R. Gongal-Reddy, and Q. J. Zhang, "Parallel EM optimization approach to microwave filter design using feature assisted neuro-transfer functions," in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, May. 2016, pp. 1-3.

- [81] F. Feng, W. Na, W. Liu, S. Yan, L. Zhu, J. Ma, and Q. J. Zhang, "Multifeature-assisted neuro-transfer function surrogate-based EM optimization exploiting trust-region algorithms for microwave filter design," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 2, pp. 531-542, Feb. 2020.
- [82] Q. S. Cheng, J. W. Bandler, S. Koziel, M. H. Bakr, and S. Ogurtsov, "The state of art of microwave CAD: EM-based optimization and modeling," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 20, no. 5, pp. 475-491, Sep. 2010.
- [83] A. Assadihaghi, *et al.*, "Design of microwave components using topology gradient optimization," in *IEEE 2006 Eur. Microw. Conf.*, Manchester, UK, Sep. 2006, pp. 462-465.
- [84] W. Wang, Y. Lu, J. S. Fu, and Y. Z. Xiong, "Particle swarm optimization and finite-element based approach for microwave filter design," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1800-1803, May 2005.
- [85] T. Kido, H. Deguchi, M. Tsuji, and M. Ohira, "Compact high-performance planar bandpass filters with arbitrarily-shaped conductor patches and slots," *IEEE 2008 Eur. Microw. Conf.*, Amsterdam, Netherlands, Oct. 2008, pp. 1165-1168.
- [86] E. Hassan, B. Scheiner, F. Michler, M. Berggren, E. Wadbro, F. Rohrl, S. Zorn, R. Weigel, and F. Lurz, "Multilayer topology optimization of wideband SIW-to-waveguide transitions," *IEEE Trans. Microw. Theory Tech.*, vol. 68, no. 4, pp. 1326-1339, Apr. 2020.

- [87] J. Byun and I. Park, "Design of dielectric waveguide filters using topology optimization technique," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1573-1576, Apr. 2007.
- [88] G. Kiziltas, D. Psychoudakis, J. L. Volakis, and N. Kikuchi, "Topology design optimization of dielectric substrates for bandwidth improvement of a patch antenna," *IEEE Trans. Antennas. Propag.*, vol. 51, no. 10, pp. 2732-2743, Oct. 2003.
- [89] N. Aage, N. A. Mortensen, and O. Sigmund, "Topology optimization of metallic devices for microwave applications," *Int. J. Numer. Meth. Eng.*, vol. 83, no. 2, pp. 228-248, Jul. 2010.
- [90] T. Nomura, M. Ohkado, P. Schmalenberg, J. Lee, O. Ahmed, and M. Bakr, "Topology optimization method for microstrips using boundary condition representation and adjoint analysis," *2013 Eur. Microw. Conf.*, Nuremberg, Germany, Oct. 2013, pp. 632-635.
- [91] J. M. Johnson and Y. Rahmat-Samii, "Genetic algorithms and method of moments (GA/MOM) for the design of integrated antennas," *IEEE Trans. Antennas. Propagat.*, vol. 47, no. 10, pp. 1606-1614, Oct. 1999.
- [92] A. Assadihaghi, S. Bila, D. Baillargeat, M. Aubourg, S. Verdeyme, C. Boichon, J. Puech, and L. Lapierre, "Optimization of microwave devices combining topology gradient and genetic algorithm," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 18, no. 5, pp. 454-463, Sep. 2008.

- [93] N. Takahashi, S. Nakazaki, and D. Miyagi, "Optimization of electromagnetic and magnetic shielding using ON/OFF method," *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3153-3156, Aug. 2010.
- [94] H. Choo and H. Ling, "Design of multiband microstrip antennas using a genetic algorithm," *IEEE Microw. Wireless Compon. Lett.*, vol. 12, no. 9, pp. 345-347, Sep. 2002.
- [95] J. L. T. Ethier and D. A. McNamara, "Antenna shape synthesis without prior specification of the feedpoint locations," *IEEE Trans. Antennas. Propagat.*, vol. 62, no. 10, pp. 4919-4934, Oct. 2014.
- [96] W. Wang, Y. Lu, and J. S. Fu, "Arbitrary planar microwave filter design by the FEM-GA approach," *Microw. Opt. Techn. Lett.*, vol. 41, no. 4, pp. 276-279, May 2004.
- [97] A. Modiri and K. Kiasaleh, "Efficient design of microstrip antennas for SDR applications using modified PSO algorithm," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1278-1281, May 2011.
- [98] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q. J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140-4155, Oct. 2019.
- [99] J. Jin, F. Feng, W. Na, J. Zhang, W. Zhang, Z. Zhao, and Q. J. Zhang, "Advanced cognition-driven EM optimization incorporating transfer function-

