BROADBAND MACROMODELING VIA A FAST IMPLEMENTATION OF VECTOR FITTING WITH PASSIVITY ENFORCEMENT

BY

PATRICK KUANLYE GOH

B.S., University of Illinois at Urbana-Champaign, 2007

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Adviser:

Professor José E. Schutt-Ainé

ABSTRACT

As operating frequencies and signal speeds continue to increase in modern devices, the effects of packages and interconnects on the overall signal integrity become increasingly important. The complex electromagnetic behaviors of these often complicated structures must be characterized in order to take their effects into account. Broadband macromodeling deals with the generation of network models of these devices in order to accurately predict their behaviors in circuit simulators. This often involves the generation of passive rational function representations of the system from the measured port responses.

In this thesis, we will employ the vector fitting algorithm to generate a rational function representation of the system along with its state space model. Various issues on the subject will be discussed, including the recently developed fast fitting method for multiport devices. Passivity of the model, which is one of the most prominent issues on the subject, will be addressed. A robust algorithm, via residue perturbation, to enforce passivity in nonpassive models will be presented. Finally, numerical results will be presented to demonstrate the performance of the overall process.

ACKNOWLEDGMENTS

"If I have seen a little further it is by standing on the shoulders of giants."

Sir Isaac Newton

I would like to take this opportunity to express my sincere gratitude to my advisor, Prof. José E. Schutt-Ainé, for his wisdom and guidance throughout this entire research. His patience and understanding have kept me going through hard and easy times.

I would also like to thank Dr. Yidnek Mekonnen for his invaluable help in the short time that I knew him when I first started this research, and also Dr. Jilin Tan at Cadence Design Systems for his helpful support and suggestions on the project.

Additionally, I am grateful to my group members – Srivatsan Jayaraman, Dmitri Klokotov and Pavle Milosevic – for their help and valuable contributions to various parts of my research and graduate life as a whole. I sincerely believe that I am made a better person because of the people around me.

Also, I would of course like to thank my parents and my sister for all their love, support and understanding as I pursued my dream. I might be a long way from where I started, but I certainly have not forgotten my way home.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

With the advancement of recent technology, there is an ongoing effort to achieve faster and smaller electronic systems. Developers continue to push the envelope in terms of operating frequencies and design densities. However, with the increase in signal speed, signal integrity issues such as crosstalk, dispersion, attenuation, reflection, delay and distortion become more significant and must be duly accounted for in a proper design. This involves taking into account the complex electromagnetic behaviors of all parts of the systems and incorporating them into the design process.

One of the most prominent researches on the subject is about the modeling of packages and interconnects such that their frequency dependent effects can be accurately simulated and integrated in the design considerations. This is important because, with the increase in density of recent designs, interconnects can take up a large portion of the final product and packages can significantly influence the input and output behavior of the components. Unfortunately, creating an accurate model for these components can be rather challenging. Most packages and interconnect structures have complicated three-dimensional configurations, which often have no closed form solutions. Also, with the increase in operating speed, lumped circuit elements can no longer provide a good approximation of the complex behavior of

these components. Thus, more often than not, the only way to obtain an accurate characterization of the system is through physical measurements or full-wave electromagnetic simulations, which often results in a frequency domain characterization in the form of a tabulated dataset. The challenge is then in generating a precise circuit model or means to perform simulations on the network from the obtained tabulated data.

While it is possible to utilize the obtained data directly in circuit simulations by combining the frequency and time domain characterizations – often through the use of the inverse fast Fourier transform – such a process can be prohibitively slow as it requires a full numerical convolution between the impulse response of the system and the input at each time step. As a result, most developers turn to the use of a macromodeling technique whereby the network can be simulated faster and more efficiently.

Macromodeling involves partitioning the network into smaller subnetworks and generating a rational function representation for each subnetwork in a pole-zero (or pole-residue) model. Once these individual models are found, they can be recombined to form the overall network representation, and due to the pole-zero form of the model, the time domain simulations can be done in a recursive fashion. This results in a much faster simulation than the conventional convolution process.

However, there are a few important aspects of the model that must be ensured in order to avoid complications in the time domain simulations. The two most prominent properties are that of stability and passivity, and each of these will be addressed in the subsequent chapters.

1.2 Organization

The materials in this thesis are organized as follows. First the vector fitting method, which is used to generate a rational function representation of a system, will be presented in Chapter 2. This includes the modifications needed to handle complex poles and to fit multiple functions using the same set of poles. In addition, the recently developed fast fitting method will be explored. Stability considerations and starting poles selection methods will also be detailed.

Next, a passivity enforcement scheme will be formulated in Chapter 3. The mathematical condition for passivity and a robust passivity assessment scheme will be discussed. For nonpassive models, a passivity enforcement process involving the perturbation of residues will be presented.

In Chapter 4, the time domain simulation method utilizing the recursive convolution algorithm will be described.

Numerical results will be shown in Chapter 5. Starting from measured S-parameters of packages and interconnects in a tabulated data form, a rational function representation will be obtained by utilizing the vector fitting method. The passivity of the generated models will be assessed and enforced in the case of nonpassive models. Finally, the time domain responses will be obtained through the use of recursive convolutions with the generated passive models.

Chapter 6 will contain the conclusion along with a discussion of possible future work on the subject.

CHAPTER 2

THE VECTOR FITTING METHODOLOGY

2.1 Overview

In this section the purpose of using vector fitting will be introduced. First consider a frequency dependent function $f(s)$ where $s = j\omega$ and ω is the angular frequency, which defines the frequency dependent effects of a system. In order to generate an accurate model of the system, it is often necessary – especially in modern high speed devices – to take this frequency dependent effect into account. For this purpose, the function $f(s)$ in its tabulated form can be used, but this requires a full numerical convolution which is computationally both slow and inefficient. A faster and more desirable method is possible if a *rational function* which approximates the original function $f(s)$ is used instead, in which case the convolution can then be done recursively. Thus, finding an accurate rational function representation of any arbitrary function becomes a paramount step in the system identification process, a step in which vector fitting [1] has emerged as the method of choice.

2.2 Rational Function Approximation

Now consider the same function $f(s)$, but written as a rational function of order N in the form

$$f(s) = \sum_{n=1}^{N} \frac{c_n}{s - a_n} + d + sh \tag{2.1}$$

where the unknowns c_n, a_n, d and h are the residues, poles, constant and linear terms, respectively. If we assume that the order N (often referred to as the *order of approximation*) is fixed beforehand, the process of finding a rational function that approximates $f(s)$ in the form of Equation (2.1) then amounts to solving for all the unknown coefficients c_n, a_n, d and h. It is obvious, however, that Equation (2.1) is nonlinear in terms of the unknowns as a_n appears in the denominator. Vector fitting solves this nonlinear problem by decomposing it into a set of two linear problems, as we will see next.

In the first stage, the poles a_n of the system are identified from the frequency sampled data using a set of initial *guessed* starting poles. This is presented in Section 2.3. In the second stage, the residues c_n are then determined based on the frequency sampled data and the poles a_n determined in stage one. This is presented in Section 2.4. Note that in both stages, vector fitting linearizes the problem by fixing the denominator of a similar nonlinear problem.

In the following discussions, we will first introduce the vector fitting method for cases with purely real poles. The modifications needed to handle complex poles will be presented in Section 2.5.

2.3 Stage One: Pole Identification

In this stage, the poles a_n of the system are solved for. The key in solving for the poles lies in the introduction of an *unknown* function $\sigma(s)$ which is defined in

its rational form as

$$\sigma(s) = \sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \tag{2.2}$$

Notice that the ambiguity in the solution for $\sigma(s)$ is removed by forcing it to approach unity at very high frequencies.

Next we assume that both $\sigma(s)$ and the product of $\sigma(s)$ and $f(s)$ (i.e., $\sigma(s)f(s)$) can be approximated by rational functions using the *same set of poles* (in this case $\tilde{a}_n$). Thus we have the augmented problem:

$$\begin{bmatrix} \sigma(s)f(s) \\ \sigma(s) \end{bmatrix} \approx \begin{bmatrix} \sum_{n=1}^{N} \frac{c_n}{s-\tilde{a}_n} + d + sh \\ \sum_{n=1}^{N} \frac{\tilde{c}_n}{s-\tilde{a}_n} + 1 \end{bmatrix} \tag{2.3}$$

Now multiplying the second row in Equation (2.3) by $f(s)$ (the tabulated data) and equating it to the first row gives

$$\left(\sum_{n=1}^{N} \frac{c_n}{s - \tilde{a}_n} + d + sh \right) \approx \left(\sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \right) f(s) \tag{2.4}$$

Expanding the right side of Equation (2.4) and rearranging the terms gives

$$\left(\sum_{n=1}^{N} \frac{c_n}{s - \tilde{a}_n} + d + sh \right) \approx \left(\sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \tilde{a}_n} \right) f(s) + f(s) \tag{2.5}$$

$$\left(\sum_{n=1}^{N} \frac{c_n}{s - \tilde{a}_n} + d + sh \right) - \left(\sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \tilde{a}_n} \right) f(s) \approx f(s) \tag{2.6}$$

Examining Equation (2.6) reveals that if the poles are *fixed beforehand*, then Equation (2.6) is linear in terms of the unknowns c_n, d, h and $\tilde{c}_n$. Since $f(s)$ is often obtained from a set of tabulated data, and the number of data points collected normally well exceeds the order of approximation N, writing Equation (2.6) for each frequency sample point results in an overdetermined set of equations

in the form of

$$Ax = b \tag{2.7}$$

To see this, we rewrite Equation (2.6) for each frequency sample point in matrix form. This gives

$$\begin{bmatrix} \frac{1}{s_1-\tilde{a}_1} & \cdots & \frac{1}{s_1-\tilde{a}_N} & 1 & s_1 & \frac{-f(s_1)}{s_1-\tilde{a}_1} & \cdots & \frac{-f(s_1)}{s_1-\tilde{a}_N} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{s_k-\tilde{a}_1} & \cdots & \frac{1}{s_k-\tilde{a}_N} & 1 & s_k & \frac{-f(s_k)}{s_k-\tilde{a}_1} & \cdots & \frac{-f(s_1)}{s_k-\tilde{a}_N} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_N \\ d \\ h \\ \tilde{c}_1 \\ \vdots \\ \tilde{c}_N \end{bmatrix} = \begin{bmatrix} f(s_1) \\ \vdots \\ \\ \\ \vdots \\ f(s_k) \end{bmatrix} \tag{2.8}$$

where k is the number of frequency sample points. This overdetermined set of equations can then be solved using any of the standard least squares methods for the unknown solution vector x that contains the residues.

From Equation (2.8), we see that solving for the solution vector x gives three sets of solutions:

1. The residues of $\sigma(s)f(s)$: $c_1 - c_N$
2. The constant and linear terms of $\sigma(s)f(s)$: d and h
3. The residues of $\sigma(s)$: $\tilde{c}_1 - \tilde{c}_N$

Of the above, only solution set 3 ($\tilde{c}_1 - \tilde{c}_N$) is needed at this point. The other two are discarded as a more accurate set can be calculated later in the residue identification stage (Section 2.4). At first glance, it might not be clear how solving

for the residues of $\sigma(s)$ can lead to the solution of the poles a_n of the system under consideration (i.e., $f(s)$). However, a closer examination of Equation (2.4) reveals how this is possible, as is presented next.

