Macromodeling and Simulation of Linear Components Characterized by Measured Parameters

Mingyang Zhang

Department of Electrical & Computer Engineering
McGill University, Montreal, Canada

February 2008

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Your file Votre référence
ISBN: 978-0-494-51480-1
Our file Notre référence
ISBN: 978-0-494-51480-1

Dedication

This thesis is dedicated to my beloved parents
Jie-Sheng Yang and *Lei Zhang*
my dear fiancée *Zhuo Li*
and my uncle *Jing-hua Yang*
for their everlasting love and support.

Abstract

Recently, microelectronics designs have reached extremely high operating frequencies as well as very small die and package sizes. This has made signal integrity an important bottleneck in the design process, and resulted in the inclusion of signal integrity simulation in the computer aided design flow. However, such simulations are often difficult because in many cases it is impossible to derive analytical models for certain passive elements, and the only available data are frequency-domain measurements or full-wave simulations. Furthermore, at such high frequencies these components are distributed in nature and require a large number of poles to be properly characterized. Simple lumped equivalent circuits are therefore difficult to obtain, and more systematic approaches are required. In this thesis we study the Vector Fitting techniques for obtaining such equivalent model and propose a more streamlined approach for preserving passivity while maintaining accuracy.

Sommaire

De nos jours, les conceptions microélectroniques atteignent de très hautes fréquences d'opération ainsi que des tailles infimes de dés et de boîtiers. Ceci a fait de l'intégrité du signal un paramètre critique dans le processus de conception, dont la simulation a du être intégrée à la conception assistée par ordinateur. Cependant de telles simulations sont souvent difficiles car dans beaucoup des cas il est impossible de dériver des modèles analytiques pour certains éléments passifs, et les seules données disponibles sont des mesures en fréquence ou des simulations d'oscillation complète. De plus, à de telles fréquences, ces composants sont de nature distribuée et requièrent l'usage de plusieurs pôles pour être convenablement caractérisés. Il est donc difficile d'obtenir de simple circuits groupés équivalents, il faut recourir a des approches bien plus systématiques. Dans cette thèse, nous étudions les techniques de Vector Fitting pour obtenir de tels modèles équivalents et proposons une approche plus rationalisée pour conserver la passivité tout en maintenant la précision.

Acknowledgments

It is my great pleasure to thank my supervisor, Professor Roni Khazaka, for his support. It is he who introduced me to circuit simulation technology. With his help, I quickly became familiar with this field and its latest technologies. He taught me efficient methods for debugging computer programs, which has made my work much more efficient. He helped generate the simulated data used in this thesis. His innovative work inspires me to think and practice as engineer, and his contributions to my research are far more than I can list here.

Mr. Moufid Radji helped me with French abstract translation. I much appreciate his generous help.

I want to thank my fiancée Zhuo Li for her encouragement on my research. She makes my everyday in Montréal rich and cheerful. With her support to my life, I can fully focus on my work and drive the progress in a faster pace.

Finally, I would like to express my highest gratitude to my parents for what they have sacrificed for me. I thank them for supporting my decision of pursuing graduate studies, both spiritually and financially. They are always beside me whenever I have difficulties. I cannot achieve this without their loving support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

With the increasing complexity and operation frequency of modern very large scale integrated circuits (VLSI) and application specific integrated circuits (ASIC), signal integrity issues, which used to be negligible, have become a critical consideration in recent design flows. Signal integrity degradation of digital signal occurs in the path from the driver to the receiver, which includes interconnects, vias and other discontinuities. Such effects are typically negligible at low frequencies. However, in high-speed circuitries, undesired or incorrect signals can be generated from multiple sources like crosstalk, reflections due to impedance mismatch, ringing, delay, and attenuation [4–24]. These effects result in signal integrity violation and lead to faulty circuits and reduced performance. In fact, signal integrity has become a performance bottleneck in modern circuit design. It must, therefore, be taken into consideration as early as possible in the design cycle in order to avoid costly prototyping and unnecessary design iterations. Therefore it is important to have access to accurate models as well as efficient simulation and optimization tools, which can take into account interconnect effects and signal integrity degradation [21, 22, 25–27].

In addition to making interconnect effects and signal integrity degradation a dominant factor in the design process, high operating frequencies lead to significant difficulties in the simulation of such structures. This is due to the fact that at high frequencies interconnects must be treated as distributed structures and modeled using partial differential equations (PDE). In order to include such components in a standard time-domain circuit simulator

such as SPICE, the PDEs must be discretized to obtain ordinary differential equations, and this leads to a very large number of equations and a high CPU cost. In order to overcome this simulation bottleneck, model order reduction methods (MOR) were proposed in the literature. The general idea behind these techniques is to reduce the large linear blocks of the system into small macromodels which contain only a few dominant poles determining the response up to the highest frequency of interest. The main MOR methods can be divided into two main categories. The first is direct moment matching methods which are based on padé approximation [4,13,28–30]. These techniques include asymptotic waveform evaluation (AWE) and complex frequency hopping (CFH) [31,32]. The second category is based on indirect projection. Methods based on the Arnoldi process, truncated balanced realization (TBR) [33] and singular value decomposition (SVD) [34] are found in this category. These methods became popular due to their ability to conserve passivity by construction. However, although these methods significantly reduce the CPU cost of simulation, they all operate on system of first order differential equations and therefore assume that analytical models for the interconnects and other passive elements and discontinuities are available. Unfortunately, this is not always the case, and in many practical high-frequency applications the only available model consists of frequency-domain data of multi-port parameters obtained from measurement or from full-wave analysis. The simulation difficulty in this case is not due to the presence of distributed elements, but also to the fact that these elements are modeled using frequency-domain measured data, which cannot be directly incorporated into time domain nonlinear SPICE-type simulation. The methods for addressing this issue can be divided into two general approaches. The first approach is to use numerical convolution to evaluate the time-domain stamp of the distributed elements at each time point. We will refer to this class of methods as convolution-based approaches. The main drawbacks of such techniques are the high CPU cost and memory requirement of the numerical convolution which must be performed at each time step with the initial condition as the starting point of the integration. This type of approaches can also exhibit accuracy and stability problems. The second class of methods for handling elements defined by measured parameters is to perform some preprocessing in order to come up with a time-domain macromodel of the system in the form of poles and zeros. Once such a macromodel is obtained, recursive convolution [28,35] can be used instead of regular convolution in order to improve the CPU and memory performance. However, more commonly, the time-domain representation is either directly stamped into the modi-

fied nodal analysis equations or used to synthesize an equivalent circuit. Such an approach eliminates the mixed frequency time domain problem, which results from the presence of frequency-domain measured parameters. The main challenge in this case is that of obtaining a suitable time-domain macromodel in the form of poles and residues. Such a macromodel must of course match the frequency-domain parameters of the original circuit, but it also must not only be stable but also passive. The stability of the macromodel is easily enforced by techniques such as [32,36,37]. These techniques either simply delete the unstable poles or shift them to the left half-plane and allow extra iterations to retrieve the accuracy. However, a macromodel which is stable but not passive can result in unstable simulations depending on the termination [7,22,38]. Enforcing passivity to a macromodel is not a trivial task. In this thesis we will study the leading method for obtaining such a macromodel and for ensuring its passivity, and we will propose improvements for the passivity enforcement algorithms. The resulting passive macromodels can be easily simulated using standard simulators such as SPICE and are automatically generated without any need for physical knowledge of the passive structure.

1.2 Organization of the thesis

The thesis is organized in the following order: In Chapter 2, literature review is carried out. Conventional techniques such as convolution-based approximation and macromodel-based approaches are discussed. In Chapter 3, an algorithm called Vector Fitting is introduced to generate macromodels without suffering computational difficulties. Chapter 4 deals with passivity verification. Upon passivity violations, a global passivity enforcement algorithm is applied for compensation. Linear optimization with special constraints is used to minimize the degradation of accuracy. Details are discussed in Chapter 5. The second half of Chapter 5 presents numerical examples to illustrate the efficiency of the algorithm. Transient simulations are also carried out for evaluating the accuracy in time domain. A potential problem associated with the algorithm is discussed in Chapter 5 and a corresponding solution is given.

Chapter 2

Literature Review

2.1 Background Introduction

As discussed in Chapter 1, in real high-speed networks, it is not always easy to find closed form analytical models for the interconnects. Instead, the interconnect behaviors are typically described by frequency-domain tabulated parameters obtained from measurement or full-wave simulation. In order to include the tabulated data in time-domain simulators, there are generally two categories of approaches one can follow. The first category contains convolution-based approaches. In these methods, the inverse Fourier transform is applied to the frequency-domain parameters to get their time-domain counterparts. The time-domain results are then convolved with the arbitrary sources to evaluate the time-domain terminal voltages or currents. Although these methods are straightforward, their computer implementation has a major limitation. The convolution process leads to a high CPU cost, especially when the network has a large number of ports. In addition, since accuracy is closely related to the number of time steps, higher accuracy requires a larger number of time steps and more iterations of convolution are therefore spent on the simulation. The details of the convolution-based approaches will be discussed in Section 2.2. The methods from the second category focus on constructing time-domain macromodels for the interconnects. The most common way of macromodel construction is based on rational function approximation. This field has been studied extensively in [1–3, 24, 39, 40]. Some typical techniques will be covered later in Section 2.3. The general idea behind these approaches is to approximate the measured/simulated parameters by a rational function in its polynomial form and calculate all the coefficients of the polynomial by solving a least-square

problem. The resultant macromodels can be implemented in time-domain simulators either through generating a SPICE compatible model consisting of equivalent circuit components or through recursive convolution [28,35]. The typical difficulty associated with these methods is that the least-square problem is easily ill-conditioned. This difficulty has become a bottleneck of its computer implementation, as the ill-conditioned matrix may go beyond the floating-point range of the computer. Many studies have been done to fix this problem [1,3,28], but they still have problem with high-order approximations. A new technique named *VectorFitting* has been proposed to overcome this difficulty [41]. This technique follows an iterative procedure to evaluate the poles of the network and the resulting least-square problem is well conditioned. A thorough review of Vector Fitting will be given in Chapter 3.

2.2 Convolution-Based Approach

2.2.1 General Concept

As mentioned in [1], the implementation of interconnects represented by the measured data into circuit simulators is computationally intensive, since nearly all interconnects are terminated by nonlinear components, such as diodes and transistors. The analysis of the linear interconnects is carried out in frequency domain, whereas, the analysis of nonlinear networks is achieved in time domain. Thus, the method must be able to combine the solutions in both domains [1,40]. The most straightforward solution is to apply the inverse fast Fourier transform (IFFT) to calculate the impulse response of the network. Then the general solution is given by convolving the impulse response with the arbitrarily given input voltage. This method is referred to as convolution-based method. It calculates the time-domain responses directly from the frequency-domain measured data. The implementation details are introduced in [40] and summarized as follows.

2.2.2 Algorithm Details

Consider a linear M-port network (shown in Fig.2.1). Suppose an ideal voltage source is connected to port j and all other $M-1$ ports are grounded. Then the current in frequency

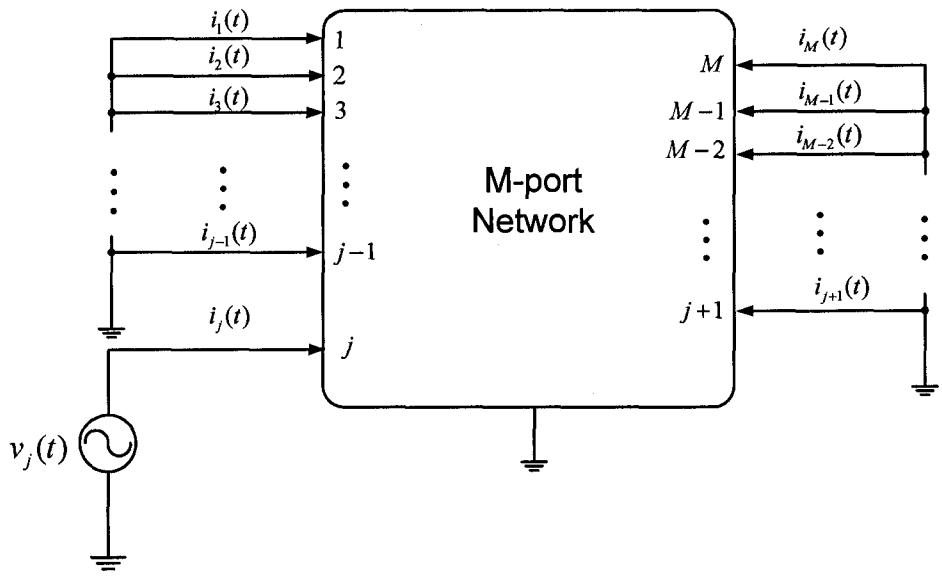

Fig. 2.1 A linear M-port network

domain at each port can be expressed as (2.1).

$$I_k(\omega) = Y_{kj}(\omega)V_{j0}(\omega); \text{ for } k=1,\ldots,M \tag{2.1}$$

where $V_{j0}(\omega)$ is the Fourier transform of $v_{j0}(t)$ and $Y_{kj}(\omega)$ is the admittance Y-parameter. A unit delta function is forced at port j such that $V_{j0}(\omega) = 1$. By doing this, the frequency-domain voltage $V_{j0}(\omega)$ is independent of frequency. Thus, it can be eliminated from (2.1) and the time domain current can be directly computed as

$$i_k(t) = i_{gkj}(t) = F^{-1}\left(Y_{kj}(\omega)\right), \tag{2.2}$$

where F^{-1} denotes the inverse Fourier transform. This set of currents is referred as the network Green's function. It is to be noted that these currents are equal to the Y-parameters in time domain regardless of input voltage, since the input voltage has been forced to be unit. In order to calculate the response due to an arbitrary input function, one needs to convolve the resulting transient response (Green's function) at the port with the arbitrary input:

$$i_k(t) = F^{-1}\left(Y_{kj}(\omega)V_{j0}(\omega)\right) = i_{gkj}(t) * v_{j0}(t) = \int_0^t i_{gkj}(t-\tau)v_{j0}(\tau)d\tau. \tag{2.3}$$

The sign "*" denotes convolution. According to the superposition principle, the total current at a specific port is the sum of the currents due to the voltage source connected to each port. Thus, we have

$$i_k(t) = \sum_{j=1}^{M} \int_0^t i_{gkj}(t-\tau) v_{j0}(\tau) d\tau \tag{2.4}$$

The calculation of the Y-parameters in (2.2) is done in the following steps. Firstly, an ideal delta-function generator is connected to one port and with all other ports shorted to ground. Secondly, find the frequency-domain current appearing at the port. Then the Green's function is calculated by applying inverse Fourier transform to the resulting current. This process should be repeated for all ports. It is to be noted that the currents should be evaluated numerically at a finite number of frequencies. The Green's functions should be also discretized in the time domain.

For the ease of computer implementation, the integrals in (2.3) should be replaced by summations:

$$i_k(t_q) = \sum_{j=1}^{M} \sum_{p=0}^{q} i_{gkj}(t_q - t_P) v_{t_p} \Delta t; \text{ for } k=1,\ldots,M \tag{2.5}$$

where the argument t_q refers to the instant of $q\Delta t$ when the currents and voltages are sampled.

The voltages are evaluated from the current-voltage relation depending on the terminal components. For example, a diode has an $I-V$ characteristics

$$i_k(t) \propto \left(exp\left(\frac{v_k(t)}{V_T}\right) - 1 \right) \tag{2.6}$$

the resulting voltages are evaluated from the above relation.

The convolution-based method is the most straightforward method to obtain the time-domain response from the frequency-domain data, since it provides a direct link between the frequency-domain data and its time-domain counterpart. However, as discussed in [40,42],

to obtain the port currents due to different port voltages, the Green's functions have to be convolved with the arbitrary functions. The convolutions are done numerically and considered to be the most CPU-expensive process, as the convolution must be done at each time step with the initial condition as the starting point of the integration. To reduce this cost, one has to keep the number of frequency points to be sampled as low as possible. This can cause restrictions on the speed of the circular convolutions, as the time spent on computing the response should not exceed a few line transit times in presence of terminal networks.

2.3 Macromodel-Based Methods

2.3.1 General Concept

Instead of convolution-based approaches, macromodel-based methods feature indirect evaluation of the time-domain response by constructing analytical macromodels for the interconnects. Polynomials are the most common way for function approximation. However, rather than polynomials, the macromodels are typically in the form of rational function. The reason for that is polynomials are not a suitable method to represent the response of electrical networks, as polynomials have limitations with representing the behaviors of the interconnects in the vicinity of poles. In contrast, rational function displays the strength of well capturing the behavior around the poles, and its partial fraction expansion form is among the most suitable approximation methods for distributed systems. Poles and residues of the system can be directly found in the partial fraction format. Then the transient response can be easily obtained by performing recursive convolution or through its equivalent time-domain macromodel. The macromodeling process may take some CPU time, but with the closed form expression of the system time-domain responses are easily calculated. Macromodel-based approaches are more CPU-efficient and, therefore, exhibit a big advantage over convolution-based approaches. Some typical techniques will be discussed later in this section.

2.3.2 Least-Square Approaches [1–3]

A network's function $H(s)$ can be expressed by a rational function in rational function form:

$$H_{m,n}(s) = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_m s^m}{1 + b_1 s + b_2 s^2 + \cdots + b_n s^n} \tag{2.7}$$

where b_0 is normalized. For a given set of sampled data at k frequency points, the task is to match (2.7) with the given data at specific frequencies:

$$\frac{Q_m(s_i)}{P_n(s_i)} = y_i; \text{ for } i=0,\ldots,k-1 \tag{2.8}$$

where y_i denotes the frequency-domain data measured at the ith frequency point and k denotes the number of sampling points. (2.8) can be rewritten in a linear format:

$$Q_m(s_i) - y_i P_n(s_i) = 0 \tag{2.9}$$

Further expanding (2.9) at each frequency point, we end up with a set of equations in the matrix form:

$$\underbrace{\begin{bmatrix} 1 & s_0 & s_0^2 & \ldots & s_0^m & -s_0y_0 & -s_0^2y_0 & \ldots & -s_0^ny_0 \\ 1 & s_1 & s_1^2 & \ldots & s_1^m & -s_1y_1 & -s_1^2y_1 & \ldots & -s_1^ny_1 \\ \vdots & & & & & & & & \\ 1 & s_{k-1} & s_{k-1}^2 & \ldots & s_{k-1}^m & -s_{k-1}y_{k-1} & -s_{k-1}^2y_{k-1} & \ldots & -s_{k-1}^ny_{k-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \end{bmatrix}}_{Y}. \tag{2.10}$$

Usually the number of frequency points k is greater than the number of coefficients (number of columns in **V**). Thus (2.10) is a full-rank over-determined equation. Since electrical network functions are analytic functions of complex variables, their real and imaginary parts satisfy Cauchy-Riemann equations. From this relation, one only needs to approximate

the real part of the rational function and the whole function can be found automatically. The real part of a rational function (2.7) can be extracted by multiplying its numerator and denominator polynomials by the complex conjugate of its denominator (i.e. $Q^*(s)$). respectively. The resulting real part is in the form

$$Re(H_{m,n}(s)) = \frac{c_0 + c_1 s^2 + c_2 s^4 + \cdots + c_m s^{2m}}{1 + p_1 s^2 + p_2 s^4 + \cdots + p_n s^{2n}} \tag{2.11}$$

The coefficients are calculated by matching (2.11) with the real part of the tabulated data at each measured frequency point

$$\underbrace{\begin{bmatrix} 1 & \omega_0^2 & \omega_0^4 & \dots & \omega_0^{2m} & -\omega_0^2 y_0^r & -\omega_0^4 y_0^r & \dots & -\omega_0^{2n} y_0^r \\ 1 & \omega_1^2 & \omega_1^4 & \dots & \omega_1^{2m} & -\omega_1^2 y_1^r & -\omega_1^4 y_1^r & \dots & -\omega_1^{2n} y_1^r \\ \vdots & & & & & & & & \\ 1 & \omega_{k-1}^2 & \omega_{k-1}^4 & \dots & \omega_{k-1}^{2m} & -\omega_{k-1}^2 y_{k-1}^r & -\omega_{k-1}^4 y_{k-1}^r & \dots & -\omega_{k-1}^{2n} y_{k-1}^r \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} y_0^r \\ y_1^r \\ \vdots \\ y_{k-1}^r \end{bmatrix}}_{b} \tag{2.12}$$

where the superscript "r" indicates the real part of a complex value. 2.12 is a least-square problem. In order to solve it without direct elimination, which can cause numerical difficulty, QR factorization is applied to matrix **A** on the LHS. Thus, (2.12) is reformed into

$$\mathbf{RX} = \mathbf{Q}^T\mathbf{b} \tag{2.13}$$

Once the coefficients are calculated, the denominator of (2.11) needs to be factored to obtain the poles of the network. The solution only contains stable poles since the roots are determined in terms of squared poles. All pure imaginary poles are rejected from the pole

set. Only stable poles are retained for constructing the partial fraction expansion of $H(s)$:

$$H(s) = k_\infty + \sum_{i=1}^{n'} \frac{k_i}{s + p_i} \tag{2.14}$$

where $n' < n$, and $n - n'$ is the number of imaginary poles discarded from the process.

The residues (k_i's in (2.14)) are calculated by equating (2.14) to the sampled data at each frequency point and matching the real and imaginary parts of the LHS with the corresponding parts of the RHS respectively. According to the above discussion, a set of linear equations can be written as

$$\begin{bmatrix} 1 & \frac{-p_1}{\omega_0^2 + p_1^2} & \frac{-p_2}{\omega_0^2 + p_2^2} & \cdots & \frac{-p_n}{\omega_0^2 + p_n^2} \\ 1 & \frac{-p_1}{\omega_1^2 + p_1^2} & \frac{-p_2}{\omega_1^2 + p_2^2} & \cdots & \frac{-p_n}{\omega_1^2 + p_n^2} \\ \vdots & & & & \\ 1 & \frac{-p_1}{\omega_{k-1}^2 + p_1^2} & \frac{-p_2}{\omega_{k-1}^2 + p_2^2} & \cdots & \frac{-p_n}{\omega_{k-1}^2 + p_n^2} \\ 0 & \frac{-\omega_0}{\omega_0^2 + p_1^2} & \frac{-\omega_0}{\omega_0^2 + p_2^2} & \cdots & \frac{-\omega_0}{\omega_0^2 + p_n^2} \\ \vdots & & & & \\ 0 & \frac{-\omega_{k-1}}{\omega_{k-1}^2 + p_1^2} & \frac{-\omega_{k-1}}{\omega_{k-1}^2 + p_2^2} & \cdots & \frac{-\omega_{k-1}}{\omega_{k-1}^2 + p_n^2} \end{bmatrix} \begin{bmatrix} k_\infty \\ k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} y_0^r \\ y_1^r \\ \vdots \\ y_{k-1}^r \\ y_1^i \\ \vdots \\ y_{k-1}^i \end{bmatrix}. \tag{2.15}$$

As discussed in [1,2], (2.12) is at high risk of ill-condition, as the powers of the frequencies are doubled from those of (2.10). This leads to a huge difference between the minimum and maximum entries in each row of $\mathbf{A}$ in (2.12). To overcome this difficulty, a transformation factor is introduced to make the problem better conditioned:

$$\omega' = 2\frac{(\omega - \omega_{min})}{(\omega_{max} - \omega_{min})} - 1. \tag{2.16}$$

Scaling the frequency points by using (2.16) can map them into the domain of $[-1, 1]$. This mapping normalizes the wide frequency range to the center of the numerical range of the computer. As a consequence, this manipulation also avoids the nth power of the frequency from exceeding the floating-point range of the computer [2]. The total computational cost of the method is one polynomial factorization, two QR factorizations (for the

poles and residues calculations) and two backsubstitutions.

