

# **Advances in Optimization and Inverse Modeling Techniques for Microwave Filters**

by

**Chao Zhang, M. Eng.**

A thesis submitted to The Faculty of Graduate Studies and Postdoctoral  
Affairs in partial fulfilment of the degree requirements of

**Doctor of Philosophy**

Ottawa-Carleton Institute for  
Electrical and Computer Engineering

Department of Electronics  
Carleton University  
Ottawa, Ontario, Canada, May 2018

Copyright ©

2018 - Chao Zhang

## Abstract

Optimization and modeling techniques are the essential part of design process of microwave filters. Space mapping is a recognized method for speeding up electromagnetic (EM) optimization, and has been applied to microwave filter design. In the first part of this thesis, a cognition-driven formulation of space mapping method is proposed and applied to EM-based filter optimization to increase optimization efficiency and the ability to avoid being trapped in local minima. This new technique utilizes two sets of intermediate feature space parameters, including feature frequency parameters and ripple height parameters. The design variables are mapped to the feature frequency parameters, which are further mapped to the ripple height parameters. By formulating the cognition-driven optimization directly in the feature space, our method increases optimization efficiency and the ability to avoid being trapped in local minima. In the second part of this thesis, a multivalued neural network is proposed to solve the non-uniqueness (multivalued) problem in inverse modeling. Our proposed technique can be effectively applied to parameter extraction of microwave filters. We propose a multivalued neural network inverse modeling technique to associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. One set of geometrical or physical parameters is called one value of our proposed inverse model. Our proposed multivalued neural network is structured to accommodate multiple values for the model output. We also propose a new training error function to focus on matching each training sample using only one value of our proposed inverse model, while other values are

free and can be trained to match other contradictory samples. In this way, our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. Therefore, our proposed technique can solve the non-uniqueness problem in a simpler and more automated way compared to existing ANN inverse modeling techniques. Both proposed techniques in this thesis are illustrated by several microwave filter examples. Our proposed techniques can be used in the design and tuning processes of microwave filters.

To my parents

*Yongxue Zhang and Benfeng Zhang*

## Acknowledgements

I would like to express my sincere appreciation to my supervisor Dr. Qi-Jun Zhang, Professor of Department of Electronics, Carleton University, for the constant support and expert guidance during the course of this research work. His encouragement, motivation and thoughtful insights during my research made my journey in realizing my goal a memorable one. It was my honor to work under his supervision and guidance.

I would like to thank my co-supervisor Dr. Ming Yu, Professor of Electronic Engineering Dept., Chinese University of Hong Kong, for his support and advice on this research work. His expert advice on microwave filters greatly assisted this research work.

Special thanks to Dr. John W. Bandler, Professor of Department of Electrical and Computer Engineering, McMaster University, for providing guidance on innovative formulation of cognition-driven optimization and valuable advices on space mapping techniques. His idea inspired me a lot to fulfill this research work.

I would also like to thank all present and former colleagues especially Dr. Feng Feng, Dr. Venu-Madhav-Reddy Gongal-Reddy, Weicong Na, Jianan Zhang, Wei Zhang, Zhihao Zhao, and Jing Jin in our research group for their helpful discussions and enthusiasm in research collaborations.

Many thanks to Blazenka Power, Anna Lee, Sylvie Beekmans, Scott Bruce, Stephen Maclaurin, and other staff in the Department of Electronics for offering a

helpful and friendly yet professional support.

Finally, I wish to thank my parents, Y. Zhang and B. Zhang, and my brother, N. Zhang for their love, belief, support and encouragement throughout my study.

# Table of Contents

|   |             |
|---|-------------|
| <b>Abstract</b>   | <b>iv</b>   |
| <b>Acknowledgements</b>   | <b>vii</b>  |
| <b>List of Figures</b>  | <b>xii</b>  |
| <b>List of Tables</b>   | <b>xxiv</b> |
| <b>1 Introduction</b>   | <b>1</b>    |
| 1.1 Introduction . . . . .  | 1           |
| 1.2 List of Contributions . . . . .   | 3           |
| 1.2.1 Cognition-Driven Formulation of Space Mapping for Equal-Ripple Optimization of Microwave Filters [11] . . . . . | 4           |
| 1.2.2 Multivalued Neural Network Inverse Modeling and Applications to Microwave Filters [18] . . . . .                | 4           |

|          |   |          |
|----------|---|----------|
| 1.3      | Thesis Organization . . . . .                   | 5        |
| <b>2</b> | <b>Literature Review</b>                        | <b>7</b> |
| 2.1      | Optimization of Microwave Filters . . . . .     | 9        |
| 2.2      | Direct EM Optimization Methods . . . . .        | 9        |
| 2.2.1    | Pattern Search Method . . . . .                 | 10       |
| 2.2.2    | Conjugate Gradient Method . . . . .             | 11       |
| 2.2.3    | Quasi-Newton Method . . . . .                   | 12       |
| 2.3      | Space Mapping Optimization . . . . .            | 13       |
| 2.3.1    | The Space Mapping Concept . . . . .             | 14       |
| 2.3.2    | Input Space Mapping . . . . .                   | 17       |
| 2.3.3    | Implicit Space Mapping . . . . .                | 17       |
| 2.3.4    | Output Space Mapping . . . . .                  | 18       |
| 2.3.5    | Tuning Space Mapping . . . . .                  | 19       |
| 2.3.6    | Coarse and Fine Mesh Space Mapping . . . . .    | 19       |
| 2.4      | Inverse Modeling of Microwave Filters . . . . . | 20       |
| 2.4.1    | Forward Model . . . . .                         | 20       |
| 2.4.2    | Inverse Model . . . . .                         | 22       |

|          |   |           |
|----------|---|-----------|
| 2.5      | Optimization Method for Inverse Problems of Microwave Filters . . .   | 23        |
| 2.6      | Learning-By-Examples (LBE) Methodologies Techniques . . . . .   | 25        |
| 2.6.1    | Support Vector Machine (SVM) . . . . .  | 27        |
| 2.6.2    | Gaussian Processes (GP) . . . . .   | 28        |
| 2.6.3    | Artificial Neural Network (ANN) . . . . .   | 29        |
| 2.7      | ANN Inverse Modeling of Microwave Filters . . . . .   | 29        |
| 2.7.1    | Direct ANN Inverse Modeling . . . . .   | 30        |
| 2.7.2    | Neural Network Inverse Modeling Technique in [17] . . . . .   | 32        |
| 2.8      | Summary . . . . .   | 33        |
| <b>3</b> | <b>Cognition-Driven Formulation of Space Mapping for Equal-Ripple<br/>Optimization of Microwave Filters</b> | <b>34</b> |
| 3.1      | Introduction . . . . .  | 35        |
| 3.2      | Original Optimization Problem . . . . .   | 37        |
| 3.3      | Proposed Space Mapping Technique for Microwave Optimization . .   | 40        |
| 3.3.1    | Feature Space Parameters . . . . .  | 40        |
| 3.3.2    | Mapping from $\mathbf{f}$ Space to $\mathbf{x}$ Space . . . . .   | 42        |
| 3.3.3    | Stage 1: Adjustment of Locations of Feature Frequencies . .   | 44        |

|          |  |           |
|----------|--|-----------|
| 3.3.4    | Stage 2: Mapping from $\mathbf{t}$ Space to $\mathbf{f}$ Space and Adjustment of Ripple Height . . . . .             | 46        |
| 3.3.5    | Update of Trust Region . . . . .   | 49        |
| 3.3.6    | Summary . . . . .  | 50        |
| 3.3.7    | Discussion . . . . .   | 52        |
| 3.4      | Application Examples . . . . .   | 54        |
| 3.4.1    | Optimization of A Four-Pole Waveguide Filter . . . . .   | 54        |
| 3.4.2    | Optimization of An Iris Coupled Microwave Cavity Filter . . . . .  | 65        |
| 3.5      | Conclusions and Summary . . . . .  | 76        |
| <b>4</b> | <b>Multivalued Neural Network Inverse Modeling and Applications to Microwave Filters</b> . . . . .                   | <b>77</b> |
| 4.1      | Introduction . . . . .   | 78        |
| 4.2      | The Proposed Multivalued Neural Network Approach to Inverse Modeling . . . . .                                       | 82        |
| 4.2.1    | The Direct Inverse Modeling Problem . . . . .  | 83        |
| 4.2.2    | The Proposed Multivalued Neural Network Inverse Model . . . . .  | 88        |
| 4.2.3    | The Proposed Training and Test Error Functions of the Proposed Multivalued Neural Network Inverse Modeling . . . . . | 90        |

|          |  |            |
|----------|--|------------|
| 4.2.4    | The Proposed Method of Selecting the Valid Values of the Proposed Inverse Model . . . . .              | 92         |
| 4.2.5    | Application of the Proposed Multivalued Neural Network Inverse Model to Parameter Extraction . . . . . | 94         |
| 4.2.6    | Discussion . . . . .   | 97         |
| 4.3      | Examples . . . . .   | 100        |
| 4.3.1    | Illustrative Example: the Inverse Modeling of a Microstrip Bandpass Filter . . . . .                   | 100        |
| 4.3.2    | The Inverse Modeling of A Fourth-Order Dielectric Resonator Filter . . . . .                           | 108        |
| 4.3.3    | The Inverse Modeling of A Sixth-Order Multicoupled Cavity Filter . . . . .                             | 117        |
| 4.3.4    | The Inverse Modeling of An Eighth-Order Dual-Band Filter . . . . .                                     | 127        |
| 4.4      | Conclusion . . . . .   | 133        |
| <b>5</b> | <b>Conclusions and Future Research</b>   | <b>134</b> |
| 5.1      | Conclusions . . . . .  | 134        |
| 5.2      | Future Research . . . . .  | 136        |
|          | <b>Bibliography</b>  | <b>137</b> |

# List of Figures

|     |   |    |
|-----|---|----|
| 2.1 | Space mapping concept following [43] . . . . .  | 14 |
| 2.2 | Mathematical representation of space mapping following [8] . . . . .  | 16 |
| 2.3 | Illustration of Implicit Space Mapping (ISM) following [27] . . . . .   | 18 |
| 2.4 | Example illustrating forward and inverse models. (a) Forward model.<br>(b) Inverse model. The inputs $x_{F3}$ and $x_{F4}$ (output $y_{F2}$ and $y_{F3}$ ) of<br>the forward model are swapped to the outputs (inputs) of the inverse<br>model, respectively. . . . . | 21 |
| 2.5 | Illustration of LBE-based techniques . . . . .  | 26 |
| 2.6 | Illustration of the SVM technique. . . . .  | 28 |
| 2.7 | Illustration of a three-layer ANN structure. . . . .  | 30 |

3.1 The optimization of the four-pole waveguide filter: (a), (c), and (e) depicts responses during quasi-Newton optimization iterations , where only 3 poles are forced to contribute passband response: (a) Initial, (c) Iteration 50, (e) Iteration 300; Subfigures (b), (d), and (f) depicts proposed cognition-driven space mapping iterations. (d) proposed method adjusts the frequency locations of reflection zeros to the passband in the first stage, (f) then adjusts the frequency locations of reflection zeros according to the ripple height parameters. One iteration of our proposed method includes multiple EM simulations in parallel. The total computation time for one iteration in our proposed method is only incrementally more than that of a single EM simulation in one quasi-Newton optimization iteration. . . . . 38

3.2 The four-pole waveguide filter. (a) the simulation structure, (b) the proposed feature parameters t and f in the filter response. . . . . 43

3.3 The flowchart of the proposed cognition-driven space mapping technique. . . . . 53

|     |  |    |
|-----|--|----|
| 3.4 | A comparison of the results for 3 different optimization methods: (a) the starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) a good equal-ripple response is obtained after 7 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum. . . . . | 56 |
| 3.5 | The objective function values of the four-pole waveguide filter response using: the proposed SM method (o) and coarse and fine mesh space mapping method ( $\times$ ). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 7 iterations. . . . .   | 57 |
| 3.6 | The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 7th iteration. Both $U_1$ and $Var(\mathbf{t})$ converge fast. . . . .  | 59 |

3.7 A comparison of the results for 3 different optimization methods: (a) the starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) good equal-ripple response is obtained after 8 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum. . . . . 61

3.8 Objective function values of the four-pole waveguide filter response with bad starting point using: the proposed space mapping method (o) and coarse and fine mesh space mapping method ( $\times$ ). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 8 iterations. . . . . 62

3.9 Feature space objective functions  $U_1$  and  $U_2$  (i.e.,  $Var(\mathbf{t})$ ) for the four-pole waveguide filter example with bad starting point. The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 7th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast. 63

3.10 The structure of the waveguide cavity filter. . . . . 65

3.11 A comparison of the results for 3 different optimization methods: (a) the starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) good equal-ripple response is obtained after 4 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum. . . . . 67

3.12 The objective function values of the cavity filter response using: the proposed space mapping method (o) and coarse and fine mesh space mapping method ( $\times$ ). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 4 iterations. . . . . 69

3.13 Feature space objective functions  $U_1$  and  $U_2$  (i.e.,  $Var(\mathbf{t})$ ) for the cavity filter. The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 4th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast. . . . . 70

3.14 Comparison of the results for three different optimization methods for the cavity filter example with bad starting point: (a) a very bad starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) good equal-ripple response is obtained after 12 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum. . . . . 72

3.15 The objective function values of the cavity filter response with bad starting point using: the proposed space mapping method (o) and coarse and fine mesh space mapping method ( $\times$ ). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 4 iterations. . . . . 73

3.16 Feature space objective functions  $U_1$  and  $U_2$  (i.e.,  $Var(\mathbf{t})$ ) for the cavity filter with bad starting point. The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 4th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast. . . . . 74

4.1 (a). A standard ANN structure for direct inverse modeling of a microwave component:  $\bar{\mathbf{x}}$  is a  $q$ -vector containing the inputs, which are usually the electrical parameters.  $\bar{\mathbf{y}}$  is a  $p$ -vector containing the outputs, which are usually the physical or geometrical parameters. (b). Our proposed multivalued neural network inverse modeling technique can associate a single set of electrical parameters with multiple sets of geometrical or physical parameters.  $\bar{\mathbf{x}}$  contains the inputs, which are usually the electrical parameters.  $\bar{\mathbf{y}}_i$  is the  $i$ th value of proposed inverse model. Each value  $\bar{\mathbf{y}}_i$  is a  $p$ -vector containing all the physical or geometrical parameters defined in the direct inverse modeling technique,  $i = 1, 2, \dots, N_v$ . The total number of output neurons in our proposed structure in (b) is  $N_v \cdot m$ . . . . . 85

4.2 An illustration example: the microstrip rectangular inductor. (a) The forward problem is from the conductor width  $W$  to the effective quality factor  $Q_{eff}$ , and the input-output relationship is unique. (b) In the inverse problem, the effective quality factor  $Q_{eff}$  is used as the input of the inverse model, and the conductor width  $W$  as the output. The inverse input-output relationship is non-unique. (c) Some training samples of the inverse model. . . . . 86

4.3 The diagram for the proposed method of selecting the valid values of the proposed multivalued neural network inverse model. . . . . 95

4.4 (a) The inverse modeling of a microstrip bandpass filter example:  $\bar{x} = \text{mag}(S_{21})|_{4GHz}$ , and  $\bar{y} = L_3$ . (b) The direct inverse model has one input which is  $\text{mag}(S_{21})|_{4GHz}$ , and one output  $L_3$ . (c) The proposed multivalued neural network inverse model with 3 values:  $\bar{y}_1$ ,  $\bar{y}_2$ , and  $\bar{y}_3$  represent Value 1, Value 2 and Value 3 of the length  $L_3$ , respectively. (d) The existing inverse modeling technique [17] using 2 sub-models. (e) The existing inverse modeling technique [17] using 4 sub-models. The symbols ( $\nabla$ ,  $\Delta$ ,  $\circ$ ,  $\times$ , and  $+$ ) are used to label the different model outputs. . . . . 101

4.5 Illustration of different inverse modeling techniques for the microstrip bandpass filter example. (a) The input-output relationship of forward problem is unique. (b) The inverse problem and the direct ANN inverse modeling technique. (c) Our proposed inverse modeling technique with 3 values ( $\nabla$  for  $\bar{y}_1$ ,  $\Delta$  for  $\bar{y}_2$ , and  $\circ$  for  $\bar{y}_3$ ) before valid value selection process. (d) After valid value selection, our proposed inverse modeling technique with 3 values ( $\nabla$  for  $\bar{y}_1$ ,  $\Delta$  for  $\bar{y}_2$ , and  $\circ$  for  $\bar{y}_3$ ) can learn all the training data automatically, matching all the contradictory data at the same time, and achieving good accuracy. (e) The existing technique in [17] with 2 sub-models. Inside the first sub-model, there is still contradictory data, therefore the sub-model #1 ( $\times$ ) is not accurate. The sub-model #2 (+) is good. (f) The existing technique in [17] with 4 sub-models can achieve good model accuracy. (g) SVM inverse model behaves well at the range where there is no contradictory data. Where there are three different solutions for the same input (i.e., multivalued output for the same input), the SVM model cannot learn all the three contradictory values. (h) GP inverse model tries to approach the average of the contradictory data. . . . . 104

4.6 Two test samples for the fourth-order filter example. Sample #1: (a) The  $S$ -parameters calculated from the 1st value  $\bar{\mathbf{y}}_1$  provided by our proposed inverse model, (b) The  $S$ -parameters calculated from the 2nd value  $\bar{\mathbf{y}}_2$  provided by our proposed inverse model. Sample #2: (c) The  $S$ -parameters calculated from the 1st value  $\bar{\mathbf{y}}_1$  provided by our proposed inverse model, (d) The  $S$ -parameters calculated from the 2nd  $\bar{\mathbf{y}}_2$  provided by our proposed inverse model. For Sample #1,  $S$ -parameters calculated from both values match the desired  $S$ -parameters very well. Both values are valid. For Sample #2, only the  $S$ -parameter calculated from the 1st value matches the desired  $S$ -parameters very well. Therefore, the 1st value is valid, and the 2nd value is dummy. Our proposed valid value selection process can distinguish valid values from the dummy ones. Our proposed inverse model with valid value selection process can always provide accurate solutions of coupling parameters for a given set of  $S$ -parameter input. 111

4.7 The slightly detuned filter case for the sixth-order cavity filter example. Our proposed inverse model can provide 4 values for the coupling parameters: (a)  $S_{11}$  calculated from the 1st value  $\bar{y}_1$  (-); (b)  $S_{11}$  calculated from the 2nd value  $\bar{y}_2$  (-); (c)  $S_{11}$  calculated from the 3rd value  $\bar{y}_3$  (-). (d)  $S_{11}$  calculated from the 4th value  $\bar{y}_4$  (-). The desired  $S$ -parameter is shown in circles (o).  $S$ -parameters calculated from  $\bar{y}_2$  and  $\bar{y}_4$  match the desired  $S$ -parameter very well. They are selected as valid values. The other two are dummy. Our proposed valid value selection process can distinguish valid values from the dummy ones. Our proposed inverse model with valid value selection process can always provide accurate solutions of coupling parameters for a given set of  $S$ -parameter. . . . . 122

4.8 The slightly and highly detuned filter cases for the sixth-order cavity filter example. (a) The slightly detuned filter case. (b) The highly detuned filter case. The circles (o) represent the desired  $S$ -parameters. The lines (-) represent the  $S$ -parameters calculated by the extracted coupling parameters, which are obtained using our proposed multi-valued neural network inverse model. The  $S$ -parameters calculated from the extracted coupling parameters in both cases can match the desired  $S$ -parameters very well. . . . . 125

4.9 Two cases for the eighth-order dual-band filter example: (a) The slightly detuned filter case. (b) The highly detuned filter case. The circles (o) represent the desired  $S$ -parameters. The lines (-) represent the  $S$ -parameters calculated by the extracted coupling parameters, which are obtained by the multi-circuit ANN process using our proposed multivalued neural network inverse model. . . . . 131

# List of Tables

|     |  |     |
|-----|--|-----|
| 3.1 | Comparison of Three Methods for The Waveguide Filter . . . . .   | 60  |
| 3.2 | Comparison of Three Methods for The Waveguide Filter with Bad Starting Point . . . . .   | 64  |
| 3.3 | Comparison of Three Methods for The Cavity Filter . . . . .  | 68  |
| 3.4 | Comparison of Three Methods for The Cavity Filter with Bad Starting Point . . . . .  | 75  |
| 4.1 | Definition of the Training and Test Data Range for the Fourth-Order Dielectric Resonator Filter Example . . . . .                    | 108 |
| 4.2 | The Comparison of Modeling Results Using Different Inverse Modeling Methods for the Fourth-Order Dielectric Resonator Filter Example | 110 |
| 4.3 | Definition of the Training and Test Data Range for the Sixth-Order Cavity Filter Example . . . . .                                   | 120 |

|     |   |     |
|-----|---|-----|
| 4.4 | The Comparison of Modeling Results Using Different Inverse Modeling Methods for the Sixth-Order Cavity Filter . . . . .     | 120 |
| 4.5 | Definition of the Training and Test Data Range for the Eighth-Order Dual-Band Filter Example . . . . .                      | 129 |
| 4.6 | The Comparison of Modeling Results Using Different Inverse Modeling Methods for the Eighth-Order Dual-Band Filter . . . . . | 130 |
| 4.7 | Comparison Between the Desired and Extracted Coupling Parameters for the Eighth-Order Dual-Band Filter Example . . . . .    | 132 |

# List of Symbols

|                    |  |
|--------------------|--|
| $\mathbf{A}$       | Jacobian matrix of $\mathbf{F}$ representing the mapping from feature frequency $\mathbf{f}$ to design variable $\mathbf{x}$                           |
| $\mathbf{A}^{(k)}$ | Jacobian matrix of $\mathbf{F}$ in the $k$ th iteration  |
| $\mathbf{a}$       | Unknown normal vector in the SVM technique.  |
| $\mathbf{B}$       | Matrix representing the mapping matrix between feature frequency $\mathbf{f}$ space to the ripple height parameter $\mathbf{t}$ space                  |
| $\mathbf{B}^{(k)}$ | Matrix representing the mapping between feature frequency $\mathbf{f}$ space to the ripple height parameter $\mathbf{t}$ space in the $k$ th iteration |
| $\mathbf{B}_I$     | Input transformation matrix of size $\Re^{N \times N}$ that transforms $\mathbf{x}_f$ to $\mathbf{x}_c$ using space mapping techniques                 |
| $\mathbf{a}$       | Unknown bias coefficients in the SVM technique.  |
| $\mathbf{c}$       | Vector containing the mapping coefficients between feature frequency $\mathbf{f}$ space to the ripple height parameter $\mathbf{t}$ space              |

|                           |   |
|---------------------------|---|
| $\mathbf{c}^{(k)}$        | Vector containing the mapping coefficients between feature frequency $\mathbf{f}$ space to the ripple height parameter $\mathbf{t}$ space in the $k$ th iteration |
| $\mathbf{c}_I$            | Input shifting vector of size $\Re^{N \times 1}$ that transforms $\mathbf{x}_f$ to $\mathbf{x}_c$ using space mapping techniques                                  |
| $\mathbf{d}_{k_s}$        | The $k_s$ th sample of $\bar{\mathbf{y}}$ corresponding to the input $\bar{\mathbf{x}}_{k_s}$ .   |
| $E(\mathbf{w})$           | Conventional training error function of this direct inverse modeling problem  |
| $\tilde{E}(\mathbf{w})$   | Proposed training error function of the proposed multivalued neural network   |
| $E_{k_s}(\mathbf{w})$     | Proposed training error for the $k_s$ th training sample of the proposed multivalued neural network   |
| $E_o(\bar{\mathbf{y}}_i)$ | Error function between $\mathbf{z}_i$ and $\bar{\mathbf{x}}$ for different $\bar{\mathbf{y}}_i$   |
| $e_{i,k_s}(\mathbf{w})$   | Error between $\bar{\mathbf{y}}_i$ and $\mathbf{d}_{k_s}$   |
| $\mathbf{F}$              | Mapping function from the design variables $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^T$ to the feature frequency parameters $\mathbf{f}$                           |
| $F_m$                     | $m$ th element of $\mathbf{F}$  |
| $\mathbf{f}$              | Vector representing the feature frequency parameters  |

|                                |   |
|--------------------------------|---|
| $\mathbf{f}_i$                 | Vector representing the feature frequency parameters for $\mathbf{x}_i$ , $i = 1, 2, \dots, 2N + 1$                     |
| $\mathbf{f}^{(k)}$             | The feature frequency parameters in the $k$ th iteration  |
| $\mathbf{f}_{FWD}$             | Input-output relationship of the forward problem  |
| $\mathbf{f}_{FWD}^{-1}$        | Input-output relationship of the inverse problem  |
| $f_H$                          | Higher frequency edge of the filter passband  |
| $f_h$                          | Upper frequency of the current frequency band   |
| $f_L$                          | Lower frequency edge of the filter passband   |
| $f_l$                          | Lower frequency of the current frequency band   |
| $\mathbf{f}_m(\mathbf{x}_F^j)$ | The measured/simulated response of the $j$ th circuit, $j = 1, 2, \dots, N_c$   |
| $\mathbf{f}_d^1$               | Vector representing the desired feature frequency parameters in the first stage   |
| $\mathbf{f}_d^{(k)}$           | Vector representing the desired frequency vector in the $k$ th iteration  |
| $\mathbf{f}_s^{(k)}$           | Vector representing the feature frequency parameters at $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ in the $k$ th iteration |
| $\mathbf{g}_{ANN}$             | Multilayer perceptron (MLP) representing our proposed multivalued neural network inverse model                          |
| $k$                            | Optimization iteration index  |

|                         |   |
|-------------------------|---|
| $k_s$                   | Training sample index   |
| $M$                     | Coupling matrix   |
| $M_{Desired}$           | Desired coupling matrix   |
| $M_{Extracted}$         | Extracted coupling matrix using our proposed multivalued neural network                               |
| $M_f$                   | Number of poles of the filter   |
| $N$                     | Number of design variables  |
| $N_c$                   | Number of circuits in the multi-Circuit optimization approach   |
| $N_f$                   | Number of frequency samples   |
| $N_v$                   | Number of values of the proposed multivalued neural network inverse model.                            |
| $\mathbf{P}_{SM}$       | Mapping function that relates fine and coarse model design parameters                                 |
| $p$                     | Number of the physical or geometrical parameters in $\bar{\mathbf{y}}$                                |
| $\mathbf{Q}_{SM}$       | Implicit space mapping function that relates fine, coarse model, and preassigned parameter parameters |
| $q$                     | Number of the electrical parameters in $\bar{\mathbf{x}}$   |
| $R(\mathbf{x}, \omega)$ | The response of microwave filters under consideration   |

|                              |   |
|------------------------------|---|
| $\mathbf{R}_c(\mathbf{x}_c)$ | Vector of coarse model responses  |
| $\mathbf{R}_f(\mathbf{x}_f)$ | Vector of fine model responses  |
| $R_f(\mathbf{x}, \omega)$    | Fine model output response  |
| $R_s(\mathbf{x}, \omega)$    | Surrogate model output response   |
| $r$                          | Control index parameter that determines the expansion or contraction of the trust radius for the trust region |
| $\mathbf{s}$                 | Vector representing the step of design variable $\mathbf{x}$ in optimization                                  |
| $\mathbf{s}^{(k)}$           | Optimization step in the $k$ th iteration   |
| $T(\mathbf{w})$              | Test error function of our proposed multivalued neural network inverse model                                  |
| $T_r$                        | Index set of all the training samples   |
| $T_v$                        | Index set of all the test samples   |
| $\mathbf{t}$                 | Vector representing the ripple height parameters  |
| $\mathbf{t}^{(k)}$           | Ripple height parameters in the $k$ th iteration  |
| $\mathbf{t}_a$               | Desired ripple height parameters  |
| $\bar{\mathbf{t}}$           | Average value of $\mathbf{t}^{(k)}$   |

|                         |   |
|-------------------------|---|
| $U$                     | Objective function, typically minimax objective function [19], which represents the error function of $R(\mathbf{x}, \omega)$ with respect to the design specifications |
| $U_i$                   | An $p \times (N_v \cdot p)$ matrix containing zeros everywhere except for the $(h, (i-1) \cdot p + h)$ th location, where the value is 1, for $h = 1, 2, \dots, p$      |
| $V$                     | Index set of valid values   |
| $Var(\mathbf{t}^{(k)})$ | Variance of ripple height parameters $\mathbf{t}^{(k)}$ in the $k$ th iteration   |
| $\mathbf{w}$            | Vector containing neural network weights  |
| $X$                     | Fine model design space   |
| $X_c$                   | Coarse model design space   |
| $\mathbf{x}$            | Vector of design variables  |
| $\mathbf{x}_F$          | A $p$ -vector containing the inputs to the forward model which includes geometrical or physical parameters of a microwave filter  |
| $\mathbf{x}_F^j$        | The physical parameters of $j$ th circuit, $j = 1, 2, \dots, N_c$   |
| $\mathbf{X}_F$          | Matrix which contains all $\mathbf{x}_F^j$ , i.e., $\mathbf{X}_F = [\mathbf{x}_F^1 \ \mathbf{x}_F^2 \ \dots \ \mathbf{x}_F^{N_c}]^T$                                    |
| $\mathbf{x}_f$          | Fine model design parameters  |
| $\mathbf{x}_c$          | Coarse model design parameters  |

|                     |   |
|---------------------|---|
| $\mathbf{x}_f^*$    | Optimal fine model design parameters satisfying the design specifications   |
| $\mathbf{x}_c^*$    | Optimal solution after coarse model optimization  |
| $\mathbf{x}^*$      | Optimal design solution   |
| $\bar{\mathbf{x}}$  | A $q$ -vector containing the inputs to the inverse model, which are usually the electrical parameters   |
| $\mathbf{y}_F$      | A $q$ -vector containing the outputs of the forward model which includes electrical parameters of a microwave filter.   |
| $\bar{\mathbf{y}}$  | A $p$ -vector containing the outputs of the inverse model, which are usually the physical or geometrical parameters; it also represents the inputs of the forward problem |
| $\mathbf{z}$        | A $q$ -vector containing the outputs of the forward model, which are usually the electrical parameters, the same as $\mathbf{y}^F$  |
| $\alpha^j$          | The appropriate weighting factors for the parameters of both $j$ th circuit and the $(j + 1)$ th circuit in multi-circuit optimization approach                           |
| $\Delta \mathbf{f}$ | Difference between the desired frequency vector $\mathbf{f}_d^{(k)}$ and the current actual frequency parameters $\mathbf{f}^{(k)}$                                       |
| $\Delta f_m$        | The $m$ th element in vector $\Delta \mathbf{f}$ , $m = 1, 2, \dots, M_f$   |

|   |  |
|---|--|
| $\Delta \mathbf{R}$                     | Correction vector which is difference between fine model and the original space mapping at the current iteration       |
| $\Delta x_n$                            | Perturbation of $x_n$ in the star distribution, $n = 1, 2, \dots, N$   |
| $\Delta \mathbf{x}_n$                   | Vector containing zero everywhere except the $n$ th element, which is $\Delta x_n$ , $n = 1, 2, \dots, N$              |
| $\Delta^*$                              | Maximum value of the trust radius  |
| $\delta^{(k)}$                          | Trust radius in the $k$ th iteration   |
| $\varepsilon$                           | Predefined constant used in the valid value selection process  |
| $\varepsilon_1$                         | User-defined threshold for the first stage   |
| $\varepsilon_2$                         | User-defined threshold for the second stage  |
| $\varepsilon_t(\mathbf{B}, \mathbf{c})$ | Training error function of the mapping $\mathbf{B}$ and $\mathbf{c}$ between $\mathbf{f}$ space and $\mathbf{t}$ space |
| $\phi(\bar{\mathbf{x}})$                | is a nonlinear function mapping original input $\bar{\mathbf{x}}$ to a higher dimensional space in the SVM technique   |
| $\omega$                                | Frequency  |

# Chapter 1

## Introduction

### 1.1 Introduction

Microwave filters have been used in many types of electronic systems such as mobile communication systems and communication satellites. For given specifications, it takes a complicated procedure for a designer to reach the final design of a microwave filter. The design procedure is usually characterized in five levels: 1. mathematical approximation; 2. L, C, R or coupling matrix; 3. distributed circuits; 4. physical dimensions; 5. optimization [1] - [3].

The last level of optimization usually relies on precise full electromagnetic (EM) simulations of microwave filters. Direct EM optimization can take a large amount of time and is impractical in general. Space mapping techniques have been proven to be very effective for microwave filter optimizations [4]-[11], [26]-[43]. Existing space mapping approaches belong to the class of surrogate-based optimization methods.

In this thesis, a cognition-driven formulation of space mapping that does not re-

quire explicit surrogates is proposed [11]. In the first part of this thesis, the proposed cognition-driven formulation of space mapping method is applied to EM based filter optimization. This new technique utilizes two sets of intermediate feature space parameters, including feature frequency parameters and ripple height parameters. The design variables are mapped to the feature frequency parameters, which are further mapped to the ripple height parameters. By formulating the cognition-driven optimization directly in the feature space, our method increases optimization efficiency and the ability to avoid being trapped in local minima. Several examples have been presented to verify the proposed optimization technique.

Furthermore, parameter extraction of the coupling matrix of microwave filters is also a main topic in the last level of optimization [12]-[16]. The accuracy of the extracted coupling parameters of microwave filters will dominate the speed of computer-aided tuning of microwave filters. Many techniques have been proposed to obtain coupling parameters [1]. One of the techniques is through optimization [2], [92], [93]. Another technique is the inverse modeling technique [1], [17], [18]. Artificial neural network (ANN) inverse modeling technique in [17] is used to extract coupling matrix of microwave filters.

In the second part of this thesis, multivalued neural network inverse modeling technique is proposed [18]. In inverse modeling of a microwave component, the inputs to the model are electrical parameters such as  $S$ -parameters, and the outputs of the model are geometrical or physical parameters. Since the analytical formula of the inverse input-output relationship does not exist, ANN becomes a logical choice

because it can be trained to learn from the data in inverse modeling. The main challenge of inverse modeling is the non-uniqueness problem. This problem in ANN inverse modeling is that different training samples with the same or very similar input values have quite different (contradictory) output values (multivalued solutions). In this part, we propose a multivalued neural network inverse modeling technique to associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. One set of geometrical or physical parameters is called one value of our proposed inverse model. Our proposed multivalued neural network is structured to accommodate multiple values for the model output. We also propose a new training error function to focus on matching each training sample using only one value of our proposed inverse model, while other values are free and can be trained to match other contradictory samples. In this way, our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. Therefore, our proposed technique can solve the non-uniqueness problem in a simpler and more automated way compared to existing ANN inverse modeling techniques. This technique is illustrated by inverse modeling and parameter extraction of four microwave filter examples.

## **1.2 List of Contributions**

The main objective of this thesis is to investigate and develop advanced optimization and inverse modeling techniques for microwave filters. Towards this objective, we

have made several technical contributions in this thesis.

### **1.2.1 Cognition-Driven Formulation of Space Mapping for Equal-Ripple Optimization of Microwave Filters [11]**

In the first part of the thesis, the following contribution were made in the development of new formulation of EM optimization for microwave filters design:

- A cognition-driven formulation of space mapping using two sets of feature parameters was proposed for equal-ripple optimization of microwave filters. The proposed method is applied to EM-based filter optimization. By formulating the cognition-driven optimization directly in the feature space, our method increased optimization efficiency and the ability to avoid being trapped in local minima.
- We formulated a robust algorithm for updating design variables using trust region method to guarantee the convergence of the feature parameters. Using our proposed algorithm, the cognition-driven space mapping optimization technique can achieve equal-ripple responses of microwave filters.

### **1.2.2 Multivalued Neural Network Inverse Modeling and Applications to Microwave Filters [18]**

In the second part of the thesis, the following significant contributions were made for inverse modeling of microwave filters:

- We formulated a new ANN structure, which was the multivalued neural network, for inverse modeling. Our proposed multivalued neural network can associate a single set of electrical parameters with multiple sets of geometrical or physical parameters, therefore allowing contradictory data.
- We proposed a new training algorithm for the multivalued neural network. Using our proposed training error function, our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. We accommodated the non-uniqueness (multivalued) solutions in a simpler and more automated way compared to existing ANN inverse modeling techniques.
- We also presented the procedure of using our proposed multivalued neural network inverse model in the multicircuit ANN process to increase the uniqueness of extracted parameters of microwave filters. The multicircuit ANN process using our proposed multivalued neural network inverse model is more efficient than the conventional multicircuit optimization process.

### 1.3 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 presents a literature review of popularly used optimization techniques and ANN inverse modeling techniques for microwave filters. An overview of classical space mapping methods such as implicit space mapping, output space mapping,

tuning space mapping and the recently developed coarse and fine mesh space mapping is reviewed. Further, an overview of LBE techniques, including SVM, GP, and ANN, is also presented. ANN inverse modeling of microwave filters is also discussed.

Chapter 3 proposes a cognition-driven formulation of space mapping that does not require explicit surrogates. The proposed method is applied to EM based filter optimization. The new technique utilizes two sets of intermediate feature space parameters, including feature frequency parameters and ripple height parameters. The design variables are mapped to the feature frequency parameters, which are further mapped to the ripple height parameters. By formulating the cognition-driven optimization directly in the feature space, our method increases optimization efficiency and the ability to avoid being trapped in local minima.

Chapter 4 presents a multivalued neural network inverse modeling technique to associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. Our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. Therefore, our proposed technique can solve the non-uniqueness problem in a simpler and more automated way compared to existing ANN inverse modeling techniques.

Finally, Chapter 5 presents the conclusions and discussions on possible directions for future work.