Consider Equation (2.4) which is repeated here for convenience:

$$\left(\sum_{n=1}^{N}\frac{c_n}{s-\tilde{a}_n}+d+sh\right)\approx\left(\sum_{n=1}^{N}\frac{\tilde{c}_n}{s-\tilde{a}_n}+1\right)f(s) \tag{2.9}$$

Solving for $f(s)$ gives

$$f(s)\approx\frac{\sum\limits_{n=1}^{N}\frac{c_n}{s-\tilde{a}_n}+d+sh}{\sum\limits_{n=1}^{N}\frac{\tilde{c}_n}{s-\tilde{a}_n}+1} \tag{2.10}$$

Now each sum of partial fractions can be rewritten as a fraction to obtain

$$f(s)\approx\frac{\frac{\prod\limits_{n=1}^{N+1}(s-z_n)}{\prod\limits_{n=1}^{N}(s-\tilde{a}_n)}}{\frac{\prod\limits_{n=1}^{N}(s-\tilde{z}_n)}{\prod\limits_{n=1}^{N}(s-\tilde{a}_n)}}=\frac{\prod\limits_{n=1}^{N+1}(s-z_n)}{\prod\limits_{n=1}^{N}(s-\tilde{z}_n)} \tag{2.11}$$

Equation (2.11) reveals *that the poles of $f(s)$ become equal to $\tilde{z}_n$, which are the zeros of $\sigma(s)$!* The initial poles $\tilde{a}_n$, which we have assumed to be known and fixed beforehand, are cancelled out in the process as both $\sigma(s)$ and $\sigma(s)f(s)$ have been formulated to have the same set of poles. Thus, if we can solve for the zeros of $\sigma(s)$, the poles of $f(s)$ may be inferred directly from it. We recall at this point that the residues of $\sigma(s)$, $(\tilde{c}_1-\tilde{c}_N)$, have been solved for in the previous step from Equation (2.8) and hence all that remains is to convert from the residues to the zeros. This can be done quite simply as is presented next.

A linear system can in general be described as a state equation realization in the form of

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.12}$$

where x is the state vector with $\dot{x} = dx/dt$, u is the input vector and y is the output vector.

For the function $\sigma(s)$, A is a diagonal matrix holding the poles $\tilde{a}_n$, B is a column vector of ones, C is a row vector holding its residues $\tilde{c}_n$, and D is unity. In order to solve for the zeros of $\sigma(s)$, we return to Equation (2.2) and rewrite it in the form of a fraction:

$$\sigma(s) = \sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 = \frac{\prod_{n=1}^{N} (s - \tilde{z}_n)}{\prod_{n=1}^{N} (s - \tilde{a}_n)} = \frac{y(s)}{u(s)} \tag{2.13}$$

Notice that the zeros of $\sigma(s)$ are equal to the poles of $1/\sigma(s)$. With Equation (2.12) as the representation for $\sigma(s)$, we can obtain the expression for $1/\sigma(s)$ by interchanging the input and the output. Solving for u in the second equation in (2.12) and plugging it into the first, we obtain

$$u = D^{-1}(y - Cx) \tag{2.14}$$

$$\dot{x} = Ax + BD^{-1}(y - Cx) = Ax + BD^{-1}y - BD^{-1}Cx = (A - BD^{-1}C)x + BD^{-1}y \tag{2.15}$$

Equation (2.15) reveals that the poles of $1/\sigma(s)$ can be calculated as

$$eig(A - BD^{-1}C) \tag{2.16}$$

Since D is unity, Equation (2.16) simplifies to

$$eig(A - BC) \tag{2.17}$$

In summary, the poles of $f(s)$, which we have shown to be equal to the zeros of $\sigma(s)$, can be calculated as the eigenvalues of the matrix $(A - BC)$, where

$$A = \begin{bmatrix} \tilde{a}_1 & & 0 \\ & \ddots & \\ 0 & & \tilde{a}_N \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} \tilde{c}_1 & \cdots & \tilde{c}_N \end{bmatrix} \tag{2.18}$$

This concludes the pole identification stage of vector fitting. Recall that in the process of solving for the poles, two pieces of information had to be known and fixed in advance. They are:

1. The order of approximation, N
2. The set of initial starting poles, $\tilde{a}_1 - \tilde{a}_N$

Consequently, the first step before running the vector fitting algorithm on any set of data is to specify these two pieces of information, which will act as inputs to the vector fitting process. Further information on how to choose these parameters can be found in Section 2.9.

2.4 Stage Two: Residue Identification

In this stage, the residues c_n of the system are solved for. While it is possible to extract the residues in the previous stage, a more accurate result is generally obtained if the newly solved poles are used in place of the starting poles that were used in stage one. We return to Equation (2.1) which is repeated here for

convenience:

$$f(s) = \sum_{n=1}^{N} \frac{c_n}{s - a_n} + d + sh \tag{2.19}$$

Since the poles a_n have been fully determined in the previous stage, Equation (2.19) is now linear in terms of the unknowns c_n, d and h. We proceed as before by writing Equation (2.19) for each frequency sample point to obtain:

$$\begin{bmatrix} \frac{1}{s_1 - a_1} & \cdots & \frac{1}{s_1 - a_N} & 1 & s_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{1}{s_k - a_1} & \cdots & \frac{1}{s_k - a_N} & 1 & s_k \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_N \\ d \\ h \end{bmatrix} = \begin{bmatrix} f(s_1) \\ \vdots \\ \\ \vdots \\ f(s_k) \end{bmatrix} \tag{2.20}$$

Equation (2.20) again results in an overdetermined problem which can be solved as before for the residues c_n, the constant d and the linear term h.

This concludes the residue identification stage of vector fitting. Along with the previous stage, we see that we have now solved for all the unknown coefficients in Equation (2.1) and thus found a rational function which approximates our system, $f(s)$. We see, however, that the solution is not guaranteed to be exact but instead depends on minimizing the error of a set of two least squares problem. Thus, at this point, one would normally compare the approximation to the original data and determine if they are within an acceptable range. If necessary, a more accurate solution can be obtained if the vector fitting algorithm is repeated on the data by using the *newly calculated poles as starting poles.* Therefore, vector fitting is often seen as an iterative scheme whereby the poles are relocated until they converge with the actual poles of the system. Normally this is achieved rather quickly and it takes an average 2 – 4 iterations to obtain an accurate result.

2.5 Modifications for Complex Poles

An important modification to the vector fitting algorithm which is often made when solving for real systems with complex poles will now be presented. For real systems, the poles must either be real or occur in complex-conjugate pairs. In addition, the residues corresponding to the real poles must be real and, similarly, the residues corresponding to the complex-conjugate pair poles must also be in complex-conjugate pairs. In order to make the necessary adjustment to Equation (2.8) in order to ensure this condition, we return to Equation (2.6) and rewrite it for systems with both real and complex poles.

Assume a system with Q real poles and L complex-conjugate pole pairs where an asterisk "*" is used as a notation to indicate complex-conjugacy. Thus for a complex pair, we would have

$$a_n = a_n^r + ja_n^i, \qquad a_n^* = a_n^r - ja_n^i \tag{2.21}$$

$$c_n = c_n^r + jc_n^i, \qquad c_n^* = c_n^r - jc_n^i \tag{2.22}$$

with the superscript r representing the real part and the superscript i representing the imaginary part. Equation (2.6) now becomes

$$\begin{aligned} &\left[\sum_{q=1}^{Q} \frac{c_q}{s-\tilde{a}_q} + \sum_{l=1}^{L}\left(\frac{c_l}{s-\tilde{a}_l} + \frac{c_l^*}{s-\tilde{a}_l^*}\right) + d + sh\right] \\ &\quad - \left[\sum_{q=1}^{Q} \frac{\tilde{c}_q}{s-\tilde{a}_q} + \sum_{l=1}^{L}\left(\frac{\tilde{c}_l}{s-\tilde{a}_l} + \frac{\tilde{c}_l^*}{s-\tilde{a}_l^*}\right)\right] \cdot f(s) \approx f(s) \end{aligned} \tag{2.23}$$

Since each complex pair consists of two poles, we have that the order of approximation $N = Q + 2L$.

The elements of the matrix A in Equation (2.7) then become

$$A_{k,q} = \frac{1}{s_k - \tilde{a}_q} \tag{2.24}$$

for each of the real poles and

$$A_{k,l} = \frac{1}{s_k - \tilde{a}_l} + \frac{1}{s_k - \tilde{a}_l^*}, \qquad A_{k,l+1} = \frac{j}{s_k - \tilde{a}_l} - \frac{j}{s_k - \tilde{a}_l^*} \tag{2.25}$$

for each of the complex pole pairs.

Equation (2.8) can now be rewritten to handle complex-conjugate pole pairs. Writing Equation (2.23) for each frequency sample point with the help of Equations (2.24) and (2.25), gives

$$\begin{bmatrix} & & 1 & s_1 & & \\ [R] & [C] & \vdots & \vdots & [G] & [H] \\ & & 1 & s_k & & \end{bmatrix} [x] = \begin{bmatrix} f(s_1) \\ \vdots \\ f(s_k) \end{bmatrix} \tag{2.26}$$

where the matrices $[R]$, $[C]$, $[G]$ and $[H]$ are as follows.

$$R = \begin{bmatrix} \frac{1}{s_1-\tilde{a}_1} & \cdots & \frac{1}{s_1-\tilde{a}_Q} \\ \vdots & \ddots & \vdots \\ \frac{1}{s_k-\tilde{a}_1} & \cdots & \frac{1}{s_k-\tilde{a}_Q} \end{bmatrix} \tag{2.27}$$

$$C = \begin{bmatrix} \frac{1}{s_1-\tilde{a}_1} + \frac{1}{s_1-\tilde{a}_1^*} & \frac{j}{s_1-\tilde{a}_1} - \frac{j}{s_1-\tilde{a}_1^*} & \cdots & \frac{1}{s_1-\tilde{a}_L} + \frac{1}{s_1-\tilde{a}_L^*} & \frac{j}{s_1-\tilde{a}_L} - \frac{j}{s_1-\tilde{a}_L^*} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{s_k-\tilde{a}_1} + \frac{1}{s_k-\tilde{a}_1^*} & \frac{j}{s_k-\tilde{a}_1} - \frac{j}{s_k-\tilde{a}_1^*} & \cdots & \frac{1}{s_k-\tilde{a}_L} + \frac{1}{s_k-\tilde{a}_L^*} & \frac{j}{s_k-\tilde{a}_L} - \frac{j}{s_k-\tilde{a}_L^*} \end{bmatrix} \tag{2.28}$$

$$G = \begin{bmatrix} \frac{-f(s_1)}{s_1-\tilde{a}_1} & \cdots & \frac{-f(s_1)}{s_1-\tilde{a}_Q} \\ \vdots & \ddots & \vdots \\ \frac{-f(s_k)}{s_k-\tilde{a}_1} & \cdots & \frac{-f(s_k)}{s_k-\tilde{a}_Q} \end{bmatrix} \tag{2.29}$$

$$H = \begin{bmatrix} \frac{-f(s_1)}{s_1-\tilde{a}_1} + \frac{-f(s_1)}{s_1-\tilde{a}_1^*} & \frac{-jf(s_1)}{s_1-\tilde{a}_1} - \frac{-jf(s_1)}{s_1-\tilde{a}_1^*} & \cdots & \frac{-f(s_1)}{s_1-\tilde{a}_L} + \frac{-f(s_1)}{s_1-\tilde{a}_L^*} & \frac{-jf(s_1)}{s_1-\tilde{a}_L} - \frac{-jf(s_1)}{s_1-\tilde{a}_L^*} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{-f(s_k)}{s_k-\tilde{a}_1} + \frac{-f(s_k)}{s_k-\tilde{a}_1^*} & \frac{-jf(s_k)}{s_k-\tilde{a}_1} - \frac{-jf(s_k)}{s_k-\tilde{a}_1^*} & \cdots & \frac{-f(s_k)}{s_k-\tilde{a}_L} + \frac{-f(s_k)}{s_k-\tilde{a}_L^*} & \frac{-jf(s_k)}{s_k-\tilde{a}_L} - \frac{-jf(s_k)}{s_k-\tilde{a}_L^*} \end{bmatrix} \tag{2.30}$$

Notice that when there are no complex poles, (i.e., $L = 0$) the matrices C and H become the empty matrix and Equation (2.26) reduces to Equation (2.8) with $N = Q$.

