

Modeling and Optimization of the Microwave PCB Interconnects Using Macromodel Techniques

Original

Modeling and Optimization of the Microwave PCB Interconnects Using Macromodel Techniques / Sedaghat, Mostafa. - (2022 Dec 15), pp. 1-100.

Availability:

This version is available at: 11583/2973989 since: 2022-12-20T08:01:16Z

Publisher:

Politecnico di Torino

Published

DOI:

Terms of use:

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)



Politecnico
di Torino

ScuDo
Scuola di Dottorato – Doctoral School
WHAT YOU ARE, TAKES YOU FAR



Doctoral Dissertation

Doctoral Program in Electronics Engineering (34th cycle)

Modeling and Optimization of the Microwave PCB Interconnects Using Macromodel Techniques

By

Mostafa Sedaghat

Supervisor(s):

Prof. Flavio G. Canavero, Supervisor

Prof. Zaker H. Firouzeh, Supervisor

Prof. Riccardo Trincherio, Co-Supervisor

Doctoral Examination Committee:

Prof. Rohit Y. Sharma, Indian Institute of Technology Ropar, India

Prof. Mihai Gabriel Telescu, University of Brest, France

Prof. Domenico Spina, UGent & IMEC, Belgium

Prof. Paolo Bardella, Politecnico di Torino, Italy

Prof. Igor S. Stievano, Politecnico di Torino, Italy

Politecnico di Torino

2022

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Mostafa Sedaghat
2022

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my loving parents

Acknowledgements

It gives me great pleasure in acknowledging the support of all the people who have helped me accomplishing this Thesis. First and foremost, I would like to thank my Isfahan University of Technology supervisor, Prof. Zaker H. Firouzeh, for his constant, irreplaceable support and presence in completion of this activity and guiding me in carrying out the joint research program with Politecnico di Torino.

I am indebted to Prof. Flavio Canavero, Prof. Riccardo Trincherro and EMC research group who provided me all the necessary things to carry out joint research and offering me this great opportunity which helped my deep dive research and cross-cultural life. Special thanks to Prof. Riccardo Trincherro for his ideas and discussions during these two fruitful years of my life.

I also acknowledge my EMC group colleagues who made the work environment enjoyable, with coffee and lunch breaks, which will definitely be cherished all my life. I owe my deepest gratitude to all the Iranian community in Torino, who have always stood with me during my stay, and felt like home; For sure Torino would remain my second home.

Last but not the least, I would like to thank my family for their constant support, prayers and love which made me who I am.

Abstract

Due to the operating frequency and signal bandwidth of communication applications, the electromagnetic simulation and modeling of modern electronic systems are essential. This is particularly problematic when attempting to solve problems that involve computationally expensive analysis (e.g., full-wave simulations) and parametric optimization. To address these problems, the use of fast and accurate surrogate models appears to be effective.

To this end, this study is divided into two parts. First, instead of attempting to characterize the entire system using Maxwell's equations, the overall system (such as a multilayer printed circuit board) is first divided into substructures (such as coupled line interconnects, vias, connectors, etc.), each of which is then further characterized using the standard *macromodels* (i.e., reduced-complexity behavioral models). As a result, the electrical interconnects at the board level and circuit components are effectively analyzed by using corresponding macromodels requiring limited computing resources.

Focusing on the black-box macromodeling approach, a rational approximation based on the integral operator is proposed. By integrating the data on several different intervals, a system of equations is formed, then the pole/residue values of the rational function are obtained by the conventional least-squares method. Finally, the stability of the model is guaranteed by using a closed-loop control technique and considering a controller coefficient. Also, using this parameter, the designer can increase the stability margin of a system with poor stability conditions. The introduced method has the potential to be used for a wide range of practical applications since there is no specific restriction on the use of this method. The only requirement that should be considered is the *Dirichlet condition* for the original data, which is usually the case for physical systems. The performance evaluation of the proposed method has been investigated in comparison with the well-known vector fitting (VF) method.

Also, the results show that the performance of the proposed method is less affected by input noise, and this is an important point because in most cases the measurement data is noisy.

In the second part, a new non-iterative optimization method based on the inverse modeling technique and its applications in the design and optimization of microwave components is presented. The proposed inverse model accepts the high-dimensional S-parameters computed at many frequency points as the input and estimates the optimal geometrical/physical parameters of the microwave component as its output. The Least-Square Support Vector Machine regression is combined with the Principal Component Analysis to simultaneously overcome both the high-dimensional input space and ill-posed challenges of the inverse modeling. We also propose a new empirical method to find the optimum number of principal components (i.e., compression level) for each example, in an automated way. This makes our proposed model general and easy to use compared with the existing data-driven inverse modeling techniques. The inverse model is trained by a set of scattering parameters computed via a 2D/3D electromagnetic solver for a few configurations of the geometrical parameters. The feasibility and the accuracy of the proposed optimization scheme are investigated by comparing its predictions with the corresponding optimal configuration estimated via a commercial solver.

Contents

List of Figures	ix
List of Tables	xii
1 Introduction	1
1.1 Research Background	1
1.2 Literature Review	2
1.3 Organization of the Thesis	4
2 Macromodeling Technique and Applications (state-of-the-art)	6
2.1 Introduction to Macromodel Technique	7
2.2 Rational Macromodel	8
2.3 Vector Fitting	11
2.3.1 Example: Microstrip Line	14
2.4 Synthesis of Equivalent Circuit	16
2.4.1 Example: Multilayer PCB Structure	18
2.5 Passivity Enforcement	20
2.6 Non-iterative Proposed Rational Macromodel based on Data Integration	22
2.6.1 Formulation	23
2.6.2 Practical Examples	28
2.7 Parameterized Macromodel	33

3	Basics of Data-driven Surrogate Modeling	37
3.1	Latin Hypercube Sampling Strategy	38
3.2	Model Identification: Least Square Support Vector Machine Regression	38
3.3	Surrogate Model Validation	41
4	Inverse Modeling Optimization	44
4.1	Literature Review	44
4.2	Inverse Model Formulation and Challenges	47
4.3	Proposed Scheme for the Inverse Model Construction	49
4.3.1	Dimensionality compression via PCA	49
4.3.2	LS-SVM regression: curse-of-dimensionality and ill-posed problems	51
4.3.3	Discussion of the ill-posed challenge	53
4.4	Inverse Model's Hyperparameters Optimization	56
4.5	Practical examples	61
4.5.1	Example I: The Inverse Modeling of a Coupled Line Microstrip Filter	61
4.5.2	Example II: The Inverse Modeling of a Wideband Common Mode Suppression Filter	68
5	Conclusion	75
5.1	Future work	77
	References	79
	Appendix A List of Contributions	87

List of Figures

1.1	A conceptual example of the application of the macromodel in accelerating the simulation of complex electronic devices. From left to right, full-wave simulation of a generic complex PCB, macromodeling of the specific passive substructure of the PCB, and its synthesized equivalent circuit netlist to be used in SPICE-like time domain solver.	2
2.1	A practical example of the signal/power integrity analysis framework using the macromodel techniques.	8
2.2	(a) Microstrip structure, and (b) amplitude of microstrip S-parameters obtained from simulation in CST.	15
2.3	Resulting S-parameters amplitude fit (red dashed line) for microstrip example after 1 iteration of the VF algorithm with $N = 20$ complex conjugate poles.	16
2.4	Macromodeling flow based on data from full-wave solvers (given from [1]).	17
2.5	Regenerated S-parameters of the synthesized netlist in ADS for microstrip example.	17
2.6	(a) Multilayer PCB structure, and (b) the S-parameters obtained from simulation in CST.	19
2.7	Fitting the S-parameters of the multilayer PCB example (blue line) using the MATLAB VF Toolbox (red dashed line) with $N = 10$ complex conjugate poles after 1 iteration.	19

2.8	(a) Netlist of the multilayer PCB example and (b) the S-parameters of the netlists generated by CST and direct state-space synthesis (VF Netlist) methods obtained from simulation in ADS.	21
2.9	Full frequency sweep of the singular values of the multilayer PCB scattering matrix before and after fitting.	22
2.10	Block diagram of a closed-loop system.	27
2.11	(a) Magnitude, and (b) phase of synthesized transfer function of theoretical example in the absence of noise.	29
2.12	(a) Magnitude, (b) phase, and (c) mean square error of synthesized transfer function of theoretical example with noise.	31
2.13	The fabricated structure of coupled microstrip line [25].	32
2.14	(a) Magnitude, and (b) phase of the synthesized far-end crosstalk of coupled microstrip line.	32
4.1	Illustration of different inverse modeling techniques for the coupled-line microstrip filter example. (a) Input-output relationship of the forward problem is unique. (b) Inverse problem relationship (solid blue line) and LS-SVM direct inverse modeling techniques training (dashed red line) results. The ML-based inverse model with only one input, i.e., $y = S_{21}(f_1 = 2500 \text{ MHz}) $, cannot match the contradictory data (multivalued solutions). (c) Training results of the direct inverse model with 2-vector input. Increasing the input dimensionality, a new well-posed inverse mapping is introduced which helps the LS-SVM regression to be well trained (dashed red line). Two S-parameter samples, i.e., $ S_{21}(f_1) $ and $ S_{21}(f_2) $, $f_1 = 2500 \text{ MHz}$ and $f_2 = 2600 \text{ MHz}$, are used as the inverse model input.	56
4.2	(a) Successive combination of ML-based inverse and forward models used in CV scheme for tuning the inverse model hyperparameter \tilde{n} . (b) Using the full computational model (e.g., ADS) for the test of the inverse model predictions in practical microwave design examples.	58
4.3	Illustrative example of K -fold cross validation scheme with $K = 10$ and 200 training samples.	59

4.4	(a) 6-stage coupled-line microstrip filter considered in Example I [81]. (b) proposed approach for design optimization based on the inverse model.	62
4.5	Analysis of PCA coefficients number \tilde{n} by proposed 5-fold cross validation Algorithm 1, and with $\hat{n} = 3$. The optimum $\tilde{n}_{opt} = 4$ is selected for building the coupled line microstrip inverse model. . . .	63
4.6	Evaluation of the trained ML-based forward model $\hat{\mathcal{M}} : \mathbb{R}^9 \rightarrow \mathbb{R}^{\hat{n}}$ of the coupled-line microstrip filter in predicting the first ten PCA coefficients (i.e., $\hat{n} = 1, 2, \dots, 10$), using 100 unseen validation samples. The LS-SVM regressions have already been trained using a separate set of 100 samples.	64
4.7	Frequency responses of the initial design (dashed gray line) and the optimized filters for the given goal specifications in section 4.5, obtained by ADS (dashed black line), inverse model via LS-SVM (dotted blue line), and proposed compressed inverse model combining LS-SVM and PCA (solid red line), also the ideal mask used as input of the inverse model (dashed-dotted green line).	66
4.8	Tunability assessment of the proposed inverse model with $\tilde{n} = 4$. (a) model tunability is limited to the range of \mathbf{x} used during the model training. (b) scattering parameters of the optimized filter by changing given filter specifications in two cases: (i) $F_C = 2700MHz$ and $BW = 70MHz$ and (ii) $F_C = 2550MHz$ and $BW = 50MHz$	68
4.9	Common-mode suppression filter with its geometrical parameters. (a) Cross-section. (b) Top view. (from [84])	69
4.10	Analysis of PCA coefficients number \tilde{n} by proposed 5-fold cross validation Algorithm 1, and with $\hat{n} = 5$. The optimum $\tilde{n}_{opt} = 3$ is selected for building the common mode suppression filter inverse model.	71
4.11	Frequency responses of the initial (dashed-dotted green line) and the optimized design for the common mode suppression filter, obtained by ADS (dashed black line), and the proposed inverse model combining LS-SVM and PCA (solid red line). Also, the ideal goal mask used as input for the inverse model (dashed-dotted blue line).	72

List of Tables

2.1	VF algorithm performance for microstrip example.	15
2.2	VF algorithm performance for multilayer PCB example.	18
2.3	Poles and residues of the TF of theoretical example	28
2.4	Comparison of the condition number of the Vandermonde and the proposed coefficients matrix for all examples.	33
4.1	Preliminar Design of the Proposed Microwave Filter	61
4.2	Filter response specifications optimized by inverse model and ADS optimizer	65
4.3	Initial Design of the Common Mode Suppression Filter	70
4.4	The Computational Cost of the Inverse Modeling and Direct Optimization of Practical Examples in section 4.5	74

Chapter 1

Introduction

1.1 Research Background

Understanding the behavior of structures and systems in practically higher frequencies requires numerical modeling and simulation [1]. For instance, evaluating the signal/power integrity issues of a generic PCB at microwave frequencies require full-wave EM simulations over a large domain, including dielectric and metal passive structures with small geometrical details connected to localized nonlinear devices as termination networks. Direct methods, such as characterizing the entire system with Maxwell's equations and then simulating it on an appropriate computer, are impractical for this purpose [1].

The hybrid simulation techniques have proven to be good alternatives for reducing this complexity and carrying out effective numerical simulations with realistic runtimes [2]. This way, the overall system is usually divided into well-defined substructures (e.g., narrow traces, vias, integrated circuits, passive interconnects, connectors) so that each one can first be separately characterized through reduced complexity surrogate models, a.k.a *macromodels* (see the conceptual Figure 1.1).

Terminal responses in the time or frequency domain can be used to identify macromodels, which results in concise behavioral representations of the structure. These macromodels are connected once they are available to create an approximation but accurate representation of the entire system, which can subsequently be solved using limited computing resources. Indeed, complex system-level evaluations based on numerical simulations can be greatly simplified using macromodels, leading

to a major acceleration in simulation time, almost without loss of accuracy (e.g., a speedup up to 700X in power integrity transient simulation of a Low DropOut (LDO) voltage regulator in electronic systems [3]). Therefore, we need reliable and stable algorithms for constructing a macromodel that makes such an efficient process possible.

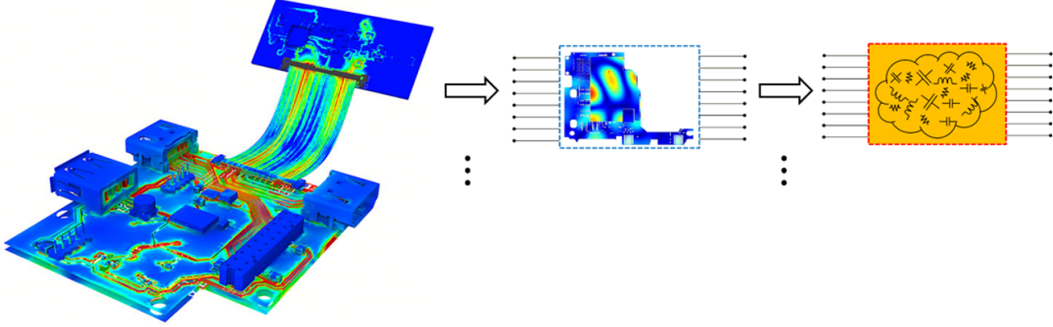


Fig. 1.1 A conceptual example of the application of the macromodel in accelerating the simulation of complex electronic devices. From left to right, full-wave simulation of a generic complex PCB, macromodeling of the specific passive substructure of the PCB, and its synthesized equivalent circuit netlist to be used in SPICE-like time domain solver.

1.2 Literature Review

A variety of macromodel techniques have been developed for passive electrical interconnects [1]. Among the black-box macromodeling techniques, two well-known are the Vector Fitting (VF) [4, 5], and the Loewner Matrix [6, 7]. The above are able to extract a rational approximation of the transfer function of the system under test through robust algorithms. The resulting mathematical expressions are easily converted to ordinary differential equations (ODEs) and synthesized by equivalent circuits (Figure 1.1) compatible with standard circuit solvers (e.g., SPICE) [8]. We assume that the component response can be reproduced by a linear time-invariant system (LTI). Therefore, such a macromodel is applicable both to inherently linear structures such as passive interconnects, and to active nonlinear components represented by a small signal transfer function (such as [3]).

According to the standard approach, a macromodel is first constructed using full-wave simulation (or measurement) data to extract the port response (e.g., frequency dependent scattering matrix) of the structure. Then, by applying the appropriate

algorithms, we obtain a rational approximation for the response so that the resulting macromodel is converted to a format compatible with circuit solvers (e.g., SPICE). However, when the structure of interest is dependent on one or more design variables, the entire process of creating the macromodel (i.e., full-wave simulations, rational approximation, etc.) must be repeated for each different combination of the parameters, making this process impractical.

In recent decades, rapid advances in the field of microwaves have led to the production of compact passive components with higher operating frequencies. Also, these high frequency components are quite sensitive to dimensional changes so a small change in geometry can lead to significant differences in their overall performance. To get the optimum dimensions for high-frequency devices, multiple simulation iterations are usually required during the design stage. As a result, there is a strong demand for microwave device models that are accurate, dimensionally dependent, fast-simulation-speed, and simulator compatible [9]. Specifically, these models should be available as needed for any configuration of design variables, including geometric parameters (such as the width and spacing of the transmission lines on PCBs, substrate height, etc.) or material parameters (such as electrical conductivity, permittivity, etc.). A model with such features is called a *parameterized macromodel*.

Other common modeling methods that are widely used for microwave components are equivalent circuit models [10], EM physical models [11] and S-parameter models [12]. Due to its fast evaluation and the capability of including geometrical parameters, the equivalent circuit models are ideal for parametric optimization. However, as the operating frequency increases, the accuracy of the equivalent circuit model degrades exponentially because a large number of parasitic elements become prominent at higher frequency ranges that are not included in the model. The primary reason for this is that accurately identifying such parasitic elements is difficult and time-consuming. In contrast, the EM physical models are generally accurate even for higher frequencies. But they require a high computation time and memory making them unsuitable for parametric simulations, in particular for cases with many geometrical parameters [13].

Data-driven surrogate models are suitable solutions to many practical modeling problems [9]. Their main advantages are versatility, low evaluation costs, a wide selection of well-established methods, as well as the availability of ready-to-use

toolboxes to implement these methods in popular programming environments such as MATLAB. However, these models have serious limitations. Major problems arise from the high cost of providing the training data, the curse of dimensionality, as well as the wide range of parameters for which the model must be valid [13]. Even in the case of high-frequency structures such as microwave components and antennas, we will face more problems, including the severity of the system's response to parameter changes, as well as dealing with vector-valued outputs [14].

1.3 Organization of the Thesis

In this research, we look for modeling methods that are much less computationally expensive than full-wave solvers, also referred to as surrogates. The focus is on modeling passive PCB interconnects and components at microwave frequencies. Considering the surrogate modeling and surrogate-based design of high-frequency structures, the thesis is organized into two main parts.

In the first part, the state-of-the-art macromodeling technique is introduced as a standard for the analysis of electrical interconnects at the chip, package, or board level, and microwave components. Chapter 2 reviews the state-of-the-art systematic method for signal integrity analysis of high-speed interconnects. The macromodel is identified from terminal responses in the frequency domain (e.g., S-parameters). Frequency samples of the scattering matrix of the whole structure are obtained from different cascaded components (transmission line, vias, connectors, packages, etc.) whose individual scattering matrix is obtained by fast 2D (for transmission lines) or full-wave (for connectors, vias, and other structures with unusual shapes) solvers. Then, the macromodel of the interconnects is constructed by fitting a rational function, which is synthesizable by equivalent circuits compatible with circuit-based solvers (e.g., SPICE, ADS) to be used in conjunction with other nonlinear components for transient time-domain simulations. To do this, a variety of nonlinear component models available from the ADS library could be used for transient simulations. In Chapter 3, a rational macromodel based on data integration is proposed with the aim of utilizing all available information, reducing the order of the model and relative resistance to noise. Finally, the stability of the model is guaranteed by using the closed-loop control technique and considering a control coefficient. The performance evaluation of the proposed method in comparison with the well-known VF method

has been investigated in several practical examples. The main goal is to develop efficient algorithms that model high-speed interconnects to be used in a circuit-based solver such as ADS.

In the second part, Chapter 4, the parameterized macromodel is introduced which includes external parameters (e.g., geometrical and physical parameters or even external incident field parameters) other than the frequency as an enabling factor for design optimization of electronic devices based on conventional iterative optimizers. Also in this chapter, we propose a data-driven model based on a completely different approach, so-called "*Inverse Modeling*", to optimize passive microwave components, which, unlike conventional repetitive optimization methods, is very efficient and does not require repetition. Accordingly, an inverse model based on the machine learning regression technique is proposed to optimize microwave components, addressing two main challenges of the inverse modeling microwave problems. We evaluate the feasibility and accuracy of the proposed method by optimizing two microwave application examples and comparing them with similar results from the commercial optimizer available in ADS. Chapter 5 deals with the conclusions and suggestions for further work.

Chapter 2

Macromodeling Technique and Applications (state-of-the-art)

The development of advanced modeling and simulation techniques enables electronic device designers to fully identify electromagnetic compatibility (EMC) related issues which have a detrimental effect on signals within a system. In this context, innovative modeling methods have been recently researched according to EM fields, electrical interconnects, circuits, and components [1].

One of the main research activities in this field is macromodeling of electrical interconnects. It is aimed at providing the designers of electronic systems with accurate and efficient models of electrical interconnections. These may include connectors, packages, transitions, junctions and discontinuities, vias on printed circuit boards (PCBs), and also transmission lines. All these structures have a significant influence on the signals that propagate through them. Our research is focused on the development of efficient macromodeling procedures leading to circuit models characterized by good accuracy and by reduced complexity. This involves the theoretical development of the macromodeling algorithms and their validation of realistic structures.

Macromodeling based on rational function approximation (rational macromodel) is a standard method in various applications. In this method, the frequency response of the desired structure is first extracted by the full-wave simulation or measurement (e.g., frequency-dependent scattering parameters). Then, by applying the appropriate curve fitting algorithms, a rational function approximation of the frequency response

is obtained. Such a macromodel is available in a form that is directly usable within standard SPICE-like circuit solvers.

2.1 Introduction to Macromodel Technique

Macromodeling techniques developed for various applications to signal and power integrity characterization and simulation, computational electromagnetics, small signal linearized modeling of biased nonlinear components, high voltage power systems, etc. These examples demonstrate that, as far as the linearity and time invariance assumptions hold, all the developed macromodels provide excellent tools for the generation of reduced complexity behavioral models [1].

One of the most common applications of macromodels is to design electronic systems under signal and power integrity constraints. Modern computing systems, such as high-end servers or laptops, smartphones, or tablets, operate at frequencies where the electromagnetic properties of the system, combined with nonideal material properties, can cause serious performance issues and malfunction. A systematic analysis must be performed from the beginning of the design process in order to estimate the transient waveforms of both sensitive signals and supply voltages and to ensure that they meet proper design constraints. For example, to ensure the appropriate operation of the logic devices connected to the power distribution network, the power supply voltage at any location of the system must not drop below a certain safety threshold.

Furthermore, high-speed signals transmission via nonideal interconnects must be immune to dispersion and attenuation caused by losses, reflection from discontinuities, crosstalk, and interference from surrounding interconnects. Even between the signal and the power distribution network, the electromagnetic coupling can occur. Signal/power integrity assessments are best performed through system-level transient simulations using circuit solvers. On the other hand, all the relevant electromagnetic effects, such as the losses, dispersion, etc., are well characterized in the frequency domain through numerical discretization of Maxwell's equations, resulting in tabulated frequency responses (e.g., S-parameters). The macromodeling technique can be seen as a promising approach to tackle this mixed frequency/time problem by providing a bridge between frequency and time domains through fitting rational macromodels that match the scattering responses from the field solvers, and which can be seamlessly

used in circuit solvers. Figure 2.1 shows the generic framework of the macromodeling application to signal/power integrity analysis of a high-speed channel in the printed circuit board.

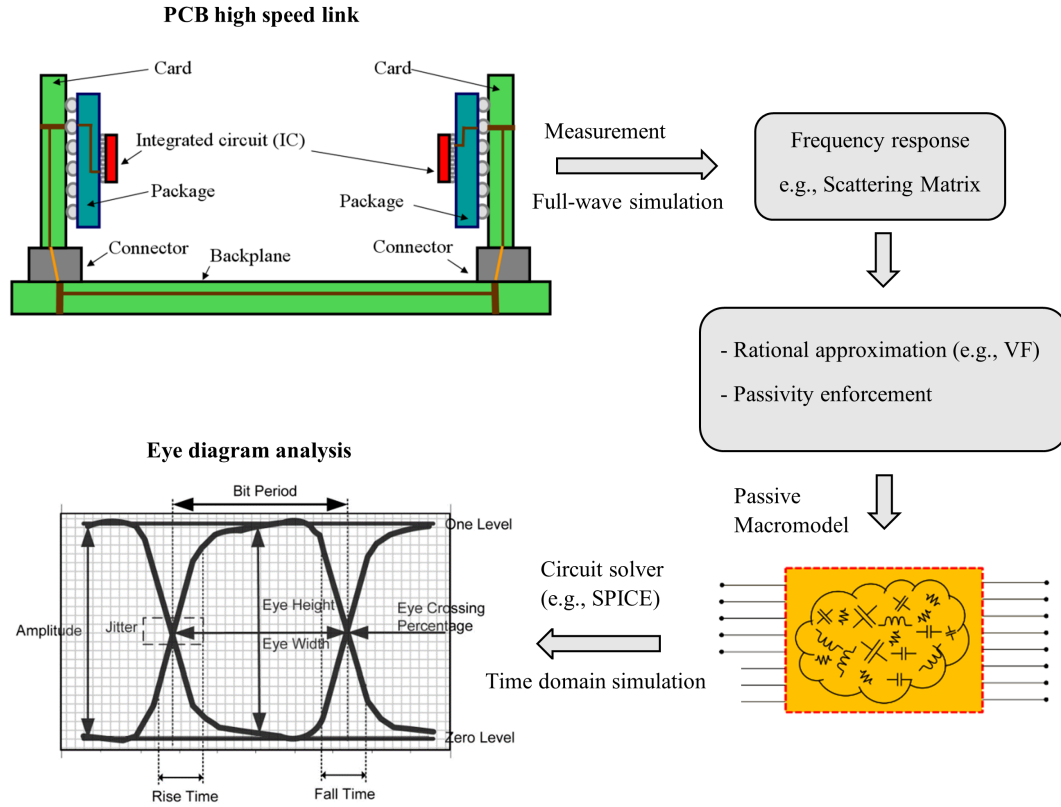


Fig. 2.1 A practical example of the signal/power integrity analysis framework using the macromodel techniques.