The next step is to obtain the time-domain macromodel. As pointed out in [1, 3], the time-domain macromodel is obtained by applying a recursive convolution to avoid the computation-intensive explicit convolution. The direct relation between the time domain and the frequency domain is the Laplace transformation:

$$Y(s) = \frac{k_i}{s + p_i} X(s) \leftrightarrow \frac{d}{dt} y(t) + p_i y(t) = k_i x(t). \tag{2.17}$$

The excitation $x(t)$ is assumed to be piecewise constant: $x(t) = c$, where $t_{n-1} \leq t \leq t_n$. (2.17) is solved using the value of $x(t)$ at current time interval as boundary conditions. The time-domain solution expressed in the recursive convolution formula for the model of (2.14) is

$$y(t_n) = k_\infty x(t_n) + \sum_{i=1}^{q} \tilde{y}_i(t_n), \tag{2.18}$$

where

$$\tilde{y}_i(t_n) = k_i(1 - e^{-p_i(t_n - t_{n-1})}) x(t_{n-1}) + e^{-p_i(t_n - t_{n-1})} \tilde{y}_i(t_{n-1}). \tag{2.19}$$

The above equation can be translated into an equivalent circuit consisting of a constant conductance, k_∞, and a current source, $-\sum_{i=1}^{q} \tilde{y}_i(t_n)$, which is updated in each iteration.

2.3.3 Macromodeling through Nonlinear Approximation

In contrast to the method of linear approximation reviewed from last section, [43] introduces a different algorithm based on solving for a nonlinear problem to obtain the coefficients of the rational functions. In this algorithm, the unknowns are included in a nonlinear equation and solved iteratively.

Assume we are given a set of tabulated data in frequency domain $(H(s_i); i = 1, 2, \ldots, N)$ obtained from full-wave simulations or measurements. The objective is to approximate the existing data by a rational function and extract a macromodel for the original network

$$Y(\mathbf{a}, \mathbf{b}, s) = \frac{b_0 + b_1 s + \cdots + b_n s^n}{1 + a_1 s + \cdots + a_m s^m} \tag{2.20}$$

where a_0 is normalized to unit. The coefficients are included in two vectors: $\mathbf{a} = [a_1, a_2, \ldots, a_n]^T$, $\mathbf{b} = [b_0, b_1, \ldots, b_m]^T$. The error of the approximation in (2.20) is evaluated by

$$E(\mathbf{a}, \mathbf{b}, s) = Y(\mathbf{a}, \mathbf{b}, s) - H(s), \tag{2.21}$$

at each measured frequency point. The minimization of (2.21) implies the minimization of the norm of the total error

$$\| E(\mathbf{a}, \mathbf{b}, s) \|_2 = \sqrt{\sum_{i=1}^{N} \| Y(\mathbf{a}, \mathbf{b}, s) - H(s) \|_2^2}. \tag{2.22}$$

Minimizing (2.22) can be further translated into minimizing
$f(\mathbf{x}) = \frac{1}{2} \| E(\mathbf{a}, \mathbf{b}, s) \|_2^2 = \frac{1}{2} \tilde{E}(\mathbf{a}, \mathbf{b}, s)^T \tilde{E}(\mathbf{a}, \mathbf{b}, s)$, where $\tilde{E} = [Re(E^T); Im(E^T)]$ and $\mathbf{x}^T = [\mathbf{a}^T; \mathbf{b}^T]$. By introducing $\tilde{E}$, all entries of the vectors $\mathbf{a}$ and $\mathbf{b}$ are real numbers so that the system avoids having complex time-domain responses. The resulting problem in (2.22) is solved by using a Gauss-Newton type method which uses a model $\tilde{M}$ for $\tilde{E}$ around $\mathbf{x}_c$

$$\tilde{M}(\mathbf{x}_c) = \tilde{E}(\mathbf{x}_c) + J(\mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c), \tag{2.23}$$

where $J(\mathbf{x}_c)$ is the Jacobian of $\tilde{E}$ at $\mathbf{x}_c$. The method minimizes $\| \tilde{E}(\mathbf{x}_c) + J(\mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c) \|_2$ in each step, subject to that the norm of the difference between the solutions from two successive iterations does not exceed some upper limit:

$$\| \delta \mathbf{x} \|_2 \leq \delta_c. \tag{2.24}$$

From (2.21), it is obvious that the Jacobian of $\tilde{E}$ is as same as that of $Y(\mathbf{a}, \mathbf{b}, s)$ separated into real and imaginary parts. From this fact, solving (2.23) is equivalent to solving

$$\underbrace{\begin{bmatrix} Re(\frac{1}{\mathbf{D}_{k-1}}\mathbf{U}_{0:m}) & \vdots & Re(-\frac{\mathbf{Y}_{k-1}}{\mathbf{D}_{k-1}}\mathbf{U}_{1:n}) \\ Im(\frac{1}{\mathbf{D}_{k-1}}\mathbf{U}_{0:m}) & \vdots & Im(-\frac{\mathbf{Y}_{k-1}}{\mathbf{D}_{k-1}}\mathbf{U}_{1:n}) \end{bmatrix}}_{A} \begin{bmatrix} \Delta\mathbf{b} \\ \Delta\mathbf{a} \end{bmatrix} = \underbrace{\begin{bmatrix} Re(Y_{k-1}(s_1) - H(s_1)) \\ \vdots \\ Re(Y_{k-1}(s_N) - H(s_N)) \\ Im(Y_{k-1}(s_1) - H(s_1)) \\ \vdots \\ Im(Y_{k-1}(s_N) - H(s_N)) \end{bmatrix}}_{b}, \quad (2.25)$$

where $\mathbf{U}_{0:m}$ is a Vandermonde matrix with $m+1$ columns containing discrete frequency points with monotonically increasing power

$$\mathbf{U}_{0:m} = \begin{bmatrix} 1 & s_1 & s_1^2 & \dots & s_1^m \\ 1 & s_2 & s_2^2 & \dots & s_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & s_N & s_N^2 & \dots & s_N^m \end{bmatrix}. \quad (2.26)$$

$\mathbf{D}_{k-1}(s)$ is a diagonal matrix

$$\mathbf{D}_{k-1}(s) = \begin{bmatrix} 1 + \sum_{i=1}^{n} a_i s_1^i & 0 & \dots & 0 \\ 0 & 1 + \sum_{i=1}^{n} a_i s_2^i & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & 1 + \sum_{i=1}^{n} a_i s_N^i \end{bmatrix}, \quad (2.27)$$

and $\mathbf{Y}_{k-1}(s)$ is also a diagonal matrix containing the approximation result from previous iteration

$$\mathbf{Y}_{k-1}(s) = \begin{bmatrix} Y_{k-1}(s_1) & 0 & \dots & 0 \\ 0 & Y_{k-1}(s_2) & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & Y_{k-1}(s_N) \end{bmatrix}. \quad (2.28)$$

(2.25) is a nonlinear equation, since A and b are also expressions of the unknowns. To solve (2.25) we start with an initial guess of $\mathbf{x}^T = [\mathbf{a}^T; \mathbf{b}^T]$ to evaluate $\mathbf{D}_{k-1}$ and $\mathbf{Y}_{k-1}$.

Then $[\Delta\mathbf{a}; \Delta\mathbf{b}]$ is calculated from (2.25). This difference is added to $\mathbf{x}^T$ to obtain a new $\mathbf{x}^T$ for the next iteration. This process is repeated until the difference $[\Delta\mathbf{a}; \Delta\mathbf{b}]$ no longer exceeds our tolerance.

Be aware of that (2.26) is ill-conditioned when frequencies are high. This difficulty is overcome by applying a robust basis transformation to (2.26) [43]. It is evident that (2.26) can be expressed by

$$\mathbf{U}_{0:m} = \begin{bmatrix} \mathbf{1} & \mathbf{S1} & \mathbf{S}^2\mathbf{1} & \dots & \mathbf{S}^m\mathbf{1} \end{bmatrix}, \tag{2.29}$$

where

$$\mathbf{S} = \begin{bmatrix} s_1 & \dots & 0 & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_N \end{bmatrix}, \tag{2.30}$$

and $\mathbf{1}$ is an N-column vector with 1 for all entries. An orthonormal basis for $\mathbf{U}_{0:m}$ can be achieved by Arnoldi process [44, 45]

$$\mathbf{S}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{v}_{m+1}\mathbf{h}_{m+1,m}, \tag{2.31}$$

where $\mathbf{V}$ is constructed by orthonormal columns and $\mathbf{H}$ is an upper Hessenberg matrix. $\mathbf{V}_{0:m} = [\mathbf{V}_m, \mathbf{v}_{m+1}]$ is the new basis spanning the same space as the columns of $\mathbf{U}_{0:m}$. Then (2.25) can be rewritten by applying the orthonormal basis

$$\begin{bmatrix} Re(\frac{1}{\mathbf{D}_{k-1}}\mathbf{V}_{0:m}) & \vdots & Re(-\frac{\mathbf{Y}_{k-1}}{\mathbf{D}_{k-1}}\mathbf{V}_{1:n}) \\ Im(\frac{1}{\mathbf{D}_{k-1}}\mathbf{V}_{0:m}) & \vdots & Im(-\frac{\mathbf{Y}_{k-1}}{\mathbf{D}_{k-1}}\mathbf{V}_{1:n}) \end{bmatrix} \begin{bmatrix} \Delta\mathbf{b}' \\ \Delta\mathbf{a}' \end{bmatrix} = \begin{bmatrix} Re\,(Y_{k-1}(s_1) - H(s_1)) \\ \vdots \\ Re\,(Y_{k-1}(s_N) - H(s_N)) \\ Im\,(Y_{k-1}(s_1) - H(s_1)) \\ \vdots \\ Im\,(Y_{k-1}(s_N) - H(s_N)) \end{bmatrix}. \tag{2.32}$$

By applying this basis transform, the high order associated with $\mathbf{U}_{0:m}$ is reduced, and the ill-condition problem is avoided.

2.3.4 A Brief Introduction to Vector Fitting

As discussed previously, the least-square based approach suffers ill-condition problem when solving equations like (2.12). The matrix on the left hand-side contains entries whose values are exponentially different from each other. This may cause a computational error when the algorithm is implemented in a computer program. To avoid this difficulty, an algorithm called Vector Fitting [7,22,41] was developed to solve the poles and residues of the system separately. The algorithm starts with an initial guess of the pole set and performs an iterative process to approximate the location of the poles. The residues are solved afterwards. By introducing the iterative process, the entries of the matrix corresponding to that of (2.12) are kept in their rational function format and the terms exponential of frequencies are effectively removed. Thus, ill-condition is effectively overcome. A deep review of its algorithm is given in Chapter 3.

2.4 Conclusion

This chapter reviewed some of the well developed techniques used for incorporating the frequency-domain parameters in time-domain simulators. The approaches based on convolution and macromodel have been discussed. The major difference between these two kinds of methods is that the macromodel-based methods generate analytical macromodels for the interconnects, while the convolution-based methods directly process the measured data to get transient responses. Their drawbacks are also discussed. The convolution-based approaches are very CPU-expensive for high accuracy simulation due to the convolution process, while the macromodel-based approaches may suffer from ill-condition which comes from the least-square problem. A brief introduction to Vector Fitting was also given. It is also classified to the category of rational-function-based approaches. But it applies iterative procedure to compute the poles of the network without suffering ill-condition and is convenient for computer implementation.

Chapter 3

Vector-Fitting-Based Rational Function Approximation

According to the discussion in Chapter 2, the conventional rational-function-based approaches aim to approximate the frequency-domain tabulated data with a ration function, normally in the form of

$$f(s) = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_m s^m}{b_0 + b_1 s + b_2 s^2 + \cdots + b_n s^n}. \tag{3.1}$$

The job focuses on finding all the coefficients $[a_0, a_1, \ldots, a_m]$ and $[b_0, b_1, \ldots, b_n]$ in the numerator and denominator of (3.1). Multiplying the left hand-side and right hand-side by the denominator $b_0 + b_1 s + b_2 s^2 + \cdots + b_n s^n$ and writing the equation at different frequency points, we can obtain an equation in the least square problem form:

$$\mathbf{AX} = \mathbf{b}. \tag{3.2}$$

(3.2) has limitation for high-order approximation, because the entries of $\mathbf{A}$ span a large orders of frequencies. Depending on the desired order of approximation, the elements may have very high power of s, causing big numerical differences between the maximum and the minimum entries. Under this circumstance, the linear equations will suffer from ill-condition problems. Thus, the traditional approaches have problem with capturing a large number of poles. On the other hand, due to the complexity of modern interconnects the number of poles of the network is usually big. To accurately capture the response around

peaks, it is desired to use a high-order approximation. Conventional pole fitting methods do not work well in this sense. [22,41] describes an accurate and robust fitting technique named *VectorFitting*, which can match complete responses without suffering ill-conditions. This chapter conducts a review of the algorithm , which was used as part of the macromodel technique presented in this thesis.

3.1 Pole Calculation by Vector Fitting

We start with a pole-residue approximation of the target function satisfying the measured data [22,41]

$$f(s) = \sum_{i=1}^{N} \frac{k_i}{s - p_i} + sh + c, \tag{3.3}$$

where N is the number of poles (residues), p_i and k_i can be either real or in the form of complex conjugate pairs. While h and c are real numbers referring to the term proportional to frequency and the coupling constant respectively. p_i, k_i, h and c are the unknowns. The task is to calculate these unknowns such that the macromodel given in (3.3) approximates the real response of the network at all the frequency points of the tabulated data.

To achieve this, we introduce a scaling function with a set of initial poles $\bar{p}_i$

$$\sigma(s) = \sum_{i=1}^{N} \frac{\tilde{k}_i}{s - \bar{p}_i} + 1. \tag{3.4}$$

Multiply the pole-residue function (3.3) by the scaling equation of (3.4) to get a scaled function $\sigma(s)f(s)$. Then this scaled function is approximated by another function $(\sigma f)(s)$, which has the same pole set as that of (3.4):

$$(\sigma f)(s) = \sum_{i=1}^{N} \frac{\hat{k}_i}{s - \bar{p}_i} + s\hat{h} + \hat{c}. \tag{3.5}$$

Considering that $(\sigma f)(s)$ is an approximation to $f(s)\sigma(s)$, we can write

$$\sigma(s)f(s) \approx (\sigma f)(s). \tag{3.6}$$

To see the number of unknowns of (3.6) the equation can be expanded as

$$\sum_{i=1}^{N} \frac{\hat{k}_i}{s-\bar{p}_i} + s\hat{h} + \hat{c} \approx (\sum_{i=1}^{N} \frac{\tilde{k}_i}{s-\bar{p}_i} + 1) f(s), \tag{3.7}$$

where $\bar{p}_i, i = 1, 2, \ldots, N$ are the initial guesses to the poles. From (3.7) the number of unknowns is $2N+2$: N $\hat{k}_i$'s, N $\tilde{k}_i$'s, one $\hat{h}$ and one $\hat{c}$. By writing (3.7) at K (normally $K > 2N+2$) frequency points at which the measured data are available and including all unknowns in one column vector, one obtains a set of linear equations in matrix form

$$\mathbf{AX} = \mathbf{b}. \tag{3.8}$$

This is an overdetermined problem, since the rank is higher than the number of unknowns. $\mathbf{X}$ is the vector of unknowns and $\mathbf{b}$ contains the frequency data. To decide the form of $\mathbf{A}$, it is necessary to separate the problem into two cases: complex poles and real poles.

For the real poles, (3.8) can be written as

$$\underbrace{\begin{bmatrix} \frac{1}{s_1-\bar{p}_1} & \cdots & \frac{1}{s_1-\bar{p}_N} & 1 & s_1 & \frac{-f(s_1)}{s_1-\bar{p}_1} & \cdots & \frac{-f(s_1)}{s_1-\bar{p}_N} \\ \frac{1}{s_2-\bar{p}_1} & \cdots & \frac{1}{s_2-\bar{p}_N} & 1 & s_2 & \frac{-f(s_2)}{s_2-\bar{p}_1} & \cdots & \frac{-f(s_2)}{s_2-\bar{p}_N} \\ \frac{1}{s_3-\bar{p}_1} & \cdots & \frac{1}{s_3-\bar{p}_N} & 1 & s_3 & \frac{-f(s_3)}{s_3-\bar{p}_1} & \cdots & \frac{-f(s_3)}{s_3-\bar{p}_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{s_K-\bar{p}_1} & \cdots & \frac{1}{s_K-\bar{p}_N} & 1 & s_K & \frac{-f(s_K)}{s_K-\bar{p}_1} & \cdots & \frac{-f(s_K)}{s_K-\bar{p}_N} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \hat{k}_1 \\ \vdots \\ \hat{k}_N \\ \hat{h} \\ \hat{c} \\ \tilde{k}_1 \\ \vdots \\ \tilde{k}_N \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} f(s_1) \\ f(s_2) \\ \vdots \\ f(s_K) \end{bmatrix}}_{b}. \tag{3.9}$$

Since real poles are always accompanied by real residues, the unknown vector X is real.

Similarly, the complex poles for each entry of the driving point admittance matrix are always accompanied by their complex conjugates and so are the residues. Thus, (3.8) can be expressed as

$$\sum_{i=1}^{N}(\frac{\hat{k}_i}{s-\bar{p}_i}+\frac{\hat{k}_i^*}{s-\bar{p}_i^*})+s\hat{h}+\hat{c}-f(s)\sum_{i=1}^{N}(\frac{\tilde{k}_i}{s-\bar{p}_i}+\frac{\tilde{k}_i^*}{s-\bar{p}_i^*})=f(s), \tag{3.10}$$

where "*" denotes the complex conjugate operator. Then the real and imaginary parts of the residues are separated

$$\begin{aligned}&\sum_{i=1}^{N}(Re(\hat{k}_i)(\frac{1}{s-\bar{p}_i}+\frac{1}{s-\bar{p}_i^*})+Im(\hat{k}_i)(\frac{j}{s-\bar{p}_i}+\frac{-j}{s-\bar{p}_i^*}))+s\hat{h}+\hat{c}\\&-f(s)\sum_{i=1}^{N}(Re(\tilde{k}_i)(\frac{1}{s-\bar{p}_i}+\frac{1}{s-\bar{p}_i^*})+Im(\tilde{k}_i)(\frac{j}{s-\bar{p}_i}+\frac{-j}{s-\bar{p}_i^*}))=f(s).\end{aligned} \tag{3.11}$$

Write (3.11) at different frequencies, the linear equations in matrix form (3.8) can be obtained. $\mathbf{A}=[\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_i,\ldots,\mathbf{A}_N,\mathbf{1},\mathbf{S}]$, where

$$\mathbf{A}_i=\begin{bmatrix}(\frac{1}{s_1-\bar{p}_i}+\frac{1}{s_1-\bar{p}_i^*}) & (\frac{j}{s_1-\bar{p}_i}-\frac{j}{s_1-\bar{p}_i^*}) & -f(s_1)(\frac{1}{s_1-\bar{p}_i}+\frac{1}{s_1-\bar{p}_i^*}) & -f(s_1)(\frac{j}{s_1-\bar{p}_i}-\frac{j}{s_1-\bar{p}_i^*})\\(\frac{1}{s_2-\bar{p}_i}+\frac{1}{s_2-\bar{p}_i^*}) & (\frac{j}{s_2-\bar{p}_i}-\frac{j}{s_2-\bar{p}_i^*}) & -f(s_2)(\frac{1}{s_2-\bar{p}_i}+\frac{1}{s_2-\bar{p}_i^*}) & -f(s_2)(\frac{j}{s_2-\bar{p}_i}-\frac{j}{s_2-\bar{p}_i^*})\\\vdots & \vdots & \vdots & \vdots\\(\frac{1}{s_K-\bar{p}_i}+\frac{1}{s_K-\bar{p}_i^*}) & (\frac{j}{s_K-\bar{p}_i}-\frac{j}{s_K-\bar{p}_i^*}) & -f(s_K)(\frac{1}{s_K-\bar{p}_i}+\frac{1}{s_K-\bar{p}_i^*}) & -f(s_K)(\frac{j}{s_K-\bar{p}_i}-\frac{j}{s_K-\bar{p}_i^*})\end{bmatrix}, \tag{3.12}$$

$$\mathbf{S}=[s_1,s_2,s_3,\ldots,s_K]^T, \tag{3.13}$$

and

$$\mathbf{X}=\left[Re(\hat{k}_1),Im(\hat{k}_1),Re(\tilde{k}_1),Im(\tilde{k}_1),\ldots,Re(\hat{k}_N),Im(\hat{k}_N),Re(\tilde{k}_N),Im(\tilde{k}_N),c,h\right]^T. \tag{3.14}$$

For both real and complex cases, the equation $\mathbf{AX}=\mathbf{b}$ is separated into real and imaginary parts

$$\underbrace{\begin{bmatrix}Re(\mathbf{A})\\Im(\mathbf{A})\end{bmatrix}}_{\hat{A}}\mathbf{X}=\underbrace{\begin{bmatrix}Re(\mathbf{b})\\Im(\mathbf{b})\end{bmatrix}}_{\hat{b}}. \tag{3.15}$$

(3.15) is solved by least-square technique. Left-multiplying both side of (3.15) by $\hat{\mathbf{A}}^T$

$$\hat{\mathbf{A}}^T\hat{\mathbf{A}}\mathbf{X} = \hat{\mathbf{A}}^T\hat{\mathbf{b}}, \tag{3.16}$$

and solving for (3.16), we get

$$\mathbf{X} = (\hat{\mathbf{A}}^T\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^T\hat{\mathbf{b}}. \tag{3.17}$$

So far the residues, coefficient of the frequency-proportional term, the coupling constant of the scaled function (3.5) and the residues of the scaling function (3.4) have been calculated. Note that (3.4) and (3.5) share the same set of initial poles. From this point on, (3.4) and (3.5) are written in the pole-zero form

$$\sigma(s) = \frac{\sum_{i=1}^{N}(s - \tilde{z}_i)}{\sum_{i=1}^{N}(s - \bar{p}_i)} \tag{3.18}$$

and

$$(\sigma f)(s) = h\frac{\sum_{i=1}^{N+1}(s - z_i)}{\sum_{i=1}^{N}(s - \bar{p}_i)}, \tag{3.19}$$

respectively. An expression for $f(s)$ is obtained by dividing both sides of (3.6) by (3.18)

$$f(s) = \frac{(\sigma f)(s)}{\sigma(s)} = h\frac{\sum_{i=1}^{N+1}(s - z_i)}{\sum_{i=1}^{N}(s - \tilde{z}_i)}. \tag{3.20}$$

From equation (3.20), it is clear that the zeros of the scaling function $\sigma(s)$ are identical to the poles of $f(s)$. Computing the poles turns out to be computing the zeros of the scaling function. Thus, the zeros of $\sigma(s)$ need to be evaluated from its residues. [22,41] tell

that the zeros of $\sigma(s)$ are the eigenvalues of the matrix

$$H = \eta - \phi, \tag{3.21}$$

where

$$\eta = \begin{bmatrix} \bar{p}_1 & 0 & \dots & 0 \\ 0 & \bar{p}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{p}_N \end{bmatrix}_{N\times N}, \tag{3.22}$$

$$\phi = \begin{bmatrix} \tilde{k}_1 & \tilde{k}_2 & \dots & \tilde{k}_N \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{k}_1 & \tilde{k}_2 & \dots & \tilde{k}_N \end{bmatrix}_{N\times N}, \tag{3.23}$$

for the case of real poles. In case of complex poles

$$\eta = \begin{bmatrix} \eta_1 & & & & \\ & \ddots & & & \\ & & \eta_i & & \\ & & & \ddots & \\ & & & & \eta_N \end{bmatrix}_{2N\times 2N}, \tag{3.24}$$

where

$$\eta_i = \begin{bmatrix} Re(\bar{p}_i) & Im(\bar{p}_i) \\ -Im(\bar{p}_i) & Re(\bar{p}_i) \end{bmatrix}_{2\times 2}, \tag{3.25}$$

and

$$\phi = \begin{bmatrix} 2Re(\tilde{k}_1) & 2Im(\tilde{k}_1) & \dots & 2Re(\tilde{k}_N) & 2Im(\tilde{k}_N) \\ 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2Re(\tilde{k}_1) & 2Im(\tilde{k}_1) & \dots & 2Re(\tilde{k}_N) & 2Im(\tilde{k}_N) \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{2N\times 2N}. \tag{3.26}$$

By computing the eigenvalues of the matrix (3.21), the zeros of the scaling function $\sigma(s)$ are found, which are also the poles of $f(s)$. It is worth noting that unstable poles may be obtained from this process. In this case, we simply reverse the signs of their real parts to

shift them to the left half-plane. The resulting set of poles is a closer guess to the poles of the network and substituted for (3.7) to start another iteration of the approximation. The above steps are repeated until the difference between the poles obtained from two successive iterations do not exceed the applied tolerance and the resulting pole set is an accurate approximation for the poles of the original system.