## Chapter 2

# Literature Review

Microwave filters play important roles in many applications including mobile and satellite communications, radar, electronics warfare, and remote sensing systems operating at microwave frequencies [1]-[2]. Design optimization of microwave filters usually relies on full electromagnetic (EM) simulations. Space mapping is a recognized method for speeding up EM optimization, and has been applied to microwave filter design [4]-[11]. Parameter extraction of microwave filters plays an important role in computer aided tuning [1]. Inverse modeling techniques can be used in efficient parameter extraction of microwave filters [17]-[18].

Optimization techniques of microwave circuits have been studied over several decades [19]-[43]. Recent advances in optimization methods made the EM based design more feasible for microwave filters [25]-[26]. The strength of the EM analysis, which includes rigorous analysis of general microwave structures, makes the design of microwave filters reliable. However, design optimization of EM structures often requires a massive amount of CPU time to find the optimum design space

parameters. Space mapping is a recognized engineering optimization methodology in the microwave area [4]-[11], [26]-[43]. Coarse models are typically empirical functions or equivalent circuits, which are computationally efficient but demonstrate low accuracy. Fine models can be provided by an EM simulator, which are accurate but computationally intensive. SM establishes a mathematical link between the coarse and fine models and directs the bulk of the CPU-intensive computations to the coarse models, while preserving the accuracy from the fine models [8].

Inverse modeling method has been studied for parameter extraction of microwave filters [17]-[18], and microwave circuit design [44]-[51]. In inverse modeling, the electrical parameters are defined as the inputs to the model and the geometrical or physical parameters are defined as the outputs of the model. Once the inverse model is obtained, it can provide the geometrical parameters immediately without iteratively evaluating the model for given electrical parameters. The formula for the inverse problem, i.e., compute the geometrical parameters from given electrical parameters, is difficult to find analytically. Therefore, learning-by-examples (LBE) methodologies, such as artificial neural network (ANN), become logical choices since the model can be trained to learn from the data of the inverse problem [17]-[18]. The difficulty in the development of the inverse model is the lack of uniqueness in the input-output relationship, that is, different training samples with the same or very similar input values have quite different (contradictory) output values (multivalued solutions).

The optimization technique and inverse modeling technique are important and

related in microwave filter design and tuning [1]. In order to find the geometrical or physical parameters from given electrical parameters, both the optimization techniques and inverse modeling can be used to solve this inverse problem [17], [92].

## 2.1 Optimization of Microwave Filters

Let  $R(\mathbf{x}, \omega)$  denote the response of microwave filters under consideration.  $\mathbf{x} \in \mathfrak{R}^{N \times 1}$  is a vector of  $N$  design variables of microwave filters and  $\omega$  represents the frequency. The original optimization of microwave filters can be formulated as

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in X} U(R(\mathbf{x}, \omega)) \quad (2.1)$$

where  $X \subseteq \mathfrak{R}^N$  represents the design space and  $U$  is a given objective function.  $U$  represents the weighted error function of microwave filter response  $R(\mathbf{x}, \omega)$  and the desired design specifications.  $\mathbf{x}^*$  represents the optimal design vector satisfying the design specifications [19] - [21].

## 2.2 Direct EM Optimization Methods

Direct EM optimization method refers to the existing built-in optimization method in the EM simulators. With the dramatic increase in the computer hardware performance, EM simulator solves Maxwell equations for circuits and arbitrary geometrical shapes. Over the last decade, advances in the technology of workstations and computers have enabled the EM simulators to run the classical optimization

methods on personal computers. The commonly used optimization methods are classified into two categories, global optimization methods and local optimization methods. Global optimization methods such as genetic algorithms [52]-[55], particle swarm optimization [56], simulated annealing [55], find the global solution(s) of a constrained optimization model (generally, in the presence of multiple local optima). Global optimization methods are usually heuristic in their approach for finding the global solution. They are referred to as evolutionary algorithms.

Local optimization methods include gradient based optimization techniques and derivative-free methods. The most popularly used gradient based methods are quasi-Newton method [57]-[60], conjugate gradient method [61]- [64], and sequential non-linear programming [65], [66]. Examples of derivative-free methods are pattern search [67]-[68], and Nelder-Mead method [69]-[70].

### **2.2.1 Pattern Search Method**

Pattern search [67]-[68] belongs to a class of numerical optimization techniques that does not require the gradient of the problem to be optimized. The situation where the pattern search is useful is when the gradient for objective function is not defined at all points.

Pattern search has two moves: exploratory move and pattern move. The exploratory move explores the local behavior and the information of the objective function. The exploratory move is performed until there is an improvement in the value of objective function in each coordinate direction by increasing or decreasing

the step size. Once, the exploratory move is performed in all directions, pattern movement is performed. The pattern move is included to move from the base point along the base line (formed using previous iterations) to a next base point. If the pattern move produces improvement in the value of the objective function, then we proceed to exploratory move for the next iteration, otherwise the step size for the exploratory move in the current iteration is reduced and the process is repeated.

### 2.2.2 Conjugate Gradient Method

The conjugate gradient methods [61]- [64] are simple and easy to implement, and generally they are superior to the steepest descent method. The conjugate gradient method is guaranteed to locate the minimum of any quadratic function of  $N$  arguments in at most  $N$  iterations. For functions which are not quadratic the process is iterative rather than  $N$ -step, and a test for convergence is required [62].

If the number  $N$  of variables is large, then the conjugate gradient methods may be more efficient than Newton-type methods. The reason is that the latter rely on matrix operations, whereas conjugate gradient methods only use vectors. Ignoring sparsity, Newton's method needs  $O(N^3)$  operations per iteration step, quasi-Newton methods need  $O(N^2)$ , but the conjugate gradient methods only use  $O(N)$  operations per iteration step. Similarly for storage: Newton-type methods require an  $N \times N$  matrix to be stored, while conjugate gradient methods only need a few vectors [63].

### 2.2.3 Quasi-Newton Method

Quasi-Newton method is based on Newton's method to find the stationary point of a objective function, where the gradient is zero. Gradient methods (conjugate gradient) assumes that the objective function can be locally approximated as a quadratic function near its optimal point. Newton method uses both first and second order derivatives to find the stationary point using least number of iterations compared to any other gradient method. Newton's method always converges to the minimum point (for a minimizing function). However,

- a) It requires the storing of  $N \times N$  Hessian matrix ( $N$  represents the number of the design variables).
- b) It is difficult to calculate the elements of the Hessian matrix directly.
- c) It requires inversion of the Hessian matrix in each iteration.

To overcome this case, quasi-Newton is developed. In the quasi-Newton method, the inverse of Hessian matrix is approximated using another matrix that is computed using only the first order derivatives of the objective function. There are several methods to approximate the Hessian matrix such as DFP (Davidson-Fletcher-Powell) [59] and BFGS (Broyden-Fletcher-Goldfarb-Shanno) [59] [23] rank-1 and rank-2 updates.

In general, existing EM simulation tools use the latest optimization methods for microwave design. A frequently used approach is the use of gradient optimization to

drive EM simulation for microwave design. These methods are mostly sequential and evaluate the next optimization update using single-point EM evaluation to minimize the CPU time per iteration. One such gradient optimization method available in EM simulation tools is the direct EM optimization approach, which involves many iterations of optimization; therefore it has to perform that many single-point EM evaluations. Space mapping (SM) techniques have been studied to improve the speed of EM optimizations [4]-[11], [27]-[43].

## 2.3 Space Mapping Optimization

Space mapping technique [4]-[11], [26]-[43] is the most popularly used surrogate based optimization technique. SM aims to use fewest possible fine model (EM model) evaluations by exploiting coarse models (e.g., empirical or equivalent circuits) during optimization, thereby increasing the speed of overall optimization. Space mapping (SM) is a recognized engineering optimization paradigm consisting of a number of efficient optimization approaches. The iterative optimization and updating of the surrogate model (coarse model) replace the direct EM optimization of an accurate, but computationally expensive fine model of interest. The formulation of SM optimization algorithm has been comprehensively explored in the recent years [8] and is shown in Fig. 2.1. Space mapping combines the computational efficiency of coarse models with the accuracy of fine models. The coarse models are typically empirical functions or equivalent circuits, which are computationally efficient but the accuracy is low. The fine models can be provided by an EM simulator,

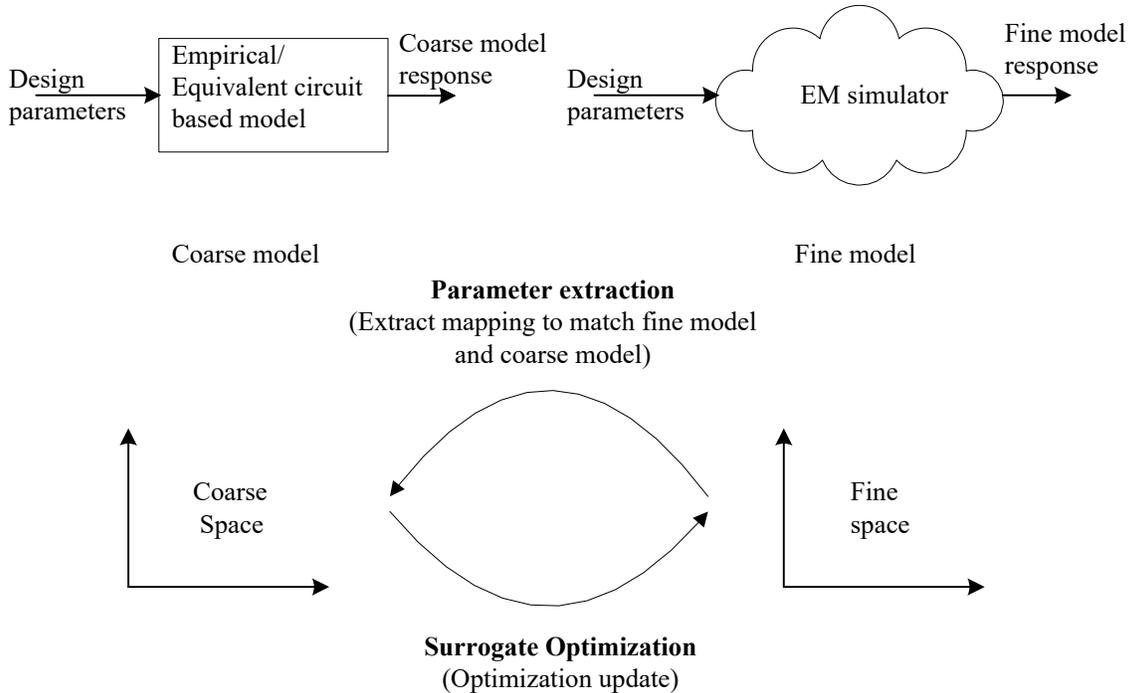


Figure 2.1: Space mapping concept following [43]

which are accurate, but computationally intensive. The space mapping establishes a mathematical link between the coarse and the fine models and directs the bulk of the CPU-intensive computations to the coarse models, while preserving the accuracy from the fine models. The mapped coarse model may be re-optimized to obtain a new design solution.

### 2.3.1 The Space Mapping Concept

The mathematical representation of space mapping methodology presented in [8] is recalled as follows. The coarse and fine model design parameters are denoted by  $\mathbf{x}_c$  and  $\mathbf{x}_f \in \mathfrak{R}^{N_f \times 1}$  respectively. Let  $\mathbf{R}_f(\mathbf{x}_f) \in \mathfrak{R}^{N_f \times 1}$  represent a vector of fine model

responses, e.g.,  $|S_{11}|$  at  $N_f$  selected frequency points  $\omega$ .  $\mathbf{R}_f(\mathbf{x}_f)$  is given by

$$\mathbf{R}_f(\mathbf{x}_f) = \begin{bmatrix} R_f(\mathbf{x}_f, \omega_1) \\ R_f(\mathbf{x}_f, \omega_2) \\ \cdot \\ \cdot \\ R_f(\mathbf{x}_f, \omega_{N_f}) \end{bmatrix} \quad (2.2)$$

Similarly,  $\mathbf{R}_c \in \Re^{N_f \times 1}$  represents a vector of coarse model responses. Space mapping aims to find a mapping function  $\mathbf{P}_{SM}$  that relates the fine and coarse model parameters as

$$\mathbf{x}_c = \mathbf{P}_{SM}(\mathbf{x}_f) \quad (2.3)$$

such that

$$\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f)) \approx \mathbf{R}_f(\mathbf{x}_f) \quad (2.4)$$

in a region of interest.

Space mapping avoids direct EM optimization that requires us to solve (2.1) to find optimal design parameter  $\mathbf{x}_f^*$ . Let  $\tilde{\mathbf{x}}_f$  be a good estimate of  $\mathbf{x}_f^*$  and is given by

$$\tilde{\mathbf{x}}_f = \mathbf{P}_{SM}^{-1}(\mathbf{x}_c^*) \quad (2.5)$$

where  $\mathbf{x}_c^*$  is the result of coarse model optimization.

Space mapping algorithms initially optimize the coarse model to obtain the

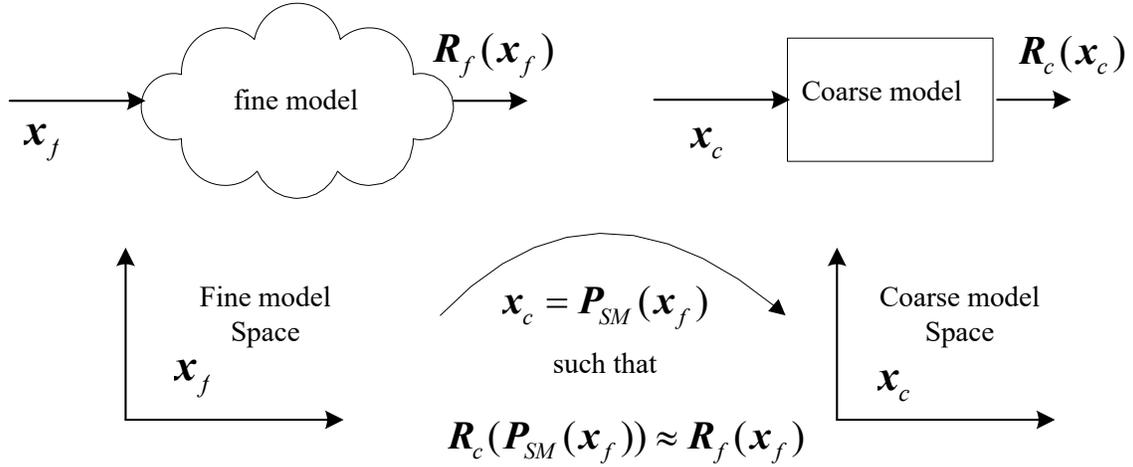


Figure 2.2: Mathematical representation of space mapping following [8]

optimal design  $\mathbf{x}_c^*$ . Subsequently, a mapped solution is found by minimizing the error between the fine model response and the coarse model response defined by

$$\mathbf{P}_{SM}(\mathbf{x}_f) = \arg \min_{\mathbf{x} \in X_c} \|\mathbf{R}_c(\mathbf{x}) - \mathbf{R}_f(\mathbf{x}_f)\| \quad (2.6)$$

where  $X_c$  is the coarse model space. Once the mapping parameters are extracted,  $\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f))$  is optimized to find the solution of the optimization problem defined in (2.1). The surrogate model in this scenario is the coarse model along with mapping parameter  $\mathbf{P}_{SM}$ , i.e.,  $\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f))$ . Thus, the optimization problem is rewritten as

$$\mathbf{x}_f^* = \arg \min_{\mathbf{x}_f \in X} U(\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f))) \quad (2.7)$$

where  $\mathbf{x}_f$  may be close to  $\mathbf{x}_f^*$  if  $\mathbf{R}_c$  is close enough to  $\mathbf{R}_f$ .  $\mathbf{x}_f^*$  obtained from (2.7) is an approximation of fine model optimum. Fig. 2.2 shows the mathematical representation of SM methodology presented in [8].

In the past decade, several advances in space mapping such as aggressive space mapping [5] - [7], implicit space mapping (ISM) [27][28], output space mapping (OSM) [29] - [31], space mapping interpolating surrogates (SMISs) [30], inverse space mapping [32] - [33], have proven successful for difficult optimization problems of microwave filters.

### 2.3.2 Input Space Mapping

Input space mapping [8], [10], [43] aims at reducing misalignment between fine and coarse model by using an affine variable transformation based on the available fine model data. Input space mapping is also called original space mapping wherein Equation (2.3) is represented as

$$\mathbf{x}_c = \mathbf{P}_{SM}(\mathbf{x}_f) = \mathbf{B}_I \mathbf{x}_f + \mathbf{c}_I \quad (2.8)$$

where matrices  $\mathbf{B}_I \in \mathfrak{R}^{N \times N}$  and  $\mathbf{c}_I \in \mathfrak{R}^{N \times 1}$  are obtained using parameter extraction procedure in (2.6).

### 2.3.3 Implicit Space Mapping

Implicit space mapping (ISM) [27]-[29] explores the flexibility of the preassigned parameter such as dielectric constant, substrate height in the design optimization process. Let  $\mathbf{x}_{aux}$  represent the preassigned parameters (auxiliary parameters), then the coarse model response vector is  $\mathbf{R}_c(\mathbf{x}_c, \mathbf{x}_{aux})$ . Therefore, implicit space mapping

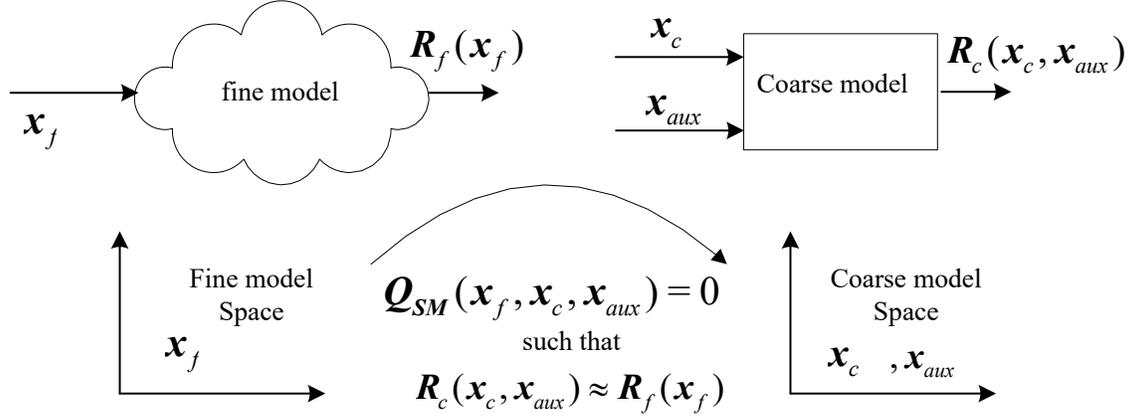


Figure 2.3: Illustration of Implicit Space Mapping (ISM) following [27]

shown in Fig. 2.3 aims at establishing an implicit mapping  $\mathbf{Q}_{SM}$  between the spaces  $\mathbf{x}_f$ ,  $\mathbf{x}_c$  and  $\mathbf{x}_{aux}$ ,  $\mathbf{Q}_{SM}(\mathbf{x}_f, \mathbf{x}_c, \mathbf{x}_{aux}) = 0$  such that  $\mathbf{R}_f(\mathbf{x}_f) \approx \mathbf{R}_c(\mathbf{x}_c, \mathbf{x}_{aux})$  over a region in a parameter space. Implicit mapping ensures to produce a good match between the coarse model and fine model in the first iteration when the input space mapping alone cannot replicate a good match.

### 2.3.4 Output Space Mapping

Output space mapping technique has been proposed [29]-[31]. Output space mapping enhances the surrogate model by a correction term (residual)  $\Delta \mathbf{R}$  which is the difference between the fine model and the original space mapping response at the current iteration so that a perfect match between these two models is ensured (zero-order consistency condition [72]-[74]), i.e.,

$$\Delta \mathbf{R} = \mathbf{R}_f(\mathbf{x}_f) - \mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f)) \quad (2.9)$$

The surrogate model (coarse model along with mapping function) is further enhanced by using the Jacobian of the  $\Delta\mathbf{R}$  (to satisfy first order consistency [72]-[74] between the surrogate model and fine model at the current design). If the misalignment between the fine and coarse models is not significant, SM-based optimization algorithms typically provide excellent results after only a few evaluations of the fine model.

### **2.3.5 Tuning Space Mapping**

Tuning space mapping (TSM) [34]-[37] is a special type of SM technique that caters to tuning of EM structures. The surrogate model is replaced by a tuning model which introduces circuit components into the fine model structure. The tuning model is optimized within a circuit simulator. With the optimal tuning parameters, thus obtained, they are mapped or transformed into the design variables using fast space-mapping surrogate or analytical formulas if available. Tuning models require a significant engineering expertise for a successful implementation of the optimization process using TSM approach.

### **2.3.6 Coarse and Fine Mesh Space Mapping**

In coarse and fine mesh SM method [39]-[42], the fine model uses a fine mesh for EM simulation and the coarse model uses coarse mesh for EM simulation. The accuracy of the fine model is achieved by the mesh convergence process. The EM simulation and mesh refinement are performed iteratively until the simulation results

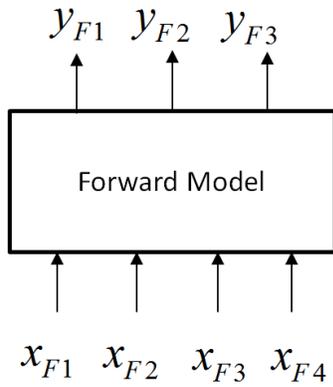
between successive iterations converge. The fine mesh simulation is accurate yet computationally expensive. The coarse model uses a coarse mesh EM simulation which only needs local meshing in EM simulation without the mesh convergence process. Also, the coarse model is computationally fast but less accurate. This method is useful when equivalent coarse model for the EM problem is not available, and when a conventional space mapping is not applicable.

Space mapping techniques aim to use fewest possible fine model (EM model) evaluations by exploiting coarse models (e.g., empirical or equivalent circuits) during optimization, thereby increasing the speed of overall optimization. Recent advances in space mapping such as three level output space mapping [31], constrained parameter extraction using implicit space mapping [75], space mapping optimization using EM-based adjoint sensitivity [76], and fast EM modeling using shape-preserving response prediction [77].

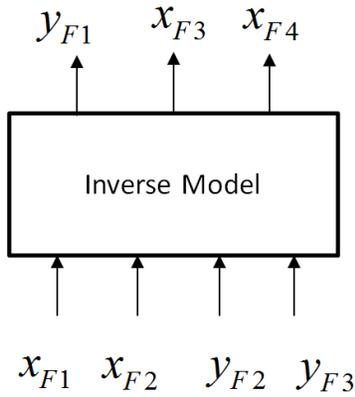
## 2.4 Inverse Modeling of Microwave Filters

### 2.4.1 Forward Model

In the forward problem of a microwave filter, the computation is from geometrical or physical parameters to the electrical parameters. Let  $p$  and  $q$  represent the number of inputs and outputs of the forward model. Let  $\mathbf{x}_F$  be a  $p$ -vector containing the inputs which includes geometrical or physical parameters of a microwave component, and  $\mathbf{y}_F$  be a  $q$ -vector containing the outputs of the forward model which includes electrical parameters of a microwave filter. The forward modeling problem can then



(a)



(b)

Figure 2.4: Example illustrating forward and inverse models. (a) Forward model. (b) Inverse model. The inputs  $x_{F3}$  and  $x_{F4}$  (output  $y_{F2}$  and  $y_{F3}$ ) of the forward model are swapped to the outputs (inputs) of the inverse model, respectively.

be expressed as

$$\mathbf{y}_F = \mathbf{f}_{FWD}(\mathbf{x}_F) \quad (2.10)$$

where  $\mathbf{x}_F = [x_{F1} \ x_{F2} \ \cdots \ x_{Fp}]^T$ ,  $\mathbf{y}_F = [y_{F1} \ y_{F2} \ \cdots \ y_{Fq}]^T$ , and  $\mathbf{f}_{FWD}$  defines the input and output relationship of the forward problem of a microwave filter [17]. An example of a forward model and its corresponding inverse model are shown in Fig. 2.4. Note that two outputs and two inputs of the forward model are swapped to the input and output of the inverse model, respectively. In general, some or all of them can be swapped from input to output or vice versa.

## 2.4.2 Inverse Model

Let  $\bar{\mathbf{x}}$  be defined as a  $q$ -vector containing the inputs to the inverse model, which usually includes the electrical parameters of a microwave filter.  $\bar{\mathbf{y}}$  is defined as the outputs of the inverse model, which usually includes the geometrical or physical parameters of a microwave filter..

$$\bar{\mathbf{y}} = \mathbf{f}_{FWD}^{-1}(\bar{\mathbf{x}}) \quad (2.11)$$

where  $\mathbf{f}_{FWD}^{-1}$  represents the input-output relationship of the inverse problem. For the example, in Fig. 2.4, the inverse model is formulated as

$$\begin{aligned} \bar{\mathbf{x}} &= [\bar{x}_1 \ \bar{x}_2 \ \bar{x}_3 \ \bar{x}_4]^T = [x_1^F \ x_2^F \ y_2^F \ y_3^F]^T \\ \bar{\mathbf{y}} &= [\bar{y}_1 \ \bar{y}_2 \ \bar{y}_3]^T = [y_1^F \ x_3^F \ x_4^F]^T \end{aligned} \quad (2.12)$$

The inverse model can be trained with the data. Usually data are generated by EM solvers originally in a forward way.

The inverse problem is necessary to be addressed for the purposes of microwave filter tuning. The inverse problem is the situation where the information is processed in the reverse direction in order to find the geometrical/physical parameters for given values of electrical parameters. There are usually two types of method to solve the inverse problem, i.e., the optimization method and inverse modeling method.

## 2.5 Optimization Method for Inverse Problems of Microwave Filters

The optimization method for inverse problems is to apply the optimization method on the forward model for design, modeling, and diagnosis of microwave filters [91], [92], [93]. This method is to find the solution of geometrical parameters or coupling parameters of microwave filters for given  $S$ -parameters (measurement data) as the target. It is to compare the target  $S$ -parameters (measurement data) to the  $S$ -parameters from a model at many frequency samples. A cost function can be defined as the difference between the target  $S$ -parameters and the  $S$ -parameters from the model at those frequency samples. Then an optimization method is applied to change geometrical or coupling parameters iteratively to reduce this cost function, i.e., to match the target  $S$ -parameters using the  $S$ -parameters from the model [1]. One of the difficulty of the optimization method for inverse problems is the non-uniqueness of the solution (geometrical parameters or coupling parameters). To improve the uniqueness, the multircuit optimization methods are proposed [92], [93].

An enhanced multicircuit optimization approach for identification of microwave filter parameters has been presented in [93]. The approach is based on  $l_1$  norm optimization [92]. It strives to force the modeled response to approximate measured response and at the same time minimize the difference of parameter values between different circuits. The proposed method can not only achieve an excellent match between the measured and the modeled responses, but also enhance the uniqueness of the identified parameters. In this technique, the microwave filter which we solve the inverse problem for is taken as the 1st circuit, and then one of the physical parameters is adjusted to obtain the 2nd circuit. Similarly, we adjust a physical parameter of the  $j$ th circuit to obtain the  $(j+1)$ th one,  $j = 1, 2, \dots, N_c - 1$ . So that we can get  $N_c$  circuits. Let  $\mathbf{x}_F^j$  be defined as the physical parameters of  $j$ th circuit. Let  $\mathbf{X}_F$  represent a matrix which contains all  $\mathbf{x}_F^j$ , i.e.,  $\mathbf{X}_F = [\mathbf{x}_F^1 \ \mathbf{x}_F^2 \ \dots \ \mathbf{x}_F^{N_c}]^T$ .

Let  $\mathbf{f}_m(\mathbf{x}_F^j)$  represent the measured/simulated response of the  $j$ th circuit and  $\mathbf{f}_{FWD}(\mathbf{x}_F^j)$  represent the  $j$ th response calculated from the approximate model.

The multi-circuit optimization method for parameter extraction can be formulated as

$$\underset{\mathbf{X}_F}{\text{minimize}} \left( \sum_{j=1}^{N_c} |\mathbf{f}_{FWD}(\mathbf{x}_F^j) - \mathbf{f}_m(\mathbf{x}_F^j)| + \sum_{j=1}^{N_c-1} |\boldsymbol{\alpha}^j (\mathbf{x}_F^{j+1} - \mathbf{x}_F^j)| \right) \quad (2.13)$$

where  $\boldsymbol{\alpha}^j$  represents the appropriate weighting factors for the parameters of both  $j$ th circuit and the  $(j+1)$ th circuit.  $\mathbf{X}_F$  represents a matrix which contains all  $\mathbf{x}_F^j$ , i.e.,  $\mathbf{X}_F = [\mathbf{x}_F^1 \ \mathbf{x}_F^2 \ \dots \ \mathbf{x}_F^{N_c}]^T$ . The optimization for parameter extraction in (2.13) is to find an set of physical parameters  $\mathbf{x}_F^j$ ,  $j = 1, 2, \dots, N_c$ , such

that the modeled responses  $\mathbf{f}_{FWD}(\mathbf{x}_F^j)$  approximate measured responses  $\mathbf{f}_m(\mathbf{x}_F^j)$  and at the same time the difference of parameter values between different circuits are minimized.

The first part of the objective function aims at matching the calculated responses of all circuit models with their corresponding measured/simulated responses. The second part is added to penalize any differences between  $\mathbf{x}_F^j$  and  $\mathbf{x}_F^{j+1}$  so that after optimization most physical parameters of  $\mathbf{x}_F^j$  and  $\mathbf{x}_F^{j+1}$  are almost the same except for those affected by the adjustment. In this way, the uniqueness of the optimal solution can be improved. This optimization approach can improve the uniqueness of the filter parameters in parameter identification process of microwave filters.

## 2.6 Learning-By-Examples (LBE) Methodologies Techniques

In inverse modeling of microwave filters, the electrical parameters are defined as the inputs to the model and the geometrical or physical parameters are defined as the outputs of the model. Once the inverse model is obtained, it can provide the geometrical parameters immediately without iteratively evaluating the model for given electrical parameters.

Several works have been done in inverse modeling. In the direct ANN inverse modeling technique [44], an ANN model is trained using data obtained by swapping the input and output data from the forward problem. In [45] - [47], inverse surrogate modeling is used for expedited geometry scaling of compact microwave passives.

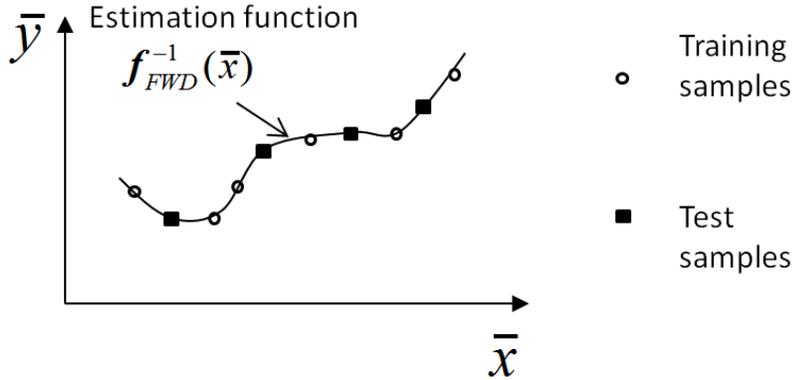


Figure 2.5: Illustration of LBE-based techniques

In [48] and [49], inverse modeling is used for rapid EM-driven antenna dimension scaling. In [50], ANN inverse modeling is applied in a microwave sensor which is used to determine the proportion of fluids in a pipeline. In [51], ANN inverse modeling is used in transmitarray antenna design.

The inverse modeling method is much faster than the conventional iterative optimization process, and can afford to be used in repetitive designs or parameter extraction process. However, the analytical function or formula of the inverse model is difficult to obtain. Therefore, learning-by-examples (LBE) techniques have been used in learning the relationship between inputs and outputs in the inverse problem.

There is a large number of problems in microwave engineering that require a fast response or in which the input-output relationship is not a-priori known or cannot be defined due to the complexity of the scenario at hand. These issues have induced researchers toward the development and application of Learning-by-Examples (LBE) techniques thanks to their extremely high computational efficiency

and to their capability to emulate the behavior of complex systems on the basis of a set of collected examples [94]. The principal LBE techniques applied to EM engineering problems are the artificial neural network (ANN), the support vector machine (SVM), and the gaussian processes (GP).

The training of a LBE-based inverse modeling technique, aimed at building the inverse model  $\mathbf{f}_{FWD}^{-1}(\bar{\mathbf{x}})$  mapping the input to the output space and at emulating the behavior of a real system, is conducted offline and before using the LBE algorithm for solving (online) the problem of interest during the testing phase. Fig. 2.5 shows a illustration of the LBE-based modeling techniques. For given training samples and selected LBE technique, LBE-based modeling technique is to define the estimation function  $\mathbf{f}_{FWD}^{-1}(\bar{\mathbf{x}})$ , which represents the inverse behavior of the real system (training samples) in order to determine the response for a specific input  $\bar{\mathbf{x}}$  during test.

### 2.6.1 Support Vector Machine (SVM)

The SVM-based procedures allow to find a trade-off between the capability of learning from the training samples and the complexity of the surrogate model [95]. As a matter of fact, SVM are built on a solid theoretical framework, the statistical learning theory, in which the definition of the control parameters is formulated as a quadratic optimization problem ensuring a global optimum [96]. Moreover, the resulting model turns out being sparse, since only training samples associated to non-vanishing coefficients (i.e. the so-called 'support vectors') are exploited to make predictions, thus controlling the model complexity and avoiding overfitting [94].

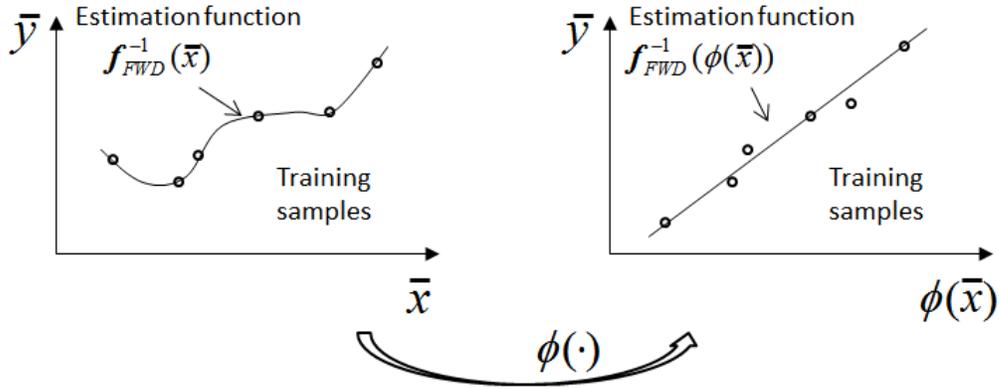


Figure 2.6: Illustration of the SVM technique.

In the SVM technique, the estimation function is given by

$$\mathbf{f}_{FWD}^{-1}(\bar{\mathbf{x}}) = \mathbf{a} \cdot \phi(\bar{\mathbf{x}}) + \mathbf{b} \quad (2.14)$$

which represents a hyperplane,  $\mathbf{a}$  represents the unknown normal vector and  $\mathbf{b}$  represents unknown bias coefficients.  $\phi(\bar{\mathbf{x}})$  is a nonlinear function mapping original input  $\bar{\mathbf{x}}$  to a higher dimensional space. Fig. 2.6 shows an illustration of the SVM technique.

## 2.6.2 Gaussian Processes (GP)

A Gaussian process is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions [97]. The GP techniques can build surrogate models which provide the

approximation error (i.e. a value of uncertainty/confidentiality) of the predictions over the whole input space without the need of testing samples [94][97].

### **2.6.3 Artificial Neural Network (ANN)**

In recent years, neural network or artificial neural network (ANN) techniques have been recognized as useful alternatives to conventional approaches in microwave optimization and modeling [78]-[90]. ANNs can be used to develop new models or to enhance the accuracy of existing models. They learn device data through an automated training process, and the trained neural networks are then used as fast and accurate models for efficient high-level circuit and system design. These models have the ability to capture multidimensional arbitrary nonlinear relationships. The theoretical basis of neural networks is based on the universal approximation theory [98], which states that a neural network with at least one hidden layer can approximate any nonlinear continuous multidimensional function to any desired accuracy. This makes neural networks a useful choice for device modeling where a mathematical model is not available. The evaluation from input to output of a neural network model is also very fast. Fig. 2.7 shows a three-layer ANN structure for inverse modeling.

## **2.7 ANN Inverse Modeling of Microwave Filters**

An artificial neural network trained to model microwave filter problems can be called the forward model where the model inputs are physical or geometrical parameters

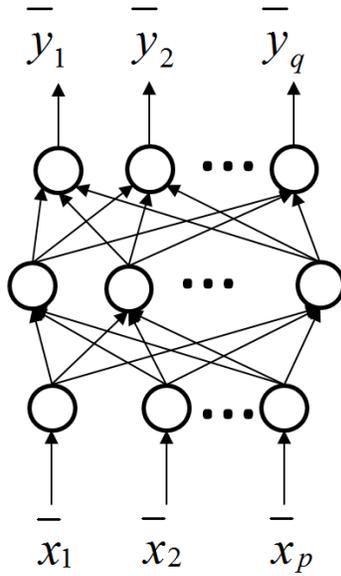


Figure 2.7: Illustration of a three-layer ANN structure.

and outputs are electrical parameters. For design purposes, the information is often processed in the reverse direction in order to find the geometrical/physical parameters for given values of electrical parameters, which is called the inverse problem.

### 2.7.1 Direct ANN Inverse Modeling

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

geometrical parameters. Training data for the neural network inverse model can be obtained simply by swapping the input and output data used to train the forward model. This method is called the direct inverse modeling and an example of this approach is [44]. Once training is completed, the direct inverse model can provide inverse solutions immediately unlike the optimization method where repetitive forward model evaluations are required. Therefore, the direct inverse model is faster than the optimization method using either the EM or the neural network forward model.