2.2 Rational Macromodel

At high frequencies, some geometric structures/devices (e.g., three-dimensional transmission lines, non-uniform transmission lines, vias, packages, on-chip passive components, etc.) show complicated electrical behavior for which there is no analytical model or an equivalent circuit. However, from the network point of view, these devices could be realized in three forms; transfer function (Y , Z , S , etc.), state-space equations, or circuit realization (R , L , C , controlled current/voltage sources, etc.).

It is well known that the port behavior of any linear network could be characterized by its transfer function (e.g., $Y/Z/S$ -parameters). In its simplest form, if it exists, the transfer function $f(x)$ of a general physical network comes in the form of a ratio of two polynomials like (2.1), which can be easily converted to other forms, e.g., pole/residue, etc.

$$f(x) = \frac{P(x)}{Q(x)} = \frac{p_mx^m + p_{m-1}x^{m-1} + \dots + p_1x + p_0}{q_nx^n + q_{n-1}x^{n-1} + \dots + q_1x + 1} \quad (2.1)$$

where, the variable x can be considered as frequency or Laplace variable s (complex frequency), and the degree of the prescribed numerator and denominator polynomials $P(x)$ and $Q(x)$ are m and n , respectively. When $m = n$, the transfer function is referred to as a "proper transfer function".

Given the following requirements, the transfer function (2.1) represents a stable physical system.

- Real-valued coefficients of $P(x)$ and $Q(x)$ (i.e., $p_i, q_i \in \mathbb{R}$)
- All the roots of $Q(x)$ and $P(x)$ are either real or come in complex-conjugate pairs
- All the poles of the transfer function are stable by being located at the LHS of the s -plane
- The highest power of the $P(x)$ and $Q(x)$ can not differ by more than unity

Given the response samples of the device under modeling $f(x_i)$ at discrete points $x_i, i = 1, 2, \dots, M$, the goal is to approximate these data by a closed-form transfer function (2.1). In general, curve fitting techniques may frequently be encountered as a general mathematical tool for finding a closed-form representation for a given data series, which is basically obtained based on the minimization of the error between the approximating function and the data. However, in rational macromodeling, despite providing an analytical approximation for a given data set, the resulting rational function needs to inherit all the important characteristics of the device under modeling, such as stability, passivity, etc.

A primitive idea can be through equating the transfer function with the data at every given frequency point, leading to the following over-determined system of

equations to solve.

$$\begin{aligned} \phi \mathbf{u} &= \mathbf{v} \quad (2.2) \\ \mathbf{u} &= [p_0 \quad \cdots \quad p_N \quad q_1 \quad \cdots \quad q_N]^T \\ \mathbf{v} &= [f(x_1) \quad f(x_2) \quad \cdots \quad f(x_M)]^T \\ \phi &= \begin{bmatrix} 1 & \cdots & x_1^N & -x_1 f(x_1) & \cdots & -x_1^N f(x_1) \\ \vdots & & & \ddots & & \vdots \\ 1 & \cdots & x_M^N & -x_M f(x_M) & \cdots & -x_M^N f(x_M) \end{bmatrix} \end{aligned}$$

It is assumed that coefficient matrix ϕ is not a zero matrix. Typically, the number of equations M is greater than the unknown number $2N + 1$, resulting in an overdetermined system. Therefore, by assuming that the matrix $\phi^T \phi$ is left-invertible, the following equation is the unique solution to the least-squares problem (2.2) [15].

$$\tilde{\mathbf{u}} = (\phi^T \phi)^{-1} \phi^T \mathbf{v} \quad (2.3)$$

It is clear that the above solution is not an exact answer for any choice of \mathbf{u} . The solution defined by (2.3) is the vector that minimizes the sum of squares of the error vector \mathbf{E} defined as the Euclidean norm of the residual as follows [16].

$$\mathbf{E} = \|\phi \tilde{\mathbf{u}} - \mathbf{v}\|^2 \quad (2.4)$$

Such approaches which are based on a direct and primitive formulation can not handle modern applications in general due to the following limitations,

- Not able to handle the wide frequency band (i.e., from DC to a few GHz)
- The matrix ϕ easily becomes ill-conditioned (a.k.a Vandermonde matrix)
- Can not achieve higher-order approximations

In the next section, the well-known *vector fitting* method is introduced which is based on a partial fraction of the pole/residue representation and is able to tackle some aforementioned limitations in the direct formulation.

It is important to remark that, rational macromodeling techniques are solely not suitable for modeling the distributed structures (i.e., when the system under

investigation is electrically large so that the effects of finite propagation speed are visible in its terminal responses) because they theoretically impose an infinite order on the rational function. One may modify the aforementioned rational model form by embedding explicit delay terms when modeling distributed structures [1].

2.3 Vector Fitting

Here, the vector fitting (VF) approach [4] for fitting a rational function (model) to a set of frequency-domain data is described. VF is largely based on solving a series of linear problems with robust numerical properties, and it finds its solution through iterations. The objective is to approximate a frequency response $f(s)$, (generally, a vector; hence, the designation VF) with a rational function

$$f(s) \approx \sum_{n=1}^N \frac{c_n}{s - a_n} + d + sh \quad (2.5)$$

where, c_n , a_n correspond to residues and poles respectively, which are either real or come in complex conjugate pairs, and d , h are real unknowns, s is the Laplace variable s (complex frequency), and N is the order of the rational function.

Consider the approximation of partial fraction form (2.5), the goal is to determine all the unknowns c_n , a_n (for $n = 1, 2, \dots, N$), d , and h in such a way that a least-squares approximation is achieved over the desired frequency range. As can be seen from (2.5), a nonlinear optimization problem is encountered since the unknowns exist in the denominator. Although numerous techniques for the numerical solution of nonlinear least-square and global nonlinear optimization have been proposed [17], their application is not strong enough for the general situation. As a result, various more reliable methods are preferable to brute-force direct nonlinear optimization.

The VF, in particular, solves the original nonlinear problem in two steps as a linear problem with known poles. First, an initial pole set \bar{a}_n is distributed over the desired frequency range to be replaced iteratively with an improved pole set through a scaling process until the desired accuracy is achieved [4]. In addition, an unknown frequency-dependent scale function $\sigma(s)$ is introduced that is precisely fitted to predetermined poles. The new pole set is then obtained from the fitted function, which is used in the second step to fit the scaleless function. The VF fitting

algorithm can be considered as a pole displacement process, meaning that it modifies an initial pole set in an iterative manner to converge very quickly to the dominant poles of the system.

Considering a set of initial poles \bar{a}_n ($n = 1, \dots, N$) and unknown scaling function $\sigma(s)$, the VF first identifies the poles of $f(s)$ by solving in the least-squares sense, the linear problem [4],

$$\sigma(s)f(s) = p(s) \quad (2.6)$$

where

$$\sigma(s) = 1 + \sum_{n=1}^N \frac{\tilde{c}_n}{s - \bar{a}_n} \quad (2.7)$$

$$p(s) = \sum_{n=1}^N \frac{c_n}{s - \bar{a}_n} + d + sh \quad (2.8)$$

where $p(s)$ is generally a vector.

Writing the (2.6) at frequency sample points s_i ($i = 1, \dots, M$) results in the following system of equations which could be solved as a least square problem to find unknowns collected in vector \mathbf{x} .

$$\begin{aligned} \phi \mathbf{x} &= \mathbf{b} \quad (2.9) \\ \mathbf{x} &= [c_1 \quad \dots \quad c_N \quad d \quad h \quad \tilde{c}_1 \quad \dots \quad \tilde{c}_n]^T \\ \mathbf{b} &= [f(s_1) \quad f(s_2) \quad \dots \quad f(s_M)]^T \\ \phi &= \begin{bmatrix} \frac{1}{s_1 - \bar{a}_1} & \dots & \frac{1}{s_1 - \bar{a}_N} & 1 & s_1 & \frac{-f(s_1)}{s_1 - \bar{a}_1} & \dots & \frac{-f(s_1)}{s_1 - \bar{a}_N} \\ \vdots & & \vdots & & & \vdots & & \vdots \\ \frac{1}{s_M - \bar{a}_1} & \dots & \frac{1}{s_M - \bar{a}_N} & 1 & s_M & \frac{-f(s_M)}{s_M - \bar{a}_1} & \dots & \frac{-f(s_M)}{s_M - \bar{a}_N} \end{bmatrix} \end{aligned}$$

Considering the rational function approximation for $f(s)$ from (2.6), the zeros and poles of $f(s)$ could be interpreted as the zeros z_n of approximated $p(s)$ and the

zeros \tilde{z}_n of $\sigma(s)$, respectively, as follows.

$$p(s) = h \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^{N+1} (s - \bar{a}_n)}, \quad \sigma(s) = \frac{\prod_{n=1}^{N+1} (s - \tilde{z}_n)}{\prod_{n=1}^{N+1} (s - \bar{a}_n)} \quad (2.10)$$

$$f(s) = \frac{p(s)}{\sigma(s)} = h \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^{N+1} (s - \tilde{z}_n)} \quad (2.11)$$

Therefore, by calculating the zeros of the approximated function $\sigma(s)$, a new set of poles of $f(s)$ is obtained to be used as starting poles for the next iteration. The iteration continues until the maximum difference between the new and old poles is less than a threshold, as a result of which the algorithm converges to the dominant poles of the system. In this process, if some new poles obtained are unstable (RHS of the s -plane), by inverting the sign of their real parts the stability is guaranteed. More details about calculation of zeros from the partial fractions representation and formation of the system of equations (2.9) for the complex-conjugate poles/residues are given in [4].

The vector fitting pseudocode algorithm is summarized as follows:

- **Step 1:** Use an initial guess of poles $a_n = \bar{a}_n$ and form the matrix equation (2.9)
- **Step 2:** First least square solution to find \mathbf{x}
- **Step 3:** Compute the zeros of scaling function $\sigma(s)$ to be used as the (new) improved poles
- **Step 4:** Repeat the steps 2 and 3 until poles converge!
- **Step 5:** For the final poles run the second round of least square to find the residues. The results are poles and residues of the $f(s)$

Given the poles $\{a_1, \dots, a_N\}$ and the residues $\{c_1, \dots, c_N, d\}$, the state-space realization of the model simply writes:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned}\tag{2.12}$$

$$\mathbf{A} = \begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_N \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T, \mathbf{C} = \begin{bmatrix} c_1 & \dots & c_N \end{bmatrix}^T, \mathbf{D} = d$$

where, $\mathbf{u}(t)$ and $\mathbf{y}(t)$ are the LTI system input and output, respectively, and $\mathbf{x}(t)$ collects the state variables.

2.3.1 Example: Microstrip Line

A single trace PCB, as shown in Figure 2.2a with a length of 10 cm is considered in this example. The width of the strip and substrate height is about 1.55 mm, 0.8 mm, respectively. The applied substrate is FR4 with a relative permittivity of $\epsilon_r = 4.3$. The S-parameters of this structure ranged between 0 and 5 GHz are extracted by CST microwave studio time-domain simulation.

The results of MATLAB and the CST built-in VF algorithms for this example are compared in Table 2.1. The CST VF algorithm is not available and only some parameters including the number and increasing rate and the maximum number of initial poles, and the desired accuracy is adjustable. The toolbox is first run with the defined initial pole number. If the desired accuracy of the approximation is not met, the VF algorithm is repeated with an increased number of poles until the desired accuracy or the maximum number of poles is reached. Therefore, it can not be claimed that the model obtained through this scenario is optimal. Here we use the results of the CST VF toolbox only as a reference for evaluating the VF algorithm implemented in MATLAB. Figure 2.3 shows a good fit of S-parameters amplitude of microstrip using MATLAB VF with $N = 20$ uniformly distributed complex conjugate poles after 1 iteration.

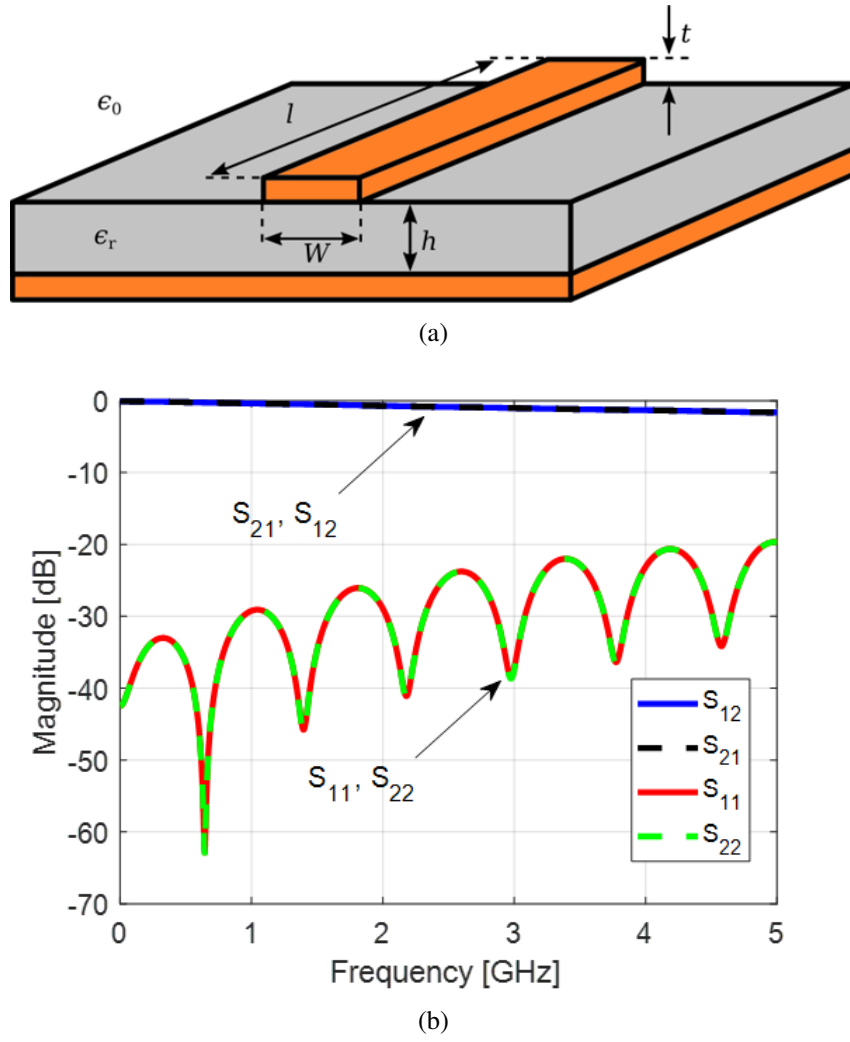


Fig. 2.2 (a) Microstrip structure, and (b) amplitude of microstrip S-parameters obtained from simulation in CST.

Table 2.1 VF algorithm performance for microstrip example.

	MATLAB VF Toolbox	CST VF Toolbox
Fitting error (MSE)	3.6379e-06	3.148827e-06
Pole number	20	20
Computational cost (sec)	0.11	0.14

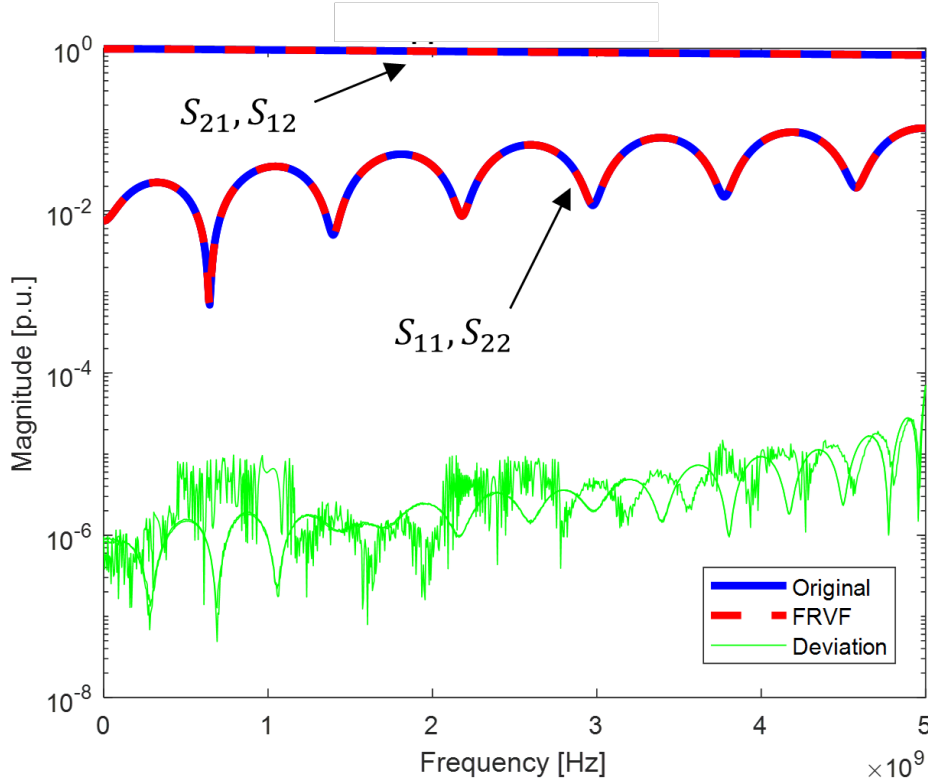


Fig. 2.3 Resulting S-parameters amplitude fit (red dashed line) for microstrip example after 1 iteration of the VF algorithm with $N = 20$ complex conjugate poles.

2.4 Synthesis of Equivalent Circuit

The general flow of the macromodeling based on data from full-wave solvers is shown in Figure 2.4. After applying the fitting algorithm and generating macromodels in a pole-residue form, it is necessary to synthesize the equivalent netlist to be used in circuit solvers for transient simulations. Various methods for equivalent circuit synthesis of the S , Y , or Z transfer functions are researched [18, 19]. Here, we have adopted the *Direct State-Space Synthesis* method [1].

Also, the ADS is used as an external circuit solver to evaluate the generated equivalent netlist (for example see Figure 2.8a). It should be noted that CST also generates the equivalent SPICE-compatible netlist, which we use it here as a reference to validate our synthesis method. To do this, the synthesized netlist of the microstrip example is imported to ADS and simulated again to regenerate the S-parameters as

shown in Figure 2.5. It can be seen that the netlist is capable of regenerating the original S-parameters in Figure 2.2.

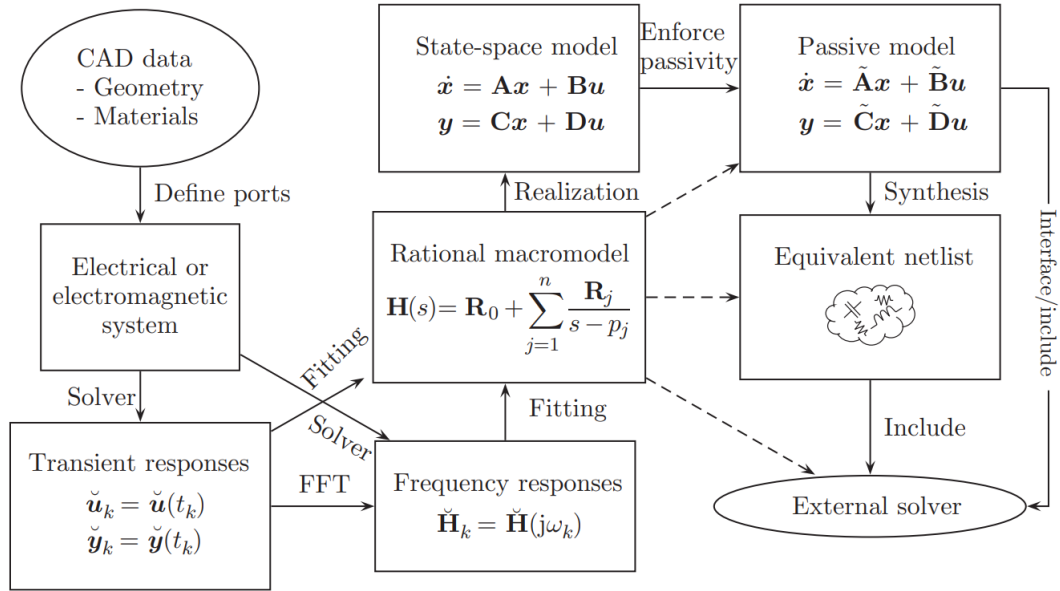


Fig. 2.4 Macromodeling flow based on data from full-wave solvers (given from [1]).

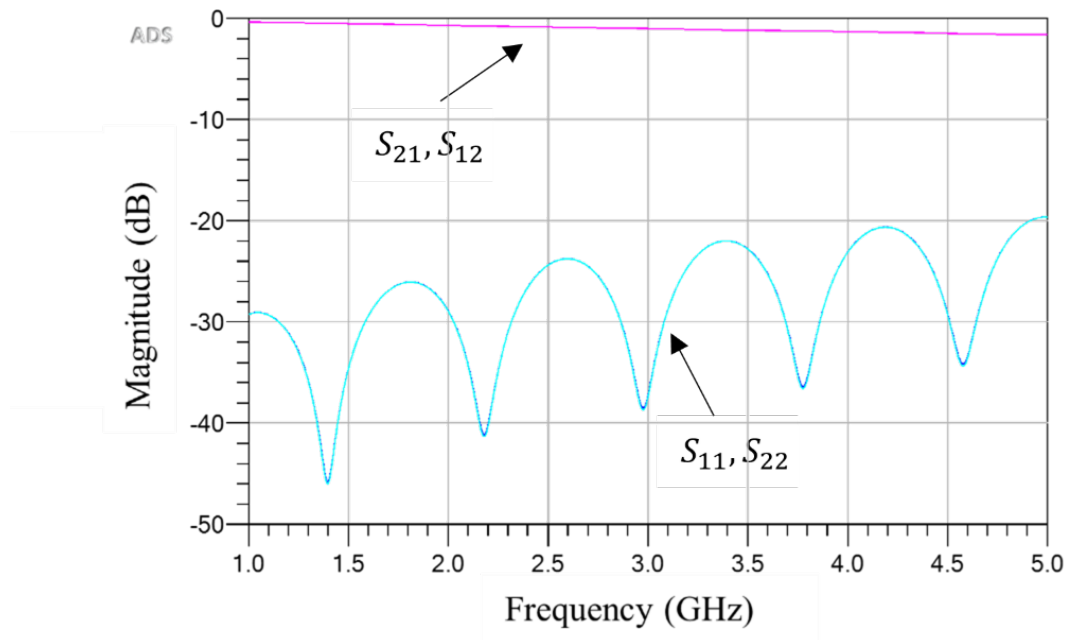


Fig. 2.5 Regenerated S-parameters of the synthesized netlist in ADS for microstrip example.

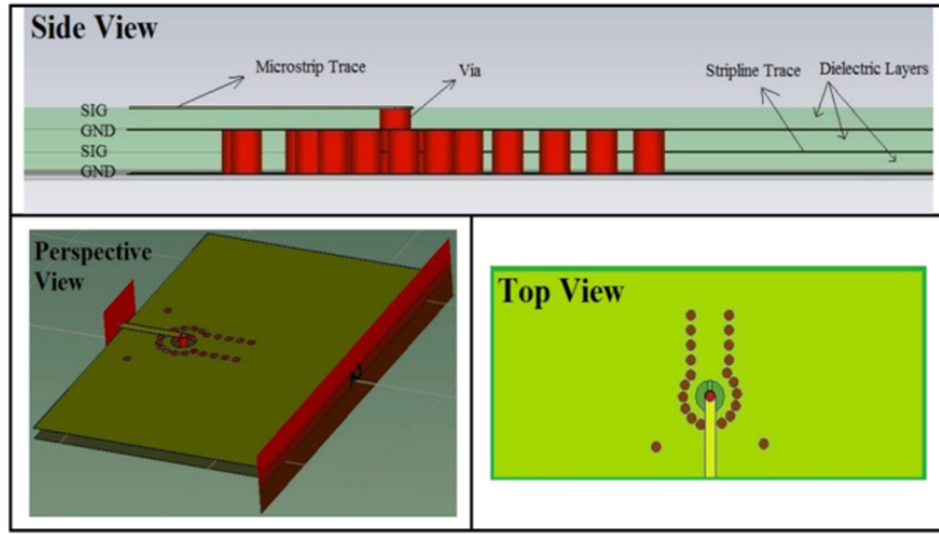
Table 2.2 VF algorithm performance for multilayer PCB example.

	MATLAB VF Toolbox	CST VF Toolbox
Fitting error (MSE)	1.8245e-04	8.06757e-04
Pole number	10	20
Computational cost (sec)	0.022618	0.03415

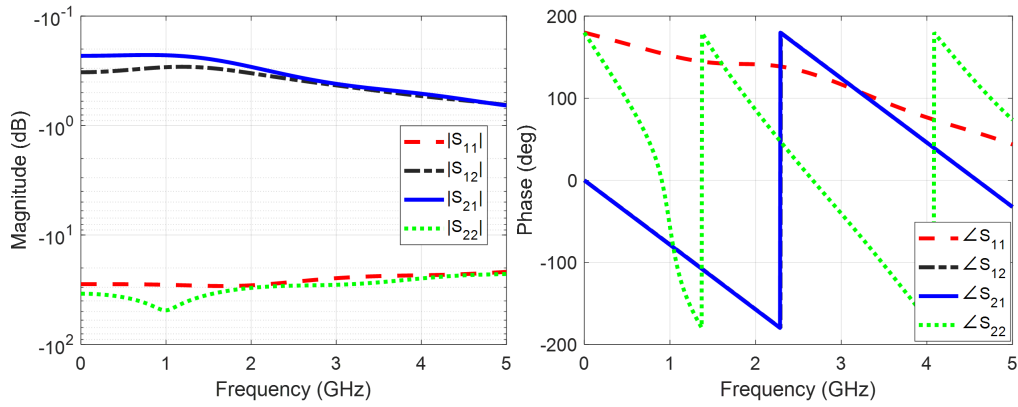
2.4.1 Example: Multilayer PCB Structure

The second example is a multilayer PCB structure including microstrip to stripline transition shown in Figure 2.6a. In this structure, a microstrip trace with a width and length of 1.18 mm, and 8.49 mm, respectively, in the signal layer is connected to a stripline trace in the third layer with a width and length of 0.47 mm, and 22.64 mm, respectively using a via. The copper planes in layers 2 and 4 are regarded as ground. Also, the signal via is surrounded by several ground vias that provide local electrical continuity between two ground planes and a shielding effect that limits the coupling of energy with other parts of the PCB. An FR4 substrate with relative permittivity, loss tangent, and height of 4.3, 0.02, and 0.6 mm, is used for this example. The structure is terminated matched load (50 Ohms impedance).

