

**IMPROVED HIGH-ORDER RATIONAL APPROXIMATION BY COMBINING  
RATIONAL INTERPOLATION WITH VECTOR FITTING METHOD**

**BY**

**YIDNEKACHEW SINTAYEHU MEKONNEN**

**B.S., Addis Ababa University, 2000**

**THESIS**

**Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2004**

**Urbana, Illinois**

# TABLE OF CONTENTS

<b>LIST OF TABLES .....</b>	<b>vi</b>
<b>LIST OF FIGURES .....</b>	<b>vii</b>
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 Background.....	1
1.2 Overview.....	2
<b>2 VECTOR FITTING METHOD .....</b>	<b>4</b>
2.1 Pole Identification.....	5
2.2 Determination of Zeros.....	10
2.3 Residues and Constant Terms Identification .....	14
2.4 Selection of Starting Poles.....	15
2.4.1 Automatic selection of initial poles .....	16
2.4.2 Rational interpolation.....	17
2.5 Hybrid Method.....	19
2.6 Order Selection and Error Estimation.....	20
<b>3 PASSIVITY ENFORCEMENT.....</b>	<b>21</b>
3.1 General Passivity Requirement of Macromodels .....	21
3.2 For One-Port Network .....	24
3.3 For Multiport Network.....	25
<b>4 NUMERICAL RESULTS .....</b>	<b>26</b>
4.1 Case Study I – Noisy Data Approximation.....	26
4.2 Case Study II – Accuracy Test.....	30
4.3 Case Study III – Convergence Test .....	33
<b>5 SPICE-COMPATIBLE EQUIVALENT CIRCUIT SYNTHESIS .....</b>	<b>37</b>
5.1 Self Admittance Term.....	38
5.2 Transadmittance Term .....	40
<b>6 CONCLUSIONS .....</b>	<b>43</b>
<b>APPENDIX – SOURCE CODE.....</b>	<b>44</b>
<b>REFERENCES.....</b>	<b>50</b>

## LIST OF TABLES

Table 1. Comparison between the methods.....	26
Table 2. Normalized poles in each iteration and the exact poles in long precision format.....	35
Table 3. RMS vs. number of iterations.....	36

## LIST OF FIGURES

Figure 1.	Magnitude of S21: Measured data and the 110 <sup>th</sup> -order rational approximation using standard method. ....	27
Figure 2.	Phase of S21: Measured data and the 110 <sup>th</sup> -order rational approximation using standard method. ....	27
Figure 3.	Magnitude of S21: Measured data and the 162 <sup>nd</sup> -order rational approximation using automatic poles selection method. ....	28
Figure 4.	Phase of S21: Measured data and the 162 <sup>nd</sup> -order rational approximation using automatic poles selection method. ....	28
Figure 5.	Magnitude of S21: Measured data and the 84 <sup>th</sup> -order rational approximation using combined method. ....	29
Figure 6.	Phase of S21: Measured data and the 84 <sup>th</sup> -order rational approximation using combined method. ....	29
Figure 7.	Root-mean squared error vs. number of iteration of the combined method .....	30
Figure 8.	An arbitrary structure layout developed for test in ADS .....	31
Figure 9.	Magnitude response of the structure under consideration, EM simulation result from ADS (original), and rational approximation obtained from proposed method. ....	32
Figure 10.	Phase response of the structure under consideration, EM simulation result from ADS (original), and rational approximation obtained from proposed method. ....	32
Figure 11.	The 3D view of the structure under consideration .....	33
Figure 12.	Pole locations with each iteration and the location of the actual poles .....	34
Figure 13.	Magnitude response of the 8 <sup>th</sup> -order system and the 12 <sup>th</sup> -order rational approximation obtained from proposed method. ....	35
Figure 14.	Phase response of the 8 <sup>th</sup> -order system and the 12 <sup>th</sup> -order rational approximation obtained from proposed method. ....	36

Figure 15. Equivalent circuit realization for a real pole in the self-admittance term .....	38
Figure 16. Equivalent circuit realization for a complex-conjugate pair in the self-admittance term.....	39
Figure 17. Equivalent circuit realization for direct constant and proportional constant in the self-admittance term.....	40
Figure 18. Equivalent circuit realization in transadmittance term of a real pole between ports $i$ and $j$ .....	41
Figure 19. Equivalent circuit realization in transadmittance term of a complex-conjugate pair poles between ports $i$ and $j$ .....	41
Figure 20. Equivalent circuit realization for the $i^{\text{th}}$ port .....	42
Figure 21. Equivalent circuit realization of the dependent sources for the transadmittance terms.....	42

# 1 INTRODUCTION

## 1.1 Background

The electrical modeling of high-speed interconnects becomes a crucial bottleneck for the system designer. Distributed models are needed to accurately simulate on- and off-chip interconnects. Various ways to obtain accurate frequency domain characterization of interconnects include full-wave electromagnetic simulation tools utilizing techniques like the Finite-Difference Time-Domain technique, transmission line simulation environments such as Advanced System Design (HP-ADS), or from measurements. The characterization data can be directly incorporated as S-parameter blocks into a larger network or input into a simulator through the use of a macromodel. Macromodeling generates an equivalent representation of the network.

An accurate simulation of interconnects requires a transient analysis of nonlinear driver and receiver, and frequency-dependent transmission-line systems. Consequently, the most efficient method of analyzing frequency-dependent component is using conventional time-domain circuit simulation, or using circuit simulation technique by generating rational approximation in order to express the interconnect governing equations using ordinary differential equations.

Circuit level simulation of interconnects with their nonlinear drivers and receivers can be costly in terms of central processor unit (CPU) time due primarily to the size of the linear interconnect portion of the model. Considering the number of interactions between interconnect and drivers/receivers (ports) will be small compared to the number of interconnect elements, it is often appropriate to partition the linear interconnect portions from the drivers and receivers. The interconnect can then be macromodeled using an efficient order reduction technique. These

macromodels are then combined with other linear blocks and nonlinear components in a circuit simulation environment.

## 1.2 Overview

Several methods have been proposed to generate rational representations of a distributed system. These methods include the asymptotic waveform evaluation (AWE) and complex frequency hopping methods (CFH) as well as their variants that are based on moment matching at single or multiple points [1]. Lanczos and Arnoldi are other reduced order modeling methods that are based on Krylov subspace techniques [2]. Robust rational interpolations that are based on the Hilbert transform of electrical parameters of passive networks or Kramers-Krönig relations between real and imaginary parts of material properties have also been used to generate rational approximation of frequency-dependent parameters [3]. In [4], a rational interpolation approach is used to approximate the transfer function of linear systems characterized by sampled data by using the three most common orthogonal polynomials: Legendre, Chebyshev of the first kind, and the second kind. The ill-conditioned Vandermonde-like interpolation matrix associated with the ordinary power series is avoided by using these orthogonal polynomials. Clenshaw's recurrence algorithm is applied in transforming the coefficients of the orthogonal polynomials to the ordinary power series. However, the efficiency, accuracy, or stability of these approximations techniques has been compromised in an effort to generate high-order approximations of frequency-dependent parameters over a wide frequency range.

In this thesis, a robust rational approximation technique for macromodel construction is discussed. Recently, the vector-fitting method was developed by Gustavsen and Semlyen in 1998 [5] and [6] to approximate the frequency-dependent parameters of transmission lines using rational functions. In Chapter 2, the vector-fitting method is discussed in detail. The step-by-

step implementation of the method is presented with mathematical proofs on some parts of the method. Some of the convergence issues using the traditional vector-fitting method are discussed. A new hybrid method is proposed by combining the rational interpolation method with the vector-fitting method [7]. Order and error estimation of this method are also discussed.

In Chapter 3, the implementation of passivity enforcement to the resulted macromodel is discussed. Passivity check is one of the major issues in macromodeling. One of the various passivity-checking methods is implemented and discussed in depth.

In Chapter 4, simulation results are presented. The examples focused on verifying the robustness, the accuracy, and the convergence of the hybrid method. An example of passive macromodel is also presented.

In Chapter 5, a Simulation Program with Integrated Circuit Emphasis (SPICE) compatible equivalent circuit representation of a macromodel is discussed. The implementation discussed is only applied for a model characterized by admittance parameter. The SPICE-like representation makes the integration of the macromodel with other linear or nonlinear lumped circuit elements easy. Finally, the conclusion and the outline of the future work to further improve the proposed method are presented in Chapter 6.



## 2 VECTOR FITTING METHOD

The basic idea of macromodeling is to construct a rational function that represents the system. The vector fitting method accurately generates high-order rational functions over a wide frequency range. The method is an iterative technique based on pole-zero relocation technique. It has two main advantages, namely, numerical stability and convergence. It does not suffer from numerical problems that occur when approximating a high-order system over a wide frequency range. It arrives at the optimal solution by solving two linear equations within a few iterations. The frequency response  $f(s)$  of any linear time-invariant passive network can be represented using rational function. For an  $N^{\text{th}}$ -order system response, the rational function  $f(s)$  can be written as

$$f(s) \approx \frac{\sum_{m=0}^M a_m s^m}{\sum_{n=0}^N b_n s^n} \approx \frac{a_0 + a_1 s + a_2 s^2 + \dots + a_M s^M}{b_0 + b_1 s + b_2 s^2 + \dots + b_N s^N} \quad (1)$$

where  $s = j\omega$ ,  $\omega$  is the angular frequency in radians per second.  $M$  and  $N$  are the orders of the numerator and denominator, respectively. For stable real systems, the difference between the numerator order  $M$  and the denominator order  $N$  of the rational function does not exceed unity.  $f(s)$  can be scattering, admittance, or impedance parameters generated from an electromagnetic simulation or from measurements. All coefficients in Equation (1) are unknown, but this nonlinear equation can be written as a linear matrix system equation. Equation (1) can be expressed as modal expression of the form

$$f(s) \approx \sum_{n=1}^N \frac{c_n}{s - a_n} + d + sh, \quad (2)$$

where  $c_n$ ,  $a_n$ ,  $d$  and  $h$  are residues, poles, real direct proportional constant, and real proportional constant, respectively. The residues and poles are either real quantities or in complex-conjugate pairs. In order to determine the above nonlinear equation, the problem can be solved by decomposing into two linear problems. The first linear problem is a pole identification problem in which the poles of the system are identified from the given frequency sample data and an initial starting poles. The second linear problem is to determine the residues and constant terms based on the frequency sample data and the new system poles determined from the first linear problem.