Even though the inverse modeling technique is much faster in the design process or parameter extraction process, the model is not easy to be trained well in many practical situations. It is because of the non-uniqueness problem of the inverse input-output relationship. The non-uniqueness problem means that different training samples with the same input values have contradictory output values (multivalued solutions). An even more challenging case is that different samples with similar but not identical input values have much different output values. This will cause the ambiguity whether these samples are contradictory or not. Since it is not possible to train an ANN model to match contradictory output values simultaneously for the same input values, the training error will stay large and the model accuracy will be low. The multivalued problem in hysteresis modeling has been studied in [99] from ANN point of view. The one-input multivalued data with hysteresis phenomena was represented by introducing a new parameter to the inputs of the model so that the one-input multivalued data becomes two-input single valued data. The non-uniqueness (multivalued) problem was addressed in [17] in microwave area.

## 2.7.2 Neural Network Inverse Modeling Technique in [17]

In the neural network inverse modeling technique [17], the data containing contradictory samples are divided into groups according to derivative information using a neural network forward model such that individual groups do not have the problem of contradictory data. Multiple inverse models are built based on divided data groups, and are then combined to form a complete model. This technique can be summarized as follows:

Step 1: Define the inputs and outputs of the model. Generate data using simulator or measurement. Swap the input and output data to obtain data for training inverse model. Train and test the inverse model. If the model accuracy is satisfied, then stop. Results obtained here is the direct ANN inverse model.

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

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

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

Step 5: Check for multivalued solutions in models training data. If none are found, then perform further segmentation by going to Step 2.

Step 6: Train a neural network forward model.

Step 7: Using the adjoint neural network of the forward model, divide the training data according to derivative criteria.

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

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

## **2.8 Summary**

In this chapter, a literature review of popularly used optimization techniques and ANN inverse modeling techniques for microwave filters has been presented. An overview of classical space mapping methods such as implicit space mapping, output space mapping, tuning space mapping and the recently developed coarse and fine mesh space mapping is reviewed. Further, an overview of LBE techniques, including SVM, GP, and ANN, is also presented. ANN inverse modeling of microwave filters is also discussed.

## Chapter 3

# Cognition-Driven Formulation of Space Mapping for Equal-Ripple Optimization of Microwave Filters

Space mapping is a recognized method for speeding up electromagnetic (EM) optimization. Existing space mapping approaches belong to the class of surrogate-based optimization methods. This chapter proposes a cognition-driven formulation of space mapping that does not require explicit surrogates. The proposed method is applied to EM based filter optimization. The new technique utilizes two sets of intermediate feature space parameters, including feature frequency parameters and ripple height parameters. The design variables are mapped to the feature frequency parameters, which are further mapped to the ripple height parameters. By formulating the cognition-driven optimization directly in the feature space, our method increases optimization efficiency and the ability to avoid being trapped in local minima. This technique is illustrated by two microwave filter examples.

## 3.1 Introduction

Space mapping is a recognized engineering optimization methodology in the microwave area [4]-[11], [26]-[43]. The space mapping concept combines the computational efficiency of coarse models with the accuracy of fine models [4]. Coarse models are typically empirical functions or equivalent circuits, which are computationally efficient but the accuracy is low. Fine models can be provided by an EM simulator, which is accurate, but computationally intensive. Space mapping establishes a mathematical link between the coarse and fine models and directs the bulk of the CPU-intensive computations to the coarse models, while preserving the accuracy from the fine models [43]. Recent progress has focused on several areas, such as a recent review points towards a cognition interpretation of space mapping [100], implicit space mapping [27], [28], output space mapping [29], [31], neural space mapping [87]-[90], generalized space mapping [102], tuning space mapping [36]-[37], portable space mapping for efficient modeling [103], inverse space mapping [32] - [33], shape-preserving response prediction [104], parallel space mapping [10] and zero-pole space mapping [105]. A software implementation of space mapping such as the SMF framework with applications such as antenna design have also been described in the literature [106].

In practical cases, equivalent circuit coarse models are not always available [40]. Some effort has been focused on this situation. In [107], to build a coarse model for waveguide filters, a small number of accessible modes in the method of moments are considered to obtain a faster simulation at the expense of solution accuracy.

In the work of [39]-[42], coarse and fine mesh EM simulations are used to enable space mapping EM optimization. Sensitivity information from EM simulations has been used to increase the effectiveness of space mapping [41], [108]-[109]. The convergence speed in this case is affected by the difference between fine and coarse mesh EM simulations, and the continuity of the coarse mesh EM response w.r.t design variables.

The technique in this chapter is a significant advance over the work of [39], in an effort to address the challenge of space mapping when explicit equivalent circuit coarse models are not available. We exploit the concept of feature parameters to assist the space mapping, as opposed to use of coarse-mesh EM in [39]. Several recent works have investigated possible feature parameters in model responses. In [2], using the differences between return loss and transmission loss at maxima in the passband and minima in the stopband as the objective function, the coefficients of the characteristic polynomial for a filter are optimized to reach equiripple passband and stopband responses. Theories for the synthesis of multiple coupled resonator filters go back many years [110]-[111], and zeros and poles of filter transfer functions are used as feature parameters for optimization in [105] and [112]. In [113]-[116], feature parameters of filter responses are used for SIW filter tuning and statistical analysis of microwave structures. In [117], the response features are utilized to develop variable-fidelity feature-based modeling.

This chapter aims to explore the use of intermediate feature space parameters in SM. We propose a cognition-driven formulation of space mapping for equal-ripple

optimization of microwave filters. It is suitable for EM-based design of Chebyshev- and elliptic-type responses. The proposed SM can proceed with neither explicit coarse models nor explicit surrogate models. In our method, the intermediate feature space parameters, including the feature frequency parameters and ripple height parameters, are used to set up two new kinds of space mapping. The design variables are mapped to feature frequency parameters, which are further mapped to the ripple height parameters, thus the concept of SM in our optimization. By formulating the cognition-driven optimization directly in the feature space, our method can increase optimization efficiency and ability to avoid being trapped in local minima. This technique is illustrated by two microwave bandpass filter examples.

We think of our technique as “cognitive” [100] in the sense that a meaningful coarse or surrogate model is implied by the engineers intuition and experience.

## 3.2 Original Optimization Problem

Let  $R(\mathbf{x}, \omega)$  denote the response corresponding to a vector of design variables  $\mathbf{x}$  and frequency  $\omega$ . The original optimization problem is formulated as follows

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} U(R(\mathbf{x}, \omega)) \quad (3.1)$$

where  $U$  is a suitable objective function, typically minimax objective function [19], which represents the error function of  $R(\mathbf{x}, \omega)$  with respect to the design specifications;  $\mathbf{x}^*$  is the optimal design to be found.

The convergence of optimization is efficient when the objective functions are

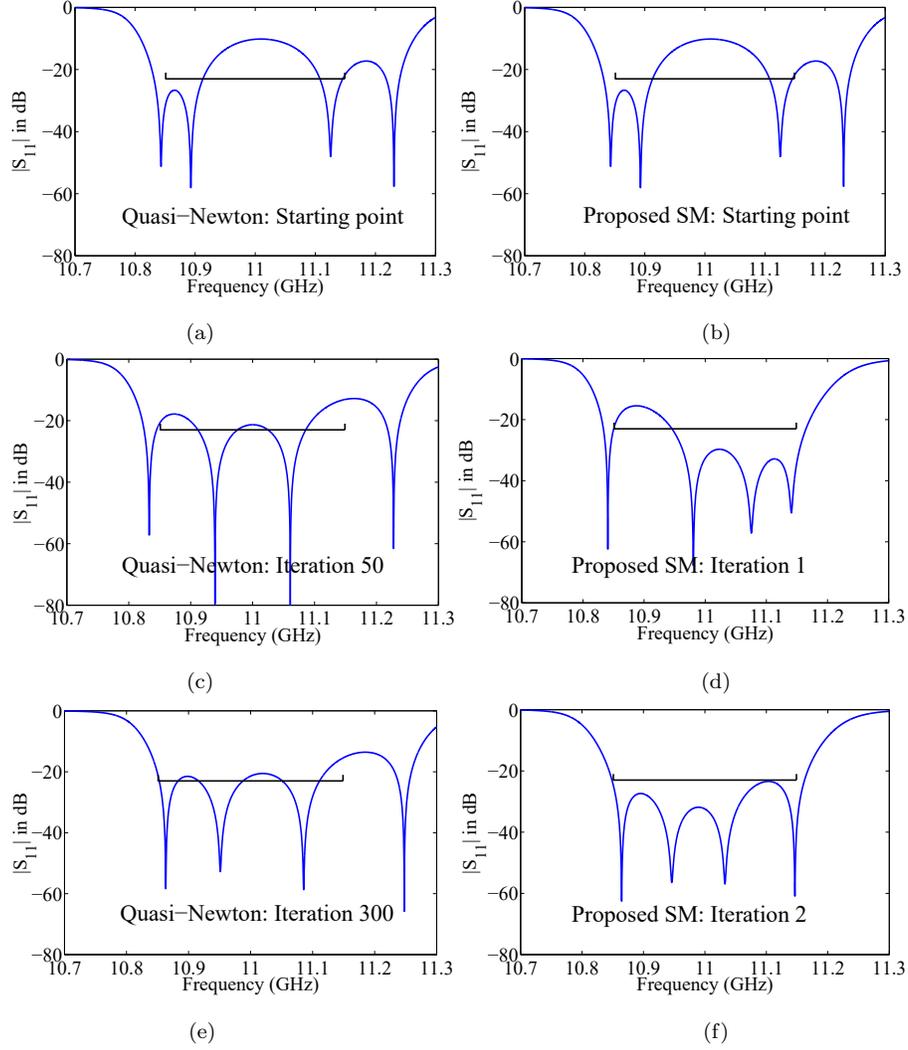


Figure 3.1: The optimization of the four-pole waveguide filter: (a), (c), and (e) depicts responses during quasi-Newton optimization iterations, where only 3 poles are forced to contribute passband response: (a) Initial, (c) Iteration 50, (e) Iteration 300; Subfigures (b), (d), and (f) depicts proposed cognition-driven space mapping iterations. (d) proposed method adjusts the frequency locations of reflection zeros to the passband in the first stage, (f) then adjusts the frequency locations of reflection zeros according to the ripple height parameters. One iteration of our proposed method includes multiple EM simulations in parallel. The total computation time for one iteration in our proposed method is only incrementally more than that of a single EM simulation in one quasi-Newton optimization iteration.

relatively smooth and/or the initial values are of good quality. However, when the objective functions contain traps of local minima and the initial values are not within the neighborhood of a good solution, gradient-based optimization may easily be trapped into local solutions. An example is the filter design shown in Fig. 3.1(a), 3.1(c), 3.1(e). As shown in the figures, because the reflection zeros of the 4-pole filters are not all within the specification range, the optimization process will force only 3 poles to contribute to the filter passband response, while the remaining 1 pole is excluded from helping the passband response. Suitable optimization techniques, such as non-gradient based methods, may help alleviate some of the above mentioned problems, usually at the cost of longer CPU time. In this chapter, we explore an efficient approach to solve the problem using feature parameters and formulations inspired by human design intuition. The direct optimization problem is changed into a different optimization problem in feature space. Because of this change, our method takes only 2 iteration to achieve a optimization result in feature space shown in Fig. 3.1(f). Compared to the result obtained using 300 quasi-Newton iterations in direct optimization method shown in Fig. 3.1(e), our method can achieve the same purpose of filter design.

## 3.3 Proposed Space Mapping Technique for Microwave Optimization

### 3.3.1 Feature Space Parameters

Our cognition-driven approach is motivated by the intuitive (cognitive) design process of the experienced filter designer. The designer would firstly adjust the frequency locations of reflection zeros relative to the passband, rather than trying to push the  $S$ -parameter values in the initial design stage. In sub-sequent design stages, the designer would adjust the ripple height using the fact that making two frequency locations of reflection zeros closer (further apart) will reduce (increase) the height of the passband ripple in the frequency response curve. This process is illustrated in Fig. 3.1(b), 3.1(d), and 3.1(f). By adopting such a concept, we reformulate the design optimization by introducing new feature parameters for the design, i.e., we define a new feature parameter space, called the feature frequency space as follows.

For an equal-ripple bandpass filter, the filter response curve (e.g.,  $S_{11}$  versus frequency) has several minima which are referred to as feature frequencies, and several maximums which are referred to as ripples. The feature frequencies correspond to the reflection zeros at which the filter has maximum transmission. We propose to use feature frequencies as feature space parameters for space mapping. For example, Fig. 3.2(a) shows a four-pole waveguide filter, and Fig. 3.2 (b) shows  $S_{11}$  in dB of this filter, where the feature frequencies  $\mathbf{f} = [f_1 \ f_2 \ f_3 \ f_4]^T$  are important features of the  $S_{11}$  response curve and are used by our technique as intermediate feature

space for a new formulation of space mapping. We perform space mapping between physical/geometrical design variables  $\mathbf{x}$  (i.e., the original optimization variables) and the feature frequency parameters  $\mathbf{f}$ .

The maximum values of  $S_{11}$  in dB between these feature frequency points are also important features of the  $S_{11}$  response curve and are represented by a new set of feature parameters called ripple height parameters  $\mathbf{t} = [t_1 \ t_2 \ t_3]^T$  similar to that used in [113][116].

In our proposed formulation of space mapping, we obtain the feature frequency parameters  $\mathbf{f} = [f_1 \ f_2 \ f_3 \ f_M]^T$  and the ripple parameters  $\mathbf{t} = [t_1 \ t_2 \ t_{M-1}]^T$  from the results of an EM simulation, where  $M_f$  represents the number of poles of the filter.

At the beginning of our proposed method, we generate multiple sample points with star distribution around the current solution point  $\mathbf{x}^{(k)}$  in the  $k$ th iteration of space mapping. We perturb  $\mathbf{x}^{(k)}$  twice along each dimension, once towards the positive direction, and once towards the negative direction. In this way, we find  $2N$  points of response, where  $N$  is the number of design variables. Let  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{2N+1}$  represent the  $2N + 1$  points of the star distribution with  $\mathbf{x}_1$  as the center, i.e.,  $\mathbf{x}_1 = \mathbf{x}^{(k)}$ , and the remaining  $2N$  points in the neighborhood of the center. When the optimization process moves to the next iteration, the center of the star distribution moves from  $\mathbf{x}^{(k)}$  to  $\mathbf{x}^{(k+1)}$ . We perform EM simulations at all the  $2N + 1$  data points to obtain the responses  $R(\mathbf{x}_i, \omega)$  at  $\mathbf{x}_i$ , using  $2N + 1$  processors in parallel [10], and subsequently obtain  $\mathbf{f}_i$  and  $\mathbf{t}_i$ , for  $i = 1, 2, \dots, 2N + 1$ . In this way, the basic data

available in the  $k$ th iteration includes the feature frequency parameters  $\mathbf{f}^{(k)}$  and the ripple height parameters  $\mathbf{t}^{(k)}$ , which equal  $\mathbf{f}_1$  and  $\mathbf{t}_1$ , respectively. Because our algorithm uses parallel computation to perform the  $2N + 1$  EM simulations simultaneously, the total computation time for the  $2N + 1$  EM simulations is similar to (or only incrementally more than) that of a single EM simulation.

### 3.3.2 Mapping from $\mathbf{f}$ Space to $\mathbf{x}$ Space

With this basic data, we can build the mapping  $\mathbf{F}$  from the design variables  $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^T$  to the feature frequency parameters  $\mathbf{f}$  as follows

$$\mathbf{f} = \mathbf{F}(\mathbf{x}) \quad (3.2a)$$

$$\mathbf{A} = \partial \mathbf{F}(\mathbf{x}) / \partial \mathbf{x} \quad (3.2b)$$

$\mathbf{A}$  is the Jacobian matrix of  $\mathbf{F}$ . We evaluate the  $\mathbf{A}$  matrix using the EM solutions over the  $2N + 1$  star distribution points as

$$\begin{aligned} [\mathbf{A}]_{m,n} &= [F_m(\mathbf{x} + \Delta \mathbf{x}_n) - F_m(\mathbf{x} - \Delta \mathbf{x}_n)] / 2\Delta x_n, \\ m &= 1, 2, \dots, M_f; \quad n = 1, 2, \dots, N \end{aligned} \quad (3.3)$$

where  $F_m$  is the  $m$ th element of  $\mathbf{F}$ , and  $\Delta x_n$  is the perturbation of  $x_n$  in the star distribution.  $\Delta \mathbf{x}_n$  is defined as a vector containing zero everywhere except the  $n$ th element, which is  $\Delta x_n$ , i.e.,

$$\Delta \mathbf{x}_n = [0 \ \cdots \ \overset{(n)}{\Delta x_n} \ \cdots \ 0]^T, n = 1, 2, \dots, N \quad (3.4)$$

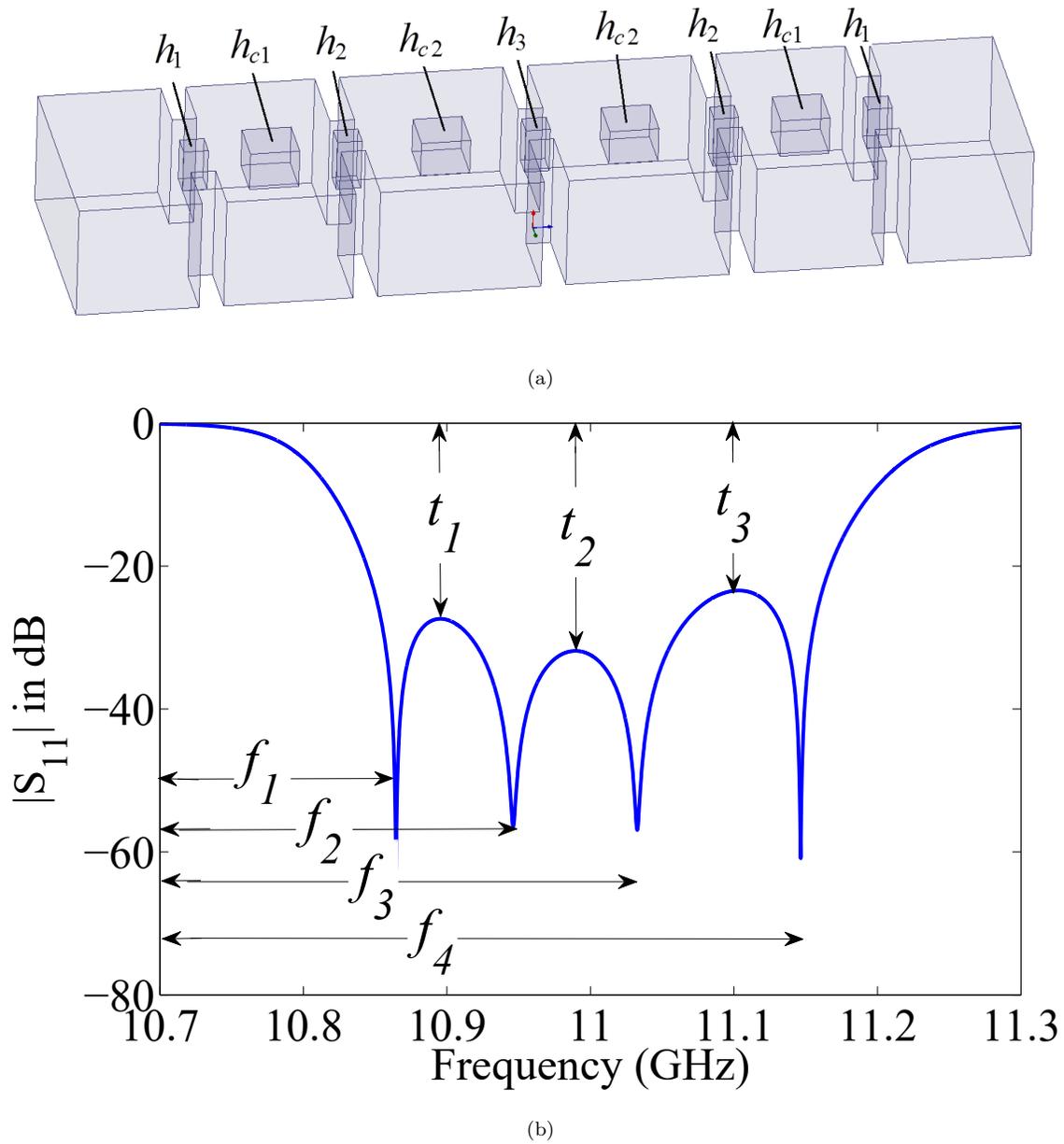


Figure 3.2: The four-pole waveguide filter. (a) the simulation structure, (b) the proposed feature parameters  $t$  and  $f$  in the filter response.

Compared to the relationship between the objective function and  $\mathbf{x}$  space in the direct optimization problem, the mapping from  $\mathbf{f}$  space to  $\mathbf{x}$  space is more linear and valid in a larger range. If the mapping from  $\mathbf{f}$  space to  $\mathbf{x}$  space is not linear in a given range, our proposed trust region method will shrink the trust radius such that the linear mapping is still valid.

We incorporate the trust region mechanism [118] into our formulation of optimization. When this mapping  $\mathbf{F}$  is built, we can solve for the step  $\mathbf{s}^{(k)}$  in the  $k$ th iteration by

$$\mathbf{s}^{(k)} = \arg \min_{\|\mathbf{s}\| \leq \delta^{(k)}} \left\| \mathbf{f}_d^{(k)} - \mathbf{f}^{(k)} - \mathbf{A}^{(k)} \mathbf{s} \right\| \quad (3.5)$$

where  $\delta^{(k)}$  is the trust radius,  $\mathbf{f}_d^{(k)}$  represents the desired frequency vector and  $\mathbf{A}^{(k)}$  is the Jacobian matrix of  $\mathbf{F}$  in the  $k$ th iteration.

### 3.3.3 Stage 1: Adjustment of Locations of Feature Frequencies

In the 1st stage, the passband specified for the filter is divided into  $(M_f - 1)$  equal parts. The desired feature frequency parameters  $\mathbf{f}_d^1$  in the first stage are designated as

$$\mathbf{f}_d^1 = [ f_L \quad f_L + \frac{f_H - f_L}{M_f - 1} \quad \cdots \quad f_L + \frac{f_H - f_L}{M_f - 1} (M_f - 2) \quad f_H ]^T \quad (3.6)$$

where  $f_L$  and  $f_H$  mean the lower and higher frequency edge of the filter passband, respectively. Both  $f_L$  and  $f_H$  are constants determined a priori according to the desired filter passband. Now we can calculate  $\mathbf{s}^{(k)}$  by solving Equ. 3.5 using  $\mathbf{f}_d^{(k)} = \mathbf{f}_d^1$  for the first stage. Once  $\mathbf{s}^{(k)}$  is determined, we generate  $2N + 1$  sample points

with star distribution in parallel around the new center  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ , and perform these  $2N + 1$  EM simulations in parallel. The feature frequency parameters  $\mathbf{f}_s^{(k)}$  and the ripple height parameters  $\mathbf{t}_s^{(k)}$  are determined at  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ . Then we calculate the adjustment index  $r$  for the trust radius as part of the trust region procedure. If the dimensions of feature space parameters change,  $r$  is set to be  $-1$ . Otherwise,  $r$  can be obtained by

$$r = \frac{U_1(\mathbf{f}^{(k)}) - U_1(\mathbf{f}_s^{(k)})}{U_1(\mathbf{f}^{(k)}) - U_1(\mathbf{f}^{(k)} + \mathbf{A}^{(k)}\mathbf{s}^{(k)})} \quad (3.7)$$

where  $U_1$  is the objective function for the first stage defined as

$$U_1(\mathbf{f}) = \|\mathbf{f}_d^1 - \mathbf{f}\| \quad (3.8)$$

The stopping criterion for the first stage is shown as follows

$$\left|f_1^{(k)} - f_L\right| + \left|f_{M_f}^{(k)} - f_H\right| \leq \varepsilon_1 \quad (3.9)$$

where  $\varepsilon_1$  is a user-defined threshold for the first stage. By doing this, all the feature frequency points will move to the passband and  $\mathbf{f}$  will have approximately equal distance between each two adjacent feature frequency points. Therefore, the first stage helps to avoid being trapped in a local minimum.

### 3.3.4 Stage 2: Mapping from $t$ Space to $f$ Space and Adjustment of Ripple Height

In the second stage, we perform the second kind of mapping, i.e., from the feature frequency parameters to the ripple height parameters as follows

$$\mathbf{B}^{(k)}(\mathbf{f}_d^{(k)} - \mathbf{f}^{(k)}) + \mathbf{c}^{(k)} = \mathbf{t}_a - \mathbf{t}^{(k)} \quad (3.10)$$

where  $\mathbf{t}_a$  represents the desired ripple height parameters; and  $\mathbf{B}^{(k)}$  and  $\mathbf{c}^{(k)}$  are mapping coefficients between the  $\mathbf{f}$  and  $\mathbf{t}$  spaces. To find the optimal filter solution, the ripples should be equal, which means every element of  $\mathbf{t}^{(k+1)}$  should be the same. Let  $\bar{t}$  be defined as the average value of  $\mathbf{t}^{(k)}$ , i.e.,

$$\bar{t} = \frac{1}{M_f - 1} \sum_{i=1}^{M_f-1} t_i \quad (3.11)$$

To make every element of  $\mathbf{t}_a$  equal, we set

$$\mathbf{t}_a = [\bar{t} \ \bar{t} \ \dots \ \bar{t}]^T. \quad (3.12)$$

By doing this, we translate optimization problem (3.1) into the  $\mathbf{t}$  space with a new criterion, which is that the variance  $Var(\mathbf{t}^{(k)})$  of  $\mathbf{t}^{(k)}$  is minimized to be smaller than a user-defined threshold  $\varepsilon_2$

$$Var(\mathbf{t}^{(k)}) = \frac{1}{M_f - 1} \sum_{t=1}^{M_f-1} (t_i - \bar{t})^2 \leq \varepsilon_2 \quad (3.13)$$

When this new criterion is reached, the optimization process stops. If not, we determine the mapping between the  $\mathbf{t}$  space and the  $\mathbf{f}$  space. To do this, we firstly

determine the matrix  $\mathbf{B}^{(k)}$  and vector  $\mathbf{c}^{(k)}$  by the following training process

$$(\mathbf{B}^{(k)}, \mathbf{c}^{(k)}) = \arg \min_{(\mathbf{B}, \mathbf{c})} \varepsilon_t(\mathbf{B}, \mathbf{c}) \quad (3.14)$$

where the training error function is defined as

$$\varepsilon_t(\mathbf{B}, \mathbf{c}) = \sum_{i=2}^{2N+1} \|\mathbf{B}(\mathbf{f}_i - \mathbf{f}_1) + \mathbf{c} - (\mathbf{t}_i - \mathbf{t}_1)\| \quad (3.15)$$

After  $\mathbf{B}^{(k)}$  and  $\mathbf{c}^{(k)}$  are found from (3.14) and (3.15), the mapping between the  $\mathbf{t}$  and  $\mathbf{f}$  spaces is determined. We will use the mapped  $\mathbf{f}$  to deduce the desired value of the  $\mathbf{f}$  vector, for the feature parameters in the  $\mathbf{f}$  space. We introduce a new vector  $\Delta\mathbf{f} = [\Delta f_1 \quad \Delta f_2 \quad \cdots \quad \Delta f_{M_f}]^T$ , which satisfies

$$\mathbf{f}_d^{(k)} = \mathbf{f}^{(k)} + \Delta\mathbf{f} \quad (3.16)$$

where  $\mathbf{f}_d^{(k)}$  represents the desired frequency vector and  $\mathbf{f}^{(k)}$  represents the current actual frequency parameters. In order to find  $\mathbf{f}_d^{(k)}$  in the second stage, we need to solve for  $\Delta\mathbf{f}$ . After the first stage, we assume that (3.6) is satisfied. Therefore, any changes on  $\Delta f_1$  and  $\Delta f_{M_f}$  will influence the passband. We need to refine the values of  $\Delta f_1$  and  $\Delta f_{M_f}$  according to  $R(\mathbf{x}^{(k)}, \omega)$ . Firstly we find two frequency points  $f_l$  and  $f_h$  in the EM responses which can satisfy

$$f_l = \min\{\omega_j | R(\mathbf{x}^{(k)}, \omega_j) \leq \bar{t} < R(\mathbf{x}^{(k)}, \omega_{j-1})\} \quad (3.17a)$$

$$f_h = \max\{\omega_j | R(\mathbf{x}^{(k)}, \omega_j) \leq \bar{t} < R(\mathbf{x}^{(k)}, \omega_{j+1})\} \quad (3.17b)$$

with  $j = 1, 2, \dots, N_f$ , and  $N_f$  is the number of frequency points per EM simulation.

These two frequencies,  $f_l$  and  $f_h$ , approximately representing the lower and upper frequencies of the current frequency band are compared with the desired lower and upper frequencies  $f_L$  and  $f_H$ . Then  $\Delta f_1$  and  $\Delta f_{M_f}$  are obtained by solving

$$\Delta f_1 = f_L - f_l \quad (3.18a)$$

$$\Delta f_{M_f} = f_H - f_h \quad (3.18b)$$

The equations (3.17) and (3.18) try to refine the passband location. Then we regard  $\Delta f_1$  and  $\Delta f_{M_f}$  as constant value and obtain  $\Delta \mathbf{f}$  by solving

$$\Delta \mathbf{f} = \arg \min_{\Delta f_2, \dots, \Delta f_{M_f-1}} \left\| \mathbf{t}_a - \mathbf{t}^{(k)} - \mathbf{B}^{(k)} [\Delta f_1 \quad \Delta f_2 \quad \dots \quad \Delta f_{M_f-1} \quad \Delta f_{M_f}]^T - \mathbf{c}^{(k)} \right\| \quad (3.19)$$

Using (3.16),  $\mathbf{f}_d^{(k)}$  can be obtained from  $\Delta \mathbf{f}$ .

Next, we perform space mapping from the  $\mathbf{f}$  space to the  $\mathbf{x}$  space by solving (3.5). Once  $\mathbf{s}^{(k)}$  is determined, we perform  $2N + 1$  EM simulations using the star distribution in parallel around the new center  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ , and subsequently we can get the feature frequency parameters  $\mathbf{f}_s^{(k)}$  and the ripple height parameters  $\mathbf{t}_s^{(k)}$  at  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ .

We calculate the adjustment index  $r$  for the trust radius. If the dimensions of feature space parameters change,  $r$  is set to be  $-1$ . Otherwise,  $r$  can be obtained by

$$r = \frac{U_2(\mathbf{t}^{(k)}) - U_2(\mathbf{t}_s^{(k)})}{U_2(\mathbf{t}^{(k)}) - U_2(\mathbf{t}^{(k)} + \mathbf{B}^{(k)} \mathbf{A}^{(k)} \mathbf{s}^{(k)} + \mathbf{c})} \quad (3.20)$$

where  $U_2$  is defined as

$$U_2(\mathbf{t}) = Var(\mathbf{t}) \quad (3.21)$$

Notice that the computation for trust region parameters in Stage 2 (described by (3.20) and (3.21)) are different from that in Stage 1 (described by (3.7) and (3.8)).

### 3.3.5 Update of Trust Region

After we get  $\mathbf{s}^{(k)}$  for both stages we should update the radius of the trust region using the following equation [118][119]

$$\delta^{(k+1)} = \begin{cases} 0.618 \|\mathbf{s}^{(k)}\|, & ifr < 0.1, \\ \min \{1.214\delta^{(k)}, \Delta^*\}, & ifr > 0.8, \\ \|\mathbf{s}^{(k)}\|, & otherwise, \end{cases} \quad (3.22)$$

where  $\Delta^*$  is the maximum value of the trust radius, and decide whether or not to accept this step  $\mathbf{s}^{(k)}$ .

If the dimension of  $\mathbf{f}_s^{(k)}$  remains the same as  $\mathbf{f}^{(k)}$ , and the following condition

$$U_1(\mathbf{f}_s^{(k)}) \leq U_1(\mathbf{f}^{(k)}) \quad (3.23)$$

is satisfied in the first stage, or the condition

$$Var(\mathbf{t}_s^{(k)}) \leq Var(\mathbf{t}^{(k)}) \quad (3.24)$$

is satisfied in the second stage, we accept the step  $\mathbf{s}^{(k)}$  and update the design

variables as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)} \quad (3.25)$$

At the same time, we set  $\mathbf{f}^{(k+1)} = \mathbf{f}_s^{(k)}$ , and  $\mathbf{t}^{(k+1)} = \mathbf{t}_s^{(k)}$ .

In this way, both the original design variables and the trust radius are updated, thereby completing one iteration of the proposed space mapping.

Otherwise (i.e., if the dimensions of feature space parameters change, or neither (3.23) nor (3.24) is satisfied),  $\mathbf{x}^{(k)}$  will be kept unchanged, and a new  $\mathbf{s}^{(k)}$  is calculated by solving (3.5) with the updated trust radius  $\delta^{(k)} = \delta^{(k+1)}$ . To make our proposed method more robust, we terminate the algorithm if one of the following conditions is satisfied:  $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < 10^{-2}$  or  $\delta^{(k+1)} < 10^{-3}$  [120].

### 3.3.6 Summary

The flowchart of the proposed space mapping technique is shown in Fig. 3.3. The proposed algorithm can be summarized as follows.

Step 1: Initialize  $\mathbf{x}^{(k)}$  and  $\delta^{(k)}$  at  $k = 0$ .

Step 2: Set  $\mathbf{x}_1$  to be equal to  $\mathbf{x}^{(k)}$ . Create multiple points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{2N+1}$  using a star distribution sampling strategy with the center  $\mathbf{x}_1$ . Evaluate multi-point fine responses  $R(\mathbf{x}_i, \omega)$  by performing EM simulation using parallel computation for  $i = 1, 2, \dots, 2N + 1$ .

Step 3: Determine the feature frequency parameters  $\mathbf{f}^{(k)}$  and the ripple height parameters  $\mathbf{t}^{(k)}$ .

Step 4: If the stopping criterion of the first stage (3.9) is not satisfied, go to Step 5, else go to Step 9.

Step 5: Obtain the first mapping matrix  $\mathbf{A}^{(k)}$  by solving (3.3)-(3.4), and get  $\mathbf{f}_d^1$  by (3.6).

Step 6: Obtain the prospective step  $\mathbf{s}^{(k)}$  by solving (3.5).

Step 7: Perform parallel EM simulations at  $2N + 1$  star distribution points around the new center  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ . Obtain the feature parameters  $\mathbf{f}_s^{(k)}$  and  $\mathbf{t}_s^{(k)}$  at  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ , and find  $r$  using (3.7)-(3.8). Update trust radius  $\delta^{(k+1)}$  by solving (3.22) according to the value of  $r$ .

Step 8: If the dimension of  $\mathbf{f}_s^{(k)}$  remains the same as  $\mathbf{f}^{(k)}$ , and  $U_1(\mathbf{f}_s^{(k)}) \leq U_1(\mathbf{f}^{(k)})$ , then we update the design variable  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ , set  $\mathbf{f}^{(k+1)} = \mathbf{f}_s^{(k)}$ , and set  $\mathbf{t}^{(k+1)} = \mathbf{t}_s^{(k)}$ . Update iteration counter  $k = k + 1$ , and go to Step 4. Otherwise, keep  $\mathbf{x}^{(k)}$  unchanged, set  $\delta^{(k)} = \delta^{(k+1)}$ , and go to Step 6.

Step 9: If the stopping criterion of the second stage (3.13) is satisfied, STOP, otherwise, go to Step 10.

Step 10: Obtain  $\mathbf{A}^{(k)}$  by solving (3.3)-(3.4) and get the second set of mapping matrices  $\mathbf{B}^{(k)}$ ,  $\mathbf{c}^{(k)}$  by solving (3.14)-(3.15), then find  $\mathbf{f}_d^{(k)}$  by solving (3.10)-(3.12) and (3.16)-(3.19).

Step 11: Obtain the prospective step  $\mathbf{s}^{(k)}$  by solving (3.5).

Step 12: Perform parallel EM simulations at  $2N + 1$  star distribution points around the new center  $(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ . Get the feature parameters  $\mathbf{f}_s^{(k)}$  and  $\mathbf{t}_s^{(k)}$  at

$(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ , and find  $r$  using (3.20)-(3.21). Update trust radius  $\delta^{(k+1)}$  by solving (3.22) according to the value of  $r$ .

Step 13: If the dimension of  $\mathbf{f}_s^{(k)}$  remains the same as  $\mathbf{f}^{(k)}$ , and  $Var(\mathbf{t}_s^{(k)}) \leq Var(\mathbf{t}^{(k)})$ , then we update the design variables  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ , set  $\mathbf{f}^{(k+1)} = \mathbf{f}_s^{(k)}$ , and set  $\mathbf{t}^{(k+1)} = \mathbf{t}_s^{(k)}$ . Update iteration counter  $k = k + 1$ , and go to Step 9. Otherwise, keep  $\mathbf{x}^{(k)}$  unchanged, set  $\delta^{(k)} = \delta^{(k+1)}$ , and go to Step 11.

### 3.3.7 Discussion

There are two kinds of challenges for EM optimization of filter design, the first being the challenge of a computationally bad starting point, but one with the correct number of feature frequencies, and the second the ability to correct the number of feature frequencies if the initial point has the wrong number of feature frequencies. In this chapter, we focus on the solution to the first challenge. The second challenge, which is equally important and a heavy task, is a possible direction of future research. In our present work, we have used an empirical approach to preprocess the starting point using derivative information of the response shape to guide the correction of the number of the feature frequencies iteratively. Once the correct number of feature frequencies is reached, the proposed trust region method in the two different optimization stages will maintain the number of feature frequencies throughout the proposed optimization process.

Our proposed cognition driven formulation of space mapping translates the direct optimization problem into a relatively easier optimization problem in feature space.