By this complicated example, we show that the netlist generated by CST is not able to perfectly regenerate the original S-parameters obtained from full-wave simulation. To this end, given the S-parameters of multilayer PCB shown in Figure 2.6, both the CST and MATLAB VF toolbox are used to build the rational macromodels with parameters collected in Table 2.2. Figure 2.7 shows the perfect fit of MATLAB VF rational macromodel. It should be noted that according to Table 2.2, the macromodel obtained from MATLAB VF Toolbox is efficient with respect to the CST VF Toolbox in terms of the accuracy and the model's order.



(a)



(b)

Fig. 2.6 (a) Multilayer PCB structure, and (b) the S-parameters obtained from simulation in CST.

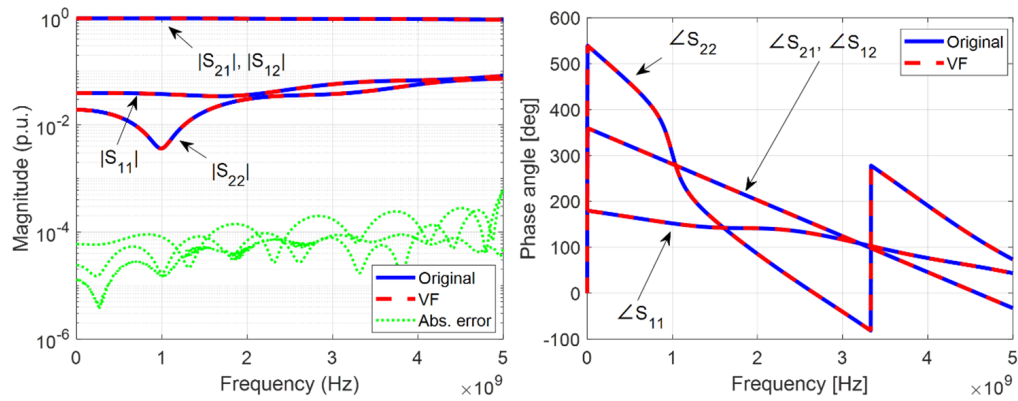


Fig. 2.7 Fitting the S-parameters of the multilayer PCB example (blue line) using the MATLAB VF Toolbox (red dashed line) with $N = 10$ complex conjugate poles after 1 iteration.

Then, the corresponding synthesized netlists are simulated in ADS to regenerate the S-parameters as shown in Figure 2.8. The results show that the CST netlist (black dotted line) has failed to regenerate the original magnitude responses S_{11} and S_{22} (e.g., the notch frequency of $|S_{22}|$ is about 300 MHz shifted from the original response) and phase response $\angle S_{11}$, as well. But our netlist (i.e., *VF Netlist* red dashed line), generated by *Direct State-Space Synthesis* [1], is able to regenerate the original S-parameters responses very well.

2.5 Passivity Enforcement

In the macromodeling procedure, the passivity of the model may be lost for several reasons [1]. When working with measurement data, the presence of noise or inappropriate calibration, etc. may cause local passivity [1]. In cases where we use full-wave simulation data, it should also be noted that the answers are obtained through a discretization process that is never accurate, and in fact, the inevitable error of numerical approximation may be the reason why the model is non-passive. Improper settings of the full-wave solver and non-physical assumptions on the materials used can also be the cause of this problem.

In addition, although the fitting algorithm can produce very accurate and stable approximations even for complex structures, these models may not be passive. The passivity of an LTI system could be simply assessed by checking the singular values of the S transfer matrix in the desired frequency range, that no value is greater than one. For example, Figure 2.9 shows the singular values of the matrix S for the multilayer PCB example before and after applying the VF algorithm. It can be seen that the macromodel approximated is not passive in the range $[0, 560]$ MHz as one of the corresponding singular values has exceeded the 1 in this range.

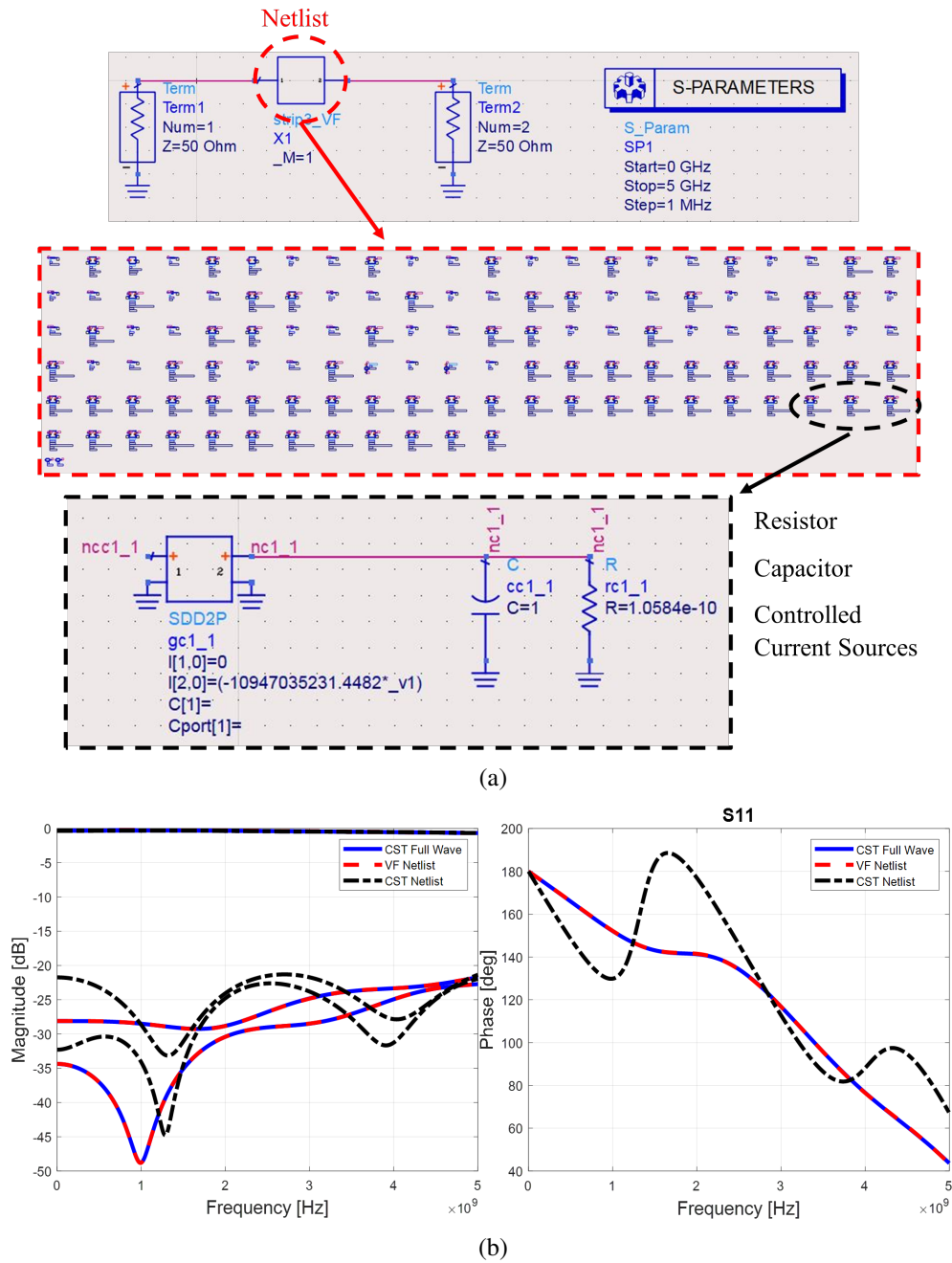


Fig. 2.8 (a) Netlist of the multilayer PCB example and (b) the S-parameters of the netlists generated by CST and direct state-space synthesis (VF Netlist) methods obtained from simulation in ADS.

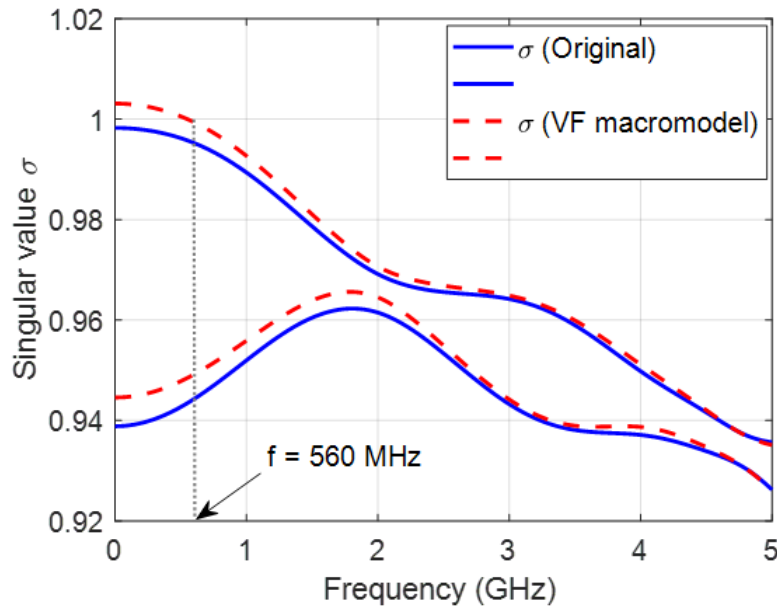


Fig. 2.9 Full frequency sweep of the singular values of the multilayer PCB scattering matrix before and after fitting.

2.6 Non-iterative Proposed Rational Macromodel based on Data Integration

In this section, a mathematical method for developing a rational-based transfer function model for practical applications is introduced; addressing the challenges of low complexity and less sensitivity to noisy data. This method is developed based on the integration of the original simulated or measurement data at several specified intervals to decrease data losses and increase the accuracy of the final outcome. For the number of integration intervals, a number of equations are obtained. The result is a system of linear equations. Then, using the Least Square Method (LSM), the required values, including poles and residues of the rational form of the model are determined. To ensure the stability condition of the final response, a closed-loop model is attributed to the understudying system. This goal will be met by defining a stability controller coefficient for the closed-loop model. Several practical examples are provided to evaluate the performance of the proposed method.

2.6.1 Formulation

As discussed in section 2.2, a direct formulation (2.2) of the rational approximation is affected by several issues. First, the high power of frequency $s^n (s = j\omega)$ in (2.2) will result in Vandermonde matrices in the least-squares problem to be solved, which are known to be ill-conditioned even for relatively modest values of N . Second, in this method, information between two adjacent samples is not used. If the goal is to use all the information, the number of unknowns, being the degree of the numerator and the denominator polynomials of (2.1), increases dramatically.

To tackle the above issues, multiplying both side of (2.1) by its denominator $Q(x)$, and integrating both sides over an interval $[x_i, x_f]$ leads to:

$$\int_{x_{min}}^{x_{max}} \{p_0 + p_1x - q_1f(x)x + p_2x^2\} dx + \int_{x_{min}}^{x_{max}} \{-q_2f(x)x^2 + \dots + p_nx^N - q_nf(x)x^N\} dx = \int_{x_{min}}^{x_{max}} f(x) dx \quad (2.13)$$

For a given desired data, the integration result for both the left and right-hand sides of the above equation could be calculated. The unknown numbers are $2N + 1$. Hence, at least $2N + 1$ independent equations are required. In practical application, it is assumed that the desired data are available from frequency interval $\omega \in [\omega_{min}, \omega_{max}]$. As mentioned before, x could be considered as a complex frequency $s = j\omega$. Hence, x is varying over the interval $x \in [x_{min}, x_{max}]$, where x_{min} and x_{max} correspond to a minimum and maximum frequency, respectively. By dividing the distance $x_{min} \leq x \leq x_{max}$ to M equal segments as the integral intervals $x_m \leq x \leq x_{m+1}$, $m = 1, 2, \dots, M$, the required number of equations will be obtained. In physical problems, the frequency is in the range of GHz. Hence, to avoid computational complexity, before incoming the process, the variable x can be normalized to its maximum value.

By specifying the upper and lower limits of the integration interval, only the coefficient's q_i, p_i remains unknown. As a result, a linear system of M equations and

$2N + 1$ unknown is made. This equation system and its solution are as follows.

$$\boldsymbol{\psi} \mathbf{u} = \mathbf{w} \quad (2.14)$$

$$\mathbf{u} = \begin{bmatrix} p_0 & \cdots & p_N & q_1 & \cdots & q_N \end{bmatrix}^T \quad (2.15)$$

$$\mathbf{w} = \begin{bmatrix} \int_{x_1}^{x_2} f(x) dx & \cdots & \int_{x_M}^{x_{M+1}} f(x) dx \end{bmatrix}^T \quad (2.16)$$

$$\boldsymbol{\psi} = \begin{bmatrix} \int_{x_1}^{x_2} dx & \cdots & \int_{x_1}^{x_2} x^N dx & -\int_{x_1}^{x_2} x f(x) dx & \cdots & -\int_{x_1}^{x_2} x^N f(x) dx \\ \vdots & & & \ddots & & \vdots \\ \int_{x_M}^{x_{M+1}} dx & \cdots & \int_{x_1}^{x_2} x^N dx & -\int_{x_1}^{x_2} x f(x) dx & \cdots & -\int_{x_M}^{x_{M+1}} x^N f(x) dx \end{bmatrix} \quad (2.17)$$

$$\tilde{\mathbf{u}} = \left(\boldsymbol{\psi}^T \boldsymbol{\psi} \right)^{-1} \boldsymbol{\psi}^T \mathbf{w} \quad (2.18)$$

where $\boldsymbol{\psi}$ is $M \times (2N + 1)$ coefficient matrix, \mathbf{u} is $(2N + 1) \times 1$ column vectors, in which holds the unknowns, and \mathbf{w} is $M \times 1$ column vectors that include the desired data.

It should be noted that the proposed linearization (2.13) (which is essentially an improved version of Levy's linearization [1] with the additional use of integral operators) inherently acts as a weighting function, hence (2.18) is the solution of a weighted LS problem, with a weighting matrix defined by the magnitude of the model denominator. In general, this produces a well-known systematic bias problem that Vector Fitting does not have [1]. In some practical cases, using equation (2.18) is not a good solution, especially when $\boldsymbol{\psi}$ is an ill-conditioned matrix, and it may cause a low accuracy in the final answer. In these cases, using a modified QR factorization technique leads to an increase in computational efficiency, where \mathbf{P} is the permutation matrix [1].

$$\tilde{\mathbf{u}} = \mathbf{P} \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{w} \quad (2.19)$$

$$\boldsymbol{\psi} \mathbf{P} = \mathbf{Q} \mathbf{R} \quad (2.20)$$

It could be seen that the integral responses in (2.14) are independent of variable x . This ensures that the proposed method is less affected by any unwanted noise or disturbance. In other words, the integral operator is resistant to noise. In former research, they use only samples of the available data. This means that, in these methods, all observable output of the system is not used properly and there is a data

loss problem, while in the proposed technique, all available data are used in the integration process, and there is not any data loss.

According to equation (2.14), three conditions should be met. First, available data should be absolutely integrable over any interval. Second, available data should be of bounded variation in any given bounded interval. Third, available data should have a finite number of discontinuities in any given bounded interval, and the discontinuities cannot be infinite. In summary, the Dirichlet condition should be met as follows [20].

$$\int_x |f(x)| dx < \infty \quad (2.21)$$

$$\int_x |f(x)x^n| dx < \infty; n \text{ is integer} \quad (2.22)$$

The computational complexity of (2.14) is dependent on the value of M . The Nyquist theorem can be helpful to determine the sampling rate Δ . For arbitrary available data, the following equation can be used to the first approximation of M [21].

$$\Delta \leq \frac{|x_{max} - x_{min}|}{4N} \quad (2.23)$$

$$M \geq \frac{8N}{|x_{max} - x_{min}|} \quad (2.24)$$

In other words, the number of integration intervals is considered equal to the number of samples. It should be noted that the above equations are obtained, assuming that the integration intervals have equal lengths. In some cases, intervals of equal length may not produce acceptable results. For these cases, the number of integral intervals of a range of x that are more important could be increased. Correspondingly, for regions of x where the original data is less important, less number of integral intervals could be considered. For example, if the original data is the frequency response of a bandpass filter, the number of integral intervals in the passband range will be considered greater than the number of integral intervals in the stopband range.

After determining the unknown coefficients q_i , p_i using (2.14), poles, and zeros of equation (2.1) should be determined. The poles and zeros are eigenvalues of the

following matrixes [22].

$$\mathbf{T_P} = \begin{bmatrix} -\frac{p_{N-1}}{p_N} & -\frac{p_{N-2}}{p_N} & \dots & -\frac{p_0}{p_N} \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix}_{N \times N} \quad (2.25)$$

$$\mathbf{T_Q} = \begin{bmatrix} -\frac{q_{N-1}}{q_N} & -\frac{q_{N-2}}{q_N} & \dots & -\frac{1}{q_N} \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix}_{N \times N} \quad (2.26)$$

By specifying the poles and zeros of the system, available data could be expressed in poles-residues form as (2.5). After specifying the poles and residue, the rational form can be easily rescaled to its normal case.

In some cases, the obtained poles may be placed in an unstable region. In the following, a simple technique will be introduced to overcome this problem. The general form of a closed-loop system with forward transfer function $G(x)$ and feedback transfer function $F(x)$ is presented in Figure 2.10. The transfer function of the closed-loop system is as follows.

$$TF(x) = \frac{G(x)}{1 + G(x)F(x)} \quad (2.27)$$

For simplicity, it is assumed that the feedback transfer function is equal to one and the forward transfer function could be expressed as a ratio of two polynomials. This system is considered a macromodel of the understudying system as follows.

$$F(x) = 1, \quad G(x) = k \frac{B(x)}{A(x)} \rightarrow TF(x) = \frac{k B(x)}{A(x) + k B(x)} \quad (2.28)$$

Therein, parameter k is a stability controller coefficient, using which the stability of the model could be controlled. By comparing equation (2.28) with (2.1), we have:

$$f(x) = \frac{kB(x)}{A(x) + kB(x)} = \frac{kB(x)}{C(x)} = k \frac{b_N x^N + b_{N-1} x^{N-1} + \dots + b_1 x + b_0}{c_N x^N + c_{N-1} x^{N-1} + \dots + c_1 x + c_0} \quad (2.29)$$

$$\begin{cases} c_n = q_n + k p_n & n = 1, 2, \dots, N \\ c_0 = 1 + k p_0 \end{cases}$$

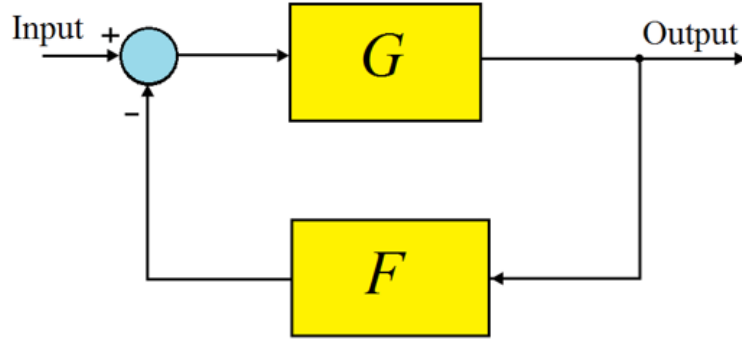


Fig. 2.10 Block diagram of a closed-loop system.

In this situation, the proposed procedure is applied to a new case. In other words, equation (2.14) should be rewritten for a new case.

$$\psi = \begin{bmatrix} \psi_b & \psi_a \end{bmatrix} \quad (2.30)$$

$$\psi_b = k \begin{bmatrix} \int_{x_1}^{x_2} h(x) dx & \cdots & \int_{x_1}^{x_2} h(x) x^N dx \\ \vdots & \ddots & \vdots \\ \int_{x_M}^{x_{M+1}} h(x) dx & \cdots & \int_{x_M}^{x_{M+1}} h(x) x^N dx \end{bmatrix} \quad (2.31)$$

$$\psi_a = \begin{bmatrix} -\int_{x_1}^{x_2} x f(x) dx & \cdots & -\int_{x_1}^{x_2} x^N f(x) dx \\ \vdots & \ddots & \vdots \\ -\int_{x_M}^{x_{M+1}} x f(x) dx & \cdots & -\int_{x_M}^{x_{M+1}} x^N f(x) dx \end{bmatrix} \quad (2.32)$$

$$h(x) = 1 - f(x) \quad (2.33)$$

As a result, by changing the parameter k , the stability condition could be met. In other words, unstable poles could be moved toward the stable region. Another solution is to plot the root locus of the system and determine the acceptable value of k that guarantees the stability condition.

Passivity can be achieved through the conventional two-step methods introduced in [1]. In this way, the first step comprises approximation with stable poles. Then, using the method of iterative perturbation in the residues, a passive macromodel could be achieved. Also, the proposed method could be easily extended for Multi Input Multi Output systems (MIMO) using the introduced procedure in chapter 8 of [1].

Table 2.3 Poles and residues of the TF of theoretical example

Poles(GHz)	Residues (GHz)
-0.6132±j3.4551	-0.9877±j0.0809
-0.3940±j7.3758	-0.2067±j0.0131
-0.0880±j14.3024	-0.1382±j0.0145
-0.4097±j17.7864	-0.1182±j0.0166
-0.2991±j28.4622	-0.2426±j0.0145
-0.6447±j35.2669	-0.4043±j0.0297
-1.0135±j37.9655	-0.6787±j0.1465
-0.5711±j57.4748	-0.2626±j0.1037
$r_0=0.1$	

2.6.2 Practical Examples

In this section, the performance of the proposed rational macromodel will be demonstrated using several examples. It should be noted that the purpose of this validation is not to show the overall superiority of the proposed method over the VF algorithm. Here, the relaxed VF algorithm "*vectfit3.m*" is used as a known method to compare the results [5].

Example: Noisy data

The first test case is a theoretical example to illustrate the performance of the proposed method for noise-infected data in frequencies ranging from 0 to 10 GHz. This example includes a synthetic transfer function with 16 poles described in Table 2.3 [23]. First, the synthetic transfer function is considered to be noise-free. Figure 2.11 shows the obtained results of the proposed method with $N=12$ and the VF algorithm with $N=16$. The performance of the proposed method is clearly better than the VF.

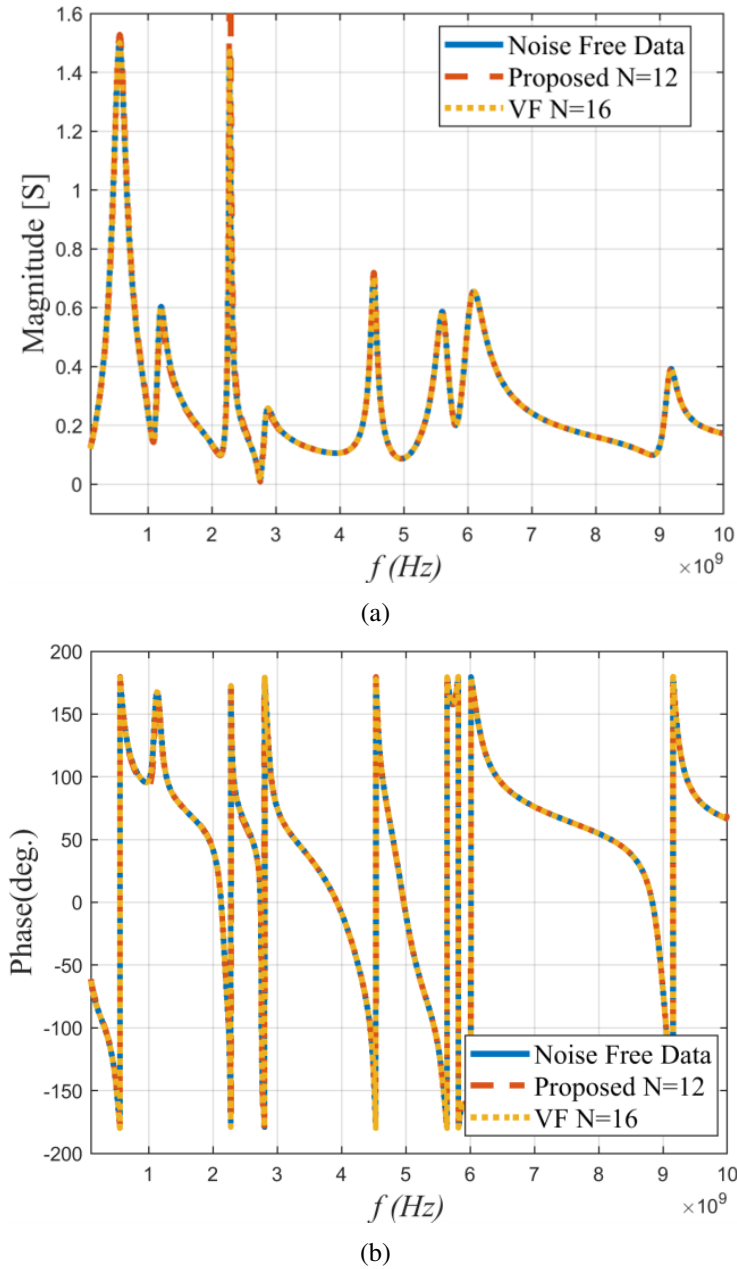
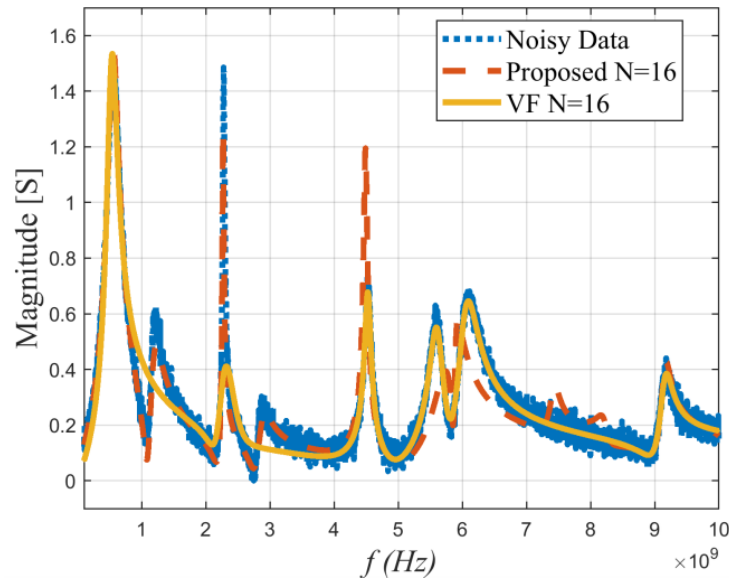


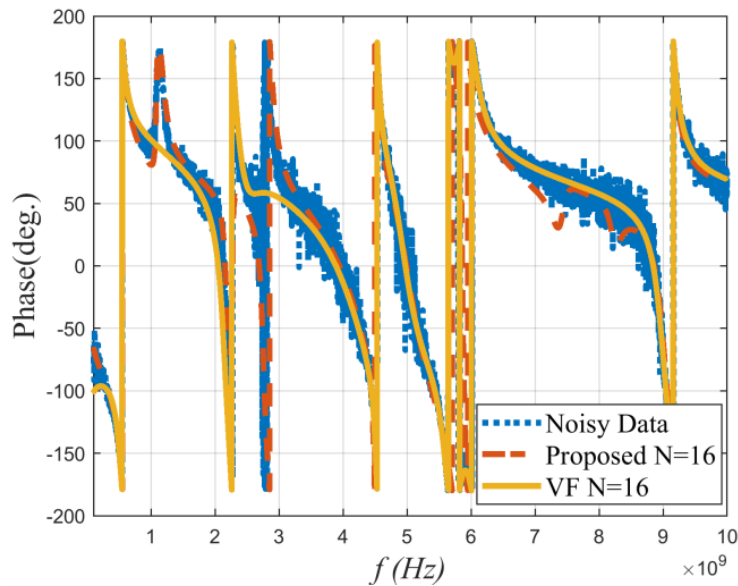
Fig. 2.11 (a) Magnitude, and (b) phase of synthesized transfer function of theoretical example in the absence of noise.