In order to make the algorithm work efficiently, the strategy for choosing initial poles becomes vital. A good initial guess enables the program to capture the response of the system in only a few iterations. There will be numerical problems due to improperly choosing of the starting poles:

a. The linear equation (3.9) will become ill-conditioned, if a set of real initial poles is chosen to treat a function with distinct resonant peaks.

b. An arbitrary choose of starting poles may lead to a large variations in the scaling function $\sigma(s)$ and $\sigma(s)f(s)$.

The solution to problem (**a**) is to choose complex poles when dealing with functions with distinct resonant peaks. The strategy to address problem (**b**) is to sensibly select the starting poles and run the iteration after that. A general rule of choosing starting poles is suggested in [41]:

1. For the function with distinct resonant peaks, a set of complex starting poles should be chosen to capture the change around peaks. For each complex pole, a sufficiently small real part should be chosen. Normally, the real part is chosen to be one hundredth of the imaginary part (i.e. for $p = \alpha + j\beta$, choose $\alpha = \beta/100$).

2. For smooth functions, it is recommended to choose real poles which are linearly spaced as a function of frequency. In the smooth function case, starting with real poles will not affect the accuracy of the fitting result.

3. Generally, the starting poles should be chosen to be located linearly across the frequency band of interest. The benefit from this strategy is that the frequency range of

interest is evenly covered. As a result, the starting poles will not be shifted far from their original locations. It is worth noting that if the initial poles are densely chosen within a small frequency interval, there will be a large variation between the LHS and RHS of equation (3.6) at the frequencies beyond this interval. A large CPU cost is needed to shift these poles over a large frequency distance to reach convergence.

Notably, the process discussed so far in this section is for one-port network. However, most networks in real world application have multiple ports. Consider an M-port network described by Y-parameters

$$\mathbf{Y}(s) = \begin{bmatrix} Y_{11}(s) & Y_{12}(s) & \dots & Y_{1M}(s) \\ Y_{21}(s) & Y_{22}(s) & \dots & Y_{2M}(s) \\ \vdots & \vdots & \ddots & \vdots \\ Y_{M1}(s) & Y_{M2}(s) & \dots & Y_{MM}(s) \end{bmatrix}. \tag{3.27}$$

Each entry of the above matrix is approximated by the pole-residue form:

$$f(s) = \sum_{n=1}^{N} \frac{k_n^{i,j}}{s - p_n^{i,j}} + sh^{i,j} + c^{i,j}; \text{ for } i,j=0,1,\dots,M. \tag{3.28}$$

In conventional methods, the rational function approximation is applied to each entry of the driving point admittance matrix, i.e.(3.28), to get the poles. A union of pole set is obtained by collecting the poles for each entry together. However, this approach will lead to a considerable redundancy of poles, which will cause inefficiencies in transient simulations [7,22]. In order to remove the redundant poles from the pole set, we take advantage of the following two propositions [7,46]:

1. Generally speaking, the pole set corresponding to each individual entry of the driving point admittance matrix (3.27) is a subset of the union of poles for all driving point admittances.

2. In a system with a large number of dominant poles, pole sets corresponding to different driving point admittances contain mostly identical poles and with only a very small number of poles differing among these pole sets.

With the above propositions, instead of fitting the individual pole set for each entry of the admittance matrix, we only fit the poles of the diagonal entries. This is because the poles of the diagonal elements form a subset of all poles of the subnetwork (proposition 1), and there's very minor difference from the poles of the off-diagonal entries (proposition 2). We apply Vector Fitting algorithm to the vector of the diagonal entries of the driving point admittance matrix

$$\mathbf{Y}_{diag}(s) = \begin{bmatrix} Y_{11}(s) \\ \vdots \\ Y_{ii}(s) \\ \vdots \\ Y_{MM}(s) \end{bmatrix}, \tag{3.29}$$

where

$$Y_{ii}(s) = \sum_{n=1}^{N} \frac{k_n^{i,i}}{s - p_n^{i,i}} + sh^{i,i} + c^{i,i}; \text{ for } i=0,1,\ldots,M. \tag{3.30}$$

A scaling function with a set of initial poles is introduced as

$$\sigma(s) = \sum_{n=1}^{N} \frac{\tilde{k}_n}{s - \bar{p}_n} + 1. \tag{3.31}$$

Similar to the case of single-port network, multiplying each entry of (3.29) by $\sigma(s)$ and approximating the resulting scaled function by $(\sigma Y_{diag})(s)$, we obtain

$$\begin{bmatrix} \sigma(s)Y_{11}(s) \\ \sigma(s)Y_{22}(s) \\ \vdots \\ \sigma(s)Y_{MM}(s) \end{bmatrix} \approx \begin{bmatrix} (\sigma Y_{11})(s) \\ (\sigma Y_{22})(s) \\ \vdots \\ (\sigma Y_{MM})(s) \end{bmatrix}. \tag{3.32}$$

Using (3.31), (3.32) can be further expanded as

$$\begin{bmatrix} (\sum_{n=1}^{N} \frac{\tilde{k}_n^{11}}{s-\bar{p}_n} + 1)Y_{11}(s) \\ (\sum_{n=1}^{N} \frac{\tilde{k}_n^{22}}{s-\bar{p}_n} + 1)Y_{22}(s) \\ \vdots \\ (\sum_{n=1}^{N} \frac{\tilde{k}_n^{MM}}{s-\bar{p}_n} + 1)Y_{MM}(s) \end{bmatrix} \approx \begin{bmatrix} \sum_{n=1}^{N} \frac{\hat{k}_n^{11}}{s-\bar{p}_n} + s\hat{h}^{11} + \hat{c}^{11} \\ \sum_{n=1}^{N} \frac{\hat{k}_n^{22}}{s-\bar{p}_n} + s\hat{h}^{22} + \hat{c}^{22} \\ \vdots \\ \sum_{n=1}^{N} \frac{\hat{k}_n^{MM}}{s-\bar{p}_n} + s\hat{h}^{MM} + \hat{c}^{MM} \end{bmatrix}. \tag{3.33}$$

Note that the poles of the scaled function $\sigma(s)Y_{diag}(s)$ are same than the ones of the approximation function $(\sigma Y_{diag})(s)$ and there are $2M + (M+1)N$ unknowns. Writing each row of (3.33) at several frequency points we obtain an overdetermined equation

$$\mathbf{AX} = \mathbf{b}, \tag{3.34}$$

where

$$\mathbf{A} = \begin{bmatrix} \frac{1}{s_1-\bar{p}_1} & \dots & \frac{1}{s_1-\bar{p}_N} & 1 & s_1 & 0 & \dots & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \frac{-Y_{11}(s_1)}{s_1-\bar{p}_1} & \dots & \frac{-Y_{11}(s_1)}{s_1-\bar{p}_N} \\ \vdots & & \vdots & \vdots & & \vdots & & & & \dots & \dots & \dots & \dots & \dots & \dots & \vdots & & \vdots \\ \frac{1}{s_K-\bar{p}_1} & \dots & \frac{1}{s_K-\bar{p}_N} & 1 & s_K & 0 & \dots & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \frac{-Y_{11}(s_K)}{s_K-\bar{p}_1} & \dots & \frac{-Y_{11}(s_K)}{s_K-\bar{p}_N} \\ 0 & \dots & 0 & 0 & 0 & \frac{1}{s_1-\bar{p}_1} & \dots & \frac{1}{s_1-\bar{p}_N} & 1 & s_1 & \dots & \dots & \dots & \dots & \dots & \frac{-Y_{22}(s_1)}{s_1-\bar{p}_1} & \dots & \frac{-Y_{22}(s_1)}{s_1-\bar{p}_N} \\ \vdots & & \vdots & \vdots & & \vdots & & & & \dots & \dots & \dots & \dots & \dots & \dots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \frac{1}{s_K-\bar{p}_1} & \dots & \frac{1}{s_K-\bar{p}_N} & 1 & s_K & \dots & \dots & \dots & \dots & \dots & \frac{-Y_{22}(s_K)}{s_K-\bar{p}_1} & \dots & \frac{-Y_{22}(s_K)}{s_K-\bar{p}_N} \\ \vdots & & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \frac{1}{s_1-\bar{p}_1} & \dots & \frac{1}{s_1-\bar{p}_N} & 1 & s_1 & \frac{-Y_{MM}(s_1)}{s_1-\bar{p}_1} & \dots & \frac{-Y_{MM}(s_1)}{s_1-\bar{p}_N} \\ \vdots & & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \frac{1}{s_K-\bar{p}_1} & \dots & \frac{1}{s_K-\bar{p}_N} & 1 & s_K & \frac{-Y_{MM}(s_K)}{s_K-\bar{p}_1} & \dots & \frac{-Y_{MM}(s_K)}{s_K-\bar{p}_N} \end{bmatrix}, \tag{3.35}$$

$$\mathbf{X} = \begin{bmatrix} \hat{k}_1^{11} & \dots & \hat{k}_N^{11} & \hat{c}^{11} & \hat{h}^{11} & \dots & \hat{k}_1^{MM} & \dots & \hat{k}_N^{MM} & \hat{c}^{MM} & \hat{h}^{MM} & \tilde{k}_1 & \dots & \tilde{k}_N \end{bmatrix}^T, \tag{3.36}$$

$$\mathbf{b} = \left[\begin{array}{cccccccccc} Y_{11}(s_1) & \dots & Y_{11}(s_K) & Y_{22}(s_1) & \dots & Y_{22}(s_K) & \dots & Y_{MM}(s_1) & \dots & Y_{MM}(s_K) \end{array} \right]^T . \tag{3.37}$$

(3.34) is solved by least-square technique. The last N entries $(\tilde{k}_1 \dots \tilde{k}_N)$ of the solution vector $\mathbf{b}$ are the residues of the scaling function and are used to calculate the new set of poles by following (3.21-3.26). The problem is solved self-consistently until convergence is reached.

3.2 Residue Calculation

Once the poles of the network are identified, we proceed to calculate the residues. With the poles, the admittance matrix of $\mathbf{Y}(s)$ can be written as

$$\mathbf{Y}(s) = \left[\begin{array}{ccccc} c^{1,1} + \sum_{n=1}^{N} \frac{k_n^{1,1}}{s - p_n} & c^{1,2} + \sum_{n=1}^{N} \frac{k_n^{1,2}}{s - p_n} & \dots & c^{1,M} + \sum_{n=1}^{N} \frac{k_n^{1,M}}{s - p_n} \\ \vdots & \vdots & \vdots & \vdots \\ c^{M,1} + \sum_{n=1}^{N} \frac{k_n^{M,1}}{s - p_n} & c^{M,2} + \sum_{n=1}^{N} \frac{k_n^{M,2}}{s - p_n} & \dots & c^{M,M} + \sum_{n=1}^{N} \frac{k_n^{M,M}}{s - p_n} \end{array} \right] . \tag{3.38}$$

The residues $k_n^{i,j}$ and coupling constant $c^{i,j}$ for each entry of (3.38) are solved individually by writing the rational function for each entry at several frequencies, $s_1, s_2, \dots, s_K$. Generally, K is larger than the number of unknowns, $N+1$. Thus, it is an overdetermined equation of the form

$$\mathbf{A}\mathbf{X}^{i,j} = \mathbf{b}^{i,j}. \tag{3.39}$$

For the case of real poles, the matrix $\mathbf{A}$, $\mathbf{b}^{i,j}$ and the unknown vector $\mathbf{X}^{i,j}$can be written as

$$\mathbf{A} = \left[\begin{array}{cccc} 1 & \frac{1}{s_1 - p_1} & \frac{1}{s_1 - p_2} & \dots & \frac{1}{s_1 - p_N} \\ 1 & \frac{1}{s_2 - p_1} & \frac{1}{s_2 - p_2} & \dots & \frac{1}{s_2 - p_N} \\ \vdots & \vdots & \dots & \vdots & \\ 1 & \frac{1}{s_K - p_1} & \frac{1}{s_K - p_2} & \dots & \frac{1}{s_K - p_N} \end{array} \right], \tag{3.40}$$

$$\mathbf{b}^{i,j} = \begin{bmatrix} Y^{i,j}(s_1) \\ Y^{i,j}(s_2) \\ \vdots \\ Y^{i,j}(s_K) \end{bmatrix}, \tag{3.41}$$

and

$$\mathbf{X}^{i,j} = \begin{bmatrix} c^{i,j} & k_1^{i,j} & k_2^{i,j} & \dots & k_N^{i,j} \end{bmatrix}^T. \tag{3.42}$$

For the case of complex poles, the residues are also in the form of complex conjugate pair. Thus, (3.28) can be written as

$$\begin{aligned} Y_{i,j}(s) = c^{i,j} + \frac{k_{1R}^{i,j} + jk_{1I}^{i,j}}{s - p_1} + \frac{k_{1R}^{i,j} - jk_{1I}^{i,j}}{s - p_1^*} + \frac{k_{2R}^{i,j} + jk_{2I}^{i,j}}{s - p_2} + \frac{k_{2R}^{i,j} - jk_{2I}^{i,j}}{s - p_2^*} + \\ \dots + \frac{k_{NR}^{i,j} + jk_{NI}^{i,j}}{s - p_N} + \frac{k_{NR}^{i,j} - jk_{NI}^{i,j}}{s - p_N^*}, \end{aligned} \tag{3.43}$$

where the subscript "R" refers to the real part, and "I" refers to the imaginary part. By separating the real and imaginary parts, the solution vector only contains real values. Expanding the above equation at several frequencies, we obtain $\mathbf{A}$, $\mathbf{b}^{i,j}$ and $\mathbf{X}^{i,j}$ as follows

$$\mathbf{A} = \begin{bmatrix} 1 & \frac{1}{s_1 - p_1} + \frac{1}{s_1 - p_1^*} & \frac{j}{s_1 - p_1} - \frac{j}{s_1 - p_1^*} & \dots & \frac{1}{s_1 - p_N} + \frac{1}{s_1 - p_N^*} & \frac{j}{s_1 - p_N} - \frac{j}{s_1 - p_N^*} \\ 1 & \frac{1}{s_2 - p_1} + \frac{1}{s_2 - p_1^*} & \frac{j}{s_2 - p_1} - \frac{j}{s_2 - p_1^*} & \dots & \frac{1}{s_2 - p_N} + \frac{1}{s_2 - p_N^*} & \frac{j}{s_2 - p_N} - \frac{j}{s_2 - p_N^*} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & \frac{1}{s_K - p_1} + \frac{1}{s_K - p_1^*} & \frac{j}{s_K - p_1} - \frac{j}{s_K - p_1^*} & \dots & \frac{1}{s_K - p_N} + \frac{1}{s_K - p_N^*} & \frac{j}{s_K - p_N} - \frac{j}{s_K - p_N^*} \end{bmatrix}, \tag{3.44}$$

$$\mathbf{b}^{i,j} = \begin{bmatrix} Y^{i,j}(s_1) \\ Y^{i,j}(s_2) \\ \vdots \\ Y^{i,j}(s_K) \end{bmatrix}, \tag{3.45}$$

$$\mathbf{X}^{i,j} = \begin{bmatrix} c^{i,j} & k_{1R}^{i,j} & k_{1I}^{i,j} & k_{2R}^{i,j} & k_{2I}^{i,j} & \dots & k_{NR}^{i,j} & k_{NI}^{i,j} \end{bmatrix}^T. \tag{3.46}$$

Equating the real and imaginary parts of LHS and RHS of (3.39) respectively, we have

$$\begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix} \mathbf{X}^{i,j} = \begin{bmatrix} Re(\mathbf{b}^{i,j}) \\ Im(\mathbf{b}^{i,j}) \end{bmatrix}. \tag{3.47}$$

(3.47) is solved by least square methods. Left-multiplying both sides of (3.47) by $[Re(\mathbf{A}), Im(\mathbf{A})]$, we have

$$\begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix}^T \begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix} \mathbf{X}^{i,j} = \begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix}^T \begin{bmatrix} Re(\mathbf{b}^{i,j}) \\ Im(\mathbf{b}^{i,j}) \end{bmatrix}. \tag{3.48}$$

The solution to the above equation is

$$\mathbf{X}^{i,j} = \left(\begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix}^T \begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix} \right)^{-1} \begin{bmatrix} Re(\mathbf{A}) \\ Im(\mathbf{A}) \end{bmatrix}^T \begin{bmatrix} Re(\mathbf{b}^{i,j}) \\ Im(\mathbf{b}^{i,j}) \end{bmatrix}. \tag{3.49}$$

By repeating (3.39)-(3.49) for each entry of (3.27), the computation of residues is completed. All the rational functions share the same set of dominant poles but different residues. The algorithm of Vector Fitting is summarized in Fig. 3.1 in pseudocode.

3.3 Conclusion

Vector Fitting technique displays a big advantage over the traditional approximation algorithms. It efficiently overcomes the problem of ill-condition, which exists in the process of conventional rational function approximation algorithms. Conventional approximation methods end up with a set of linear equations $\mathbf{AX} = \mathbf{b}$, where the matrix $\mathbf{A}$ is normally Vandermonde-like. In other words, the entries of $\mathbf{A}$ are expressed in form of power of frequencies depending on the accuracy requirement of the approximation. This leads to the potential of ill-condition when we write the equation at higher frequencies or when a high-order approximation is desired. On the other hand, Vector Fitting technique is immune to this problem, as the entries of the matrix $\mathbf{A}$ are in the form of single power of frequencies. Due to this property, with appropriate initial poles provided, Vector Fitting can fit the poles at high frequencies in high-order approximation without suffering computational difficulties. This advantage enables the algorithm to accurately locate the poles throughout the bandwidth of interest.

Step 1. Choose an appropriate set of initial poles by following the guideline in Section 3.1.

Step 2. Construct Equation (3.34) using (3.35) and (3.37).

Step 3. Solve the resulting equation from Step 2 by least-square technique to obtain the residues of the scaling function $\sigma(s)$.

Step 4. Follow the guideline given by Equation (3.21) - (3.26) to evaluate the zeros of $\sigma(s)$ which are considered as the new poles. Inverse the sign of the real part of any unstable pole.

Step 5. If the poles obtained from Step 4 converge, go directly to Step 6. Otherwise repeat Step 2 to Step 5 for another iteration.

Step 6. Calculate the residues and coupling constant for each entry of (3.38) by following Equation (3.39) – (3.49).

Fig. 3.1 Pseudocode of Vector Fitting

Chapter 4

The Passivity of Macromodels

In previous chapters, we have reviewed the methods developed to extract accurate macromodels from measured/simulated data. In this chapter we will have discussion on another important property of electrical networks - *passivity*. This chapter begins with a brief introduction to passivity. Methods used for passivity check will be reviewed later. In the last part, an efficient method used for checking passivity and precisely locating the regions of passivity violation will be introduced.

4.1 What Is Passivity?

The techniques reviewed in previous chapters can generate fairly accurate macromodels despite the fact that they perform distinctly in numerical robustness and efficiency. However, as pointed out in Chapter 1, passivity is another concern associated with the macromodel besides accuracy. Passivity implies that a network can never generate more energy than it absorbs [22,47,48]. Unfortunately, these methods cannot guarantee the passivity of the macromodels. In other words, the passivity of the original network under measurement may not be inherited by its macromodel. Loss of passivity is a major defect of the macromodel, even though it can accurately represent the behavior of the original network in a frequency band of interest. Passivity is a vital issue because when a stable but nonpassive macromodel is terminated by some arbitrary passive macromodels, the transient simulation of the whole system may suffer from artificial oscillations [7]. The reason for this is that the poles are relocated during the connection of the macromodels. When stable but nonpassive macromodels are connected to even passivity-guaranteed macromodels, it is possible

that the poles of the new system become unstable. To illustrate this point, consider a second-order macromodel shown in Fig. 4.1. The macromodel is stable but nonpassive. It has a pair of stable poles. After it is connected to a passive model, unstable poles are generated causing oscillations in the transient simulation. Thus, it is important to ensure that each macromodel is passive before the system-level transient simulation. To reach this goal, an efficient passivity checking algorithm is desired. Some widely used methods will be introduced in the following sections of this chapter.

4.2 Passivity Check Techniques

In this section, we will review traditional methods developed for checking passivity. Generally, the methods can be classified into two categories: frequency-domain based method and time-domain based method.

4.2.1 Frequency-Domain Passivity Check

A passive multi-port network, expressed in (4.1)

$$\mathbf{Y}(s) = \begin{bmatrix} Y_{11}(s) & Y_{12}(s) & \dots & Y_{1M}(s) \\ Y_{21}(s) & Y_{22}(s) & \dots & Y_{2M}(s) \\ \vdots & \vdots & \ddots & \vdots \\ Y_{M1}(s) & Y_{M2}(s) & \dots & Y_{MM}(s) \end{bmatrix}. \tag{4.1}$$

must satisfy the frequency-domain passivity criteria, given by [38, 47–49]

a) $\mathbf{Y}(s^*) = \mathbf{Y}^*(s)$, where "*" denotes the complex conjugate operation.
b) $\mathbf{Y}(s)$ is a positive real (PR) matrix.