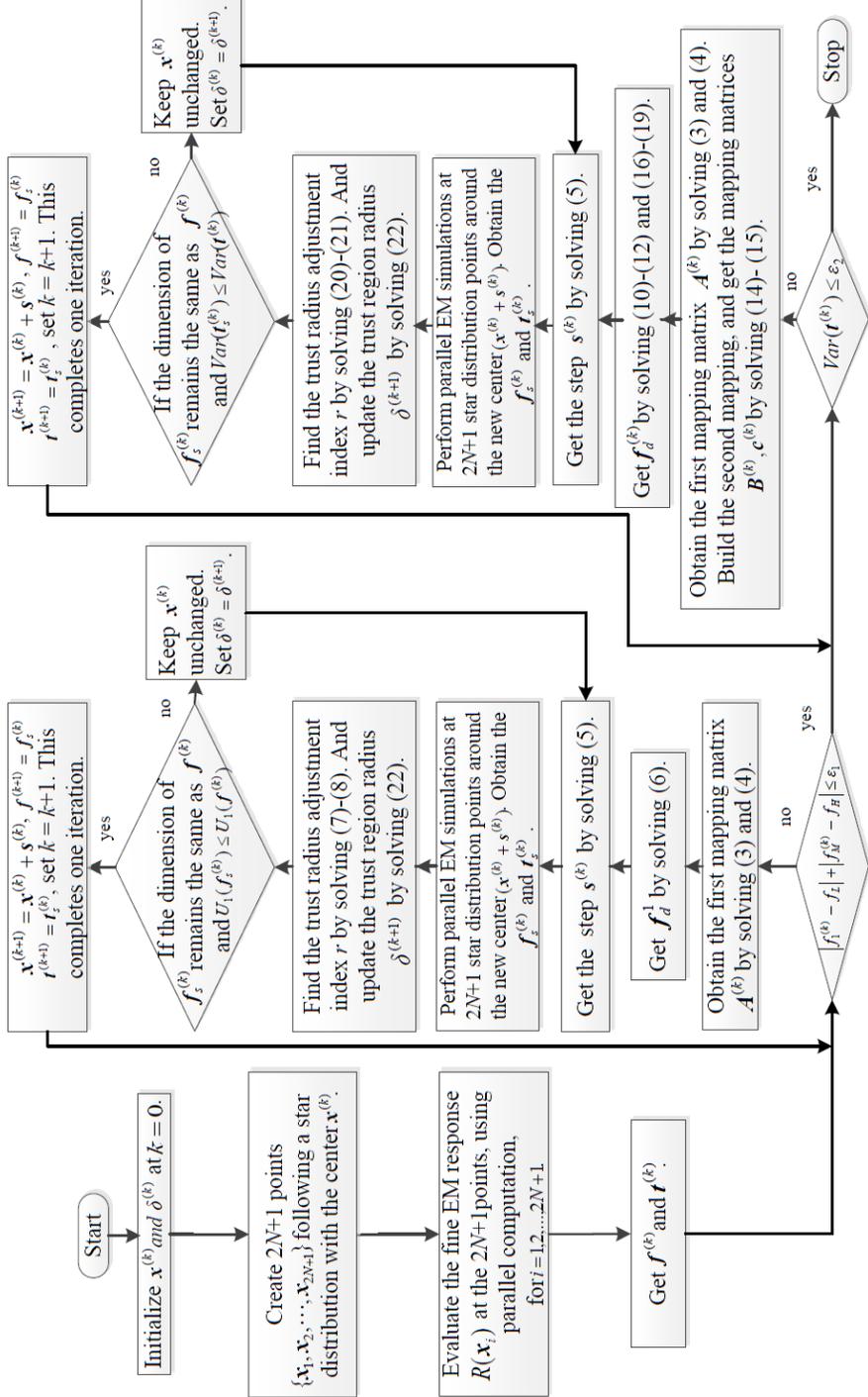


Figure 3.3: The flowchart of the proposed cognition-driven space mapping technique.

The mapping from feature space to design variable space is more linear and valid in a larger range, compared to the original problem in the direct optimization. Therefore, our method can improve the efficiency and ability to avoid being trapped into local minima. Using the trust region method, our method is guaranteed to converge. The final optimization result is still a local solution, which is much better than that obtained in the direct optimization method.

In this subsection, we also discuss the limitation of our proposed method. Firstly, our technique is well-suited to the EM based design of Chebyshev- and elliptic-type filters, which are characterized by equal-ripple responses. For other types of filters and others applications of microwave components, our proposed method requires features to be identified, which may or may not be an easy task. This could be a possible direction of future research. Secondly, our proposed technique in this chapter can proceed with the starting point with the correct number of feature parameters. Another possible direction is to develop a method which can deal with the starting point with incorrect number of feature parameters.

## **3.4 Application Examples**

### **3.4.1 Optimization of A Four-Pole Waveguide Filter**

The first example under consideration is a four-pole wave-guide filter [107]. The tuning elements are penetrating posts of square cross section placed at the center of each cavity and each coupling window, shown in Fig. 3.2 (a). The input and output waveguides, as well as the resonant cavities, are standard WR-75 waveguides

( $a = 19.050\text{mm}$ ,  $b = 9.525\text{mm}$ ). The thickness of all the coupling windows is set to  $2\text{mm}$ .  $h_1$ ,  $h_2$  and  $h_3$  are the heights of posts in the coupling windows, and  $h_{c1}$ ,  $h_{c2}$  are the height of the posts in the resonant cavities. The design variables are  $\mathbf{x} = [h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2}]^T$ .

EM evaluation is performed by *HFSS* [121] EM simulator using fast simulation feature. With this simulation feature, we can get the response of the entire frequency band without a discrete frequency sweep. The desired filter responses have been chosen to be standard four-pole Chebyshev curves of 300-MHz bandwidth, 0.02-dB ripple, and centered at 11 GHz. The starting point is  $\mathbf{x}^{(0)} = [3.3 \ 4.389 \ 3.991 \ 3.28 \ 2.914]^T$  (all values in mm). From the second iteration, the  $\mathbf{t}$  parameters are found to be  $\mathbf{t}^{(2)} = [-15.473 \ -29.709 \ -32.851]^T$  (all values in dB).

Using the proposed technique, the optimal solution  $\mathbf{x}^{(7)} = [3.562 \ 4.291 \ 3.790 \ 3.237 \ 2.955]^T$  (all values in mm) is obtained after 7 iterations, and the final ripple vector  $\mathbf{t}^{(7)} = [-27.088 \ -27.512 \ -27.783]^T$  (all values in dB). The responses from the initial point, the first and the last iterations are shown in Fig. 3.4(a), 3.4(b) and 3.4(c), respectively. Fig. 3.4(b) shows that all the feature frequencies move to the passband after the first iteration. Fig. 3.4(c) shows that our proposed method can avoid being trapped in a local minimum and that a good equal-ripple filter response is obtained after 7 iterations. The values of objective function for all the iterations are shown in Fig. 3.5. From the figure, we observe that using our proposed space mapping method the filter response can satisfy the design specifications in 2

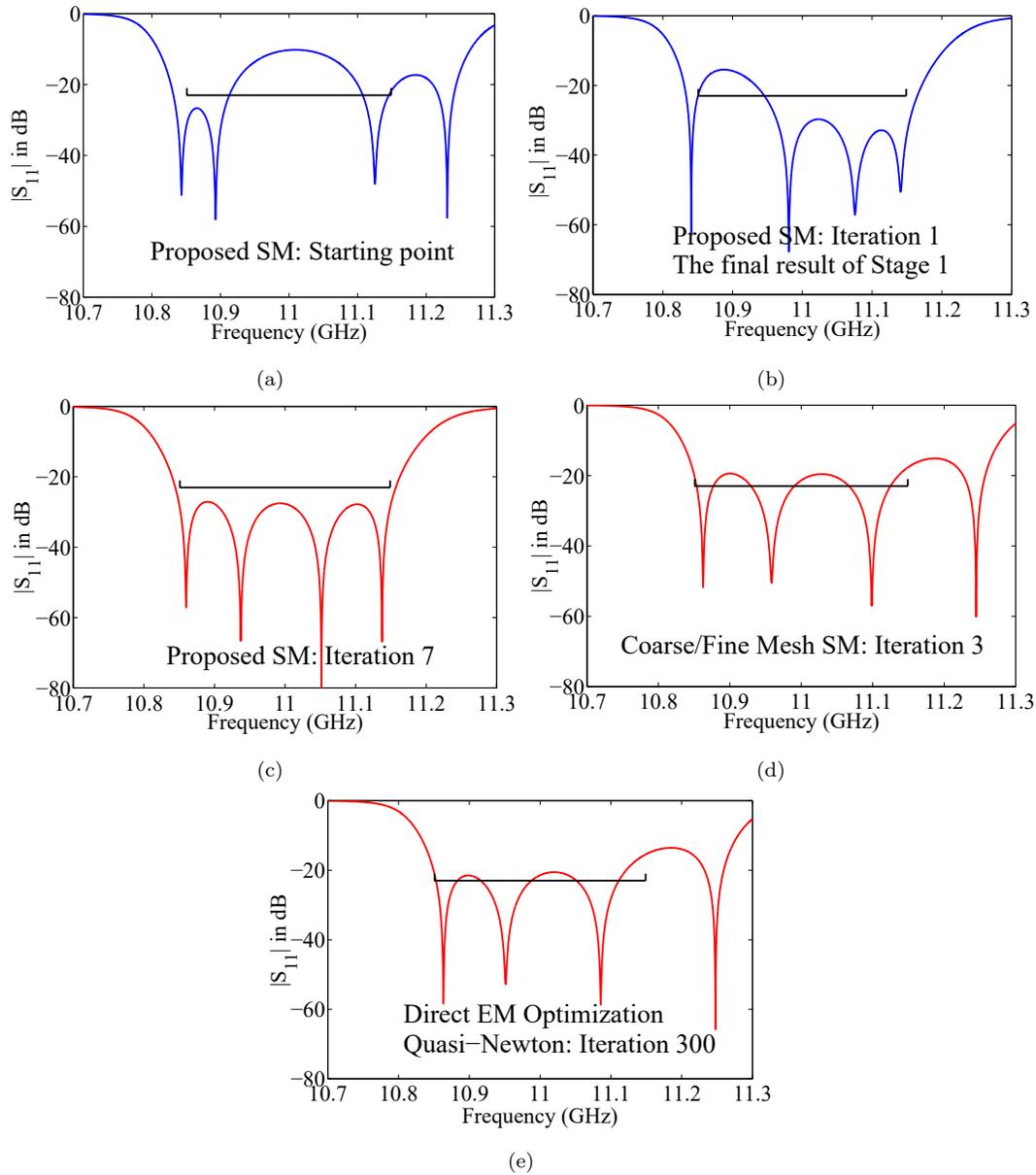


Figure 3.4: A comparison of the results for 3 different optimization methods: (a) the starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) a good equal-ripple response is obtained after 7 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum.

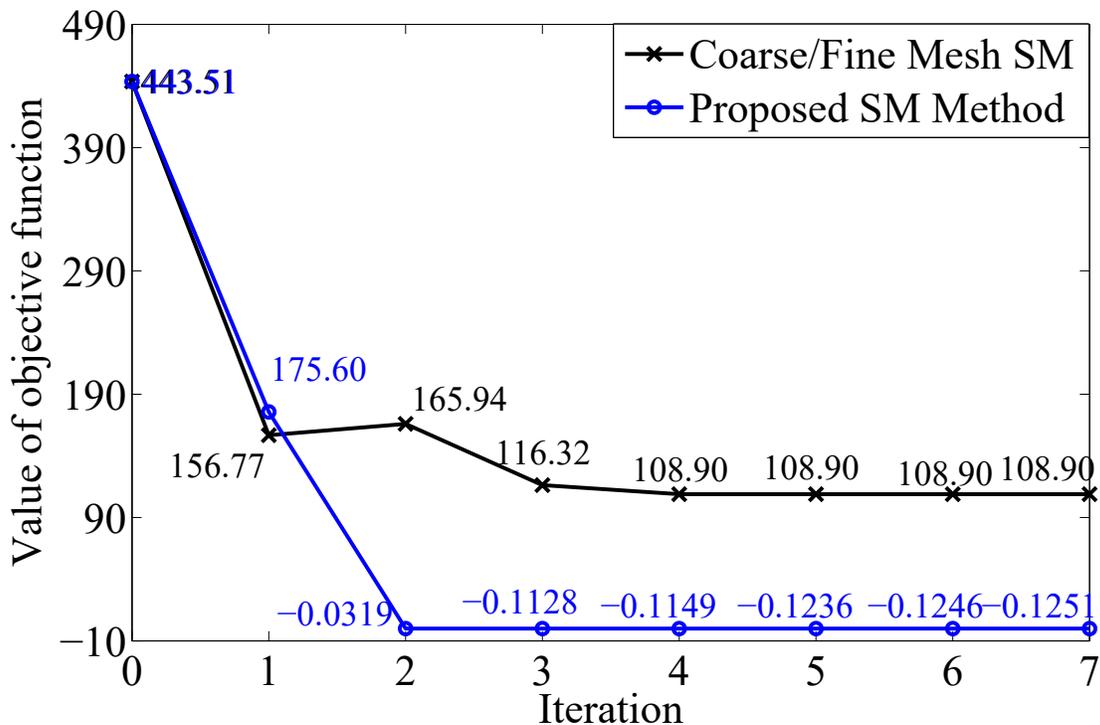


Figure 3.5: The objective function values of the four-pole waveguide filter response using: the proposed SM method (o) and coarse and fine mesh space mapping method ( $\times$ ). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 7 iterations.

iterations and it can even exceed the specifications in subsequent iterations.  $U_1$  in the first stage and the variances of  $\mathbf{t}$  in the second stage of our proposed method are shown in Fig. 3.6. We set the threshold  $\varepsilon_2 = 0.2$ . Smaller (or larger) values for  $\varepsilon_2$  tend to make the ripple heights more (or less) uniform using more (or less) overall computation time. In this example, both  $U_1$  and the variance of  $\mathbf{t}$  converge fast.

For comparison purposes, we use the baseline coarse and fine mesh space map-

ping optimization method [39] to optimize this filter. The coarse mesh EM optimizations for the surrogate training and for surrogate optimization are both carried out using *HFSS*'s internal quasi-Newton optimizer. The value of objective function for each iteration is shown in Fig. 3.5, from which we found that the coarse and fine mesh space mapping optimization process falls into a local minimum. The space mapping iterations stopped at this point because the updates by surrogate optimization cannot lead to any improvement in the fine model responses. The comparison of the results for different methods, including direct EM optimization (using *HFSS Optimetrics* quasi-Newton optimization), coarse and fine mesh space mapping method and our proposed method is shown in Fig. 3.4 and Table 3.1. In the coarse and fine mesh space mapping method, the training time and the design optimization time are very long because of coarse mesh EM simulation. In our method, the training process, which means solving mapping matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{c}$ , and the design optimization process, which is to solve Equ. (3.5), only do algebraic calculations, so our feature space mapping method saves much time. The fine EM evaluation time (6 min) of our proposed method is the time for generating 11 (i.e.,  $2N + 1$ ) star distribution points in parallel. Therefore the total computation time (6 min) for the 11 EM simulations is only incrementally more than that of a single EM simulation (4 min). As observed in Table 3.1, our method can increase the optimization efficiency and find a better result within less time compared to coarse and fine mesh EM space mapping and direct EM optimization.

As a further experiment about robustness of the proposed optimization for this filter example, we use a much worse starting point  $\mathbf{x}^{(0)} = [3.020 \ 4.680 \ 4.314$

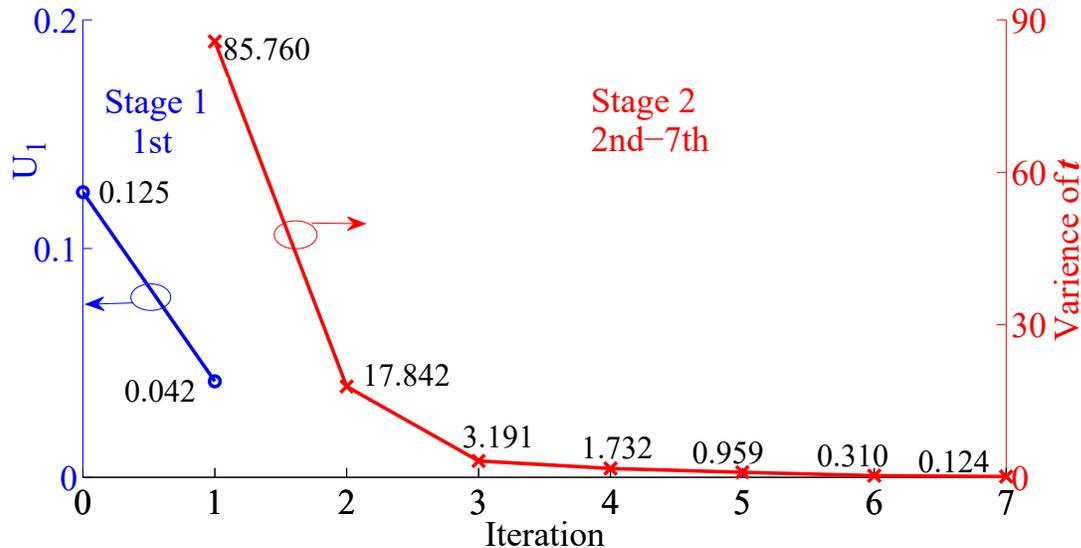


Figure 3.6: The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 7th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast.

$3.480 \ 3.028]^T$  (all values in mm) to rerun our proposed method. Using the proposed technique, the optimal solution  $\mathbf{x}^{(8)} = [3.424 \ 4.116 \ 3.608 \ 3.291 \ 2.977]^T$  (all values in mm) is obtained after 8 iterations, and the final ripple vector  $\mathbf{t}^{(8)} = [-24.316 \ -24.062 \ -24.793]^T$  (all values in dB). The responses from the initial point, the last iteration of the first stage and the last iteration of proposed method are shown in Fig. 3.7(a), 3.7(b) and 3.7(c), respectively. A preprocessing step was used to create a usable starting point, which is shown in Fig. 3.7(a), with the correct number of feature frequencies, i.e., 4. During the optimization process, if EM simulations occasionally produce the wrong number of (e.g., 3) feature frequencies, our trust region mechanism will shrink the step size until the number of feature frequencies returns to 4, at which point the optimization update then takes place.

Table 3.1: Comparison of Three Methods for The Waveguide Filter

| Optimization Method               | Direct EM optimization                     | Coarse and fine mesh SM optimization         | Proposed Space Mapping |
|-----------------------------------|--|--|------------------------|
| No. of Iterations                 | 300  | 4  | 7                      |
| Fine model Evaluation Time        | 300×4 min                                  | 5×4 min                                      | 8×6 min                |
| Training Time                     | -  | 3×1h   | 7×1 min                |
| Design Optimization Time          | -  | 4×1h   | 7×1 min                |
| Total time                        | 20 h                                       | 7h 20min                                     | 1h 2min                |
| Final Value of Objective Function | 69.14*<br>(being trapped in local minimum) | 108. 90*<br>(being trapped in local minimum) | -0.1251                |

\* - Design specifications are not satisfied.

Fig. 3.7(b) shows that all the feature frequencies move to the passband after the first stage. Fig. 3.7(c) shows that our proposed method can avoid being trapped in local minima and that a good equal-ripple filter response is obtained after 8 iterations. The values of objective function for all the iterations are shown in Fig. 3.8. From the figure, we observe that using our proposed space mapping method the filter response can satisfy the design specifications in 8 iterations.  $U_1$  in the first stage and the variances of  $t$  in the second stage of our proposed method are shown in Fig. 3.9.

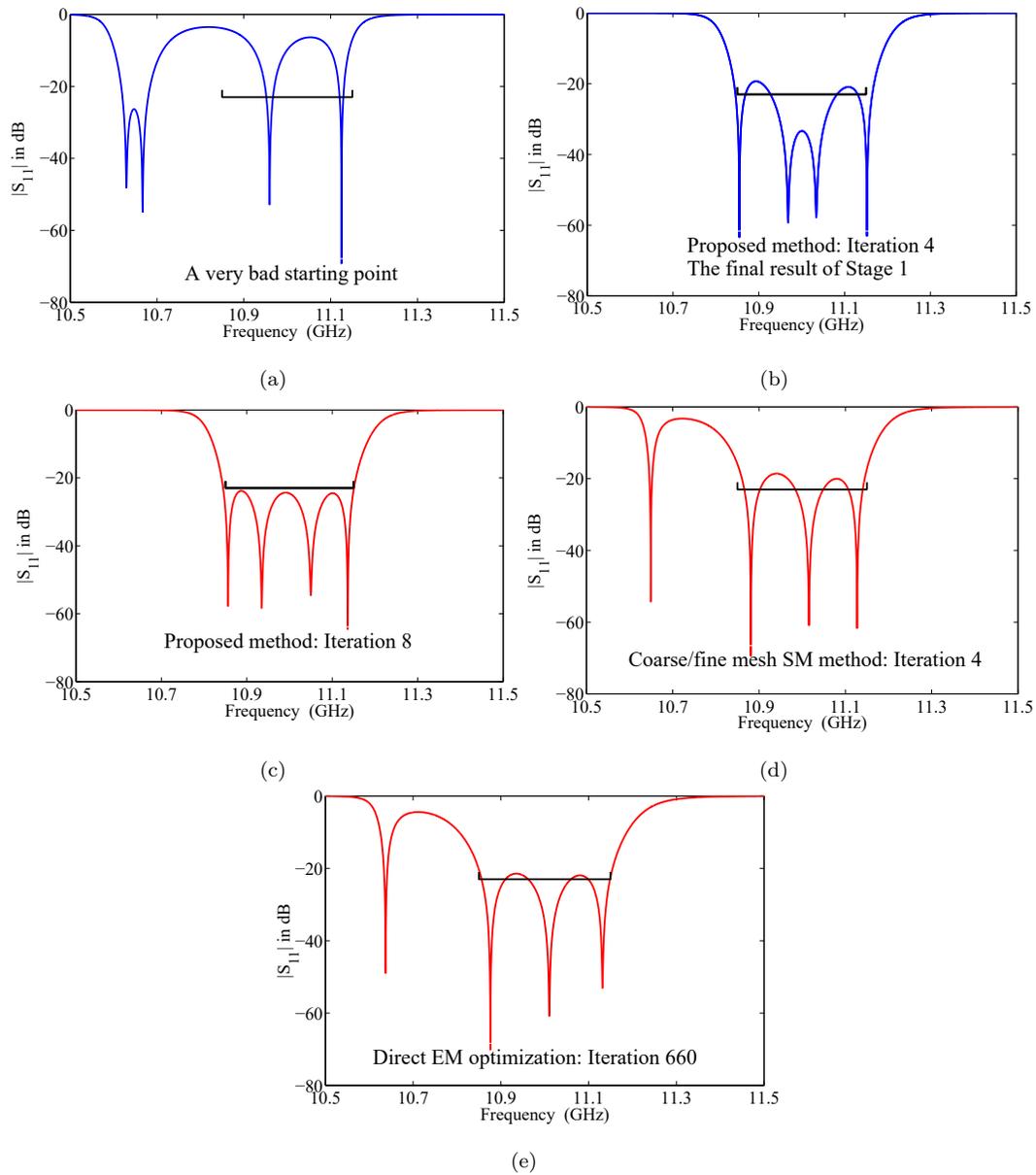


Figure 3.7: A comparison of the results for 3 different optimization methods: (a) the starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) good equal-ripple response is obtained after 8 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum.

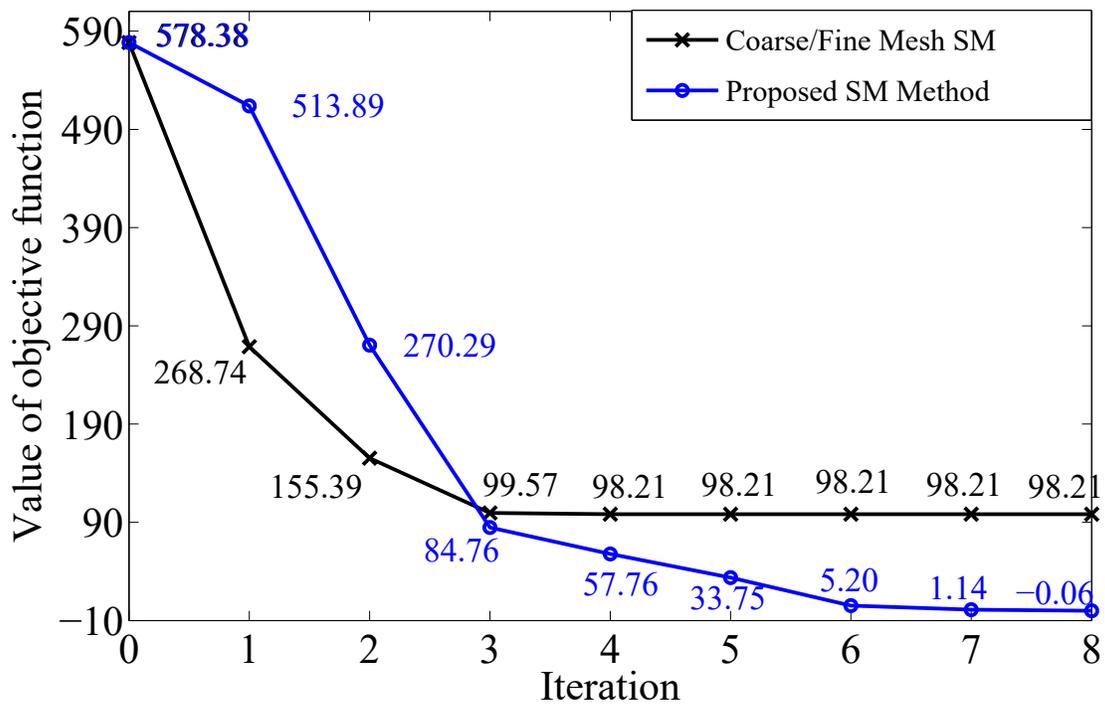


Figure 3.8: Objective function values of the four-pole waveguide filter response with bad starting point using: the proposed space mapping method (o) and coarse and fine mesh space mapping method (x). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 8 iterations.

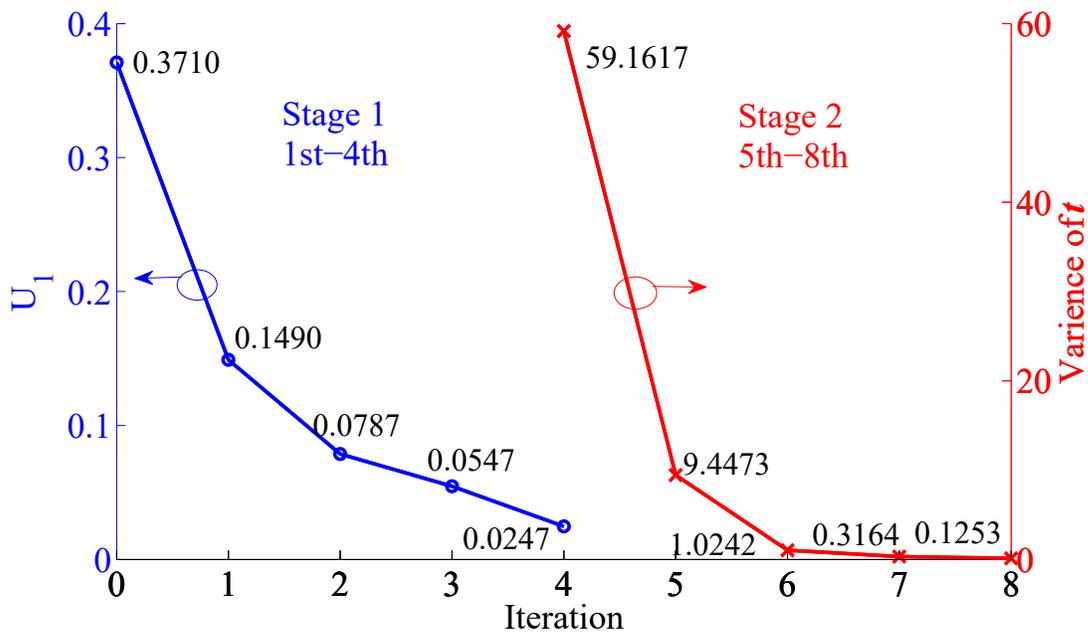


Figure 3.9: Feature space objective functions  $U_1$  and  $U_2$  (i.e.,  $Var(\mathbf{t})$ ) for the four-pole waveguide filter example with bad starting point. The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 7th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast.

Table 3.2: Comparison of Three Methods for The Waveguide Filter with Bad Starting Point

| Optimization Method               | Direct EM optimization                    | Coarse and fine mesh SM optimization      | Proposed Space Mapping |
|-----------------------------------|---|---|------------------------|
| No. of Iterations                 | 660                                       | 4   | 8                      |
| Fine model Evaluation Time        | 660×4 min                                 | 5×4 min                                   | (9* + 6)×6 min         |
| Training Time                     | -   | 3×1h                                      | 8×1 min                |
| Design Optimization Time          | -   | 4×1h                                      | 8×1 min                |
| Total time                        | 44 h                                      | 7h 20min                                  | 1h 46min               |
| Final Value of Objective Function | 25.42<br>(being trapped in local minimum) | 98.21<br>(being trapped in local minimum) | -0.06                  |

\* - The number of EM simulations, which are not accepted during trust region adjustment.

For comparison purposes, we also use the baseline coarse and fine mesh space mapping optimization method [39] to optimize this filter. The value of objective function for each iteration is shown in Fig. 8, from which we found that the coarse and fine mesh space mapping optimization process falls into a local minimum with filter responses still violating design specifications. We also performed direct EM optimization of this filter example for a further comparison. With the same starting point and same design specifications, the direct EM optimization also falls into a

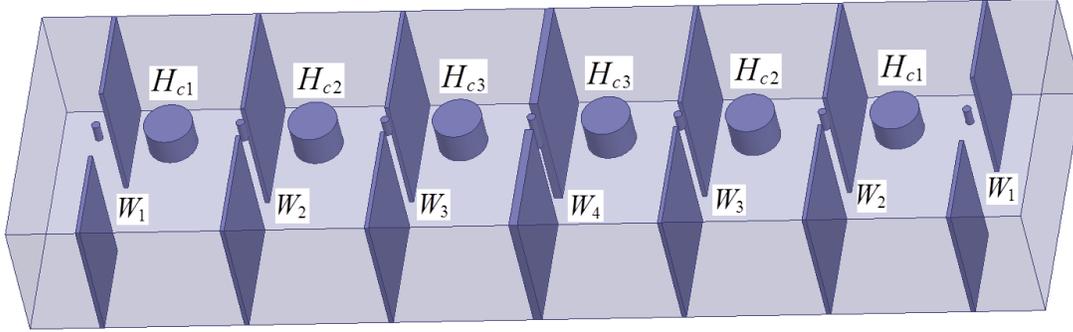


Figure 3.10: The structure of the waveguide cavity filter.

local minimum with filter responses violating design specifications. The comparison of the results for different methods, including direct EM optimization, coarse and fine mesh space mapping method and our proposed method is shown in Fig. 3.7 and Table 3.2, demonstrating that the proposed method has the shortest optimization time and best quality of solutions among the different methods compared.

### 3.4.2 Optimization of An Iris Coupled Microwave Cavity Filter

Consider an iris coupled cavity microwave bandpass filter shown in Fig. 3.10 [122]. The filter has seven geometrical design parameters described as follows. The heights of the big cylinders  $H_{c1}$ ,  $H_{c2}$ , and  $H_{c3}$  positioned at the cavity centers are responsible for tuning the frequencies in the cavity. The required coupling bandwidths are accomplished via the iris widths  $W_1$ ,  $W_2$ ,  $W_3$ , and  $W_4$  for a pre-tuning. The design variables are  $\mathbf{x} = [W_1 \ W_2 \ W_3 \ W_4 \ H_{c1} \ H_{c2} \ H_{c3}]^T$ .

EM evaluation is performed by the *HFSS* EM simulator using the fast simulation feature. We provide the design specifications  $|S_{11}| \leq -20$  dB at frequency range of 703 MHz to 713 MHz, and  $|S_{21}| \leq -10$  dB at frequency range of 690 MHz to 701 MHz and 715 MHz to 720MHz. The starting point is  $\mathbf{x}^{(0)} = [115.289 \ 51.276 \ 46.897 \ 51.084 \ 42.816 \ 50.287 \ 50.402]^T$  (all values in mm). From the second iteration, the  $\mathbf{t}$  parameters are found to be  $\mathbf{t}^{(2)} = [-16.787 \ -14.932 \ -32.553 \ -22.344 \ -17.701]^T$  (all values in dB).

Using the proposed technique, the optimal solution  $\mathbf{x}^{(4)} = [115.557 \ 50.061 \ 44.226 \ 47.605 \ 42.734 \ 50.331 \ 50.461]^T$  (all values in mm) is obtained after 5 iterations, and the final ripple vector  $\mathbf{t}^{(4)} = [-20.16 \ -20.48 \ -20.32 \ -20.28 \ -20.60]^T$  (all values in dB). The responses from the initial point, the first and the last iterations are shown in Fig. 3.11, from which we observe that our method can avoid being trapped in a local minimum and achieve a good equal-ripple filter response in 4 iterations. The values of objective function are shown in Fig. 3.12, which further shows our method avoids being trapped in a local minimum. We set the threshold  $\varepsilon_2 = 0.3$ .  $U_1$  in the first stage and the variances of  $\mathbf{t}$  in the second stage of our proposed method are shown in Fig. 3.13, and both  $U_1$  and  $Var(\mathbf{t})$  converge fast. From the final filter response in Fig. 3.11(c), we find that the best equal-ripple filter solution has already been found.

For comparison purposes, we use the coarse and fine mesh space mapping method [39] to optimize this filter, and the value of objective function for each iteration is shown in Fig. 3.12, from which we observe that the coarse and fine mesh space

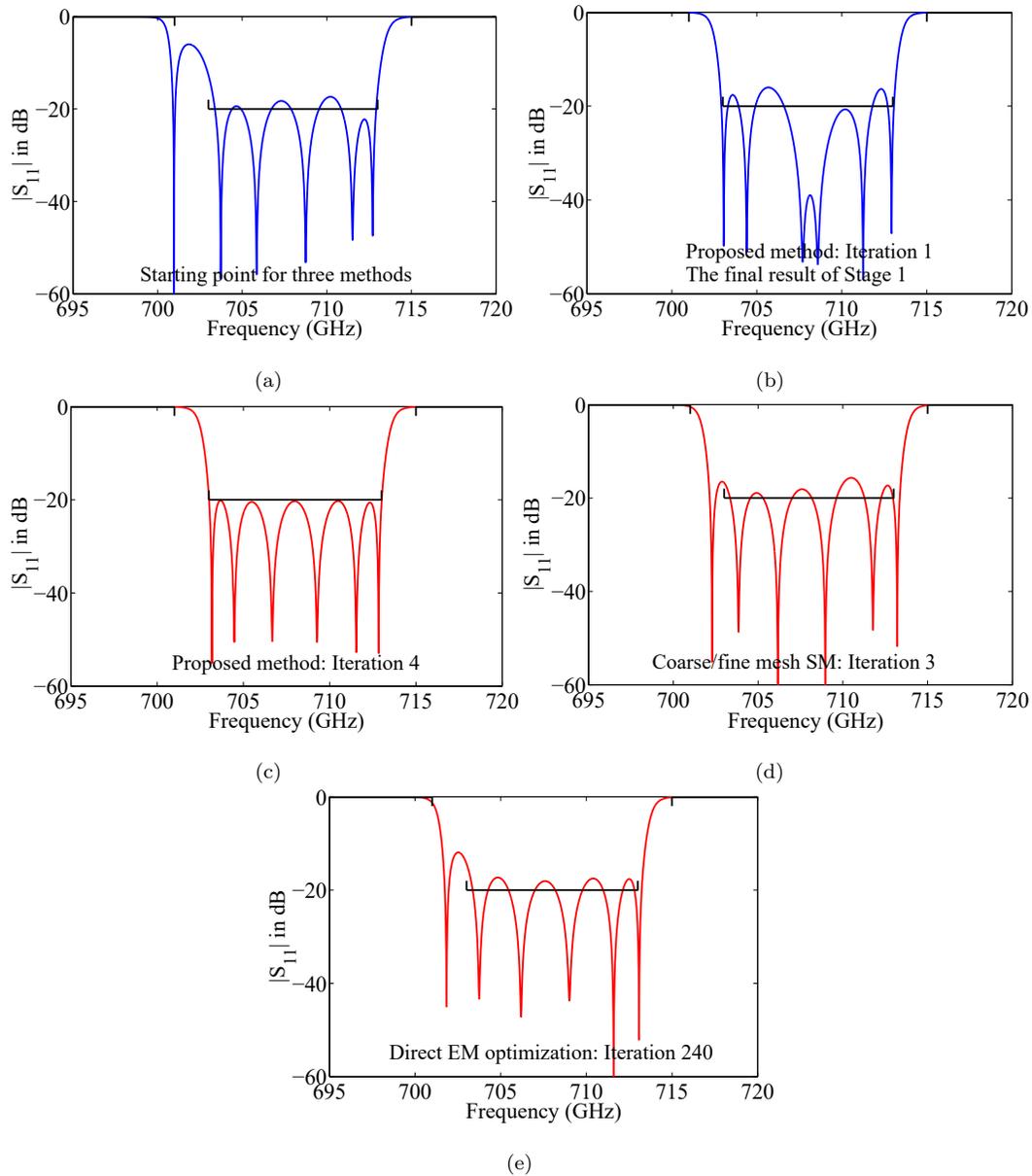


Figure 3.11: A comparison of the results for 3 different optimization methods: (a) the starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) good equal-ripple response is obtained after 4 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum.