## 2.1 Pole Identification

In this stage, the system poles are fully identified from the given frequency sample data and initial starting poles in the frequency range. The methodology of determining the poles is discussed as follows. First, multiply the given frequency response  $f(s)$  with an unknown augmented function  $\sigma(s)$ . The augmented function  $\sigma(s)$  is defined strictly as

$$\sigma(s) = \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1, \quad (3)$$

where  $\tilde{c}_n$ ,  $\tilde{a}_n$  are unknown constants.  $\sigma(s)$  is intentionally forced to approach unity at a very high frequencies. The whole equation under consideration can be expressed as

$$\begin{bmatrix} \sigma(s)f(s) \\ \sigma(s) \end{bmatrix} \approx \begin{bmatrix} \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \\ \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \end{bmatrix}. \quad (4)$$

In Equation (4) the rational approximation for  $\sigma(s)$  has the same poles as the approximation for  $\sigma(s)f(s)$ . These poles are the initial starting poles. Detailed information

about how to choose these starting poles is given in Section 2.4.2. Multiplying the second row in (4) with  $f(s)$  yields

$$\left( \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \right) \approx \left( \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \right) f(s), \quad (5)$$

$$(\sigma f)_{fit}(s) \approx \sigma_{fit}(s) f(s). \quad (6)$$

Equation (6) is the same representation as Equation (5). Equation (5) reveals that a linear equation results with unknown  $c_n, \tilde{c}_n, d, h$ . This equation can be rewritten for each frequency sample points as follows:

$$\left( \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \right) - \left( \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} \right) f(s) \approx f(s). \quad (7)$$

In matrix form, it becomes

$$\begin{pmatrix} \frac{1}{s_1 - \tilde{a}_1} & \dots & \frac{1}{s_1 - \tilde{a}_N} & \frac{-f(s_1)}{s_1 - \tilde{a}_1} & \dots & \frac{-f(s_1)}{s_1 - \tilde{a}_N} & 1 & s_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{1}{s_k - \tilde{a}_1} & \dots & \frac{1}{s_k - \tilde{a}_N} & \frac{-f(s_k)}{s_k - \tilde{a}_1} & \dots & \frac{-f(s_k)}{s_k - \tilde{a}_N} & 1 & s_k \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \tilde{c}_1 \\ \vdots \\ \tilde{c}_N \\ d \\ h \end{pmatrix} = \begin{pmatrix} f(s_1) \\ \vdots \\ \vdots \\ f(s_k) \end{pmatrix}, \quad (8)$$

where  $k$  is the number of frequency sample points. Since Equation (8) often results in an over-determined linear equation, the least square solution can easily be obtained using one of the standard methods.

Since the system considered is real, all poles must appear as real or complex-conjugate pair poles in order to insure stability of the macromodel. In addition, for every complex-

conjugate pair pole, the residue must appear in complex-conjugate form to insure that the approximate rational function represents a real system because it results a real coefficients. Similarly, a real pole will result a real residue. But solving for the unknown in Equation (8) does not guarantee this condition. The initial starting poles must also satisfy real or complex-conjugate pair form. The above complex matrix is modified to ensure that the residues will be in complex-conjugate form for complex-conjugate as

$$a_n = a'_n + ja''_n, \quad a_n^* = a'_n - ja''_n, \quad (9)$$

$$c_n = c'_n + jc''_n, \quad c_n^* = c'_n - jc''_n, \quad (10)$$

where  $a'_n$  is a real part of the complex pole,  $a''_n$  is the imaginary part of the complex pole, and  $a_n^*$  is the complex-conjugate of  $a_n$ . Similarly,  $c'_n$  is the real part of the residue,  $c''_n$  is the imaginary part of the residue, and  $c_n^*$  is the complex-conjugate of  $c_n$ .

The governing modal equation can be rewritten in more detail form as having  $Q$  real poles and  $L$  complex-conjugate pole pairs as follows:

$$\left( \sum_{i=1}^Q \frac{c_{ir}}{s - \tilde{a}_{ir}} + \sum_{n=1}^L \left( \frac{c_n}{s - \tilde{a}_n} + \frac{c_n^*}{s - \tilde{a}_n^*} \right) + d + sh \right) - \left( \sum_{i=1}^Q \frac{\tilde{c}_{ir}}{s - \tilde{a}_{ir}} + \sum_{n=1}^L \left( \frac{\tilde{c}_n}{s - \tilde{a}_n} + \frac{\tilde{c}_n^*}{s - \tilde{a}_n^*} \right) \right) f(s) \approx f(s). \quad (11)$$

For a real pole, two elements in the matrix have the following form:

$$A_{k,i} = \frac{1}{s_k - \tilde{a}_{ir}}, \quad A_{k,i+2N} = \frac{-f(s_k)}{s_k - \tilde{a}_{ir}} \quad (12)$$

For a complex-conjugate pole pair, four elements of the matrix have the following form:

$$\begin{aligned}
A_{k,n} &= \frac{1}{s_k - \tilde{a}_n} + \frac{1}{s_k - \tilde{a}_n^*}, & A_{k,n+2N} &= \frac{-f(s_k)}{s_k - \tilde{a}_n} + \frac{-f(s_k)}{s_k - \tilde{a}_n^*} \\
A_{k,n+1} &= \frac{j}{s_k - \tilde{a}_n} - \frac{j}{s_k - \tilde{a}_n^*}, & A_{k,n+2N+1} &= \frac{-jf(s_k)}{s_k - \tilde{a}_n} + \frac{jf(s_k)}{s_k - \tilde{a}_n^*}
\end{aligned} \tag{13}$$

In matrix form, it becomes

$$\begin{pmatrix} & & & & 1 & s_1 \\ [R] & [C] & [G] & [H] & \vdots & s_2 \\ & & & & \vdots & \vdots \\ & & & & 1 & s_k \end{pmatrix} x = \begin{pmatrix} f(s_1) \\ f(s_2) \\ \vdots \\ f(s_k) \end{pmatrix}, \tag{14}$$

where  $x$  is the unknowns, and  $[R]$ ,  $[C]$ ,  $[G]$ , and  $[H]$  are matrices of the following form:

$$R = \begin{pmatrix} \frac{1}{s_1 - \tilde{a}_{1r}} & \cdots & \frac{1}{s_1 - \tilde{a}_{qr}} \\ \vdots & \ddots & \vdots \\ \frac{1}{s_k - \tilde{a}_{1r}} & \cdots & \frac{1}{s_k - \tilde{a}_{qr}} \end{pmatrix}, \tag{15}$$

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

$$\begin{aligned}
G &= -f(s_i) \cdot [R] \\
H &= -f(s_i) \cdot [C]
\end{aligned} \tag{16}$$

and the unknown  $x$  has the form:

$$x = \left( c_{1r} \quad \cdots \quad c_{Qr} \quad c_1^r \quad c_1^i \quad \cdots \quad c_L^r \quad c_L^i \right. \\ \left. \tilde{c}_{1r} \quad \cdots \quad \tilde{c}_{Qr} \quad \tilde{c}_1^r \quad \tilde{c}_1^i \quad \cdots \quad \tilde{c}_L^r \quad \tilde{c}_L^i \quad d \quad h \right)^T. \quad (17)$$

The given data has only positive frequencies, and the complex matrix  $A$  can be modified in terms of real quantities by separating the real and imaginary part of the complex matrix as

$$\begin{bmatrix} A' \\ A'' \end{bmatrix} x = \begin{bmatrix} b' \\ b'' \end{bmatrix}, \quad (18)$$

where  $A'$  and  $A''$  are the real and imaginary part of matrix  $A$ , respectively, with the corresponding  $b'$  and  $b''$  vectors for the real and imaginary parts of the frequency response data, respectively. Finally, the complex-conjugate residues are formed from  $x$  as

$$c_m = c_m^r \pm c_m^i \\ \tilde{c}_m = \tilde{c}_m^r \pm \tilde{c}_m^i. \quad (19)$$

Now, the augmented function  $\sigma(s)$  and the product of the augmented function and frequency response  $\sigma(s)f(s)$  can be changed from modal form to fractions of numerator and denominator and rewritten as

$$(\sigma f)_{fit}(s) = \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^N (s - \tilde{a}_n)}, \quad \sigma_{fit}(s) = \frac{\prod_{n=1}^N (s - \tilde{z}_n)}{\prod_{n=1}^N (s - \tilde{a}_n)}. \quad (20)$$

From Equations (20),  $f(s)$  can be obtained by dividing one from the other as

$$f(s) = \frac{(\sigma f)_{fit}(s)}{\sigma_{fit}(s)} = \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^N (s - \tilde{a}_n)} * \frac{\prod_{n=1}^N (s - \tilde{a}_n)}{\prod_{n=1}^N (s - \tilde{z}_n)} = \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^N (s - \tilde{z}_n)}. \quad (21)$$

From Equation (21), the poles of  $f(s)$  become the zeros of  $\sigma(s)$ . The initial starting poles are canceled out in this division process because both arguments have developed to have the same poles. Therefore, if the zeros of  $\sigma(s)$  are determined, the system poles can be directly found. The next section describes how to determine the system zeros to systems described in state space representation.

## 2.2 Determination of Zeros

A linear time invariant system can be described by a set of equations of the form

$$\begin{aligned}\bar{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\quad (22)$$

where  $\bar{x} := dx/dt$  and  $A, B, C$ , and  $D$  are constant matrices for time invariant system.  $u(t)$ ,  $x(t)$ , and  $y(t)$  are the input, state, and output of the linear system, respectively. For a linear system that has  $p$  inputs and  $q$  outputs,  $u(t)$  is a  $p \times 1$  vector and  $y(t)$  is a  $q \times 1$  vector. If the linear system has  $n$  state variables, then  $x(t)$  is a  $n \times 1$  vector. Therefore, the matrices  $A$ ,  $B$ ,  $C$ , and  $D$  must be  $n \times n$ ,  $n \times p$ ,  $q \times n$ , and  $q \times p$  matrices, respectively.

Taking the Laplace transform of Equation (22) and  $x(0)$  as initial condition of the state,

$$\begin{aligned}sX(s) - x(0) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s),\end{aligned}\quad (23)$$

where  $X(s)$ ,  $U(s)$ , and  $Y(s)$  are the Laplace transform of  $x(t)$ ,  $u(t)$ , and  $y(t)$ , respectively.

Equation (23) can be rewritten as

$$\begin{aligned}(sI - A)X(s) - BU(s) &= x(0) \\ CX(s) + DU(s) &= Y(s),\end{aligned}\quad (24)$$

where  $I$  is an identity matrix of the same size as  $A$ .