Condition **a)** can be translated into that the rational function $\mathbf{Y}(s)$ has only real coefficients in its numerator and denominator. This condition is automatically satisfied by driving point admittance matrices, since the poles and residues of driving point admittance matrices are either real numbers or complex conjugate pairs. For real s, $\mathbf{Y}(s)$ is real also. By ensuring Condition **b)**, the driving point function has to satisfy the following inequation

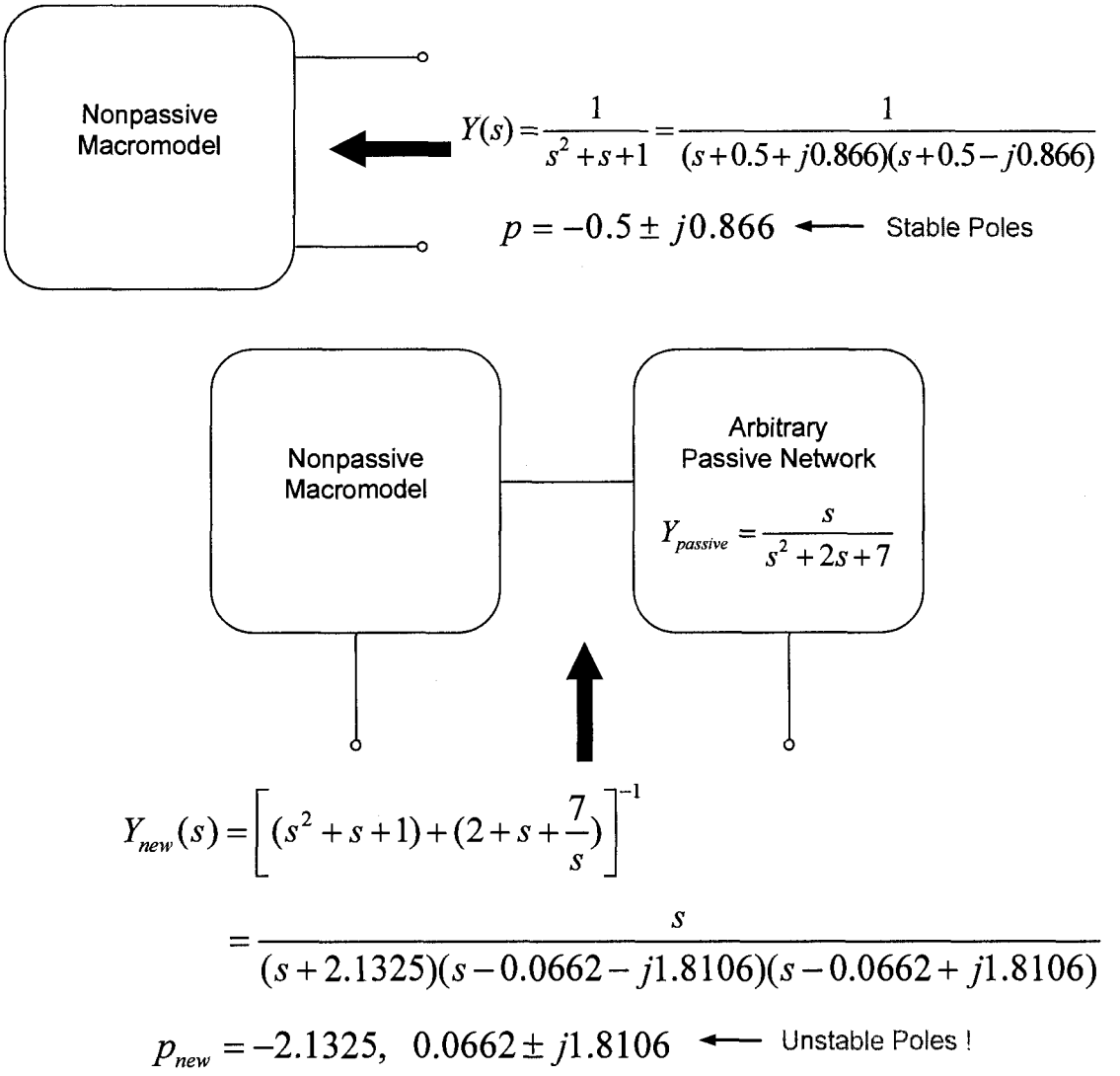

Fig. 4.1 Illustration of the significance of passivity: A stable but nonpassive macromodel connected to an arbitrary passive macromodel leads to an unstable system.

$$\mathbf{z}^* \left[\mathbf{Y}^T(s^*) + \mathbf{Y}(s)\right] \mathbf{z} \geq 0, \tag{4.2}$$

for any complex frequency with $Re(s) > 0$ and for any arbitrary vector $\mathbf{z}$. It is to be noted that $\mathbf{Y}(s)$ is a complex symmetric matrix. It can be written as the sum of a Hermitian matrix, which is the real part, and an anti-Hermitian matrix, which is the imaginary part. With this property, $\left[\mathbf{Y}^T(s^*) + \mathbf{Y}(s)\right]/2$ refers to the Hermitian part or the real part of $\mathbf{Y}(s)$. Thus, Condition **b)** is satisfied by ensuring

$$Re(\mathbf{Y}(s)) = \frac{\left[\mathbf{Y}^T(s^*) + \mathbf{Y}(s)\right]}{2} \geq 0; \text{ for all s with } Re(s) > 0. \tag{4.3}$$

From [7,48], for a matrix rational function without poles on the closed right half-plane, Condition **b)** implies

$$Re(\mathbf{Y}(j\omega)) = \frac{\left[\mathbf{Y}^T(j\omega^*) + \mathbf{Y}(j\omega)\right]}{2} \geq 0; \text{ for } \omega \in \Re \cup \infty. \tag{4.4}$$

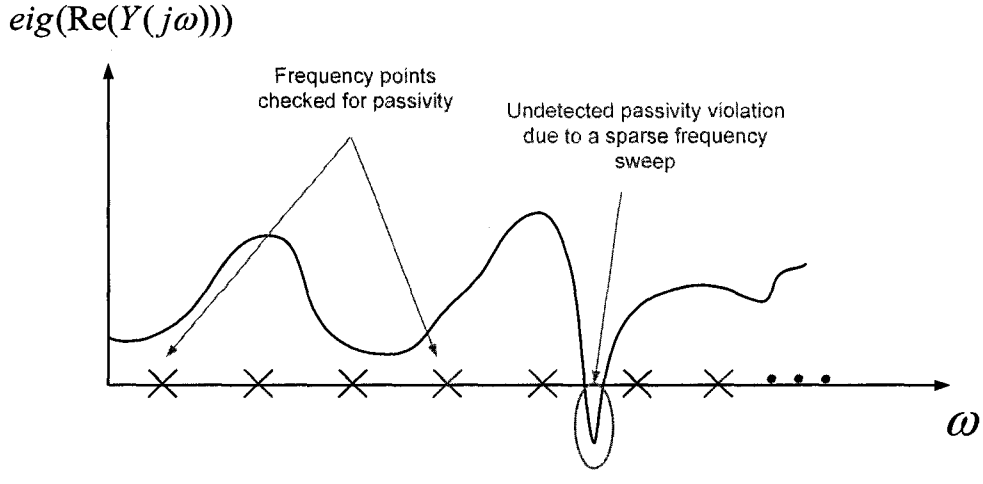

Fig. 4.2 An insufficient frequency sweep leads to misjudgment.

(4.4) tells us the real part of $\mathbf{Y}(j\omega)$ must be positive semi-definite for all $j\omega$. By the properties of positive semi-definite, $Re(\mathbf{Y}(j\omega))$ should contain only semi-positive eigenvalues ($\lambda_i \geq 0$, for $i = 1, 2, \ldots, M$). Therefore, the most straightforward way to check passivity is accomplished by carrying out a frequency sweep of the eigenvalues of the real part of the admittance matrix $\mathbf{Y}(s)$. If its eigenvalues at all frequency points are nonneg-

ative, the macromodel is passive. However, as pointed out in [38] this method suffers from several drawbacks. For example, it is hard to say up to what frequency the sweep may stop and how fine the sweep should be. Evaluation of the eigenvalues at very fine frequency grids is extremely CPU-expensive, especially for large networks with high number of ports. Even so, an exhaustive sweep can never be achieved, as it is impossible to evaluate every frequency point. On the other hand, a sparse sweep is much faster, but the passivity violation taking place between two consecutive frequency points may be missed. This kind of insufficient evaluation directly leads to a misjudgment. The mechanism of the failure is shown in Fig. 4.2.

4.2.2 Time Domain Passivity Check

A multi-port lumped linear network can be described by a set of state-space equations [22, 50]

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{4.5}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{4.6}$$

such that

$$\mathbf{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{4.7}$$

where $\dot{\mathbf{x}}(t)$ denotes the derivative of $\mathbf{x}$. For a M-input M-output network with n state variables, $\mathbf{x}$ is a vector of the size $\Re^{n\times 1}$, $\mathbf{u}$ is the input vector of the size $\Re^{M\times 1}$, $\mathbf{A}$ is the state matrix of the size $\Re^{n\times n}$, $\mathbf{B}$ is a $n \times M$ matrix relating the input vector $\mathbf{u}$ to the state variables $\mathbf{x}$, $\mathbf{C}$ is a $M \times n$ matrix linking the state variables to the output, and $\mathbf{D}$ is a $M \times M$ matrix directly coupling the input $\mathbf{u}$ to the output. Once we obtain the macromodel of the system, *i.e.* the poles, residues and coupling constants of the rational approximation are known, the state-space representations $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ can be easily derived.

Firstly, let's consider a macromodel of a two-port network with two real poles

$$\mathbf{Y}(s) = \begin{bmatrix} c^{1,1} + \sum_{i=1}^{2} \frac{k_i^{1,1}}{s - p_i} & c^{1,2} + \sum_{i=1}^{2} \frac{k_i^{1,2}}{s - p_i} \\ c^{2,1} + \sum_{i=1}^{2} \frac{k_i^{2,1}}{s - p_i} & c^{2,2} + \sum_{i=1}^{2} \frac{k_i^{2,2}}{s - p_i} \end{bmatrix}. \tag{4.8}$$

The corresponding state-space representations are

$$\mathbf{A} = \begin{bmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_1 & 0 & 0 \\ 0 & 0 & p_2 & 0 \\ 0 & 0 & 0 & p_2 \end{bmatrix} \tag{4.9}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}^T \tag{4.10}$$

$$\mathbf{C} = \begin{bmatrix} k_1^{1,1} & k_1^{1,2} & k_2^{1,1} & k_2^{1,2} \\ k_1^{2,1} & k_1^{2,2} & k_2^{2,1} & k_2^{2,2} \end{bmatrix} \tag{4.11}$$

$$\mathbf{D} = \begin{bmatrix} c^{1,1} & c^{1,2} \\ c^{2,1} & c^{2,2} \end{bmatrix}. \tag{4.12}$$

Secondly, in general, macromodels contain poles in the form of complex conjugate pairs. In this case, (4.5) and (4.6) are complex equations, which do not have physical meaning in the time domain. They therefore need to be rewritten in Jordan-form [22]

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_1^* \end{bmatrix}}_{\mathbf{A}'} \mathbf{x} + \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_1^* \end{bmatrix}}_{\mathbf{B}'} \mathbf{u}, \tag{4.13}$$

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_1^* \end{bmatrix}}_{\mathbf{C}'} \mathbf{x} + \mathbf{D} \tag{4.14}$$

where $\mathbf{A}_1$ contains the complex poles, and $\mathbf{A}_1^*$ contains their conjugates. Consider a two-port macromodel containing both complex and real poles, which is the most common case in real application. Without loss of generality, assume this macromodel has one pair of complex poles and one real pole: $p_{1,2} = \alpha \pm i\beta, p_3 = \gamma$ and the corresponding residues are $c_{1,2}^{i,j} = (r \pm iv)^{i,j}, c_3^{i,j} = k^{i,j}$. Then the resulting state-space matrices are

$$\mathbf{A}' = \begin{bmatrix} \alpha+i\beta & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha+i\beta & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha-i\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha-i\beta & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma \end{bmatrix} \tag{4.15}$$

$$\mathbf{B}' = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}^T \tag{4.16}$$

$$\mathbf{C}' = \begin{bmatrix} r^{1,1}+jv^{1,1} & r^{1,2}+jv^{1,2} & r^{1,1}-jv^{1,1} & r^{1,2}-jv^{1,2} & k^{1,1} & k^{1,2} \\ r^{2,1}+jv^{2,1} & r^{2,2}+jv^{2,2} & r^{2,1}-jv^{2,1} & r^{2,2}-jv^{2,2} & k^{2,1} & k^{2,2} \end{bmatrix} \tag{4.17}$$

$$\mathbf{D} = \begin{bmatrix} c^{1,1} & c^{1,2} \\ c^{2,1} & c^{2,2} \end{bmatrix}. \tag{4.18}$$

In order to make $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ real matrices, an equivalent transformation is applied to (4.13) and (4.14)

$$\mathbf{V}\dot{\mathbf{x}} = \underbrace{\left(\mathbf{V}\mathbf{A}'\mathbf{V}^{-1}\right)}_{\mathbf{A}}(\mathbf{V}\mathbf{x}) + \underbrace{\left(\mathbf{V}\mathbf{B}'\right)}_{\mathbf{B}}\mathbf{u}. \tag{4.19}$$

$$\mathbf{y} = \underbrace{\left(\mathbf{C}'\mathbf{V}^{-1}\right)}_{\mathbf{C}}(\mathbf{V}\mathbf{x}) + \mathbf{D} \tag{4.20}$$

where $\mathbf{V}$ is the transformation matrix of the same size as $\mathbf{A}'$. It has the form of

$$\mathbf{V} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & 0 \\ \mathbf{I}i & -\mathbf{I}i & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \tag{4.21}$$

where $\mathbf{I}$ is the identity matrix. The resulting matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in (4.19) and (4.20) are real matrices and the value of the output $\mathbf{y}$ is preserved.

The time-domain state-space representation is then checked for passivity. By [22, 51], the state-space system is passive, if there exists a matrix $\mathbf{X}$ that is symmetric and real positive definite to satisfy the continuous-time algebraic Riccati equation

$$\mathbf{A}^T\mathbf{X} + \mathbf{X}\mathbf{A} + (\mathbf{X}\mathbf{B} - \mathbf{C}^T)(\mathbf{D} + \mathbf{D}^T)^{-1}(\mathbf{X}\mathbf{B} - \mathbf{C}^T)^T = 0, \tag{4.22}$$

where all eigenvalues of $\mathbf{A}$ must have negative real parts. The $(\mathbf{A}, \mathbf{B})$ pair must be stabilizable, and $\mathbf{D} + \mathbf{D}^T > 0$. These properties can be automatically satisfied by the macromodel. Eigenvalues of $\mathbf{A}$ has only negative real parts, as only stable poles (with negative real part) are obtained from Vector Fitting. And $\mathbf{D} + \mathbf{D}^T > 0$ can be guaranteed by enforcing the diagonal entries of $\mathbf{D}$ to be greater than zero and off-diagonal entries to be equal to zero during the residue calculation process [7]. This time-domain passivity check algorithm is independent of frequency and very fast, but it still fails to give any information about the location of the passivity violation. It is simply an ideal method to check the passivity of a macromodel.

4.3 An Efficient Method of Passivity Verification

In the previous section, we have discussed frequency-domain and time-domain methods used to check the passivity of macromodels. Their common drawback is that they fail to tell the location of passivity violation. However, this information is required for the later compensation process. In order to achieve this function, a new passivity check method was developed in [38], which is efficient and informative. The description of the method starts with two theorems.

Theorem 1 [51]: The state-space system described by $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is passive if the Hamiltonian matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} - \mathbf{B}(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{C} & \mathbf{B}(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{B}^T \\ \mathbf{C}^T(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{C} & -\mathbf{A}^T + \mathbf{C}^T(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{B}^T \end{bmatrix} \tag{4.23}$$

has no pure imaginary eigenvalues.

This method is similar to the time-domain passivity check method, as it is also based

on the state-space representation. The advantage of Theorem 1 becomes clear when it is combined with Theorem 2.

Theorem 2 [52–55] The real part of the symmetric admittance matrix $\mathbf{F}(j\omega_0) = Real(\mathbf{Y}(j\omega_0))$ is singular if $j\omega_0$ is an eigenvalue of the corresponding Hamiltonian matrix $\mathbf{M}$ in Theorem 1, provided $\mathbf{D} + \mathbf{D}^T$ is a positive definite matrix.

Theorem 2 relates the imaginary eigenvalue of the Hamiltonian matrix $\mathbf{M}$ to the frequencies where the macromodel becomes nonpassive. It implies that any imaginary eigenvalue of the Hamiltonian corresponds to the frequencies at which the real part becomes singular. In other words, these frequencies points are exactly the zero-crossing points of the eigenvalues of the real part of $\mathbf{Y}(j\omega)$. The constraint of Theorem 2 (i.e., $\mathbf{D} + \mathbf{D}^T > 0$) implies the macromodel should be asymptotically passive at $\omega = \infty$. This can be easily ensured by enforcing the coupling constants corresponding to the diagonal entries of $\mathbf{Y}(s)$ to be positive and the ones corresponding to the off-diagonal entries to be zero during the Vector Fitting process [7].

However, Theorem 2 only tells us the frequency points at which the macromodel becomes singular. They are not necessarily the exact locations of the passivity violation regions. The exact locations are unknown, unless the slopes of the eigenvalues of $\mathbf{F}(j\omega_0)$ at each frequency point detected by Theorem 1 are determined. This calculation can be done as follows.

Assuming that λ is an eigenvalue of $\mathbf{F}(j\omega_0)$ and $\mathbf{u}$ is the corresponding right eigenvector, we have

$$(\mathbf{F}(j\omega) - \lambda\mathbf{I})\mathbf{u} = 0. \tag{4.24}$$

Differentiating the above equation with respect to ω, we have

$$\left(\frac{d}{d\omega}\mathbf{F}(j\omega) - \frac{d\lambda}{d\omega}\mathbf{I}\right)\mathbf{u} + (\mathbf{F}(j\omega) - \lambda\mathbf{I})\frac{d\mathbf{u}}{d\omega} = 0. \tag{4.25}$$

Then (4.25) is left-multiplied by $\mathbf{v}^T$, which is the transposed left eigenvector of $\mathbf{F}(j\omega)$

$$\mathbf{v}^T \frac{d}{d\omega}\mathbf{F}(j\omega)\mathbf{u} - \mathbf{v}^T \frac{d\lambda}{d\omega}\mathbf{u} + \mathbf{v}^T \left(\mathbf{F}(j\omega) - \lambda\mathbf{I}\right) \frac{d\mathbf{u}}{d\omega} = 0. \tag{4.26}$$

By the definition of left eigenvector, $\mathbf{v}^T \left(\mathbf{F}(j\omega) - \lambda\mathbf{I}\right)$ in the last term of (4.26) is equal to zero. Taking advantage of this property, (4.26) can be simplified as

$$\mathbf{v}^T \frac{d}{d\omega}\mathbf{F}(j\omega)\mathbf{u} = \mathbf{v}^T \frac{d\lambda}{d\omega}\mathbf{u} \tag{4.27}$$

or

$$\frac{d\lambda}{d\omega} = \frac{\mathbf{v}^T \frac{d}{d\omega}\mathbf{F}(j\omega)\mathbf{u}}{\mathbf{v}^T\mathbf{u}}. \tag{4.28}$$

Notice that $\mathbf{F}(j\omega)$ can be expressed using (4.7)

$$\begin{aligned} Real(\mathbf{Y}(j\omega)) &= \mathbf{F}(j\omega) \\ &= -\mathbf{CA}(\omega^2\mathbf{I} + \mathbf{A}^2)^{-1}\mathbf{B} + \mathbf{D}, \end{aligned} \tag{4.29}$$

then the derivative of $\mathbf{F}(j\omega)$ with respect to ω on the right hand-side of (4.28) is easily derived as

$$\frac{d}{d\omega}\mathbf{F}(j\omega) = \mathbf{CA}(\omega^2\mathbf{I} + \mathbf{A}^2)^{-2}2\omega\mathbf{B}. \tag{4.30}$$

By substituting (4.30) for (4.28), we obtain the formula for calculating the slope of the eigenvalue of $\mathbf{F}(j\omega)$

$$\frac{d\lambda}{d\omega} = \frac{\mathbf{v}^T \left(\mathbf{CA}(\omega^2\mathbf{I} + \mathbf{A}^2)^{-2}2\omega\mathbf{B}\right)\mathbf{u}}{\mathbf{v}^T\mathbf{u}}. \tag{4.31}$$

Most practical interconnect networks are multi-port networks. The corresponding admittance matrices are of the same size and have multiple eigenvalues. One should make sure that when evaluating (4.31), the correct eigenvectors are used. For example, we evaluate the slope of $\mathbf{F}(j\omega)$ at ω_k for a three-port network. So $\mathbf{F}(j\omega)$ is a three-by-three matrix and has three eigenvalues. It is not obvious that the eigenvectors ($\mathbf{u}$ and $\mathbf{v}$) belonging to which eigenvalue should be used in (4.31) to evaluate the slope. It is useful to note that ω_k corresponds to the frequency where $\mathbf{F}(j\omega_k)$ is singular. Thus, the solution is to evaluate the three eigenvalues of $\mathbf{F}(j\omega_k)$ and find the one closest to zero (It should be ideally zero, and

numerical noise is considered here.). The eigenvectors corresponding to that eigenvalue are applied to (4.31).

In order to determine the location of each violation region, the follows steps are used:

1. Construct the Hamiltonian matrix using Equation (4.23).
2. Calculate the eigenvalues of the resulting Hamiltonian matrix $\mathbf{M}$ and collect the imaginary eigenvalues in a vector, whose entries are arranged in an ascending order $\mathbf{G} = [\omega_1, \omega_2, \ldots, \omega_N]$ such that $\omega_1 < \omega_2 < \cdots < \omega_N$.
3. Evaluate the eigenvalue of $\mathbf{F}(j\omega)$ at the frequency point corresponding to each entry of $\mathbf{G}$ using (4.31). Since the macromodel is asymptotically passive at $\omega = \infty$ ($\mathbf{D}+\mathbf{D}^T > 0$), the slope at the highest frequency (ω_N) must always be positive.
4. Count the number of positive and negative slopes from the highest frequency point (ω_N). When the numbers of positive slopes and negative slopes become equal stop counting. Assume the count stops at ω_k, the first passivity violation region is located at $[\omega_k, \omega_N]$.
5. Reset the counter and start counting from ω_{k-1}. Repeat Step 3 and Step 4 until all imaginary eigenvalues are exhausted.

One should pay special attention to Step 2, because the imaginary eigenvalues of $\mathbf{M}$ may not be *purely* imaginary due to the numerical noise generated during computation [38]. Fortunately, the eigenvalue spectrum of the Hamiltonian matrix is symmetric with reference to both real and imaginary axes. Taking advantage of this property, the pure imaginary eigenvalues are detected by checking their eigenvalues which are symmetric only with respect to the real axis. As a result, the effect of numerical noise is removed.

It is also worth noting that it is not always true that the numbers of positive slopes and negative slopes are equal for the last violation region, since the passivity violation may start from origin (Fig. 4.3). In this case, the violation region is simply from origin to the frequency point where the current-round count starts at.

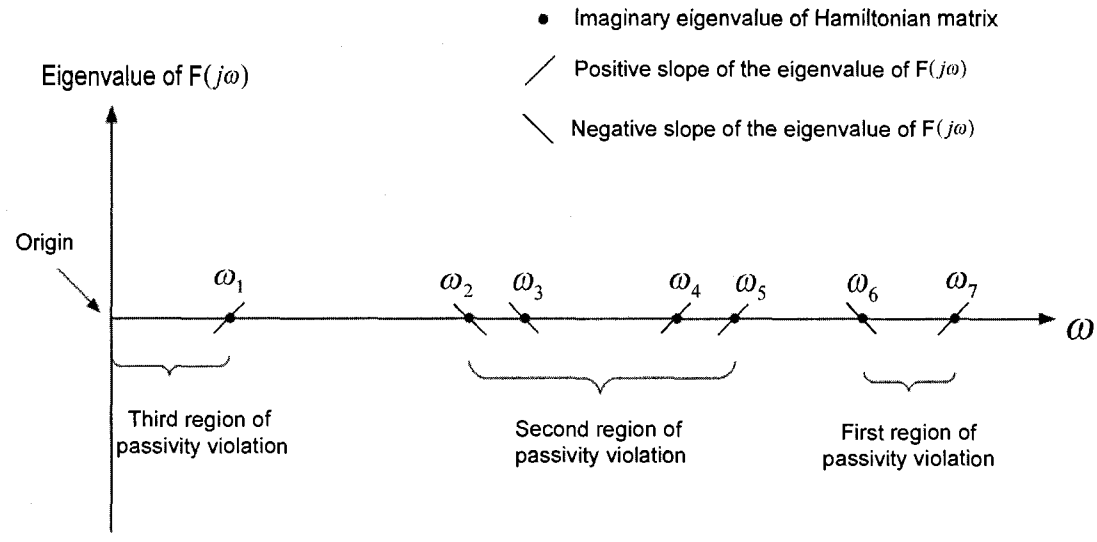

Fig. 4.3 Regions of passivity violation identified. The third violation region starts from the origin.

4.4 Conclusion

In this chapter, we discussed and compared some of the most widely used methods for passivity check. The method based on frequency-domain passivity criteria is straightforward but suffers from high CPU cost and misjudgment. It is therefore not recommended, but can be used as a supplementary method for result verification. The methods based on the space-state representation are used as the major methods for passivity check in this thesis. Most significantly, the method discussed in the previous section can also identify the exact location of passivity violation. This merit is very useful for the compensation process that is going to be discussed in Chapter 5.