Now, consider the same transfer function in the presence of noise. Both real and imaginary parts of the transfer function are infected by white Gaussian noise. The noise level considered for this example is set to 20 dB signal-to-noise ratios (SNRs). The synthesized results of both VF and the proposed method with noisy

data are depicted in Figure 2.12. Due to numerous fluctuations, it is not possible to compare the results from the figures correctly. For this reason, the diagram of the Mean Squares Error (MSE) in dB for both methods is also shown. It is observed that the proposed method has a lower MSE than VF, for about 2.8 dB average. In other words, the proposed method has more immunity with respect to noise. It should be noted that the noisy transfer function does not show smooth behavior and includes several sudden jumps versus frequency.



(a)



(b)

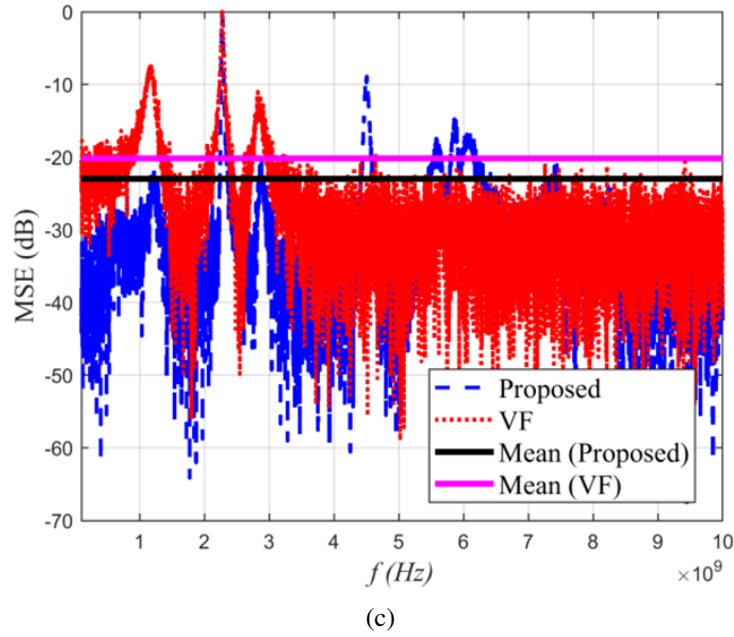


Fig. 2.12 (a) Magnitude, (b) phase, and (c) mean square error of synthesized transfer function of theoretical example with noise.

Example: Coupled Microstrip Line

As the second example, a coupled microstrip line using TLY062 substrate with relative permittivity of 2.2, a thickness of 1.56 mm, and loss tangent 0.009 is considered [24]. The length and width of the board are 50 mm. The culprit and victim strip width are set to 4.8 mm, and the distance between the two traces is about 0.5 mm. The measured far-end crosstalk is regarded as original data [24]. The fabricated of the understudying structure is shown in Figure 2.13 [24].

The magnitude and phase of rational approximation using the proposed method with $N = 9, 10$, and vector fitting algorithm with $N = 14$ poles are shown in Figure 2.14. It can be seen that the proposed method with $N = 9$ shows a small deviation in magnitude in low frequencies. However, by increasing the pole number to $N = 10$, the synthesized error is decreased. Although the accuracy of VF and the proposed method for $N = 10$ is almost acceptable, the proposed method with a smaller number of poles shows better performance.

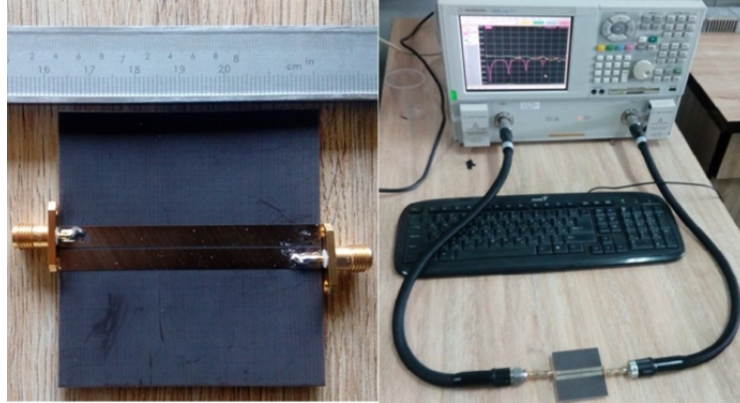


Fig. 2.13 The fabricated structure of coupled microstrip line [25].

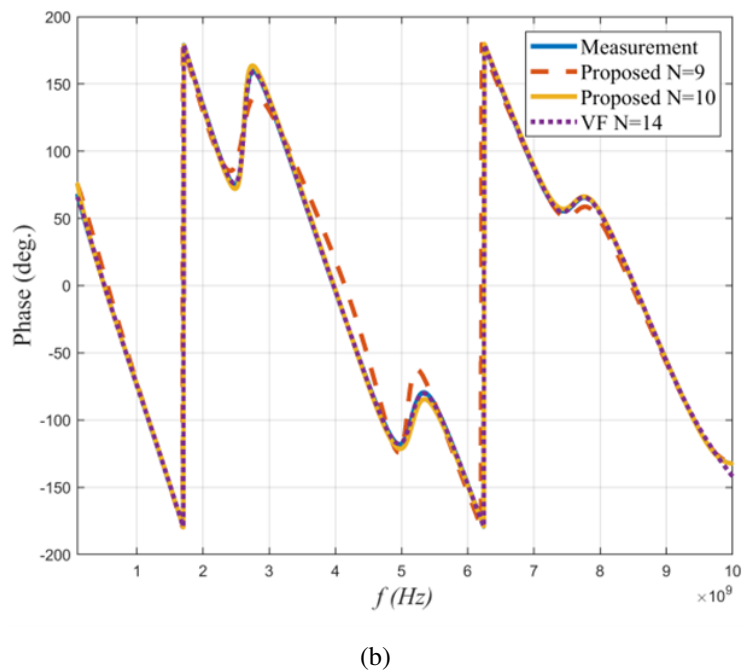
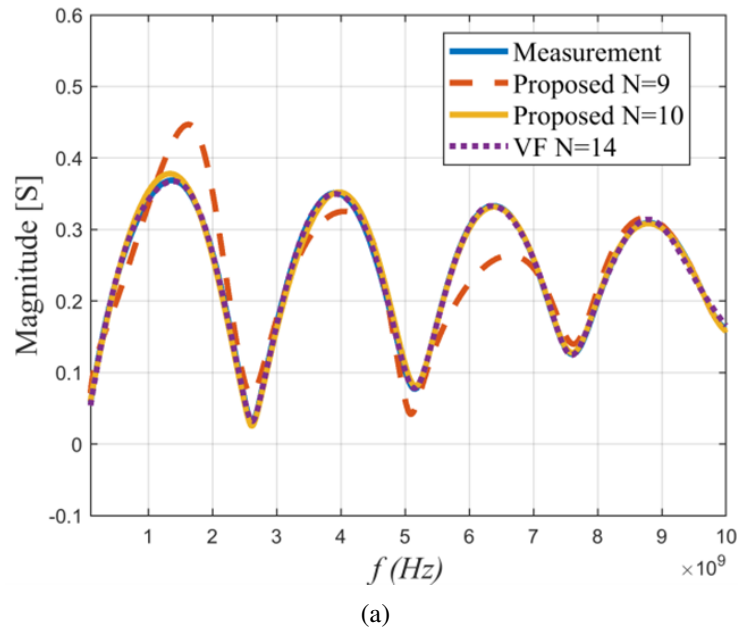


Fig. 2.14 (a) Magnitude, and (b) phase of the synthesized far-end crosstalk of coupled microstrip line.

Table 2.4 Comparison of the condition number of the Vandermonde and the proposed coefficients matrix for all examples.

	N	Vandermonde matrix	Proposed method
Noisy data	16	3.4643e+163	6.1996e+13
Coupled microstrip line	10	9.6081e+103	3.4439e+09

The condition numbers of the Vandermonde (2.2), and the proposed coefficient matrix (2.18) for all examples are reported in Table 2.4. It can be seen that the condition number of the proposed coefficient matrix is very small in comparison to the Vandermonde matrix in all application examples. Large condition numbers mean numerical difficulties in the computation of the poles and residues of the final model [1]. So, Table 2.4 shows that the proposed method has created an extreme improvement in the condition number of the coefficient matrix compared to the LSM method. This has led to a significant reduction in the computational error of the introduced method.

2.7 Parameterized Macromodel

Due to the complexity of modern electronic systems, we need fully automated design and optimization workflows. In this scenario, the physical layout of a given structure or subsystem is finalized as a result of lengthy optimization processes that select the best candidate among many possible options while maximizing some cost functions related to system performance. This optimization is usually achieved by repeating system-level simulations for different combinations of design variables; Therefore, the availability of a model of each component is necessary for this process. These models should be available as needed for any configuration of design variables, including geometric parameters (such as the width and spacing of the transmission lines on PCBs, substrate height, etc.) or material parameters (such as electrical conductivity, permittivity, etc.). A model with such features is called a *parameterized macromodel*.

According to the standard approach (see Figures 2.1 and 2.4), a macromodel is first constructed using full-wave simulation (or measurement) data to extract the port response (e.g., frequency-dependent scattering matrix) of the structure. Then,

by applying the appropriate algorithms, we obtain a rational approximation for the response so that the resulting macromodel is converted to a format compatible with circuit solvers (e.g., SPICE). However, when the structure of interest is dependent on one or more design variables, the entire process of creating the macromodel (i.e., full-wave simulations, rational approximation, etc.) must be repeated for each different combination of the parameters, making this process impractical.

In previous sections, we discussed the rational macromodels in which the model transfer function had only one independent variable (i.e., the complex frequency s). But in practice, in addition to frequency, other geometrical or material parameters are also involved in the design process which should be determined based on performance optimization. Standard macromodeling approaches can be generalized in such cases to include a closed-form dependence on a few external parameters, λ_i ($i = 1, \dots, l$), (e.g., geometrical or material parameters) into the model transfer function, resulting in parameterized macromodel as follows.

$$H(s; \boldsymbol{\lambda}) = H(s; \lambda_1, \dots, \lambda_l) \quad (2.34)$$

In the simplest form, the rational approximation (2.1) discussed before could be simply rewritten as parametric form (2.35), in which the numerator and denominator coefficients (or poles/residues equivalently) are considered as a function of external parameters.

$$H(s; \boldsymbol{\lambda}) \approx \frac{P(s; \boldsymbol{\lambda})}{Q(s; \boldsymbol{\lambda})} = \frac{p_m(\boldsymbol{\lambda})s^m + p_{m-1}(\boldsymbol{\lambda})s^{m-1} + \dots + p_1(\boldsymbol{\lambda})s + p_0(\boldsymbol{\lambda})}{q_n(\boldsymbol{\lambda})s^n + q_{n-1}(\boldsymbol{\lambda})s^{n-1} + \dots + q_1(\boldsymbol{\lambda})s + 1} \quad (2.35)$$

Constructing the parameterized macromodels in a functional form, for example, the rational closed-form (2.35), is quite challenging, mainly due to the stability of the polynomial coefficients (or poles/residues) in the whole desired frequency and parameter ranges, because these formulations will be severely degraded when dependent on external parameters. Various parameterized model forms have been proposed, with related fitting algorithms for transmission line structures, connectors, vias, passive elements, etc. [25–28].

Most of these methods are efficient in applications with a limited number of external parameters (e.g., $|\boldsymbol{\lambda}| = 1, 2$) and show poor performance in high dimensional cases [1]. In particular, the complexity of these methods directly depends on the parameter space dimensionality. To overcome this dependence, a parameterized

macromodel based on radial basis functions is proposed, which is successful in applications with ten external parameters (i.e., $|\boldsymbol{\lambda}| = 10$) [29].

Different from the closed-form models, Artificial Neural Networks (ANNs) have been recently demonstrated as efficient parameterized macromodels which are able to learn the port response behavior with respect to external parameter changes and can be further used in EM design optimization where the geometrical variables are repetitively changed [30]. The neuro-transfer function (neuro-TF) method [31] is an advanced ANN parameterized modeling approach that addresses the situation where a suitable equivalent circuit model is not available. Transfer functions are employed as prior knowledge to develop the neuro-TF parameterized macromodel.

Let $\boldsymbol{\lambda}$, \mathbf{w} , and \tilde{H} represent the input, weighting, and output of the ANN model, respectively. Defining the vector \mathbf{p} containing all the transfer function parameters (e.g., rational function coefficients, or poles/residues), i.e., $\mathbf{p} = [p_0 p_1 \dots p_m q_1 \dots q_n]$, they are neural network functions of $\boldsymbol{\lambda}$ and \mathbf{w} . Hence, the output of the neuro-TF model is generally formulated as

$$H(s; \boldsymbol{\lambda}) \approx \tilde{H}(s; \mathbf{p}(\boldsymbol{\lambda}, \mathbf{w})) \quad (2.36)$$

Given a set of training samples $\{\boldsymbol{\lambda}_k\}_{k=1}^L$ and $\{H_{k,j}(s_j; \boldsymbol{\lambda}_k)\}_{k,j=1}^{L,F}$, where L and F are the number of training samples (various geometrical configurations) and frequency points, respectively, and $H_{k,j}$ denotes the response samples from EM solvers of the k th sample at the j th frequency. First, before neuro-TF model training, the vector fitting is performed to generate the transfer function parameters \mathbf{p} versus geometrical parameters $\boldsymbol{\lambda}$ for all training samples. Then, the ANN model is trained by optimizing the weighting parameters to get the minimum training error. The interested reader is referred to [31] for a detailed description of the two-stage neuro-TF model training.

Although the parameterized macromodeling is a fundamental enabling factor for the design process, the design optimization is usually performed by repeating system-level simulations to minimize a cost function. In such a repetitive optimization scheme, the convergence and the overall runtime depend on several factors like the optimizer algorithm type used or the nonlinearity relation of the transfer function to the design parameters variation.

Feature-based optimization approaches [32], using the fact that the nonlinear behavior of frequency response characteristics (e.g., bandwidth and central frequency

of a microwave filter, etc.) relative to the change of design parameters is much less than nonlinearity of frequency response samples, allow improving the convergence and runtime of the repetitive optimizer. However, the application is limited because defining the cost function based on response features depends on each problem at hand, and sometimes it is difficult to accurately extract the response features.

Considering these issues, in Chapter 4, we propose a completely different approach called *Inverse Modeling* to design optimization of microwave components, which differs from conventional repetitive optimizers, is very efficient, and does not require any iteration. Given the desired response samples (e.g., frequency dependent S-parameter samples) of a microwave component, the goal is to build a data-driven macromodel that could be used to design and optimize the geometrical parameters of that component. Since the S-parameters of the system under consideration depend on external parameters other than the independent variable (here the complex frequency s), the proposed inverse model could be considered as a parameterized macromodel which models the input-output relation of a parametric transfer function.

Chapter 3

Basics of Data-driven Surrogate Modeling

Due to its versatility, low evaluation cost, and widespread available toolboxes, data-driven model is popular among surrogate models [33]. In this chapter, a brief overview of approximation-based modeling including sampling strategies, different modeling methods (e.g., support vector regression), and model validation approaches is introduced.

The following are the steps involved in creating a data-driven surrogate:

1. *Sampling Strategy*: defines the training samples allocation within the design space and depends on the data source (e.g., full-wave simulations) and available computational budget.
2. *Training Data Acquisition*: In microwave applications, data is usually obtained from 2D or 3D simulations
3. *Model Identification*: finding the model's parameters either analytically, by addressing an appropriate linear regression problem [34], or by minimizing a suitably formulated cost function (e.g., Kriging [35]), or neural networks [36].
4. *Model Validation*: Verifying the approximation capability (i.e., approximating the training data) and the generalization capability (i.e., ability to predict unseen data) of the model. In practice, a trade-off between model approximation and generalization capabilities is required [37].

In general, the desired quality of approximating the training samples (a.k.a *model's bias*) could be decreased by using the complex models, but this usually causes increasing the sensitivity of the model's output (a.k.a *model's variance*) to particular data sets. In contrast, using regularization (i.e., penalizing the model complexity to achieve a smooth model) helps to reduce the model's variance but results in increasing the bias. To reduce both the bias and variance, a larger data set is usually required which is not feasible if the computational budget is limited.

3.1 Latin Hypercube Sampling Strategy

The sampling strategy is an important step of the surrogate modeling process because it defines how information about the system of interest is gathered. The amount of data about the system is clearly dependent on the size of the training set. Regardless, the number of samples that can be assigned is usually limited by the available computing budget. There are three main categories of sampling strategies: factorial, space-filling, and sequential sampling methods [33]. This section outlines one of the most popular space-filling sampling strategies, i.e., Latin Hypercube Sampling (LHS) [38], which is used in this research. In cases the system of interest's data comes from deterministic simulations, it is not necessary to spread out the samples for reducing the random errors resulting from imperfect measurements. Therefore, the space-filling strategy is utilized to allocate the sample points uniformly within the design space. In particular, when our knowledge of the system is limited.

In the LHS algorithm, to allocate L samples, each parameter range is divided into L bins yielding a total of L^n bins for a n -dimensional design space. Then, the samples are divided into the bins according to the following two rules: (i) each sample is randomly placed inside a bin, and (ii) there is exactly one sample in each bin for all 1D projections of the L samples and bins.

3.2 Model Identification: Least Square Support Vector Machine Regression

There are a number of well-known and commonly used data-driven modeling techniques, including neural networks [39], support vector regression [36], gaussian

process regression [40], polynomial regression [34], polynomial chaos expansion [41], kriging [35], etc.

In this section, the basics of the Least Square Support Vector Machine (LS-SVM) regression are introduced to be used in the next sections for building our data-driven model. In the following, the training samples will be denoted as $\{\mathbf{x}_i\}_{i=1}^L$, whereas the corresponding response samples as $f(\mathbf{x}_i)$. The data-driven model is constructed by approximating the data pairs $\{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^L$.

The LS-SVM is a statistical-learning strategy that replaces the original SVM's quadratic loss function with a least-squares linear scheme [36]. The LS-SVM approach has been effectively used for pattern recognition, time-series prediction, and nonlinear-function estimation issues due to its unique qualities such as its simple algorithm, quick convergence, and strong prediction ability [42]. It has been mathematically proven that SVM with RBF kernel can approximate any continuous nonlinear function arbitrarily well over a compact set to any degree of accuracy (a.k.a *Universal Approximation*) [43]. The RBF kernel-based LS-SVM is more effective at estimating nonlinear functions and can reduce the system noise, hence it has a good generalization ability.

Let us consider the nonlinear function for system response f of the following form:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b, \quad (3.1)$$

where, $\mathbf{x} \in \mathbb{R}^n$, $f(\mathbf{x}) \in \mathbb{R}$ and function $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ is the mapping from the input space to a higher dimensional (potentially infinite dimensional) feature space. The $\mathbf{w} \in \mathbb{R}^{n_h}$, and b are the weight vector in primal space, and bias term, respectively.

The following constrained optimization problem in the primal weight space could then be developed given a training set $\{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^L$.

$$\begin{aligned} \min_{\mathbf{w}, b, e} J(\mathbf{w}, e) &= \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{2} \gamma \sum_{i=1}^L e_i^2 \\ \text{s.t. } e_i &= f(\mathbf{x}_i) - \mathbf{w}^T \phi(\mathbf{x}_i) - b, \text{ for } i = 1, \dots, L. \end{aligned} \quad (3.2)$$

where γ is the regularization parameter. The optimization problem in (3.2) tries to minimize both the squared of the error computed on the training samples and the squared of the ℓ_2 -norm of the coefficients in the vector \mathbf{w} , also known as *Tikhonov*

Regularizer. The regularizer parameter γ , usually referred to as hyperparameter, provides a trade-off between the model accuracy and its flatness.

It should be noted that (3.2) is nothing more than a feature space formulation of a ridge regression cost function. It can be seen that the primal space formulation in (3.1) turns out to be a parametric model in which the number of regression unknowns (i.e., the cardinality of the weight vector \mathbf{w}) is equal to the number of basis functions collected in the vector $\boldsymbol{\phi}$. Therefore, the above formulation suffers from the curse-of-dimensionality. In particular, when \mathbf{w} becomes infinitely dimensional, this primal problem is hard to solve [36].

Therefore, by constructing a Lagrangian including both the cost function and the constraints in (3.2), the dual problem could be derived by finding the saddle point of the following function,

$$\mathcal{L}(\mathbf{w}, b, e; \boldsymbol{\beta}) = J(\mathbf{w}, e) - \sum_{i=1}^L \beta_i \{f(\mathbf{x}_i) - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - b - e_i\} \quad (3.3)$$

where, $\beta_i \in \mathbb{R}$ are Lagrange multipliers which are used as support values. According to the Kuhn-Tucker theorem, there are values of Lagrange multipliers for which the minimax of (3.3) equals the minimum of (3.2) [44].

The following solution is obtained by employing optimality conditions (i.e., $\frac{\partial \mathcal{L}}{\partial \mathbf{w}}, \frac{\partial \mathcal{L}}{\partial b}, \frac{\partial \mathcal{L}}{\partial e}$, and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = 0$) and removing the variables \mathbf{w} and e .

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \boldsymbol{\Omega} + \frac{\mathbf{I}}{\gamma} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix} \quad (3.4)$$

where $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_L)]^T$, $\mathbf{1} = [1, \dots, 1]^T$ and $\boldsymbol{\beta} = [\beta_1, \dots, \beta_L]^T$. The $\boldsymbol{\Omega}$ matrix elements are calculated by the so-called kernel trick [36] as,

$$\Omega_{kl} = \boldsymbol{\phi}(\mathbf{x}_k)^T \boldsymbol{\phi}(\mathbf{x}_l) = K(\mathbf{x}_k, \mathbf{x}_l) \quad (3.5)$$

The positive real constant γ determines the relative importance of $K(\mathbf{x}_k, \mathbf{x}_l)$ terms; in this case, a smaller γ value could be used to avoid overfitting. The resulting LS-SVM regression model for estimating the nonlinear function (with scalar output $f(\mathbf{x}) \in \mathbb{R}$) becomes then,

$$f(\mathbf{x}) = \sum_{i=1}^L \beta_i K(\mathbf{x}_i, \mathbf{x}) + b, \quad (3.6)$$

where $K(.,.)$ is the kernel function of the two vectors, and β_i and b are called regression coefficients and bias, respectively, which are the solution to the linear system (3.4).

For the kernel, the different nonlinear transformations could be applied, i.e., polynomial, radial basis function (RBF) and hyperbolic tangent kernel [45]. Throughout this research, the RBF kernel is used as,

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{\sigma^2}\right), \quad (3.7)$$

The complexity of the model is regulated by the kernel function parameter σ and the regularization parameter γ , which should be adjusted appropriately so that the model does not over-fit the training data. A large γ value will result in a model with less error, but overfitting can occur, while a small γ value minimizes the complexity of the model. Also, the larger σ indicates a stronger smoothing. These parameters, known as hyperparameters, are adjusted during model training, generally, through cross-validation [45].

The above dual formulation, i.e., primal space (3.1) and dual space (3.6) of the LS-SVM regression equations provides a non-parametric model. In fact, the dual space formulation in (3.6) provides a non-parametric model in which the number of regression coefficients β_i is independent of the dimensionality of \mathbf{x} . This allows for reducing the number of regression unknowns and thus improving the convergence of the model accuracy with respect to the number of training samples.

3.3 Surrogate Model Validation

The goal of model validation is to assess the predictive power of the model. Two main parts of the validation process are the error function selection and the approach used for estimating the predictive power capability which is briefly discussed in this section.

The error function, as well as the surrogate's target accuracy, are often problem dependent. In most cases, some awareness of the structure of the system response is necessary, as well as a thorough understanding of the meaning of the generalization estimator to be utilized [46]. Absolute and relative error functions are the two types

of error functions. Because relative error metrics are more context-independent, they are often preferred. some of the most prominent relative error functions write [33]:

$$\text{Average relative error (ARE): } \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \tilde{y}_i|}{|y_i|} \quad (3.8)$$

$$\text{Root relative square error (RRSE): } \sqrt{\frac{\sum_{i=1}^N (y_i - \tilde{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2}} \quad (3.9)$$

$$\text{Bayesian estimation error quotient (BEEQ): } \left(\prod_{i=1}^N \frac{\sum_{i=1}^N |y_i - \tilde{y}_i|}{\sum_{i=1}^N |y_i - \bar{y}_i|} \right)^{\frac{1}{N}} \quad (3.10)$$

where, N , y_i , \tilde{y}_i , and \bar{y}_i denote the number of samples, the actual, the predicted, and averaged response samples, respectively.