Writing the equation in matrix form,

$$\begin{pmatrix} (sI - A) & -B \\ C & D \end{pmatrix} \begin{pmatrix} X(s) \\ U(s) \end{pmatrix} = \begin{pmatrix} x(0) \\ Y(s) \end{pmatrix} \quad (25)$$

$$P(s) \begin{pmatrix} X(s) \\ U(s) \end{pmatrix} = \begin{pmatrix} x(0) \\ Y(s) \end{pmatrix},$$

where  $P(s)$  is called the system matrix. The transfer function of the system,  $H(s)$ , which is the transfer function of the system, can be derived from the state equation of the system. Again, applying Laplace transform to Equation (22) and rearranging the equations yields

$$\begin{aligned} sX(s) &= AX(s) + BU(s) \\ X(s) &= (sI - A)^{-1}BU(s) \\ Y(s) &= CX(s) + DU(s) \\ Y(s) &= [C(sI - A)^{-1}B + D]U(s) \\ H(s) &= \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D. \end{aligned} \quad (26)$$

The transfer function response representation in Equation (26) will be used to drive the zeros of the system defined by state space representation. For a minimal realization of a system defined by Equation (22), a system zero is defined as a number that makes the output of the system zero for a time greater than or equal to zero. For such system, a necessary and sufficient condition for an arbitrary input of the form  $u(t) = ge^{\lambda t}l(t)$ , where  $l(t)$  is a unit step function and  $g$  and  $\lambda$  are arbitrary real constants, to yield a state  $x(t) = x_0e^{\lambda t}l(t)$  and output  $y(t) = 0$  for  $t \geq 0$ , is when Equation (25) is determined at  $t = 0$  with the input  $u(t)$  and gives

$$P(\lambda) \begin{pmatrix} x_0 \\ g \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (27)$$

assuming the initial condition of the state  $x(0)$  is zero.



Therefore, a number  $\lambda \in \mathbb{C}$ , where  $\mathbb{C}$  is complex space, is a *zero* of system in Equation (22) if and only if there exists  $x_0 \neq 0$  where  $x_0 \in \mathbb{C}^n$  and  $g \in \mathbb{C}^m$  such that the triple  $\lambda, x_0, g$  satisfies Equation (27).

The following theorem with its proof provides the construction of the system zeros based on the definitions given above. The detailed theorem can be found in [8] - [11].

**Theorem:**

A number  $\lambda \in \mathbb{C}$  is a zero of a minimal system of Equation (22) if

- $\lambda$  is the eigenvalue of  $(A - BD^+C)$  and there exists an eigenvector  $x_0$  of  $(A - BD^+C)$  corresponding to  $\lambda$ .
- $x_0$  must be in the null space of  $(I_r - DD^+)C$ , then  $\lambda, x_0, g$  must satisfy Equation (27).

The matrix  $D^+$  is called the pseudo-inverse or the Moore-Penrose generalized inverse of D [12].

$D^+$  is defined as  $D^+ = (D^T D)^{-1} D^T$ , where  $D^T$  is the transpose of D.  $I_r$  is an identity matrix.

**Proof:**

Let us take an input  $g = -D^+ C x_0$ . The eigen value equation can be written as

$$\begin{aligned} \lambda x_0 &= (A - BD^+C)x_0 \\ (\lambda I - A)x_0 &= -BD^+ C x_0 \\ (\lambda I - A)x_0 &= Bg \end{aligned} \tag{28}$$

If  $x_0$  is in the null space of  $(I_r - DD^+)C$ , then

$$\begin{aligned} (I_r - DD^+)C x_0 &= 0 \\ C x_0 - DD^+ C x_0 &= 0 \\ C x_0 + Dg &= 0 \end{aligned}$$

$$\begin{aligned} \begin{pmatrix} (\lambda I - A) & -B \\ C & D \end{pmatrix} \begin{pmatrix} x_0 \\ g \end{pmatrix} &= 0 \\ P(\lambda) \begin{pmatrix} x_0 \\ g \end{pmatrix} &= 0. \end{aligned} \quad (29)$$

which shows that  $\lambda$  is the zeros of the system defined by Equation (22) because the last result is the same as Equation (27). ■

The next step is to construct a state space model for the augmented function  $\sigma(s)$ . The poles of the system response  $f(s)$  are the zeros of the augmented function  $\sigma(s)$ . Therefore, the modal canonical form of  $\sigma(s)$  can be written as

$$\begin{aligned} \bar{x}(t) &= \begin{pmatrix} a_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & a_N \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} u(t) = Ax(t) + Bu(t) \\ y(t) &= [c_1 \cdots c_N] x(t) + [d] u(t) = Cx(t) + Du(t) \end{aligned} \quad (30)$$

where  $A$  is a diagonal matrix whose elements are the poles of  $\sigma(s)$ , which is the initial poles,  $B$  is a column vector whose elements are 1,  $C$  is a row vector whose elements are the residues of the  $\sigma(s)$ , and  $D$  is 1. In order to find the zeros of  $\sigma(s)$ , according to the theorem, the eigenvalue of the matrix  $(A - BD^+C)$  must be determined. Since  $D = 1$ ,

$$D^+ = (D^T D)^{-1} D^T = 1. \quad (31)$$

Therefore, the zeros of  $\sigma(s)$  are the eigenvalue of the matrix  $(A - BC)$ , which are equally the poles of the frequency response  $f(s)$ .

By using this method the poles of the system are identified. At this point, unstable poles, poles with a positive real part, can be deleted or modified to be stable poles with negative real part. This insures stability of the approximation process. In this pole identification process, initial poles are assigned at first and iteration can be used to refine the pole locations. These

initial poles must be stable poles. One can set a maximum root mean square error to end up the iteration process. There are ways to select a good starting poles, which are discussed in Section 2.4.

## 2.3 Residues and Constant Terms Identification

Once the system poles are identified, the unknown residues and the constant term can be directly be calculated from

$$f(s) = \sum_{n=1}^N \frac{c_n}{s - a_n} + d + sh \quad (32)$$

Now, the system poles  $a_n$  are already determined from the pole identification process in Section 2.2. The only remaining unknowns are  $c_n, d$ , and  $h$ . The system poles are either real or complex-conjugate form. Therefore, the corresponding residue values must be the same form as the system poles. The governing modal equation  $f(s)$  can be rewritten again in more detail as having  $Q$  real poles  $a_{ir}$  and  $L$  complex-conjugate pole pairs  $a_n$

$$f(s) = \sum_{i=1}^Q \frac{c_{ir}}{s - a_{ir}} + \sum_{n=1}^L \left( \frac{c_n}{s - a_n} + \frac{c_n^*}{s - a_n^*} \right) + d + sh. \quad (33)$$

In matrix form, it becomes

$$\begin{pmatrix} & & 1 & s_1 \\ [R] & [C] & \vdots & s_2 \\ & & \vdots & \vdots \\ & & 1 & s_k \end{pmatrix} x = \begin{pmatrix} f(s_1) \\ f(s_2) \\ \vdots \\ f(s_k) \end{pmatrix}. \quad (34)$$

[R] and [C] are matrices defined in Equation (15) and the unknown  $x$  have the form

$$x = \left( c_{1r} \quad \cdots \quad c_{Qr} \quad c_1^r \quad c_1^i \quad \cdots \quad c_L^r \quad c_L^i \quad d \quad h \right)^T. \quad (35)$$

Again, the complex matrix is modified in terms of real quantities by separating the real and imaginary part of the complex matrix as shown in Equation (18).

Finally, the complex-conjugate residues are formed from  $x$  as:

$$c_m = c_m^r \pm c_m^i. \quad (36)$$

Therefore, by following the above method, all the unknown residues and constant terms are obtained. The residues obtained guarantee complex-conjugate form for complex-conjugate poles.

## 2.4 Selection of Starting Poles

The vector fitting method needs initial poles to approximate the given data in rational function. However, the accuracy and convergence of the vector fitting method depends strongly on the number and the location of the starting poles. The standard methods for selecting the orders and initial poles are in most part heuristics [5] and [6]. So far, there has not been a study on the effects of the initial poles on the convergence of the vector fitting method. A possible way of selecting a set of starting poles was introduced in [5] and [6]. In [5], poles are selected in logarithmic or linear scale in the frequency range. But this way of selecting poles does not guarantee capturing the behavior of the data at the maximums and minimums. The number of the starting poles determines the order of approximation. The initial pole assignment proposed in [5] is to start with poles covering the approximation frequency range with the relationship between the imaginary and real poles

$$p_i = -\alpha \pm j\beta, \quad \text{where} \quad \alpha = \beta/100$$

In this thesis, new ways of selecting starting poles are presented.

### 2.4.1 Automatic selection of initial poles

One possible way of selecting the initial poles is performed by automatically parsing the data within the frequency range. This method is based on finding frequency points at which the local minimums or maximums of the given data occur. These frequency points are used to create the starting poles. The advantage of this method is that it does not require guessing the initial poles by users. One of the disadvantages of this method is that it generates a high-order approximation and can easily be affected by noise spikes in the given data. The fitting curves obtained from such a method of selecting the initial pole do not often give a smooth curve. The resulting starting poles from this method are all in conjugate complex form.

The procedure is summarized as follows:

1. Normalize the whole frequency range using the maximum frequency.
2. Starting from the beginning of the frequency range to the end, find the local minimum and maximum points of the data.
3. Store the frequency point at which these extreme occurs. Include the starting and the ending points of the frequency range.
4. Set the imaginary part of the pole,  $\beta = 2 * \pi * \text{extreme frequency points}$ .
5. Create the initial poles,  $p_i = -\alpha \pm j\beta$ , where  $\alpha = \beta/100$ .

In this thesis, a new hybrid method of selecting initial poles for the vector fitting method is proposed. The set of initial poles is obtained by performing rational interpolation over partitioned frequency ranges. The method takes the advantage of the robustness of the rational interpolation in the extracting stable poles by partitioning the frequency range. These poles improve the accuracy and convergence of the vector fitting method. The new proposed method uses as initial poles the set of poles obtained by performing rational interpolation over a partitioned frequency range using the method developed in [3] and described below.

## 2.4.2 Rational interpolation

The starting poles can be obtained by partitioning the frequency range and by finding the poles of each partition region by rational approximation using the real part of the partition [3].

The whole frequency range can be partitioned equally or in certain other criteria.