Chapter 5

Passivity Compensation

In this chapter we proceed to the issue of passivity compensation. In case there is passivity violation detected by any method discussed in Chapter 4, a compensation process is necessary to fix that. The algorithm used in this thesis is the one based on [28]. The major advantage of this algorithm is that it features *global* passivity enforcement; no additional passivity violation is generated during the compensation process. But for applications in this thesis, a different method of calculating the perturbing value of the residues (i.e. $\Delta\mathbf{C}$) is proposed. The method can guarantee the minimum change in response.

5.1 Theory

In order to demonstrate the compensation process, we start with a M-port macromodel generated using Vector Fitting algorithm

$$\mathbf{Y}(s) = \begin{bmatrix} Y_{11}(s) & Y_{12}(s) & \dots & Y_{1M}(s) \\ Y_{21}(s) & Y_{22}(s) & \dots & Y_{2M}(s) \\ \vdots & \vdots & \ddots & \vdots \\ Y_{M1}(s) & Y_{M2}(s) & \dots & Y_{MM}(s) \end{bmatrix}. \tag{5.1}$$

Each element of the above matrix is in the pole-residue form

$$Y_{i,j}(s) = \sum_{n=1}^{N} \frac{k_n^{i,j}}{s - p_n^{i,j}} + c^{i,j}; \text{ for } i,j=0,1,\dots,M \tag{5.2}$$

where N is the number of poles/residues. Then the state-space model ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$) is obtained by following the guideline in Chapter 4. The resulting macromodel described by the state-space representation is checked for passivity by evaluating the eigenvalues of the Hamiltonian matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} - \mathbf{B}(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{C} & \mathbf{B}(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{B}^T \\ \mathbf{C}^T(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{C} & -\mathbf{A}^T + \mathbf{C}^T(\mathbf{D} + \mathbf{D}^T)^{-1}\mathbf{B}^T \end{bmatrix}. \tag{5.3}$$

In case of pure imaginary eigenvalues detected (i.e. nonpassive), the exact location of each passivity violation is identified using the method described in Chapter 4. The next step is to find the location and magnitude of the maximum violation (i.e. the most negative eigenvalue of $\mathbf{F}(j\omega)$ given by (4.29)), for each region of violation. This can be done by solving the following problem

$$min(eig(\mathbf{F}(j\omega))) \;\; \omega \in (\omega_l, \omega_h), \tag{5.4}$$

where ω_l and ω_h are respectively the lower and upper boundaries of the same region of passivity violation. The most negative value is considered as the worst case of passivity violation. In order to make the macromodel passive at the frequency at which the maximum violation is located, we need to offset its eigenvalue at that frequency point by the absolute value of (5.4). To achieved this goal, the real part of $\mathbf{Y}(j\omega)$ (i.e. $\mathbf{F}(j\omega)$) need to be added by some $\Delta\mathbf{F}(j\omega)$, such that

$$\mathbf{F}(j\omega) + \Delta\mathbf{F}(j\omega) \geq 0. \tag{5.5}$$

Then recall

$$\mathbf{F}(j\omega) = -\mathbf{C}\mathbf{A}(\omega^2\mathbf{I} + \mathbf{A}^2)^{-1}\mathbf{B} + \mathbf{D}. \tag{5.6}$$

From (5.6), if we keep the matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{D}$ unchanged and slightly perturb $\mathbf{C}$, the resulting change in $\mathbf{C}$, denoted by $\Delta\mathbf{C}$, can lead to a desired $\Delta\mathbf{F}(j\omega)$ that satisfies (5.5). From this point on, to offset the negative eigenvalue of $\mathbf{F}(j\omega)$ (or make $\mathbf{F}(j\omega)$ positive real) at the frequency of maximum passivity violation, we look for some $\Delta\mathbf{C}$, such that

$$\Delta\mathbf{F}(j\omega) = -\Delta\mathbf{C}\mathbf{A}(\omega^2\mathbf{I} + \mathbf{A}^2)^{-1}\mathbf{B} + \mathbf{D}. \tag{5.7}$$

By applying the eigenvalue perturbation theory [56] to the above equation, we are able to relate the unknown $\Delta\mathbf{C}$ to the most negative eigenvalue of $\mathbf{F}(j\omega)$ given by (5.4):

$$\begin{aligned}\Delta\lambda &= \frac{\mathbf{y}^T\mathbf{F}(j\omega)\mathbf{x}}{\mathbf{y}^T\mathbf{x}} \\ &= \frac{\mathbf{y}^T\Delta\mathbf{C}A(\omega^2\mathbf{I}+\mathbf{A}^2)^{-1}\mathbf{B}\mathbf{x}}{\mathbf{y}^T\mathbf{x}},\end{aligned} \tag{5.8}$$

where $\mathbf{y}$ and $\mathbf{x}$ are respectively the left and right eigenvectors of $\mathbf{F}(j\omega)$ at the frequency of the maximum violation and $\Delta\mathbf{C}$ contains the unknowns. Before solving the above equation, it is necessary to simplify it first. Be aware of the fact that $\mathbf{F}(j\omega)$ is a real symmetric matrix (real part of $\mathbf{Y}(j\omega)$). Its left eigenvector is equal to its right eigenvector. Moreover, if the eigenvectors are normalized, the denominator of (5.8) is equal to the scalar 1. So (5.8) is simplified to

$$\Delta\lambda = \mathbf{y}^T\Delta\mathbf{C}A(\omega^2\mathbf{I}+\mathbf{A}^2)^{-1}\mathbf{B}\mathbf{x}. \tag{5.9}$$

Since the unknowns in (5.9) are not explicitly indicated, the above equation can be further converted to a more convenient expression by taking advantage of the property of the Kronecker product [57]

$$\begin{aligned}\Delta\lambda &= \left(\mathbf{A}(\omega^2\mathbf{I}+\mathbf{A}^2)^{-1}\mathbf{B}\mathbf{x}\right)^T \otimes \left(\mathbf{y}^T\right) \mathrm{vec}\left(\Delta\mathbf{C}\right) \\ &= \underbrace{\mathbf{x}^T\mathbf{B}^T\left(\left(\omega^2\mathbf{I}+\mathbf{A}^2\right)^{-1}\right)^T\mathbf{A}^T \otimes \left(-\mathbf{y}^T\right)}_{\Theta}\underbrace{\mathrm{vec}(\Delta\mathbf{C})}_{\mathbf{Q}},\end{aligned} \tag{5.10}$$

where "$\otimes$" denotes the Kronecker product operator, "Θ" is a row vector and "$\mathbf{Q}$" is the column vector containing the unknown perturbed value of $\Delta\mathbf{C}$. Notice that since the residues obtained from Vector Fitting algorithm is already accurate, the macromodel must suffer some accuracy degradation after the compensation operation. To minimize the changes in responses due to the compensation, only a limited number of residues are perturbed. Actually, the perturbation should be only effective on the real parts of the residues corresponding to the poles located in the vicinity of the passivity violation. By setting this constraint, the accuracy degradation will be confined within the vicinity of passivity violation; it will not be spreaded throughout the whole bandwidth of interest. The selection of the poles is done by comparing the contribution of each pole in the vicinity of passivity violation to

the real part of the diagonal entries of the admittance matrix. This can be achieved by integrating the square of the real part of the frequency response, (i.e. $Real(\mathbf{Y}(j\omega))$) over the frequency range of passivity violation. That is

$$\int_{\omega_L}^{\omega_H} Real^2\left(\mathbf{Y}\left(j\omega\right)\right) d\omega, \tag{5.11}$$

where ω_L and ω_H are the lower and upper boundaries of the passivity violation. Only the poles with significant contribution (i.e. with comparably large integration value in (5.11)) are selected for future compensation. In the next step, the chosen residues need to be mapped to the residue matrix $\mathbf{C}$. This process is illustrated by the following example. For an M-port macromodel with N pairs of complex conjugate poles and the corresponding residues $c_{p,q}^k = r_{p,q}^k \pm j g_{p,q}^k$, for $k = 1, 2, \ldots, N$ and $p, q = 1, 2, \ldots, M$, by (4.20), we have

$$\mathbf{C} = \begin{bmatrix} r_{11}^1 & r_{12}^1 & \cdots & r_{1M}^1 & \cdots & r_{11}^N & r_{12}^N & \cdots & r_{1M}^N & \cdots & g_{11}^1 & g_{12}^1 & \cdots & g_{1M}^1 & \cdots & g_{11}^N & g_{12}^N & \cdots & g_{1M}^N \\ r_{21}^1 & r_{22}^1 & \cdots & r_{2M}^1 & \cdots & r_{21}^N & r_{22}^N & \cdots & r_{2M}^N & \cdots & g_{21}^1 & g_{22}^1 & \cdots & g_{2M}^1 & \cdots & g_{21}^N & g_{22}^N & \cdots & g_{2M}^N \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{M1}^1 & r_{M2}^1 & \cdots & r_{MM}^1 & \cdots & r_{M1}^N & r_{M2}^N & \cdots & r_{MM}^N & \cdots & g_{M1}^1 & g_{M2}^1 & \cdots & g_{MM}^1 & \cdots & g_{M1}^N & g_{M2}^N & \cdots & g_{MM}^N \end{bmatrix}. \tag{5.12}$$

If the k^{th} complex pole pair is identified by (5.11), the resulting $\Delta\mathbf{C}$ is represented as

$$\Delta\mathbf{C} = \begin{bmatrix} 0 & \ldots & 0 & \Delta r_{11}^k & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \Delta r_{22}^k & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & & & \vdots & & \ddots & \vdots & \vdots & \ldots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 0 & \Delta r_{MM}^k & 0 & \ldots & 0 \end{bmatrix}, \tag{5.13}$$

with unknowns appearing only in the corresponding positions in 5.12. The vector $\mathbf{Q}$ is formulated as $\mathbf{Q} = \left[\Delta r_{11}^k, \Delta r_{22}^k, \ldots, \Delta r_{MM}^k\right]^T$. In case more than one pair of complex poles are selected by (5.11), $\Delta\mathbf{C}$ will have block diagonal entries corresponding to the selected poles. For example, if the k^{th} and l^{th} pole pairs are selected, ΔC will be in the form of

$$\Delta\mathbf{C} = \begin{bmatrix} 0 & \dots & 0 & \Delta r_{11}^k & 0 & \dots & 0 & 0 & \dots & \Delta r_{11}^l & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \Delta r_{22}^k & \dots & 0 & 0 & \dots & 0 & \Delta r_{22}^l & \dots & 0 & \dots & 0 \\ \vdots & & & \vdots & & \ddots & \vdots & \vdots & \dots & \vdots & & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \Delta r_{MM}^k & 0 & \dots & 0 & \dots & 0 & \Delta r_{MM}^l & \dots & 0 \end{bmatrix}, \tag{5.14}$$

and consequently $\mathbf{Q} = [\Delta r_{11}^k, \Delta r_{22}^k, \dots, \Delta r_{MM}^k, \Delta r_{11}^l, \Delta r_{22}^l, \dots, \Delta r_{MM}^l]^T$. In general, all the non-zero entries of $\Delta\mathbf{C}$ are stacked vertically to form the column vector $\mathbf{Q}$. Then, (5.10) is solved for $\mathbf{Q}$ or $\Delta\mathbf{C}$. Notice that (5.10) is an under-determined problem that has a rank less than the number of unknowns (i.e. There are infinity sets of possible solutions satisfying this equation.). Hence, a linear optimization is required to obtain the optimum solution to (5.10). Based on this discussion, the problem can be formulated as

minimizing $\| \mathbf{Q} \|^2$ *subject to the constraints*:

a) $\Delta\lambda = \Theta\mathbf{Q}$

b) *All entries of* $\mathbf{Q}$ *are positive numbers*

where $\| \mathbf{Q} \|^2$ denotes the Frobenius norm of $\mathbf{Q}$ squared. By minimizing the Frobenius norm of the unknown vector we can keep the loss of accuracy in response as low as possible. The problem (5.10) is solved by enforcing Constraint **a**). Constraint **b**) is used to guarantee the change in the real part of the selected residues is always positive. Notice that the real part of an arbitrary function $\mathbf{H}(j\omega)$ due to a complex pole pair, $p_{1,2} = -\alpha \pm j\beta$, and the corresponding residues $k_{1,2} = r \pm jg$ is [38, 58]:

$$real(\mathbf{H}(j\omega)) = \frac{2r\alpha(\alpha^2 + \beta^2 + \omega^2) - 2g\beta(\alpha^2 + \beta^2 - \omega^2)}{(\alpha^2 + \beta^2 - \omega^2)^2 + 4\alpha^2\omega^2}. \tag{5.15}$$

It is evident the change in $real(\mathbf{H}(j\omega))$ due to some perturbation of the real part of the residue Δd_1 can be expressed as

$$\Delta(real(\mathbf{H}(j\omega))) = \frac{2\Delta r\alpha(\alpha^2 + \beta^2 + \omega^2)}{(\alpha^2 + \beta^2 - \omega^2)^2 + 4\alpha^2\omega^2}. \tag{5.16}$$

(5.16) implies that $\Delta(real(\mathbf{H}(j\omega)))$ is linearly proportional to Δr. Taking advantage of this property, we have all the perturbations of the diagonal entries of $\mathbf{F}(j\omega)$ positive (i.e. $\Delta\mathbf{F}_{11}(j\omega), \Delta\mathbf{F}_{22}(j\omega), \ldots, \Delta\mathbf{F}_{MM}(j\omega) > 0$), as Constraint **b**) is enforced to guarantee a positive Δr. Once this perturbed value is added to the original $\mathbf{F}(j\omega)$, the eigenvalues of $\mathbf{F}(j\omega)$ are offset by the values given by

$$\Delta\lambda = \frac{\mathbf{y}^T \Delta\mathbf{F}(j\omega)\mathbf{x}}{\mathbf{y}^T\mathbf{x}}, \tag{5.17}$$

where $\mathbf{x}$ and $\mathbf{y}$ are respectively the right and left eigenvectors of $\mathbf{F}(j\omega)$. In fact, the left eigenvector is equal to the right eigenvector, since $\mathbf{F}(j\omega)$ is a real symmetric matrix. if the eigenvectors are further normalized, for an M-port network the above equation can be expanded as

$$\Delta\lambda = \Delta\mathbf{F}_{11}(j\omega)x_1^2 + \Delta\mathbf{F}_{22}(j\omega)x_2^2 + \cdots + \Delta\mathbf{F}_{MM}(j\omega)x_M^2. \tag{5.18}$$

From (5.18) it is evident that the perturbation only adds positively to the eigenvalues of $\mathbf{F}(j\omega)$, since $\Delta\mathbf{F}_{11}(j\omega), \Delta\mathbf{F}_{22}(j\omega), \ldots, \Delta\mathbf{F}_{MM}(j\omega)$ have been proved to be positive numbers. Hence, it becomes a nice feature of this method since the compensation only fixes the existing passivity violations and never introduces additional passivity violations to the macromodel.

The method compensates the region of passivity violation iteratively. In each iteration only the highest passivity violation (the one of the highest frequency) is compensated. After each iteration the macromodel is again checked for passivity and the regions of passivity violation are updated for another iteration. The general procedure of the compensation operation is summarized in Fig. 5.1. This method features a fixed direction of passivity compensation. Thus, it never generates additional regions of passivity violation during the compensation process.

Step 1. Construct the Hamiltonian matrix from the macromodel using Equation (4.23) and evaluate its eigenvalues. If no pure imaginary eigenvalues are detected, stop the iteration. Otherwise proceed to Step 2.

Step 2. Follow the procedure outlined in Section 4.3 to determine the location for the region of passivity violation.

Step 3. Determine the frequency and the magnitude of the maximum (most negative) point within the highest region of passivity violation (the one of the highest frequency) using Equation (5.4).

Step 4. Select the significant poles for compensation by evaluating Equation (5.11) for each pole in the vicinity of the highest passivity violation.

Step 5. Formulate Equation (5.10) and solve it with the linear optimization (discussed in Section 5.1) to compensate the highest violation ***only.***

Step 6. Update the residues and go back to *Step 1*.

Fig. 5.1 Pseudocode of the compensation algorithm

5.2 Numerical Results

In this section, three numerical examples will be given as a better illustration of the mechanism of the method. Both frequency-domain and time-domain responses will be compared with those of the original network. A limitation of the method will also be discussed, and a proper solution to that will be given.

5.2.1 Example 1

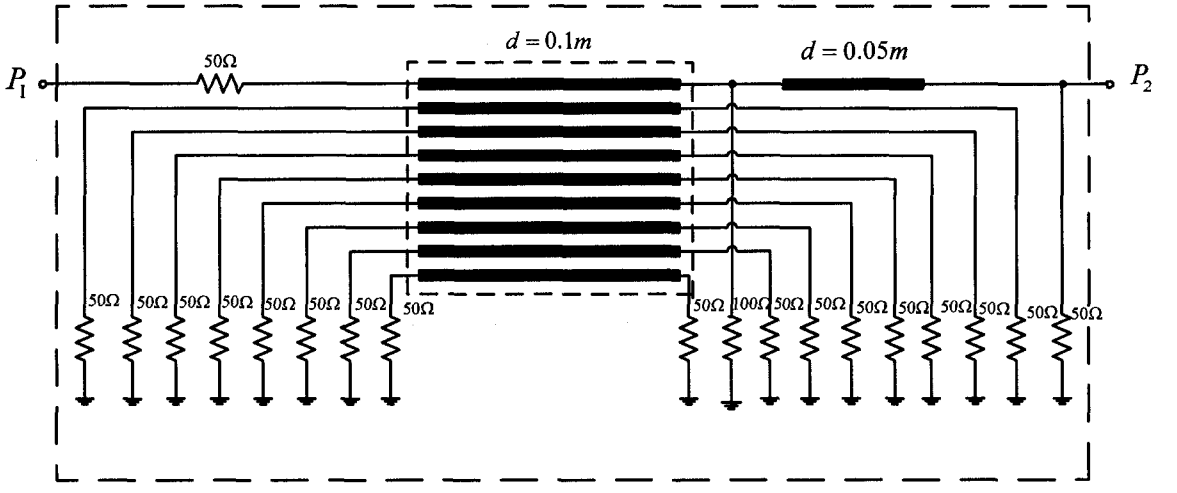

Fig. 5.2 Example 1: Two-port interconnect containing coupled transmission lines

The first example is a two-port network consisting of coupled transmission lines and resistors, shown in Fig. 5.2. The subnetwork is characterized by a set of Y-parameter simulated from DC to 4 GHz. The macromodel was constructed using Vector Fitting algorithm. Thirty-four complex poles and two real poles are required to obtain a reliable accuracy. The Y-parameters calculated from the macromodel were compared with those simulated from the original network. A very good match was achieved (Fig. 5.3). Next, the state-space model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ was constructed from the poles and residues, and the resulting model was checked for passivity by evaluating the eigenvalues of the Hamiltonian matrix. Since no pure imaginary eigenvalue was detected, the macromodel is passive and there's no need for compensation. To make sure the judgment is correct, the conventional method of plotting the eigenvalues of the real part of $\mathbf{Y}(j\omega)$ has been carried out as well

in Fig. 5.4. The first and second eigenvalues are plotted to 12 GHz and no negative values were detected. Following that, the SPICE model was constructed using the method described in Appendix **A** and used as the netlist file of HSPICE for the transient simulation. The voltage at Port 2 was simulated and compared with the simulation of the original network Fig. 5.5). A pulse of 0.4 GHz with both rise and fall times equal to 0.25 ns was connected to Port 1 and the simulation stopped at 10 ns. The comparison shows that the results agree with each other.