Table 3.3: Comparison of Three Methods for The Cavity Filter

| Optimization Method               | Direct EM optimization                    | Coarse and fine mesh SM optimization      | Proposed Space Mapping |
|-----------------------------------|---|---|------------------------|
| No. of Iterations                 | 240                                       | 3   | 4                      |
| Fine model Evaluation Time        | 240×30 min                                | 4×30 min                                  | 5×40 min               |
| Training Time                     | -   | 2×5h                                      | 4×1 min                |
| Design Optimization Time          | -   | 3×4h                                      | 4×1 min                |
| Total time                        | 120 h                                     | 24h                                       | 3h 28min               |
| Final Value of Objective Function | 82.36<br>(being trapped in local minimum) | 63.24<br>(being trapped in local minimum) | 0.36                   |

mapping optimization process falls into a local minimum. The comparison of the results for different methods, including direct EM optimization, coarse and fine mesh space mapping method and our proposed method is shown in Fig. 3.11 and Table 3.3. The fine EM evaluation time (40 min) of our proposed method is the time for generating 15 (i.e.,  $2N + 1$ ) star distribution points in parallel. Therefore the total computation time (40 min) for the 15 EM simulations is only incrementally more than that of a single EM simulation (30 min). Our space mapping method can increase the optimization efficiency and find a better result within less time compared to coarse and fine mesh EM space mapping and direct EM optimization.

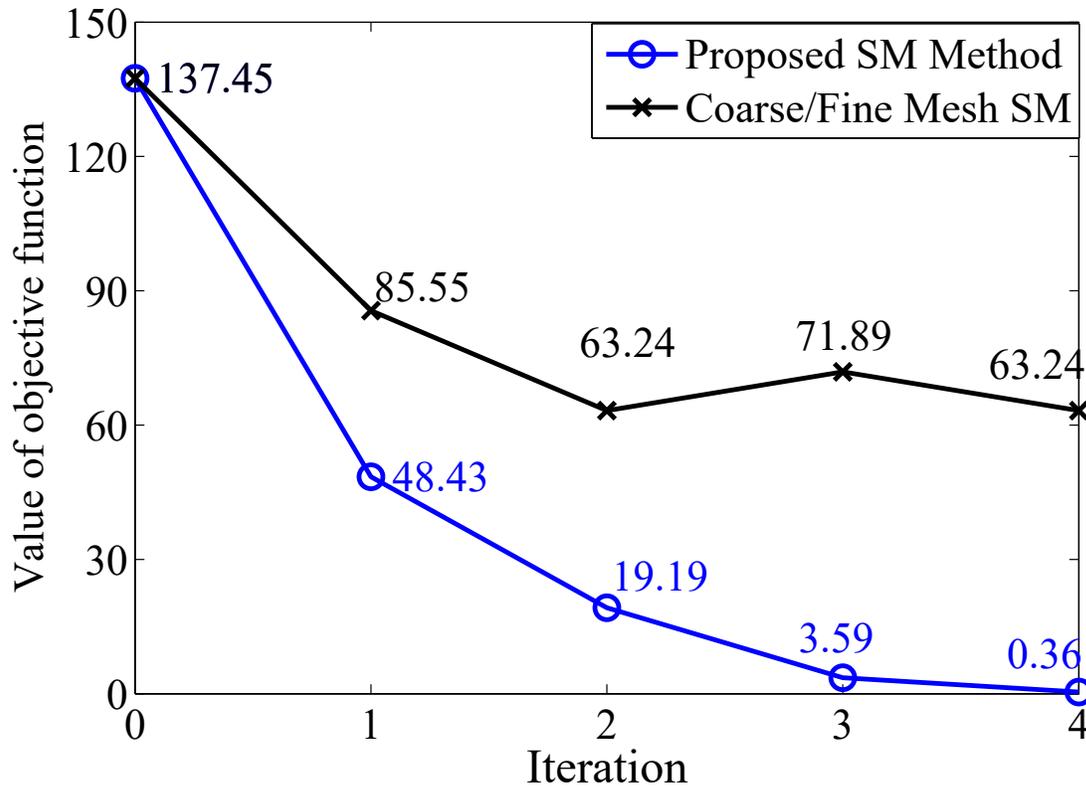


Figure 3.12: The objective function values of the cavity filter response using: the proposed space mapping method (o) and coarse and fine mesh space mapping method ( $\times$ ). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 4 iterations.

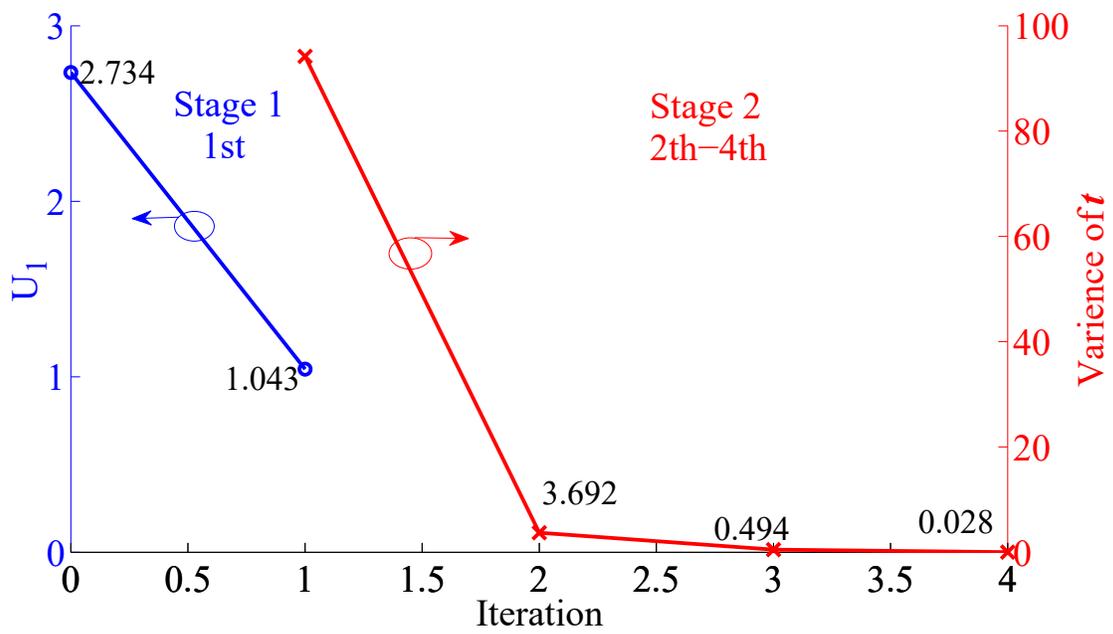


Figure 3.13: Feature space objective functions  $U_1$  and  $U_2$  (i.e.,  $Var(\mathbf{t})$ ) for the cavity filter. The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 4th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast.

As a further experiment on the robustness of the proposed optimization for this filter example, we use a much worse starting point  $\mathbf{x}^{(0)} = [111.910 \ 56.822 \ 53.916 \ 57.906 \ 43.145 \ 49.385 \ 49.626]^T$  (all values in mm) to rerun our proposed method. Using the proposed technique, the optimal solution  $\mathbf{x}^{(12)} = [119.331 \ 51.953 \ 45.099 \ 48.278 \ 41.881 \ 50.186 \ 50.385]^T$  (all values in mm) is obtained after 7 iterations, and the final ripple vector  $\mathbf{t}^{(12)} = [-24.258 \ -24.487 \ -23.979 \ -23.949 \ -25.188]^T$  (all values in dB). The responses from initial point, the last iteration of the first stage and the last iterations are shown in Fig. 3.14(a), 3.14(b) and 3.14(c), respectively. A preprocessing step was used to create a usable starting point, which is shown in Fig. 3.14(a), with correct number of feature frequencies. Fig.3.14(b) shows that all the feature frequencies move to the passband after the first stage. Fig. 3.14(c) shows that our proposed method can avoid being trapped in a local minimum and that a good equal-ripple filter response is obtained after 12 iterations. The values of objective function for all the iterations are shown in Fig. 3.15. From the figure, we observe that using our proposed space mapping method the filter response can satisfy the design specifications in 11 iterations.  $U_1$  in the first stage and the variances of  $\mathbf{t}$  in the second stage of our proposed method are shown in Fig. 3.16. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast.

For comparison purposes, we also use the baseline coarse and fine mesh space mapping optimization method [39], and the value of objective function for each iteration is shown in Fig. 3.15, from which we observe that the coarse and fine mesh space mapping optimization process falls into a local minimum. With the same starting point and same design specifications, the direct EM optimization is

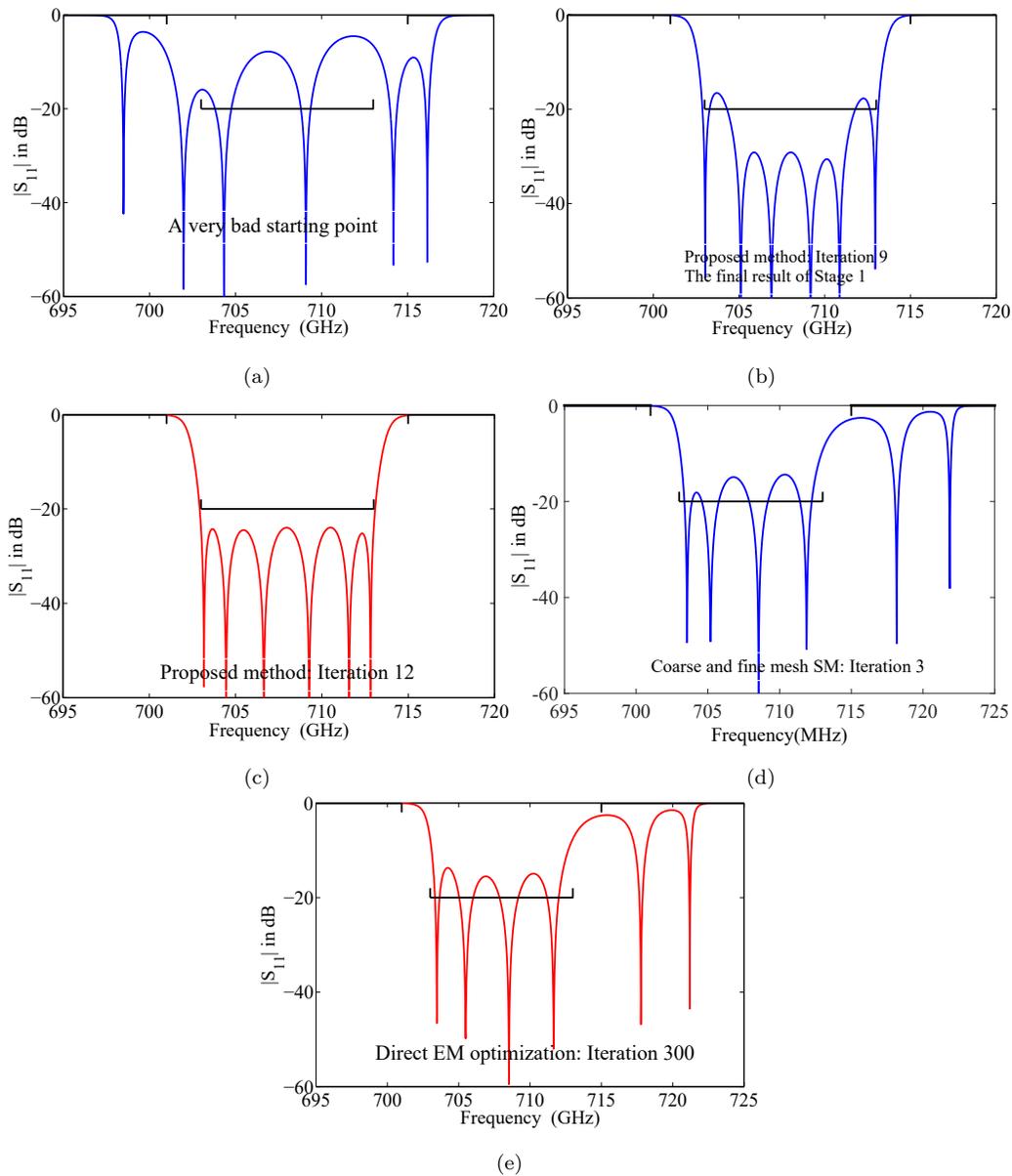


Figure 3.14: Comparison of the results for three different optimization methods for the cavity filter example with bad starting point: (a) a very bad starting point for all three methods, (b) using our proposed space mapping method, all the feature frequencies move to the passband after the first stage, (c) good equal-ripple response is obtained after 12 iterations, and our method can avoid being trapped in a local minimum; (d) using coarse and fine mesh space mapping, the optimization process falls into a local minimum; (e) using direct EM optimization, the optimization process falls into a local minimum. 72

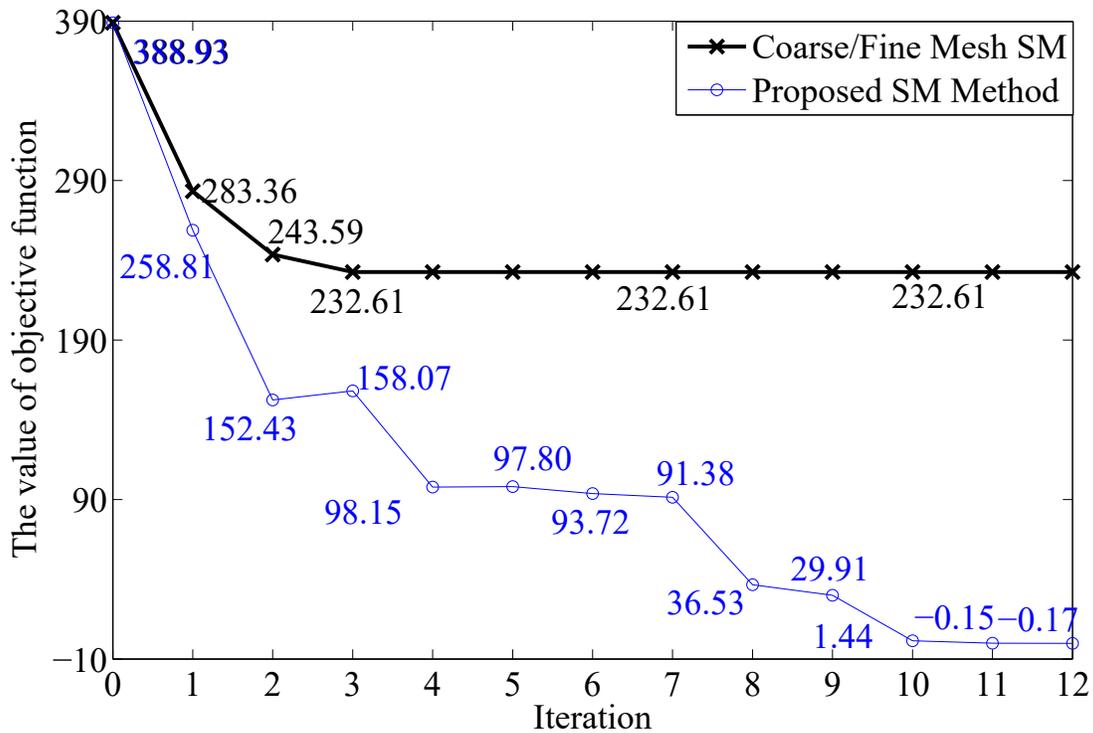


Figure 3.15: The objective function values of the cavity filter response with bad starting point using: the proposed space mapping method (o) and coarse and fine mesh space mapping method (x). The objective function in the coarse and fine mesh space mapping optimization cannot be reduced further because it falls into a local minimum. Our proposed method can avoid being trapped in a local minimum and finds a good filter response in 4 iterations.

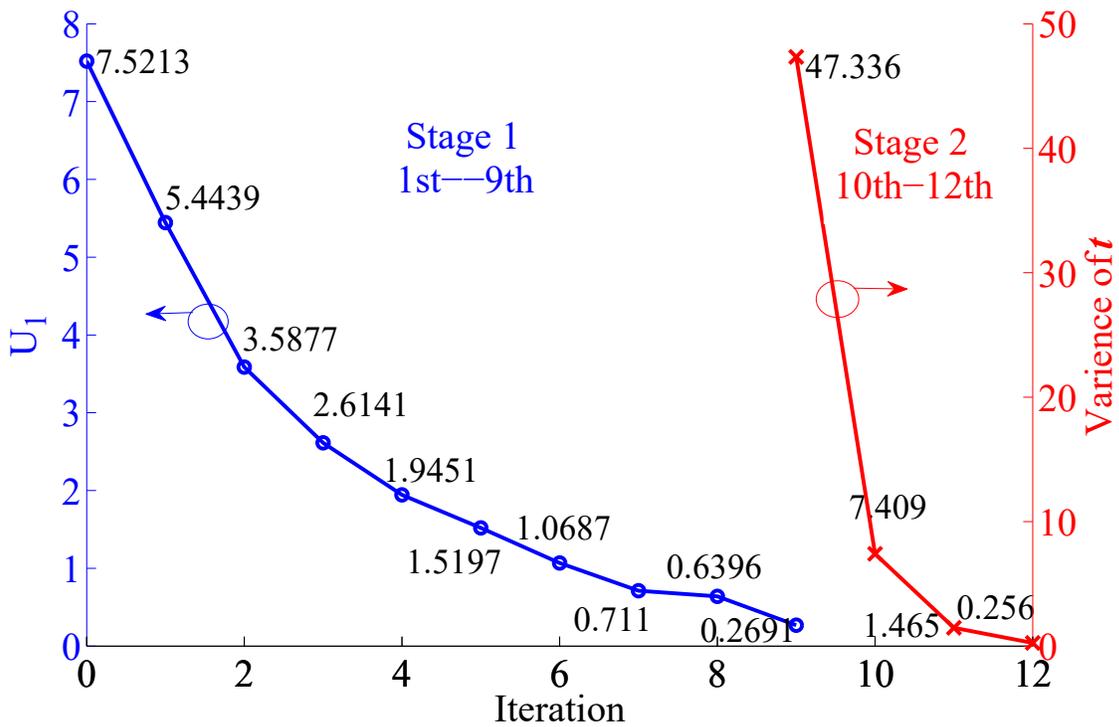


Figure 3.16: Feature space objective functions  $U_1$  and  $U_2$  (i.e.,  $Var(\mathbf{t})$ ) for the cavity filter with bad starting point. The first stage goes in the first iteration, and the second stage runs from the 2nd iteration to the 4th iteration. Both  $U_1$  and  $Var(\mathbf{t})$  converge fast.

Table 3.4: Comparison of Three Methods for The Cavity Filter with Bad Starting Point

| Optimization Method               | Direct EM optimization                     | Coarse and fine mesh SM optimization       | Proposed Space Mapping |
|-----------------------------------|--|--|------------------------|
| No. of Iterations                 | 300  | 3  | 12                     |
| Fine model Evaluation Time        | 300×30 min                                 | 4×30 min                                   | (13 + 3*)×40 min       |
| Training Time                     | -  | 2×5h                                       | 12×1 min               |
| Design Optimization Time          | -  | 3×4h                                       | 12×1 min               |
| Total time                        | 150 h                                      | 24h  | 11h                    |
| Final Value of Objective Function | 239.77<br>(being trapped in local minimum) | 232.61<br>(being trapped in local minimum) | -0.17                  |

\*The number of EM simulations, which are not accepted during trust region adjustment.

also performed for additional comparison. The direct EM optimization also falls into a local minimum with filter responses violating the design specifications. The comparison of the results for different methods, including direct EM optimization, coarse and fine mesh space mapping method and our proposed method is shown in Fig. 3.14 and Table 3.4. Our space mapping method can increase the optimization efficiency and find a better result within less time compared to coarse and fine mesh EM space mapping and direct EM optimization.

## 3.5 Conclusions and Summary

In this chapter, a cognition-driven formulation of space mapping optimization of microwave filters has been proposed. In our method, two sets of intermediate feature space parameters, including the feature frequency and ripple height parameters, are used to build two kinds of mapping. A trust region approach has been incorporated to control the optimization updates, thus ensuring convergence of the proposed cognition-drive optimization. By using the proposed cognition-driven formulation of optimization directly in the feature space, our method can increase optimization efficiency and the ability to avoid being trapped in a local minimum over our baseline approaches of coarse and fine mesh EM space mapping and direct EM optimization. This technique is well-suited to the EM based design of Chebyshev- and elliptic-type filters, which are characterized by equal-ripple responses. We believe that further exploration of the cognition-driven formulation and the use of feature parameters for design optimization is highly promising. One possible future direction is to develop a systematic method for preprocessing the optimization starting point to correct the number of feature frequencies. Another possible direction is to incorporate more specific filter design knowledge into the cognition-driven formulation to further enhance optimization. The third possible direction is to expand the cognition-driven concept with more general feature parameters to advance EM based design beyond Chebyshev- and elliptic-type filters to more generic filters and other microwave circuits.

## Chapter 4

# Multivalued Neural Network Inverse Modeling and Applications to Microwave Filters

This chapter presents a new technique for artificial neural network (ANN) inverse modeling and applications to microwave filters. In inverse modeling of a microwave component, the inputs to the model are electrical parameters such as  $S$ -parameters, and the outputs of the model are geometrical or physical parameters. Since the analytical formula of the inverse input-output relationship does not exist, ANN becomes a logical choice because it can be trained to learn from the data in inverse modeling. The main challenge of inverse modeling is the non-uniqueness problem. This problem in ANN inverse modeling is that different training samples with the same or very similar input values have quite different (contradictory) output values (multivalued solutions). In this chapter, we propose a multivalued neural network inverse modeling technique to associate a single set of electrical parameters with

multiple sets of geometrical or physical parameters. One set of geometrical or physical parameters is called one value of our proposed inverse model. Our proposed multivalued neural network is structured to accommodate multiple values for the model output. We also propose a new training error function to focus on matching each training sample using only one value of our proposed inverse model, while other values are free and can be trained to match other contradictory samples. In this way, our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. Therefore, our proposed technique can solve the non-uniqueness problem in a simpler and more automated way compared to existing ANN inverse modeling techniques. This technique is illustrated by inverse modeling and parameter extraction of four microwave filter examples.

## 4.1 Introduction

Artificial neural networks (ANNs) have been recognized as a powerful tool in microwave modeling and design [80],[124]. This technique has been applied to parametric modeling of microwave components [125], [126], optimization of microwave circuits [81], [127], [128], modeling of nonlinear microwave devices [129], [130], modeling of RF power amplifiers [131], and design of microwave filters [17], [132], etc.

ANN can learn the nonlinear relationships between the behaviors of microwave devices and the geometrical/physical parameters. ANN trained to model the original microwave device problems can be called forward model where the inputs to the

model are the geometrical or physical parameters and outputs of the model are the electrical parameters. For design purposes or the parameter extraction of microwave filters, the information is often processed in the reverse direction in order to find the geometrical or physical parameters from given electrical parameters, which is called the inverse problem. There are usually two ways to solve the inverse problem, i.e., the optimization method and inverse modeling method. The optimization method is the most conventional way [22], [93]. For a given electrical target, the forward models will be evaluated iteratively to find the optimal geometrical parameters such that the circuit response matches the electrical target. Furthermore, for different electrical targets, this optimization process to find the geometrical or physical parameters will be repeated again and again, which is time consuming.

The other way to solve the inverse problem is the inverse modeling method [17]. In inverse modeling, the electrical parameters are defined as the inputs to the model and the geometrical or physical parameters are defined as the outputs of the model. Once the inverse model is obtained, it can provide the geometrical parameters immediately without iteratively evaluating the model for given electrical parameters. This method is much faster than the conventional iterative optimization process, and can afford to be used in repetitive designs or parameter extraction process. However, the analytical function or formula of the inverse model is difficult to obtain. Therefore, ANN has been used in learning the relationship between inputs and outputs in the inverse problem. Several works have been done in inverse modeling. In the direct inverse modeling technique [44], an ANN model is trained using data obtained by swapping the input and output data from the forward problem.

In [45] - [47], inverse surrogate modeling is used for expedited geometry scaling of compact microwave passives. In [48] and [49], inverse modeling is used for rapid EM-driven antenna dimension scaling. In [50], ANN inverse modeling is applied in a microwave sensor which is used to determine the proportion of fluids in a pipeline. In [51], ANN inverse modeling is used in transmitarray antenna design.

Even though the inverse modeling technique is much faster in the design process or parameter extraction process, the model is not easy to be trained well in many practical situations. It is because of the non-uniqueness problem of the inverse input-output relationship. The non-uniqueness problem means that different training samples with the same input values have contradictory output values (multivalued solutions). An even more challenging case is that different samples with similar but not identical input values have much different output values. This will cause the ambiguity whether these samples are contradictory or not. Since it is not possible to train an ANN model to match contradictory output values simultaneously for the same input values, the training error will stay large and the model accuracy will be low. The multivalued problem in hysteresis modeling has been studied in [99] from ANN point of view. The one-input multivalued data with hysteresis phenomena was represented by introducing a new parameter to the inputs of the model so that the one-input multivalued data becomes two-input single valued data. The non-uniqueness (multivalued) problem was addressed in [17] in microwave area. The data containing contradictory samples are divided into groups according to derivative information using a neural network forward model such that individual groups do not have the problem of contradictory data. Multiple inverse models

are built based on divided data groups, and are then combined to form a complete model.

In this chapter, we aim to further simplify the inverse modeling process and solve the non-uniqueness problem in a simpler and more automated way. In low dimension, the non-uniqueness problem can be solved by manual data segmentation and training multiple sub-models. While in high dimensional spaces, the non-uniqueness problem will become even more complex, as the numbers of the inputs and outputs increase. In this case, manual data segmentation or division can not solve the non-uniqueness problem easily. Furthermore, at some difficult situations, the number of conflicting data within a small range is more than that the derivative information can distinguish. Some contradictory data can have the same derivative signs. In this case, even after the derivative information is exhaustively used, the information of signs of derivatives is still not enough to distinguish multiple conflicts. Therefore, we need a more systematic method to solve the non-uniqueness problem.

In this chapter, we propose a multivalued neural network inverse modeling technique to associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. One set of geometrical or physical parameters is called one value of our proposed inverse model. Our proposed multivalued neural network is structured to accommodate multiple values for the model output. We also propose a new training error function to focus on matching each training sample using only one value of our proposed inverse model, while other values are free and can be trained to match other contradictory samples. In this way, our proposed multivalued

neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. Therefore, our proposed technique can solve the non-uniqueness problem in a simpler and more automated way compared to existing neural network inverse modeling techniques. This technique is illustrated by inverse modeling and parameter extraction of four microwave filter examples.

This chapter is organized as follows. In Section 4.2, the new structure of our proposed multivalued neural network inverse model is described, and a new error function to train our proposed inverse model is presented. A valid value selection process for our proposed inverse model is also proposed in Section 4.2. In Section 4.3, we demonstrate our proposed method using four microwave filter examples.

## 4.2 The Proposed Multivalued Neural Network Approach to Inverse Modeling

The forward problem of a microwave component is from geometrical or physical parameters to the electrical parameters. Let  $p$  and  $q$  represent the number of inputs and outputs of the forward problem. Let  $\bar{\mathbf{y}}$  be defined as a  $p$ -vector containing the inputs of the forward problem, which are usually the physical or geometrical parameters. Let  $\mathbf{z}$  be defined as a  $q$ -vector containing the outputs, which are usually the electrical parameters. The forward problem can be expressed as

$$\mathbf{z} = \mathbf{f}_{FWD}(\bar{\mathbf{y}}) \quad (4.1)$$

where  $\mathbf{f}_{FWD}$  represents the input-output relationship of the forward problem.

For design purposes or the parameter extraction of microwave filters, the information is often processed in the reverse direction in order to find the geometrical or physical parameters from given electrical parameters. This is called the inverse problem, where the inputs and outputs are the reverse of those in the forward problem.

#### 4.2.1 The Direct Inverse Modeling Problem

The inverse modeling technique is one of the methods to solve the inverse problem. The inputs to the inverse model are electrical parameters, and the outputs of the inverse model are geometrical or physical parameters. Since the analytical formula of the inverse input-output relationship does not exist, ANN becomes a logical choice because it can be trained to learn from the data in inverse modeling. In the direct inverse modeling technique [44], an ANN model is trained using data obtained by swapping the input and output data from the forward problem. Here  $q$  and  $p$  also represent the number of inputs and outputs of the inverse model of a microwave component. Let  $\bar{\mathbf{x}}$  be defined as a  $q$ -vector containing the inputs to the inverse model, which are usually the electrical parameters.  $\bar{\mathbf{y}}$  is also used as the outputs of the inverse model, which are usually the physical or geometrical parameters. The direct neural network inverse model can be expressed as

$$\bar{\mathbf{y}}(\bar{\mathbf{x}}, \mathbf{w}) = \mathbf{f}_{FWD}^{-1}(\bar{\mathbf{x}}) \quad (4.2)$$

where  $\mathbf{f}_{FWD}^{-1}$  represents the input-output relationship of the inverse problem,  $\bar{\mathbf{y}}(\bar{\mathbf{x}}, \mathbf{w})$  represents the neural network with  $\bar{\mathbf{x}}$  as inputs and  $\bar{\mathbf{y}}$  as outputs, and  $\mathbf{w}$  is defined as a vector containing neural network weights. The elements in the  $\mathbf{w}$  vector include bias values for all the neurons in each layer and the weight parameters linking neurons in different layers. Fig. 4.1(a) shows a standard ANN structure for the direct inverse modeling.

The direct inverse model can be obtained by a training process. The  $k_s$ th training sample pair is defined as  $(\bar{\mathbf{x}}_{k_s}, \mathbf{d}_{k_s})$ ,  $k_s \in T_r$ , where the subscript  $k_s$  is the sample index and  $T_r$  represents the index set of all the training samples.  $\bar{\mathbf{x}}_{k_s}$  is defined as the  $k_s$ th sample of  $\bar{\mathbf{x}}$ .  $\mathbf{d}_{k_s}$  is defined as the  $k_s$ th sample of  $\bar{\mathbf{y}}$  corresponding to the input  $\bar{\mathbf{x}}_{k_s}$ . The conventional training error function of this direct inverse modeling problem is defined as

$$E(\mathbf{w}) = \sum_{k \in T_r} \frac{1}{2} \|\bar{\mathbf{y}}(\bar{\mathbf{x}}_{k_s}, \mathbf{w}) - \mathbf{d}_{k_s}\|^2 \quad (4.3)$$

where  $\bar{\mathbf{y}}(\bar{\mathbf{x}}_{k_s}, \mathbf{w})$  is the neural network output for the input  $\bar{\mathbf{x}}_{k_s}$  [80].

When the inverse problem is relatively easy, e.g., the input-output relationship is unique, the training is easy and the model accuracy is high. In many situations, there will be the non-uniqueness problem of the inverse input-output relationship in the training data. The model can not be trained well, and training error will be large.

Here we illustrate the non-uniqueness problem through a microstrip rectangular inductor inverse modeling problem [17]. The effective quality factor  $Q_{eff}$  of the

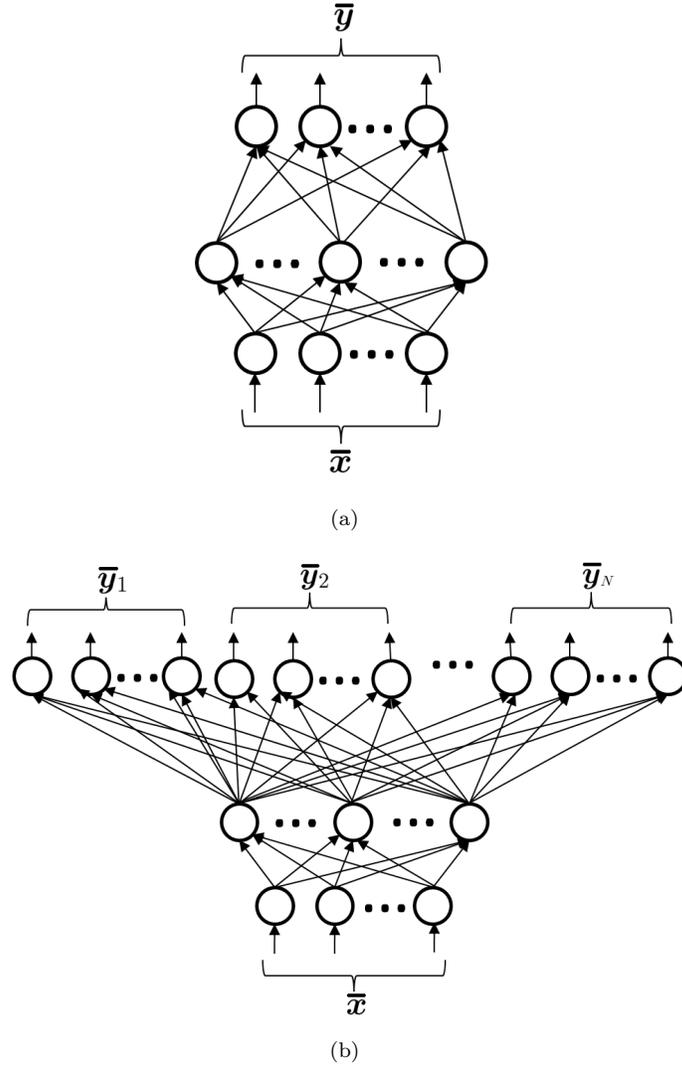
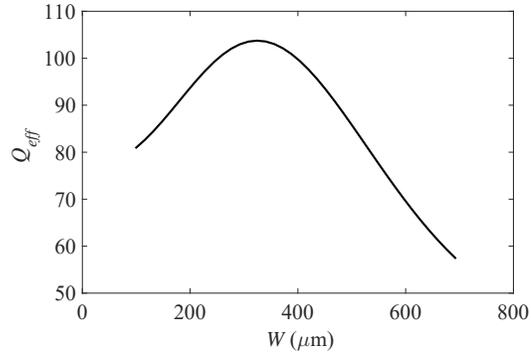
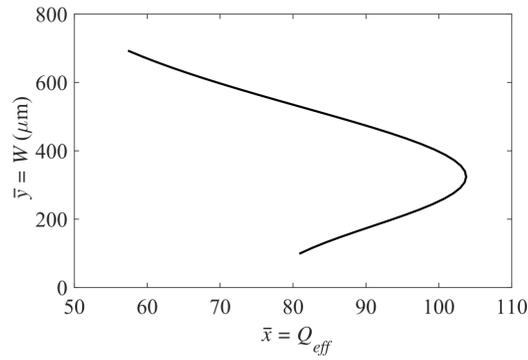


Figure 4.1: (a). A standard ANN structure for direct inverse modeling of a microwave component:  $\bar{\mathbf{x}}$  is a  $q$ -vector containing the inputs, which are usually the electrical parameters.  $\bar{\mathbf{y}}$  is a  $p$ -vector containing the outputs, which are usually the physical or geometrical parameters. (b). Our proposed multivalued neural network inverse modeling technique can associate a single set of electrical parameters with multiple sets of geometrical or physical parameters.  $\bar{\mathbf{x}}$  contains the inputs, which are usually the electrical parameters.  $\bar{\mathbf{y}}_i$  is the  $i$ th value of proposed inverse model. Each value  $\bar{\mathbf{y}}_i$  is a  $p$ -vector containing all the physical or geometrical parameters defined in the direct inverse modeling technique,  $i = 1, 2, \dots, N_v$ . The total number of output neurons in our proposed structure in (b) is  $N_v \cdot m$ .



(a)



(b)

| Sample No. | $Q_{eff}$ | $W$ ( $\mu\text{m}$ ) |
|------------|-----------|-----------------------|
| 1          | 99.78     | 250                   |
| 2          | 101.57    | 270                   |
| 3          | 102.85    | 290                   |
| 4          | 103.73    | 320                   |
| 5          | 102.84    | 360                   |
| 6          | 101.56    | 380                   |
| 7          | 99.78     | 400                   |

(c)

Figure 4.2: An illustration example: the microstrip rectangular inductor. (a) The forward problem is from the conductor width  $W$  to the effective quality factor  $Q_{eff}$ , and the input-output relationship is unique. (b) In the inverse problem, the effective quality factor  $Q_{eff}$  is used as the input of the inverse model, and the conductor width  $W$  as the output. The inverse input-output relationship is non-unique. (c) Some training samples of the inverse model.

inductor is used as the input to the inverse model, and the conductor width  $W$  as the output. The forward problem is shown in Fig. 4.2(a). The output of this forward problem is a single value (unique solution) for every input value. The inverse problem is shown in Fig. 4.2(b). It can be seen that in the inverse problem the output for some input values will have non-unique values (multivalued solutions) even though output value for every input is unique in the forward problem. This is why inverse modeling is usually more difficult than forward modeling. The table in Fig. 4.2(c) shows some training samples of inverse modeling of this microstrip inductor. From the table in Fig. 4.2(c), we can find that different training samples with the same input value have contradictory output values (multivalued solutions). For example, for Sample 1 and Sample 7, when  $Q_{eff} = 99.78$ ,  $W$  can be  $250 \mu\text{m}$  or  $400 \mu\text{m}$ . In inverse modeling, for the input value  $99.78$ , there will be 2 values (i.e.,  $250$  and  $400$ ) for the output. In this case, Sample 1 and Sample 7 are contradictory data. This will create conflicts between learning Sample 1 and learning Sample 7 while training ANN to learn all the samples. An even more challenging case is that different samples with similar but not identical input values have much different output values, such as Sample 2:  $Q_{eff} = 101.57$ ,  $W = 270 \mu\text{m}$ , and Sample 6:  $Q_{eff} = 101.56$ ,  $W = 380 \mu\text{m}$ . This will cause the ambiguity whether these samples are contradictory or not. Since it is not possible to train an ANN model to match several contradictory output values simultaneously for the same input values, the training errors in (4.3) will remain large. After training, the inverse model will not be accurate even in the training range, and the test error will also be large. In high dimension, the non-uniqueness problem in inverse modeling is even more

complex because the contradictory data in multi-dimension can not be visualized graphically. Manual data segmentation or division can not solve the non-uniqueness problem easily. Therefore, a systematic algorithm for solving the non-uniqueness problem is very important.