We are now ready to solve for the unknown residues for the case with complex poles. Because it is often desirable to solve the overdetermined set of equations in terms of real numbers, and since two complex numbers are only equal if both their real and imaginary quantities match, Equation (2.7) is rewritten in terms of real quantities by separating the real and imaginary parts as follows:

$$\begin{bmatrix} Re(A) \\ Im(A) \end{bmatrix} [x] = \begin{bmatrix} Re(b) \\ Im(b) \end{bmatrix} \tag{2.31}$$

This has the effect that the number of equations in the overdetermined set is changed from k complex equations to $2k$ real equations, where k is the number of frequency sample points.

Equation (2.31) can then be solved for the solution vector x using any of the standard least squares methods to yield

$$x = \begin{bmatrix} c_1 & \cdots & c_Q & c_1^r & c_1^i & \cdots & c_L^r & c_L^i & d & h & \tilde{c}_1 & \cdots & \tilde{c}_Q & \tilde{c}_1^r & \tilde{c}_1^i & \cdots & \tilde{c}_L^r & \tilde{c}_L^i \end{bmatrix}^T \tag{2.32}$$

where all the elements are purely real. The complex residues are then formed from Equation (2.22), where we would have

$$c_l = c_l^r + jc_l^i, \qquad c_{l+1} = c_l^* = c_l^r - jc_l^i \tag{2.33}$$

$$\tilde{c}_l = \tilde{c}_l^r + j\tilde{c}_l^i, \qquad \tilde{c}_{l+1} = \tilde{c}_l^* = \tilde{c}_l^r - j\tilde{c}_l^i \tag{2.34}$$

As in the case with purely real poles, we collect the residues of $\sigma(s)$, $(\tilde{c}_1 - \tilde{c}_N)$ and discard the other values as a more accurate result can be obtained in the residue identification stage. The poles of the system $f(s)$ can then be solved as before by solving Equation (2.17). However, note that each of the matrix entries A and C in Equation (2.18) can now be complex, resulting in the entire matrix $(A - BC)$ being complex. To account for this, we modify the matrices A, B, and C for each complex entry via a similarity transformation to yield the submatrices

$$\hat{A} = \begin{bmatrix} Re(\tilde{a}) & Im(\tilde{a}) \\ -Im(\tilde{a}) & Re(\tilde{a}) \end{bmatrix}, \qquad \hat{B} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \qquad \hat{C} = \begin{bmatrix} Re(\tilde{c}) & Im(\tilde{c}) \end{bmatrix} \tag{2.35}$$

As a result, the matrices are now real matrices and any complex eigenvalue will come along with its complex-conjugate pair, thus preserving the properties of a real system.

Once the poles of the system are solved for, the residues can then be calculated as before. We again return to Equation (2.19) and rewrite it for cases with complex poles to yield

$$f(s) = \sum_{q=1}^{Q} \frac{c_q}{s - a_q} + \sum_{l=1}^{L} \left(\frac{c_l}{s - a_l} + \frac{c_l^*}{s - a_l^*} \right) + d + sh \tag{2.36}$$

The matrix in Equation (2.20) now becomes

$$\begin{bmatrix} & & 1 & s_1 \\ [R] & [C] & \vdots & \vdots \\ & & 1 & s_k \end{bmatrix} [x] = \begin{bmatrix} f(s_1) \\ \vdots \\ f(s_k) \end{bmatrix} \tag{2.37}$$

where the matrices $[R]$ and $[C]$ are the same as in Equations (2.27) and (2.28), respectively. Solving Equation (2.37) yields the unknown vector $[x]$ in the form of

$$x = \begin{bmatrix} c_1 & \cdots & c_Q & c_1^r & c_1^i & \cdots & c_L^r & c_L^i & d & h \end{bmatrix}^T \tag{2.38}$$

and the complex residues can be formed as

$$c_l = c_l^r + jc_l^i, \qquad c_{l+1} = c_l^* = c_l^r - jc_l^i \tag{2.39}$$

This concludes the process as all the poles, residues, constant and proportional terms have been identified for cases with complex poles.

2.6 Modification for Fitting Vector Functions

So far we have considered the case for fitting a scalar or a single function. However, it is sometimes desirable to fit a vector or multiple functions using the same set of poles since this would result in an increase in efficiency in the time domain convolutions. The modification for fitting vectors is rather straightforward and is presented below.

Consider a vector of Nc functions:

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{Nc} \end{bmatrix} \tag{2.40}$$

For this function, Equation (2.6) now becomes

$$\begin{bmatrix} \sum_{n=1}^{N} \frac{c_n^1}{s-\tilde{a}_n} + d^1 + sh^1 \\ \sum_{n=1}^{N} \frac{c_n^2}{s-\tilde{a}_n} + d^2 + sh^2 \\ \vdots \\ \sum_{n=1}^{N} \frac{c_n^{Nc}}{s-\tilde{a}_n} + d^{Nc} + sh^{Nc} \end{bmatrix} - \begin{bmatrix} f_1 \sum_{n=1}^{N} \frac{\tilde{c}_n}{s-\tilde{a}_n} \\ f_2 \sum_{n=1}^{N} \frac{\tilde{c}_n}{s-\tilde{a}_n} \\ \vdots \\ f_{Nc} \sum_{n=1}^{N} \frac{\tilde{c}_n}{s-\tilde{a}_n} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{Nc} \end{bmatrix} \tag{2.41}$$

The residues can then be solved from Equation (2.7) where the matrix given in Equation (2.8) now becomes

$$\begin{bmatrix} [X_{\sigma f}] & 0 & 0 & 0 & -f_1 [X_\sigma] \\ 0 & [X_{\sigma f}] & 0 & 0 & -f_2 [X_\sigma] \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & 0 & [X_{\sigma f}] & -f_{Nc} [X_\sigma] \end{bmatrix} \begin{bmatrix} [Y_1] \\ [Y_2] \\ \vdots \\ [Y_{Nc}] \\ \left[\tilde{Y}\right] \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{Nc} \end{bmatrix} \tag{2.42}$$

where

$$X_{\sigma f} = \begin{bmatrix} \frac{1}{s_1-\tilde{a}_1} & \cdots & \frac{1}{s_1-\tilde{a}_N} & 1 & s_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{1}{s_k-\tilde{a}_1} & \cdots & \frac{1}{s_k-\tilde{a}_N} & 1 & s_k \end{bmatrix} \tag{2.43}$$

$$X_\sigma = \begin{bmatrix} \frac{1}{s_1-\tilde{a}_1} & \cdots & \frac{1}{s_1-\tilde{a}_N} \\ \vdots & \ddots & \vdots \\ \frac{1}{s_k-\tilde{a}_1} & \cdots & \frac{1}{s_k-\tilde{a}_N} \end{bmatrix} \tag{2.44}$$

$$Y_{nc} = \begin{bmatrix} c_1^{nc} & \cdots & c_N^{nc} & d^{nc} & h^{nc} \end{bmatrix}^T, \qquad nc \in 1, 2 \ldots Nc \tag{2.45}$$

$$\tilde{Y} = \begin{bmatrix} \tilde{c}_1 & \cdots & \tilde{c}_N \end{bmatrix}^T \tag{2.46}$$

After solving this, we again collect the residues of $\sigma(s)$, $(\tilde{c}_1 - \tilde{c}_N)$ and solve for the poles using Equation (2.17) where the matrix elements are given in Equation (2.18). This has the effect that a single set of poles that minimizes the least squares error in all elements of Equation (2.40) is obtained. The residues of the individual functions can then be solved for by carrying out the residue identification stage independently for each element in Equation (2.40). It should also be noted that if complex poles are used, the modifications presented in Section 2.5 should also be carried out for each element of the vector.

2.7 Modification for Fast Fitting Vector Functions

A method to improve the speed of the vector fitting process when fitting multiple functions using the same set of poles will now be presented. As presented before, the first step of the vector fitting method is to solve for the residues of $\sigma(s)$ from an overdetermined set of equations. When fitting multiple functions using the same set of poles, the size of this overdetermined set of equations may get prohibitively large. However, note that only part of the solution vector was used while the other part was simply discarded, which can be thought of as a waste in computational resources. When fitting multiple functions using the same set of poles, this waste get significantly larger. For example, in Equation (2.42), only the

solution vector $\tilde{Y}$ is needed while the others $(Y_1 - Y_{Nc})$ were discarded. A method to minimize this waste have been recently proposed which results in a significant speedup of the overall process [2].

When fitting multiple functions simultaneously, instead of solving the large Equation (2.42) for the residues, a QR decomposition is first applied to the least squares equations of the single element

$$\begin{bmatrix} [X_{\sigma f}] & -f\,[X_{\sigma}] \end{bmatrix} = [Q] \begin{bmatrix} R^{11} & R^{12} \\ R^{21} & R^{22} \end{bmatrix} \tag{2.47}$$

Once all the Q and R submatrices have been extracted, an overall overdetermined set of equations is formed to solve for the residues of $\sigma(s)$ as

$$\begin{bmatrix} R_1^{22} \\ R_2^{22} \\ \vdots \\ R_{Nc}^{22} \end{bmatrix} \left[\tilde{Y}\right] = \begin{bmatrix} Q_1^T f_1 \\ Q_2^T f_2 \\ \vdots \\ Q_{Nc}^T f_{Nc} \end{bmatrix} \tag{2.48}$$

This has the effect that the new overdetermined set of equations is now significantly smaller than before and the solution vector is only the residues of $\sigma(s)$. Although it requires solving the QR decomposition of each individual element to be fitted, that process is often less time consuming since the matrices are much smaller. When the number of elements to be fitted increases, for example in multiport devices with a large number of ports, the savings could be enormous.

2.8 Stability Consideration

For a causal system to be stable, the poles of the system must lie in the left half-plane of the s-domain [3]. In the vector fitting process, however, it is possible to obtain unstable poles when solving Equation (2.17). This can easily be corrected by one of two different methods:

1. Discard any unstable poles that were obtained from Equation (2.17).
2. Flip unstable poles into the left half-plane by reflecting it on the imaginary axis of the s-domain.

While these two methods are equally effective at eliminating unstable poles, performing the former will result in a reduction in the original order of approximation, which might not be desirable. For that reason, in this thesis we will adopt the latter method.

2.9 Starting Pole Selection Method

In this section, a method to select the initial starting poles will be introduced. In Section 2.3, we see that the vector fitting method requires that an initial set of starting poles be specified to be used as a preliminary guess of the actual poles. Although these starting poles cancel out in the subsequent formulation, a poor choice of these values can result in a large variation between the original function and the fitted function as the vector fitting method relies on solving Equation (2.7) in a least squares sense.

A generally good fit is obtained if the starting poles are selected to be in complex conjugate pairs situated along a line close to the imaginary axis [1]:

$$\tilde{a}_n = -\alpha + j\beta, \qquad \tilde{a}_{n+1} = -\alpha - j\beta \tag{2.49}$$

where α and β are real numbers.

Also, choosing poles too far left in the complex plane results in the real part dominating the matrix entries [4]

$$\frac{1}{s_k - \tilde{a}_n} = \frac{1}{j\left(\omega_k - Im(\tilde{a}_n)\right) - Re(\tilde{a}_n)} \approx \frac{-1}{Re(\tilde{a}_n)} \tag{2.50}$$

which results in a poor conditioning of the system of linear equations. Thus we choose

$$\alpha = \beta/100 \tag{2.51}$$

such that the real parts of the starting poles are much smaller than the imaginary parts. This often results in an extremely accurate fit within a few iterations.