Any network parameter of passive system can be approximated as a ratio of polynomials in the frequency domain as Equation (1). The real part of a network function can be specified as the even part of  $f(s)$  replacing  $-s^2$  with  $\omega^2$ . The real part of the original function is fitted with the real rational polynomial function of squared variable as

$$\text{Re}(f_{p,q}(s)) = \frac{a_0 + a_1 s^2 + a_2 s^4 + \dots + a_p s^{2p}}{1 + b_1 s^2 + b_2 s^4 + \dots + b_q s^{2q}}. \quad (37)$$

The poles of the even function of  $f(s)$  are those of both  $f(s)$  and  $f(-s)$ , those belonging to  $f(s)$  lie in the left half-plane. Thus the denominator coefficients of  $f(s)$  in Equation (1) can be obtained from Equation (37). The following system equations results from matching the real parts of the original function with Equation (37) at the set of frequencies, as in

$$\begin{bmatrix} 1 & \omega_0^2 & \omega_0^4 & \dots & \omega_0^{2p} & -\omega_0^2 y_0^r & -\omega_0^4 y_0^r & \dots & -\omega_0^{2p} y_0^r \\ 1 & \omega_1^2 & \omega_1^4 & \dots & \omega_1^{2p} & -\omega_1^2 y_1^r & -\omega_1^4 y_1^r & \dots & -\omega_1^{2p} y_1^r \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega_{k-1}^2 & \omega_{k-1}^4 & \dots & \omega_{k-1}^{2p} & -\omega_{k-1}^2 y_{k-1}^r & -\omega_{k-1}^4 y_{k-1}^r & \dots & -\omega_{k-1}^{2p} y_{k-1}^r \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_p \\ p_1 \\ p_2 \\ \vdots \\ p_q \end{bmatrix} = \begin{bmatrix} y_0^r \\ y_1^r \\ \vdots \\ y_{k-1}^r \end{bmatrix}, \quad (38)$$

where the subscript “ $r$ ” indicates the real part of the complex data given.  $k$  is the number of frequency data points given. Once the coefficients of the denominator polynomial are obtained from Equation (38), the roots of the denominator are calculated from

$$1 + p_1\omega^2 + p_2\omega^4 + \dots + p_q\omega^{2q} = 0 \quad (39)$$

The poles are assumed to be distinct. Repeated poles are not likely because of the nature of the problem and the fact that the poles are computed numerically. If repeated poles are obtained, they can be modified slightly to make them distinct. Factoring the denominator and taking only the left half-plane roots, the partial fraction expansion of the transfer function is constructed. Stability of the poles is guaranteed, because the polynomial roots are determined in terms of the square poles. The purely imaginary single poles on the imaginary axis are rejected as spurious because the rational function has double roots. The remaining poles are used to formulate the stable initial poles for the vector-fitting method.

The transposed Vandermonde-like interpolation matrix in Equation (38) can be artificially ill-conditioned; the condition number of the matrix can be improved by normalizing the frequency range to unity. Therefore, the whole frequency range under consideration is normalized by the maximum frequency under consideration. The transposed Vandermonde-like matrix in Equation (38), even for a moderate order, is notoriously ill-conditioned in the sense that the entries along each row are simple powers of the corresponding frequency values. If the span of the frequencies being considered is large, then the magnitudes of the entries on some of the rows will be much larger than those in rows corresponding to low frequency values. For this reason, the frequency range under consideration is partitioned into smaller frequency range, and by doing this the risk of solving an ill-conditioned matrix is minimized. The frequency range can be divided equally. One possible way of partitioning the frequency range is by determining the number of extremes in the partition in such a way that each partition has equal extremes and comparable frequency range. A sufficient order of approximation should be set for this method to work. One possible way of guessing good order of approximation for a partition is again by determining the number of extremes in the partition. These poles are then used as starting poles

for the vector-fitting algorithm. The resulting estimate poles at each iteration are used as the starting pole for the next iteration.

## 2.5 Hybrid Method

The proposed method [7] combines the robustness and efficiency of the rational interpolation and the numerical stability and fast convergence of the vector-fitting method. The rational interpolation method is first used to generate accurate poles of the partition frequency range. Then the poles are used in the vector-fitting method to generate a high-order rational function valid over wide frequency range taking advantage of numerical stability and fast convergence properties of the method.

The procedure is summarized as follows:

1. Normalize the entire frequency range with the maximum frequency.
2. Partition the frequency range into subsections and select the order of approximation to each partition.
3. For each partition perform rational interpolation.
  - 3.1. Construct the real part of the matrix and solve for denominator coefficients in Equation (38).
  - 3.2. Factor the denominator polynomial and filter the stable poles.
4. Collect all the poles from each partition and construct the initial guess poles, and perform vector fitting.
  - 4.1. Set the least square solution of the augmented functions at the frequency points using the initial poles.
  - 4.2. Solve for new set of poles.
  - 4.3. Find the residues and constants using the new set of poles.



4.4. Calculate the root-mean square (RMS). If the error is not acceptable, repeat step 4 using the new set of poles as initial guesses. If the error is acceptable, go to passivity checking and enforcement.

The proposed combined method provides a more accurate and fast convergence rational approximation.

## **2.6 Order Selection and Error Estimation**

An important step in any order reduction scheme is order selection. Unfortunately, in most methods predicting the order of approximation is very difficult. The approach is usually to start with a low order and increase the order until a convergence criterion is satisfied. Ideally the approximate response is compared to the exact response at the desired frequency points. If the error is not acceptable, the order can be increased until the desired root-mean square error is achieved. Many frequency-sampling points should be taken in order to get a better approximation in continuous frequency.

## 3 PASSIVITY ENFORCEMENT

### 3.1 General Passivity Requirement of Macromodels

For transient time-domain analysis, a passive macromodel needs to satisfy three conditions:

1. The macromodel has to have real coefficients in its rational approximation function.
2. The macromodel should have stable poles.
3. The macromodel must be passive.

The passivity condition requires that the circuit does not create energy. The passivity conditions for a multiport macromodel are:

1.  $[H(s^*)] = [H^*(s)]$  for all  $s$ , where  $*$  is the complex-conjugate operator.
2.  $[H(s)]$  is a positive real matrix. Thus, the product  $z^{*T} [H^T(s^*) + H(s)] z \geq 0$  for all  $s$  with  $Re(s) > 0$  and any arbitrary vector  $z$ .

These conditions translate into the following constraints:

1.  $[H(s)]$  does not contain poles on the right half of the  $s$ -plane.
2.  $[H(s)]$  does not have multiple poles on the imaginary axis of the  $s$ -plane.
3. The coefficients of  $[H(s)]$  are real.
4. The real part of  $[H(s)]$  must be positive semidefinite for all frequencies. This implies that the eigenvalues of the real part of  $[H(s)]$  are positive or zero for all frequencies.

The first and the second constraints are included as part of stability condition while the third condition is equivalent to simply satisfying the real coefficients of the system. Therefore, the passivity condition needs to ensure real coefficients and satisfy the stability condition.

Furthermore, all the eigenvalues of the real part of the rational function matrix must be semipositive.

The next important step is how to satisfy these conditions and construct a passive and stable macromodel. The constraint for real coefficients can be satisfied by construction. In the construction of the passive macromodel, if all poles and residues are constructed in such a way that they appear as a real or complex-conjugate form, it is guaranteed that the whole rational function matrix will have real coefficients. Therefore, poles and residues must appear as real or as conjugate pairs. The stability conditions can be satisfied by ensuring the poles of the rational function matrix lie on the left half of the complex  $s$ -plane and eradicating those poles that do not satisfy this condition. The last passivity constraint must be satisfied and guaranteed over the infinite frequency bandwidth. Therefore, the last constraint should be developed independent of frequency bandwidth.

Since poles are characteristic frequencies of the system, common poles should be used to represent the multiport passive system when approximating by rational function matrix  $[H(s)]$ . Different approaches are available to enforce passivity. Some of these approaches are frequency range dependent. In [13], an eigenvalue approach has been discussed, which enforces passivity of the macromodel by directly compensating the poles and residues of the rational function using linearization and constrained minimization through Quadratic Programming. However, this method that is based on searching the frequency band of violation, is computationally expensive and uses discrete, band-limited frequency samples for enforcing passivity of the macromodel. Hence, the generated macromodel can still violate passivity over a continuous frequency and outside the band-limited frequency response since the macromodel is tested at discrete frequency samples.

Recently, the rational function matrix  $[H(s)]$  without common poles has been used for representing the passive macromodel for multiport circuits, and a compensation method has been proposed [14]. The compensation method can detect the frequency band violating passivity using the associated Hamiltonian matrix. In [15] and [16], a different approach is used to assure passive macromodel. In this thesis, the method developed in [15] is used and discussed.

The roots of the polynomial in Equation (1) are either real or complex-conjugate pairs. It is assumed that the root is simple or nonoverlapping. The pole-residue form can be rewritten as

$$H(s) = \sum_{n=1}^{CPN/2} \left( \frac{\alpha_n + j\beta_n}{s - p_{nr} - jp_{ni}} + \frac{\alpha_n - j\beta_n}{s - p_{nr} + jp_{ni}} \right) + \sum_{m=1}^{RPN} \frac{\gamma_m}{s - p_{mr}} + d + sh, \quad (40)$$

where the coefficients  $p_{nr}, \gamma_m, p_{nr}, p_{ni}, \alpha_n, \beta_n, d, h$  are real values.  $CPN$  and  $RPN$  represent the number of complex-conjugate poles and real poles, respectively.  $p_{nr} \pm jp_{ni}$  are the complex-conjugate poles with the corresponding residues  $\alpha_n \pm j\beta_n$ .  $p_{mr}$  is a real pole with a residue  $\gamma_m$ .  $d$  is a direct constant term independent of frequency.  $h$  is a proportional constant term dependent on frequency. By combining the complex-conjugate poles, Equation (40) can be rewritten as

$$H(s) = \sum_{n=1}^{CPN/2} \frac{2\alpha_n(s - p_{nr}) - 2\beta_n p_{ni}}{(s - p_{nr})^2 + p_{ni}^2} + \sum_{m=1}^{RPN} \frac{\gamma_m}{s - p_{mr}} + d + sh. \quad (41)$$

For a distributed multiport network with common system poles, Equation (41) can be generalized as

$$[H(s)] = \sum_{n=1}^{CPN/2} \frac{2[\alpha_n](s - p_{nr}) - 2[\beta_n]p_{ni}}{(s - p_{nr})^2 + p_{ni}^2} + \sum_{m=1}^{RPN} \frac{[\gamma_m]}{s - p_{mr}} + [d] + [h]s. \quad (42)$$

where the matrices  $[\alpha_n], [\beta_n], [\gamma_m], [d], [h]$  are  $P \times P$  residue and constant matrices for a  $P$ -port network.