- based feature surrogate for microwave filters,” *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 1, pp. 15-28, Jan. 2021.
- [100] J. Jin, F. Feng, J. Zhang, J. Ma, and Q. J. Zhang, “Efficient EM topology optimization incorporating advanced Matrix Padé via Lanczos and genetic algorithm for microwave design,” *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 8, pp. 3645-3666, Aug. 2021.
- [101] K. C. Gupta, “Emerging trends in millimeter-wave CAD,” *IEEE Trans. Microw. Theory Techn.*, vol. 46, No. 6, pp. 747-755, Jun. 1998.
- [102] F. Wang, V. K. Devabhaktuni, C. Xi, and Q.J. Zhang, “Neural network structures and training algorithms for microwave applications,” *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 9, No. 3, pp. 216-240, May 1999.
- [103] F. E. Rangel-Patino, J. E. Rayas-Sánchez, A. Viveros-Wacher, J. L. Chávez-Hurtado, E. A. Vega-Ochoa, and N. Hakim, “Post-silicon receiver equalization metamodeling by artificial neural networks,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, No. 4, pp. 733-740, Apr. 2019.
- [104] S. K. Mandal, S. Sural, and A. Patra, “ANN- and PSO-based synthesis of on-chip spiral inductors for RF ICs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, No. 1, pp. 188-192, Jan. 2008.
- [105] J. E. Rayas-Sánchez, “Power in simplicity with ASM: Tracing the aggressive space mapping algorithm over two decades of development and engineering applications,” *IEEE Microw. Mag.*, vol. 17, no. 4, pp. 64-76, Apr. 2016.

- [106] L.-Y. Xiao, W. Shao, F.-L. Jin, B.-Z. Wang, "Multiparameter modeling with ANN for Antenna Design," *IEEE Trans. Antennas. Propag.*, vol. 66, no. 7, pp. 3718-3723, July 2018.
- [107] V. K. Devabhaktuni, C. Xi, F. Wang, and Q. J. Zhang, "Robust training of microwave neural models," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 12, no. 1, pp. 109-124, Jan. 2002.
- [108] E. A. Soliman, M. H. Bakr, and N. K. Nikolova, "Modeling of microstrip lines using neural networks: Applications to the design and analysis of distributed microstrip circuits," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 14, no. 2, pp. 166-173, Mar. 2004.
- [109] Q. J. Zhang and M. S. Nakhla, "Signal integrity analysis and optimization of VLSI interconnects using neural network models," *IEEE Int. Circuits Syst. Symp.*, London, England, May 1994, pp. 459-462.
- [110] A. Veluswami, M. S. Nakhla, and Q. J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high speed VLSI circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 5, pp. 712-723, May 1997.
- [111] D. E. Root, "Future device modeling trends," *IEEE Microw. Mag.*, vol. 13, no. 7, pp. 45-59, Nov. 2012.

- [112] S. Zhang, J. Xu, Q. J. Zhang, and D. E. Root, "Parallel matrix neural network training on cluster systems for dynamic FET modeling from large datasets," *IEEE MTT-S Int. Microw. Symp.*, San Francisco, CA, USA, May 2016, pp. 1-3.
- [113] X. Ding, V. K. Devabhaktuni, B. Chattaraj, M. C. E. Yagoub, M. Doe, J. J. Xu, and Q. J. Zhang, "Neural-network approaches to electromagnetic-based modeling of passive components and their applications to high-frequency and high-speed nonlinear circuit optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 436-449, Jan. 2004.
- [114] X. Du, M. Helaoui, A. Jarndal, T. Liu, B. Hu, X. Hu, and F. M. Ghannouchi, "ANN-based large-signal model of AlGaIn/GaN HEMTs with accurate buffer-related trapping effects characterization," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 7, pp. 3090-3099, Jul. 2020.
- [115] Z. Zhao, L. Zhang, F. Feng, W. Zhang, and Q. J. Zhang, "Space mapping technique using decomposed mappings for GaN HEMT modeling," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 8, pp. 3318-3341, Aug. 2020.
- [116] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins, and M. C. Calcaterra,, "Artificial neural networks for fast and accurate EM-CAD of microwave circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 5, pp. 794-802, May 1997.
- [117] S. Jeong, T.-H. Lin, and M. M. Tentzeris, "A real-time range-adaptive impedance matching utilizing a machine learning strategy based on neural net-