2.10 Conclusion

An overall flowchart showing the whole vector fitting process is shown in Figure 2.1.

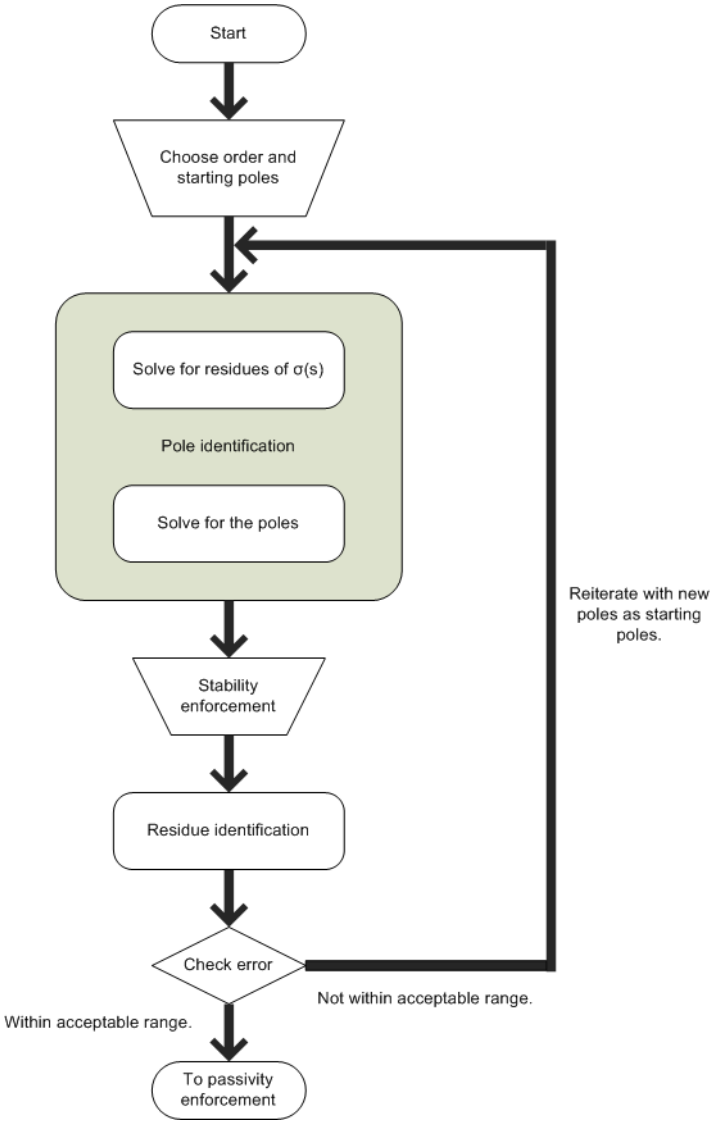


Figure 2.1: Flowchart of the vector fitting process.

CHAPTER 3

PASSIVITY ASSESSMENT AND ENFORCEMENT

3.1 Overview

In this section, the methodology used for passivity assessment and enforcement will be introduced. In Chapter 2 we have presented the vector fitting method which is used to generate a macromodel of the system. While we have enforced stability in the macromodel (by ensuring that all the poles of the system are located in the left half-plane of the s-domain), we have not considered another important characteristic of rational models, namely passivity. Passivity is defined as the inability of the system to generate energy in any termination condition [5]. If the system being modeled is passive, then the macromodel generated must be passive as well, since stable but nonpassive models can result in unstable systems when connected to other passive components [6, 7]. Thus, ensuring passivity of the model is a crucial step in the macromodel generation process. The discussion in this chapter will be organized as follows. First the mathematical condition for a passive system will be presented in Section 3.2. Then the passivity assessment process will be detailed in Section 3.3. Finally the steps involved in enforcing passivity will be explained in Section 3.4.

3.2 Passivity of a System Characterized by Scattering Parameters

A precise mathematical definition of passivity depends on the adopted representation [3]. For a system characterized by the scattering parameters $S(s)$, the condition for passivity is [8, 9]

1. $S(s^*) = S^*(s)$ where "*" denotes the complex conjugate operator.

2. $S(s)$ is bounded real.

   I.e., $\|S(j\omega)\| \leq 1$ or $eig\left(I - S(j\omega)^H S(j\omega)\right) \geq 0, \qquad \omega \in \mathbb{R}$

Condition 1 is always satisfied in our macromodel since, in the vector fitting process, the complex poles and residues are always considered along with their conjugates, thus leading to only real coefficients in $S(s)$. Consequently, enforcing passivity of the macromodel then amounts to enforcing condition 2.

3.3 Passivity Assessment

In this section, general methods used to check for passivity of a system characterized by the scattering parameters will be discussed. From the above conditions, we see that a fast and easy way to check for passivity is by evaluating the norm of the scattering matrix $S(j\omega)$ or the eigenvalues of the dissipation matrix $I - S(j\omega)^H S(j\omega)$ and determine if condition 2 above is satisfied. However, a major drawback of this method of passivity assessment is that the above condition must be evaluated at discrete frequency points and thus it is practically impossible to check for passivity at all frequency values. In addition, whenever a passivity violation is found, the exact location of the violation (which becomes important in the passivity enforcement step) cannot be accurately determined for the same

reason as above. Thus, recent literature has resorted to a more robust method of passivity assessment, which will be presented next.

Consider an m-port system characterized by the S-parameters written in rational form of order N as

$$S(s) = [S_{ij}(s)], \qquad S_{ij}(s) \approx \sum_{n=1}^{N} \frac{c_n^{ij}}{s - a_n} + d^{ij}, \qquad i, j \in m \tag{3.1}$$

This rational function representation can be obtained by utilizing the vector fitting method presented in Chapter 2. The state-space representation of the corresponding system is then

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3.2}$$

where $A \in \mathbb{R}^{N \times N}$ is the state matrix containing the poles, $B \in \mathbb{R}^{N \times m}$ is the input mapping matrix, $C \in \mathbb{R}^{m \times N}$ is the output matrix containing the residues and $D \in \mathbb{R}^{m \times m}$ consists of the direct coupling terms. The system is passive if and only if the Hamiltonian matrix M has no imaginary eigenvalues [8], where the Hamiltonian is given by

$$M = \begin{bmatrix} A + BKD^T C & BKB^T \\ -C^T LC & -A^T - C^T DKB^T \end{bmatrix} \tag{3.3}$$

where

$$\begin{aligned} K &= \left(I - D^T D\right)^{-1} \\ L &= \left(I - DD^T\right)^{-1} \end{aligned} \tag{3.4}$$

Since the Hamiltonian is independent of frequency, only a single evaluation is necessary to determine whether or not the system is passive. In addition, if any purely imaginary eigenvalue is found, it has been shown that it corresponds to the

point where a singular value of the scattering matrix becomes equal to one (and hence where the eigenvalues of the dissipation matrix $I - S(j\omega)^H S(j\omega)$ are equal to zero) [8]. Thus, this information can be used to pinpoint the exact locations of passivity violations, as will be seen next.

Consider a plot of the eigenvalues of the dissipation matrix $I - S(j\omega)^H S(j\omega)$ of a general m-port scattering matrix shown in Figure 3.1 (note that only plots of two eigenvalues are shown). From Figure 3.1, we see that there are four points (marked #1 to #4) where the eigenvalues of the dissipation matrix are equal to zero, thus defining potential points where the system crosses from being passive to nonpassive. As mentioned before, these points can be obtained by solving for the eigenvalues of the Hamiltonian matrix, specifically those that are purely imaginary. In order to obtain the bands of passivity violations, we return to condition 2 above and check whether or not the system is passive for a short distance right before and after the potential crossover frequency. If the system is found to be passive right before the point of consideration but not passive after, the point is defined as a crossover frequency where the system crosses from being passive to nonpassive (i.e., point #1 in Figure 3.1). If the system is not passive right before the point of consideration but is passive after, the point is defined as a crossover frequency where the system crosses from being nonpassive to passive (i.e., point #4 in Figure 3.1). On the other hand, if the system is both nonpassive right before and after the point of consideration, then it is concluded that the point is contained within a larger passivity violation band due to the other eigenvalues (i.e., points #2 and #3 in Figure 3.1). Thus we are able to determine the exact band of passivity violation by arranging the points in order and determining all the crossover frequencies. In the example given in Figure 3.1, the band would be from ω_1 to ω_4.

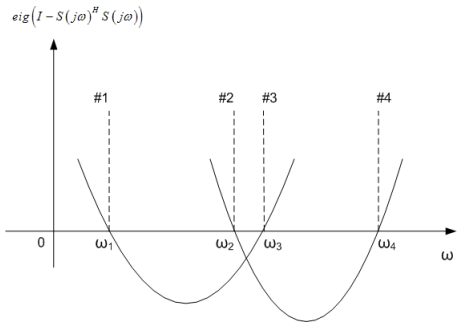


Figure 3.1: Determination of the band of passivity violation.

So far we have seen how the bands of passivity violations are determined with the use of the Hamiltonian matrix. Before we proceed to the passivity enforcement section, let us first see how two other important quantities which are needed for passivity enforcement are determined. These two quantities are the frequency of maximum violation and the magnitude of maximum violation in each violation band. These locations can be found by solving [9]

$$\lambda = \max \left| eig\left(I - S(j\omega)^H S(j\omega)\right)\right|, \qquad \omega \in \omega_l, \omega_h \tag{3.5}$$

where λ is the magnitude of maximum violation and ω_l and ω_h are the boundaries of the passivity violation band. This is easily solved by doing a fine sweep of each frequency violation band that was found and recording the maximum value and the corresponding frequency point as given in Equation (3.5). With this information, we are now able to proceed to enforce passivity for nonpassive systems.

3.4 Passivity Enforcement

Consider again the state-space representation of a system given in Equation (3.2). The scattering matrix of this system can be obtained from

$$S(j\omega) = C\left(j\omega I - A\right)^{-1} B + D \tag{3.6}$$

For this system to be passive, it must obey condition 2 presented in Section 3.2

$$eig\left(Q(j\omega)\right) \geq 0 \tag{3.7}$$

at all frequency points, where $Q(j\omega)$ denotes the dissipation matrix

$$Q(j\omega) = I - S(j\omega)^H S(j\omega) \tag{3.8}$$

If the system is nonpassive, we can attempt to restore passivity by perturbing the representation of the system given in Equation (3.6) by a small amount such that the new system satisfies Equation (3.7) at the frequency points of violation. This can be done by perturbing any or all of the matrices associated with the right-hand side of Equation (3.6). In the algorithm presented, we will perturb only the residues (contained in matrix C) associated with the system.