Following the selection of the error function, the model generalization capability can be calculated using a variety of methods. It is obvious that evaluating the model solely based on the training set is not reliable, because the interpolative models have always a zero error therein. Some modeling techniques identify a surrogate model as well as an estimate of the associated approximation error (e.g., kriging [35]). Alternatively, there are approaches that can be utilized independently to test a model's prediction capabilities beyond the set of training points. The *split-sample* method [34] is probably the most often used strategy, in which part of the available data set (the training set) is used to develop the surrogate, while the remaining (the test set) is used solely for model validation. The error predicted by this method, however, is highly dependent on how the set of data samples is partitioned. Furthermore, if only a few testing points are available, it may produce very biased results.

The model generalization error can be more accurately estimated via cross validation [45]. The method divides the data set into K subsets and uses each of these subsets as a testing set for a surrogate built on the other $(K - 1)$ subsets in a sequential manner. All of the K error metrics obtained during this method can be used to determine the prediction error (e.g., as an average value). In its most severe form, the Leave-One-Out error, only one point at a time is used to estimate error. Cross-validation produces a less biased error estimate than the split-sample method. This method has the drawback of requiring the surrogate to be built multiple times. It is worth mentioning that choosing the error function, calculating the generalization capability, and determining the target accuracy are all difficult tasks. In practice,

these factors are assumed at first, and the model is built after the training data is obtained. A visual evaluation of the model answers and their consistency with the system outputs is usually the deciding stage. Depending on the results of this evaluation, the modeling step may need to be repeated with the new arrangement [33].

The data-driven modeling flow, which includes allocating samples, collecting data, identifying, and validating the models, can be repeated until the desired surrogate accuracy level is achieved. Each iteration adds a new set of training samples to the previous ones. Some sample allocation strategies, try to improve the model's overall accuracy by introducing new samples where the predicted modeling error is highest [33].

Chapter 4

Inverse Modeling Optimization

In this chapter, we propose a completely different approach called *Inverse Modeling* to design optimization of microwave components, which is different from conventional repetitive optimizers, is very efficient, and does not require any iteration. Given the desired response samples (e.g., frequency dependent S-parameter samples) of a microwave component, the goal is to build a data-driven macromodel that could be used to design and optimize the geometrical parameters of that component.

4.1 Literature Review

In recent years, machine learning (ML) regression techniques have been recognized as powerful tools in microwave modeling and design [47]. Applications have been reported in parametric modeling of microwave components [48], microstrip patch antenna and printed microstrip radial stub [49], broadband Doherty and continuous class-F power amplifier design [50, 51], and compact microwave components [13], all of which use the ML techniques as a forward model according to common practice. Forward modeling is the process of training an ML-based technique to directly model the electromagnetic (EM) behavior of generic microwave components in terms of frequency/time responses (e.g., S-parameters, voltage/current waveforms, etc.), by using as input its physical and electrical parameters. For design purposes, the problem to be solved goes in the opposite direction, i.e., given the desired specification of electrical responses, we are looking for the optimal configuration of the geometrical/physical parameters of the microwave component. The optimization

based method is the traditional way to solve such an inverse problem [52–54], where the forward model is evaluated iteratively until the optimal geometrical/physical parameters are found, such that the model output matches the electrical goals and constraints. However, especially for the case of multi-objective optimization, such a conventional optimization scheme can be rather expensive, since it may require many iterations to converge [55, 56].

Inverse models represent a promising alternative to solve the inverse problem required by the optimization task[57]. Unlike the common optimization method based on the forward map, starting from the desired EM behavior of the device under modeling, the inverse model allows predicting directly the corresponding values of geometrical/physical parameters, without requiring any iteration or iterative algorithm based on the forward model. As a result, the inverse modeling approach outperforms the optimization-based method in terms of computational time. Indeed, inverse surrogate models can be used to speed up the geometry scaling of compact microwave passives [58], as well as the rapid EM-driven antenna dimension scaling [59].

The inverse model is usually trained via a data-driven regression approach since in most cases its analytical formulation is not available [57]. One may argue that providing the training data (which is usually acquired from full-wave simulations) for an ML-based inverse model is still time-consuming, but it should be noted that this step could be accelerated using the parallelized computation techniques [60], while parallelization is not applicable to *plain iterative optimization* schemes[61], where the result of each iteration is required to be known before starting a new iteration.

Even if the optimization based on the inverse modeling approach is usually faster compared with the performance of classical optimization schemes adopted for microwave design optimization, two main challenges usually exist in practice. First, an inverse model is generally ill-posed, in the sense that a given set of electrical responses might be generated by more than one combination of the component parameters. This means that the inverse model is usually a one-to-many map suffering from nonuniqueness [57]. Since it is generally impossible to train a data-driven model to fit multiple output values for the same input values at the same time, the training error will remain high, and the model accuracy will be poor. As a result, it is difficult to train the model in many practical inverse problems [62]. Second, the construction of the inverse model for microwave design problems is rather complex

and challenging, since the dimensionality of its input space is defined by the number of frequency/time points used to characterize the response of interest of the component (e.g., for the case of S-parameters, the dimensionality of the input space turns out to be defined as the number of frequency samples multiplied by the number of S-parameters of interest). This means that the regression technique used for training the inverse model should be able to learn a nonlinear function in high-dimensional input space. This might lead to accuracy issues, a curse of dimensionality, and an enormous demand for training samples [63].

Regularization methods have been demonstrated to be an effective tool for solving ill-posed inverse problems [64]. Also, ML techniques have emerged as a data-driven alternative to traditional inversion approaches in recent years, thus showing significant advances in dealing with both the ill-posedness and the high dimensionality of a variety of inverse problems, such as imaging [65, 66], tomography [67], nonlinear electromagnetic inversion [68]. A specific review of the neural network based inverse modelling techniques for microwave applications is presented in [69]. Both the direct inverse modeling with derivative division [57], and the multivalued neural network inverse modeling technique [62] methods address the problem of nonuniqueness. Also, a hybrid deep neural network inverse model with high dimensional inputs has been proposed in [63] for the inverse problems without nonuniqueness issue, where a big data set including thousands of training samples is required to train the inverse model. However, none of these existing strategies are solely applicable to inverse problems with both nonuniqueness and high-dimensional inputs issues [69].

Here, we propose a compressed ML-based inverse model [70] to handle both of the issues discussed above. Starting from the desired frequency-domain ideal mask, the inverse model built via this approach is able to inexpensively predict the optimal geometrical parameters of the microwave filters, without using computationally expensive iterative optimization algorithms. Due to its simplicity, the capability of providing compact non-parametric models, and implicit regularizer, the least-square support vector machine (LS-SVM) regression [36] has been considered to construct the advocated inverse model. Also, a data compression strategy based on the principal component analysis (PCA) has been utilized to mitigate the high dimensionality of the input space issue, followed by a qualitative approach [71] based on variability analysis to find the appropriate number of PCA coefficients.

The computational advantages are due to the compressed training set having a lower dimensionality than the original one, which translates into the better predictive power of the inverse model even when utilizing limited-cardinality training datasets [71]. Nevertheless, the advantages of the discussed inverse modeling methodology are deteriorated by the high sensitivity of the inverse model accuracy to the number of PCA coefficients used as its input. Indeed, this hyperparameter plays a significant role in the proposed inverse modeling scheme to simultaneously mitigate the ill-posed and high-dimensional input space challenges, thus needs to be accurately tuned. Therefore, also a new method is proposed (see section 4.4) to find the optimum number of PCA coefficients (i.e., the compression level) for each problem at hand. This way, the ill-posedness of the inverse problem is tackled by choosing the optimal dimension for the new compressed input space with orthonormal PCA coefficients, along with the regularization technique.

In order to validate the inverse model performance, a new error function is proposed as a result of combining the inverse and forward ML-based models, which improves the reliability and the interpretability of the resulting error. The presented approach has a significant advantage over the prior development that makes it general and easy to use for microwave applications compared with the existing data-driven inverse modeling techniques.

In summary, the main contributions of this section include significantly: a detailed theoretical formulation and discussion of the proposed compact inverse model; a deeper discussion of the ill-posed issue; a new complicated microwave example; and the proposal of a new empirical technique to automatically tune the number of PCA coefficients.

4.2 Inverse Model Formulation and Challenges

Most of the optimization algorithms used for the design of microwave components rely on the information provided by the forward model. The forward model provides a map between the configuration of the device parameters (e.g., geometrical, physical, and electrical parameters), and the corresponding responses in frequency- or time-domain (e.g., S-parameters, voltage/current waveforms, etc...). Without loss

of generality, the mathematical formulation of a forward model writes:

$$\mathbf{y} = \mathcal{M}(\mathbf{x}) \quad (4.1)$$

where the forward model is a vector function $\mathcal{M} : \mathbb{R}^m \rightarrow \mathbb{R}^{(n \times n_o)}$ mapping the system parameters $\mathbf{x} = [x_1, \dots, x_m]^T$ into the output vector $\mathbf{y} = [y_1, \dots, y_{(n \times n_o)}]^T$, which can be interpreted as vectorization of a matrix $\mathbf{D}_Y \in \mathbb{R}^{n \times n_o}$ collecting n time/frequency points and for each of the n_o -variable of interest. As an example, for the case of the S-parameters of a 2-port structure, n would be the number of frequency samples and $n_o = 4$ represents the number of S-parameters S_{ij} for $i, j = 1, 2$. The forward model \mathcal{M} can be a full-computational model (e.g., analytic formula, equivalent circuits, full-wave simulations) or a fast-to-evaluate data-driven model (e.g., ML-based model) constructed using the training samples usually provided by full-computational models [9].

Given a set of constraints on the responses of a generic microwave component, the forward model is called several times during the optimization process with the aim of exploring the parameters space and getting the optimal configuration of the input parameters. Such an iterative approach is required within the design optimization because the forward model is used to solve an inverse problem. Indeed, the most natural way to tackle an optimization problem is to use the so-called *inverse model*. The underlying idea is to build a model in which the inputs and outputs are reversed with respect to the forward model (i.e., $\mathcal{M}^{-1} : \mathbb{R}^{(n \times n_o)} \rightarrow \mathbb{R}^m$), such that:

$$\mathbf{x} = \mathcal{M}^{-1}(\mathbf{y}). \quad (4.2)$$

The inverse model goes in the opposite direction with respect to the forward model and allows to efficiently and inexpensively perform the optimization task via a simple function evaluation, without requiring any iteration. Unfortunately, in most cases, the inverse model is not available in closed form, since it would require an accurate and invertible analytical formulation of the forward model. Therefore, a numerical approximation $\hat{\mathcal{M}}^{-1} \approx \mathcal{M}^{-1}$ is usually constructed by means of data-driven regressions starting from a set of training samples computed via the standard forward model [47].

Despite the apparent simplicity and extreme efficiency, the construction of the inverse model for microwave applications via regression techniques is rather

challenging and complicated. Specifically, this contribution deals with the development of a modeling approach that is able to deal with the following challenges:

- the dimensionality of the input space of the inverse model (i.e., the cardinality of $|\mathbf{y}| = n \times n_o$) can be extremely high;
- the problem is ill-posed;
- the inverse model should be trained using a *small* set of training samples (e.g., $\ll 1000$ samples).

4.3 Proposed Scheme for the Inverse Model Construction

This section describes the modeling procedure adopted hereafter to overcome the above challenges. The proposed methodology is based on a two-step modeling scheme: (i) the dimensionality of the \mathbf{y} vector used as input for the inverse model is reduced via the compression provided by the PCA; (ii) the compressed representation of the training set is used to train the inverse model based on the LS-SVM regression.

4.3.1 Dimensionality compression via PCA

In microwave applications, the dimensionality of the input space of the inverse model (i.e., $|\mathbf{y}| = n \times n_o$) is usually extremely high. Indeed, the vector \mathbf{y} collects all the frequency/time points of the responses for each variable of interest. Obviously, such high-dimensionality of the input of the inverse model has a big impact on the model accuracy [71]. Increasing the input dimension of a nonlinear function makes the input-output relationship describing the inverse model more complicated to learn, thus leading to a large training set and accuracy issues [63]. Such a problem can be easily overcome by considering a subset of “pre-selected” frequency/time points to describe the component responses, the so-called response features [72]. However, the frequency response feature selection directly depends on the specific problem at hand, which limits the model’s applicability and brings additional complexity to the modeling process, since it would require a human-intensive feature definition and extraction process.

Here, we introduce an alternative approach based on PCA compression. Starting from the complete set of responses (i.e., from the full vector \mathbf{y}), PCA-based data compression is adopted to exploit and remove redundant information from the responses in the training set, thus leading to a compressed approximation with a lower dimensionality with respect to the original data [73]. To this aim, the training responses $\{\mathbf{y}_i\}_{i=1}^L$ are collected in the matrix $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_L] \in \mathbb{R}^{(n \times n_o) \times L}$. The matrix \mathbf{Y} is used within the PCA algorithm to find out the smallest set of “principal components”, approximating the training responses $\{\mathbf{y}_i\}_{i=1}^L$ as [74]:

$$\mathbf{y}_i \approx \boldsymbol{\mu} + \sum_{n=1}^{\tilde{n}} \tilde{y}_{n,i} \mathbf{u}_n \quad (4.3)$$

for $i = 1, \dots, L$, where L is the number of training samples.

The PCA coefficients $\tilde{y}_{n,i}$ writes:

$$\tilde{y}_{n,i} = \mathbf{u}_n^T (\mathbf{y}_i - \boldsymbol{\mu}), \quad (4.4)$$

where $\boldsymbol{\mu}$ is the row-wise mean, and the orthonormal principal components $\{\mathbf{u}_n\}_{n=1}^{\tilde{n}}$ are the left singular vectors calculated via the singular value decomposition applied to matrix \mathbf{Y} . According to (4.3) and (4.4), the training samples $\{\mathbf{y}_i\}_{i=1}^L$ with $\mathbf{y}_i \in \mathbb{R}^{(n \times n_o)}$ can be approximated (with a tunable accuracy) via a compressed training set $\{\tilde{\mathbf{y}}_i\}_{i=1}^L$, in which $\tilde{\mathbf{y}}_i = [\tilde{y}_{1,i}, \dots, \tilde{y}_{\tilde{n},i}]^T \in \mathbb{R}^{\tilde{n}}$, where $\tilde{n} \ll (n \times n_o)$. Now, the compressed representation of the training input can be used to train a compact inverse model $\tilde{\mathcal{M}}^{-1} : \mathbb{R}^{\tilde{n}} \rightarrow \mathbb{R}^m$, such that:

$$\mathbf{x} = \tilde{\mathcal{M}}^{-1}(\tilde{\mathbf{y}}), \quad (4.5)$$

where $\tilde{\mathbf{y}} \in \mathbb{R}^{\tilde{n}}$ is the compressed representation of the vector $\mathbf{y} \in \mathbb{R}^{(n \times n_o)}$.

The number of PCA coefficients \tilde{n} must be carefully tuned since it has a huge impact on the performance of the inverse model. Specifically, if \tilde{n} is chosen too small (i.e., high compression is used for the input space samples), the inverse map that must be learned by the inverse model is *simpler* but is more affected by the nonuniqueness issue. On the other hand, if \tilde{n} is too large (i.e., low compression), the inverse model can suffer the infamous curse-of-dimensionality. A wise strategy is to use only the PCA components, which show the largest variability in the compressed

training set [71]. However, the above approach provides qualitative information on the sensitivity of each PCA component, without providing a full understanding of the model error as a function of the parameter \tilde{n} . Therefore a more advanced technique for tuning the parameter \tilde{n} based on a double cross-validation scheme is introduced in section 4.4.

4.3.2 LS-SVM regression: curse-of-dimensionality and ill-posed problems

Unavoidably, the accuracy and reliability of the inverse model depend on the regression technique used to train it. The regression technique should be able to limit in somehow the detrimental effect of the ill-posedness of the inverse model, but at the same time, it should be able to accurately predict the nonlinear input-output relationship of the inverse model with a small set of training samples.

In the above scenario, LS-SVM regression (discussed in Chapter 3, section 3.2) can be seen as a promising regression technique for constructing the advocated inverse model. First of all, the LS-SVM uses a Tikhonov regularizer which allows for limiting the issue related to the ill-posed nature of the inverse problem. Second, thanks to the kernel trick, the LS-SVM regression admits a nonparametric formulation (see (3.6)) in which the number of regression unknowns is independent of the dimensionality of the input space, and it is equal to the number of training samples. The latter feature limits the curse-of-dimensionality and improves the method convergence in terms of training samples [36] compared to other techniques based on the more flexible neural network and deep learning techniques [63].

Given the compressed training set $\{(\mathbf{x}_l, \tilde{\mathbf{y}}_l)\}_{l=1}^L$, where $\mathbf{x}_l \in \mathbb{R}^m$ and $\tilde{\mathbf{y}}_l \in \mathbb{R}^{\tilde{n}}$, the multioutput inverse model $\tilde{\mathcal{M}}^{-1}$ trained in the primal space formulation via the LS-SVM regression writes [36]:

$$x_j(\tilde{\mathbf{y}}) = \tilde{\mathcal{M}}_j^{-1}(\tilde{\mathbf{y}}) = \langle \mathbf{w}_j, \boldsymbol{\phi}(\tilde{\mathbf{y}}) \rangle + b_j, \quad (4.6)$$

for $j = 1, \dots, m$, where m is the number of physical parameters.

In the above equation, $\boldsymbol{\phi}(\tilde{\mathbf{y}}) = [\phi_1(\tilde{\mathbf{y}}), \dots, \phi_{n_h}(\tilde{\mathbf{y}})]^T$ is a vector function mapping the input space into the feature space, such as $\boldsymbol{\phi}(\cdot) : \mathbb{R}^{\tilde{n}} \rightarrow \mathbb{R}^{n_h}$ and $\langle \cdot, \cdot \rangle$ represented the inner product in \mathbb{R}^{n_h} . The weight vector $\mathbf{w}_j \in \mathbb{R}^{n_h}$, and the bias term b_j are the

regression unknowns estimated by solving the following optimization problem [36],

$$\begin{aligned} \min_{\mathbf{w}_j, b_j, \mathbf{e}_j} J(\mathbf{w}_j, e_j) &= \frac{1}{2} \|\mathbf{w}_j\|^2 + \frac{1}{2} \gamma_j \sum_{l=1}^L e_{j,l}^2 \\ \text{s.t. } e_{j,l} &= x_{j,l} - \langle \mathbf{w}_j, \boldsymbol{\phi}(\tilde{\mathbf{y}}_l) \rangle - b_j, \text{ for } l = 1, \dots, L. \end{aligned} \quad (4.7)$$

Fortunately, the LS-SVM regression admits an equivalent dual formulation which writes [36]:

$$x_j = \mathcal{M}_j^{-1}(\tilde{\mathbf{y}}) = \sum_{l=1}^L \beta_{j,l} K_j(\tilde{\mathbf{y}}_l, \tilde{\mathbf{y}}) + b_j, \quad (4.8)$$

where, $K_j(\tilde{\mathbf{y}}_l, \tilde{\mathbf{y}}) = \langle \boldsymbol{\phi}(\tilde{\mathbf{y}}_l), \boldsymbol{\phi}(\tilde{\mathbf{y}}) \rangle$ is the so-called kernel function such that $K_j(\cdot, \cdot) : \mathbb{R}^{\tilde{n} \times \tilde{n}} \rightarrow \mathbb{R}$, and the regression unknowns are the coefficient $\beta_{j,i}$ and the bias b_j . For the kernel, the RBF kernel is used as,

$$K_j(\tilde{\mathbf{y}}_l, \tilde{\mathbf{y}}) = \exp \left(-\frac{\|\tilde{\mathbf{y}}_l - \tilde{\mathbf{y}}\|^2}{\sigma_j^2} \right), \quad (4.9)$$

where the parameter σ_j is the kernel hyperparamter.

It is important to remark that the dual space formulation in (4.8) provides a non-parametric model in which the number of regression coefficients $\beta_{j,l}$ is independent of the dimensionality of $\tilde{\mathbf{y}}$. This allows for reducing the number of regression unknowns and thus improving the convergence of the model accuracy with respect to the number of training samples. Moreover, the regularizer used by the LS-SVM regression allows limiting the detrimental effects introduced by the considered ill-posed inverse problem [36].

The LS-SVM regression is already available in MATLAB within LS-SVMLab Toolbox version 1.8 [75]. In such a tool, the regression hyperparameters (i.e., γ_j and σ_j) are adjusted during model training via a cross validation scheme, being the latter a common approach when a small set of samples are available [45]. However, the tuning of the new hyperparameter \tilde{n} introduced by the PCA compression is not addressed by the above algorithm. Therefore, the LS-SVM training algorithm should be embedded within the advanced tuning procedure presented in section 4.4.

Throughout this section, the training samples for the specific case of the inverse model have been generated around an initial design [62]. However, it is important to remark that, the initial design is not directly used by the proposed inverse model.

Indeed, different from the standard optimization scheme based on the forward model, the inverse model is not influenced by the initial guess used during the training phase.

4.3.3 Discussion of the ill-posed challenge

Within the proposed modeling framework, the ill-posed nature of the inverse model is tackled by combining both the benefits of the built-in regularizer of the LS-SVM regression and by tuning the dimensionality of the input space of the inverse model (i.e., the number of PCA coefficients used as input by the inverse model).

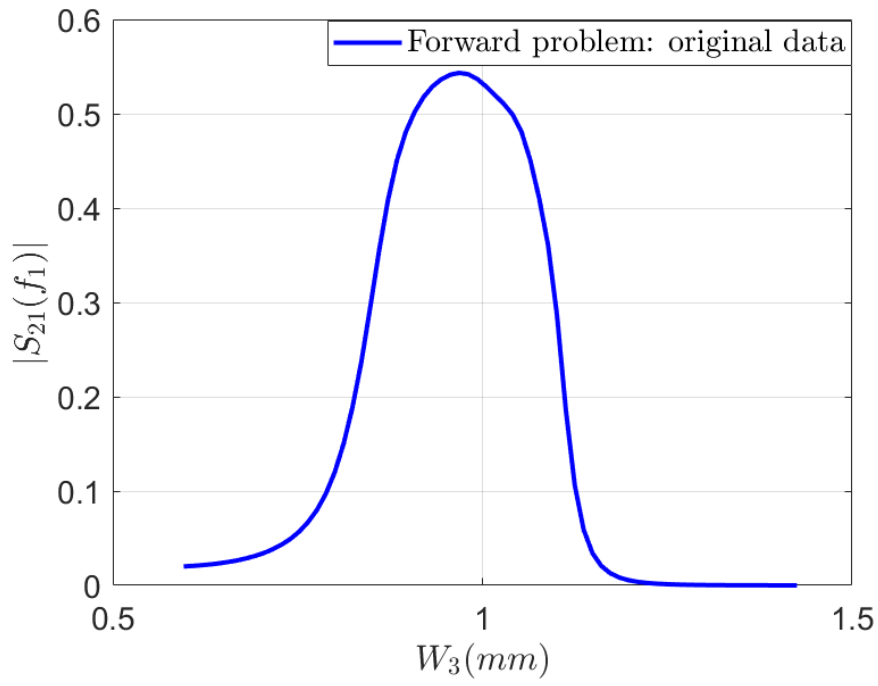
Indeed, even if the Tikhonov regularizer used by the LS-SVM regression allows to limit ill-posedness issues related to the overfitting [76], it does not completely fix the nonuniqueness issue [62] and might lead to inaccurate predictions. Therefore, in our approach we are trying to overcome the ill-posedness of the inverse problem by increasing the dimensionality of the input space of the inverse model (an increased number of input data on the desired response must be used in order to get an accurate prediction of the design parameters \mathbf{x} [77]), thus assuming that the new inverse map will be well-posed.

Here, we illustrate the ill-posed problem through a coupled-line microstrip filter inverse modeling problem [71], shown in Fig. 4.4(a). The filter response magnitude $|S_{21}|$ at $f_1 = 2500 \text{ MHz}$ is used as the input to the inverse model, and the width of the parallel coupled-lined section W_3 is used as the output. The forward problem relationship is shown in Fig. 4.1(a), in which for every input value, the output of this forward problem is a single value (unique solution). The plot clearly highlights that even if the forward map is well-posed, the inverse problem relationship (shown in Fig. 4.1(b) with a solid blue line) is ill-posed. For instance, there are two different values of W_3 for which $|S_{21}(f_1)|=0.3$, namely 0.85, and 1.1 mm. The multivalued output solutions for the same input, i.e., $(x, y) = (0.85, 0.3)$ and $(x, y) = (1.1, 0.3)$, result in contradictory data for training a generic data-driven model, making the inverse model ill-posed.

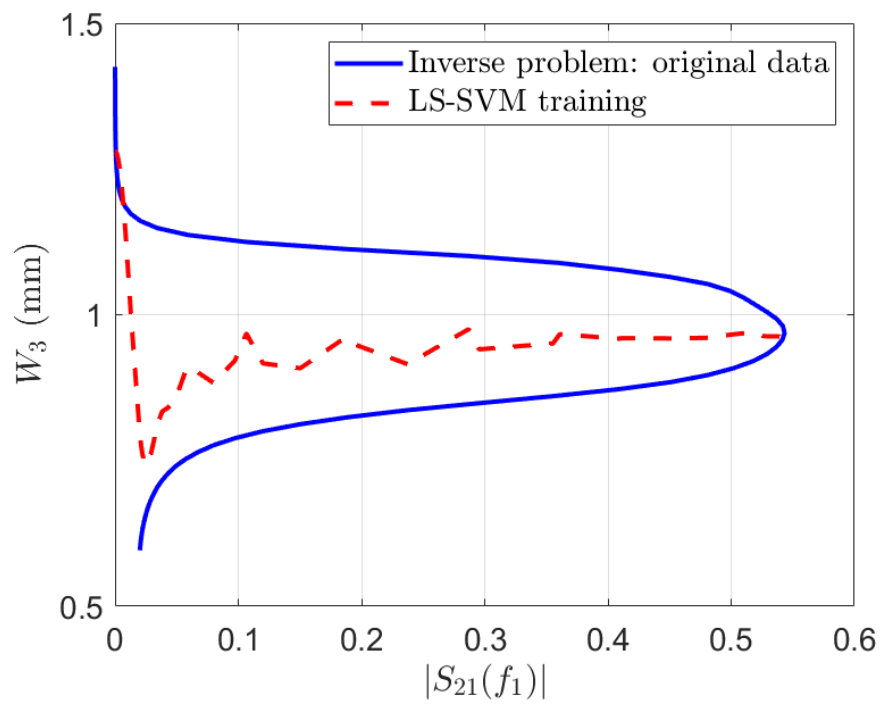
Indeed, the predictions in Fig. 4.1(b) (dashed red line) are of a plain inverse model trained via the LS-SVM regression. The results show that the direct LS-SVM regression inverse model with a single input, i.e., $y = |S_{21}(f_1)|$, cannot match the nonunique input-output relation. A wise strategy is to use vector-valued inputs to the inverse model containing multiple samples of the frequency response at different

frequency points, with the aim that increasing the input dimensionality helps to reduce the possibility of an ill-posed problem. For example, using an inverse model with two samples of S-parameter magnitude $|S_{21}|$, i.e., $\mathcal{M}^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}$, improves the inverse model accuracy as shown in Fig. 4.1(c), by introducing a new inverse mapping with increased input dimensionality which is not ill-posed anymore.