- works for wireless power transfer systems,” *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 12, pp. 5340-5347, Dec. 2019.
- [118] Y. Fang, M. C. E. Yagoub, F. Wang, and Q. J. Zhang, “A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks,” *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2335-2344, Dec. 2000.
- [119] T. Liu, S. Boumaiza, and F. M. Ghannouchi, “Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025-1033, Mar. 2004.
- [120] W. Zhang, A. Hasan, F. M. Ghannouchi, M. Helaoui, Y. Wu, and Y. Liu, “Novel calibration algorithm of multiport wideband receivers based on real-valued time-delay neural networks,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025-1033, Mar. 2004.
- [121] J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, “Neural based dynamic modeling of nonlinear microwave circuits,” *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 12, pp. 2769-2780, Dec. 2002.
- [122] Y. Cao, R. T. Ding, and Q. J. Zhang, “State-space dynamic neural network technique for high-speed IC applications: Modeling and stability analysis,” *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 6, pp. 2398-2409, Jun. 2006.
- [123] F. Wang and Q. J. Zhang, “Knowledge based neural models for microwave design,” *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 12, pp. 2333-2343, Dec. 1997.

- [124] Q. Zhang and A Benveniste, "Wavelet Networks," *IEEE Trans. Neural Networks*, vol. 3, no. 6, pp. 889-898, Nov. 1992.
- [125] E. K. Murphy and V. V. Yakovlev, "RBF network optimization of complex microwave systems represented by small FDTD modeling data sets," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 7, pp. 3069-3083, Jul. 2006.
- [126] L.-Y. Xiao, F.-L. Jin, B.-Z. Wang, W. T. Joines, and Q.-H. Liu, "Semisupervised radial basis function neural network with an effective sampling strategy," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 4, pp. 1260-1269, Apr. 2020.
- [127] S. Bila, *et al.*, "An accurate wavelet neural-network-based model for electromagnetic optimization of microwave circuits," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 9, No. 3, pp. 297-306, May 1999.
- [128] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of IEEE*, vol. 105, no. 12, pp. 2295-2329, Dec. 2017.
- [129] J. Jin, F. Feng, J. Zhang, S. Yan, W. Na, and Q. J. Zhang, "A novel deep neural network topology for parametric modeling of passive microwave components," *IEEE Access*, vol. 8, pp. 82273-82285, May 2020.
- [130] R. Hongyo, Y. Egashira, T. M. Hone, and K. Yamaguchi, "Deep neural network-based digital predistorter for Doherty power amplifiers," *IEEE Microw. Wireless Compon. Lett.*, vol. 29, no. 2, pp. 146-148, Feb. 2019.

- [131] H. M. Yao, W. E. I. Sha, and L. Jiang, “Two-step enhanced deep learning approach for electromagnetic inverse scattering problems,” *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2254-2258, Nov. 2019.
- [132] Z. Liu, X. Hu, T. Liu, X. Li, W. Wang, and F. M. Ghannouchi, “Attention-based deep neural network behavioral model for wideband wireless power amplifiers,” *IEEE Microw. Wireless Compon. Lett.*, vol. 30, no. 1, pp. 82-85, Jan. 2020.
- [133] G. Pan, Y. Wu, M. Yu, L. Fu, and H. Li, “Inverse modeling for filters using a regularized deep neural network approach,” *IEEE Microw. Wireless Compon. Lett.*, vol. 30, no. 5, pp. 457-460, May. 2020.
- [134] W. Shao and Y. Du, “Microwave imaging by deep learning network: feasibility and training method,” *IEEE Trans. Antennas Propag.*, vol. 68, no. 7, pp. 5626-5635, Jul. 2020.
- [135] Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional neural networks and applications in vision,” *Proc. IEEE Int. Symp. Circuits Syst.*, Paris, France, Jun. 2010.
- [136] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, J. Schmidhuber, “A system for robotic heart surgery that learns to tie knots using recurrent neural networks,” *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 2006.
- [137] S. Koziel, D. E. Ciaurri, and L. Leifsson, *Surrogate-Based Methods*. In: Koziel S., Yang XS. (eds) Computational Optimization, Methods and Algorithms.