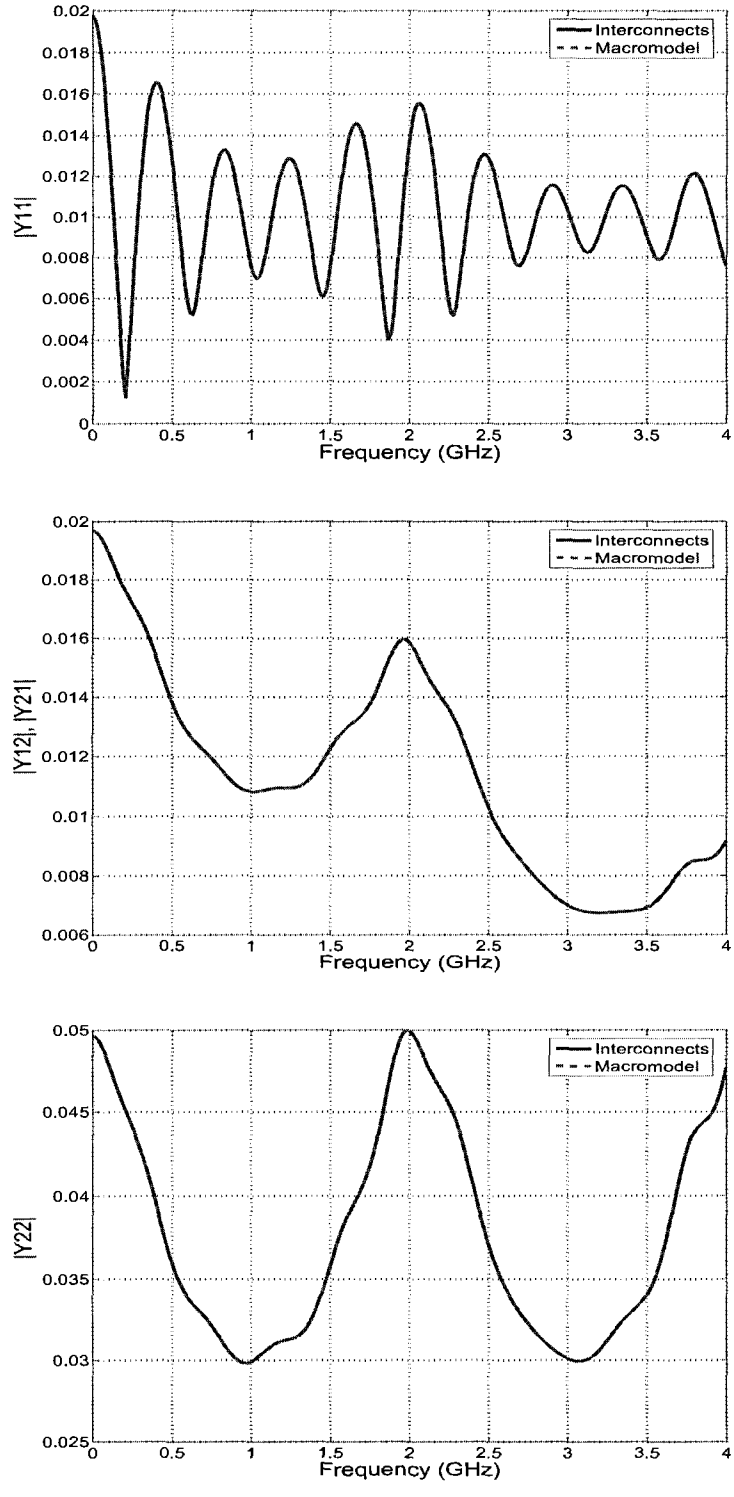

Fig. 5.3 Admittance parameter comparison - Example 1

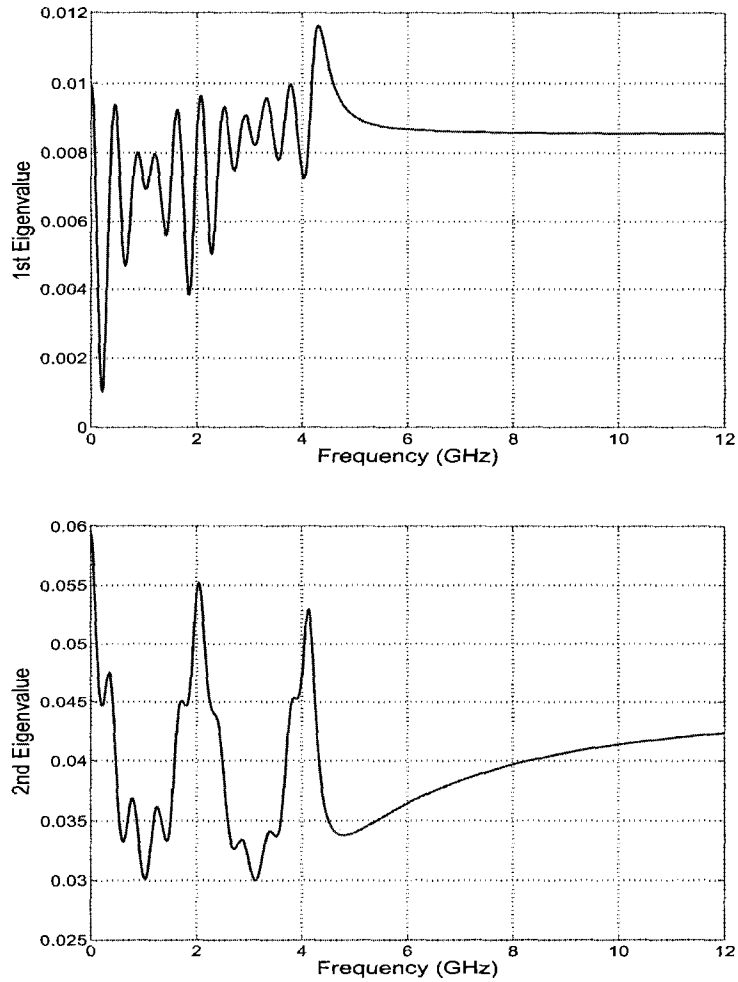

Fig. 5.4 Eigenvalues of the real part of $\mathbf{Y}(j\omega)$ - Example 1

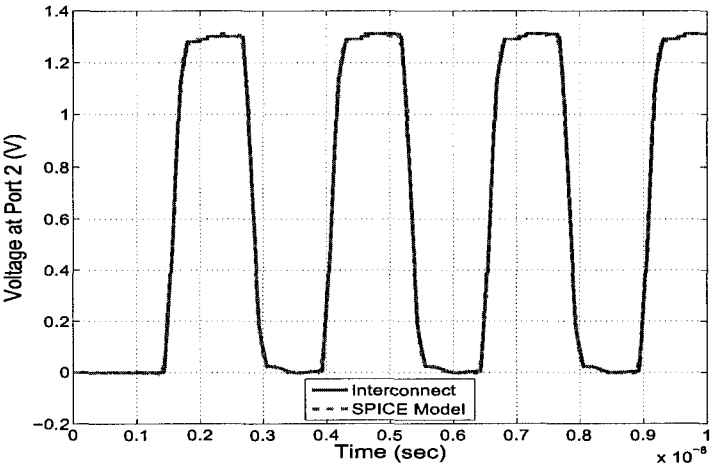

Fig. 5.5 Transient response comparison: voltage at Port 2 - Example 1

5.2.2 Example 2

The second example was created based on Example 1. The difference is changes in some of the components' values in order to make the network more susceptive to passivity violation. The resulting network was simulated to 3 GHz to get the Y-parameters. Vector Fitting was applied to get the macromodel for the tabulated data. Thirty poles (twenty-eight complex poles and two real poles) are required to achieve a good accuracy. Then, the corresponding state-space model was checked for passivity and passivity violation was detected out of band (Fig. 5.7). Thus, the macromodel was compensated for passivity using the method discussed in the previous section. The eigenvalue after compensation is shown in Fig. 5.8 and no more negative value was detected. The compensated macromodel was simulated to get the responses, which were compared with the original Y-parameters obtained from the network (Fig. 5.9 and Fig. 5.10). A good match has been achieved showing that the compensation process did not affect the accuracy. Details about this process are summarized in Table 5.1. The transient simulation of the SPICE model was carried out by connecting a pulse of 0.4 GHz with both rise and fall times equal to 0.25 ns to Port 1 and simulating the voltage at Port 2. The comparison in Fig. 5.11 shows that the macromodel is accurate in the time domain as well.

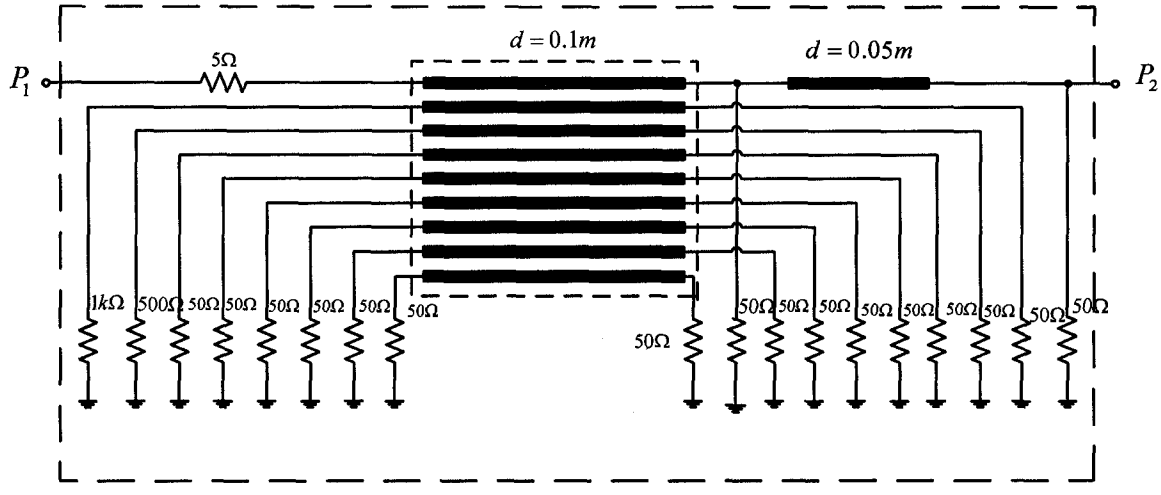

Fig. 5.6 Example 2: Two-port interconnect susceptive to passivity violation

Table 5.1 Details of the compensation process: boundaries of passivity violation, maximum violation location and value, pole(s) selected for compensation, residue(s) prior to compensation, residue(s) after compensation and evaluation of error - Example 2

		Violation region 1
Violation information	f_L	3.57 GHz
	f_H	3.66 GHz
	f_{max}	3.62 GHz
	$\min(\lambda)$	$-1.183E-3$
Pole(s) & residue(s) selected for compensation	pole(s)	-0.9533 ± 22.4067
	$Y_{1,1}$	$-2.4347E-2 \mp 2.6487E-2i$
	$Y_{2,2}$	$-3.1483E-2 \mp 3.2417E-2i$
Residue(s) after compensation	$Y_{1,1}$	$-2.3032E-2 \mp 2.6487E-2i$
	$Y_{2,2}$	$-3.1476E-2 \mp 3.2417E-2i$
$\frac{\|\Delta \mathbf{C}\|_{fro}}{\|\mathbf{C}\|_{fro}}$		$4.2E-3$

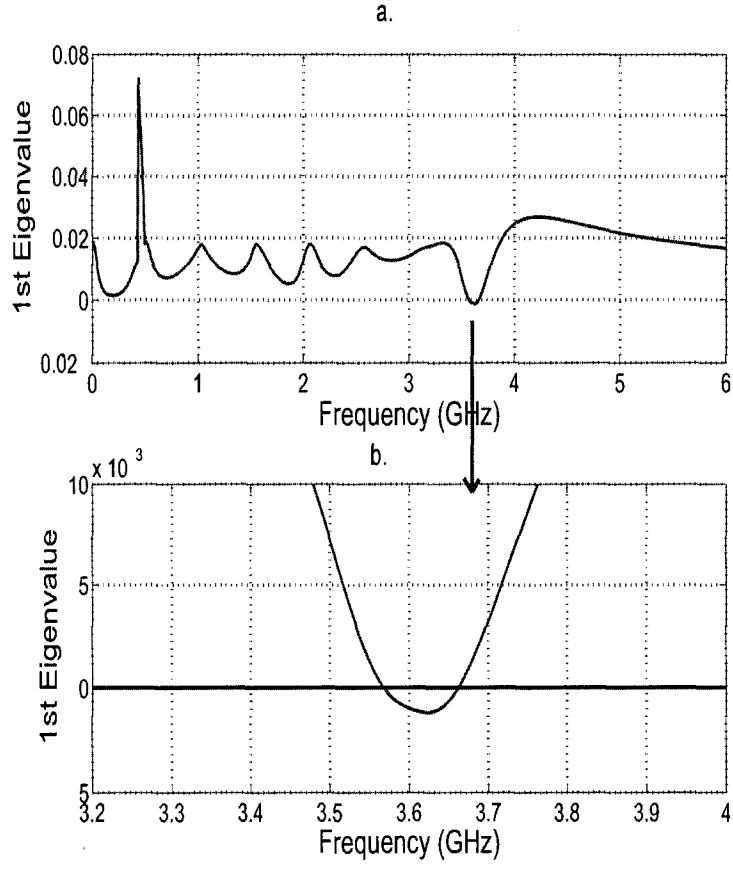

Fig. 5.7 **a.** The first eigenvalue of the real part of $\mathbf{Y}(j\omega)$ **b.** Enlarged view of the passivity violation

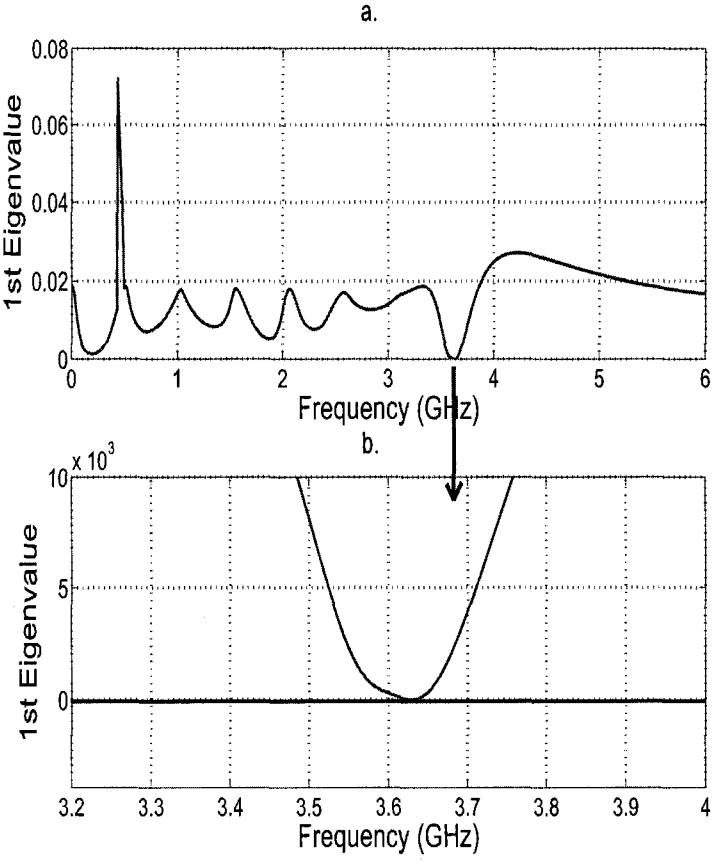

Fig. 5.8 **a.** The first eigenvalue of the real part of $\mathbf{Y}(j\omega)$ after compensation of passivity violation **b.** Enlarged view of the change in eigenvalue

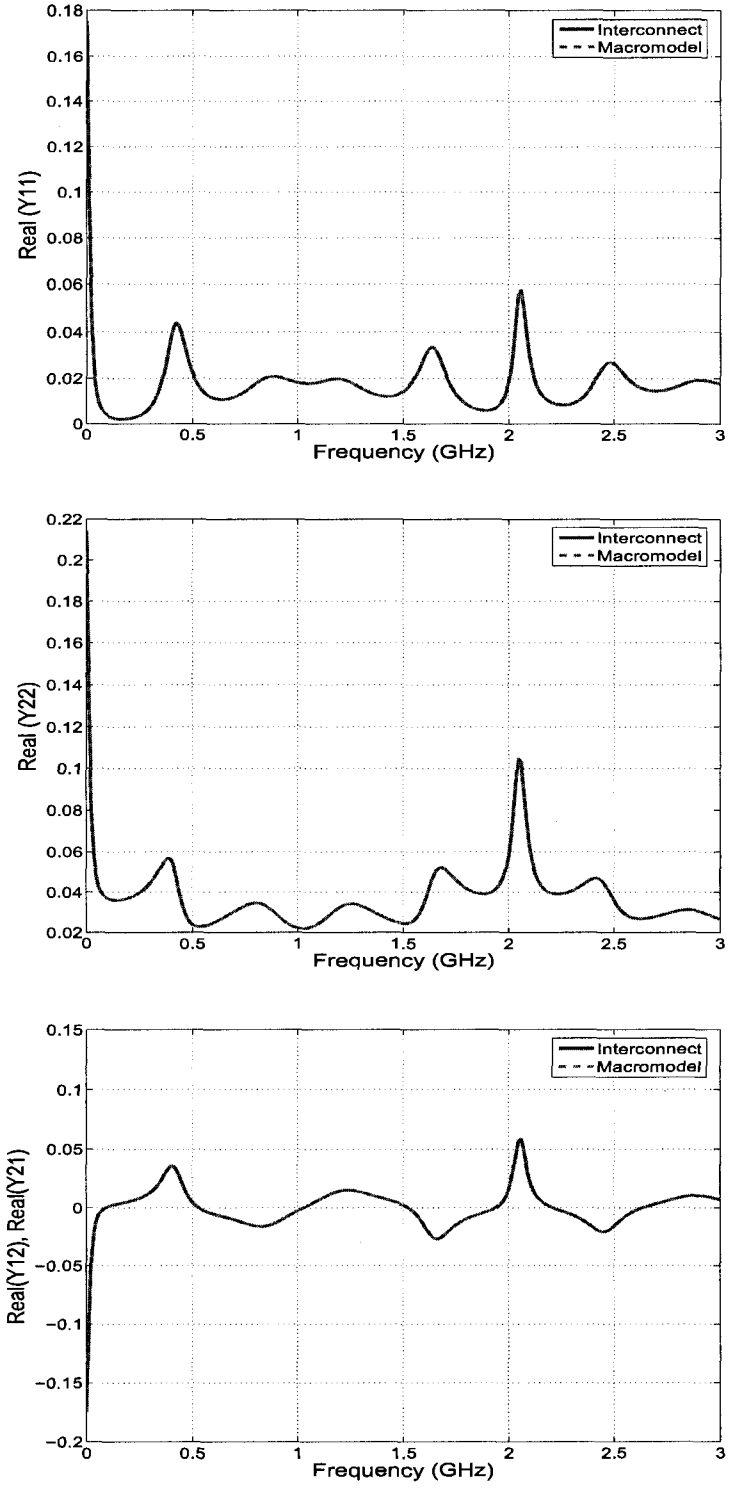

Fig. 5.9 Y-parameters(real part) comparison after compensation - Example 2

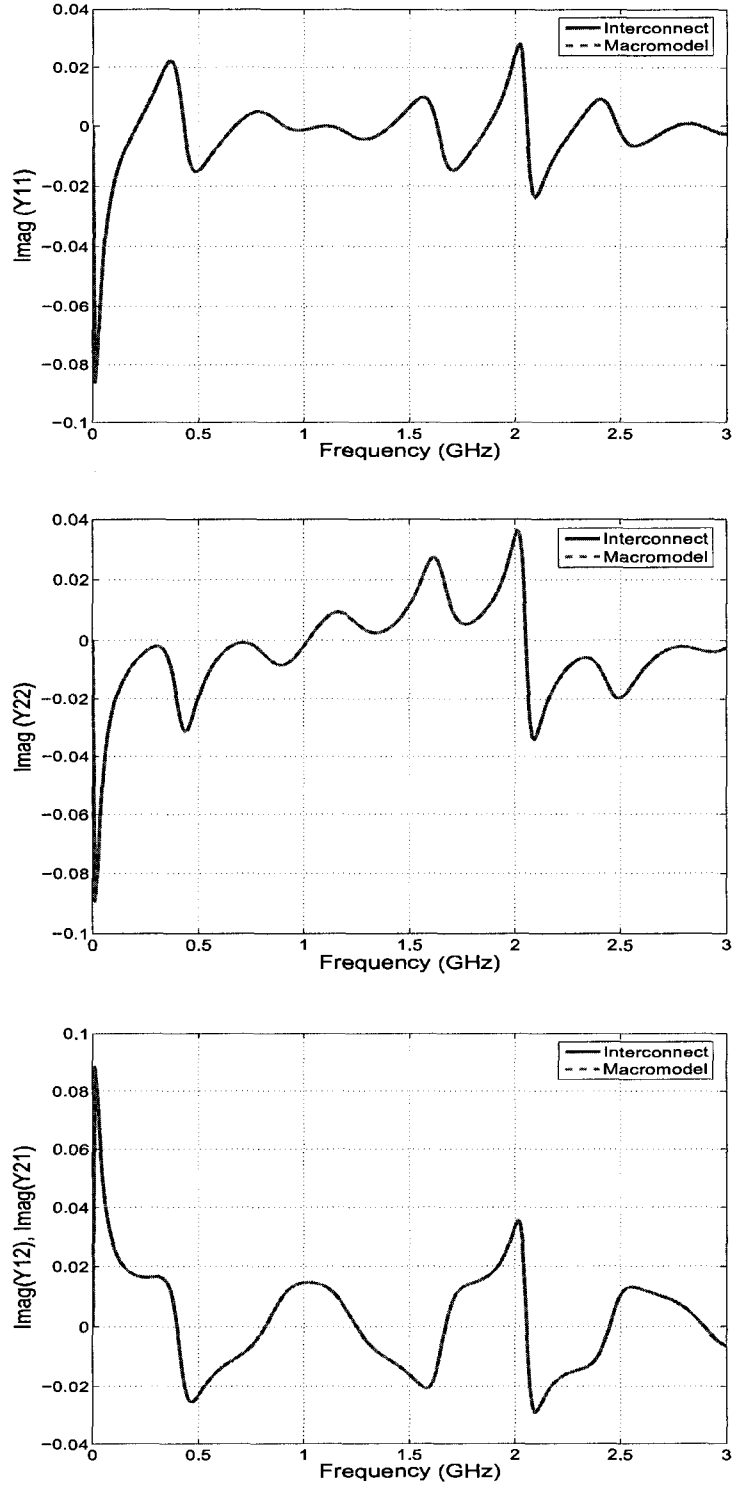

Fig. 5.10 Y-parameters(imaginary part) comparison after compensation - Example 2

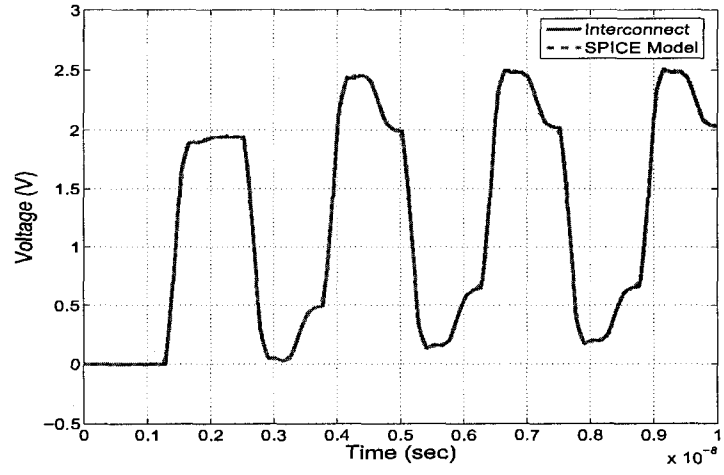

Fig. 5.11 Transient response comparison: voltage at Port 2 - Example 2

5.2.3 Example 3

Example 3 is a three-port interconnect consisting of R, L, C components and transmission lines. The network was simulated from DC to 4 GHz. The resulting Y-parameters were fitted using Vector Fitting algorithm followed by generation of state-space model. By checking the passivity of the macromodel, passivity violation was found right out of band (shown in Fig. 5.14). Then the passivity violation was compensated and the resulting macromodel was again checked for passivity. It was found the passivity violation was fixed and no additional violation was created (Fig. 5.15). To evaluate the preservation of accuracy after compensation, the responses of the macromodel were compared with the Y-parameter from measurement(Fig. 5.16 and Fig. 5.17). From the comparison, we can see very good accuracy has been preserved except for the imaginary part of Y_{33}. The reason and solution for that is given in the later part of this chapter. The details of the compensation process are summarized in Table 5.2. The transient simulation with the same source as Example 1 and 2 was applied to Example 3 to verify its accuracy in the time domain. Good match was achieved between the SPICE model and the original interconnect (Fig. 5.18).