### 4.2.2 The Proposed Multivalued Neural Network Inverse Model

In order to solve the non-uniqueness problem in inverse modeling in a simple and automated way, we propose a multivalued neural network inverse modeling technique to associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. One set of geometrical or physical parameters is called one value of our proposed inverse model. Our proposed multivalued neural network inverse model is structured to accommodate multiple values for the model output. Instead of manually detecting the contradictory data in the training set, we want to use ANN learning ability to automatically redirect contradictory information into different values of the proposed inverse model. Here we define  $N_v$  as the number of values of the proposed inverse model. We propose a multivalued neural network inverse model by repeating the outputs of the direct inverse model in (4.2)  $N_v$  times as follows

$$[\bar{\mathbf{y}}_1^T \ \bar{\mathbf{y}}_2^T \ \cdots \ \bar{\mathbf{y}}_{N_v}^T]^T = \mathbf{g}_{ANN}(\bar{\mathbf{x}}, \mathbf{w}) \quad (4.4)$$

where  $\mathbf{g}_{ANN}$  is a multilayer perceptron (MLP) representing our proposed multivalued neural network inverse model. For the hidden layer, the activation function is the sigmoid function. For the output layer, the activation function is the linear function.

$\bar{\mathbf{y}}_i$  is the  $i$ th value of the proposed inverse model for input  $\bar{\mathbf{x}}, i = 1, 2, \dots, N_v$ . Each value  $\bar{\mathbf{y}}_i$  of our proposed multivalued neural network is a  $p$ -vector containing all the physical or geometrical parameters defined in the direct inverse modeling technique. Fig. 4.1(b) shows an example of our proposed multivalued neural network.

For the convenience of representing each value  $\bar{\mathbf{y}}_i$  using our proposed  $\mathbf{g}_{ANN}$ , we define an  $p \times (N_v \cdot p)$  matrix  $\mathbf{U}_i$  containing zeros everywhere except for the  $(h, (i - 1) \cdot p + h)$ th location, where the value is 1, for  $h = 1, 2, \dots, p$ . Then  $\bar{\mathbf{y}}_i$  can be calculated by

$$\begin{aligned} \bar{\mathbf{y}}_i &= \mathbf{U}_i \cdot \mathbf{g}_{ANN}(\bar{\mathbf{x}}, \mathbf{w}); \\ i &= 1, 2, \dots, N_v. \end{aligned} \tag{4.5}$$

Since we have increased the total number of outputs of  $\mathbf{g}_{ANN}$  to  $N_v \cdot m$ , the training error function in (4.3) is not suitable for our proposed multivalued neural network. The brute force method to solve the training problem of this proposed neural network is to reformat all the training data, such that each training sample contains all the contradictory outputs for the same inputs. However, for those conflicts, where the input values are not exactly the same but very similar while the output values are quite different, the contradictory data can not be reformatted together. Even if all the contradictory outputs for every input could be found, the consistency of sequences of contradictory outputs for different inputs could not be ensured. Thus the data reformatting method fails in training our proposed inverse model. Therefore, we propose a new error function for training our proposed multivalued neural network  $\mathbf{g}_{ANN}$ , such that our proposed multivalued neural network

can redirect contradictory information into different values of the proposed inverse model, and learn all the training data bypassing the difficulty of having to detect and sort out all the multivalued solutions.

### 4.2.3 The Proposed Training and Test Error Functions of the Proposed Multivalued Neural Network Inverse Modeling

To make sure our proposed multivalued neural network can automatically learn contradictory data in the training set, such that different values  $\bar{\mathbf{y}}_i$ , for  $i = 1, 2, \dots, N_v$ , can match different conflicts for the same or very similar input  $\bar{\mathbf{x}}$ , we propose a new training error as

$$\tilde{E}(\mathbf{w}) = \sum_{k_s \in T_r} E_{k_s}(\mathbf{w}) \quad (4.6)$$

where  $T_r$  represents the index set of all the training samples and  $E_{k_s}(\mathbf{w})$  is the training error for the  $k_s$ th training sample which is defined as

$$E_{k_s}(\mathbf{w}) = \left( \sum_{i=1}^{N_v} \frac{1}{e_{i,k_s}(\mathbf{w})} \right)^{-1} \quad (4.7)$$

where  $e_{i,k_s}(\mathbf{w})$  is defined as the error between  $\bar{\mathbf{y}}_i$  and  $\mathbf{d}_{k_s}$  as follows

$$\begin{aligned} e_{i,k_s}(\mathbf{w}) &= \frac{1}{2} \|\bar{\mathbf{y}}_i - \mathbf{d}_{k_s}\|^2 \\ &= \frac{1}{2} \|\mathbf{U}_i \cdot \mathbf{g}_{ANN}(\bar{\mathbf{x}}_{k_s}, \mathbf{w}) - \mathbf{d}_{k_s}\|^2, \\ &i = 1, 2, \dots, N_v \end{aligned} \quad (4.8)$$

where  $\bar{\mathbf{y}}_i$  is the  $i$ th value of our proposed inverse model  $\mathbf{g}_{ANN}(\bar{\mathbf{x}}_{k_s}, \mathbf{w})$  for the input  $\bar{\mathbf{x}}_{k_s}$ .

When one value  $\bar{\mathbf{y}}_i$  of the proposed inverse model fits the training sample  $\mathbf{d}_{k_s}$  perfectly, it means the error  $e_{i,k_s}(\mathbf{w})$  for this value is very small. The small error  $e_{i,k_s}(\mathbf{w})$  of one value will result in a small value in the proposed overall error function  $E_{k_s}(\mathbf{w})$  in (4.7) for the  $k_s$ th sample. Meanwhile the error  $e_{i,k_s}(\mathbf{w})$  for other values can be large, which means the other values are free and can be different from the current training sample  $\mathbf{d}_{k_s}$ . Therefore, the proposed error function  $E_{k_s}(\mathbf{w})$  in (4.7) focuses on matching the training sample with one of the values of multivalued neural network inverse model which is the closest to the current training sample while ignoring other values. If there is another contradictory sample in the training data, our proposed error function will automatically match it using another value of our multivalued neural network inverse model. This feature gives our proposed multivalued neural network the ability to learn the contradictory data in the same training. During training, the multiple values of our proposed model output can automatically fit multiple contradictory samples, thus solving the non-uniqueness problem.

At the beginning of the training process, the ANN weights  $\mathbf{w}$  are initialized as small random numbers [133]. An optimization method, such as quasi-Newton method or conjugate gradient method, is applied to minimize the training error in (4.6) with  $\mathbf{w}$  as the optimization variables. ANN weights  $\mathbf{w}$  will be updated by optimization during the training process.

After training is finished, we perform the test of the inverse model using test data. Because our multivalued neural network inverse model is structured to have multiple values for the model output, the conventional formula for test error, which is usually defined similarly to the training error in (4.3), is not suitable for our proposed model. Therefore we define a new test error function of our proposed multivalued neural network inverse model as

$$\begin{aligned}
T(\mathbf{w}) &= \sum_{k_s \in T_v} \min_i \left\{ \frac{1}{2} \|\bar{\mathbf{y}}_i - \mathbf{d}_{k_s}\|^2 \right\} \\
&= \sum_{k_s \in T_v} \min_i \left\{ \frac{1}{2} \|\mathbf{U}_i \cdot \mathbf{g}_{ANN}(\bar{\mathbf{x}}_{k_s}, \mathbf{w}) - \mathbf{d}_{k_s}\|^2 \right\}; \\
&\quad i = 1, 2, \dots, N_v
\end{aligned} \tag{4.9}$$

where  $T_v$  represents the index set of all the test samples. In our proposed test error function (4.9), we calculate the error for every value of our multivalued neural network inverse model, and the value which is closest to the test data is selected towards the calculation of the total test error. The error function proposed in (4.9) will produce a measure of the test error compatible to that calculated for conventional neural network using the error function (4.3).

#### 4.2.4 The Proposed Method of Selecting the Valid Values of the Proposed Inverse Model

To get an accurate inverse model, the number of values  $N_v$  of our proposed multivalued neural network should be equal to the maximum number of the conflicts in the training data for all possible inputs. However, after the training of our proposed multivalued neural network using error functions (4.6) - (4.8), the number of the

conflicts may not always be equal to  $N_v$  for a particular given input. For some inputs, the numbers of the conflicts are equal to  $N_v$ . In this case, all values of the proposed multivalued neural network inverse model are valid. For some inputs, the numbers of conflicts can be fewer than  $N_v$ . In this case, out of  $N_v$  values provided by our proposed inverse model, some values are valid and some values are dummy. Since our proposed inverse model will always provide  $N_v$  values for the model output for any given input, we need to propose a method to distinguish the valid value(s) from dummy values out of all the values  $\bar{\mathbf{y}}_i$ , for  $i = 1, 2, \dots, N_v$ .

Motivated by the sub-model combining technique in [17], we propose to use the forward model to help decide which values are the valid values. Each  $\bar{\mathbf{y}}_i$  obtained in our proposed inverse model is fed into the forward model in (4.1) as inputs,  $i = 1, 2, \dots, N_v$ . After forward model evaluation, we get  $N_v$  sets of output.  $\mathbf{z}_i$  represents the output of the forward model for  $\bar{\mathbf{y}}_i$ ,  $i = 1, 2, \dots, N$ . Apparently, the output  $\mathbf{z}$  of the forward model and the input  $\bar{\mathbf{x}}$  of the inverse model contain the same electrical parameters. The errors  $E_o$  between  $\mathbf{z}_i$  and  $\bar{\mathbf{x}}$  for different  $\bar{\mathbf{y}}_i$  can help us to decide which value(s) is (are) the valid value(s). We define  $E_o(\bar{\mathbf{y}}_i)$  as

$$E_o(\bar{\mathbf{y}}_i) = \|\mathbf{z}_i - \bar{\mathbf{x}}\| \quad (4.10)$$

where

$$\mathbf{z}_i = \mathbf{f}_{FWD}(\bar{\mathbf{y}}_i); \quad i = 1, 2, \dots, N_v \quad (4.11)$$

and  $\mathbf{f}_{FWD}(\cdot)$  is the forward model defined in (4.1) with  $\bar{\mathbf{y}}_i$  as the input and  $\mathbf{z}_i$  as

the output. If the condition

$$E_o(\bar{\mathbf{y}}_i) = \|\mathbf{f}_{FWD}(\bar{\mathbf{y}}_i) - \bar{\mathbf{x}}\| \leq \varepsilon \quad (4.12)$$

is satisfied,  $\bar{\mathbf{y}}_i$  is called a valid value. Here  $\varepsilon$  is a predefined constant. If

$$E_o(\bar{\mathbf{y}}_i) = \|\mathbf{f}_{FWD}(\bar{\mathbf{y}}_i) - \bar{\mathbf{x}}\| > \varepsilon \quad (4.13)$$

$\bar{\mathbf{y}}_i$  is called an invalid (dummy) value.

Let  $V$  be defined as the index set of valid values, i.e.,

$$V = \{i | E_o(\bar{\mathbf{y}}_i) \leq \varepsilon; i = 1, 2, \dots, N_v\}. \quad (4.14)$$

The valid values of our proposed inverse model can be expressed as

$$\bar{\mathbf{y}} \in \{\bar{\mathbf{y}}_i | i \in V\} \quad (4.15)$$

Fig. 4.3 shows the diagram for the method of selecting the valid values of the proposed multivalued neural network inverse model.

#### 4.2.5 Application of the Proposed Multivalued Neural Network Inverse Model to Parameter Extraction

After training in Section II-C and selecting valid values in Section II-D, our proposed inverse model can provide one or more sets of valid values for a given set of inputs. It means the proposed inverse model will provide one or more sets of geometrical or physical parameters for a given set of electrical parameters, such as  $S$ -parameters.

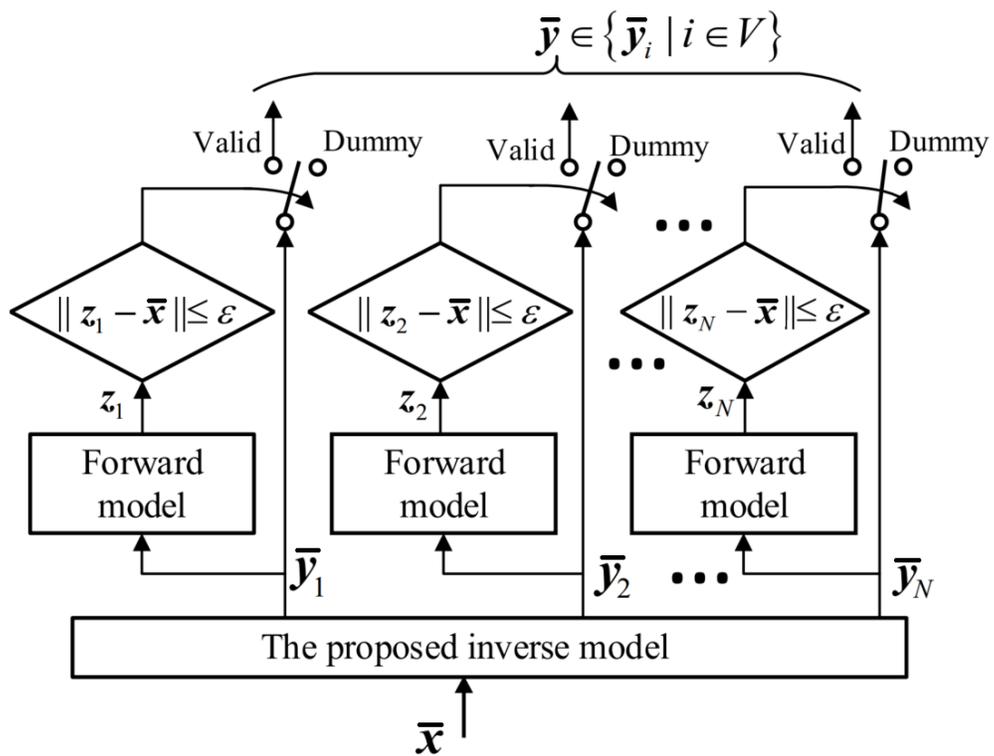


Figure 4.3: The diagram for the proposed method of selecting the valid values of the proposed multivalued neural network inverse model.

For given  $S$ -parameters, there could be multiple solutions of geometrical or physical parameters (multivalued solutions). However, one such solution of geometrical or physical parameters is true for a particular circuit. This single solution of geometrical or physical parameters is called the desired parameters in this chapter. In this subsection, we further use our proposed multivalued neural network inverse model to convert multiple valid solutions to a single solution.

Motivated by the multi-circuit optimization concept in [22], we use our proposed multivalued neural network inverse model in the multi-circuit process to extract the desired parameters among the valid values of our proposed inverse model. The circuit under consideration, of which the parameters are to be extracted, is taken as the 1st circuit. Then one of the circuit parameters is changed to obtain the 2nd circuit. The electrical parameters of both circuits are fed into our proposed inverse model. After the valid value selection process in Section II-D, we can get multiple valid sets of geometrical or physical parameters for both circuits. Then we compare each valid set for the 1st circuit with each valid set for the 2nd circuit. If the difference between one set for the 1st circuit and one set for the 2nd circuit matches the change of the circuit parameters, that valid set of geometrical or physical parameters for the 1st circuit are the extracted parameters. This is called a 2-circuit ANN process for parameter extraction using our proposed inverse model.

An extension to 2-circuit ANN process for parameter extraction is the multi-circuit ANN process. If the process with 2 circuits still leads to multiple solutions (i.e., the differences of multiple pairs of valid values match the adjustment of the

physical parameter), more circuits can be added in a similar way to increase the uniqueness of the extracted parameters.

The multi-circuit ANN process for parameter extraction using our proposed inverse model only needs several evaluations of the inverse model and the forward model for one extraction process. ANN evaluation is very fast and can be done instantaneously. This is in contrast with the conventional multi-circuit optimization process in [93] which needs repetitive evaluation of forward model in an iterative optimization process. Our proposed process is much faster than the multi-circuit optimization process. Furthermore, for parameter extractions with different electrical parameters, our proposed process can extract the desired parameters instantly. For the same situation, the conventional multi-circuit optimization process will need to repeat all the iterative optimization process again. Therefore, the multi-circuit ANN process using our proposed multivalued neural network inverse model is more efficient.

#### **4.2.6 Discussion**

In this subsection, we firstly discuss the differences between the techniques in this chapter and that in the existing publications [17] and [99]. Direct inverse modeling with a single set of outputs cannot match multivalued solutions in training data. In order to separate the non-unique or multivalued solutions, additional information is needed. Existing techniques to solve the multivalued problem, e.g., [17] and [99], are to add additional input information (derivative information [17] or neighborhood

information [99]). This is because the new input information, such as derivatives inputs [17], is different for the different values at the output. In the present chapter, we propose an alternative method by using multiple sets of outputs to allow the contradictory information to be learned by the multivalued outputs of the model.

Our proposed multivalued neural network can solve the non-uniqueness problem by repeating the outputs of the direct inverse model in (4.2)  $N_v$  times which is shown in (4.4). Compared to the existing technique [17], one of the drawbacks of our proposed technique is the long training time because we may need to train the inverse model several times with different values of  $N_v$ . The final value of  $N_v$  is the one with the lowest training error. However, the benefit of our technique over [17] is the elimination of human-intensive processing of signs of derivatives which becomes complex when the model inputs have many dimensions.

The number of the training samples is usually affected by the nonlinearity of the input-output relationship, and the input dimension (number of model inputs). For a particular inverse modeling problem within a fixed range, repeating outputs  $N_v$  times does not increase the nonlinearity over original inverse model with single outputs. Usually adding more inputs will increase the number of training data. In our proposed formulation, the input dimension remains the same as that of conventional inverse model. Therefore, the repetition of the outputs will not necessarily require more training samples.

Secondly, we discuss the potential of competitive learning [134] for microwave inverse modeling. In the competitive learning technique, the mechanism permits

the multiple outputs of an ANN to compete for the right to respond to a given set of inputs, such that only one output is active at a time. This method has been used in classification and cognition science, e.g., [135]. Possible applications of competitive learning in microwave inverse modeling could be to help decide which value among all the contradictory values will win for a given input. In fact, in order to permit competitive learning for microwave inverse modeling, it may be necessary to use our proposed multivalued output so that competition between different values is possible. Therefore, our proposed method to repeat the output multiple times opens the door to permit competitive learning, which is otherwise not possible in existing single-value formulation of inverse modeling. How to perform competitive learning for microwave inverse modeling would be a potential future topic.

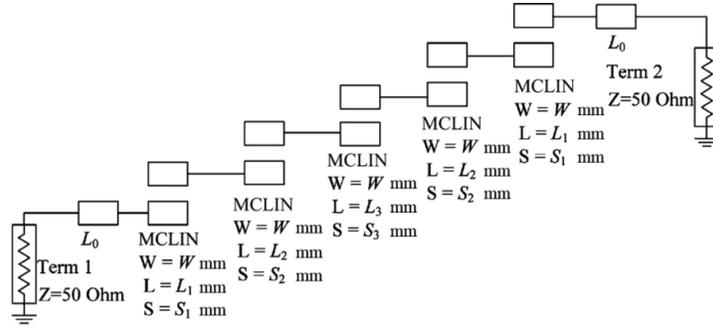
Thirdly, we discuss the possibility of different learning-by-examples (LBEs) methodologies for inverse modeling. For typical LBEs methods, such as ANN techniques, support vector machine (SVM) regression [96], and Gaussian process (GP) approach [97], the LBEs models work well when the training data is not contradictory. For the standard LBEs methods, each output of the model will only provide a single value for a given input. Therefore, these LBEs methods face the same problem when they are used to learn contradictory data, i.e., multivalued solutions in inverse modeling. To solve the problem of multivalued solutions, we need new formulations of the inverse model regardless whether ANN, SVM or GP are used. The choice between different LBEs methods (such as ANN, SVM and GP) may affect the accuracy of modeling (under the same formulation of model inputs and outputs), e.g., [136] and [137]. SVM and GP have good generalization capability when training data is lim-

ited. ANN is well suited to the case when the amount of training data is large. In our examples, the challenge is contradictory data in a large data base. Therefore, ANN has been used in this chapter. However, it is also possible to apply the proposed multivalued output formulation to enable SVM and GP to solve multivalued solution problems. This can be a possible future direction of research.

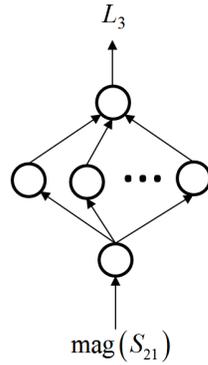
## 4.3 Examples

### 4.3.1 Illustrative Example: the Inverse Modeling of a Microstrip Bandpass Filter

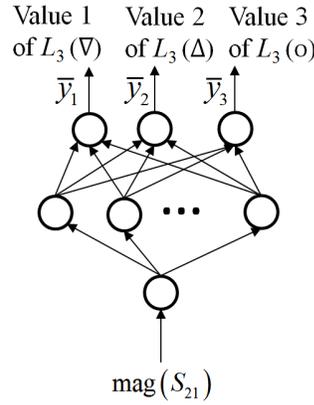
In this example, we illustrate our proposed technique through a simple inverse modeling problem of a microstrip bandpass filter, shown in Fig. 4.4(a). For simplicity of illustration, here we assume that the length of the parallel coupled-lined section  $L_3$  is the input of the forward model, and assume that the magnitude of  $S_{21}$  at 4 GHz is the output of the forward model. The forward problem is shown in Fig. 4.5(a). In inverse modeling of this microstrip bandpass filter, we assume the magnitude of  $S_{21}$  at 4 GHz as the input, and  $L_3$  as the output, i.e.,  $\bar{x} = \text{mag}(S_{21})|_{4\text{GHz}}$ , and  $\bar{y} = L_3$ . The inverse problem is shown in Fig. 4.5(b). From the figure, we can see that the input-output relationship of inverse problem is non-unique even though the input-output relationship of forward problem is unique. In Fig. 4.5(b), in the range  $\bar{x} \in [0.55, 0.8)$ , the output  $\bar{y}$  has unique solution. In the range  $\bar{x} \in [0.8, 1]$ , the output  $\bar{y}$  has multivalued solutions for the same input value, e.g., if  $\bar{x} = 0.9$ ,  $\bar{y}$  can have three different values: 3.96, 3.99, and 4.05. In other words, for  $\text{mag}(S_{21})|_{4\text{GHz}} = 0.9$ ,



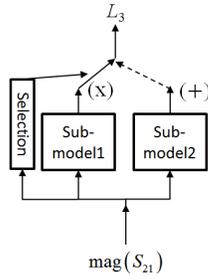
(a)



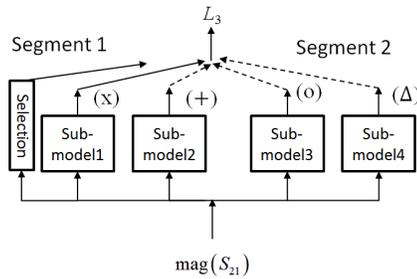
(b)



(c)



(d)



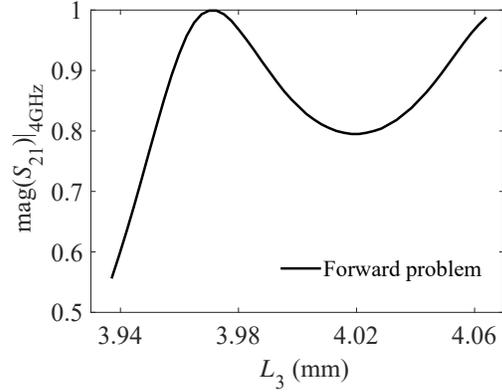
(e)

Figure 4.4: (a) The inverse modeling of a microstrip bandpass filter example:  $\bar{x} = \text{mag}(S_{21})|_{4GHz}$ , and  $\bar{y} = L_3$ . (b) The direct inverse model has one input which is  $\text{mag}(S_{21})|_{4GHz}$ , and one output  $L_3$ . (c) The proposed multivalued neural network inverse model with 3 values:  $\bar{y}_1$ ,  $\bar{y}_2$ , and  $\bar{y}_3$  represent Value 1, Value 2 and Value 3 of the length  $L_3$ , respectively. (d) The existing inverse modeling technique [17] using 2 sub-models. (e) The existing inverse modeling technique [17] using 4 sub-models. The symbols ( $\nabla$ ,  $\Delta$ ,  $\circ$ ,  $\times$ , and  $+$ ) are used to label the different model outputs.

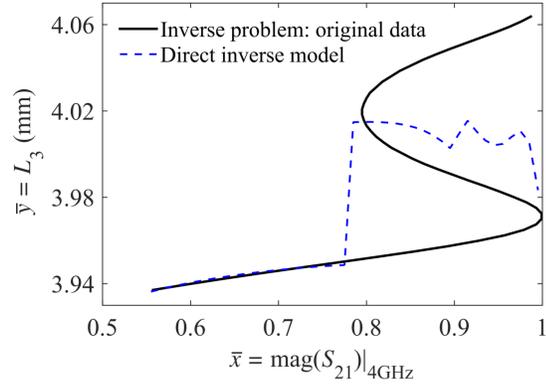
there are three different solutions of length  $L_3$ , i.e.,  $3.96mm$ ,  $3.99mm$ , and  $4.05mm$ . From the ANN perspective, the multivalued solutions of  $\bar{y}$  for the same input  $\bar{x}$  mean contradictory data or conflicting information for ANN training. For example, the training data of  $(\bar{x} = 0.9, \bar{y} = 3.96)$  is contradictory with the training data of  $(\bar{x} = 0.9, \bar{y} = 4.05)$ . In other words, the two sets of training data, i.e.,  $(\bar{x} = 0.9, \bar{y} = 3.96)$  and  $(\bar{x} = 0.9, \bar{y} = 4.05)$ , provide conflicting information for ANN training.

We firstly illustrate the inverse modeling problem using the conventional direct ANN inverse modeling technique. The structure of the direct inverse model is shown in Fig. 4.4(b). The direct inverse model has one input which is  $\text{mag}(S_{21})|_{4GHz}$ , and one output  $L_3$ . The training result is shown in Fig. 4.5(b). In the range where there is no conflict, i.e., in the range  $\bar{x} \in [0.55, 0.8)$ , the direct inverse model behaves well. However, in the range where there are contradictory data, i.e., in the range  $\bar{x} \in [0.8, 1]$ , the single output in direct inverse modeling technique cannot match all the contradictory data at the same time. The result is a compromise, where the output is approximately the average of the contradictory values (multivalued solutions). The training error in this range is always large, and can never be reduced even if more training iterations are used.

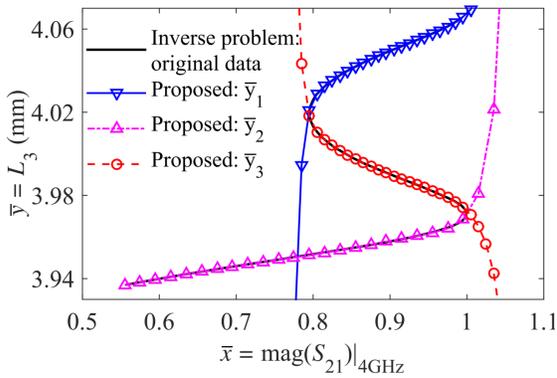
Then we use our proposed multivalued neural network inverse model with 3 values to solve this example. Our proposed multivalued neural network is structured to have multiple values for the model output, which is shown in Fig. 4.4(c).  $\bar{y}_1$ ,  $\bar{y}_2$ , and  $\bar{y}_3$  represent Value 1, Value 2 and Value 3 of the length  $L_3$ , respectively. Fig. 4.5(c) shows the modeling results using our proposed multivalued neural network



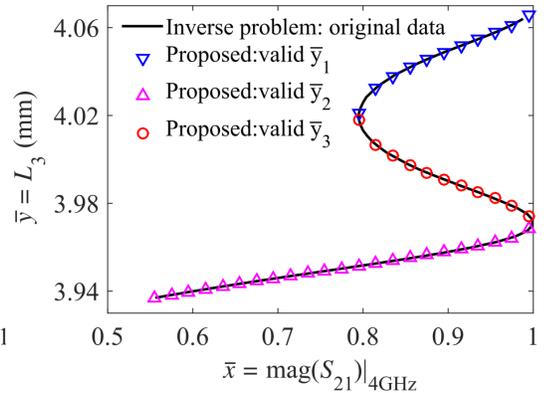
(a)



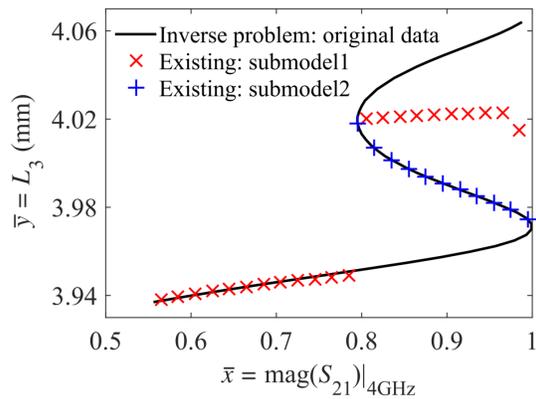
(b)



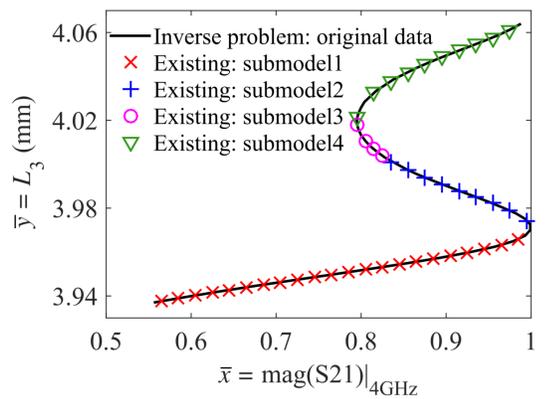
(c)



(d)



(e)



(f)

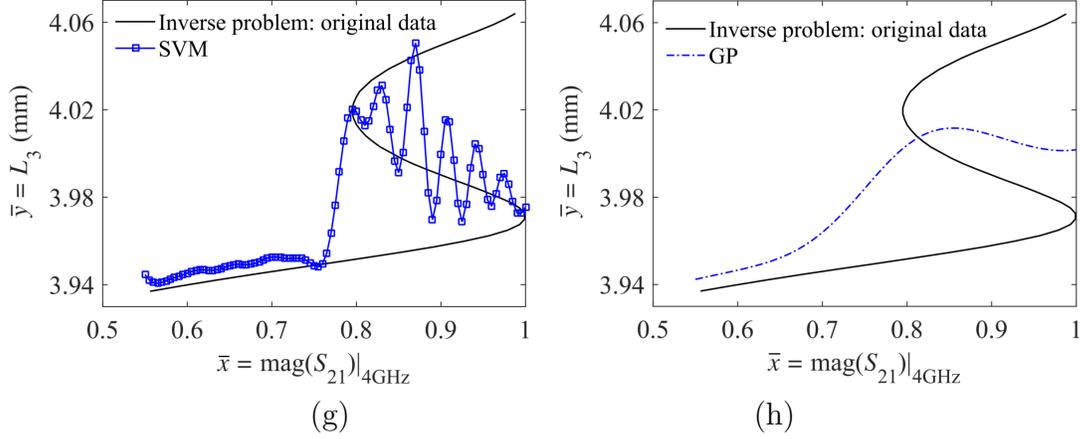


Figure 4.5: Illustration of different inverse modeling techniques for the microstrip bandpass filter example. (a) The input-output relationship of forward problem is unique. (b) The inverse problem and the direct ANN inverse modeling technique. (c) Our proposed inverse modeling technique with 3 values ( $\nabla$  for  $\bar{y}_1$ ,  $\Delta$  for  $\bar{y}_2$ , and  $\circ$  for  $\bar{y}_3$ ) before valid value selection process. (d) After valid value selection, our proposed inverse modeling technique with 3 values ( $\nabla$  for  $\bar{y}_1$ ,  $\Delta$  for  $\bar{y}_2$ , and  $\circ$  for  $\bar{y}_3$ ) can learn all the training data automatically, matching all the contradictory data at the same time, and achieving good accuracy. (e) The existing technique in [17] with 2 sub-models. Inside the first sub-model, there is still contradictory data, therefore the sub-model #1 ( $\times$ ) is not accurate. The sub-model #2 (+) is good. (f) The existing technique in [17] with 4 sub-models can achieve good model accuracy. (g) SVM inverse model behaves well at the range where there is no contradictory data. Where there are three different solutions for the same input (i.e., multivalued output for the same input), the SVM model cannot learn all the three contradictory values. (h) GP inverse model tries to approach the average of the contradictory data.

without valid value selection. From Fig. 4.5(c), we can find that two different values (such as  $\bar{y}_1$  and  $\bar{y}_3$ , or  $\bar{y}_2$  and  $\bar{y}_3$  in Fig. 5(c)) of our proposed multivalued neural network inverse model approach the same point, where the derivative of the input-output relationship changes signs. At such points, the errors  $e_{i,k_s}$  in (4.8) for two different  $i$  will be both minimized, which results in a small training error in (4.7). In other words, the values for both sets of outputs of our proposed model at such points will approach the same training data. After using the selection method proposed in Section II-D, the valid values are shown in Fig. 4.5(d). As we can see from the figure, in the range where there are three different solutions for the same input, the values  $\bar{y}_1$ ,  $\bar{y}_2$ , and  $\bar{y}_3$  match three different solutions, respectively, thus solving the non-uniqueness problem. So in this range,  $\bar{y}_1$ ,  $\bar{y}_2$ , and  $\bar{y}_3$  are all valid values. In the range where there is no conflict, only  $\bar{y}_2$  fits the training data. So  $\bar{y}_2$  will be the valid value, and both  $\bar{y}_1$  and  $\bar{y}_3$  are invalid (dummy) values. From Fig. 4.5(c) we can find that our proposed inverse modeling technique not only fits the training data very well, but also distinguishes valid values from invalid ones.

For comparison purposes, we also use the existing inverse modeling technique in [17] to solve this problem. In the data range from  $L_3 = 3.93mm$  to  $L_3 = 4.07mm$ , we separate the training data into two groups according to the signs of derivative information  $\frac{\partial|S_{21}|}{\partial L_3}$ : Group #1. according to the positive derivative, the data range from  $L_3 = 3.93mm$  to  $L_3 = 3.97mm$  and the data range from  $L_3 = 4.02mm$  to  $L_3 = 4.07mm$  belong to the first group; Group #2. according to the negative derivative, the data range from  $L_3 = 3.97mm$  to  $L_3 = 4.02mm$  belongs to the second group. Since the signs of derivative information in this case can only distinguish

between two groups, the existing inverse modeling technique can not solve this three-value non-uniqueness problem in this data range. As we can see from Fig. 4.5(d), the sub-model #1 (represented by “×” in Fig. 4.5(d)) is not accurate. The sub-model #2 (represented by “+” in Fig. 4.5(d)) is good. From the comparison, our proposed inverse modeling technique can achieve better model accuracy without having to separate the training data and then train sub-models. The proposed inverse modeling technique solves the non-uniqueness problem in a much simpler and more automated way. Furthermore, our proposed technique can increase the reliability of the solution over that of [17].

To further improve the model accuracy using the existing inverse modeling technique in [17], we further segment this small data range from  $L_3 = 3.93mm$  to  $L_3 = 4.07mm$  into two parts. Without prior knowledge of this data range in this example, we usually segment the data into several equal parts along the geometrical parameter  $L_3$  to solve the inverse modeling problem. The data segmentation will become even more difficult in high dimension especially when the contradictory data can not be graphically visualized. Thus the training data is segmented into two equal parts: Segment 1 includes the data range from  $L_3 = 3.93mm$  to  $L_3 = 4.00mm$ , and Segment 2 includes data range from  $L_3 = 4.00mm$  to  $L_3 = 4.07mm$ . Within each segment, the derivative information will be used to separate the contradictory data into different groups. Thus, we will have four groups: Group #1. according to the positive derivative in Segment 1, the data range from  $L_3 = 3.93mm$  to  $L_3 = 3.97mm$  belongs to the first group; Group #2. according to the negative derivative in Segment 1, the data range from  $L_3 = 3.97mm$  to  $L_3 = 4.00mm$  belongs

to the second group; Group #3. according to the negative derivative in Segment 2, the data range from  $L_3 = 4.00mm$  to  $L_3 = 4.02mm$  belongs to the third group; Group #4. according to the positive derivative in Segment 2, the data range from  $L_3 = 4.02mm$  to  $L_3 = 4.07mm$  belongs to the fourth group. The model structure is shown in Fig. 4.4(e). After training four sub-models using four different groups of data, the result is shown in Fig. 4.5(e). The existing technique in [17] using four sub-models can achieve good model accuracy. Compared to this existing technique, our proposed inverse modeling technique can solve this inverse problem using only one proposed ANN structure with three values after one overall training, and our proposed technique does not need to separate training data nor train sub-models. Therefore, our proposed technique can achieve good accuracy in a much simpler and more automated way.

We also provide a comparison with standard SVM and GP approaches. The inverse modeling results using SVM with a single output (one value) and GP with a single output (one value) are shown in Fig. 4.5(g) and 4.5(h), respectively. The SVM inverse model with a single output behaves well at the range where there is no contradictory data. Where there are three different solutions, the SVM model with a single output cannot learn all the three contradictory values. GP inverse model with a single output tries to approach the average of the contradictory data. The output fails to match any valid training data. Therefore, the standard SVM and GP methods with a single set of output (one value) cannot solve the problem of multivalued solutions in inverse modeling. In Fig. 4.5(d), it shows that by formulating the inverse model with multiple outputs, i.e., repeating the outputs

Table 4.1: Definition of the Training and Test Data Range for the Fourth-Order Dielectric Resonator Filter Example

|          | Min     | Nominal | Max    |
|----------|---------|---------|--------|
| $M_{12}$ | 0.1106  | 0.9106  | 1.7106 |
| $M_{23}$ | -0.1001 | 0.6999  | 1.4999 |
| $M_{34}$ | 0.1106  | 0.9106  | 1.7106 |

of the direct ANN inverse model  $N_v$  times, our proposed method can solve the multivalued problem of the ANN inverse modeling.