The macromodel constructed needs to satisfy stability and passivity for a linear time-invariant passive system. The stability requires that all poles must lie in the left half of the  $s$ -plane. This can be achieved easily during the construction of the macromodel by applying  $p_{nr} < 0$  and  $p_{mr} < 0$  in Equations (41) and (42). The passivity requires that a passive circuit does not create energy. Nonpassive macromodels combined with a stable circuit can generate an unstable time-domain response. Based on the maximum modulus theorem [17], the passivity condition for a one-port network can be written as

$$\operatorname{Re}\{H(s = j\omega)\} \geq 0 \quad \forall \omega. \quad (43)$$

The one port network rational function  $H(s)$  can be separated into real and imaginary parts and results as follows:

$$H(s = j\omega) = Hr(j\omega) + jHi(j\omega) \quad (44)$$

$$Hr(j\omega) = \sum_{n=1}^{CPN/2} \frac{2\{\omega^2(-\alpha_n p_{nr} + \beta_n p_{ni}) + (p_{nr}^2 + p_{ni}^2)(-\alpha_n p_{nr} - \beta_n p_{ni})\}}{(p_{nr}^2 + p_{ni}^2 - \omega^2)^2 + (2p_{nr}\omega)^2} + \sum_{m=1}^{RPN} \frac{-\gamma_m p_{mr}}{p_{mr}^2 + \omega^2} + d \quad (45)$$

$$Hi(j\omega) = \sum_{n=1}^{CPN/2} \frac{2\omega\{\alpha_n(-p_{nr}^2 + p_{ni}^2 - \omega^2) - 2\beta_n p_{ni} p_{nr}\}}{(p_{nr}^2 + p_{ni}^2 - \omega^2)^2 + (2p_{nr}\omega)^2} + \sum_{m=1}^{RPN} \frac{-\gamma_m \omega}{p_{mr}^2 + \omega^2} + \omega h. \quad (46)$$

### 3.2 For One-Port Network

The passivity of each subnetwork in Equations (41) and (42) can be satisfied using the analytical formulas for the passivity. Applying the passive constraint for a one-port network, Equation (43), the following constraints can be derived from Equation (45):

$$\begin{aligned} -\alpha_n p_{nr} \pm \beta_n p_{ni} &\geq 0 \\ \gamma_m &\geq 0 \\ d &\geq 0. \end{aligned} \quad (47)$$

Note that due to the stability constraint,  $p_{nr} < 0$  and  $p_{mr} < 0$ . It is important to note that the summation of a passive subnetwork is always results in a passive network. The rational function  $H(s)$  is regarded as a summation of subnetworks consisting of complex-conjugate poles and real poles with corresponding residues. Therefore, if every subnetwork in Equations (41) and (42) satisfies the passivity condition, the rational function  $H(s)$  satisfies the passivity condition as well.

### 3.3 For Multiport Network

For a multiport network, the rational function  $[H(s)]$  in Equation (42) has to be positive semidefinite at all frequencies according to the passivity constraints discussed before. Using the property of positive semidefiniteness, the one-port passivity formulae in Equation (47) can be extended to multiport passivity formulae as

$$\begin{aligned}
 & \text{Eigenvalues of } [\gamma_m] \geq 0, \\
 & \text{Eigenvalues of } [-\alpha_n p_{nr} \pm \beta_n p_{ni}] \geq 0, \\
 & \text{Eigenvalues of } [d] \geq 0.
 \end{aligned} \tag{48}$$

From Equation (48), the multiport passivity formulae only depend on the poles and residues matrices, which are independent of frequency. Hence, the passivity of the circuit is satisfied over an infinite frequency. The multiport passive formulae are only enforced on each subnetwork of  $[H(s)]$ , and there is no relationship for passivity between sub networks except that they contribute to the overall frequency response of the macromodel. This adds simplicity to the computation. For compensating negative eigenvalues in Equation (48), there are two free matrix variables  $[\alpha_n]$  and  $[\beta_n]$  related to two free variables of complex-conjugate poles  $p_{nr} \pm jp_{ni}$ , a free matrix variable  $[\gamma_m]$  related to a real pole  $p_{mr}$ , and a free matrix variable  $[d]$  that can be changed. Simulation results are presented in the next chapter.

## 4 NUMERICAL RESULTS

### 4.1 Case Study I – Noisy Data Approximation

Measured scattering parameters of interconnect are approximated over a frequency range using the proposed methods in Chapter 2. For comparison, the result of methods, the standard vector fitting, and the proposed method are presented.

First, the measured data is approximated by using the initial poles selection method proposed in [5]. Figures 1 and 2 show the result of the magnitude and phase of the S21 of the given interconnect, respectively. Next, the measured data is approximated using automatic starting pole selection. In this method, the order depends on the number of extremes in the frequency range under consideration. The order of approximation is 162. High order is expected due to high number of picks. The root-mean squared error and the number of iterations needed for the vector fitting to converge are  $2.9941 \times 10^{-4}$  and 5, respectively. Figures 3 and 4 show the result of the magnitude and phase of the S21 of the given interconnect, respectively. The same data is approximated using the proposed combined method. Figures 5 and 6 show the magnitude and phase of the S21, respectively. Table 1 shows the comparison between these methods.

Table 1. Comparison between the methods.

	Standard Method	Automatic Selection Method	Combined Method
Order of Approximation	110	162	84
RMS	$7.07 \times 10^{-3}$	$2.9941 \times 10^{-4}$	$7.7001 \times 10^{-4}$
Number of Iterations	8	5	8

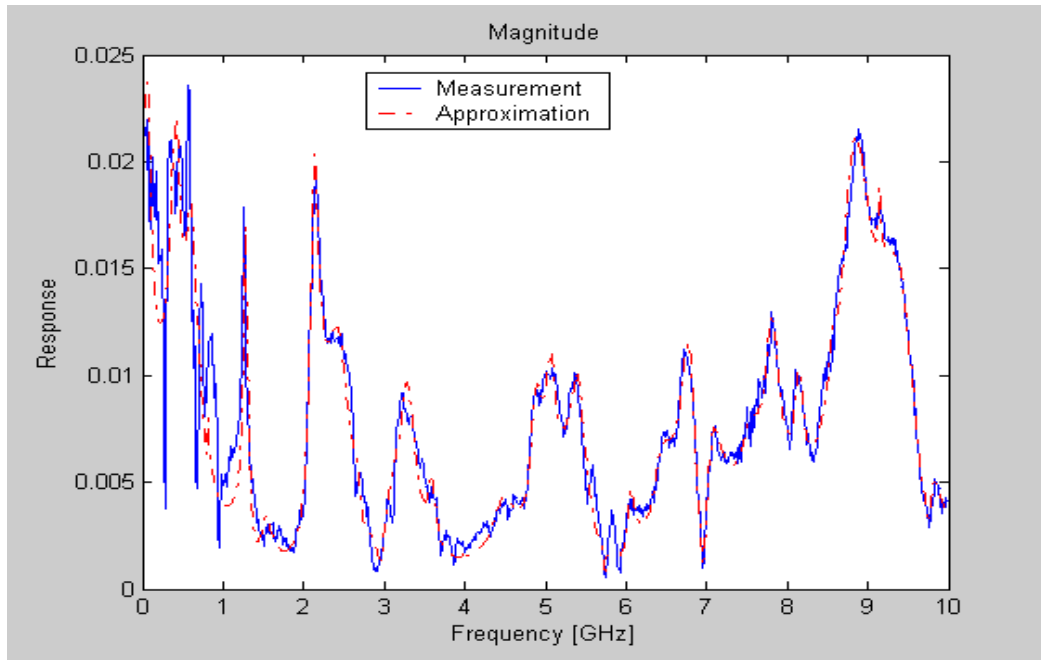


Figure 1. Magnitude of S21: Measured data and the 110<sup>th</sup>-order rational approximation using standard method.

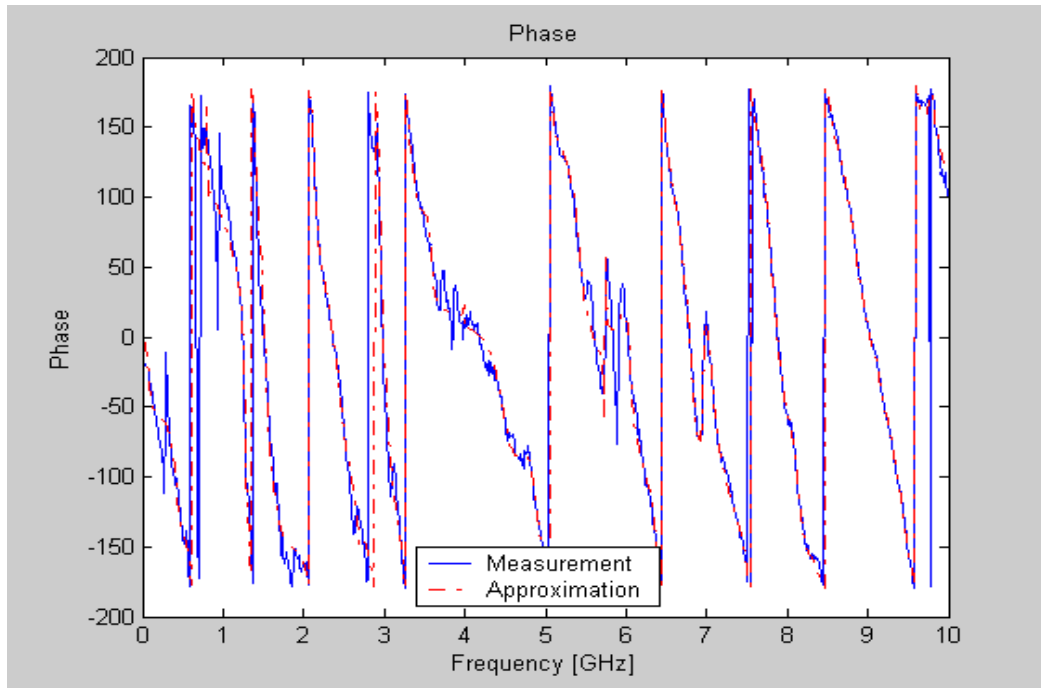


Figure 2. Phase of S21: Measured data and the 110<sup>th</sup>-order rational approximation using standard method.