The above simplified problem highlights the importance of two key points in the proposed inverse modeling method, i.e., the regularization technique and the input space dimension. The ill-posed issue in high dimensions is even more challenging since it is not possible to visualize the contradicting data in multi-dimension space. Therefore, a systematic approach to solving the ill-posed inverse problem is very important. On the other hand, in our proposed method all available S-parameter samples, i.e., $|S_{21}(f_i)|$ and $|S_{11}(f_i)|$, $\forall i$, could be used thanks to the PCA compression technique which exploits and removes the redundant information from the scattering parameters, thus leading to a compressed approximation with orthonormal PCA coefficients[78], to be used as input to the LS-SVM regression model. This way, by using any PCA coefficient, new information (since is orthogonal to others) is being used as additional input to the inverse model.



(a)



(b)

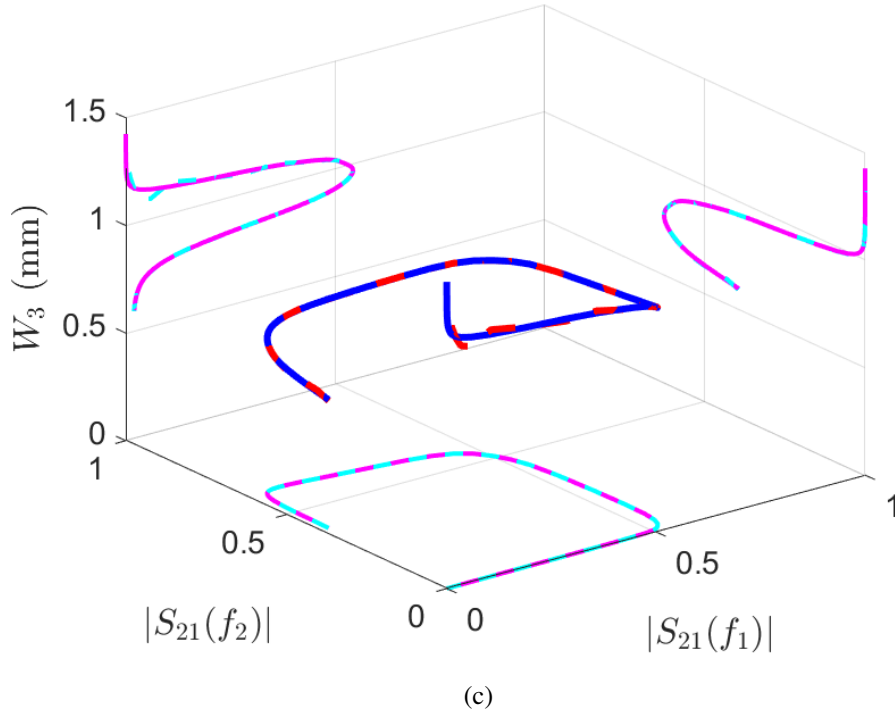


Fig. 4.1 Illustration of different inverse modeling techniques for the coupled-line microstrip filter example. (a) Input-output relationship of the forward problem is unique. (b) Inverse problem relationship (solid blue line) and LS-SVM direct inverse modeling techniques training (dashed red line) results. The ML-based inverse model with only one input, i.e., $y = |S_{21}(f_1 = 2500 \text{ MHz})|$, cannot match the contradictory data (multivalued solutions). (c) Training results of the direct inverse model with 2-vector input. Increasing the input dimensionality, a new well-posed inverse mapping is introduced which helps the LS-SVM regression to be well trained (dashed red line). Two S-parameter samples, i.e., $|S_{21}(f_1)|$ and $|S_{21}(f_2)|$, $f_1 = 2500 \text{ MHz}$ and $f_2 = 2600 \text{ MHz}$, are used as the inverse model input.

4.4 Inverse Model's Hyperparameters Optimization

The accuracy of the proposed inverse model mainly depends on the tuning of the hyperparameter representing the number of PCA coefficients \tilde{n} used as input for the inverse model. Such parameter provides information on the original response data (e.g., scattering parameters), and also determines the input space dimensionality. In this section, a new double cross validation technique based on the combination of the inverse and forward model is proposed to find the optimal hyperparameter \tilde{n} in an automated way.

Unfortunately, the standard algorithms for the hyperparameters optimization are not directly applicable for tuning \tilde{n} in our inverse model, since its output is not unique in general [77]. As an example, let us assume to tune \tilde{n} by using the error computed by the inverse model on a validation set $\mathcal{V} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{|\mathcal{V}|}$, by using the following error function[47]:

$$\begin{aligned} err(\tilde{n}) &= \frac{1}{2} \sum_{i=1}^{|\mathcal{V}|} \|\mathbf{x}_i - \mathcal{M}^{-1}(\tilde{\mathbf{y}}_i; \tilde{n})\|^2 \\ &= \frac{1}{2} \sum_{i=1}^{|\mathcal{V}|} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2. \end{aligned} \quad (4.10)$$

In the above equation, the error $err(\tilde{n})$ is calculated in the output space of the inverse model. This can reduce the reliability and the interpretability of the resulting error since the above estimator is not able to deal with the inherent ill-posedness nature of the problem [77].

On the other hand, a combination of the inverse and forward ML-based models could be used as a more reliable scheme for validating the inverse model performance [57]. Therefore, a forward model $\mathcal{M} : \mathbb{R}^m \rightarrow \mathbb{R}^{\hat{n}}$ is trained once, using the available training set $\mathcal{T} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{|\mathcal{T}|}$, for which a fixed number of PCA coefficients \hat{n} is considered in its $\tilde{\mathbf{y}}_i$ samples, that in general, \hat{n} can be different from \tilde{n} . It should be noted that due to the large variability of the parameter \mathbf{x} used as input, the forward model might not be able to accurately approximate the behavior of a large set of PCA coefficients. However, by evaluating the forward model on the validation set and looking at the scatter plots [77], the most first accurate PCA coefficients could be selected as the optimal \hat{n} (see Fig. 4.6). It is important to remark that the experiment results show that the proposed empirical method for tuning the \tilde{n} is not very sensitive to the chosen \hat{n} .

As shown in Fig. 4.2 (a), the trained forward model accepts the $\hat{\mathbf{x}}_i$ as its input and estimates an \hat{n} -value output vector $\hat{\mathbf{y}}$ containing the corresponding PCA coefficients. Then, by using the transformation (4.3) with \hat{n} , the full vector of responses $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_{n \times n_o}]^T$ is reconstructed as an estimation of the true response \mathbf{y} in the validation set, such that $\hat{\mathbf{y}} = \mathcal{M}(\mathcal{M}^{-1}(\mathbf{y}; \tilde{n}))$.

As a result, now the test error could be computed in the response domain (i.e., full vector \mathbf{y} -domain). In order to do this, we can recast the samples in the \mathbf{y} -domain into a

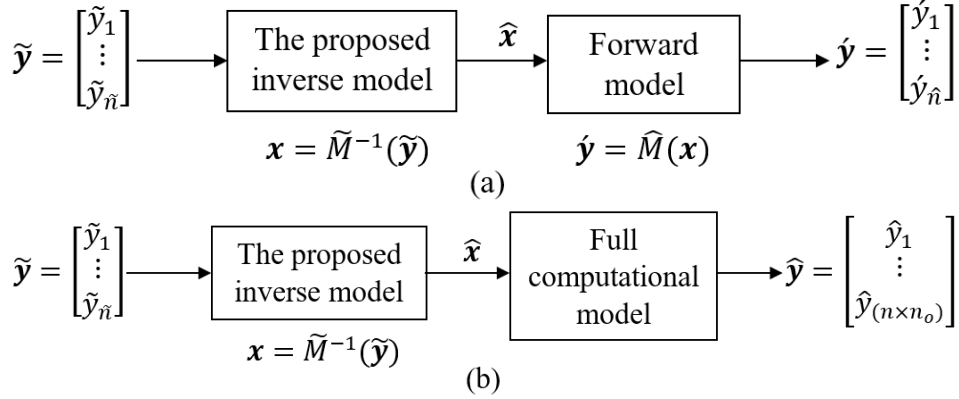


Fig. 4.2 (a) Successive combination of ML-based inverse and forward models used in CV scheme for tuning the inverse model hyperparameter \tilde{n} . (b) Using the full computational model (e.g., ADS) for the test of the inverse model predictions in practical microwave design examples.

tensor $\mathcal{Y} \in \mathbb{R}^{n \times n_o \times |\mathcal{V}|}$, where the element $y_{i,j,v}$ corresponds to the i -th frequency/time point of the j -th design response for the v -th validation sample. Using the above notation, a new error function can be defined as:

$$err(\tilde{n}) = \frac{1}{|\mathcal{V}|} \sum_{j=1}^{n_o} \sum_{v=1}^{|\mathcal{V}|} \frac{\sqrt{\sum_{i=1}^n |y_{i,j,v} - \hat{y}_{i,j,v}|^2}}{\sqrt{\sum_{i=1}^n |y_{i,j,v}|^2}} \quad (4.11)$$

As discussed in section 3.3, since a small training set is available (i.e., L is small), the more reliable way of evaluating the above error function is the cross validation (CV) scheme [45]. This allows us to use all data both for training and for validation, resulting in a less biased model compare to other methods. In K -fold CV, we randomly split the available training set \mathcal{T} into K partitions with almost equal sizes, called “folds”. Then going over K iterations, in each iteration, one of the folds is reserved as \mathcal{V} subset for model validation, while the $K - 1$ remaining folds are used as the \mathcal{T}' subset for training the model. Fig. 4.3. shows an illustration of 10-fold CV scheme (i.e., $K = 10$) for the case with $L = 200$ training samples. Thus, by changing the model parameter $\tilde{n} \in \mathcal{S}$ and using the K -fold CV scheme, an K -by- $|\mathcal{S}|$ test error matrix **err** is built from (4.11), in which the $\mathbf{err}(k, \tilde{n}) = err_{k, \tilde{n}}$ denotes the test error of the model with hyperparameter \tilde{n} computed over the k -th validation fold. Also, \mathcal{S} is a pre-defined set of the candidate PCA coefficients numbers that are being investigated for tuning the hyperparameter \tilde{n} . For each value

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Step 1	20	20	20	20	20	20	20	20	20	20
Step 2	20	20	20	20	20	20	20	20	20	20
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Step 10	20	20	20	20	20	20	20	20	20	20

Legend: Validation fold, Train fold

Fig. 4.3 Illustrative example of K -fold cross validation scheme with $K = 10$ and 200 training samples.

of \tilde{n} considered during the CV, the averaged $CV_{err}(\tilde{n})$ and standard deviation $CV_{std}(\tilde{n})$ of the test error over K iterations is defined as follows [79],

$$CV_{err}(\tilde{n}) = \frac{1}{K} \sum_{k=1}^K err_{k,\tilde{n}}, \quad (4.12)$$

and

$$CV_{std}(\tilde{n}) = \sqrt{\frac{1}{K-1} \sum_{k=1}^K |err_{k,\tilde{n}} - CV_{err}(\tilde{n})|^2}. \quad (4.13)$$

To find the optimum value \tilde{n}_{opt} , the criteria used is that, after a significant decrease in mean error $CV_{err}(\tilde{n})$ curve, the PCA coefficient \tilde{n} corresponding to the minimum standard deviation of the CV_{err} , i.e., $CV_{std}(\tilde{n})$, is selected as the optimum number \tilde{n}_{opt} (see application examples in section 4.5). This guarantee a low sensitivity of the model accuracy with respect to both the training and validation samples.

The proposed CV scheme for tuning the hyperparameter \tilde{n} is illustrated in Algorithm 1. In this algorithm, what we mean by "Use \mathcal{T}' to build the model", is that an LS-SVM regression model is built using the subset \mathcal{T}' via the process discussed in section 4.3, i.e., tuning the hyperparameters (γ, σ) and training the model. Therefore, Algorithm 1, is a double CV scheme so that in the first level the subsets \mathcal{T}' and \mathcal{V} are being used by the first CV scheme for tuning parameter \tilde{n} , and at the same time, the subset \mathcal{T}' is further split by the second CV scheme to tune and train the hyperparameters (γ, σ) of the LS-SVM regression models. It should be noted that the use of disjoint subsets (i.e., $\mathcal{V} \cup \mathcal{T}' = \mathcal{T}$ and $\mathcal{V} \cap \mathcal{T}' = \emptyset$) in the

Algorithm 1: K -fold CV for tuning parameter \tilde{n}

1. Build $\mathcal{T} = \{\mathbf{x}_i, \tilde{\mathbf{y}}_i\}_{i=1}^{|\mathcal{T}|}$ with maximum numbers of PCA coefficients in $\tilde{\mathbf{y}}_i$, using (4.4)
 2. Randomly split \mathcal{T} into K partitions (folds) with almost equal sizes
 3. **for** $k = 1$ to K **do**
 4. $\mathcal{V} \leftarrow$ partition k from \mathcal{T}
 5. $\mathcal{T}' \leftarrow \mathcal{T} \setminus \mathcal{V}$
 6. **for** $\tilde{n} \in \mathcal{S}$ **do**
 7. Use \mathcal{T}' to build the inverse model $\tilde{\mathcal{M}}^{-1}$
 8. Use the trained inverse model $\tilde{\mathcal{M}}^{-1}$ to predict \mathcal{V}
 9. $err_{k,\tilde{n}} \leftarrow$ Use the forward model $\hat{\mathcal{M}}$ and (4.3) to evaluate predicted \mathcal{V} in \mathbf{y} -domain, using (4.11)
 10. **end for**
 11. **end for**
-

proposed method, allows avoiding the so-called cheating in machine learning [37]. Although a total of $K \times |\mathcal{S}| \times m$ inverse models are being built through Algorithm 1, which may affect its computational cost, the tuning cost could be controlled by choosing an appropriate small set \mathcal{S} (e.g., a logarithmically spaced numbers) via a coarse and fine-tuning strategy. Also, the common choice for K is usually 2, 5, or 10 [37], as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

It is important to remark that, even though the direct implementation of the full computational model within the proposed CV scheme would provide a more precise and reliable estimation of the hyperparameter \tilde{n} , compare to the proposed approach based on a forward model $\hat{\mathcal{M}}$, such a solution could be extremely inefficient in terms of computation time in general. Indeed, since the simulation of many complicated microwave structures requires a full-wave EM solver (e.g., the application Example II section 4.5), using the full computational forward model makes the proposed Algorithm 1 inefficient. However, developing an ML-based forward model could be avoided when fast circuitual simulations (e.g., less than 1 second) are provided by a full computational forward model for the specific problem at hand.

Table 4.1 Preliminar Design of the Proposed Microwave Filter

W_1	1.0878 mm	S_1	0.9564 mm	L_1	11.2553 mm
W_2	1.1925 mm	S_2	3.3199 mm	L_2	11.0942 mm
W_3	1.1929 mm	S_3	3.8338 mm	L_3	11.0889 mm

4.5 Practical examples

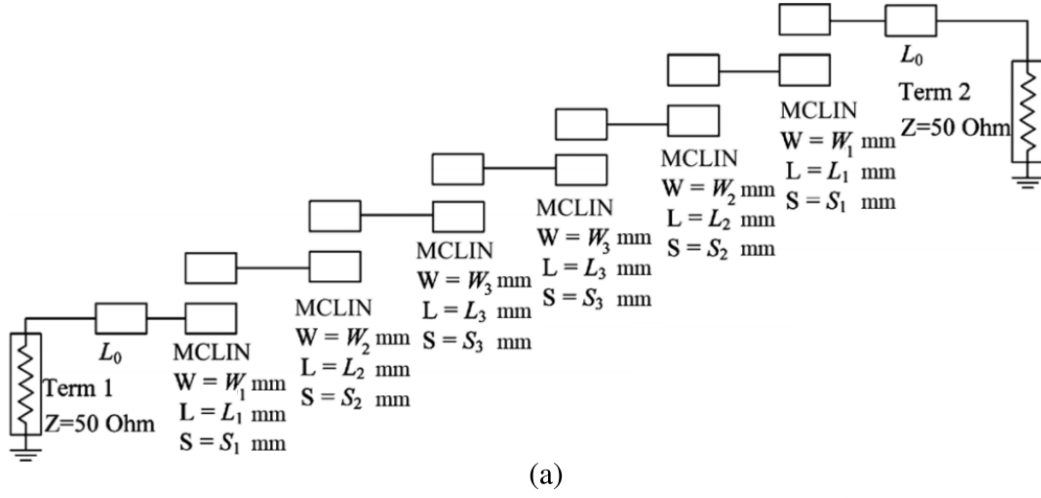
4.5.1 Example I: The Inverse Modeling of a Coupled Line Microstrip Filter

Let us consider the design of a pass-band microstrip filter with following specifications:

- Center frequency: $F_C = 2400$ MHz
- Bandwidth: $BW = 100$ MHz
- Insertion loss: $IL = 5$ dB
- Return loss: $RL = 14$ dB
- Out of band rejection: 60 dB (@ $f = 2300$ MHz).

Without loss of generality, we will focus on the 6-stage symmetric coupled-line microstrip filter structure inspired by [80], and shown in Fig. 4.4(a). The filter is designed for a 1.27 mm FR4 substrate with $\epsilon_r=9.6$, $\tan \delta=0.002$, and $\sigma = 5.78 \times 10^6$ S/m. The overall filter structure is characterized by 9 geometrical parameters, i.e., length L_i , width W_i , and separation S_i of the lines, for $i = 1, \dots, 3$. An initial design of the above parameters obtained via analytic formulas [81] is collected in Table 4.1, for which the corresponding filter frequency response is completely deviated from the given design specifications. Hence, further tuning through design optimization is required.

Our goal is to construct an inverse model $\mathbf{x} = \mathcal{M}^{-1}(\mathbf{y})$ to estimate the geometrical parameters of the filter $\mathbf{x} = [S_1, S_2, S_3, W_1, W_2, W_3, L_1, L_2, L_3]^T \in \mathbb{R}^{m=9}$ directly from the desired spectra scattering parameters $|S_{21}|$ and $|S_{11}|$ (i.e., $n_o = 2$) defined for $n=1000$ frequency points each, such that $\mathbf{y} = [|S_{11}|, |S_{21}|]^T \in \mathbb{R}^{2000}$, where $|S_{ij}| =$



$$\mathbf{x} = [S_1, S_2, S_3, W_1, W_2, W_3, L_1, L_2, L_3]^T$$

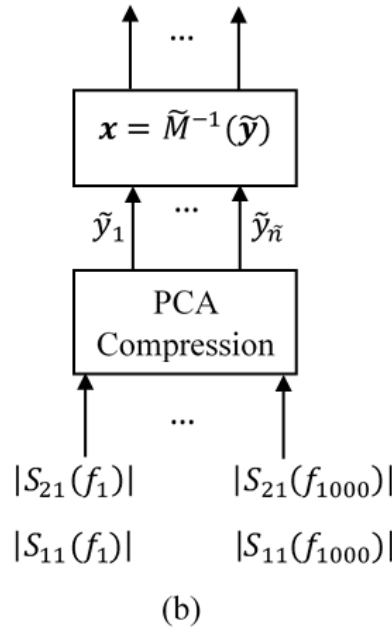


Fig. 4.4 (a) 6-stage coupled-line microstrip filter considered in Example I [81]. (b) proposed approach for design optimization based on the inverse model.

$[|S_{ij}(f_1)|, \dots, |S_{ij}(f_n)|]$. The inverse model is trained using a training set $\{(\mathbf{x}_l, \mathbf{y}_l)\}_{l=1}^L$ consisting of $L=200$ training pairs computed via a full computational forward model implemented in ADS. The configurations of the geometrical parameters corresponding to input vectors $\{\mathbf{x}_l\}_{l=1}^L$ are randomly generated via a Latin Hypercube Sampling (LHS) scheme by considering a uniform variation of 50% for W_i , S_i and

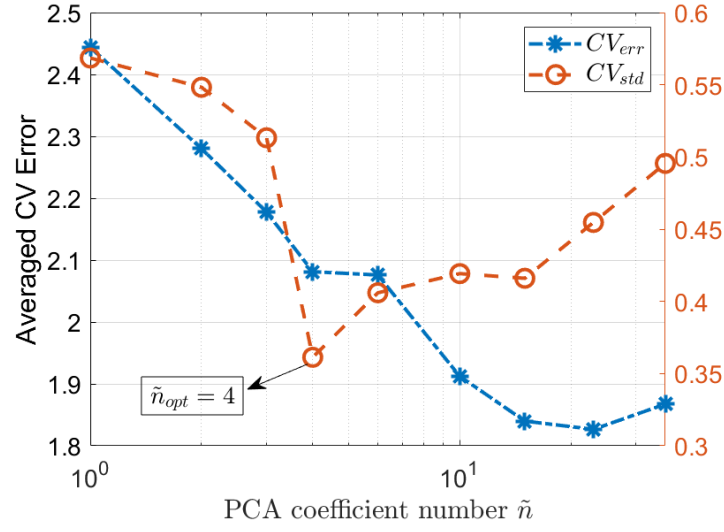


Fig. 4.5 Analysis of PCA coefficients number \tilde{n} by proposed 5-fold cross validation Algorithm 1, and with $\hat{n} = 3$. The optimum $\tilde{n}_{opt} = 4$ is selected for building the coupled line microstrip inverse model.

10% for L_i variables around the mean values specified in Table 4.1. The above configurations are then used as input for a set of parametric simulations in ADS to calculate the magnitude of the scattering parameters $|S_{21}|$ and $|S_{11}|$ collected in the set $\{\mathbf{y}_l\}_{l=1}^L$. For this practical example, each simulation to acquire the training data is evaluated in less than one second (i.e., about 0.23 s from Table 4.4), using a computer with a 3-GHz Core i7 CPU and 32-GB RAM. Such a low computational cost is mainly due to the circuitual simulations which are fortunately possible by the 2D model (i.e., MCFil [82]) in ADS to design a microstrip coupled-line filter component.

The PCA compression has been applied on the training samples $\{\mathbf{y}_l\}_{l=1}^L$. Then, the new compressed training set has been used to train the inverse model based on the LS-SVM regression. The optimal value of the hyperparameter related to the number of PCA coefficients $\tilde{n}_{opt} = 4$ is estimated by analyzing the CV error calculated via Algorithm 1. The mean and the standard deviation of the CV error is shown in Fig. 4.5, for which a set \mathcal{S} containing the candidates for \tilde{n} , logarithmically spaced in the range $[1, 40]$, is used. The CV_{err} curve shows that an \tilde{n} from 4 to 15 could be a choice for the number of PCA coefficients to be used in the inverse modeling, but the optimal number is the one with minimum CV_{std} which is $\tilde{n} = 4$. As discussed in section 4.4, an ML-based forward model $\hat{\mathcal{M}} : \mathbb{R}^9 \rightarrow \mathbb{R}^{\hat{n}=3}$ has been developed to

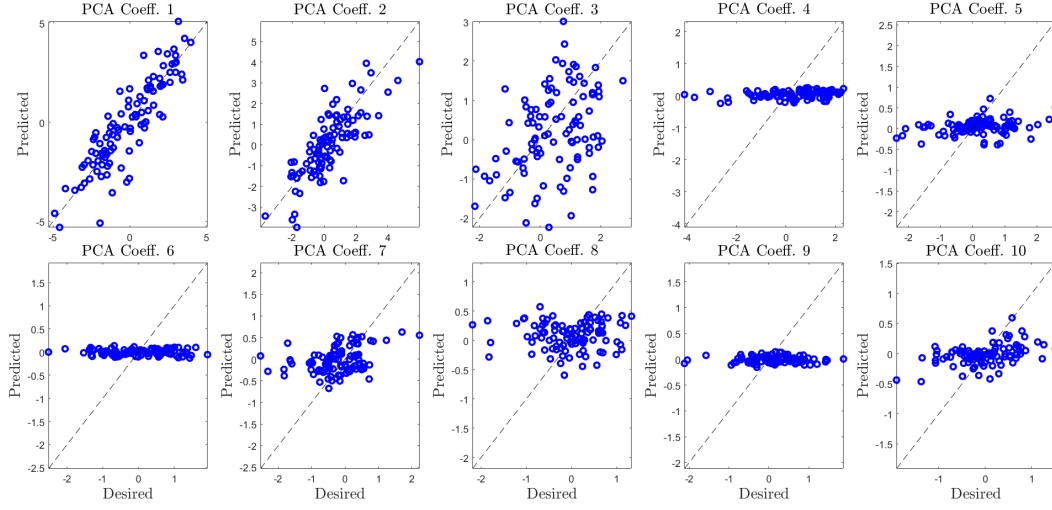


Fig. 4.6 Evaluation of the trained ML-based forward model $\mathcal{M} : \mathbb{R}^9 \rightarrow \mathbb{R}^{\hat{n}}$ of the coupled-line microstrip filter in predicting the first ten PCA coefficients (i.e., $\hat{n} = 1, 2, \dots, 10$), using 100 unseen validation samples. The LS-SVM regressions have already been trained using a separate set of 100 samples.

be used in the aforementioned CV scheme, with the parameter $\hat{n} = 3$ determined by scatter plot evaluation (shown in Fig. 4.6). As can be seen, the advocated forward model accurately approximates the behavior of the first three PCA coefficients, so the parameter \hat{n} is set to 3.