For a small perturbation ΔC, matrix C becomes

$$\hat{C} = C + \Delta C \tag{3.9}$$

This results in a change of the scattering matrix given in Equation (3.6), where we now have

$$\hat{S}(j\omega) = \left(C + \Delta C\right)\left(j\omega I - A\right)^{-1} B + D \tag{3.10}$$

which can be written as

$$\hat{S}(j\omega) = S(j\omega) + \Delta S(j\omega) \tag{3.11}$$

where

$$\Delta S(j\omega) = \Delta C \left(j\omega I - A\right)^{-1} B = \Delta C V \tag{3.12}$$

with

$$V = \left(j\omega I - A\right)^{-1} B \tag{3.13}$$

In order to ensure passivity, the new scattering matrix $\hat{S}(j\omega)$ given in Equation (3.11) must obey condition 2 given in Section 3.2:

$$eig\left(\hat{Q}(j\omega)\right) \geq 0 \tag{3.14}$$

where $\hat{Q}(j\omega)$ is the new dissipation matrix

$$\hat{Q}(j\omega) = I - \hat{S}(j\omega)^H \hat{S}(j\omega) \tag{3.15}$$

Substituting Equation (3.11) into Equation (3.15) gives (dropping $j\omega$ for simplicity)

$$\hat{Q} = I - \hat{S}^H \hat{S} = I - S^H S - S^H \Delta S - \Delta S^H S - \Delta S^H \Delta S \tag{3.16}$$

Neglecting the second-order term in Equation (3.16), we get

$$\hat{Q} \approx I - S^H S - S^H \Delta S - \Delta S^H S \tag{3.17}$$

Comparing Equation (3.17) to Equation (3.8) reveals that the perturbation results in a change of

$$\Delta Q = -S^H \Delta S - \Delta S^H S \tag{3.18}$$

from the unperturbed system. Thus, if the unperturbed nonpassive system violates Equation (3.7) at a particular frequency by an amount λ, we can restore passivity at that point by perturbing the system such that the change in the dissipation matrix given by Equation (3.18) results in a change of its eigenvalue by an amount equal and opposite to λ. To do this, we invoke the first-order eigenvalue perturbation formula [10] which states that a matrix K perturbed by an amount ΔK will result in a change of $\Delta\lambda$ in its eigenvalue given by

$$\Delta\lambda = \frac{y^T \Delta K x}{y^T x} \tag{3.19}$$

where y and x are the left and right eigenvectors of K, respectively. Therefore, a matrix Q given in Equation (3.8) perturbed by an amount ΔQ given by Equation (3.18) would result in a change in its eigenvalue by an amount

$$\Delta\lambda = \frac{v^T \left(-S^H \Delta S - \Delta S^H S\right) u}{v^T u} \tag{3.20}$$

where v and u are the left and right eigenvectors of Q, respectively. Since for a matrix A, the eigenvalues and eigenvectors can be solved such that $A = VDV^{-1}$ where V is a modal matrix (its columns are the eigenvectors of A) and D is the canonical form of A (a diagonal matrix with the eigenvalues of A on the main diagonal), the eigenvectors can be scaled such that $v^T u$ would result in unity for a given eigenvalue. Thus, dropping the term $v^T u$ and substituting Equation (3.12) in Equation (3.20) gives

$$\Delta\lambda = v^T \left(-S^H \Delta C V - (\Delta C V)^H S\right) u \tag{3.21}$$

which can be written as

$$\Delta\lambda = v^T \left(-S^H \Delta C V - V^H \Delta C^H S\right) u \tag{3.22}$$

Since ΔC is a real matrix, $\Delta C^H = \Delta C^T$

$$\Delta\lambda = v^T \left(-S^H \Delta C V - V^H \Delta C^T S\right) u = -v^T S^H \Delta C V u - v^T V^H \Delta C^T S u \tag{3.23}$$

Next we invoke an identity of the Kronecker product $\otimes$ which states that for a given matrix Y, A, X, and B [11]

$$Y = AXB \;\Leftrightarrow\; vec\,(Y) = \left(B^T \otimes A\right) vec\,(X) \tag{3.24}$$

$$Y = AXB \;\Leftrightarrow\; wec\,(Y) = \left(A \otimes B^T\right) wec\,(X) \tag{3.25}$$

where $vec\,(.)$ denotes the vectorization of the matrix (.) formed by column-ordering the matrix (.) into a single column vector and $wec\,(.)$ denotes the vectorization of the matrix (.) formed by row-ordering the matrix (.) into a single column vector. Applying Equations (3.24) and (3.25) along with the fact that $\Delta\lambda$ is a scalar on Equation (3.23), results in

$$\Delta\lambda = -\left((Vu)^T \otimes v^T S^H\right) vec\,(\Delta C) - \left(v^T V^H \otimes (Su)^T\right) wec\left(\Delta C^T\right) \tag{3.26}$$

Since

$$wec\left(\Delta C^T\right) = vec\,(\Delta C) \tag{3.27}$$

we have

$$\begin{aligned} \Delta\lambda &= -\left((Vu)^T \otimes v^T S^H\right) vec\,(\Delta C) - \left(v^T V^H \otimes (Su)^T\right) vec\,(\Delta C) \\ &= -\left[\left((Vu)^T \otimes v^T S^H\right) + \left(v^T V^H \otimes (Su)^T\right)\right] vec\,(\Delta C) \end{aligned} \tag{3.28}$$

which has the form

$$\Delta\lambda = g \cdot vec\,(\Delta C) \tag{3.29}$$

with

$$g = -\left[\left((Vu)^T \otimes v^T S^H\right) + \left(v^T V^H \otimes (Su)^T\right)\right] \tag{3.30}$$

which can be shown to be a row vector. Thus, for a passivity violation at a particular frequency, Equation (3.29) provides the means for restoring passivity at that point.

When a passivity violation band is detected, Equation (3.29) is applied to the point of maximum violation which was obtained by solving Equation (3.5). For cases where there are more than one violation band, passivity compensation can be done simultaneously for all violation bands by setting up Equation (3.29) for each band, resulting in a set of least-squares equations in the form of

$$\Delta\lambda = G \cdot vec\,(\Delta C) \tag{3.31}$$

where $\Delta\lambda$ is a vector formed by the magnitudes of maximum violations in each band and G is a matrix consisting of several rows of g's.

In order to retain the accuracy of the model, we minimize the change in the scattering matrix as passivity enforcement is carried out. To do this, we return to Equation (3.11) which defines the change in the scattering matrix after passivity compensation and relate that to the perturbation of the residues ΔC. It can be

shown [12, 13] that

$$\|\Delta S\|^2 = trace\left(\Delta C P \Delta C^T\right) = vec\left(\Delta C\right)^T H vec\left(\triangle C\right) \tag{3.32}$$

where P is the controllability Grammian obtained by solving the Lyapunov equation [14]

$$AP + PA^H + BB^H = 0 \tag{3.33}$$

and H is a matrix formed by stacking P on the diagonal

$$H = \begin{bmatrix} P & 0 & \cdots & 0 \\ 0 & P & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P \end{bmatrix}_{(m\times N)by(m\times N)} \tag{3.34}$$

Equations (3.31) and (3.32) together result in an optimization problem which can be solved iteratively to satisfy passivity while minimizing the change in the response. Since the objective function given in Equation (3.32) is quadratic in nature, the problem is solved by utilizing a quadratic programming routine where the overall problem is

$$\min\left(vec\left(\Delta C\right)^T H vec\left(\triangle C\right)\right) \qquad \text{subject to} \qquad \Delta\lambda = G \cdot vec\left(\Delta C\right) \tag{3.35}$$

3.5 Conclusion

The overall process of passivity enforcement is summarized in the flowchart given in Figure 3.2.

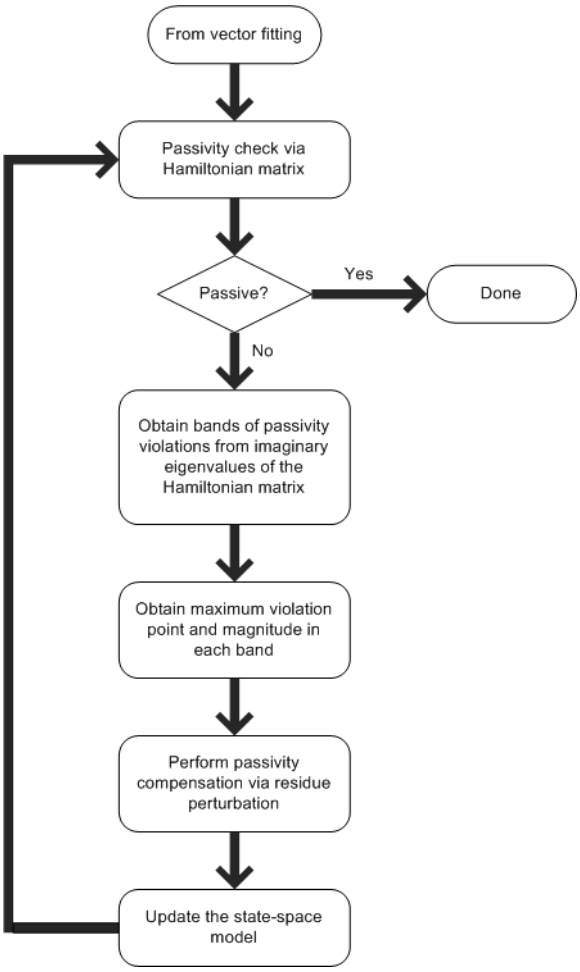


Figure 3.2: Flowchart of the passivity enforcement process.

CHAPTER 4

TIME DOMAIN SIMULATION

4.1 Overview

In this chapter, the recursive convolution algorithm which is used to obtain the time domain response of the system will be introduced. For a system characterized by rational functions, the transfer function $H(s)$ can be approximated to take the form

$$H(s) = \sum_{n=1}^{N} \frac{c_n}{s + a_n} + d \tag{4.1}$$

where the coefficients c_n, a_n and d can be obtained by using the vector fitting method presented in Chapter 2. Note that for the discussions in this chapter, the negative sign in the denominator of Equation (2.1) has been absorbed into a_n in Equation (4.1). For this transfer function, its input-output relationship is given by

$$Y(s) = H(s) \cdot X(s) \tag{4.2}$$

where $X(s)$ is the input function and $Y(s)$ is the output function. For each term in the summation of Equation (4.1), we have

$$Y_n(s) = H_n(s) \cdot X(s) = \left[\frac{c_n}{s + a_n} \right] X(s) \tag{4.3}$$

In the time domain this corresponds to

$$y_n(t) = c_n e^{-a_n t} * x(t) \tag{4.4}$$

where "*" denotes the convolution operator. Equation (4.4) can be evaluated most effectively using the recursive convolution [15–17] method which will be presented next.

4.2 The General Recursive Convolution Algorithm

The goal is to evaluate the function

$$y(t) = Ae^{-\alpha t} * x(t) \tag{4.5}$$

This is equivalent to

$$y(t) = \int_0^t Ae^{-\alpha\tau} x(t-\tau)d\tau = \int_0^h Ae^{-\alpha\tau} x(t-\tau)d\tau + \int_h^t Ae^{-\alpha\tau} x(t-\tau)d\tau \tag{4.6}$$

Assuming a step invariant (constant) behavior of the input function, the first integral can be written as

$$\int_0^h Ae^{-\alpha\tau} x(t-\tau)d\tau = Ax(t-h)\int_0^h e^{-\alpha\tau}d\tau \tag{4.7}$$

This can be evaluated to yield

$$\int_0^h Ae^{-\alpha\tau} x(t-\tau)d\tau = \frac{Ax(t-h)}{\alpha}\left(1-e^{-\alpha h}\right) \tag{4.8}$$

Setting $\tau = \tau' + h$ in the second integral yields

$$\begin{aligned} \int_h^t Ae^{-\alpha\tau}x(t-\tau)d\tau &= \int_0^{t-h} Ae^{-\alpha(\tau'+h)}x(t-\tau'-h)d\tau' \\ &= e^{-\alpha h}\int_0^{t-h} Ae^{-\alpha\tau'}x(t-\tau'-h)d\tau' \\ &= e^{-\alpha h}y(t-h) \end{aligned} \tag{4.9}$$

Thus the overall result is then

$$y(t) = \frac{Ax(t-h)}{\alpha}\left(1-e^{-\alpha h}\right) + e^{-\alpha h}y(t-h) \tag{4.10}$$

Equation (4.10) is the general *recursive convolution* formula for a step-invariant approximation.