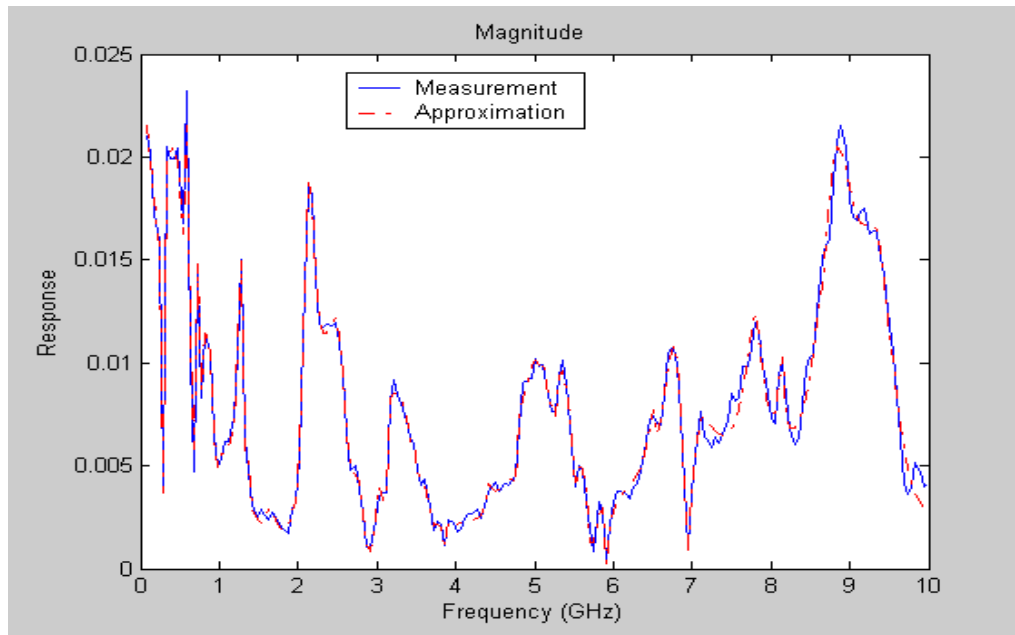


Figure 3. Magnitude of S21: Measured data and the 162<sup>nd</sup> -order rational approximation using automatic poles selection method.

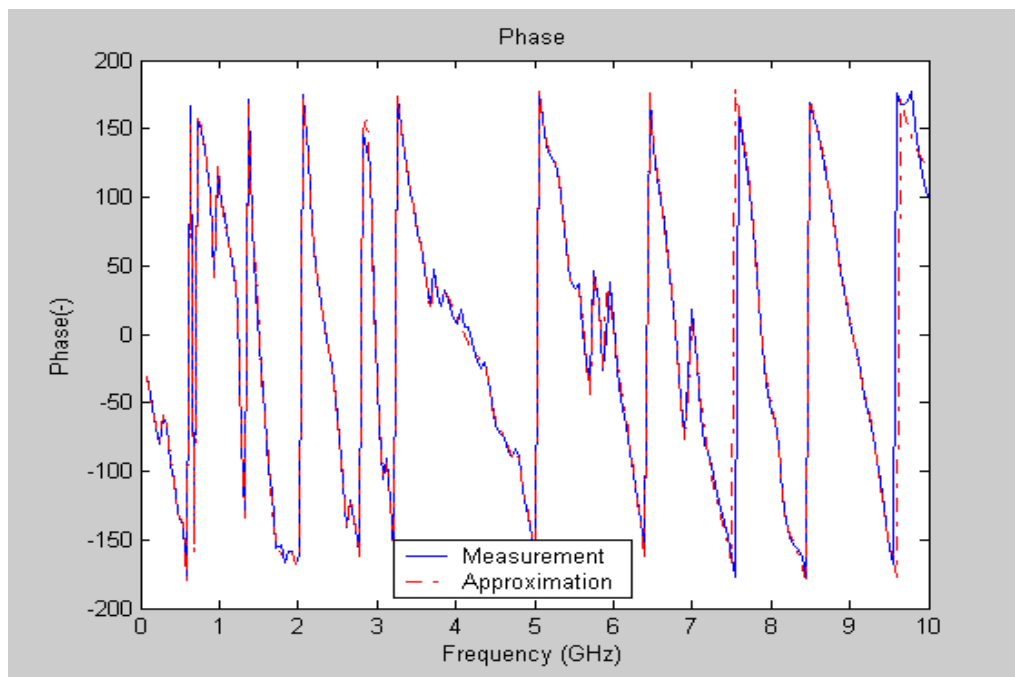


Figure 4. Phase of S21: Measured data and the 162<sup>nd</sup> -order rational approximation using automatic pole selection method.

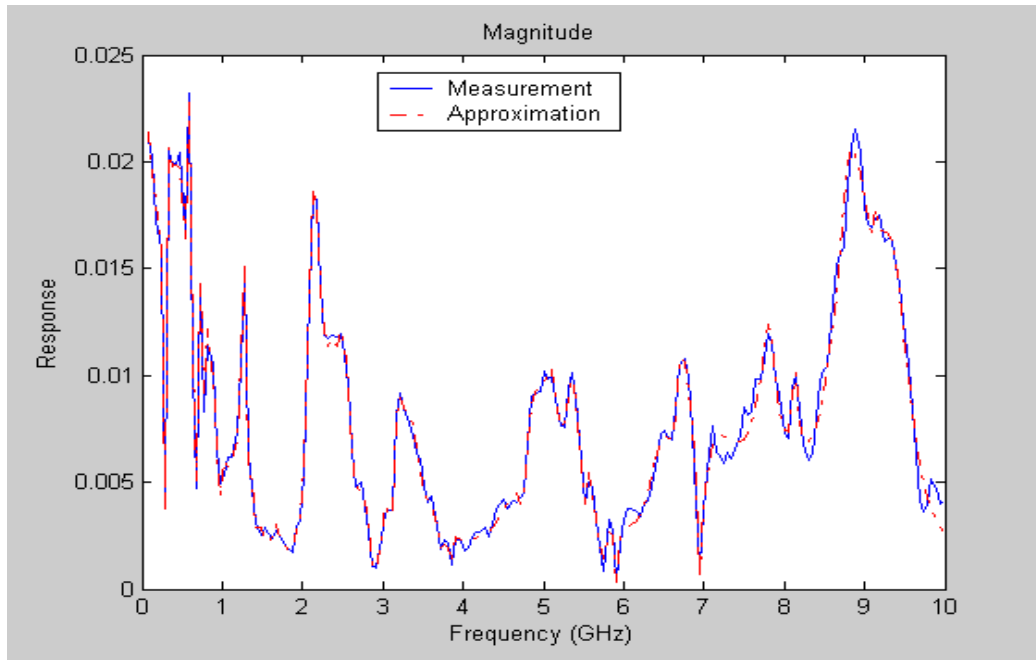


Figure 5. Magnitude of S21: Measured data and the 84<sup>th</sup>-order rational approximation using combined method.

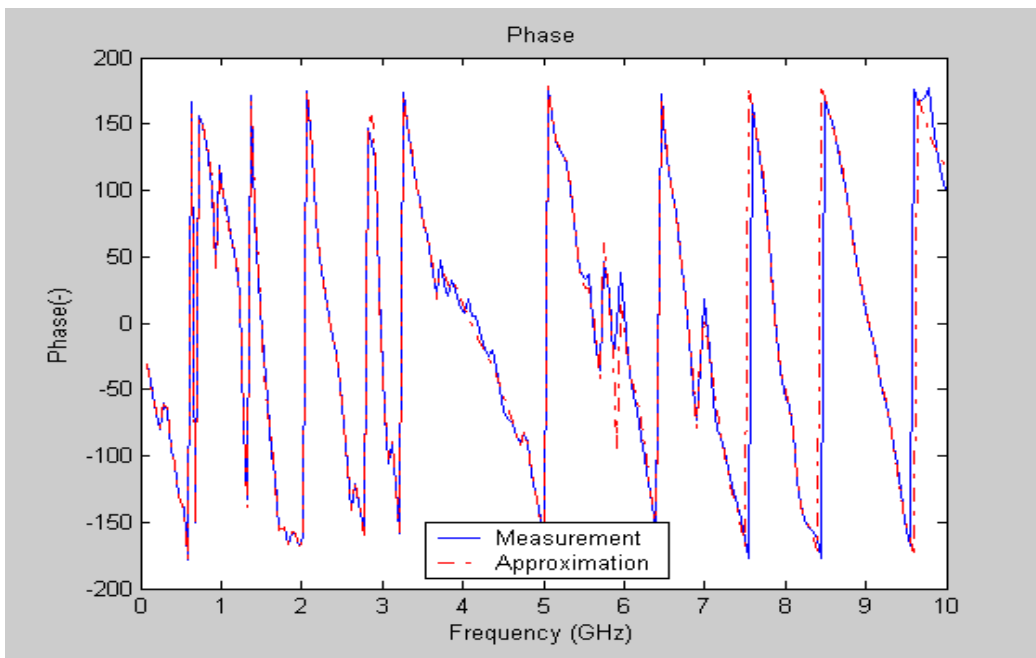


Figure 6. Phase of S21: Measured data and the 84<sup>th</sup>-order rational approximation using the combined method.

As can be seen in Table 1, the starting poles generated using the suggested rational interpolation method give better results with a lower order approximation. Even though the combined method needs more iterations than the standard vector fitting method for this example, the comparisons over many other examples show that the two methods require about the same number of iterations. The root-mean squared error is calculated at each iteration and run for 30 iterations using the combined method. Figure 7 shows the root-mean squared error versus the number of iterations. The error drops very fast in the first few iterations and stagnates at high iterations. For example, the root-mean squared error at the second iteration is 0.0012 while at the 15<sup>th</sup> iteration the root-mean squared error is 0.00062 and stays constant for higher numbers of iterations.

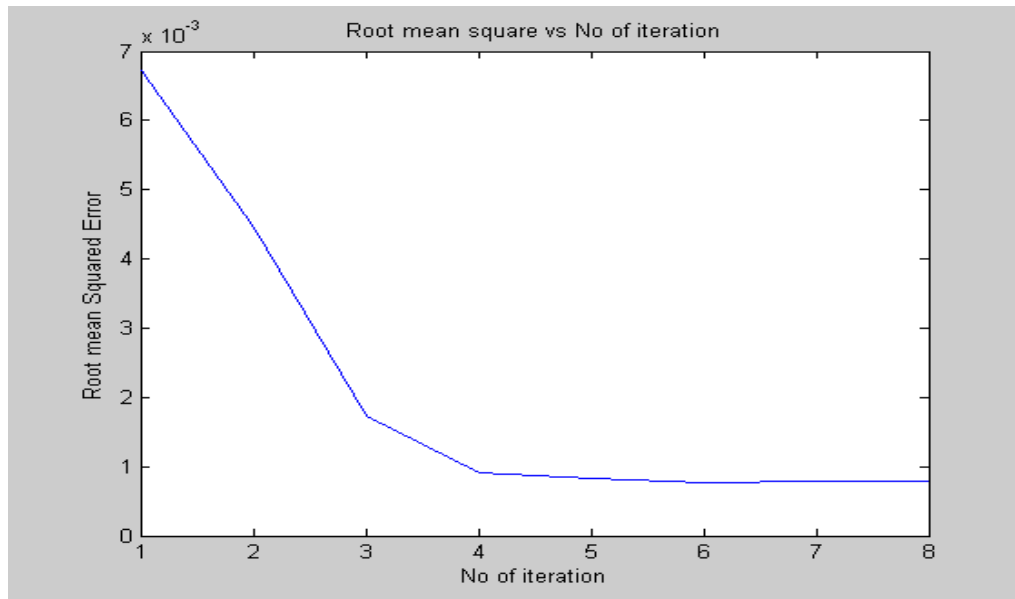


Figure 7. Root-mean squared error vs. number of iteration of the combined method.