To evaluate the performance of the proposed inverse model, the model is fed with the ideal Chebyshev mask shown in Fig. 4.7 (dashed-dotted green line), incorporating all the specifications given by the problem. Then as shown in Fig. 4.2(b), the filter geometry predicted by the inverse model has been used as input for the full-computational forward model implemented in ADS. Fig. 4.7 (solid red line) shows the optimized results computed via the proposed inverse model with $\tilde{n}=4$. The results highlight the capability of the proposed compressed inverse model of providing an accurate estimation of the optimal filter geometry with improved performance with respect to the direct inverse modeling approach [77] in which the LS-SVM regression is used alone without any data compression (dotted blue line) [71].

Moreover, the results of the inverse model are compared with the ones predicted by a conventional optimization approach available in ADS [82] by considering following optimization “goals”:

- Goal 1: $|S_{21}| \leq 5$ dB in the band pass;
- Goal 2: $|S_{21}| \geq 60$ dB @ 2300 MHz;
- Goal 3: $|S_{11}| \geq 14$ dB in the band pass.

The ADS gradient optimizer[82] converged after 125 iterations in 14 min and 28 s (Table 4.4). The scattering parameters $|S_{21}|$ and $|S_{11}|$ for the optimized filter geometry based on the ADS are shown in Fig. 4.7 (dashed black line) and are reasonably in agreement with the ones obtained by the proposed inverse model. However, as compared in Table 4.4, the whole proposed inverse modeling approach (i.e., data acquisition, tuning the hyperparameters, and training) is faster for this specific application example. Filter response specifications are collected in Table 4.2.

Table 4.2 Filter response specifications optimized by inverse model and ADS optimizer

	F_C (MHz)	Insertion loss (dB)	BW_{3dB} (MHz)
ADS Optimizer	2403	4.4	125
Inverse Model	2424	5.25	99
Initial Design	2550	6.64	90

It is important to remark that different from the standard optimization approaches based on the forward model, the obtained inverse model is tunable and can be suitably adopted to optimize the filter geometry for a generic set of filter specifications, just by changing the desired mask. To this aim, its performances have been assessed by using two new specifications: (i) $F_C = 2700\text{MHz}$ and $BW = 70\text{MHz}$ and (ii) $F_C = 2550\text{MHz}$ and $BW = 50\text{MHz}$. Two new masks, generated according to the above specifications, have been used as input for the inverse model with $\tilde{n}=4$. Obviously, the model tunability is limited to the range of \mathbf{x} used during the model training as confirmed by results of the optimized geometrical parameters in Fig. 4.8(a). Then, the scattering parameters obtained from the optimal parameters predicted via the inverse model are shown in Fig. 4.8(b). The results clearly highlight the capability of the proposed inverse model of optimizing structures for different specifications.

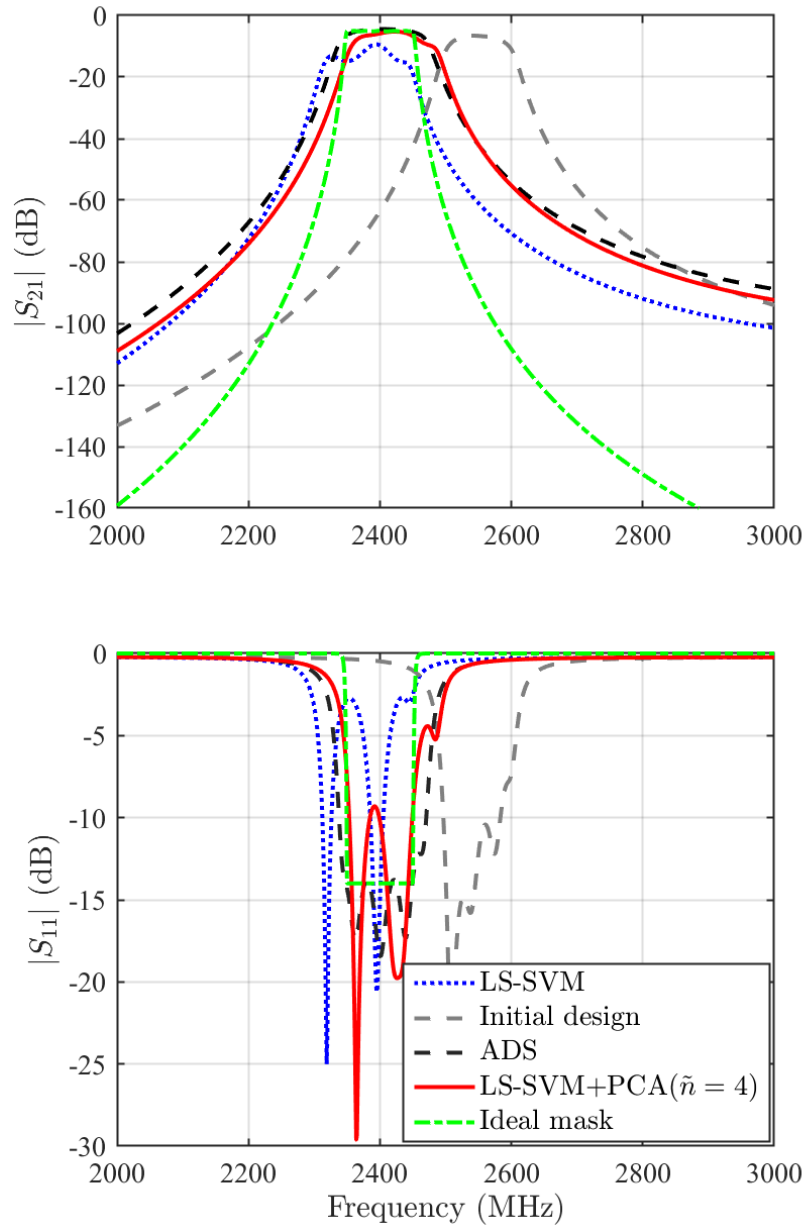
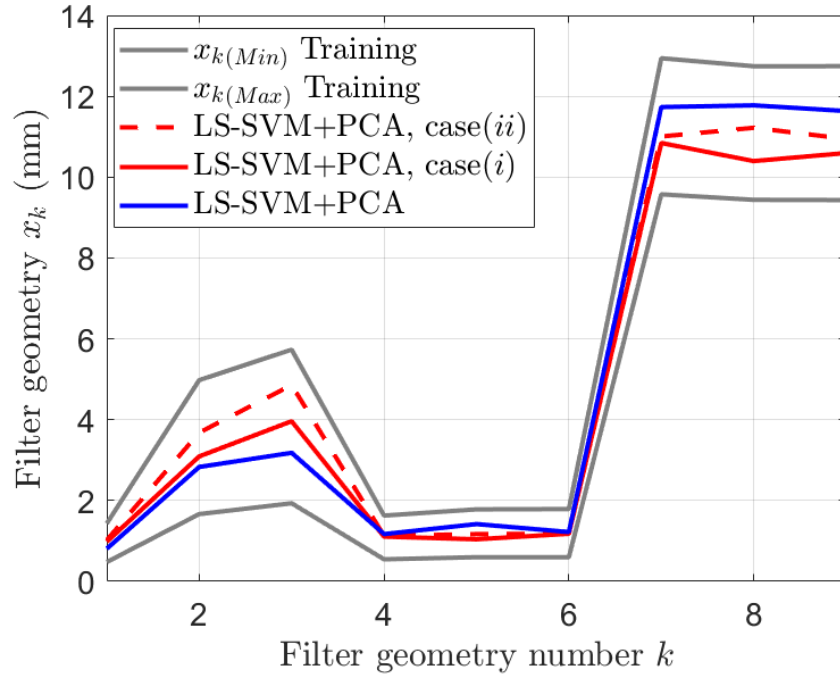


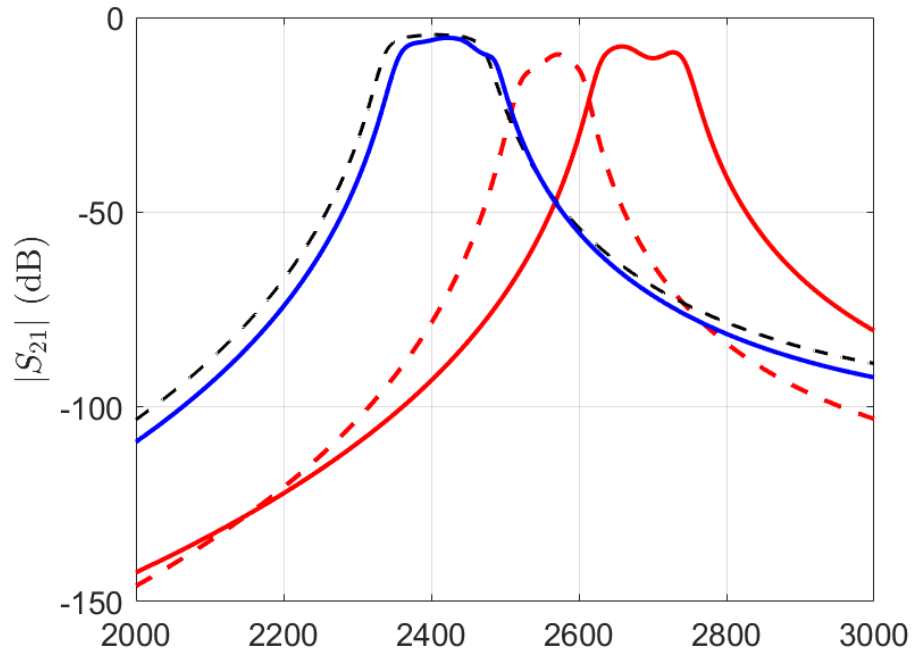
Fig. 4.7 Frequency responses of the initial design (dashed gray line) and the optimized filters for the given goal specifications in section 4.5, obtained by ADS (dashed black line), inverse model via LS-SVM (dotted blue line), and proposed compressed inverse model combining LS-SVM and PCA (solid red line), also the ideal mask used as input of the inverse model (dashed-dotted green line).

Although the geometrical parameters predicted by the inverse model (as shown in Fig. 4.8(a)) should belong to the range of variability (domain) assumed for

the design parameters, there is no guarantee that for the case of the predictions obtained by a standard LS-SVM regression problem. However, the consistency (i.e.,



(a)



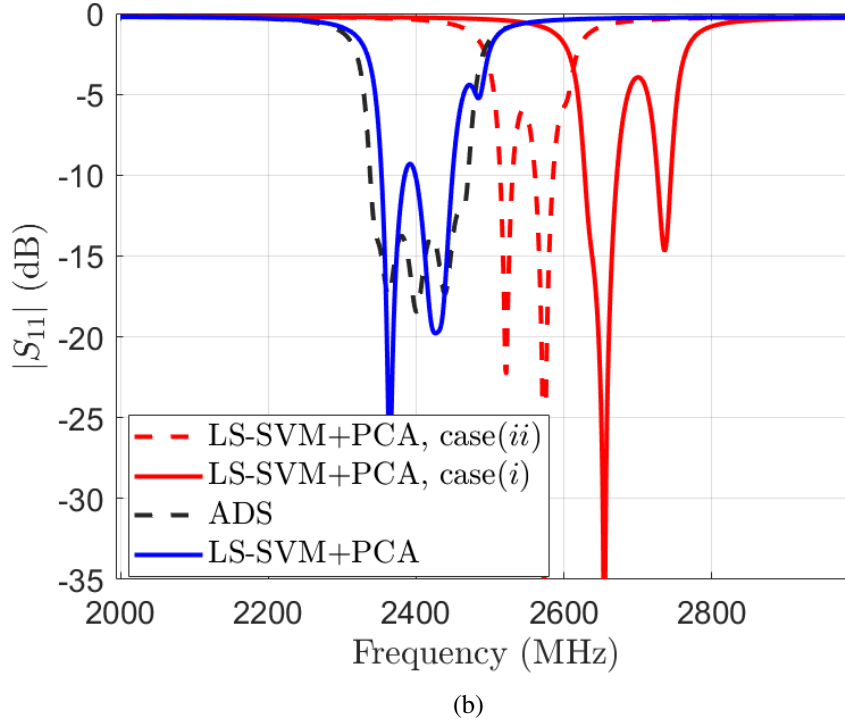


Fig. 4.8 Tunability assessment of the proposed inverse model with $\tilde{n}=4$. (a) model tunability is limited to the range of \mathbf{x} used during the model training. (b) scattering parameters of the optimized filter by changing given filter specifications in two cases: (i) $F_C = 2700\text{MHz}$ and $BW = 70\text{MHz}$ and (ii) $F_C = 2550\text{MHz}$ and $BW = 50\text{MHz}$.

$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}$) of the optimal configuration of the design parameters estimated by the inverse model can be easily checked a posteriori, also, such a constraint could be used directly during the training of the inverse model, by reformulating the training algorithm in term of constrained optimization regression. How to guarantee a physically realizable optimized solution would be a potential future topic.

4.5.2 Example II: The Inverse Modeling of a Wideband Common Mode Suppression Filter

In this example, the proposed method is further illustrated through the inverse modeling of the compact common-mode suppression filter shown in Fig. 4.9 [83]. The filter structure is characterized by 6 geometrical parameters, i.e., line width w_i , spacing s_i and length l_i of coupled microstrip lines for $i = 1, 2$, and designed

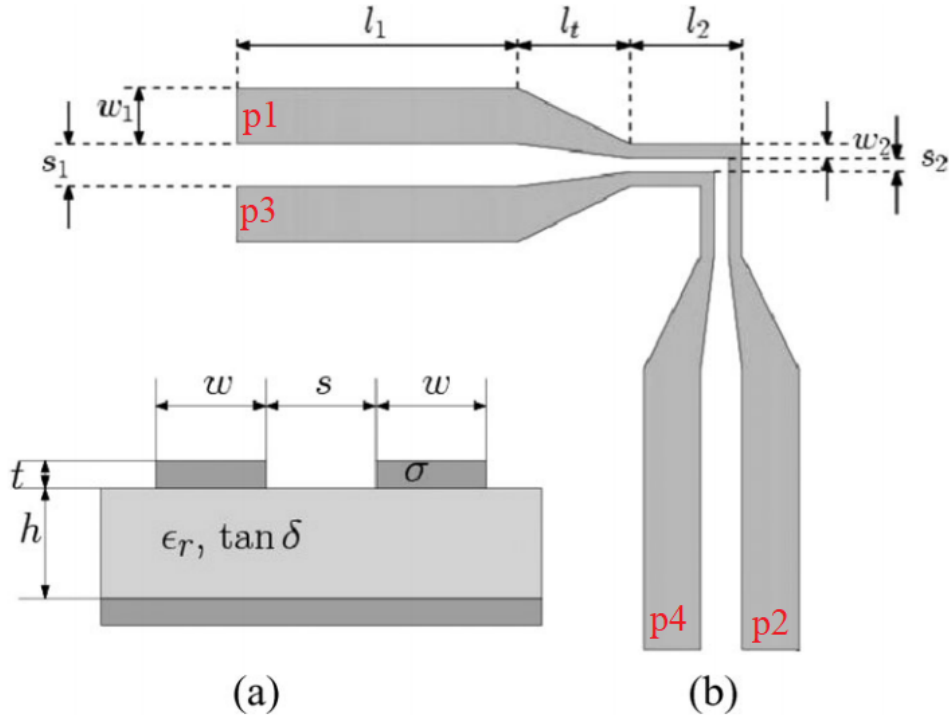


Fig. 4.9 Common-mode suppression filter with its geometrical parameters. (a) Cross-section. (b) Top view. (from [84])

on Rogers RO4350B substrate with $h=1.524$ mm, $\epsilon_r=3.66$, $\tan \delta = 0.003$, and $\sigma = 4.1 \times 10^7$ S/m as a low-loss one allowing precise simulations close to measurements. The performance of such a structure could be described using the following modal S-parameters matrix [83],

$$\mathbf{S}_m = \begin{bmatrix} S_{cc11} & S_{cc12} & S_{cd11} & S_{cd12} \\ S_{cc21} & S_{cc22} & S_{cd21} & S_{cd22} \\ S_{dc11} & S_{dc12} & S_{dd11} & S_{dd12} \\ S_{dc21} & S_{dc22} & S_{dd21} & S_{dd22} \end{bmatrix} = \begin{bmatrix} CC & CD \\ DC & DD \end{bmatrix}, \quad (4.14)$$

where modal ports 1 and 2 consist of the single ended ports p1, p3 and p2, p4 in Fig. 4.9, respectively, and could be excited by a common or differential mode signals. The matrix entries S_{ccij} and S_{ddij} are the scattering parameter related to a common or differential mode signal going from port j to i ($i, j=1,2$), respectively. Whereas, S_{cdij} refers to the conversion of a differential to common mode signal from port j to i , and S_{dcij} is the opposite conversion. Only the S_{ddij} and S_{cdij} parameters will be

Table 4.3 Initial Design of the Common Mode Suppression Filter

w_1	1.8 mm	s_1	0.7 mm	l_1	28.55 mm
w_2	0.6 mm	s_2	0.2 mm	l_2	10 mm

considered in the following optimization since coupled microstrip lines are typically utilized to transmit differential signals.

The optimization “goals” for the considered structure are the following:

- Goal 1: $|S_{cd21}| \geq 20 \text{ dB}$, $10\text{MHz} < f < 6\text{GHz}$
- Goal 2: $|S_{dd21}| \leq 0.6 \text{ dB}$, $10\text{MHz} < f < 6\text{GHz}$

In this example the inverse model uses as input the magnitude of the scattering parameters $|S_{cd21}|$ and $|S_{dd21}|$ (i.e., $n_o = 2$) defined for $n=600$ frequency points equally spaced in $[10, 6000]$ MHz, such that $\mathbf{y} = [|S_{cd21}|, |S_{dd21}|]^T \in \mathbb{R}^{1200}$, whereas it provides as output $m = 6$ geometrical parameters collected in the vector $\mathbf{x} = [s_1, w_1, l_1, s_2, w_2, l_2]^T \in \mathbb{R}^6$. Hence, in this example we are looking for an inverse model such that $\mathcal{M}^{-1} : \mathbb{R}^{1200} \rightarrow \mathbb{R}^6$.

A training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^L$ with $L=200$ is randomly generated via an LHS scheme by considering a uniform variation of 50% for w_i , s_i and l_i variables around the mean values specified in Table 4.3, taken from [84]. The corresponding scattering parameters $\{\mathbf{y}_i\}_{i=1}^L$ are computed via full-wave simulation in ADS Momentum tool. As shown in Table 4.4, training data acquisition takes 1 h 55 min for this practical example using a computer with a 3-GHz Core i7 CPU and 32-GB RAM. It should be noted that in this practical example where the full computational model (i.e., the full-wave EM model) is computationally expensive, it is necessary to develop a surrogate forward model to be efficiently used in the proposed double CV scheme Algorithm 1. To this end, an ML-based forward model $\hat{\mathcal{M}} : \mathbb{R}^6 \rightarrow \mathbb{R}^{\hat{n}}$ has been developed with the parameter $\hat{n}=5$ determined by scatter plots evaluation.

Having the aforementioned dataset, the proposed cross validation Algorithm 1 is used to find the optimum number of PCA coefficients to be $\tilde{n}_{opt}=3$ as shown in Fig. 4.10. Then, the 6 inverse models in (4.8) have been built by using the compressed training set $\{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^L$ in which $\tilde{n}=3$ PCA coefficients in (4.3) are used to approximate original set.

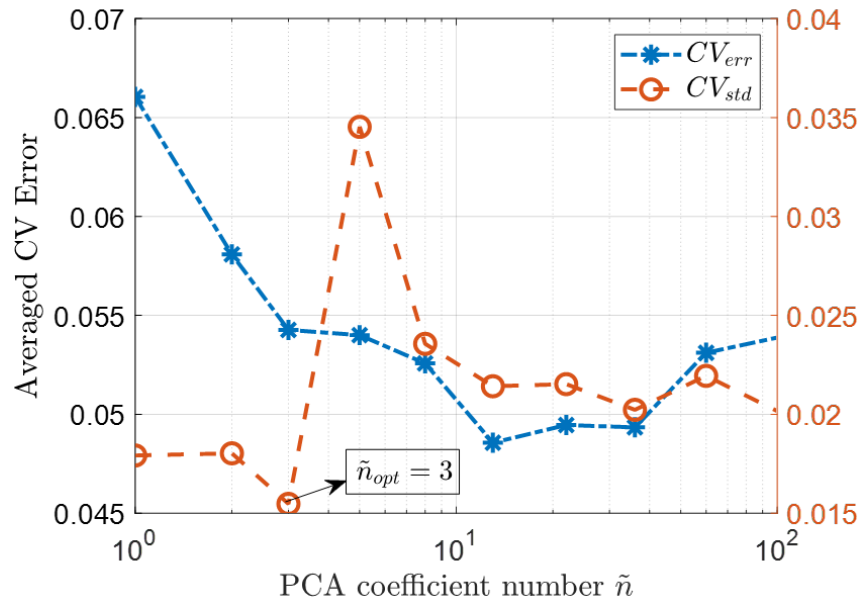
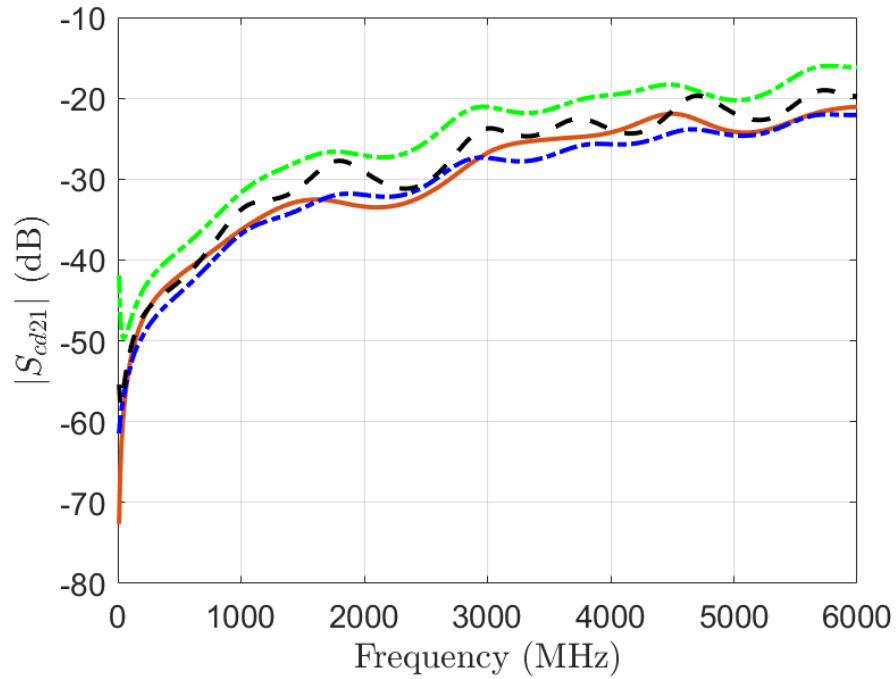


Fig. 4.10 Analysis of PCA coefficients number \tilde{n} by proposed 5-fold cross validation Algorithm 1, and with $\hat{n} = 5$. The optimum $\tilde{n}_{opt} = 3$ is selected for building the common mode suppression filter inverse model.



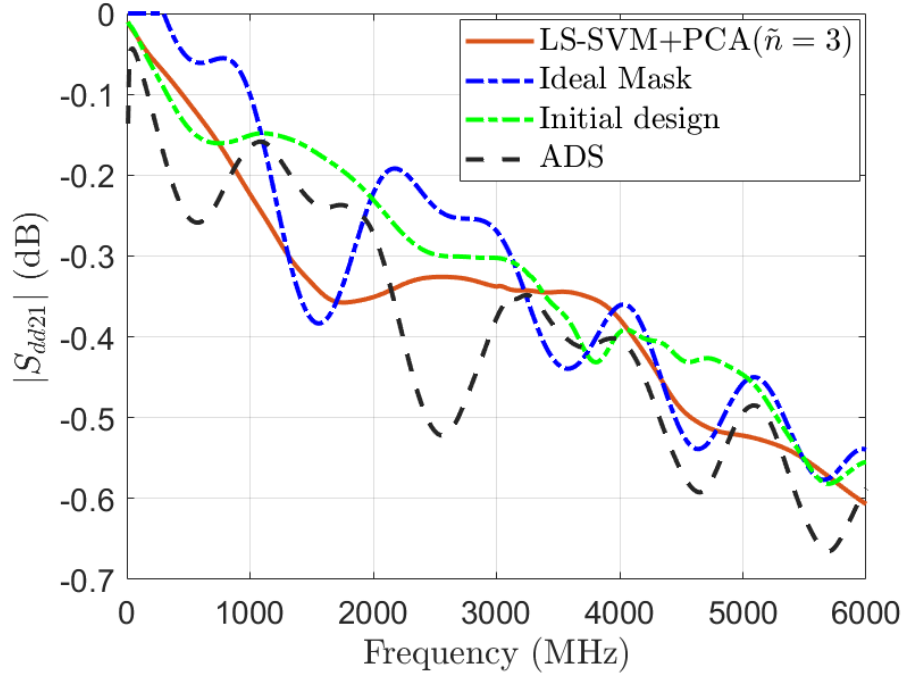


Fig. 4.11 Frequency responses of the initial (dashed-dotted green line) and the optimized design for the common mode suppression filter, obtained by ADS (dashed black line), and the proposed inverse model combining LS-SVM and PCA (solid red line). Also, the ideal goal mask used as input for the inverse model (dashed-dotted blue line).

To evaluate the performance of the proposed inverse model, the ideal mask shown in Fig. 4.11 (dashed-dotted blue line), calculated starting from the average of the training responses based on the design goals, has been used as input for the trained inverse models. Then as shown in Fig. 4.2(b), the optimal filter geometry estimated by the inverse models is simulated via ADS to get corresponding scattering parameters shown in Fig. 4.11 (solid red line). Also in this example, the performances of the proposed inverse model have been compared with the ones of the conventional ADS optimizer based on the full-wave simulations run in the Agilent ADS Momentum tool. The ADS gradient solver[82] has been converged after 14 iterations in 1 h and 42 min (Table 4.4). The S-parameters $|S_{cd21}|$ and $|S_{dd21}|$ obtained by the ADS optimizer and by the proposed inverse model are shown in Fig. 4.11. The ADS results (dashed black line) show a good agreement with the given specifications, even if a small violation of Goal 2 (i.e., $|S_{dd21}| \leq 0.6dB$) is observed at the end of the frequency band. As an overall metric of common-mode voltage suppression, the average value of $|S_{cd21}|$ over the frequency range for initial design, ADS optimizer,

and inverse model prediction are 24.97, 28.1 and 29.85 dB, respectively, which shows the 1.75 dB superiority of the common mode suppression ability of filter designed by the inverse model compared to ADS optimizer result. On the other hand, the results of the inverse model (solid red line) highlight once again the capability of the proposed approach of providing a design satisfying the optimization goals.