- Studies in Computational Intelligence, vol 356. Springer, Berlin, Heidelberg, 2011.
- [138] A. Khalatpour, R. K. Amineh, Q. S. Cheng, M. H. Bakr, N. K. Nikolova, and J. W. Bandler, "Accelerating space mapping optimization with adjoint sensitivities," *IEEE Microw. Wireless Compon. Lett.*, vol. 21, no. 6, pp. 280-282, June 2011.
- [139] J. Zhang, F. Feng, W. Zhang, J. Jin, J. Ma, and Q. J. Zhang, "A novel training approach for parametric modeling of microwave passive components using Padé via Lanczos and EM sensitivity," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 6, pp. 2215-2233, Jun. 2020.
- [140] A. Kozak and W. K. Gwarek, "Unrestricted arbitrary shape optimization based on 3D electromagnetic simulation," *IEEE MTT-S Int. Microw. Symp. Dig.*, Baltimore, MD, USA, Jun. 1998, pp. 17-20.
- [141] P. Arcioni, M. Bozzi, M. Bressan, G. Conciauro, and L. Perregri, "Fast optimization, tolerance analysis, and yield estimation of H-/E-plane waveguide components with irregular shapes," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 319-328, Jan. 2004.
- [142] F. Campelo, J. Ramirez, and H. Igarashi, "A survey of topology optimization in electromagnetics: considerations and current trends," *Academia*, 46 (2010), Oct. 2010.

- [143] S. Zhou, W. Li, and Q. Li, "Design of 3-D periodic metamaterials for electromagnetic properties," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 4, pp. 910-916, Apr. 2010.
- [144] S. Zhou, W. Li, and Q. Li, "Level-set based topology optimization for electromagnetic dipole antenna design," *J. Comput. Phys.*, vol. 229, no. 19, pp. 6915-6930, Sep. 2010.
- [145] Z. Q. Qu, *Model Order Reduction Techniques With Applications in Finite Element Analysis*. London, U.K.: Springer, 2013.
- [146] A. Quarteroni and G. Rozza, *Reduced Order Methods for Modeling and Computational Reduction*. Cham, Switzerland: Springer, 2014.
- [147] Y.-R. Jeong, I.-P. Hong, K.-W. Lee, J.-H. Lee, and J.-G. Yook, "Fast frequency sweep using asymptotic waveform evaluation technique and thin dielectric sheet approximation," *IEEE Trans. Antennas. Propagat.*, vol. 64, no. 5, pp. 1800-1806, May 2016.
- [148] R. D. Slone, J.-F. Lee, and R. Lee, "Automating multipoint Galerkin AWE for a FEM fast frequency sweep," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 637-640, Mar. 2002.
- [149] R. D. Slone and R. Lee, "Applying Padé via Lanczos to the finite element method for electromagnetic radiation problems," *Radio Science*, vol. 35, no. 2, pp. 331-340, Mar. 2000.

- [150] J. Gong and J. L. Volakis, "AWE implementation for electromagnetic FEM analysis," *Electron. Lett.*, vol. 32, no. 24, pp. 2216-2217, Nov. 1996.
- [151] J. W. Bandler and Q. J. Zhang, "Large change sensitivity analysis in linear systems using generalized Householder formula," *Circuit Theory Appl.*, vol 14, no. 2 , pp. 89-101, Apr. 1986.
- [152] A. B. Manic, D. I. Olcan, M. M. Ilic, and B. M. Notaros, "Diakoptic approach combining finite-element method and method of moments in analysis of inhomogeneous anisotropic dielectric and magnetic scatterers," *Electromagnetics*, vol 34, no. 3-4 , pp. 222-238, 2014.
- [153] Y. Bengio, "Learning deep architectures for AI," *Foundations Trends[®] Mach. Learn.*, vol. 2, no. 1, pp. 1-127, Jan. 2009.
- [154] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, 1986, pp 318-362.
- [155] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness, Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107-116, Apr. 1998.
- [156] A. Viveros-Wacher and J. E. Rayas-Sánchez, "Analog fault identification in RF circuits using artificial neural networks and constrained parameter extrac-

- tion,” *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Reykjavik, Iceland, Aug. 2018, pp. 1-3.
- [157] J. Jin, F. Feng, W. Zhang, J. Zhang, Z. Zhao, and Q. J. Zhang, “Recent advances in deep neural network technique for high-dimensional microwave modeling,” *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Hangzhou, China, Dec. 2020, pp. 1-3.
- [158] V. Miraftab and R. R. Mansour, “A robust fuzzy-logic technique for computer-aided diagnosis of microwave filters,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 450-456, Jan. 2004.
- [159] P. Harscher, R. Vahldieck, and S. Amari, “Automated filter tuning using generalized low-pass prototype networks and gradient-based parameter extraction,” *IEEE Trans. Microw. Theory Techn.*, vol. 49, no. 12, pp. 2532-2538, Dec. 2001.
- [160] M. Kahrizi, S. Safavi-Naeini, and S. K. Chaudhuri, “Computer diagnosis and tuning of microwave filters using model-based parameter estimation and multi-level optimization,” *IEEE MTT-S Int. Microw. Symp. Dig.*, Boston, MA, USA, Jun. 2000, pp. 1641-1644.
- [161] G. Pepe, F. J. Gortz, and H. Chaloupka, “Computer-aided tuning and diagnosis of microwave filters using sequential parameter extraction,” *IEEE MTT-S Int. Microw. Symp. Dig.*, Fort Worth, TX, USA, Jun, 2004, pp. 1373-1376.
- [162] J. Jin, C. Zhang, and Q. J. Zhang, “An enhanced multi-circuit optimization approach for identification of microwave filter parameters,” *Proc. IEEE MTT-S*