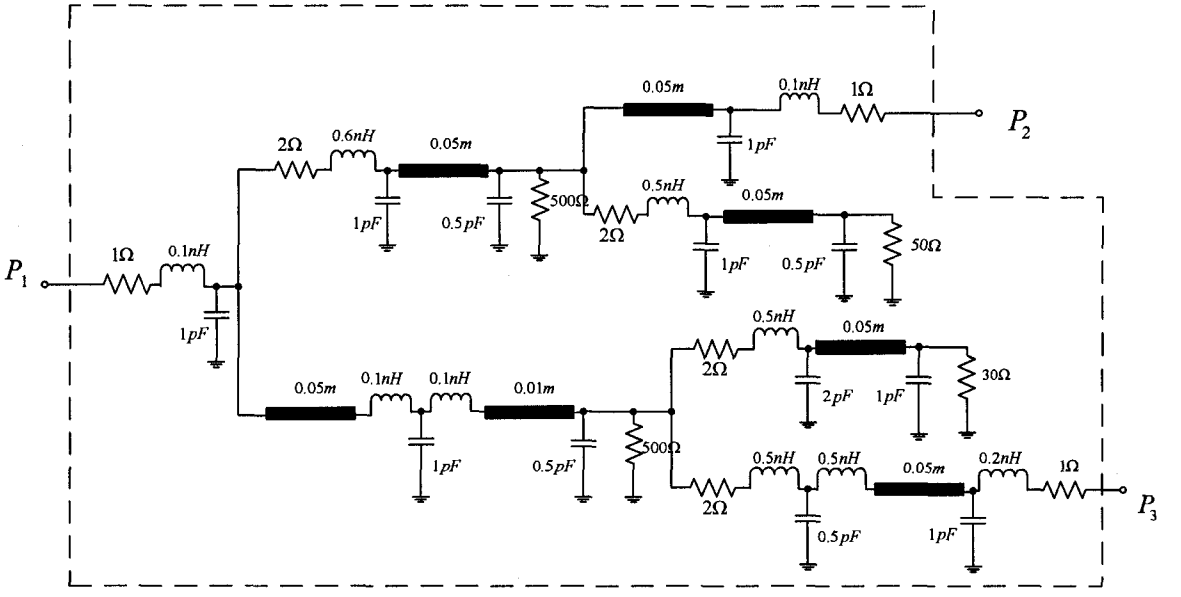

Fig. 5.12 Example 3

Table 5.2 Details of the compensation process: boundaries of passivity violation, maximum violation location and value, pole(s) selected for compensation, residue(s) prior to compensation, residue(s) after compensation and evaluation of error - Example 3

		Violation region 1
Violation information	f_L	4.06 GHz
	f_H	4.34 GHz
	f_{max}	4.11 GHz
	$\min(\lambda)$	-0.0166
Pole(s)& residue(s) selected for compensation	pole(s)	$-0.1696 \pm 25.6969i$
	$Y_{1,1}$	$-4.5128E-4 \pm 1.7952E-4i$
	$Y_{2,2}$	$7.7466E-6 \mp 2.8332E-3i$
	$Y_{3,3}$	$-4.4557E-4 \mp 5.6845E-3i$
Residue(s) after compensation	$Y_{1,1}$	$4.3588E-3 \pm 1.7952E-4i$
	$Y_{2,2}$	$7.3231E-4 \mp 2.8332E-3i$
	$Y_{3,3}$	$2.3318E-4 \mp 5.6845E-3i$
$\frac{\|\Delta C\|_{fro}}{\|C\|_{fro}}$		$1.74E-2$

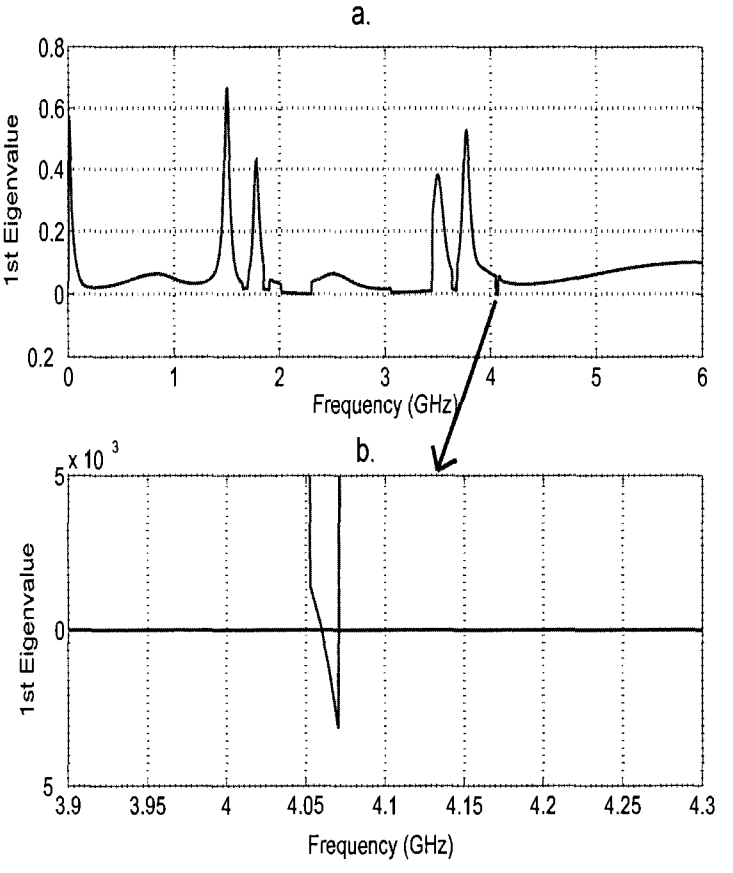

Fig. 5.13 **a.** The first eigenvalue of the real part of $\mathbf{Y}(j\omega)$ **b.** Enlarged view of the passivity violation

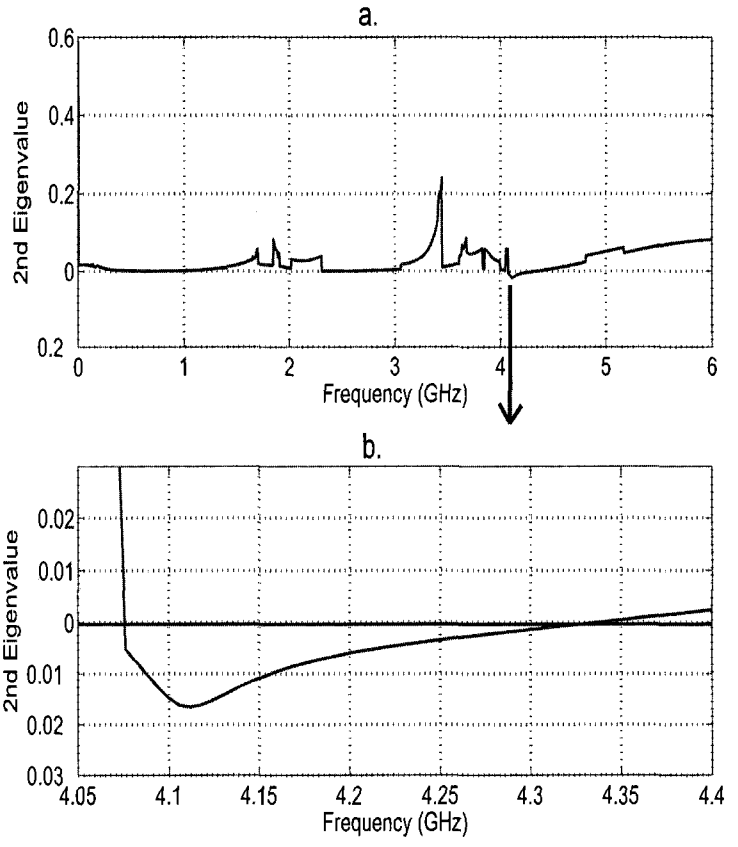

Fig. 5.14 **a.** The second eigenvalue of the real part of $\mathbf{Y}(j\omega)$ **b.** Enlarged view of the passivity violation

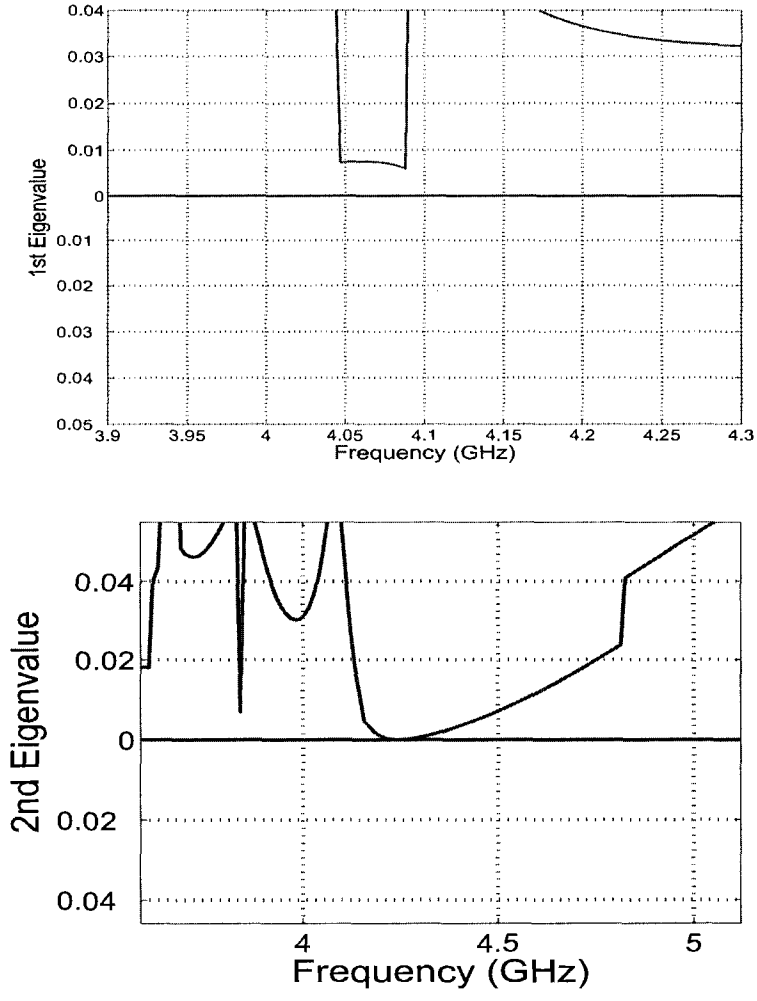

Fig. 5.15 Eigenvalues of the real part of $\mathbf{Y}(j\omega)$ after compensation (enlarged view) - Example 3

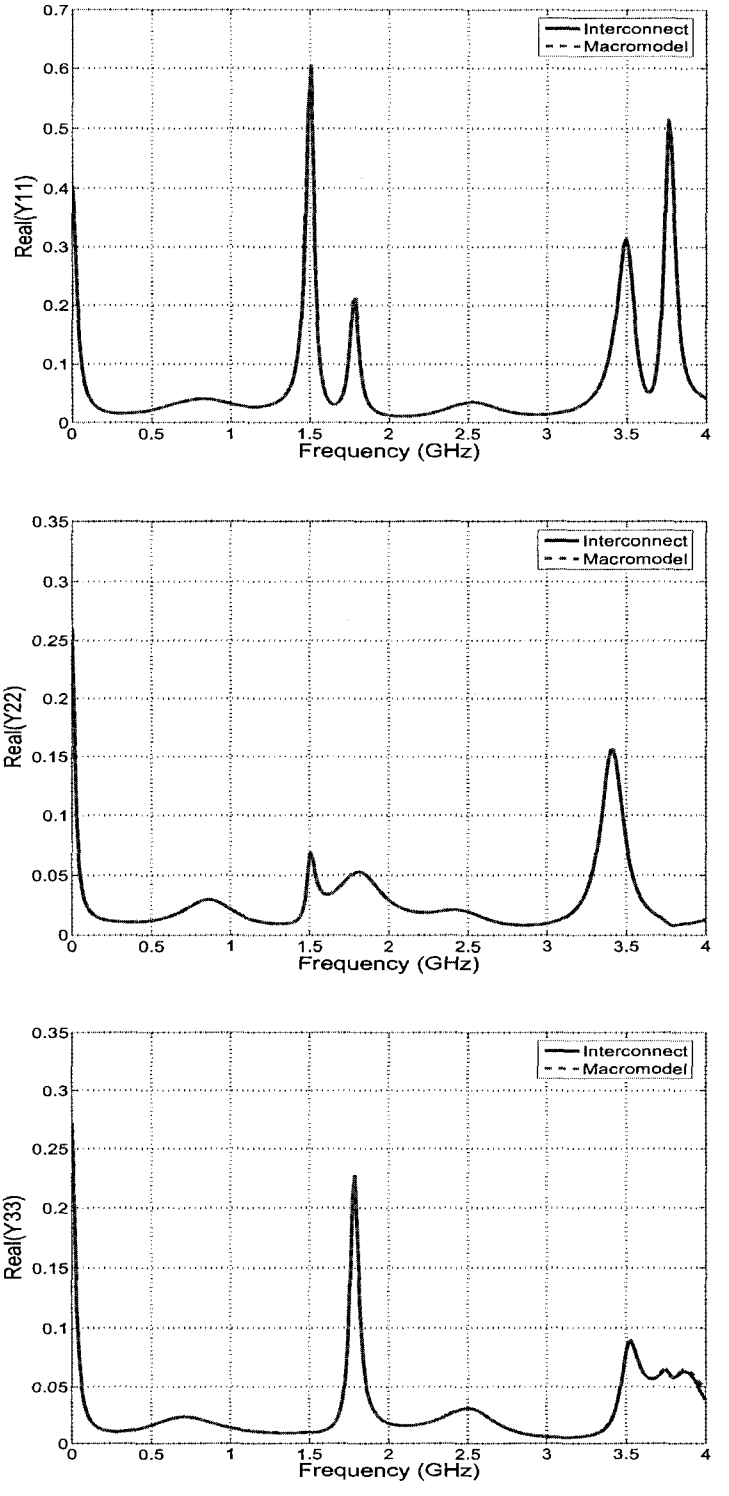

Fig. 5.16 Y-parameters (real part) comparison after compensation - Example 3

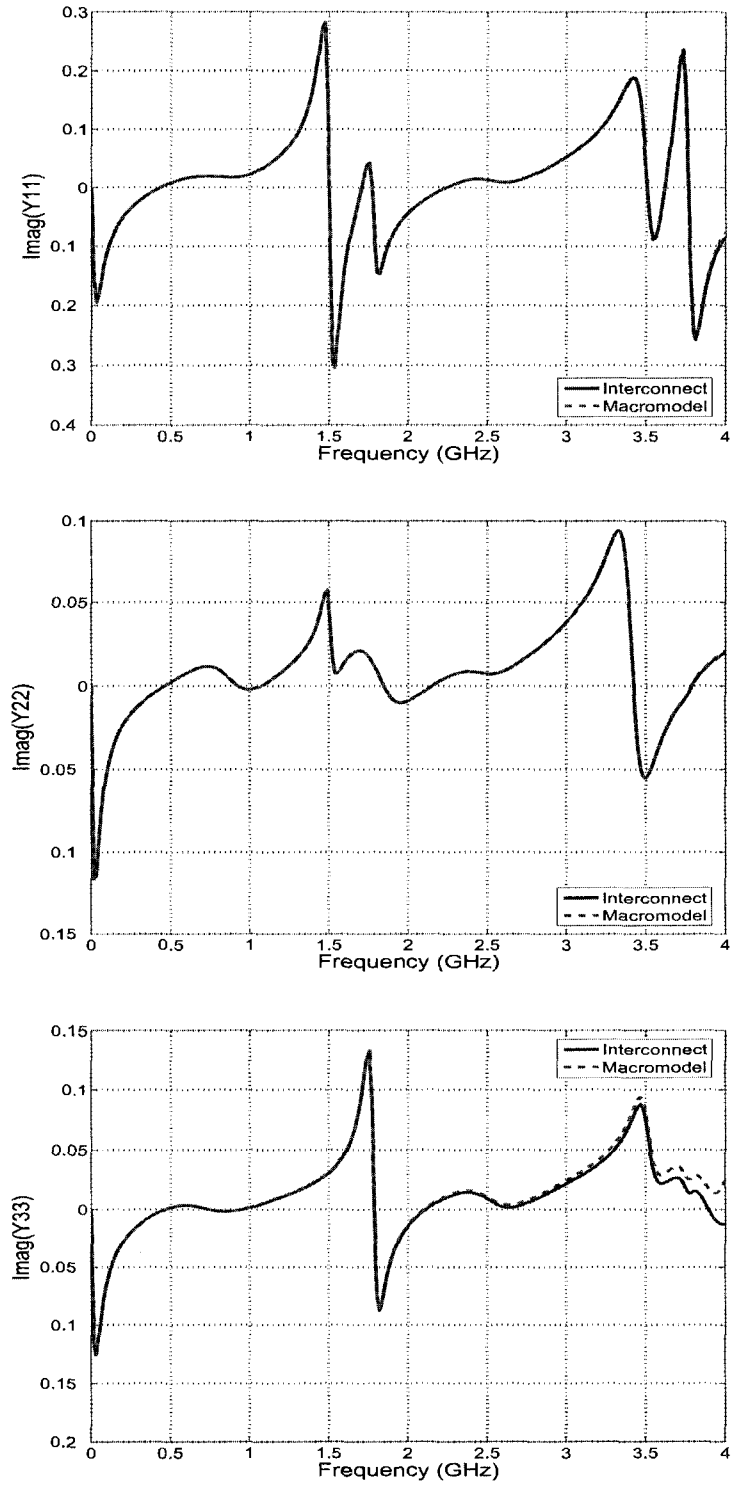

Fig. 5.17 Y-parameters (imaginary part) comparison after compensation - Example 3

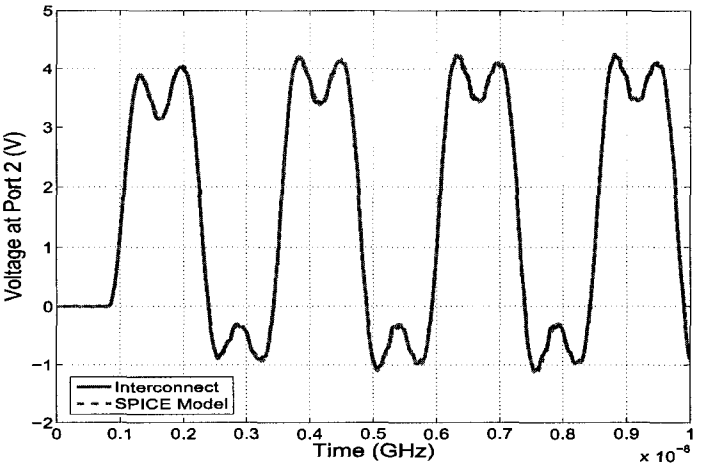

Fig. 5.18 Transient response comparison: voltage at Port 2 - Example 3

5.3 Compensation-Introduced Inaccuracy and Solution

We have seen that there is an observable difference in the imaginary part of Y_{33} between the compensated macromodel and the interconnect in the region close to the passivity violation (happening out of band). The accuracy of the macromodel decreases due to the compensation process. This could happen especially when the passivity violation takes place beyond and close to our highest frequency of interest. The reason lies in that Vector Fitting only guarantees the in-band accuracy, it does not have control over the out-band response of the macromodel. Consequently, given that sufficient in-band information is provided and a fine Vector Fitting is applied, it is extremely occasional to detect passivity violation in band (In case of in-band passivity violation, it should be extremely small so that some slight compensation is sufficient to offset it and the frequency response will not be affected much.). In addition, remember that we have ensured the macromodel to be asymptotically passive at $\omega = \infty$ by adding constrains to the coupling constants ($\mathbf{D}$), so the most possible location of passivity violation is in the vicinity of the highest frequency of interest right out of band, like Example 3 (Example 2 also shows an out-band violation, but the violation is so small that it has very little affect on the accuracy). In contrast to in-band passivity violations, out-band violations are usually large and require more efforts to compensate. The degradation of accuracy due to the compensation process is consequently

large. Fortunately, since only the poles in the vicinity of passivity violation are selected for compensation, the resulting inaccuracy is efficiently confined to the vicinity of the highest frequency of interest, like the imaginary part of Y_{33} in Fig. 5.17. Based on this discussion, a straightforward solution to the problem is to give some appropriate frequency margin to our frequency range of interest. For an illustration, we return to Example 3. Assume one cares about the accuracy of the macromodel up to 4 GHz. Instead of simulating the original network to 4 GHz (as what was done in Example 3), we simulate it to 5 GHz. The passivity violation happens right out of band. Fig. 5.19 shows the negative eigenvalues. After a proper compensation, the responses of the macromodel are compared with those of the original network (Fig. 5.21 and Fig. 5.22). Good matches are obtained except for the imaginary part of Y_{22}, as a small mismatch is detected around 5 GHz. Since our highest frequency of interest is 4 GHz, from Fig. 5.22 the 1 GHz margin efficiently offsets the accuracy degradation.

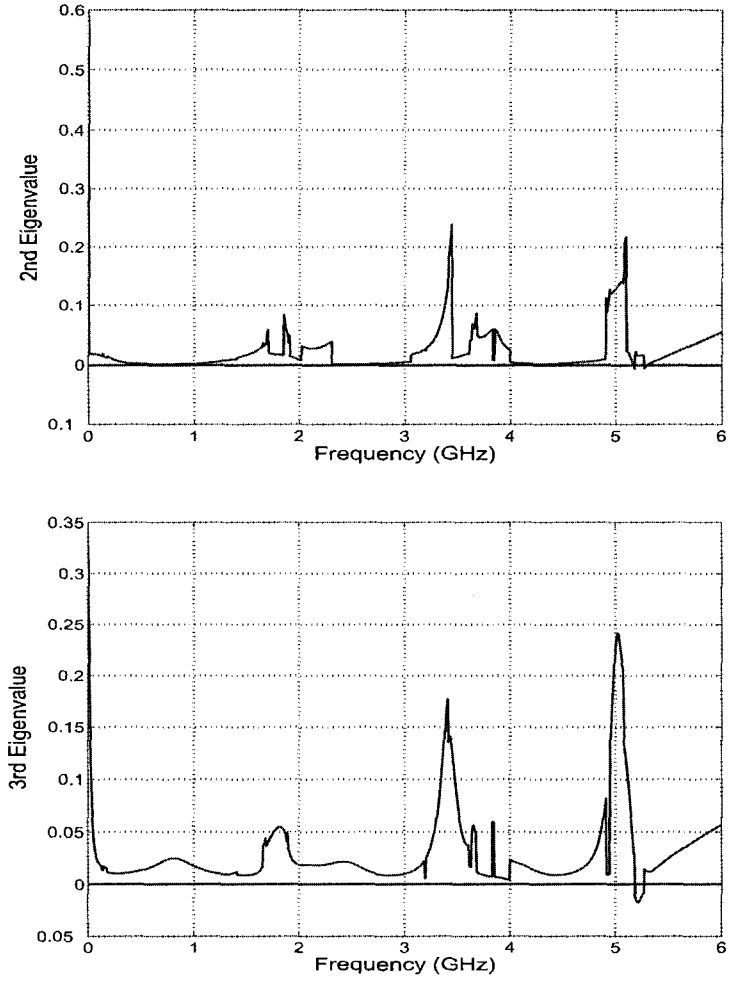

Fig. 5.19 The second and third eigenvalues of the real part of $\mathbf{Y}(j\omega)$ - Example 3 with frequency margin

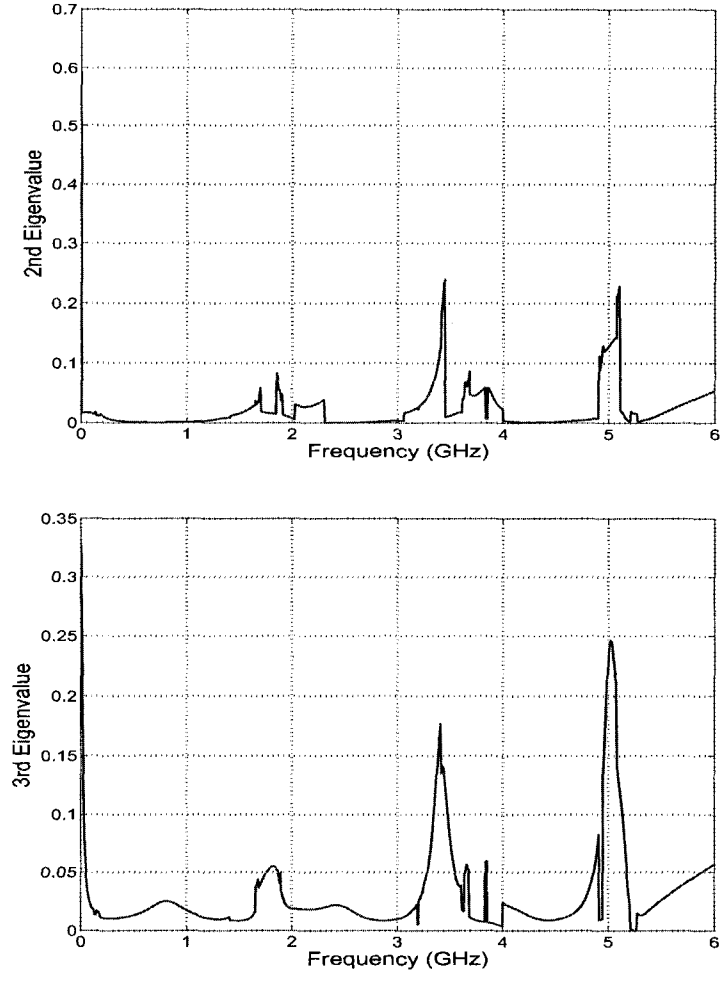

Fig. 5.20 The second and third eigenvalues of the real part of $\mathbf{Y}(j\omega)$ after compensation - Example 3 with frequency margin