### 4.3.2 The Inverse Modeling of A Fourth-Order Dielectric Resonator Filter

In this example, we illustrate the proposed technique through the inverse modeling of a fourth-order dielectric resonator filter [2]. This filter is supposed to have the following specifications: the filter order is 4; the center frequency is 1930 MHz; the bandwidth is 15 MHz; the return loss is 20 dB; the ideal coupling parameters are

$$M = \begin{bmatrix} 0 & 0.9106 & 0 & 0 \\ 0.9106 & 0 & 0.6999 & 0 \\ 0 & 0.6999 & 0 & 0.9106 \\ 0 & 0 & 0.9106 & 0 \end{bmatrix} \quad (4.16)$$

The diagonal elements  $M_{11} = M_{22} = M_{33} = M_{44} = 0$ , and  $M_{ij} = M_{ji}$ ,  $i \neq j$ . As a result, the non-zero coupling parameters are  $M_{12}$ ,  $M_{23}$  and  $M_{34}$ .

For this example, we try to build an inverse model between the coupling parameters and the  $S$ -parameter. Here we choose both  $S_{11}$  and  $S_{21}$  (in dB) at 11 frequency samples as the inputs  $\bar{\mathbf{x}}$ , and 3 coupling parameters as the outputs, i.e.,  $\bar{\mathbf{y}} = [M_{12} \ M_{23} \ M_{34}]^T$ . Therefore,  $n = 22$ ,  $m = 3$ . To generate the training and test data, we choose a tolerance of  $\pm 0.8$  for every coupling parameter, which is shown in Table 4.1. Then we use uniformly distributed random values in this range to get 100000 sets of input-output data as the training data, and 50000 sets as the test data.

We firstly illustrate the inverse modeling problem using the conventional direct inverse modeling technique. We train a 3-layer MLP with 22 input neurons, 60 hidden neurons, and 3 output neurons, i.e., MLP: 22-60-3, to learn all the training data. The training error and test error are shown in Table 4.2. From the table, we can find that even though we try to reduce the training error to a relatively small value with enough hidden neurons and training iterations, the test error still stays large because of the non-uniqueness problem. The large test error of the direct inverse modeling technique in Table 4.2 means that, for given  $S$ -parameter inputs, the coupling parameters provided by the direct inverse model can not match the desired coupling parameters in the test data.

Then we apply our proposed multivalued neural network inverse model with 2 values to this example. The same number of hidden neurons are used in our proposed multivalued neural network with 2 values. There are 22 input neurons and 6 output neurons (2 values of 3 coupling parameters) in our proposed multivalued neural

Table 4.2: The Comparison of Modeling Results Using Different Inverse Modeling Methods for the Fourth-Order Dielectric Resonator Filter Example

| Inverse modeling methods          | Training error* | Test error |
|-----------------------------------|-----------------|------------|
| Direct inverse modeling technique | 5.09%           | 10.43%     |
| Proposed technique with 2 values  | 1.63%           | 1.84%      |

\*To be compatible with the errors in the direct inverse modeling technique, both the training error and test error in the proposed modeling technique are calculated by (4.9).

network, i.e., MLP: 22-60-6. The comparison of the modeling results is shown in Table 4.2. Our proposed multivalued neural network inverse model is obtained using our proposed training error functions (4.6) - (4.8) in one overall training. Both the training error and test error of our proposed inverse model are very small. The small training and test errors of the proposed inverse modeling technique in Table 4.2 mean that, for given  $S$ -parameter inputs, one of the two values provided by the proposed inverse model for the coupling parameters can match the desired coupling parameters in the training and test data very well. Therefore, our proposed technique can solve the non-uniqueness problem in a simple and automated way.

After being trained, our inverse model (with 2 values) will always provide two sets of coupling parameters for any set of  $S$ -parameter input. With valid value selection process, our proposed inverse model can always provide reliable solutions of coupling parameters. Here we use two test samples to illustrate how our valid value selection process works.

For Sample #1, the desired  $S$ -parameters (both  $S_{11}$  and  $S_{21}$ ) of this filter sample

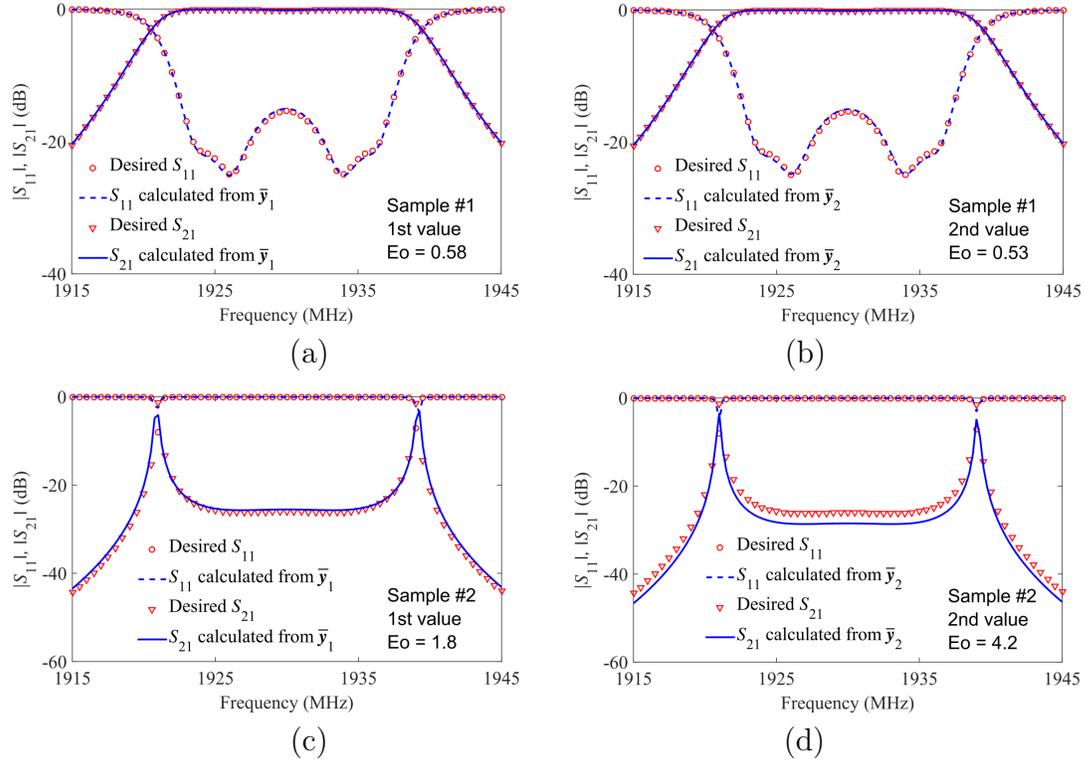


Figure 4.6: Two test samples for the fourth-order filter example. Sample #1: (a) The  $S$ -parameters calculated from the 1st value  $\bar{\mathbf{y}}_1$  provided by our proposed inverse model, (b) The  $S$ -parameters calculated from the 2nd value  $\bar{\mathbf{y}}_2$  provided by our proposed inverse model. Sample #2: (c) The  $S$ -parameters calculated from the 1st value  $\bar{\mathbf{y}}_1$  provided by our proposed inverse model, (d) The  $S$ -parameters calculated from the 2nd  $\bar{\mathbf{y}}_2$  provided by our proposed inverse model. For Sample #1,  $S$ -parameters calculated from both values match the desired  $S$ -parameters very well. Both values are valid. For Sample #2, only the  $S$ -parameter calculated from the 1st value matches the desired  $S$ -parameters very well. Therefore, the 1st value is valid, and the 2nd value is dummy. Our proposed valid value selection process can distinguish valid values from the dummy ones. Our proposed inverse model with valid value selection process can always provide accurate solutions of coupling parameters for a given set of  $S$ -parameter input.

are shown in Fig. 4.6(a). After the  $S$ -parameters are fed into our proposed inverse model, two values of coupling parameters provided by our inverse model are

$$\bar{\mathbf{y}}_1 = [0.8888 \ 0.6587 \ 0.9502]^T \quad (4.17a)$$

$$\bar{\mathbf{y}}_2 = [0.9497 \ 0.6575 \ 0.8878]^T \quad (4.17b)$$

The  $S$ -parameters calculated from  $\bar{\mathbf{y}}_1$  and  $\bar{\mathbf{y}}_2$  are shown in Fig. 4.6(a) and Fig. 4.6(b), respectively. The  $S$ -parameters calculated from both values match the desired  $S$ -parameters very well. The errors  $E_o$  for both values are

$$E_o(\bar{\mathbf{y}}_1) = 0.58 \quad (4.18a)$$

$$E_o(\bar{\mathbf{y}}_2) = 0.53 \quad (4.18b)$$

For given  $\varepsilon = 2$ , both  $\bar{\mathbf{y}}_1$  and  $\bar{\mathbf{y}}_2$  are selected as the valid values. In this situation, it proves that for a given set of  $S$ -parameter, there would be multivalued solutions of coupling parameters. Our proposed multivalued neural network can associate a single set of  $S$ -parameter to multiple sets of coupling parameters. By combining our proposed inverse model with valid value selection process, we can always obtain reliable solutions of the coupling parameters for a given set of  $S$ -parameters.

For Sample #2, the desired  $S$ -parameters (both  $S_{11}$  and  $S_{21}$ ) of this filter sample are shown in Fig. 4.6(c). After the  $S$ -parameters are fed into our proposed inverse

model, two values of coupling parameters provided by our inverse model are

$$\bar{\mathbf{y}}_1 = [0.1571 \ 1.2021 \ 0.2176]^T \quad (4.19a)$$

$$\bar{\mathbf{y}}_2 = [0.2218 \ 1.1934 \ 0.1088]^T \quad (4.19b)$$

The  $S$ -parameters calculated from  $\bar{\mathbf{y}}_1$  and  $\bar{\mathbf{y}}_2$  are shown in Fig. 4.6(c) and Fig. 4.6(d), respectively. Only the  $S$ -parameter calculated from  $\bar{\mathbf{y}}_1$  matches the desired  $S$ -parameters very well. The errors  $E_o$  for both values are

$$E_o(\bar{\mathbf{y}}_1) = 1.8 \quad (4.20a)$$

$$E_o(\bar{\mathbf{y}}_2) = 4.2 \quad (4.20b)$$

For given  $\varepsilon = 2$ ,  $\bar{\mathbf{y}}_1$  is selected as the valid value, and  $\bar{\mathbf{y}}_2$  is dummy. Therefore, our proposed valid value selection process can distinguish the valid value(s) from the dummy value(s). By combining our proposed inverse model with valid value selection process, we can always obtain reliable solutions of the coupling parameters for a given set of  $S$ -parameter.

Next we further demonstrate the use of our proposed multivalued neural inverse model to extract the coupling parameters for these two filter samples.

For Sample #1, this filter sample is slightly detuned from the nominal filter shown in (4.16). Two valid values provided by our proposed multivalued neural network inverse model are in (4.17). From (4.18), we have already known that both values provided by our inverse model are valid. To further distinguish the true solution between these two values to get a unique set of desired coupling parameters,

the 2-circuit ANN process described in Section II-E using our proposed multivalued neural network inverse model is applied.

This slightly detuned filter is used as the 1st circuit in the 2-circuit ANN extraction process using our proposed multivalued neural network inverse model. The desired coupling parameters of the 1st circuit are

$$M_{Desired}^1 = \begin{bmatrix} 0 & 0.8850 & 0 & 0 \\ 0.8850 & 0 & 0.6595 & 0 \\ 0 & 0.6595 & 0 & 0.9493 \\ 0 & 0 & 0.9493 & 0 \end{bmatrix} \quad (4.21)$$

Then we change one filter parameter, which mainly affects the first coupling parameter, i.e.,  $M_{12}$ , to obtain the 2nd circuit. The desired coupling matrix of the 2nd circuit is

$$M_{Desired}^2 = \begin{bmatrix} 0 & \underline{0.5634} & 0 & 0 \\ \underline{0.5634} & 0 & 0.6612 & 0 \\ 0 & 0.6612 & 0 & 0.9131 \\ 0 & 0 & 0.9131 & 0 \end{bmatrix} \quad (4.22)$$

We can obtain the  $S$ -parameters (both  $S_{11}$  and  $S_{21}$ ) of the 2nd circuit. After the  $S$ -parameters are fed into our proposed inverse model, the model produces two values

of coupling parameters for the 2nd circuit as follows

$$\bar{\mathbf{y}}_1 = [0.5601 \ 0.6632 \ 0.9127]^T \quad (4.23a)$$

$$\bar{\mathbf{y}}_2 = [0.9118 \ 0.6638 \ 0.5591]^T \quad (4.23b)$$

We calculate  $E_o$  (the errors between the  $S$ -parameters calculated from these values and the desired  $S$ -parameters) for both values. According to  $E_o$ , both values for the 2nd circuit are selected as valid values. Then we need to compare each value in (4.19) with each value in (4.23). According to the adjustment of the filter parameter, the change of the first coupling parameter should be large, and others small. We can find the correct combination as follows

$$\text{1st circuit: } \bar{\mathbf{y}}_1 = [0.8888 \ 0.6587 \ 0.9502]^T \quad (4.24a)$$

$$\text{2st circuit: } \bar{\mathbf{y}}_1 = [0.5601 \ 0.6632 \ 0.9127]^T \quad (4.24b)$$

which satisfies the pattern of parameter change between the two circuits. Therefore, the extracted coupling parameters for this slightly detuned filter sample are

$$M_{Extracted} = \begin{bmatrix} 0 & 0.8888 & 0 & 0 \\ 0.8888 & 0 & 0.6587 & 0 \\ 0 & 0.6587 & 0 & 0.9502 \\ 0 & 0 & 0.9502 & 0 \end{bmatrix} \quad (4.25)$$

Comparing the extracted coupling parameters in (4.25) with the desired coupling parameters in (4.21), we can find that our proposed multivalued neural network inverse model can be used to extract the coupling parameter accurately. The total

computation effort in this process includes two evaluations of the proposed inverse model, four evaluations of the forward model, and comparisons between the solutions of 2 circuits. Since the evaluation of ANN is very fast and the comparison time between vectors is very short, the multi-circuit ANN process using our proposed multivalued neural network is very efficient and can be done instantaneously.

For Sample #2, this filter sample is highly detuned from the nominal filter shown in (4.16). The desired coupling parameters of this highly detuned filter sample are

$$M_{Desired} = \begin{bmatrix} 0 & 0.1512 & 0 & 0 \\ 0.1512 & 0 & 1.1856 & 0 \\ 0 & 1.1856 & 0 & 0.2099 \\ 0 & 0 & 0.2099 & 0 \end{bmatrix} \quad (4.26)$$

From (4.20), we have known that the only valid value is  $\bar{\mathbf{y}}_1 = [0.1571 \ 1.2021 \ 0.2176]^T$ .

In this situation, the extracted coupling matrix is

$$M_{Extracted} = \begin{bmatrix} 0 & 0.1571 & 0 & 0 \\ 0.1571 & 0 & 1.2021 & 0 \\ 0 & 1.2021 & 0 & 0.2176 \\ 0 & 0 & 0.2176 & 0 \end{bmatrix} \quad (4.27)$$

Comparing the extracted coupling parameters in (4.27) and the desired coupling parameters in (4.26), our proposed inverse model can provide a reliable extraction result.

For comparison purposes, we also use the existing 2-circuit optimization approach in [93] to extract the coupling parameters in both cases mentioned above. The starting points of the 2-circuit optimization approach in both cases are the nominal coupling parameters in (4.16). For the slightly detuned filter in Sample #1, the extraction result obtained by the existing 2-circuit optimization method is also very accurate. However, the conventional 2-circuit optimization method takes about 90 gradient based optimization iterations to find the solution. For the highly detuned filter in Sample #2, the optimization converges to a wrong solution, since the starting point of the optimization is far from the desired coupling parameters. The optimization solution is found to be wrong, because the optimized values of coupling parameters are outside the practical range even though the corresponding  $S$ -parameters have approximated the desired data. The multi-circuit ANN process using our proposed multivalued neural network inverse model does not suffer from the problem of a bad starting point, because our process does not need iterative model evaluation. Therefore, our proposed multivalued neural network inverse model is more efficient for filter parameter extraction over the conventional multi-circuit optimization approach.

### 4.3.3 The Inverse Modeling of A Sixth-Order Multicoupled Cavity Filter

In this example, we deal with a sixth-order multicoupled cavity filter used in microwave communication system [92]. This filter is centered at 4 GHz with 40 MHz

bandwidth. The coupling matrix is shown as follows

$$M = \begin{bmatrix} 0 & M_{12} & 0 & 0 & 0 & M_{16} \\ M_{21} & 0 & M_{23} & 0 & M_{25} & 0 \\ 0 & M_{32} & 0 & M_{34} & 0 & 0 \\ 0 & 0 & M_{43} & 0 & M_{45} & 0 \\ 0 & M_{52} & 0 & M_{54} & 0 & M_{56} \\ M_{61} & 0 & 0 & 0 & M_{65} & 0 \end{bmatrix} \quad (4.28)$$

where  $M_{ij} = M_{ji}$ ,  $i \neq j$ . There are 7 non-zero coupling parameters, i.e.,  $M_{12}$ ,  $M_{23}$ ,  $M_{34}$ ,  $M_{45}$ ,  $M_{56}$ ,  $M_{16}$  and  $M_{25}$ . The nominal coupling parameters are shown in Table 4.3.

For this example, we try to build an inverse model between the coupling parameters and the  $S$ -parameter. Here we choose  $S_{11}$  (in dB) at 33 frequency samples as the inputs  $\bar{\mathbf{x}}$ , and 7 coupling parameters as the outputs, i.e.,  $\bar{\mathbf{y}} = [M_{12} \ M_{23} \ M_{34} \ M_{45} \ M_{56} \ M_{16} \ M_{25}]^T$ . Therefore,  $n = 33, m = 7$ . To generate the training and test data, we choose a tolerance of  $\pm 0.2$  for every coupling parameter, which is shown in Table 4.3. Then we use uniformly distributed random values in this range to get 100000 sets of input-output data as the training data, and 50000 sets as test data.

We firstly illustrate the inverse modeling problem using the conventional direct inverse modeling technique. We train a 4-layer MLP with 60 neurons in each hidden layer, 33 input neurons, and 7 output neurons, i.e., MLP: 33-60-60-7, to learn all

the training data. After training with enough iterations, the training error and test error are shown in Table 4.4. From the table, we can see that even though we try to train the conventional direct inverse model with enough hidden neurons and training iterations, the training error and test error are always large because of the non-uniqueness problem.

Then we use our proposed multivalued neural network inverse modeling technique with 2, 3 and 4 values to solve this example. We will also use 4-layer MLPs in our proposed technique. The number of input neurons remains 33. The same number of neurons (i.e., 60) in each hidden layer are used. The total numbers of output neurons of the proposed multivalued neural network inverse models with 2, 3, and 4 values are 14 ( $2 \cdot m$ ), 21 ( $3 \cdot m$ ), and 28 ( $4 \cdot m$ ), respectively. The comparison results are shown in Table 4.4. With the increase of the number of the values, both the training error and test error are reduced using our proposed inverse modeling technique. The proposed multivalued neural network inverse model with 4 values is the most accurate one for this sixth-order filter example.

Next, we use the proposed multivalued neural network inverse model (with 4 values) to extract the coupling parameters for two filter cases, i.e., slightly (Case 1) and highly (Case 2) detuned from the nominal filter.

In the first case, our proposed multivalued neural network inverse model is used to extract coupling parameters for a slightly detuned filter sample. The 2-circuit ANN process is applied in this case. This filter is used as the 1st circuit in the extraction process. The desired coupling parameters of the 1st circuit are

Table 4.3: Definition of the Training and Test Data Range for the Sixth-Order Cavity Filter Example

|          | Min       | Nominal   | Max       |
|----------|-----------|-----------|-----------|
| $M_{12}$ | 0.659956  | 0.859956  | 1.059956  |
| $M_{23}$ | 0.326602  | 0.526602  | 0.726602  |
| $M_{34}$ | 0.591894  | 0.791894  | 0.991894  |
| $M_{45}$ | 0.326602  | 0.526602  | 0.726602  |
| $M_{56}$ | 0.659956  | 0.859956  | 1.059956  |
| $M_{16}$ | -0.112707 | 0.087293  | 0.287293  |
| $M_{25}$ | -0.593686 | -0.393686 | -0.193686 |

Table 4.4: The Comparison of Modeling Results Using Different Inverse Modeling Methods for the Sixth-Order Cavity Filter

| Inverse modeling methods          | *Training error | Test error |
|-----------------------------------|-----------------|------------|
| Direct inverse modeling technique | 5.64%           | 5.91%      |
| Proposed technique with 2 values  | 2.67%           | 3.05%      |
| Proposed technique with 3 values  | 2.10%           | 2.66%      |
| Proposed technique with 4 values  | 1.86%           | 1.94%      |

\*To be compatible with the errors in the direct inverse modeling technique, both the training error and test error in the proposed modeling technique are calculated by (4.9).

$$M_{Desired}^1 = \begin{bmatrix} 0 & .9351 & 0 & 0 & 0 & .0734 \\ .9351 & 0 & .5631 & 0 & -.4037 & 0 \\ 0 & .5631 & 0 & .8032 & 0 & 0 \\ 0 & 0 & .8032 & 0 & .5309 & 0 \\ 0 & -.4037 & 0 & .5309 & 0 & .8503 \\ .0734 & 0 & 0 & 0 & .8503 & 0 \end{bmatrix} \quad (4.29)$$

The desired  $S$ -parameter is shown in Fig. 4.7(a). After the desired  $S$ -parameter is fed into our proposed inverse model, four values provided by our inverse model are

$$\bar{\mathbf{y}}_1 = [.9105 \ .5721 \ .7984 \ .5036 \ .8584 \ .0755 \ - \ .4112]^T \quad (4.30a)$$

$$\bar{\mathbf{y}}_2 = [.8329 \ .5038 \ .8024 \ .5581 \ .9373 \ .0714 \ - \ .4127]^T \quad (4.30b)$$

$$\bar{\mathbf{y}}_3 = [.8688 \ .5003 \ .8025 \ .5588 \ .8964 \ .0738 \ - \ .4173]^T \quad (4.30c)$$

$$\bar{\mathbf{y}}_4 = [.9425 \ .5631 \ .8011 \ .4995 \ .8308 \ .0756 \ - \ .4174]^T \quad (4.30d)$$

The  $S$ -parameters calculated from these values  $\bar{\mathbf{y}}_1$ ,  $\bar{\mathbf{y}}_2$ ,  $\bar{\mathbf{y}}_3$ , and  $\bar{\mathbf{y}}_4$  are shown in Fig. 4.7(a), Fig. 4.7(b), Fig. 4.7(c), and Fig. 4.7(d), respectively. The  $S$ -parameters calculated from  $\bar{\mathbf{y}}_2$  and  $\bar{\mathbf{y}}_4$  match the desired  $S$ -parameter very well. The errors  $E_o$  for all these four values are

$$E_o(\bar{\mathbf{y}}_1) = 7.1 \quad (4.31a)$$

$$E_o(\bar{\mathbf{y}}_2) = 1.5 \quad (4.31b)$$

$$E_o(\bar{\mathbf{y}}_3) = 8.7 \quad (4.31c)$$

$$E_o(\bar{\mathbf{y}}_4) = 1.9 \quad (4.31d)$$

For given  $\varepsilon = 2$ , both  $\bar{\mathbf{y}}_2$  and  $\bar{\mathbf{y}}_4$  are selected as the valid values.

To further distinguish between these two valid solutions to get a unique set of desired coupling parameters, the 2-circuit ANN process described in Section II-E using our proposed multivalued neural network inverse model is applied. We change one filter parameter, which mainly affects the second coupling parameter in  $\bar{\mathbf{y}}$ , i.e.,  $M_{23}$ , to obtain the 2nd circuit. The desired coupling matrix of the 2nd circuit is

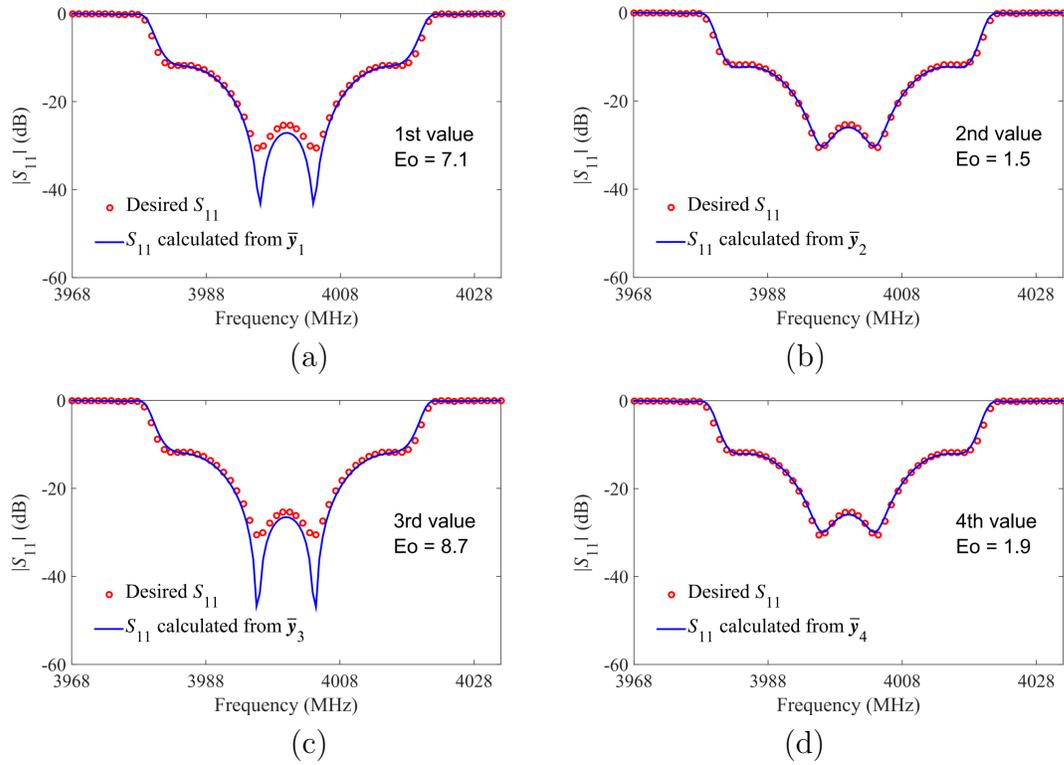


Figure 4.7: The slightly detuned filter case for the sixth-order cavity filter example. Our proposed inverse model can provide 4 values for the coupling parameters: (a)  $S_{11}$  calculated from the 1st value  $\bar{y}_1$  (-); (b)  $S_{11}$  calculated from the 2nd value  $\bar{y}_2$  (-); (c)  $S_{11}$  calculated from the 3rd value  $\bar{y}_3$  (-). (d)  $S_{11}$  calculated from the 4th value  $\bar{y}_4$  (-). The desired  $S$ -parameter is shown in circles (o).  $S$ -parameters calculated from  $\bar{y}_2$  and  $\bar{y}_4$  match the desired  $S$ -parameter very well. They are selected as valid values. The other two are dummy. Our proposed valid value selection process can distinguish valid values from the dummy ones. Our proposed inverse model with valid value selection process can always provide accurate solutions of coupling parameters for a given set of  $S$ -parameter.

$$M_{Desired}^2 = \begin{bmatrix} 0 & .9367 & 0 & 0 & 0 & .0739 \\ .9367 & 0 & \underline{.3622} & 0 & -.4053 & 0 \\ 0 & \underline{.3622} & 0 & .8012 & 0 & 0 \\ 0 & 0 & .8012 & 0 & .5311 & 0 \\ 0 & -.4053 & 0 & .5311 & 0 & .8507 \\ .0739 & 0 & 0 & 0 & .8507 & 0 \end{bmatrix} \quad (4.32)$$

We can get the  $S$ -parameter for the 2nd circuit. After the  $S$ -parameter of the 2nd circuit is fed into our proposed inverse model, the model produces four values of coupling parameters for the 2nd circuit as follows

$$\bar{\mathbf{y}}_1 = [.8449 \ .5257 \ .7994 \ .3581 \ .9209 \ .0625 \ -.4039]^T \quad (4.33a)$$

$$\bar{\mathbf{y}}_2 = [.8309 \ .4826 \ .7878 \ .4016 \ .9556 \ .0647 \ -.4056]^T \quad (4.33b)$$

$$\bar{\mathbf{y}}_3 = [.9279 \ .3562 \ .8044 \ .5211 \ .8390 \ .0600 \ -.4097]^T \quad (4.33c)$$

$$\bar{\mathbf{y}}_4 = [.9568 \ .3994 \ .7893 \ .4829 \ .8315 \ .0681 \ -.4095]^T \quad (4.33d)$$

We calculate  $E_o$  (the errors between the  $S$ -parameters calculated from these values and the desired  $S$ -parameter) for all these four values. According to  $E_o$ ,  $\bar{\mathbf{y}}_1$  and  $\bar{\mathbf{y}}_3$  are selected as the valid values. Then we compare  $\bar{\mathbf{y}}_2$ ,  $\bar{\mathbf{y}}_4$  of the 1st circuit in (4.30) with  $\bar{\mathbf{y}}_1$ ,  $\bar{\mathbf{y}}_3$  of the 2nd circuit in (4.33). From the comparison, we find that the difference between  $\bar{\mathbf{y}}_4$  of the 1st circuit and  $\bar{\mathbf{y}}_3$  of the 2nd circuit matches the pattern of parameter change between the two circuits. According to the change pattern of the circuit parameter, the change of second coupling parameter should be large, while the changes of other coupling parameters are small, as demonstrated below

$$\text{1st circuit: } \bar{\mathbf{y}}_4 = [.9425 \ .5631 \ .8011 \ .4995 \ .8308 \ .0756 \ - .4174]^T \quad (4.34a)$$

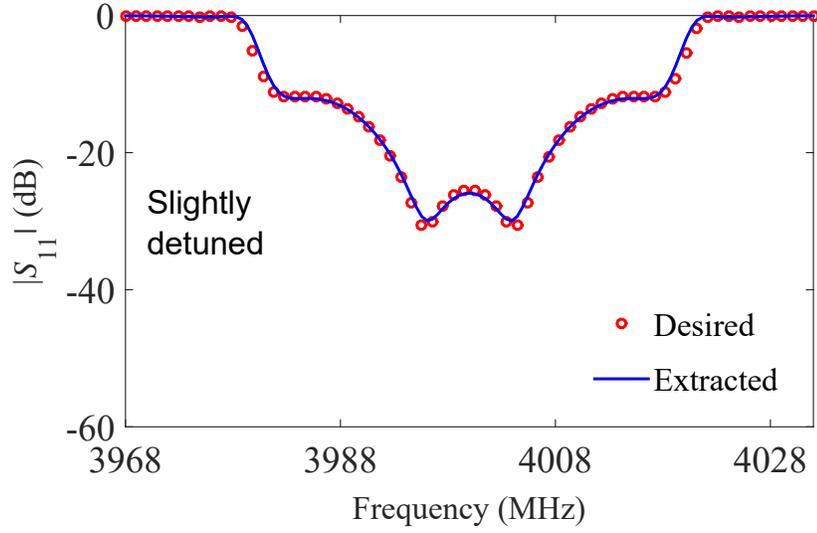
$$\text{2nd circuit: } \bar{\mathbf{y}}_3 = [.9279 \ .3562 \ .8044 \ .5211 \ .8390 \ .0600 \ - .4097]^T \quad (4.34b)$$

Therefore, the extracted coupling parameters of this slightly detuned filter are

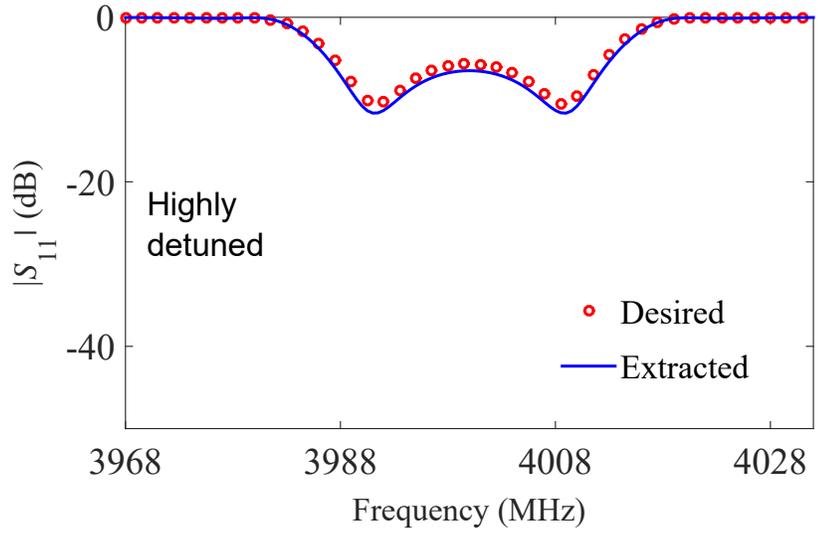
$$M_{Extracted} = \begin{bmatrix} 0 & .9425 & 0 & 0 & 0 & .0756 \\ .9425 & 0 & .5631 & 0 & -.4174 & 0 \\ 0 & .5631 & 0 & .8011 & 0 & 0 \\ 0 & 0 & .8011 & 0 & .4995 & 0 \\ 0 & -.4174 & 0 & .4995 & 0 & .8308 \\ .0756 & 0 & 0 & 0 & .8308 & 0 \end{bmatrix} \quad (4.35)$$

Comparing the extracted coupling parameters in (4.35) with the desired coupling parameters in (4.29), we can find that the 2-circuit ANN process using our proposed multivalued neural network inverse model is reliable. The total computation effort in this process includes two evaluations of the proposed inverse model, eight evaluations of the forward model, and comparisons between the solutions of 2 circuits. The multi-circuit ANN process using our proposed multivalued neural network is very efficient and can be done instantaneously.

In the second case, our proposed multivalued neural network inverse model is used to extract coupling parameters for a highly detuned filter sample. The similar procedure as that in the first case is done. Both the desired and extracted coupling parameters for this filter sample are shown below



(a)



(b)

Figure 4.8: The slightly and highly detuned filter cases for the sixth-order cavity filter example. (a) The slightly detuned filter case. (b) The highly detuned filter case. The circles (o) represent the desired  $S$ -parameters. The lines (-) represent the  $S$ -parameters calculated by the extracted coupling parameters, which are obtained using our proposed multivalued neural network inverse model. The  $S$ -parameters calculated from the extracted coupling parameters in both cases can match the desired  $S$ -parameters very well.

$$M_{Desired} = \begin{bmatrix} 0 & 1.057 & 0 & 0 & 0 & .0501 \\ 1.057 & 0 & .3873 & 0 & -.2171 & 0 \\ 0 & .3873 & 0 & .6005 & 0 & 0 \\ 0 & 0 & .6005 & 0 & .6209 & 0 \\ 0 & -.2171 & 0 & .6209 & 0 & 1.057 \\ .0501 & 0 & 0 & 0 & 1.057 & 0 \end{bmatrix} \quad (4.36a)$$

$$M_{Extracted} = \begin{bmatrix} 0 & 1.046 & 0 & 0 & 0 & .0643 \\ 1.046 & 0 & .4018 & 0 & -.2453 & 0 \\ 0 & .4018 & 0 & .6124 & 0 & 0 \\ 0 & 0 & .6124 & 0 & .5848 & 0 \\ 0 & -.2453 & 0 & .5848 & 0 & 1.009 \\ .0643 & 0 & 0 & 0 & 1.009 & 0 \end{bmatrix} \quad (4.36b)$$

As we can see, the coupling parameters extracted by the multi-circuit ANN process using our proposed multivalued neural network inverse model match the desired coupling parameters very well. The  $S$ -parameters calculated from the extracted coupling parameters in both cases can also match the desired  $S$ -parameters very well, which is shown in Fig. 4.8.

For comparison purposes, we also use the existing 2-circuit optimization approach in [93] to extract the coupling parameters for both cases mentioned above. In the first case, the result obtained by the existing 2-circuit optimization approach is also very accurate. However, the conventional 2-circuit optimization method takes about 250 gradient based optimization iterations to find the solution. In the second case, the optimization converges to a wrong solution, since the starting point is far from the desired coupling parameters. The multi-circuit ANN process using our proposed multivalued neural network inverse model does not suffer from the problem of a bad starting point. Therefore, our proposed multivalued neural network

inverse model is more efficient for filter parameter extraction over the conventional multi-circuit optimization approach.

#### 4.3.4 The Inverse Modeling of An Eighth-Order Dual-Band Filter

In this example, we deal with an eighth-order dual-band filter [2]. Two passbands for this filter are from 3.901 GHz to 3.960 GHz and from 4.040 GHz to 4.101 GHz.