4.3 Time Domain Evaluation Using the Recursive Convolution Algorithm

Returning to Equation (4.4) and applying the result obtained in Equation (4.10) with a time step $T = h$ gives

$$y_n(t) = e^{-a_n T}y_n(t-T) + \frac{c_n}{a_n}x(t-T)\left(1-e^{-a_n T}\right) \tag{4.11}$$

Therefore the complete solution at each time step is given by

$$y(t) = d\cdot x(t-T) + \sum_{n=1}^{N} y_n(t) \tag{4.12}$$

where $y_n(t)$ is given in Equation (4.11). Equation (4.12) can be evaluated recursively to yield the time domain solution for a particular period $t_{start} - t_{end}$.

Before concluding this chapter, let us examine a special case when the poles and residues appear in complex conjugate pairs. In that instance, Equation (4.11) would yield

$$y_n(t) = e^{-a_n T} y_n(t-T) + \frac{c_n}{a_n} x(t-T)\left(1 - e^{-a_n T}\right) \tag{4.13}$$

$$y_{n+1}(t) = e^{-a_n^* T} y_{n+1}(t-T) + \frac{c_n^*}{a_n^*} x(t-T)\left(1 - e^{-a_n^* T}\right) \tag{4.14}$$

where the asterisk "*" indicates complex conjugacy. It can be shown that $y_{n+1}(t) = y_n^*(t)$. Therefore, this leads to

$$y_n(t) = e^{-a_n T} y_n(t-T) + \frac{c_n}{a_n} x(t-T)\left(1 - e^{-a_n T}\right) \tag{4.15}$$

$$y_{n+1}(t) = e^{-a_n^* T} y_n^*(t-T) + \frac{c_n^*}{a_n^*} x(t-T)\left(1 - e^{-a_n^* T}\right) \tag{4.16}$$

Examining Equations (4.15) and (4.16) reveals that $y_n(t) + y_{n+1}(t)$ results in a real quantity. Thus the properties of a real system are preserved by ensuring that each complex pole appears along with its complex conjugate and that the residues corresponding to those poles also come in complex conjugate pairs as was done in Chapter 2.

CHAPTER 5

NUMERICAL RESULTS

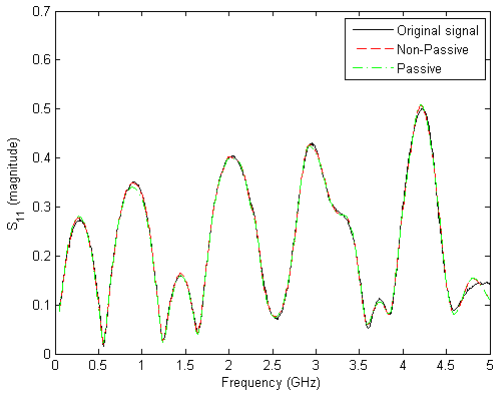
5.1 Overview

In this chapter, various numerical results of using the vector fitting method along with the presented passivity enforcement routine will be demonstrated. In all examples, the data used will be S-parameter data obtained from measurement on a network analyzer. The system in consideration will be treated as a black box, illustrating a black-box macromodeling process, whereby no prior knowledge is needed to generate a model for the system. The results after the vector fitting method and passivity enforcement will be shown and the accuracy of the process will be examined. In all cases, the timing information cited is obtained on a Pentium 4 2.4 GHz processor.

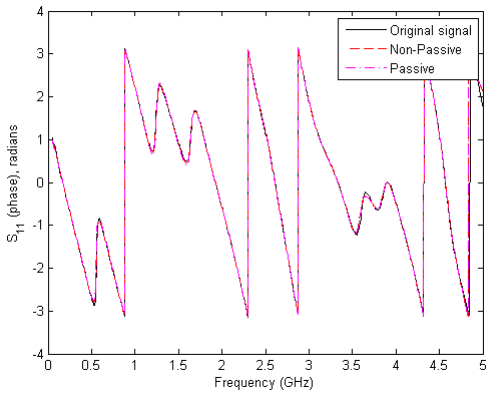
5.2 Example I

The scattering parameters of a 2-port interconnect structure are obtained in the frequency range of 50 MHz – 5 GHz. The vector fitting method is used to obtain a model for the system, fitting all the elements of the 2-port system using the same set of poles with an order of 36. Four vector fitting iterations were used, which took a total of 4.485 s. The passivity of the system was analyzed and the Hamiltonian matrix revealed two passivity violation regions. Passivity enforcement was carried

out which converged after two iterations, lasting an additional 4.656 s. Plots of all the S-parameters are shown in Figures 5.1 – 5.4. Table 5.1 shows the

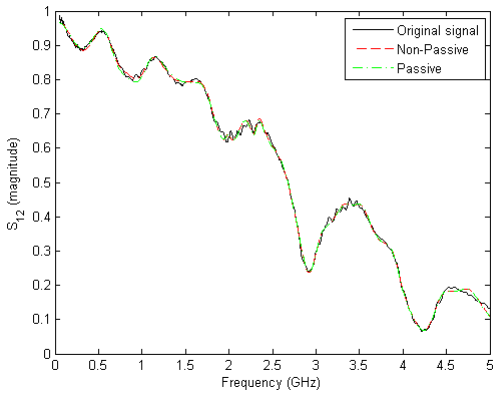


(a) Magnitude

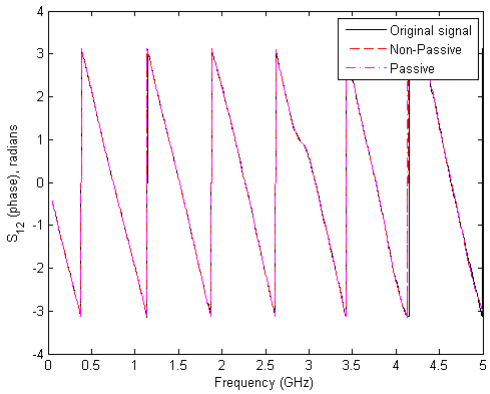


(b) Phase

Figure 5.1: Comparison of S_{11} of the measured data and the model for Example I.



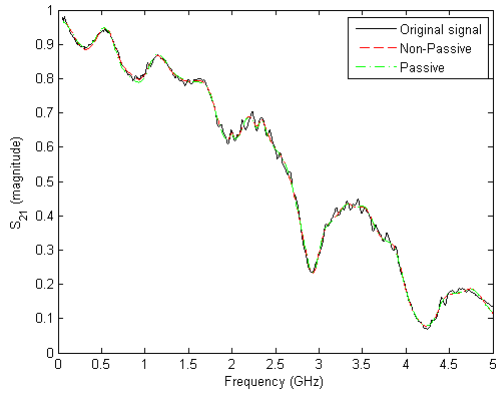
(a) Magnitude



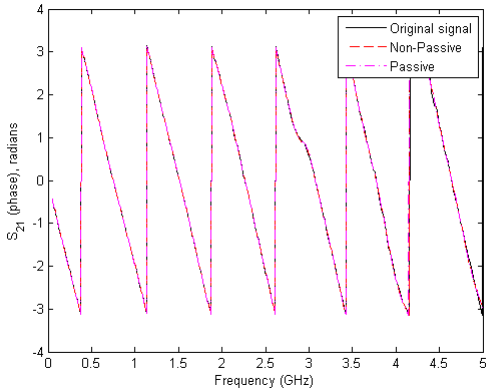
(b) Phase

Figure 5.2: Comparison of S_{12} of the measured data and the model for Example I.

root-mean-square (RMS) error of the model compared to the original signal before and after passivity enforcement. We see that any additional error resulting from the passivity compensation process is minimal and the overall accuracy of the model is retained. A plot of the eigenvalues of the dissipation matrix is shown in Figure 5.5, verifying the passivity compensation process. A time domain simulation is done by utilizing the recursive convolution process with the model developed. A single pulse

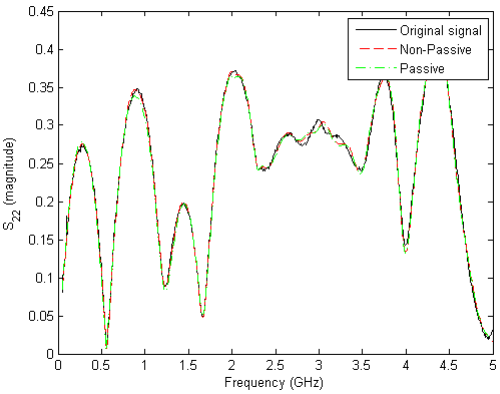


(a) Magnitude

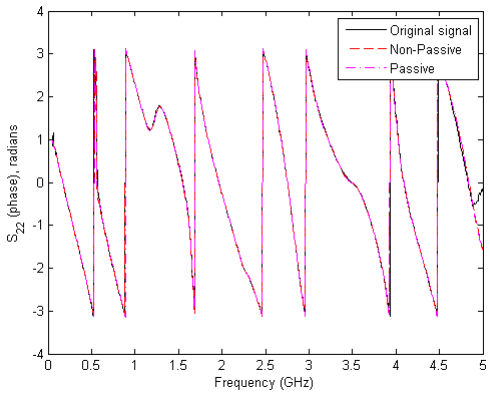


(b) Phase

Figure 5.3: Comparison of S_{21} of the measured data and the model for Example I.



(a) Magnitude



(b) Phase

Figure 5.4: Comparison of S_{22} of the measured data and the model for Example I.

Table 5.1: RMS error of the model compared to the original signal before and after passivity enforcement for Example I.

	S_{11}	S_{12}	S_{21}	S_{22}
From vector fitting	0.008523	0.011352	0.013270	0.008541
After passivity enforcement	0.009831	0.012353	0.014109	0.009795

with rise and fall time of 1 ns and with a pulse width of 12 ns is sent at port 1 and the responses at both ports were evaluated. The result is shown in Figure 5.6.

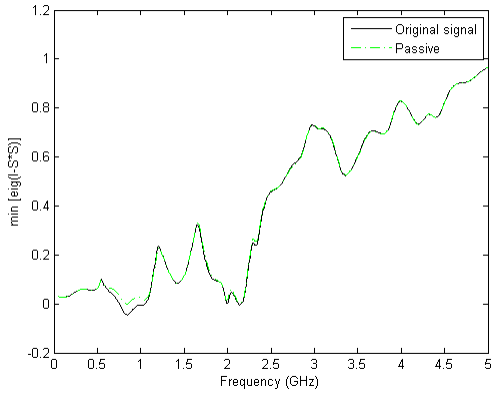


Figure 5.5: Eigenvalues of the dissipation matrix of Example I. Negative values indicate passivity violation.

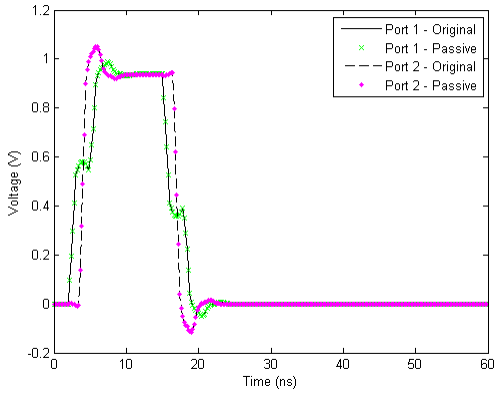
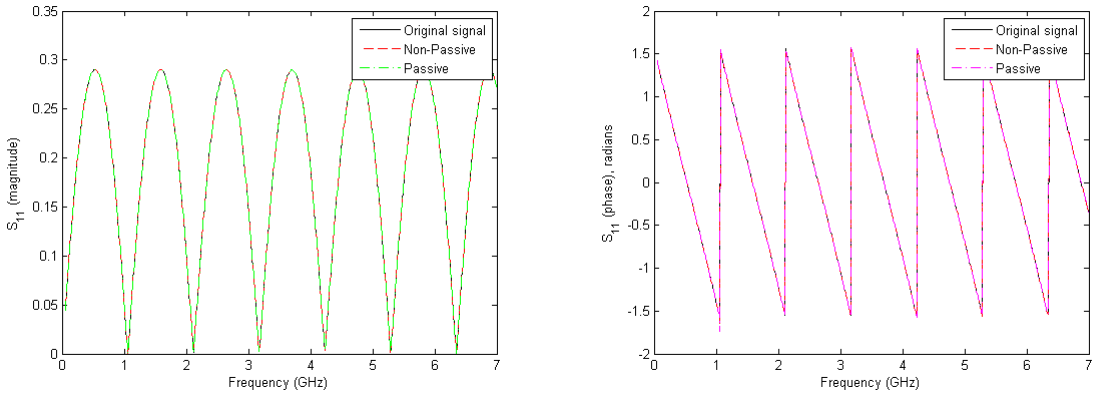


Figure 5.6: Time domain response of Example I.