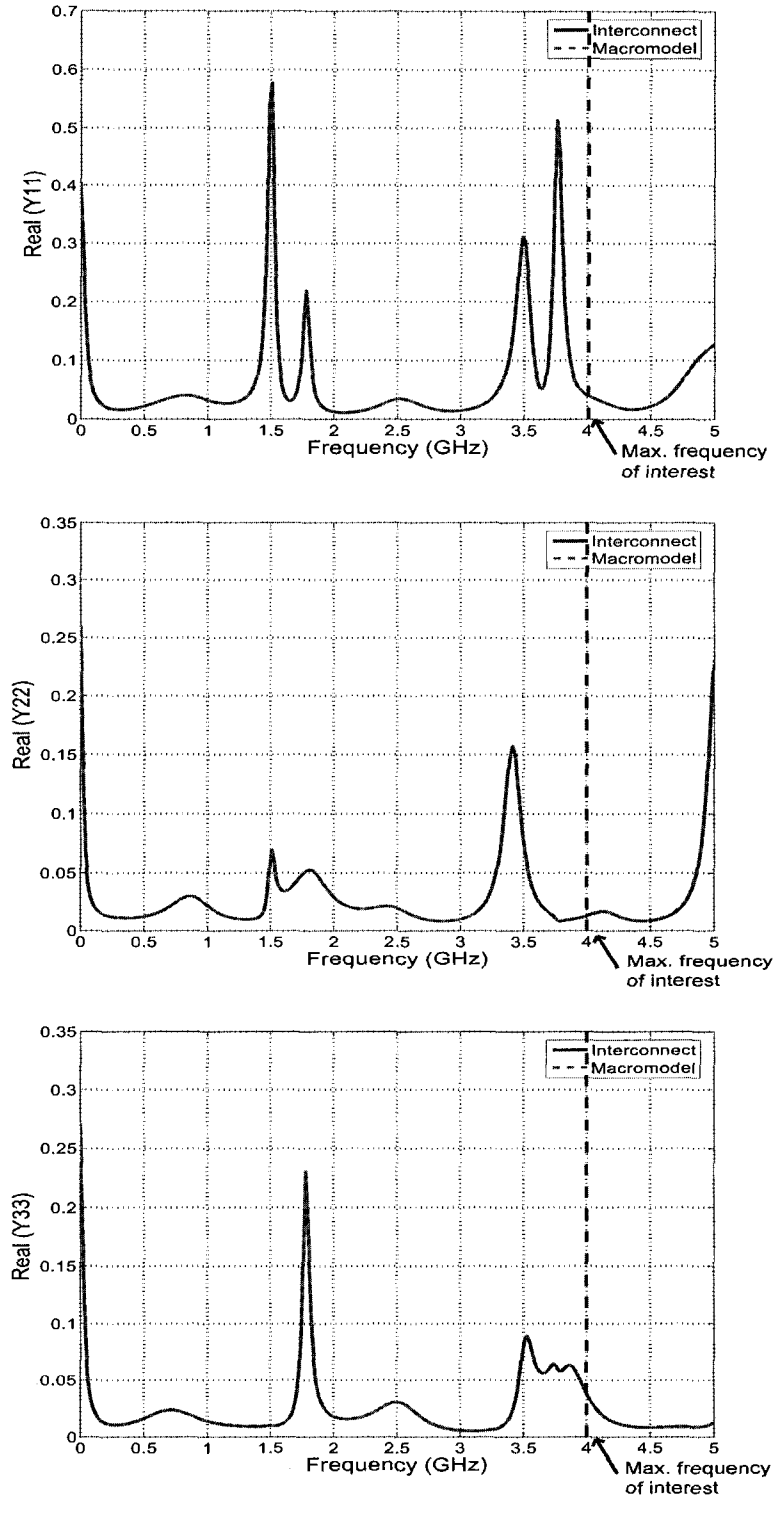

Fig. 5.21 Y-parameters (real part) comparison after compensation - Example 3 with frequency margin

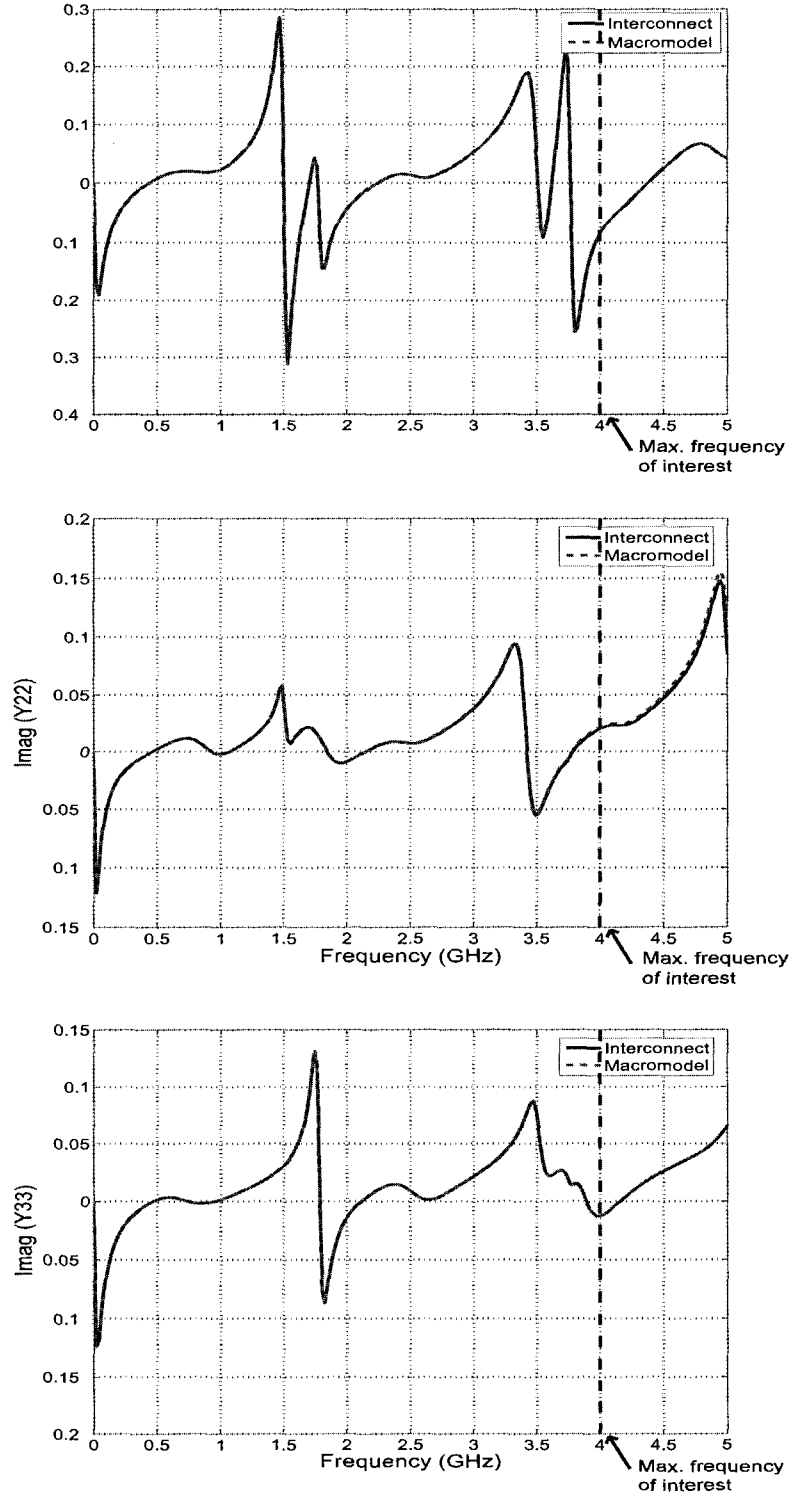

Fig. 5.22 Y-parameters (imaginary part) comparison after compensation - Example 3 with frequency margin

Chapter 6

Conclusions and Future Work

6.1 Conclusion

This thesis reviews some typical techniques developed for incorperating the frequency-domain measured data into time-domain simulators. Two main classes of methods have been discussed: convolution-based methods and the ones based on macromodels. The convolution-based methods are usually not CPU-efficient as the convolutions have to be done numerically. The macromodel-based approaches involve rational function approximation and therefore suffer from ill-condition when a high-order approximation is required. In contrast to these methods, Vector Fitting algorithm efficiently avoids ill-condition by applying iterative solution. This feature allows it to fit more complicated curves in high-order approximations.

Several methods for checking the passivity of the resulting macromodel are presented in Chapter 4. The method used in this thesis is based on evaluating the eigenvalues of the Hamiltonian matrix. The advantage lies in that this method provides a direct link between pure imaginary eigenvalues of the Hamiltonian and the regions of passivity violation. This helps us precisely locate the regions. Based on this information, the algorithm discussed in Chapter 5 compensates the regions of passivity violation along a positive direction without introducing additional passivity violations. The algorithm perturbs the real part of the residues for the diagonal entries of $\mathbf{Y}(j\omega)$ with the poles and coupling constants unchanged. By applying a linear optimization the perturbed values offset the maximum violation and keep the change in response at the minimum level. A potential problem of accuracy degra-

dation associated with the algorithm is illustrated by Example 3. It is pointed out that adding some initial frequency margin to Vector-fitting can effectively fix the problem.

6.2 Future Work

1. The compensation process is based on perturbing the residues of the diagonal entries of $\mathbf{Y}(j\omega)$, and the relative change in response is a measure to the impact of the compensation. In case the frequency response is small (close to zero) throughout the bandwidth, a minor compensation may cause relatively large change or even offset the original response. Future work is required to relate the magnitude of the change to the magnitude of the frequency response. Thus, the degradation of accuracy can be kept *relatively* small.

2. Efforts will be made to develop a passivity-guaranteed approximation algorithm. The focus may be moved from post-macromodeling compensation to pre-macromodeling manipulation, which avoids passivity check and compensation and therefore saves CPU cost.

3. Interconnects containing large delay lines usually require comparatively high-order macromodels to represent. The resulting complexity of macromodels leads to intensive CPU efforts in transient simulation. A delay extraction operation is required to reduce the delay before the rational function approximation [59].

4. The approximation methods discussed in this thesis are only for linear network macromodel construction. In contrast to linear networks, nonlinear networks are characterized in the time domain, and nonlinear macromodeling techniques need to be developed to address this problem.

Appendix A

Conversion of Macromodels to Equivalent Circuits

The conversion of macromodels in their state-space form to equivalent circuits can be easily achieved. Consider (4.5) and (4.6)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{v}(t) \tag{A.1}$$

$$\mathbf{I}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{v}(t). \tag{A.2}$$

For a better illustration, let us assume we have a simple macromodel with two ports and two states. Hence, the above two differential equations can be rewritten as

$$\dot{x}_1 = a_{11}x_1 + a_{12}x_2 + b_{11}v_1 + b_{12}v_2 \tag{A.3}$$

$$\dot{x}_2 = a_{21}x_1 + a_{22}x_2 + b_{21}v_1 + b_{22}v_2 \tag{A.4}$$

$$\dot{I}_1 = c_{11}x_1 + c_{12}x_2 + d_{11}v_1 + d_{12}v_2 \tag{A.5}$$

$$\dot{I}_2 = c_{21}x_1 + c_{22}x_2 + d_{21}v_1 + d_{22}v_2, \tag{A.6}$$

where I_1, I_2 and v_1, v_2 denote the port currents and voltages respectively. x_1, x_2 stand for the state variables. (A.3-A.6) can be translated into an equivalent network shown in Fig. A.1 [60]. The state variables can be represented by capacitor or node voltages v_{n1}, v_{n2}. In another word, each state variable is represented by one independent node like Fig.

A.1(c)(d). Next, the equivalent circuits for the equations of the output currents are given by Fig. A.1(a)(b). Terms such as $a_{11}, a_{12}, a_{21}, a_{22}$ in (A.3) and (A.4) are represented by voltage controlled current sources controlled by the node voltages. It is always true that the number of the subnetworks in Fig. A.1(a)(b) should be equal to the number of ports and the number of the subnetworks in Fig. A.1(c)(d) should be equal to the number of states. The resulting equivalent circuits are easily implemented for transient simulation.

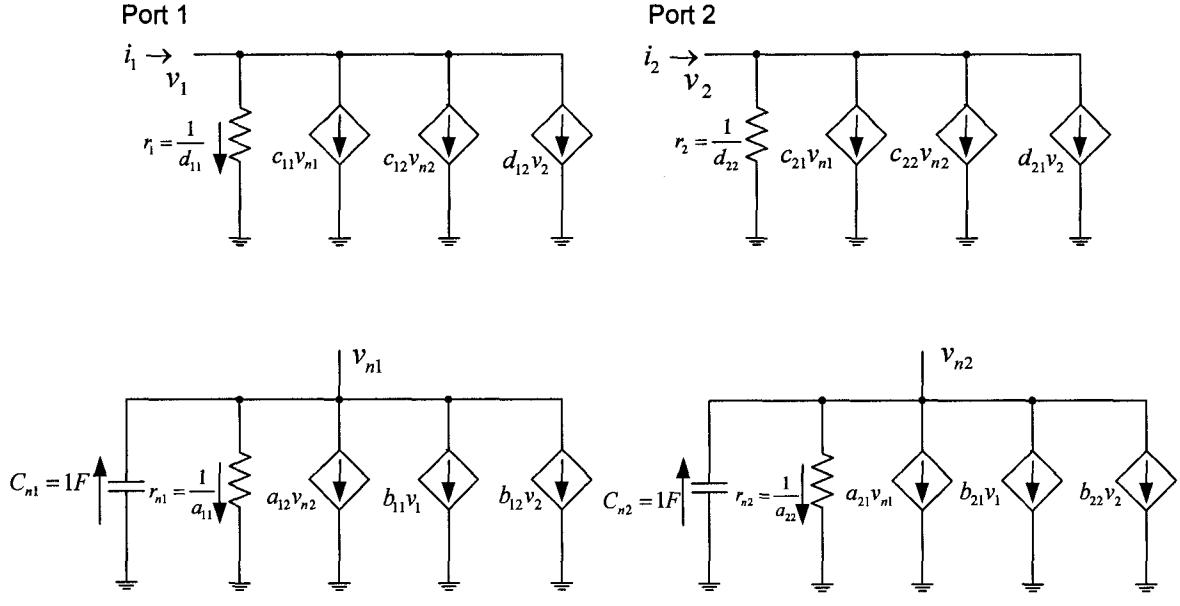

Fig. A.1 Equivalent circuits of the state-space model

References

[1] W. T. Beyene and J. E. Schutt-Aine, "Efficient transient simulation of high-speed interconnects characterized by sampled data," *IEEE Trans. Comp.,Packag. Manufact. Technol. A*, vol. 21, pp. 105–114, Feb. 1998.

[2] M. Elzinga, K. L. Virga, L. Zhao, and J. L. Prince, "Pole-residue formulation for transient simulation of high-frequncy interconnects using householder ls curve-fitting techniques," *IEEE Trans. Adv. Packag.*, vol. 23, pp. 142–147, May 2000.

[3] W. T. Beyene and J. E. Schutt-Aine, "Interconnect simulation using order reduction and scattering parameters," *IEEE Electronic Components and Technology Conference*, pp. 627–631, May 1998.

[4] R. Achar and M. S. Nakhla, "Simulation of high-speed interconnects," *Proceedings IEEE*, vol. 89, pp. 693–728, May 2001.

[5] H. B. Bakoglu, *Circuit, Interconnections and Packaging for VLSI.* Reading, MA: Addison-Wesley, 1990.

[6] M. S. Nakhla and R. Achar, *High-Speed Circuit and Interconnect Analysis.* multimedia Learning course, OT: Omniz Global Knowledge Corporation (www.omniz.com), 2001.

[7] D. Saraswat, R. Achar, and M. S. Nakhla, "A fast algorithm and practical considerations for passive macromodeling of measured/simulated data," *IEEE Trans. Adv. Packag.*, vol. 27, Feb. 2004.

[8] W. W. M. Dai, "Special issue on modeling and simulation of high-speed and high-density interconnects," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 857–982, Nov. 1992.

[9] M. S. Nakhla and A. Ushida, "Special issue on modeling and simulation of high-speed interconnects," *IEEE Trans. Circuits Syst.*, vol. 47, pp. 239–305, Apr. 2000.

[10] A. Deutsch, "Electrical charactersitics of interconnections for high performance systems," *Proceedings IEEE*, vol. 86, pp. 315–355, Feb. 1998.

[11] R.Goyal, "Managing signal integrity," *IEEE Spectr.*, pp. 54–62, Mar. 1994.

[12] J. B. Faria, *Multiconductor Transmission Line Structures.* New York: Wiley, 1993.

[13] C. Paul, *Analysis of Multiconductor Transmission Lines.* New York: Wiley, 1994.

[14] K. C. Gupta and R. Grag, *Microstrip Lines and Slotlines.* Boston MA: Artech House, 1996.

[15] A. E. Ruehli, "Equivalent circuit models for three dimentional multiconductor systems," *IEEE Trans. Microwave Theory Tech.*, vol. 22, pp. 216–224, Mar. 1974.

[16] P. K. Wolff and A. E. Ruehli, "Inductance computations for complex three dimensional geometries," *IEEE Trans. Circuits Syst.*, pp. 16–19, 1981.

[17] J. Cullum, A. E. Ruehli, and T. Zhang, "A method of reduced-order modeling and simulation of large interconnect circuits and its application to peec models with retardation," *IEEE Trans. Circuits Syst.*, pp. 261–273, Apr. 2000.

[18] A. E. Ruehli and H. Heeb, "Circuit models for three dimentional geometries including dielectrics," *IEEE Trans. Microwave Theory Tech.*, pp. 1507–1516, Mar. 1992.

[19] A. E. Ruehli and P. A. Brennan, "Equivalent capacitance calculations for three dimensional multiconductor systems," *IEEE Trans. Microwave Theory Tech.*, pp. 76–82, Feb. 1973.

[20] B. Young, *Digital Signal Integrity: Modeling and Simulation with Interconnects and Packages.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[21] M. Nakhla and R. Achar, *Interconnect modeling and simulation,in The VLSI Handbook*, ch. XVII. Boca Raton, FL: CRC Press, 2000.

[22] D. Saraswat, "Passive macromodeling of linear subnetworks characterized by measured/simulated data," masters thesis, Carleton University, Ottawa ON, Canada, April 2003.

[23] N. Nakhla, A. Dounavis, R. Achar, and M. Nakhla, "Passive macromodeling and sensitivity analysis of multiconductor transmission lines," *IEEE International North Eastern Symposium on Circuits and Systems*, pp. 33–36, June 2004.

[24] C. P. Coelho, J. Phillips, and L. M. Silveira, "A convex programming approach for generating guaranteed passive approximations to tabulated frequency-data," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, Feb. 2004.

[25] J. Kim and M. Swaminathan, "Modeling of multilayered power distribution planes using transmission matrix method," *IEEE Trans. Adv. Packag.*, vol. 25, May 2002.

[26] N. Orhanovic, P. Wang, and V. K. Tripathi, "Generalized method of characteristics for time domain simulation of multiconductor lossy transmission lines analysis of lossy coupled transmission lines," *Proc. IEEE Symp. on Circuits Syst.*, May 1990.

[27] F. Y. Chang, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *IEEE Trans. Microwave Theory Tech.*, vol. 37, pp. 2028–2038, Dec. 1989.

[28] D. Saraswat, *Global Compact Passive Macromodeling Algorithms for High-Speed Circuits.* Ph.d. thesis, Carleton University, Ottawa ON, Canada, 2006.

[29] M. S. Nakhla and R. Achar, *Multimedia Book Series on Signal Integrity.* ON: Omniz Global Knowledge Corporation (www.omniz.com), 2002.

[30] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.

[31] E. Chiprout and M. Nakhla, *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis.* Boston, MA: Kluwer Academic Publishers, 1993.

[32] E. Chiprout and M. S. Nakhla, "Analysis of interconnect networks using complex frequency hopping(CFH)," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 186–200, Feb. 1995.

[33] B. Moore, "Principal component analysis in linear systems: Controllability, observability and model reduction," *IEEE Trans. Automatic Control*, pp. 17–32, Feb. 1981.

[34] M. Ma and R. Khazaka, "Efficient projection based macromodel for interconnect networks," *Proc. 9th IEEE Workshop on Signal Propagation on Interconnects, Germany*, pp. 181–184, May 2005.

[35] S. Lin and E. S. Kuh, "Transient simulation of lossy interconnects based on the recursive convolution formulation," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 879–892, Nov. 1992.

[36] L. M. Silveira, I. M. Elfadel, J. K. White, M. Chilukura, and K. S. Kundert, "Efficient frequency-domain modeling and circuit simulation of transmission lines," *IEEE Trans. Comp.,Packag. Manufact. Technol. A*, vol. 17, pp. 505–513, Nov. 1994.

[37] T. McKelvey, H. Akçay, and L. Ljung, "Subspace-based multivariable system identification from frequency response data," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 960–979, July 1996.

[38] D. Saraswat, R. Achar, and M. S. Nakhla, "Global passivity enforcement algorithm for macromodels of interconnect subnetworks characterized by tabulated data," *IEEE Trans. Very Large Scale Integration Systems*, vol. 13, no. 7, pp. 819–832.

[39] J. E. Schutt-Aine and R. Mittra, "Scattering parameter transient analysis of transmission lines loaded with nonlinear terminations," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 529–536, Mar. 1998.

[40] A. R. Djordjevic, T. K. Sarkar, and R. F. Harrington, "Analysis of lossy transmission lines with arbitrary nonlinear terminal networks," *IEEE Trans. Microwave Theory Tech.*, vol. 34, pp. 660–666, June 1986.

[41] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. power delivery*, vol. 14, pp. 1052–1061, July 1999.

[42] D. Winklestein, M. B. Steer, and R. Pomerleau, "Simulation of arbitrary transmission line networks with nonlinear terminations," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 418–422, Apr. 1991.

[43] C. P. Coelho, J. R. Phillips, and L. M. Silveira, "Robust rational function approximation algorithm for model generation," *Proc. 36th Design Automation Conference, New Orleans*, vol. 23, no. 2, pp. 207–212, 1999.

[44] Y. Saad, *Iterative Methods for Sparse Linear Systems.* Pws Publishing Co., 1996.

[45] L. N. Trefethen and D. Bau, *Numerical Linear Algebra.* Philadelphia PA: SIAM, 1999.

[46] R. Archar, P. Gunupudi, M. Nakhla, and E. Chiprout, "Passive interconnect reduction algorithm for distributed/measured networks," *TCAS-II*, vol. 47, pp. 287–301, Apr. 2000.

[47] L. Weinberg, *Network Analysis and Synthesis.* New York: McGraw-Hill Book Company Inc., 1962.

[48] E. S. Kuh and R. Rohrer, *Theory of Active Linear Networks.* San Fransisco: Holden-day Inc., 1967.

[49] M. E. V. Valkenberg, *Introduction to Modern Network Synthesis.* New York: John Wiley and Sons Inc., 1960.

[50] C. T. Chen, *Linear System Theory and Design.* New York: Oxford University Press, 1999.

[51] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory.* Philadelphia PA: SIAM, 1994.

[52] D. Saraswat, R. Archar, and M. Nakhla, "Passive macromodels of microwave subnetworks characterized by measured/simulated data," *Proc. 7th IEEE Workshop Signal Propagation Interconnects, Siena, Italy*, pp. 999–1002, June 2003.

[53] D. Saraswat, R. Archar, and M. Nakhla, "On passivity check and compensation of macro-models from tabulated data," *Proc. Int. Microwave Symp. 2003 Dig., Philadelphia, PA*, pp. 25–28, May 2003.

[54] S. Grivet-Talocia, "Passivity enforcement via perturbation of hamiltonian matrices," *IEEE Trans. Circuits Syst.*, vol. 51, pp. 1755–1769, Sept. 2004.

[55] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* Ph.d. dissertation, Dept. of Control and Dynamical Systems, California Institudte of Technology, Pasadena, 2000.

[56] G. W. Stewart and J. Sun, *Matrix Perturbation Theory.* Boston, MA: Academic Press, 1990.

[57] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis.* Cambridge University Press, 1991.

[58] E. A. Guillemin, *Synthesis of passive networks.* New York: John Wiley & Sons, 1957.

[59] A. Dounavis, N. Nakhla, R. Achar, and M. Nakhla, "Delay extraction and passive macromodeling of lossy coupled transmission lines," *Proc. IEEE conf. on Electrical Performance of Electronic Packaging*, pp. 251–254, Oct. 2003.

[60] R. Archar, *Model-Reduction Techniques for High-Speed interconnect Analysis.* Ph.d. thesis, Carleton University, Ottawa ON, Canada, September 1998.