The coupling matrix is shown as follows

$$M = \begin{bmatrix} 0 & M_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ M_{21} & 0 & M_{23} & 0 & 0 & 0 & M_{27} & 0 \\ 0 & M_{32} & 0 & M_{34} & 0 & M_{36} & 0 & 0 \\ 0 & 0 & M_{43} & 0 & M_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{54} & 0 & M_{56} & 0 & 0 \\ 0 & 0 & M_{63} & 0 & M_{65} & 0 & M_{67} & 0 \\ 0 & M_{72} & 0 & 0 & 0 & M_{76} & 0 & M_{78} \\ 0 & 0 & 0 & 0 & 0 & 0 & M_{87} & 0 \end{bmatrix} \quad (4.37)$$

where  $M_{ij} = M_{ji}$ ,  $i \neq j$ . There are 9 non-zero coupling parameters, i.e.,  $M_{12}$ ,  $M_{23}$ ,  $M_{34}$ ,  $M_{45}$ ,  $M_{56}$ ,  $M_{67}$ ,  $M_{78}$ ,  $M_{27}$  and  $M_{36}$ . The nominal coupling parameters are shown in Table 4.5.

For this example, we try to build an inverse model between the coupling parameters and the  $S$ -parameter. Here we choose  $S_{11}$  (in dB) at 51 frequency samples as

the inputs  $\bar{\mathbf{x}}$ , and 9 coupling parameters as the outputs, i.e.,  $\bar{\mathbf{y}} = [M_{12} \ M_{23} \ M_{34} \ M_{45} \ M_{56} \ M_{67} \ M_{78} \ M_{27} \ M_{36}]^T$ . Therefore,  $n = 51, m = 9$ . To generate the training and test data, we choose a tolerance of  $\pm 0.15$  for every coupling parameter, which is shown in Table 4.5. Then we use uniformly distributed random values in this range to get 200000 sets of input-output data as the training data, and 100000 sets as test data.

First of all, we illustrate the inverse modeling problem using the conventional direct inverse modeling technique. We train a 4-layer MLP with 51 input neurons, 75 neurons in each hidden layer, and 9 output neurons, i.e., MLP: 51-75-75-9, to learn all the training data. The training error and test error are shown in Table 4.6. From the table, we can see the accuracy of the conventional inverse model is low. The training error and the test error can not be reduced because of the non-uniqueness problem.

Then we use our proposed multivalued neural network inverse modeling technique with 2, 3, 4, 5 and 6 values to solve this example. The same numbers of hidden layers (i.e., 2) and hidden neurons (75 in each layer) as that used in direct inverse modeling technique are used in our proposed multivalued neural network. The numbers of total output neurons of the proposed multivalued neural networks with 2, 3, 4, 5 and 6 values are 18 ( $2 \cdot m$ ), 27 ( $3 \cdot m$ ), 36 ( $4 \cdot m$ ), 45 ( $5 \cdot m$ ) and 54 ( $6 \cdot m$ ), respectively. The comparison results are shown in Table 4.6. For the modeling results in Table 4.6, we have tried with MLP structures with different number of hidden neurons and different initial weights to achieve the results. During train-

Table 4.5: Definition of the Training and Test Data Range for the Eighth-Order Dual-Band Filter Example

|          | Min     | Nominal | Max     |
|----------|---------|---------|---------|
| $M_{12}$ | 0.6967  | 0.8467  | 0.9967  |
| $M_{23}$ | 0.2642  | 0.4142  | 0.5642  |
| $M_{34}$ | 0.1254  | 0.2754  | 0.4254  |
| $M_{45}$ | 0.1332  | 0.2832  | 0.4332  |
| $M_{56}$ | 0.1254  | 0.2754  | 0.4254  |
| $M_{67}$ | 0.2642  | 0.4142  | 0.5642  |
| $M_{78}$ | 0.6967  | 0.8467  | 0.9967  |
| $M_{27}$ | -0.4400 | -0.2900 | -0.1400 |
| $M_{36}$ | -0.5670 | -0.4170 | -0.2670 |

ing, as soon as the model reaches the specified accuracy threshold, the training will stop. The training result is not necessarily required to be a global optimum. With the increase of the number of the values, both the training error and test error are reduced using our proposed inverse modeling technique. From Table 4.6, we can find that the proposed multivalued neural network inverse model with 6 values is the most accurate one for this eighth-order dual-band filter example. Our proposed technique can automatically learn all the training data by redirecting the contradictory information into multiple values of our proposed multivalued neural network.

Next, we use our proposed multivalued neural network inverse model to extract the coupling parameters for two filter cases, i.e., slightly (Case 1) and highly (Case 2)

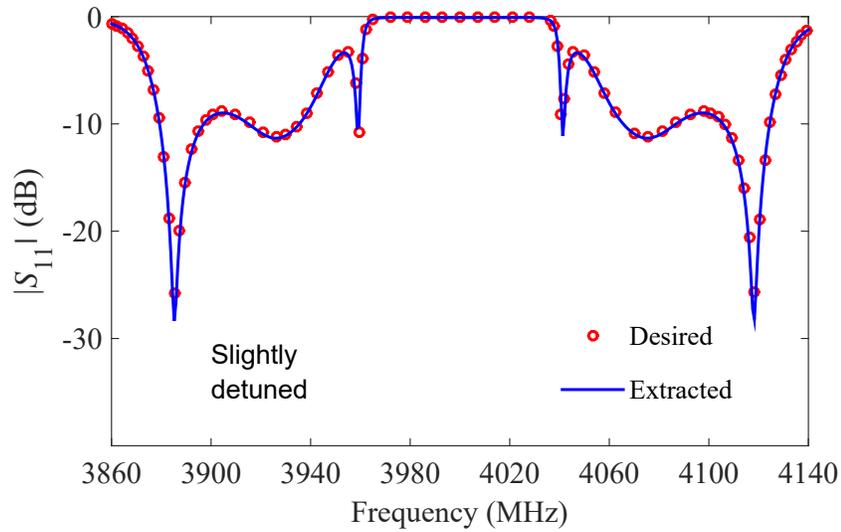
Table 4.6: The Comparison of Modeling Results Using Different Inverse Modeling Methods for the Eighth-Order Dual-Band Filter

| Inverse modeling methods          | *Training error | Test error |
|-----------------------------------|-----------------|------------|
| Direct inverse modeling technique | 5.09%           | 5.20%      |
| Proposed technique with 2 values  | 3.28%           | 3.89%      |
| Proposed technique with 3 values  | 2.89%           | 3.11%      |
| Proposed technique with 4 values  | 2.54%           | 2.67%      |
| Proposed technique with 5 values  | 2.27%           | 2.31%      |
| Proposed technique with 6 values  | 1.94%           | 2.00%      |

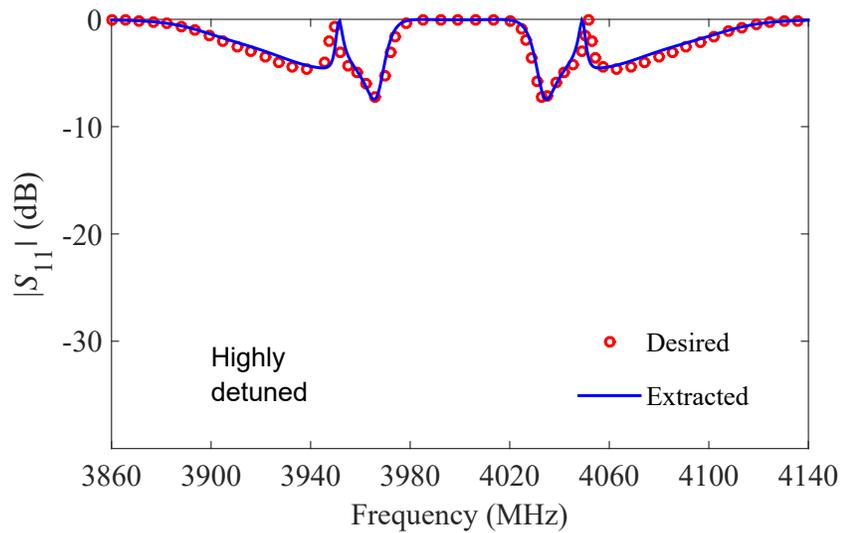
\*To be compatible with the errors in the direct inverse modeling technique, both the training error and test error in the proposed modeling technique are calculated by (4.9).

detuned from the nominal filter shown in Table 4.5. The multi-circuit ANN process is applied in both cases. The extracted coupling parameters are shown in Table 4.7. From the table, we can see that the extracted coupling parameters using proposed multivalued neural network inverse model match the desired coupling parameters very well. The  $S$ -parameters calculated from the extracted coupling parameters in the first and second cases are shown in Fig. 4.9 (a) and (b), respectively. From the figures, we can find that  $S$ -parameters calculated from the extracted coupling parameters match the desired  $S$ -parameters very well.

For comparison purposes, we apply the existing multi-circuit optimization approach in [93] to both cases mentioned above. In the first case, the conventional multi-circuit optimization process can also get an accurate result. However, it takes 26 quasi-Newton iterations, which includes about 150 forward model evaluations. In



(a)



(b)

Figure 4.9: Two cases for the eighth-order dual-band filter example: (a) The slightly detuned filter case. (b) The highly detuned filter case. The circles (o) represent the desired  $S$ -parameters. The lines (-) represent the  $S$ -parameters calculated by the extracted coupling parameters, which are obtained by the multi-circuit ANN process using our proposed multivalued neural network inverse model.

Table 4.7: Comparison Between the Desired and Extracted Coupling Parameters for the Eighth-Order Dual-Band Filter Example

| Case     | Slightly detuned |           | Highly detuned |           |
|----------|------------------|-----------|----------------|-----------|
|          | Desired          | Extracted | Desired        | Extracted |
| $M_{12}$ | 0.9347           | 0.9382    | 0.6998         | 0.6878    |
| $M_{23}$ | 0.4029           | 0.4049    | 0.2734         | 0.2578    |
| $M_{34}$ | 0.3026           | 0.3085    | 0.1491         | 0.1533    |
| $M_{45}$ | 0.3100           | 0.3083    | 0.4246         | 0.4021    |
| $M_{56}$ | 0.2962           | 0.3026    | 0.3023         | 0.2913    |
| $M_{67}$ | 0.5090           | 0.5096    | 0.3039         | 0.3048    |
| $M_{78}$ | 0.8985           | 0.8959    | 0.9852         | 1.0038    |
| $M_{27}$ | -0.3369          | -0.3349   | -0.2455        | -0.2431   |
| $M_{36}$ | -0.5639          | -0.5606   | -0.2874        | -0.2978   |

the second case, the optimization converges to a wrong solution, since the starting point is far from the desired coupling parameters. The multi-circuit ANN process using our proposed multivalued neural network inverse model does not suffer from the problem of a bad starting point. Therefore, our proposed multivalued neural network inverse model is more efficient for filter parameter extraction over the conventional multi-circuit optimization approach.

## 4.4 Conclusion

This chapter has presented a multivalued neural network inverse modeling technique to address a major challenge in inverse modeling, which is the non-uniqueness problem. The proposed technique can associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. Our proposed inverse model is structured to accommodate multiple values for the model output. Our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. As illustrated in the examples, our proposed technique can solve the non-uniqueness problem in a simpler and more automated way, compared to the existing ANN inverse modeling techniques. Furthermore, our proposed multivalued neural network inverse models have been used to enhance the efficiency and the reliability of the filter parameter extraction process over the conventional multi-circuit optimization approach.

# Chapter 5

## Conclusions and Future Research

### 5.1 Conclusions

In this thesis, we have investigated advanced computer-aided-design (CAD) techniques for optimization and inverse modeling of microwave filters. Aimed at speeding up design optimization and improving the accuracy of inverse modeling of microwave filters, we have proposed a cognition-driven formulation of space mapping optimization of microwave filters and a multivalued neural network inverse modeling technique.

In our proposed cognition-driven formulation of space mapping optimization, two sets of intermediate feature space parameters, including the feature frequency and ripple height parameters, have been used to build two kinds of mapping. To control the optimization updates and ensure convergence of the proposed cognition-drive optimization, a trust region approach has been incorporated into our technique. By using the proposed cognition-driven formulation of optimization directly in the

feature space, our method can increase optimization efficiency and the ability to avoid being trapped in a local minimum over our baseline approaches of coarse and fine mesh EM space mapping and direct EM optimization.

In our proposed multivalued neural network inverse modeling technique, the proposed inverse model can associate a single set of electrical parameters with multiple sets of geometrical or physical parameters. Our proposed inverse model has been structured to accommodate multiple values for the model output. Our proposed multivalued neural network can learn all the training data by automatically redirecting contradictory information into different values of the proposed inverse model. As illustrated in the examples, our proposed technique can solve the non-uniqueness (multivalued) problem in a simpler and more automated way, compared to the existing ANN inverse modeling techniques. Furthermore, our proposed multivalued neural network inverse models have been applied to enhance the efficiency and the reliability of the filter parameter extraction process over the conventional multi-circuit optimization approach.

Both proposed techniques in this thesis have been illustrated by several microwave filter examples. Our proposed techniques have been used to improve the efficiency and accuracy of the design and parameter extraction processes of microwave filters.

## 5.2 Future Research

Initiated in this thesis towards addressing optimization and inverse modeling challenges of microwave filters, some interesting future directions are stated as follows

- The cognition-driven formulation of space mapping has been studied for equal ripple optimization of microwave filters in this thesis. Optimization of many other microwave structures, such as antennas, currently still involves high cost EM simulations. One future direction is to expand the cognition-driven concept to advance EM based design not only for filters, but also for more general microwave circuits, such as antennas. The radiation pattern of antennas can be used as feature parameters. More formulation need to be studied to realize an accurate mapping from radiation pattern to design variables. Thus the radiation pattern can be optimized by changing design variables using the newly formulated mapping.
- Yield optimization is an important topic for the fabrication process of microwave filters. Manufactured outcomes are spread over a region around the nominal design. In order to improve the yield, the statistical design variables should be taken into consideration. We can further expand the cognition-driven formulation of space mapping to the yield driven optimization or the statistical analysis of microwave filters. Using our proposed cognition-driven space mapping, we can find the statistic behavior of the feature parameters. By optimizing the statistical design variables, we can improve the margin of

the feature parameters. Therefore, the yield can be improved.

- The proposed multivalued neural network has been applied to parameter extraction of filters. It is possible to further apply our technique to design optimization. We need to investigate how to build an accurate inverse model between  $S$ -parameters and design variables. New formulations need to be studied to obtain the desired  $S$ -parameters from given design specifications. The optimal design variables can be obtained at the inverse model output with the desired  $S$ -parameters as the inverse model inputs.
- Another possible future direction is to combine both cognition-driven concept and multivalued neural network inverse modeling technique for design optimization. It is possible to build a mapping from feature parameters to design variables using an inverse model. The feature parameters can be used as the inputs of the inverse model, and the design variables can be used as the outputs of the inverse model. The multivalued neural network inverse modeling technique can solve the multivalued problem when the inverse model between feature parameters and design variables is trained. New formulations need to be studied to get the desired feature parameters from given design specifications. It is possible to obtain the optimal design variables instantly at the inverse model output with the desired feature parameters as inputs. Therefore, we can probably further improve the optimization efficiency.

# References

- [1] M. Yu and Y. Wang, "Synthesis and Beyond," *IEEE Microwave Magazine*, vol. 12, no. 6, pp. 62-76, Oct. 2011.
- [2] R. J. Cameron, C. M. Kudsia, and R. T. Mansour, *Microwave Filters for Communication Systems: Fundamentals, Design, and Applications*, Hoboken, New Jersey: John Wiley & Sons, Inc, 2007.
- [3] D. Swanson and G. Macchiarella, "Microwave filter design by synthesis and optimization," *IEEE Microwave Magazine*, vol. 8, no. 2, pp55-69, Apr. 2007.
- [4] 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.
- [5] J. W. Bandler, R. M. Biernacki, S. H. Chen, R. H. Hemmers, and K. Madsen, "Electromagnetic optimization exploiting aggressive space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 43, no. 12, pp. 2874-2882, Dec. 1995.

- [6] J. E. Rayas-Sánchez, “Power in simplicity with ASM: tracing the aggressive space mapping algorithm over two decades of development and engineering applications,” *IEEE Microwave Magazine*, vol. 17, no. 4, pp. 64-76, Apr. 2016.
- [7] J. C. Cervantes-González, J. E. Rayas-Sánchez, C. A. López, J. R. Camacho-Pérez, Z. Brito-Brito, and J. L. Chavez-Hurtado, “Space mapping optimization of handset antennas considering EM effects of mobile phone components and human body,” *Int. J. RF and Microwave CAE*, vol. 26, no. 2, pp. 121-128, Feb. 2016.
- [8] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard, “Space mapping: The state of the art,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 337-361, Jan. 2004.
- [9] M. A. Ismail, D. Smith, A. Panariello, Y. Wang, and M. Yu, “EM based design of large-scale dielectric-resonator filters and multiplexers by space mapping,” *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 1, part 2, pp. 386392, Jan. 2004
- [10] Feng 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, May 2014.
- [11] C. Zhang, F. Feng, V. 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 Techn.*, vol. 63, no. 7, pp. 2154-2165, Jul. 2015.
- [12] M. Yu, “Simulation/design techniques for microwave filters - An engineering perspective,” in *IEEE MTT-S Int. Microwave Symp. Dig., Workshop WSA: State-of-the-Art Filter Design Using EM and Circuit Simulation Techniques*, Phoenix, AZ, May 2001.
- [13] G. Pepe, F.-J. Gortz, and H. Chaloupka, “Computer-aided tuning and diagnosis of microwave filters using sequential parameter extraction,” in *IEEE MTT-S Int. Microwave Symp. Dig.*, Apr. 2004, vol. 3, pp. 1373-1376.
- [14] M. Kahrizi, S. Safavi-Naeini, S. K. Chaudhuri, and R. Sabry, “Computer diagnosis and tuning of RF and microwave filters using model-based parameter estimation,” *IEEE Trans. Circuits Syst. I*, vol. 49, no. 9, pp. 1263-1270, Sept. 2002.
- [15] M. Meng and K.-L. Wu, “An analytical approach to computer-aided diagnosis and tuning of lossy microwave coupled resonator filters,” *IEEE Trans. Microwave Theory Tech.*, vol. 57, no. 12, pp. 3188-3195, Dec. 2009.
- [16] M. Yu, “Robotic computer-aided tuning,” *Microwave J.*, pp. 136-138, Mar. 2006.
- [17] 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.

- [18] 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.*, Accepted May 2018.
- [19] J. W. Bandler, and S. H. Chen, "Circuit optimization: the state of the art," *IEEE Trans. Microw. Theory Techn.*, vol. 36, no. 2, pp. 424-443, Feb. 1988.
- [20] F. Feng and Carleton University. Theses and Dissertations. Engineering, Electrical and Computer, *Parametric Modeling of EM Behaviors of Microwave Components using Combined Neural Networks and Pole-Residue Transfer Functions*. 2017.
- [21] Gongal Reddy, Venu Madhav Reddy and Carleton University. Theses and Dissertations. Engineering, Electrical and Computer, *Parallel Computational Approach to Gradient Based EM Optimization of Microwave Structures*. 2016.
- [22] 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. MTT-34, No. 12, Dec. 1986.
- [23] J. W. Bandler, W. Kellermann, and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design," *IEEE Trans. Microw. Theory Techn.*, vol. 33, no. 12, pp. 1519-1530, Dec. 1985.
- [24] M. B. Steer, J. W. Bandler, and C. M. Snowden, "Computer-aided design of RF and microwave circuits and systems," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 3, pp. 996-1005, Mar. 2002.

- [25] 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.
- [26] S. Koziel, S. Ogurtsov, J. W. Bandler, and Q. S. Cheng, "Reliable space-mapping optimization integrated with EM-based adjoint sensitivities," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 10, pp. 3493-3502, Oct. 2013.
- [27] 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.
- [28] J. W. Bandler, M. A. Ismail, and J. E. Rayas-Sánchez, "Expanded space-mapping EM-based design framework exploiting preassigned parameters," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 12, pp. 1833-1838, Dec. 2002.
- [29] J. W. Bandler, Q. S. Cheng, D. H. Gebre-Mariam, K. Madsen, F. Pedersen, and J. Sondergaard, "EM-based surrogate modeling and design exploiting implicit, frequency and output space mappings," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Philadelphia, PA, Jun. 2003.
- [30] J. W. Bandler, D. M. Hailu, K. Madsen, and F. Pedersen, "A space mapping interpolating surrogate algorithm for highly optimized EM-based design of

- microwave devices,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 11, pp. 2593-2600, Nov. 2004.
- [31] 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.
- [32] J. E. Rayas-Sánchez, F. Lara-Rojo, and E. Martínez-Guerrero, “A linear inverse space mapping (LISM) algorithm to design linear and nonlinear RF and microwave circuits,” *IEEE Trans. Microwave Theory Techn.*, vol. 53, no. 3, pp. 960-968, Mar. 2005.
- [33] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sánchez, and Q. J. Zhang, “Neural inverse space mapping (NISM) for EM-based microwave design,” *Int. J. RF and Microwave CAE*, vol. 13, no. 2, pp. 136-147, Mar. 2003.
- [34] J. Meng, S. Koziel, J. W. Bandler, M. H. Bakr, and Q. S. Cheng, “Tuning space mapping: A novel technique for engineering design optimization,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, Atlanta, Georgia, pp. 991-994, Jun. 2008.
- [35] Q. S. Cheng, J. W. Bandler, and J. E. Rayas-Sánchez, “Tuning-aided implicit space mapping,” *Int. J. RF and Microwave CAE*, vol. 18, no. 5, pp. 445-453, Sep. 2008.
- [36] S. Koziel, J. Meng, J. W. Bandler, M. H. Bakr, and Q. S. Cheng, “Accelerated microwave design optimization with tuning space mapping,” *IEEE Trans. Microwave Theory Techn.*, vol. 57, no. 2, pp. 383-394, Feb. 2009.

- [37] S. Koziel, J. W. Bandler, and Q. S. Cheng, "Tuning space mapping design framework exploiting reduced electromagnetic models," *IET Microw. Antennas Propag.*, vol. 5, no. 10, pp. 1219-1226, Jul. 2011.
- [38] S. Koziel and J. W. Bandler, "Space-mapping with adaptive response correction for microwave design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 2, pp. 478-486, Feb. 2009.
- [39] C. Zhang, Feng Feng, and Q. J. Zhang, "EM optimization using coarse and fine mesh space mapping," in *Proc. Asia-Pacific Microw. Conf.*, Seoul, Korea, pp. 824-826, Dec. 2013.
- [40] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, "Exploitation of coarse grid for electromagnetic optimization," in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Diego, CA, USA, pp. 381-384, May 1994.
- [41] F. Feng, C. Zhang, V. Gongal-Reddy, and Q. J. Zhang, "Knowledge-based coarse and fine mesh space mapping approach to EM optimization," in *IEEE MTT-S Int. Conf. on Numerical Electromagnetic Modeling and Optimization*, Pavia, Italy, May 2014.
- [42] S. Koziel and S. Ogurtsov, *Antenna Design by Simulation-Driven Optimization*, Springer, 2014.
- [43] S. Koziel, Q. S. Cheng, and J. W. Bandler, "Space mapping," *IEEE Microw. Mag.*, vol. 9, no. 6, pp. 105-122, Dec. 2008.

- [44] S. Selleri, S. Manetti, and G. Pelosi, "Neural network applications in microwave device design," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 12, pp. 90-97, Jan. 2002.
- [45] S. Koziel and A. Bekasiewicz, "Expedited geometry scaling of compact microwave passives by means of inverse surrogate modeling," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4019-4026, Dec. 2015.
- [46] S. Koziel, and A. Bekasiewicz, "Inverse surrogate modeling for low-cost geometry scaling of microwave and antenna structures," *Eng. Comp.*, vol. 33, no. 4, pp. 1095-1116, 2016.
- [47] S. Koziel and A. Bekasiewicz, "Computationally-efficient surrogate-assisted dimension scaling of compact dual-band couplers," *IET Microw. Antennas Propag.*, vol. 11, no. 4, pp. 465-470, 2017.
- [48] S. Koziel and A. Bekasiewicz, "Rapid dimension scaling for notch frequency re-design of UWB band-notch antennas," *J. Electromagnetic Waves Appl.*, vol. 30, no. 17, pp. 2280-2292, 2016.
- [49] S. Koziel, A. Bekasiewicz, and L. Leifsson, "Rapid EM-driven antenna dimension scaling through inverse modeling," *IEEE Antennas and Wireless Propag. Lett.*, vol. 15, pp. 714-717, Mar. 2016.
- [50] H. Acikgoz, B. Jannier, Y. Le Bihan, O. Dubrunfaut, O. Meyer, and L. Pichon, "Direct and inverse modeling of a microwave sensor determining the proportion of fluids in a pipeline," *IEEE Trans. Magnetics*, vol. 45, no. 3, Mar. 2009.

- [51] G. Gosal, E. Almajali, D. McNamara, and M. Yagoub, "Transmitarray antenna design using forward and inverse neural network modeling," *IEEE Antennas and Wireless Propag. Lett.*, vol. 15, pp. 1483-1486, May 2016.
- [52] B. Liu, H. Yang, and M. J. 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.
- [53] D. S. Weile and E. Michielssen, "Genetic algorithm optimization applied to electromagnetics: a review," *IEEE Trans. Antennas. Propag.*, vol. 45, no. 3, pp. 343-353, Mar. 1997.
- [54] G. Crevecoeur, P. Sergeant, L. Dupre, and R. Van de Walle, "A two-level genetic algorithm for electromagnetic optimization," *IEEE Trans. Magn.*, vol. 46, no. 7, pp. 2585-2595, Jul. 2010.
- [55] C. M. Coleman, E. J. Rothwell, and J. E. Ross, "Investigation of simulated annealing, ant-colony optimization, and genetic algorithms for self-structuring antennas," *IEEE Trans. Antennas Propag.*, vol. 52, no. 4, pp. 1007-1014, Apr. 2004.
- [56] J. Zhang, K. Ma, F. Feng, and Q. J. Zhang, "Parallel gradient-based local search accelerating particle swarm optimization for training microwave neural network models," *IEEE MTT-S Int. Microw. Symp. Dig.*, Phoenix, Arizona, USA, May 2015.

- [57] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Math. Comp.*, vol. 24, pp. 647-656, 1970.
- [58] J. W. Bandler, S. H. Chen, S. Daijavad, and K. Madsen, "Efficient optimization with integrated gradient approximations," *IEEE Trans. Microw. Theory Techn.*, vol. 36, pp. 444-455, Feb. 1988.
- [59] S. S. Rao, *Engineering Optimization*, NJ, John Wiley & Sons, 1996.
- [60] S. Koziel, *Simulation-Driven Design Optimization and Modeling for Microwave Engineering*, Imperial College Press, 2013.
- [61] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems". *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, 1952.
- [62] R. Fletcher, C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149-154, 1964.
- [63] K. Madsen and H. B. Nielsen, *Introduction to Optimization and Data Fitting*, 2008.
- [64] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM J. Optim.*, vol. 10, no. 1, pp. 177-182, 1999.
- [65] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numer.*, vol. 4, pp. 1-51, Jan. 1995.

- [66] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 2006.
- [67] E. D. Dolan, R. M. Lewis, and V. J. Torczon, “On the local convergence of pattern search,” *SIAM Journal on Optim.*, vol. 14, no. 2, pp. 567-583, 2003.
- [68] R. Hooke and T. A. Jeeves, ““Direct search,” solution of numerical and statistical problems,” *Journal of the ACM*. vol. 8, no. 2, pp. 212-229, 1961.
- [69] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *Computer Journal*, vol. 7, pp.308 - 313, 1965.
- [70] K. I. M. McKinnon, “Convergence of the Nelder-Mead simplex method to a non-stationary point,” *SIAM J Optimization* vol.9, pp. 148-158, 1999.
- [71] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the Nelder-Mead simplex method in low dimensions,” *SIAM J. Optim.*, vol. 9, no. 1, pp. 112-147, 1998.
- [72] N. M. Alexandrov, J. E. Dennis Jr., R. M. Lewis, and V. Torczon, “A trust-region framework for managing the use of approximation models in optimizations,” *Struct. Optim.*, vol. 15, no. 1, pp. 16-23, Feb. 1998.
- [73] N. M. Alexandrov and R. M. Lewis, “An overview of first-order model management for engineering optimization,” *Optim. Eng.*, vol. 2, no. 4, pp. 413-430, Dec. 2001.

- [74] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman, "Approximation and model management in aerodynamic optimization with variable-fidelity models," *J. of Aircraft*, vol. 38, no. 6, pp. 1093-1101, Dec. 2001.
- [75] S. Koziel, J. W. Bandler, and Q. S. Cheng, "Constrained parameter extraction for microwave design optimization using implicit space mapping," *IET Microw., Antennas Prop.*, vol. 5, no. 10, pp. 1156-1163, Jul. 2011.
- [76] S. Koziel, S. Ogurtsov, J. W. Bandler, and Q. S. Cheng, "Reliable space-mapping optimization integrated with EM-based adjoint sensitivities," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 10, pp. 3493-3502, Oct. 2013.
- [77] S. Koziel, Q. S. Cheng, and J. W. Bandler, "Fast EM modeling exploiting shape-preserving response prediction and space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 3, pp. 399-407, Mar. 2014.
- [78] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Boston: Artech House, 2000
- [79] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microwave Theory Tech.*, vol. 51, pp. 1339-1350, Apr. 2003.
- [80] 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.

- [81] 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.
- [82] V. K. Devabhaktuni, M. C. E. Yagoub, Y. Fang, J. J. Xu, and Q. J. Zhang, “Neural networks for microwave modeling: Model development issues and nonlinear modeling techniques,” *Int. J. RF Microwave Computer-Aided Eng.*, vol. 11, pp. 4-21, Jan. 2001.
- [83] M. Joodaki and G. Kompa, “A systematic approach to a reliable neural model for pHEMT using different numbers of training data,” in *IEEE MTT-S Int. Microwave Symp. Dig.*, Seattle, WA, Jun. 2002, pp. 1105-1108.
- [84] P. Burrascano and M. Mongiardo, “A review of artificial neural networks applications in microwave CAD,” *Int. J. RF Microwave Computer-Aided Eng.*, vol. 9, pp. 158-174, May 1999.
- [85] P. M. Watson, K. C. Gupta, and R. L. Mahajan, “Development of knowledge based artificial neural networks models for microwave components,” in *IEEE MTT-S Int. Microwave Symp. Dig.*, Baltimore, MD, Jun. 1998, pp. 912.
- [86] F. Wang, and Q. J. Zhang, “Knowledge based neuromodels for microwave design,” *IEEE Trans. Microwave Theory Techn.*, vol. 45, pp. 2333-2343, Dec. 1997.
- [87] V. Gutierrez-Ayala and J. E. Rayas-Sánchez, “Neural input space mapping optimization based on nonlinear two-layer perceptrons with optimized nonlin-

- earity,” *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 20, no. 5, pp. 512-526, Sep. 2010.
- [88] M. H. Bakr, J. W. Bandler, M. A. Ismail, J. E. Rayas-Snchez, and Q. J. Zhang, “Neural space-mapping optimization for EM-based design,” *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2307-2315, Dec. 2000.
- [89] 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.
- [90] 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.
- [91] 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. MTT-34, No. 12, pp. 1282-1293, 1986.
- [92] J. W. Bandler, W. Kellermann, and K. Madsen. “A nonlinear  $l_1$  optimization algorithm for design, modeling, and diagnosis of networks,” *IEEE Trans. Circuits Systems.*, vol. CAS-34, no. 2, pp. 174-181, 1987.
- [93] 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.
- [94] A. Massa, G. Oliveri, M. Salucci, N. Anselmi, and P. Rocca, "Learning by-examples techniques as applied to electromagnetics," *Journal of Electromagnetic Waves and Applications*, vol. 32, no. 4, pp. 516-541, Nov. 2017.
- [95] C. Cortes, V. Vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, No. 3, pp 273-297, Sept. 1995,
- [96] P. H. Chen, R. E. Fan, and C. J. Lin, "A study on SMO-type decomposition methods for support vector machines," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 893-908, 2006.
- [97] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press. 2006.
- [98] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Network*, vol. 2, no. 5, pp. 359-366, 1989.
- [99] M. Kuczmann and A. Ivanyi, "A new neural-network-based scalar hysteresis model," *IEEE Trans. Magnetics*, vol. 38, no. 2, pp. 857-860, May 2016.
- [100] J. W. Bandler, "Have you ever wondered about the engineer's mysterious 'feel' for a problem?" *IEEE Canadian Review* vol. 70, pp. 50-60, Summer. 2013.

- [101] S. Koziel and J. W. Bandler, "Space mapping optimization with adaptive surrogate model," *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 3, pp. 541-547, Mar. 2007.
- [102] S. Koziel, J. W. Bandler, and K. Madsen, "A space mapping framework for engineering optimization: theory and implementation," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 10, pp. 3721-3730, Oct. 2006.
- [103] L. Zhang, P. H. Aaen, and J. Wood, "Portable space mapping for efficient statistical modeling of passive components," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 441-450, Mar. 2012.
- [104] S. Koziel, "Shape-preserving response prediction for microwave design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 11, pp. 2829-2837, Nov. 2010.
- [105] N. Leszczynska, L. Szydlowski, and M. Mrozowski, "Zero-pole space mapping for CAD of filters," *IEEE Microwave Wireless Comp. Lett.*, vol. 24, no. 9, pp. 581-583, Sep. 2014.
- [106] J. Zhu, J. W. Bandler, N. K. Nikolova, and S. Koziel, "Antenna optimization through space mapping," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 651-658, Mar. 2007.
- [107] J. V. Morro, P. Soto, H. Esteban, V. E. Boria, C. Bachiller, M. Taroncher, S. Cogollos, and B. Gimeno, "Fast automated design of waveguide filters using

- aggressive space mapping with a new segmentation strategy and a hybrid optimization algorithm,” *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 4, pp. 1130-1142, Apr. 2005.
- [108] M.H. Bakr, J.W. Bandler, N. Georgieva, and K. Madsen, “A hybrid aggressive space mapping algorithm for EM optimization,” *IEEE Trans. Microw. Theory Techn.*, vol. 47, no. 12, pp. 2440-2449, Dec. 1999.
- [109] S. Koziel, S. Ogurtsov, J. W. Bandler, and Q. S. Cheng, “Reliable space-mapping optimization integrated with EM-based adjoint sensitivities,” *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 10, pp. 3493-3502, Oct. 2013.
- [110] W. A. Atia, K. A. Zaki, and A. E. Atia, “Synthesis of general topology multiple coupled resonator filters by optimization,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, vol. 2, pp. 821-824, June 1998.
- [111] S. Amari, “Synthesis of cross-coupled resonator filters using an analytical gradient-based optimization technique,” *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 9, pp. 1559-1564, Sep. 2000.
- [112] P. Kozakowski and M. Mrozowski, “Automated CAD of coupled resonator filters,” *IEEE Microw. Wirel. Compon. Lett.*, vol. 12, no. 12, pp. 470-472, Dec. 2002.
- [113] O. Glubokov and S. Koziel, “EM-driven tuning of substrate integrated waveguide filters exploiting feature-space surrogates,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, Tampa, FL, USA, Jun. 2014.

- [114] S. Koziel and A. Bekasiewicz, “Fast simulation-driven feature-based design optimization of compact dual-band microstrip branch-line coupler,” *Int. J. RF and Microwave CAE*, vol. 26, no. 1, pp.13-20, 2016.
- [115] S. Koziel and J.W. Bandler: ‘Rapid yield estimation and optimization for microwave structures exploiting feature-based statistical analysis’, *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 1, pp. 107-114, 2015.
- [116] S. Koziel and J. W. Bandler, “Feature-based statistical analysis for rapid yield estimation of microwave structures,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, Tampa, FL, USA, Jun. 2014.
- [117] S. Koziel and J. W. Bandler, “Reliable microwave modeling by means of variable-fidelity response features,” *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4247- 4254, 2015.
- [118] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon, “A trust region framework for managing use of approximation models in optimization,” *Struct. Multidisciplinary Optim.*, vol. 15, no. 1, pp. 16-23, 1998.
- [119] 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.
- [120] S. Koziel, J. W. Bandler, and Q. S. Cheng, “Robust trust-region space-mapping algorithms for microwave design optimization,” *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 8, pp. 2166-2174, Aug. 2010.

- [121] HFSS ver. 15.0, ANSYS Corporation, Pittsburgh, PA, 2011.
- [122] Y. Cao, S. Reitzinger, and Q. J. Zhang, "Simple and efficient high-dimensional parametric modeling for microwave cavity filters using modular neural network," *IEEE Microw. Wirel. Compon. Lett.*, vol. 21, no. 5, pp. 258-260, May 2011.
- [123] Q.-J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA, USA: Artech House, 2000.
- [124] D. E. Root, "Future device modeling trends," *IEEE Microw. Mag.*, vol. 13, no. 7, pp. 45-59, Nov./Dec. 2012.
- [125] 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.
- [126] 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, Jan. 2017.
- [127] 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.
- [128] M. M. Vai, S. Wu, B. Li, and S. Prasad, "Reverse modeling of microwave circuits with bidirectional neural network models," *IEEE Trans. Microw. Theory Techn.*, vol. 46, no. 10, pp. 1492-1494, Oct. 1998.

- [129] 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, Jan. 2017.
- [130] 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.
- [131] 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.
- [132] Y. Cao, S. Reitzinger, and Q.-J. Zhang, "Simple and efficient highdimensional parametric modeling for microwave cavity filters using modular neural network," *IEEE Microw. Wireless Compon. Lett.*, vol. 21, no. 5, pp. 258-260, May 2011.
- [133] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Boston, MA: Artech House, 2000.
- [134] S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," *Cognit. Sci.*, vol. 11, no. 1, pp. 23-63, Jan.-Mar. 1987.
- [135] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cogniti. Sci.* vol. 9, no. 1, pp. 75-112, 1985.
- [136] D. Gorissen, L. D. Tommasi, K. Crombecq, and T. Dhaene, "Sequential modeling of a low noise amplifier with neural networks and active learning," *Neural Comput. and Applic.*, vol. 18, no. 5, pp. 485-494, 2009.

- [137] J. L. Chavez-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.