## 4.2 Case Study II – Accuracy Test

Figure 8 shows an arbitrary structure developed using momentum in ADS. The frequency response of this two-port system is simulated with the quasi-static electromagnetic solver (EM) in ADS 2003C with Momentum MomEngine® 5.0 in radio frequency (RF) simulation mode.

The mesh is generated at 20-GHz mesh frequency with 20 cells per wavelength. The single layer substrate has 25-mil thickness with permittivity 9.6 and perfect ground. The two ports are characterized with 50-Ω termination. The simulation frequency range is from 45-MHz to 20-GHz. An S-parameter frequency response is obtained from the ADS in the frequency range. The S-parameters obtained from the EM simulation are approximated with the proposed method. Figures 9 and 10 show the magnitude and phase response obtained from the EM solver in ADS and the rational approximation using the proposed method, respectively.

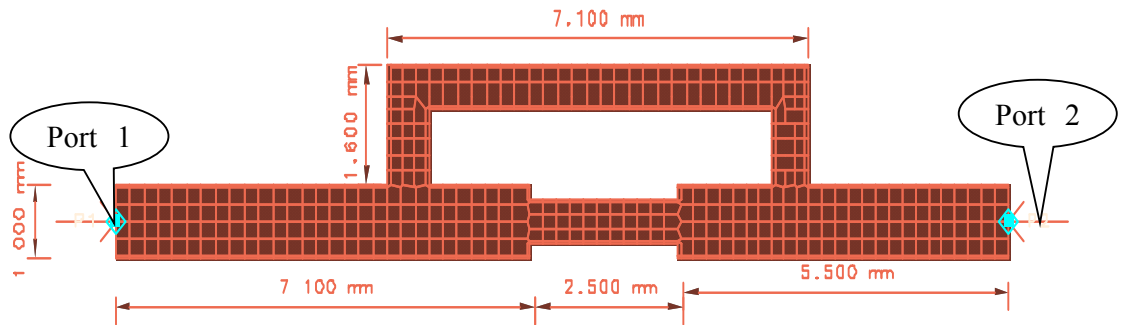


Figure 8. An arbitrary structure layout developed for test in ADS.

From Figures 9 and 10, the approximation is well fit with the original data. Since the system is a real system, the proposed method uses common system poles for each parameter. The order of approximation resulted from the method is 28 and it took only four iterations to reach a RMS of  $1.3492 \times 10^{-4}$ , which shows that a well fitted macromodel can be generated for any arbitrary structure with a wide frequency range. Figure 11 shows the 3D view of the structure.

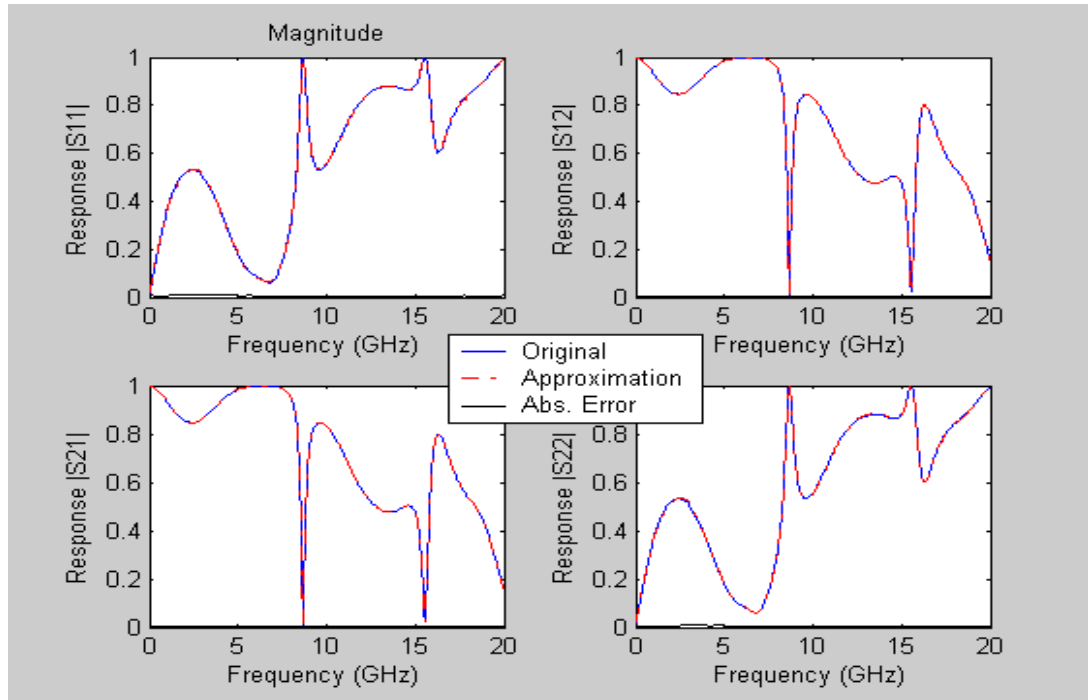


Figure 9. Magnitude response of the structure under consideration, EM simulation result from ADS (original), and rational approximation obtained from proposed method.

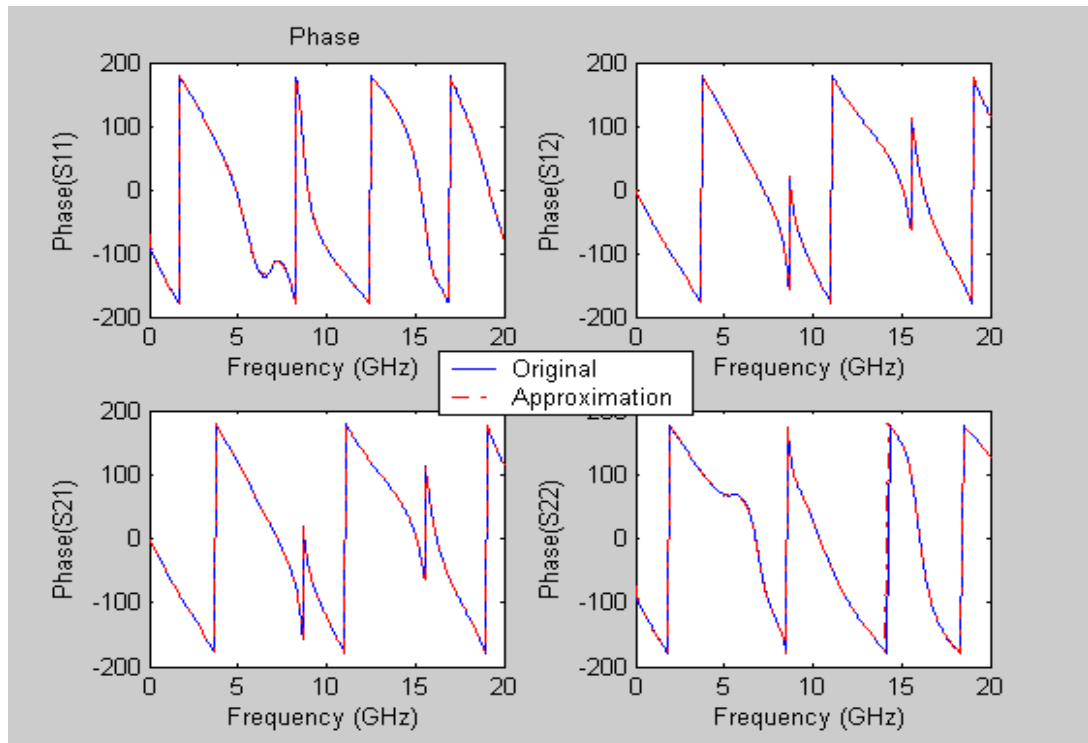
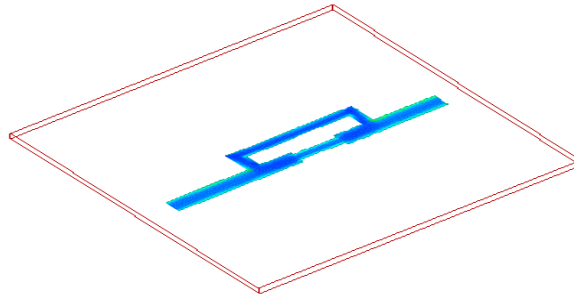


Figure 10. Phase response of the structure under consideration, EM simulation result from ADS (original), and rational approximation obtained from proposed method.



Tue Jun 15 00:20:06 2004

disc\_tran

Figure 11. The 3D view of the structure under consideration.

### 4.3 Case Study III – Convergence Test

The next case study is on the convergence of the proposed combined method. First a frequency data, which ranges from 0.1-GHz to 10-GHz, is generated from a known eight-order system so that the locations of the actual poles are already known. The frequency data is partitioned into four equal frequency ranges. The rational interpolation is applied on each partition such that good initial poles are extracted, that are going to be used in the vector fitting method. To show the process of convergence, a 12-order macromodel is generated using the proposed method so that the order of approximation is 12. Therefore, the macromodel generated has four more additional poles than the actual system. One way of observing the convergence of the combined method is by tracing the movements of the poles at each iteration. Figure 12 shows how the poles are moved with iteration and fit in to the actual pole position. Initially, the poles move a significant distance, but as the poles convergence, they move less and less. The poles mainly move in the horizontal directions, indicating that the imaginary parts of the initial and estimate poles are very close. Through iteration, those excess poles are pushed to be located as far as possible from the imaginary axis, or to be as close as possible to one of the actual poles

by the combined proposed method. If the poles are far from the imaginary axis, the transient time response of the whole system due to these poles is insignificant. On the other hand, if the excess poles are somehow close to the actual poles, the effect of these poles is incorporated together. This phenomenon is pole clustering and is discussed in [18]. One can apply pole clustering to reduce the high-order macromodel to a lower model by combining poles near to each other. It also shows that the excess poles are near to the exact poles or far from the imaginary axis. Table 2 presented the value of the poles in long precision at each iteration and the exact poles of the system. From the table, a fast convergence is achieved even at the first iteration, which shows poles generated by the rational interpolation as a starting pole results in accurate poles to be used by the vector fitting method.