From Table 4.4, we can find that even though the proposed inverse modeling approach is overall more expensive than the direct optimization by ADS optimizer, the major part of this cost is due to the data acquisition via full-wave simulations in this particular example. Indeed, considering the fact that this step (i.e., data acquisition) could be accelerated using the parallelized computation techniques [60], the proposed inverse model itself is extremely efficient compared to the conventional direct optimization techniques. It is important to remark that, for the sake of a specific and technically solid comparison of the computational costs, the same initial design values and the same variation of design parameters around them are used for design optimizations implemented on the same computer (i.e., 3-GHz Core i7 CPU and 32-GB RAM).

In both application examples, $L=200$ training samples are used to build the inverse models. Generally, the amount of data required for different examples depends on many factors: the complexity of the problem, the complexity of the learning algorithm, etc. To the best of our knowledge, there is not a consolidated scheme for automatically selecting the number of training samples so far. However, the LS-SVM regression has good generalization capability when training data are limited. Here, considering the performance and computational cost of the proposed Algorithm 1, we have chosen $L=200$ and $K=5$ which results in a validation and training set of size 40 and 160, respectively, to be used in each iteration of a 5-fold CV scheme. Moreover, by varying L and looking at the output of Algorithm 1, a lower bound for the L values could be defined such that finding an optimal \tilde{n}_{opt} is possible. Based on our experiments, the minimum required L for both application examples is $L=110$.

Table 4.4 The Computational Cost of the Inverse Modeling and Direct Optimization of Practical Examples in section 4.5

	Example I	Example II
Dataset acquisition (200 samples)	46 s	1 h 55 min
Forward model development	10 s	10 s
Proposed Algorithm 1	3 min	2 min 25 s
Inverse model training	7 s	5 s
Direct optimization (Gradient)	14 min 28 s	1 h 42 min

Chapter 5

Conclusion

This research is conducted in two parts. In the first part, a mathematical method for developing a rational transfer function model (i.e., rational macromodel) in practical applications with the aim of low complexity was presented. In the proposed method, the data obtained from the measurement/simulation is approximated in a closed-form rational function which is easily synthesizable to an equivalent circuit (netlist) compatible with SPICE-like circuit solvers. To this end, a rational macromodel based on data integration is proposed with the aim of utilizing all available information, reducing the order of the model and relative resistance to noise. Finally, the stability of the model is guaranteed by using the closed-loop control technique and considering a control coefficient. The performance evaluation of the proposed method in comparison with the well-known VF method has been investigated in several practical examples. The results show that the performance of the proposed rational macromodel is less affected by noise in the data, and this is an important feature because in most cases the measurement data is usually accompanied by noise. The main goal is to develop efficient algorithms that model high-speed interconnects to be used in a circuit-based solver such as ADS.

In the second part, a completely different approach called "Inverse Modeling" is introduced to optimize passive microwave components, which, unlike conventional optimization methods based on iterative algorithms, is very efficient and does not require either an external optimizer algorithm or repetition. In this context, an efficient and compact inverse model based on the machine learning regression technique is proposed to address two major challenges in the inverse modeling of

microwave devices, namely the ill-posed problem and high-dimensional input space issues.

The proposed inverse model is constructed by combining the LS-SVM regression with the PCA compression technique, which allows estimating the optimal geometry of microwave devices directly from the desired frequency domain mask without the need for computationally expensive iterative optimization algorithms. In addition, a new method based on cross validation is proposed to automatically determine the optimal number of PCA coefficients to be used for decreasing the input space dimensionality of the inverse problem at hand. In comparison to existing data-driven inverse modeling techniques, this makes our method more general for design optimization in microwave applications. The latter is critical for overcoming both the high-dimensional input space and the ill-posed challenges in the proposed inverse modeling approach. A distinctive feature of the proposed inverse modeling approach over existing methods is that it could be constructed using a limited dataset (e.g., 200 samples as shown in Table 4.4). The feasibility, accuracy, and tunability of the proposed optimization scheme were investigated using two examples in microwave applications by comparing its predictions with those computed using a commercial solver.

As a data-driven model, the general attractive feature of the proposed inverse model are:

- It is based on data obtained from the system of interest (usually through full-wave EM simulation in the case of high-frequency structures), hence no prior knowledge of the system of interest is required.
- It is generic, thus it can be used in a variety of microwave applications that use S-parameters.
- Since the model is based on explicit analytical formulations, It is computationally inexpensive to evaluate.

Specifically, the proposed inverse model has several advantages making it generic to be used in microwave applications. First, the proposed inverse model is a data-driven model, which uses data obtained from the system of interest. Second, the inverse model is constructed based on S-parameters computed at many frequency points (as commonly available data in various microwave applications) and there is

no need to extract the frequency response features. Lastly, an automated algorithm based on a cross-validation technique is proposed to find the optimum parameters of the model by using only a limited dataset. However, once the inverse model is built, a frequency domain ideal mask (incorporating all the specifications given by the problem) should be used as input for the inverse model to predict the optimal geometry of the microwave component.

5.1 Future work

The advantages and challenges of the proposed inverse model are summarized as follows:

- A compact and efficient macromodel to design optimization of microwave components without any requirement of external repetitive algorithms
- Due to using the S-parameters, the proposed inverse model could be generalized in various microwave applications, not depending on response features
- The capability of solving multi-objective optimization microwave problems by using vectorized multiple S-parameter responses
- An inverse model for high-dimensional and ill-posed microwave problems
- Efficient inverse data-driven model for applications with a relatively high number of design variables and minimum available data set
- The predictive power of the model is limited to the range of geometrical parameters training set
- The algorithm used for tuning the number of PCA coefficients is not efficient in complex applications
- Unlike conventional optimizers, in the proposed inverse approach, the weighting of design specifications (response features) is not directly possible.

Although the inverse model proposed in this thesis yielded good results and proved to be effective in microwave applications. The followings things can be considered as future work in this framework.

1. Increasing the predictive power of the inverse model by reformulation the regression problem (i.e., (4.7)) as a constrained optimization with a pre determined range of geometrical parameters.
2. Constraining the initial dataset to relatively ideal responses using the domain confinement [85] to improve the proposed inverse model accuracy and efficiency.
3. Assessment and development of the proposed inverse model in applications with nonlinear and active microwave components (such as mixers, oscillators, power amplifiers, etc.) considering their special features such as transient instability, dispersion, etc.
4. Using the proposed inverse model in combination with the ML classification techniques [86] in applications with multiple structural microwaves such as nanophotonics, metasurface, etc.
5. Development of an inverse model in order to use the training set obtained from measurements, considering two specific characteristics of noise and the limited number of frequency samples. In such cases, other information like phase response, delay, etc. may be used along with amplitudes.

References

- [1] Stefano Grivet-Talocia and Bjorn Gustavsen. *Passive macromodeling: Theory and applications*. John Wiley & Sons, 2015.
- [2] Bart Boesman, Georges GE Gielen, Guy AE Vandenbosch, and Davy Pisssoort. Spherical wave based macromodels for efficient system-level emc analysis in circuit simulators part i: Optimized derivation and truncation criteria. *IEEE Transactions on Electromagnetic Compatibility*, 58(5):1494–1505, 2016.
- [3] T Bradde, P Toledo, M De Stefano, A Zanco, S Grivet-Talocia, and P Croveti. Enabling fast power integrity transient analysis through parameterized small-signal macromodels. In *2019 International Symposium on Electromagnetic Compatibility-EMC EUROPE*, pages 759–764. IEEE, 2019.
- [4] Bjorn Gustavsen and Adam Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on power delivery*, 14(3):1052–1061, 1999.
- [5] Dirk Deschrijver, Michal Mrozowski, Tom Dhaene, and Daniel De Zutter. Macromodeling of multiport systems using a fast implementation of the vector fitting method. *IEEE Microwave and wireless components letters*, 18(6):383–385, 2008.
- [6] Sanda Lefteriu and Athanasios C Antoulas. A new approach to modeling multiport systems from frequency-domain data. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(1):14–27, 2009.
- [7] Muhammad Kabir and Roni Khazaka. Macromodeling of distributed networks from frequency-domain data using the loewner matrix approach. *IEEE transactions on microwave theory and techniques*, 60(12):3927–3938, 2012.
- [8] Stefano Grivet-Talocia and Riccardo Trinchero. Behavioral, parameterized, and broadband modeling of wired interconnects with internal discontinuities. *IEEE Transactions on Electromagnetic Compatibility*, 60(1):77–85, 2017.
- [9] Slawomir Koziel and Anna Pietrenko-Dabrowska. *Performance-driven surrogate modeling of high-frequency structures*. Springer Nature, 2020.

- [10] Taha A Elwi, Zaid Asaad Abdul Hassain, and Omar Almukhtar Tawfeeq. Hilbert metamaterial printed antenna based on organic substrates for energy harvesting. *IET Microwaves, Antennas & Propagation*, 13(12):2185–2192, 2019.
- [11] Slawomir Koziel, Adrian Bekasiewicz, Piotr Kurgan, and John W Bandler. Rapid multi-objective design optimisation of compact microwave couplers by means of physics-based surrogates. *IET Microwaves, Antennas & Propagation*, 10(5):479–486, 2016.
- [12] Ahmad Khusro, Mohammad S Hashmi, Abdul Quaiyum Ansari, Aditya Mishra, and Mohammad Tarique. An accurate and simplified small signal parameter extraction method for gan hemt. *International Journal of Circuit Theory and Applications*, 47(6):941–953, 2019.
- [13] Slawomir Koziel, Anna Pietrenko-Dabrowska, et al. Design-oriented two-stage surrogate modeling of miniaturized microstrip circuits with dimensionality reduction. *IEEE Access*, 8:121744–121754, 2020.
- [14] Sotirios Goudos. *Microwave systems and applications*. InTech, 2017.
- [15] Mohammad GH Alijani and Mohammad H Neshati. A new closed-form expression for dispersion characteristics of fundamental mode of siw by least squares method. *The Applied Computational Electromagnetics Society Journal (ACES)*, pages 930–933, 2015.
- [16] Jerry M Mendel. *Lessons in estimation theory for signal processing, communications, and control*. Pearson Education, 1995.
- [17] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.
- [18] Giulio Antonini. Spice equivalent circuits of frequency-domain responses. *IEEE transactions on electromagnetic compatibility*, 45(3):502–512, 2003.
- [19] Jorge Fernández Villena and Luís Miguel Silveira. Circuit synthesis for guaranteed positive sparse realization of passive state-space models. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(6):1576–1587, 2017.
- [20] David Hilbert. *Methods of mathematical physics*. CUP Archive, 1930.
- [21] Mohammad GH Alijani and Mohammad H Neshati. Development a new technique based on least square method to synthesize the pattern of equally space linear arrays. *International Journal of Engineering*, 32(11):1620–1626, 2019.
- [22] Victor Y Pan. Solving a polynomial equation: some history and recent progress. *SIAM review*, 39(2):187–220, 1997.

- [23] Amir Beygi and Anestis Dounavis. An instrumental variable vector-fitting approach for noisy frequency responses. *IEEE transactions on microwave theory and techniques*, 60(9):2702–2712, 2012.
- [24] Nastaran Soleimani, Mohammad GH Alijani, and Mohammad H Neshati. Crosstalk analysis of multi-microstrip coupled lines using transmission line modeling. *International Journal of RF and Microwave Computer-Aided Engineering*, 29(6):e21677, 2019.
- [25] Piero Triverio, Stefano Grivet-Talocia, and Michel S Nakhla. A parameterized macromodeling strategy with uniform stability test. *IEEE Transactions on Advanced Packaging*, 32(1):205–215, 2009.
- [26] Dirk Deschrijver and Tom Dhaene. Parametric macromodeling of time domain responses. In *2008 12th IEEE Workshop on Signal Propagation on Interconnects*, pages 1–2. IEEE, 2008.
- [27] P Triverio, S Grivet-Talocia, and Michel S Nakhla. An improved fitting algorithm for parametric macromodeling from tabulated data. In *2008 12th IEEE Workshop on Signal Propagation on Interconnects*, pages 1–4. IEEE, 2008.
- [28] P Triverio, Michel S Nakhla, and S Grivet-Talocia. Parametric macromodeling of multiport networks from tabulated data. In *2007 IEEE Electrical Performance of Electronic Packaging*, pages 51–54. IEEE, 2007.
- [29] Alessandro Zanco and Stefano Grivet-Talocia. Toward fully automated high-dimensional parameterized macromodeling. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(9):1402–1416, 2021.
- [30] Qi-Jun Zhang, Kuldip C Gupta, and Vijay K Devabhaktuni. Artificial neural networks for rf and microwave design-from theory to practice. *IEEE transactions on microwave theory and techniques*, 51(4):1339–1350, 2003.
- [31] Feng Feng, Chao Zhang, Jianguo Ma, and Qi-Jun Zhang. Parametric modeling of em behavior of microwave components using combined neural networks and pole-residue-based transfer functions. *IEEE Transactions on Microwave Theory and Techniques*, 64(1):60–77, 2015.
- [32] Slawomir Koziel and Adrian Bekasiewicz. Fast simulation-driven feature-based design optimization of compact dual-band microstrip branch-line coupler. *International Journal of RF and Microwave Computer-Aided Engineering*, 26(1):13–20, 2016.
- [33] Slawomir Koziel and Anna Pietrenko-dabrowska. Surrogate modeling for high-frequency design: Recent advances. 2022.
- [34] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005.

- [35] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. *European journal of operational research*, 192(3):707–716, 2009.
- [36] Johan AK Suykens, Tony Van Gestel, and Jos De Brabanter. *Least squares support vector machines*. World scientific, 2002.
- [37] Benyamin Ghojogh and Mark Crowley. The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial. *arXiv preprint arXiv:1905.12787*, 2019.
- [38] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.
- [39] Jerome H Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. springer open, 2017.
- [40] CE Rasmussen and CKI Williams. Gaussian processes for machine learning. the mit press, cambridge, massachusetts, usa, london, 2006.
- [41] Géraud Blatman and Bruno Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25(2):183–197, 2010.
- [42] Johan AK Suykens. Nonlinear modelling and support vector machines. In *IMTC 2001. proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)*, volume 1, pages 287–294. IEEE, 2001.
- [43] Junping Wang, Quanshi Chen, and Yong Chen. Rbf kernel based support vector machine with universal approximation and its application. In *International symposium on neural networks*, pages 512–517. Springer, 2004.
- [44] Craig Saunders, Alexander Gammernan, and Volodya Vovk. Ridge regression learning algorithm in dual variables. 1998.
- [45] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.
- [46] You Ling and Sankaran Mahadevan. Quantitative model validation techniques: New insights. *Reliability Engineering & System Safety*, 111:217–231, 2013.
- [47] Humayun Kabir, Lei Zhang, Ming Yu, Peter H Aaen, John Wood, and Qi-Jun Zhang. Smart modeling of microwave devices. *IEEE Microwave Magazine*, 11(3):105–118, 2010.
- [48] Li-Ye Xiao, Wei Shao, Xiao Ding, and Bing-Zhong Wang. Dynamic adjustment kernel extreme learning machine for microwave component design. *IEEE Transactions on Microwave Theory and Techniques*, 66(10):4452–4461, 2018.

- [49] G Angiulli, M Cacciola, and M Versaci. Microwave devices and antennas modelling by support vector regression machines. *IEEE Transactions on Magnetism*, 43(4):1589–1592, 2007.
- [50] Peng Chen, Brian M Merrick, and Thomas J Brazil. Support vector regression for harmonic optimization in continuous class-f power amplifier design. In *2015 Integrated Nonlinear Microwave and Millimetre-wave Circuits Workshop (INMMiC)*, pages 1–3. IEEE, 2015.
- [51] Peng Chen and Thomas J Brazil. Optimization-oriented method for broadband doherty power amplifier designs using support vector regression. In *2018 IEEE MTT-S International Microwave Workshop Series on 5G Hardware and System Technologies (IMWS-5G)*, pages 1–3. IEEE, 2018.
- [52] Feng Feng, Chao Zhang, Qi-Jun Zhang, Jianguo Ma, et al. Parallel space-mapping approach to em optimization. *IEEE transactions on microwave theory and techniques*, 62(5):1135–1148, 2014.
- [53] G Pepe, F-J Gortz, and H Chaloupka. Computer-aided tuning and diagnosis of microwave filters using sequential parameter extraction. In *2004 IEEE MTT-S International Microwave Symposium Digest (IEEE Cat. No. 04CH37535)*, volume 3, pages 1373–1376. IEEE, 2004.
- [54] Jing Jin, Chao Zhang, and Qi-Jun Zhang. An enhanced multi-circuit optimization approach for identification of microwave filter parameters. In *2016 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pages 1–2. IEEE, 2016.
- [55] Anna Pietrenko-Dabrowska and Slawomir Koziel. Accelerated multiobjective design of miniaturized microwave components by means of nested kriging surrogates. *International Journal of RF and Microwave Computer-Aided Engineering*, 30(4):e22124, 2020.
- [56] Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23(9):3137–3166, 2019.
- [57] Humayun Kabir, Ying Wang, Ming Yu, and Qi-Jun Zhang. Neural network inverse modeling and applications to microwave filter design. *IEEE Transactions on Microwave Theory and Techniques*, 56(4):867–879, 2008.
- [58] Slawomir Koziel and Adrian Bekasiewicz. Expedited geometry scaling of compact microwave passives by means of inverse surrogate modeling. *IEEE Transactions on Microwave Theory and Techniques*, 63(12):4019–4026, 2015.
- [59] Slawomir Koziel, Adrian Bekasiewicz, and Leifur Leifsson. Rapid em-driven antenna dimension scaling through inverse modeling. *IEEE Antennas and Wireless Propagation Letters*, 15:714–717, 2015.

- [60] Wei Zhang, Feng Feng, Jianan Zhang, Zhihao Zhao, Jianguo Ma, and Qi-Jun Zhang. Parallel decomposition approach to wide-range parametric modeling with applications to microwave filters. *IEEE Transactions on Microwave Theory and Techniques*, 68(12):5288–5306, 2020.
- [61] Matjaž Depolli, Roman Trobec, and Bogdan Filipič. Asynchronous master-slave parallelization of differential evolution for multi-objective optimization. *Evolutionary computation*, 21(2):261–291, 2013.
- [62] Chao Zhang, Jing Jin, Weicong Na, Qi-Jun Zhang, and Ming Yu. Multivalued neural network inverse modeling and applications to microwave filters. *IEEE Transactions on Microwave Theory and Techniques*, 66(8):3781–3797, 2018.
- [63] Jing Jin, Chao Zhang, Feng Feng, Weicong Na, Jianguo Ma, and Qi-Jun Zhang. Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters. *IEEE Transactions on Microwave Theory and Techniques*, 67(10):4140–4155, 2019.
- [64] Martin Benning and Martin Burger. Modern regularization methods for inverse problems. *arXiv preprint arXiv:1801.09922*, 2018.
- [65] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.
- [66] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.
- [67] Andreas Hauptmann and Ben T Cox. Deep learning in photoacoustic tomography: Current approaches and future directions. *Journal of Biomedical Optics*, 25(11):112903, 2020.
- [68] Hao-Jie Hu, Li-Ye Xiao, Jun-Nan Yi, and Qing Huo Liu. Nonlinear electromagnetic inversion of damaged experimental data by a receiver approximation machine learning method. *IEEE Antennas and Wireless Propagation Letters*, 2021.
- [69] Jing Jin, Feng Feng, Weicong Na, Shuxia Yan, Wenyuan Liu, Lin Zhu, and Qi-Jun Zhang. Recent advances in neural network-based inverse modeling techniques for microwave applications. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 33(6):e2732, 2020.
- [70] Mostafa Sedaghat, Riccardo Trinchero, Zaker Hossein Firouzeh, and Flavio G Canavero. Compressed machine learning-based inverse model for design optimization of microwave components. *IEEE Transactions on Microwave Theory and Techniques*, 2022.

- [71] Mostafa Sedaghat, Riccardo Trinchero, and Flavio Canavero. Compressed machine learning-based inverse model for the design of microwave filters. In *2021 IEEE MTT-S International Microwave Symposium (IMS)*, pages 13–15. IEEE, 2021.
- [72] Slawomir Koziel. Fast simulation-driven antenna design using response-feature surrogates. *International Journal of RF and Microwave Computer-Aided Engineering*, 25(5):394–402, 2015.
- [73] Vahid Yaghoubi, Stefano Marelli, Bruno Sudret, and Thomas Abrahamsson. Sparse polynomial chaos expansions of frequency response functions using stochastic frequency transformation. *Probabilistic engineering mechanics*, 48:39–58, 2017.
- [74] Paolo Manfredi and Riccardo Trinchero. A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses. *IEEE Access*, 8:92019–92027, 2020.
- [75] Version LS-SVMLab. 1.8; department of electrical engineering (esat), 2011.
- [76] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.
- [77] R Trinchero, M Ahadi Dolatsara, K Roy, M Swaminathan, and FG Canavero. Design of high-speed links via a machine learning surrogate model for the inverse problem. In *2019 Electrical Design of Advanced Packaging and Systems (EDAPS)*, pages 1–3. IEEE, 2019.
- [78] Benyamin Ghogh and Mark Crowley. Unsupervised and supervised principal component analysis: Tutorial. *arXiv preprint arXiv:1906.03148*, 2019.
- [79] Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, 5(Sep):1089–1105, 2004.
- [80] Yanling Hao, Bingfa Zu, and Ping Huang. An optimal microstrip filter design method based on advanced design system for satellite receiver. In *2008 IEEE International Conference on Mechatronics and Automation*, pages 903–907. IEEE, 2008.
- [81] Jia-Shen G Hong and Michael J Lancaster. *Microstrip filters for RF/microwave applications*, volume 167. John Wiley & Sons, 2004.
- [82] Tuning, optimization and design, statistical, May 2003.
- [83] Celina Gazda, Ivo Couckuyt, Hendrik Rogier, Dries Vande Ginste, and Tom Dhaene. Constrained multiobjective optimization of a common-mode suppression filter. *IEEE transactions on electromagnetic compatibility*, 54(3):704–707, 2012.

- [84] Celina Gazda, Dries Vande Ginste, Hendrik Rogier, Ruey-Beei Wu, and Daniël De Zutter. A wideband common-mode suppression filter for bend discontinuities in differential signaling using tightly coupled microstrips. *IEEE transactions on advanced packaging*, 33(4):969–978, 2010.
- [85] Slawomir Koziel, Anna Pietrenko-Dabrowska, and Ubaid Ullah. Low-cost modeling of microwave components by means of two-stage inverse/forward surrogates and domain confinement. *IEEE Transactions on Microwave Theory and Techniques*, 69(12):5189–5202, 2021.
- [86] Xuehua Lu, Wenbin Li, Zhihui Zhu, Yongqiang Hu, Ziyi Tang, Wenpeng Zhang, Ke Liu, Yarong Su, Jie Zheng, Weidong Chen, et al. Classification and inverse design of metasurface absorber in visible band. *Advanced Theory and Simulations*, 5(3):2100338, 2022.
- [87] Mostafa Sedaghat, Zaker Hossein Firouzeh, and Hadi Aliakbarian. Development of a non-iterative macromodeling technique by data integration and least square method. *International Journal of Engineering*, 34(11):2408–2417, 2021.

Appendix A

List of Contributions

1. **Development of a Non-Iterative Macromodeling Technique by Data Integration and Least Square Method [87].**

M Sedaghat, ZH Firouzeh, H Aliakbarianb

International Journal of Engineering (IJE) 34 (11), 10

Abstract: In this paper, a new method is introduced to synthesize the original data obtained from simulation or measurement results in the form of a rational function. The integration of the available data is vital to the performance of the proposed method. The values of poles and residues of the rational model are determined by solving the system of linear equations using the conventional Least Square Method (LSM). To ensure the stability condition of the provided model, a controller coefficient is considered. Also, using this parameter, the designer can increase the stability margin of a system with poor stability conditions. The introduced method has the potential to be used for a wide range of practical applications since there is no specific restriction on the use of this method. The only requirement that should be considered is the Dirichlet condition for the original data, usually the case for physical systems. To verify the performances of the proposed method, several application test cases are investigated and the obtained results are compared with those gathered by the well-known vector fitting algorithm. Also, the examinations show that the method is efficient in the presence of noisy data.

2. **Compressed Machine Learning-Based Inverse Model for the Design of Microwave Filters [71].**

M Sedaghat, R Trinchero, F Canavero

2021 IEEE MTT-S International Microwave Symposium (IMS)

Abstract: This paper presents an inverse model for the optimization of the geometrical parameters of a parallel coupled-line pass-band filter. Given the overall structure of the filter, the least square support vector machine is combined with the principal component analysis with the aim of constructing an inverse model able to estimate the geometrical parameters of the filter starting from a frequency-domain mask. Such model is trained via a set of scattering parameters computed via a 2D solver for few configurations of the filter geometrical parameters. The feasibility and the accuracy of the proposed optimization scheme is investigated by comparing its predictions with the corresponding optimal configuration estimated via a commercial tool.

3. **Compressed Machine Learning-Based Inverse Model for Design Optimization of Microwave Components [70].**

M Sedaghat, R Trinchero, ZH Firouzeh, F Canavero

IEEE Transactions on Microwave Theory and Techniques, 13

Abstract: This article presents a new noniterative inverse modeling technique based on machine learning regression and its applications to microwave design optimization. The proposed inverse model accepts the high-dimensional S-parameters computed at many frequency points as the input and estimates the optimal geometrical/physical parameters of the microwave component as its output. The least-squares support vector machine regression is combined with the principal component analysis to simultaneously overcome both the high-dimensional input space and ill-posed challenges of the inverse modeling. We also propose a new empirical method to find the optimum number of principal components (i.e., the compression level) for each example in an automated way. This makes our proposed model general and easy to use compared with the existing data-driven inverse modeling techniques. The inverse model is trained by a set of scattering parameters computed via a 2-D/3-D solver for few configurations of the geometrical parameters. The feasibility and the accuracy of the proposed optimization scheme are investigated by comparing its predictions with the corresponding optimal configuration estimated via a commercial solver.