5.3 Example II

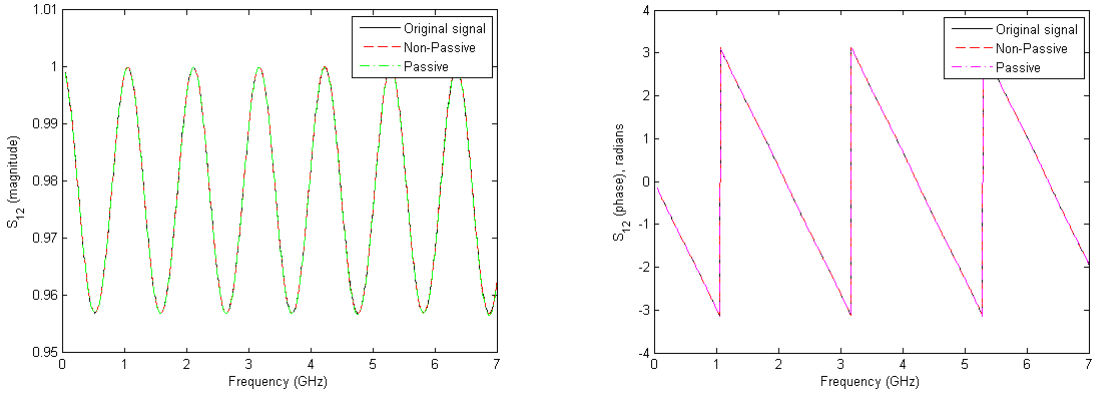
This example illustrates the passivity compensation process when many passivity violation bands are present. The scattering parameters of a 2-port interconnect structure are obtained in the frequency range of 50 MHz – 7 GHz. The vector fitting method is used to obtain a model for the system, fitting all the elements of the 2-port system using the same set of poles with an order of 18. Four vector fitting iterations were used, which took a total of 1.531 s. The passivity of the system was analyzed and the Hamiltonian matrix revealed nine passivity

violation regions. Passivity enforcement was carried out which converged after seven iterations, lasting an additional 52.047 s. Plots of all the S-parameters are shown in Figures 5.7 – 5.10. Table 5.2 shows the root-mean-square (RMS) error of



(a) Magnitude (b) Phase

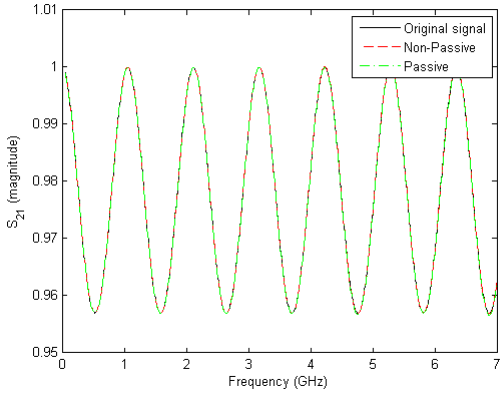
Figure 5.7: Comparison of S_{11} of the measured data and the model for Example II.



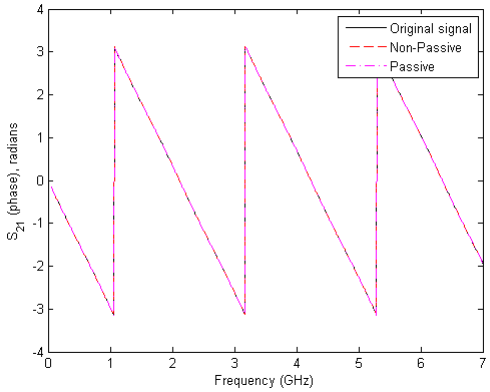
(a) Magnitude (b) Phase

Figure 5.8: Comparison of S_{12} of the measured data and the model for Example II.

the model compared to the original signal before and after passivity enforcement. We see again that the accuracy of the model is retained. A plot of the eigenvalues of the dissipation matrix is shown in Figure 5.11, verifying the passivity compensation process. A time domain simulation is done by utilizing the recursive convolution process with the model developed. A single pulse with rise and fall

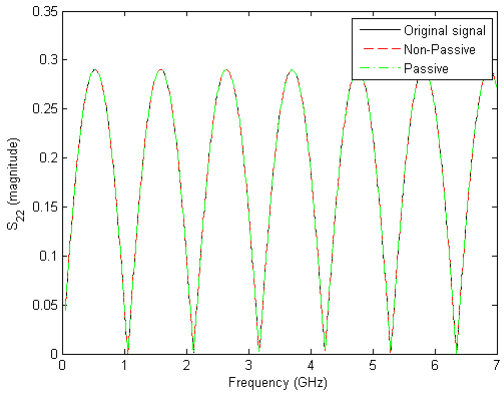


(a) Magnitude

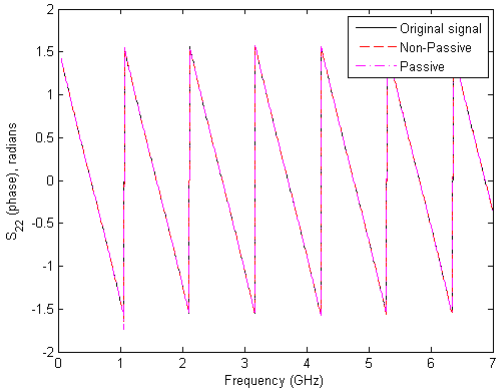


(b) Phase

Figure 5.9: Comparison of S_{21} of the measured data and the model for Example II.



(a) Magnitude



(b) Phase

Figure 5.10: Comparison of S_{22} of the measured data and the model for Example II.

Table 5.2: RMS error of the model compared to the original signal before and after passivity enforcement for Example II.

	S_{11}	S_{12}	S_{21}	S_{22}
From vector fitting	0.0000491	0.0001068	0.0001068	0.0000491
After passivity enforcement	0.0001236	0.0001662	0.0001662	0.0001240

time of 1 ns and with a pulse width of 12 ns is sent at port 1 and the responses at both ports were evaluated. The result is shown in Figure 5.12.

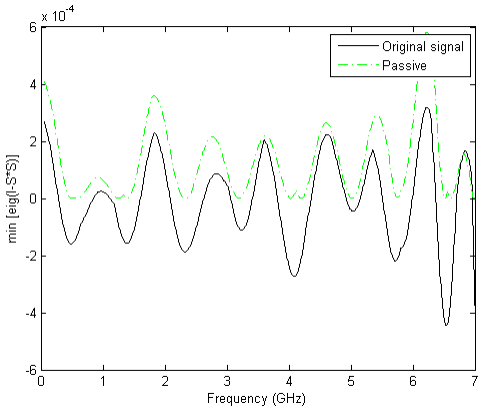


Figure 5.11: Eigenvalues of the dissipation matrix of Example II. Negative values indicate passivity violation.

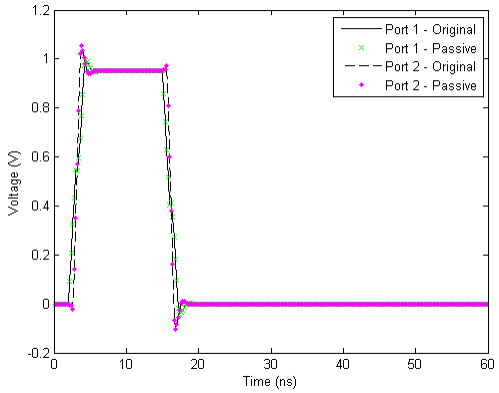
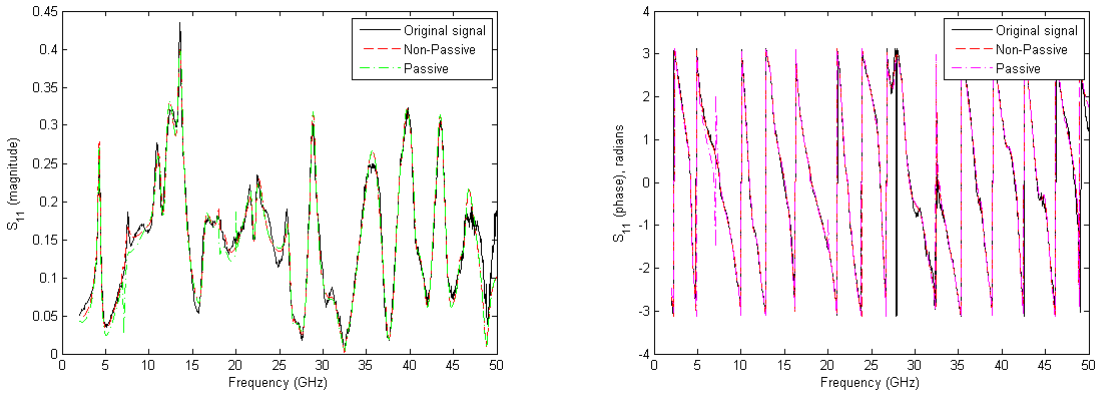


Figure 5.12: Time domain response of Example II.

5.4 Example III

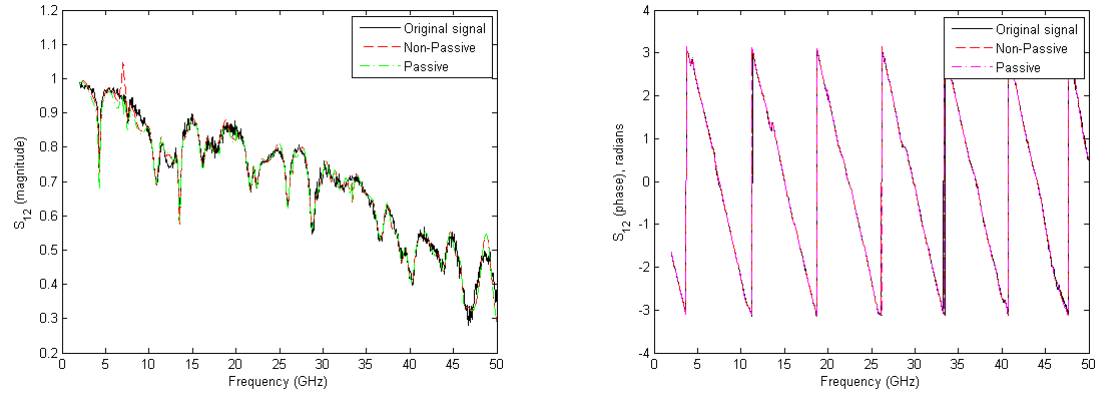
This example illustrates the performance of the macromodeling technique for a high order system when the measurement data is also contaminated with noise. The scattering parameters of a 2-port interconnect structure are obtained in the frequency range of 2 GHz – 50 GHz. The vector fitting method is used to obtain a model for the system, fitting all the elements of the 2-port system using the same set of poles with an order of 90. Four vector fitting iterations were used, which took a total of 24.875 s. The passivity of the system was analyzed and the Hamiltonian

matrix revealed five passivity violation regions. Passivity enforcement was carried out which converged after nine iterations, lasting an additional 22 min and 7.329 s. Plots of all the S-parameters are shown in Figures 5.13 – 5.16. Table 5.3 shows



(a) Magnitude (b) Phase

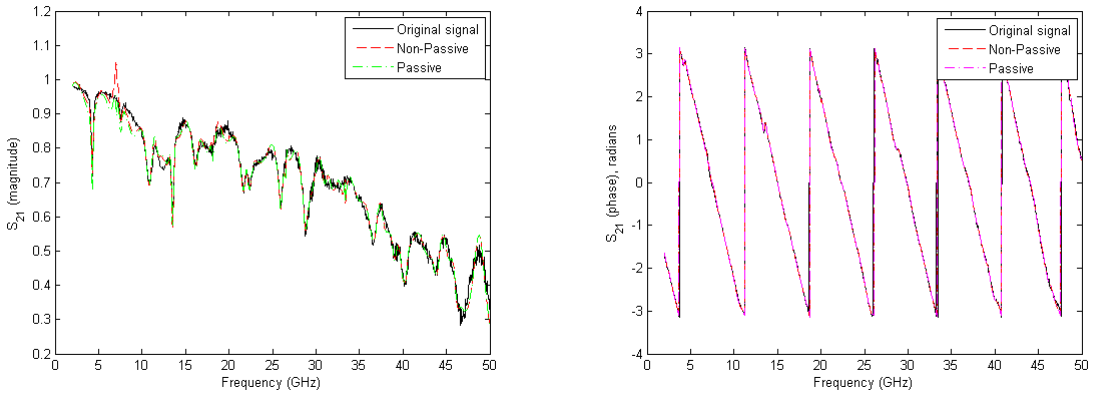
Figure 5.13: Comparison of S_{11} of the measured data and the model for Example III.