- Int. Conf. Numer. Electromagn. Multiphys. Modeling Optim. (NEMO)*, Beijing, China, Jul. 2016, pp. 1-2.
- [163] H.-T. Hsu, Z. Zhang, K. A. Zaki, and A. E. Atia, "Parameter extraction for symmetric coupled-resonator filters," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 12, pp. 2971-2978, Dec. 2002.
- [164] J. W. Bandler, S. H. Chen, and S. Daijavad, "Microwave device modeling using efficient l_1 optimization: a novel approach," *IEEE Trans. Microw. Theory Techn.*, vol. 34, no. 12, pp. 1282-1293, Dec. 1986.
- [165] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052-1061, Jul. 1999.
- [166] W. Zhang, F. Feng, S. Yan, W. Na, J. Ma, and Q. J. Zhang, "EM-centric multiphysics optimization of microwave components using parallel computational approach," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 2, pp. 479-489, Feb. 2020.
- [167] S. Liao, H. Kabir, Y. Cao, J. Xu, Q. J. Zhang, and J. Ma, "Neural-network modeling for 3-D substructures based on spatial EM-field coupling in finite-element method," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 1, pp. 21-38, Jan. 2011.
- [168] F. Feng, J. Zhang, J. Jin, W. Na, S. Yan, and Q. J. Zhang, "Efficient FEM-based EM optimization technique using combined Lagrangian method with New-

- ton's method," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 6, pp. 2194-2205, Jun. 2020.
- [169] J. M. Jin, *The Finite Element Method in Electromagnetics*. New York, NY, USA: Wiley, 2002.
- [170] Y. Rahmat-Samii and E. Michielssen, *Electromagnetic Optimization by Genetic Algorithms*. New York: J. Wiley. Print, 1999.
- [171] T. T. Lu and S. H. Shiou, "Inverses of 2×2 block matrices," *Comput. Math. Appl.* vol. 43, no. 1-2, pp. 119-129, Jan. 2002.
- [172] S.-J. Wu and P.-T. Chow, "Steady-state genetic algorithms for discrete optimization of trusses", *Comput. Struct.*, vol. 56, no. 6, pp. 979-991, Sep. 1995.
- [173] X. Chen, C. Gu, Y. Zhang, and R. Mittra, "Analysis of partial geometry modification problems using the partitioned-inverse formula and Sherman-Morrison-Woodbury formula-based method," *IEEE Trans. Antennas. Propagat.*, vol. 66, no. 10, pp. 5425-5431, Oct. 2018.
- [174] C. Geuzaine and J.-F. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post- processing facilities", *Int. J. Numer. Methods Eng.*, vol. 79, no. 11, pp. 1309-1331, Sep. 2009.
- [175] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé Approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 5, pp. 639-649, May 1995.

- [176] J. R. Perez and J. Basterrechea, "Comparison of different heuristic optimization methods for near-field antenna measurements," *IEEE Trans. Antennas. Propagat.*, vol. 55, no. 3, pp. 549-555, Mar. 2007.
- [177] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 132-141, Feb. 2013.
- [178] E. D. P. Puchta, H. V. Siqueira, and M. d. S. Kaster, "Optimization tools based on metaheuristics for performance enhancement in a Gaussian adaptive PID controller," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1185-1194, Mar. 2020.
- [179] H. Zhou, M. Song, and W. Pedryczb, "A comparative study of improved GA and PSO in solving multiple traveling salesmen problem," *Appl. Soft Comput.*, vol. 64, pp. 564-580, Mar. 2018.
- [180] S. A. Hosseini, A. Hajipour, and H. Tavakoli, "Design and optimization of a CMOS power amplifier using innovative fractional-order particle swarm optimization," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. No. 105831.
- [181] J. Liu, Y. Mei, and X. Li, "An analysis of the inertia weight parameter for binary particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 666-681, Oct. 2016.
- [182] S. Sanogo and F. Messine, "Design of space thrusters: a topology optimization problem solved via a branch and bound method," *J. Global Optim.*, vol. 64, no. 2, pp. 273-288, 2016.