(a) Magnitude (b) Phase

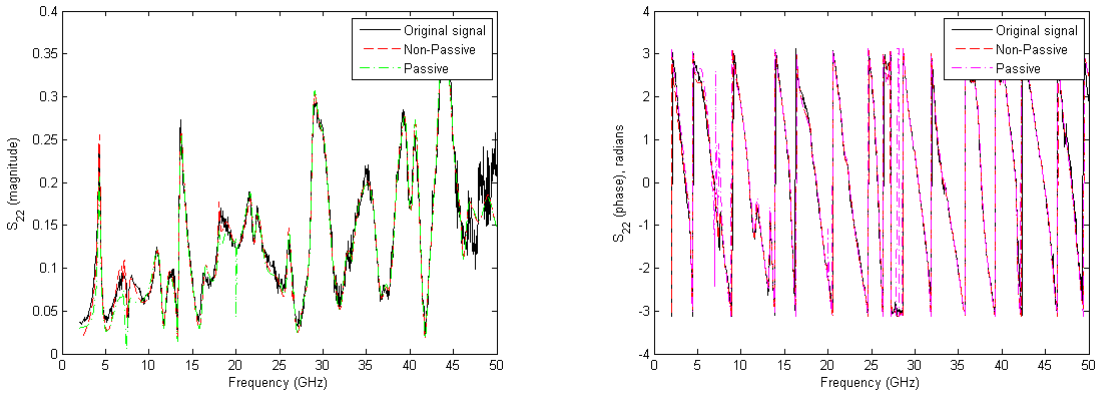
Figure 5.14: Comparison of S_{12} of the measured data and the model for Example III.

the root-mean-square (RMS) error of the model compared to the original signal before and after passivity enforcement. We see again that the accuracy of the model is retained. A plot of the eigenvalues of the dissipation matrix is shown in Figure 5.17, verifying the passivity compensation process. Notice that the largest



(a) Magnitude (b) Phase

Figure 5.15: Comparison of S_{21} of the measured data and the model for Example III.



(a) Magnitude (b) Phase

Figure 5.16: Comparison of S_{22} of the measured data and the model for Example III.

Table 5.3: RMS error of the model compared to the original signal before and after passivity enforcement for Example III.

	S_{11}	S_{12}	S_{21}	S_{22}
From vector fitting	0.01917	0.03068	0.02959	0.02277
After passivity enforcement	0.03079	0.03851	0.03957	0.02929

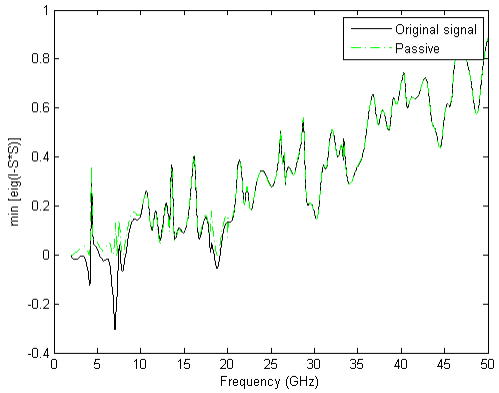


Figure 5.17: Eigenvalues of the dissipation matrix of Example III. Negative values indicate passivity violation.

violation region, which occurs from 5.278 GHz to 7.437 GHz, is due in part to a poor fitting from the vector fitting process over that frequency range. However, after the passivity compensation process, this violation (along with all the others) is removed and the plots of the S-parameters again show a good agreement between the original signal and the model. A time domain simulation is done by utilizing the recursive convolution process with the model developed. A single pulse with rise and fall time of 1 ns and with a pulse width of 12 ns is sent at port 1 and the responses at both ports were evaluated. The result is shown in Figure 5.18.

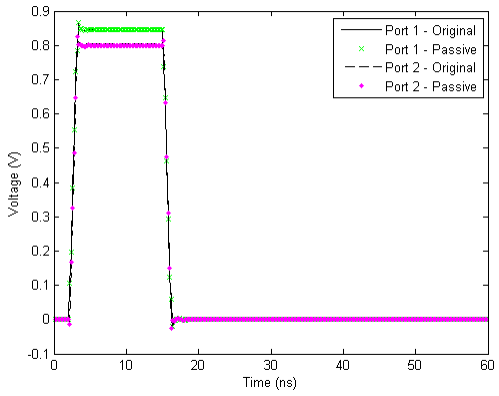


Figure 5.18: Time domain response of Example III.

5.5 Impact of Fast Fitting

In Section 2.7, a method for improving the speed of vector fitting was presented. In order to examine the impact of the presented modification, the three examples presented above were fitted without the modification for fast fitting and the amount of time required in each case was recorded. Table 5.4 shows the comparison between the conventional vector fitting and the fast vector fitting. We see a significant improvement in the latter method.

Table 5.4: Timing comparison between the conventional vector fitting and the fast vector fitting.

	Time required (s)	
	Conventional vector fitting	Fast vector fitting
Example I	264.984	4.485
Example II	63.156	1.531
Example III	762.656	24.875

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In summary, a macromodeling technique utilizing the vector fitting method has been presented. The rational form of the model generated permits the time domain simulation to be done recursively, resulting in a large speedup in computational time. Various issues surrounding generation of macromodels, such as stability and passivity, have been addressed. Stability was easily enforced by reflecting any unstable poles into the left half-plane of the s-domain, while an iterative scheme of passivity compensation relying on residue perturbation was employed. The algorithm was tested on various S-parameter data files obtained from packages and interconnects and the robustness of the method was illustrated. The model was able to closely capture the behavior of the system, and the accuracy was retained through the passivity compensation process. Examples of time domain simulations were also shown to demonstrate the use of the macromodel in typical situations.

6.2 Future Work

Some aspects of the macromodeling technique could be improved. The vector fitting method, for example, could be improved through the use of orthogonal base functions [18], which reduces the sensitivity of the process to the starting poles, thus

resulting in a faster convergence. It has also been shown that performing the vector fitting process in the z-domain [19] results in a lower order and faster convergence as the poles are bounded inside the unit circle in the z-domain as opposed to the entire left half-plane in the s-domain. Given more time, these algorithms could have been utilized to improve the rational function generation process.

The passivity enforcement routine could also be improved. Currently, passivity enforcements are often done as a postprocess, whereby nonpassive models are iteratively perturbed to account for the passivity violations. Even though a significant amount of work was done to preserve the accuracy of the model – through the use of constrained minimization routines – the process could still result in a significant deviation in the model after passivity compensation. This is especially true with models having regions of large passivity violations. A possibly better approach would be to incorporate the passivity enforcement procedure in the vector fitting process, thus removing the need to perform any postprocessing. The passivity assessment and compensation process can also be time-consuming, especially for multiport systems with a high order of approximation. In those cases, the Hamiltonian matrix can be prohibitively large and finding the eigenvalues can be computationally very expensive. It might be possible to develop special eigenvalue solvers that would exploit any properties of the Hamiltonian matrix to speed up the calculations.

All in all, while there has been significant progress in this area over the past few years, there is still much room for improvement for future researchers on the subject.

REFERENCES

[1] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052–1061, July 1999.

[2] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microw. Wireless Compon. Lett.*, vol. 18, no. 6, pp. 383–385, June 2008.

[3] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, "Stability, causality and passivity in electrical interconnect models," *IEEE Trans. Adv. Packag.*, vol. 30, no. 4, pp. 795–808, Nov. 2007.

[4] W. Hendrickx, D. Deschrijver, and T. Dhaene, "Some remarks on the vector fitting iteration," in *Proc. Eur. Conf. Mathematics Industry*, 2006, pp. 134–138.

[5] M. R. Wohlers, *Lumped and Distributed Passive Networks*. New York, NY: Academic, 1969.

[6] D. Saraswat, R. Achar, and M. Nakhla, "Global passivity enforcement algorithm for macromodels of interconnect subnetworks characterized by tabulated data," *IEEE Trans. VLSI Syst.*, vol. 13, no. 7, pp. 819–832, July 2005.

[7] S. Grivet-Talocia and A. Ubolli, "On the generation of large passive macromodels for complex interconnect structures," *IEEE Trans. Adv. Packag.*, vol. 29, no. 1, pp. 39–54, Feb. 2006.

[8] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Hoboken, NJ: Wiley, SIAM Studies in Applied Mathematics, vol. 15, 1994.

[9] D. Saraswat, R. Achar, and M. S. Nakhla, "Fast passivity verification and enforcement via reciprocal systems for interconnects with large order macromodels," *IEEE Trans. VLSI Syst.*, vol. 15, no. 1, pp. 48–59, Jan. 2007.

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

[11] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. New York, NY: Cambridge University Press, 1996.

[12] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 51, no. 9, pp. 1755–1769, Sep. 2004.

[13] K. Zhou and D. J. C. Doyle, *Essentials of Robust Control.* Upper Saddle River, NJ: Prentice Hall, 1998.

[14] C. T. Chen, *Linear System Theory and Design.* New York, NY: Holt, Rinehart and Winston, 1984.

[15] J. Umoto and T. Hara, "A new digital analysis of surge performance in electric power networks utilizing the convolution integral," *Journal of the Institute of Electrical Engineers in Japan*, vol. 91, no. 3, pp. 48–57, 1971.

[16] A. Semlyen and A. Dabuleanu, "Fast and accurate switching transient calculations on transmission lines with ground return using recursive convolutions," *IEEE Trans. Power App. Syst.,* vol. PAS-94, pp. 561–571, Mar./Apr. 1975.

[17] T. Hu, B. Zhong, S. L. Dvorak, and J. L. Prince, "Application of recursive convolution to transient simulation of interconnects using a hybrid phase-pole macromodel simulation," *IEEE Trans. Adv. Packag.,* vol. 27, no. 4, pp. 603–610, 2004.

[18] D. Deschrijver, B. Haegeman, and T. Dhaene, "Orthogonal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 216–225, May 2007.

[19] Y. S. Mekonnen and J. E. Schutt-Ainé, "Broadband macromodeling of sampled frequency data using z-domain vector-fitting method," in *SPI Conference*, 2007, pp. 45–48.