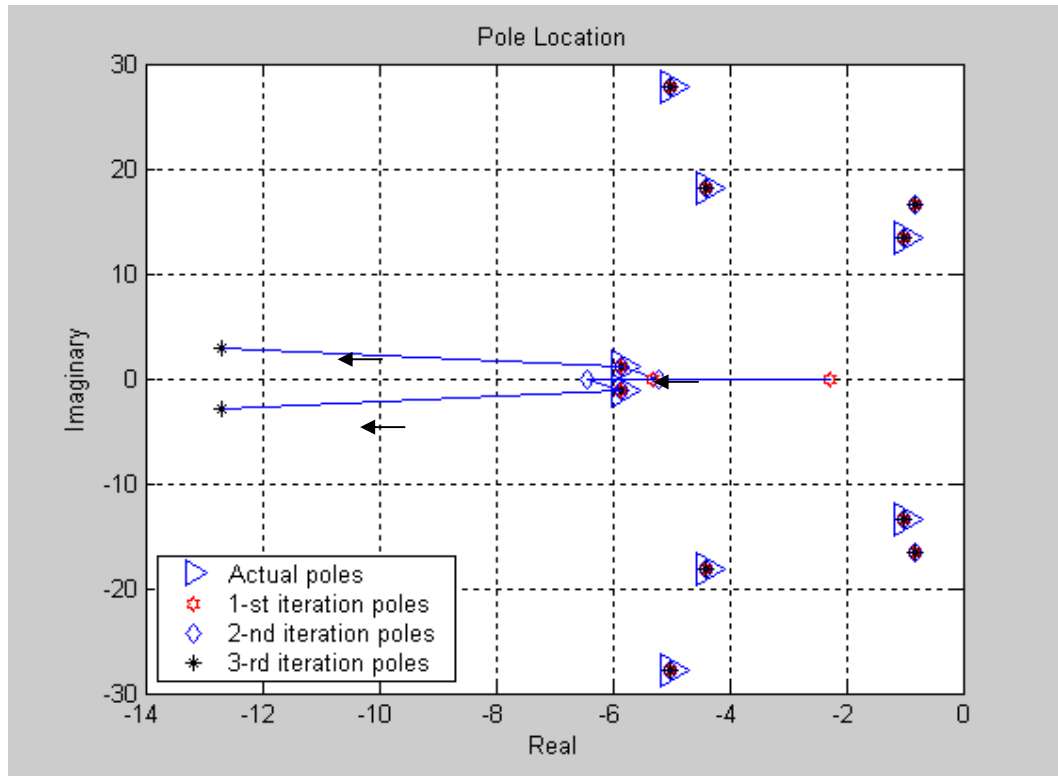


Figure 12. Pole locations with each iteration and the location of the actual poles.

Table 2. Normalized poles in each iteration and the exact poles in long precision format.

Poles after 1 <sup>st</sup> Iteration	Poles after 2 <sup>nd</sup> Iteration	Poles after 3 <sup>rd</sup> Iteration	Exact Poles
-2.28083083074210	-5.19615227575899	--	--
-5.29575863719451	-6.42868008638183	-12.69377218231458 ±2.95161226622251i	--
-5.84740000001007 ±1.15450000012744i	-5.84740000000000 ±1.15450000000114i	-5.84740000000016 ±1.15449999999966i	-5.84740000000000 ±1.15450000000000i
-1.03112699999914 ±13.35900000004103i	-1.03112699999998 ±13.35900000000030i	-1.03112700000009 ±13.35900000000028i	-1.03112700000000 ±13.35900000000000i
-0.82122437903587 ±16.54782858783110i	-0.82122437903587 ±16.54782858783111i	-0.82122437903588 ±16.54782858783112i	--
-4.40499999996669 ±18.20299999992384i	-4.40500000000000 ±18.20300000000002i	-4.40500000000000 ±18.20300000000002i	-4.40500000000000 ±18.20300000000000i
-5.01520000001838 ±27.74100000003333i	-5.01519999999999 ±27.7409999999998i	-5.01519999999999 ±27.74100000000000i	-5.01520000000000 ±27.74100000000000i

Figures 13 and 14 show the magnitude and the phase plot of the generated frequency data together with the approximate result.

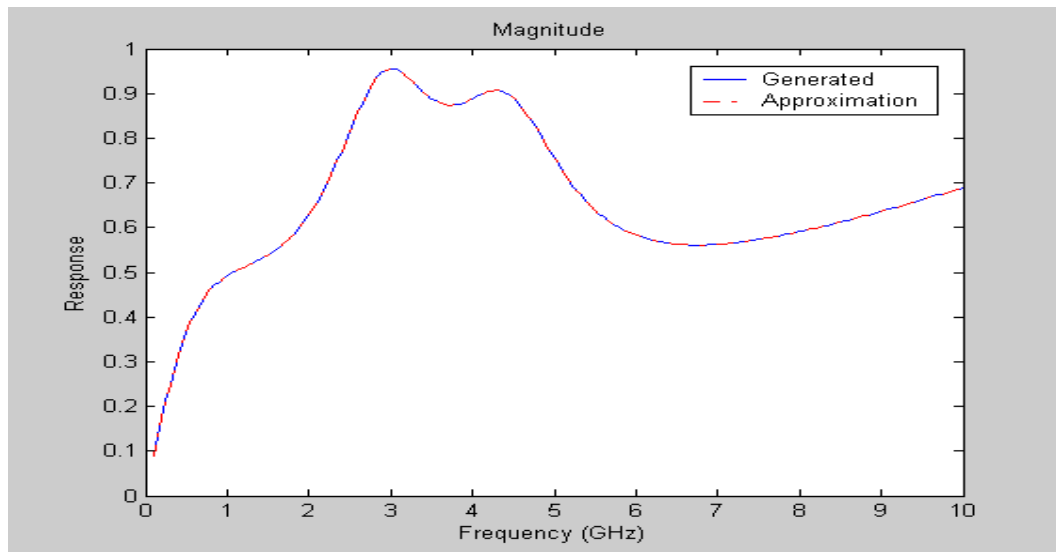


Figure 13. Magnitude response of the 8<sup>th</sup>-order system and the 12<sup>th</sup>-order rational approximation obtained from proposed method.



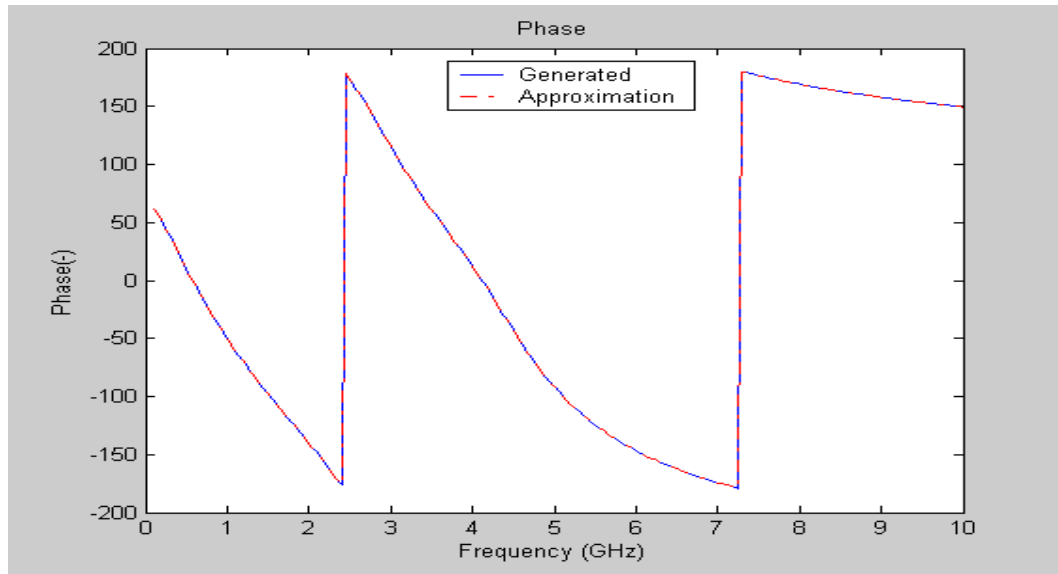


Figure 14. Phase response of the 8<sup>th</sup>-order system and the 12<sup>th</sup>-order rational approximation obtained from proposed method.

Table 3 shows the root-mean square error resulted at each iteration step. This table clearly shows that a fast convergence is achieved with few iterations, and once it reaches minimum RMS, it stays there for higher numbers of iterations.

Table 3. RMS vs. number of iterations.

Number of Iterations	RMS
1	$2.046443874377408 \times 10^{-12}$
2	$1.444960154069353 \times 10^{-15}$
3	$1.420157181164892 \times 10^{-15}$

## 5 SPICE-COMPATIBLE EQUIVALENT CIRCUIT SYNTHESIS

A pole-residue representation of a macromodel can easily be integrated into SPICE-like circuit simulator [19]. The pole-residue representation of the rational function matrix  $[G(s)]$  is used to easily construct a lumped circuit; elements consist of resistors, inductors, capacitors, and controlled sources. The method of extraction can be easily demonstrated using an admittance network. For a one-port admittance network, RL networks can be used to represent a real pole and residue. RLC networks can be used to represent a pair of complex-conjugate poles and residues. A resistor can be used to represent the direct constant. A capacitor can be used to represent the proportional constant. For a passive multiport admittance macromodel, the sign of all the equivalent elements must be positive. A closed-form admittance matrix representation for macromodel with  $N$  ports can be written the same as previously:

$$[Y(s)] = \sum_{n=1}^{CPN/2} \left( \frac{[\alpha_n] + j[\beta_n]}{s - p_{nr} - jp_{ni}} + \frac{[\alpha_n] - j[\beta_n]}{s - p_{nr} + jp_{ni}} \right) + \sum_{m=1}^{RPN} \frac{[\gamma_m]}{s - p_{mr}} + [d] + s[h] \quad (49)$$

$$[Y(s)] = \sum_{n=1}^{CPN/2} \frac{2[\alpha_n](s - p_{nr}) - 2[\beta_n]p_{ni}}{(s - p_{nr})^2 + p_{ni}^2} + \sum_{m=1}^{RPN} \frac{[\gamma_m]}{s - p_{mr}} + [d] + s[h], \quad (50)$$

where  $RPN$  is the number of real poles,  $CPN$  is the total number of complex poles, and the same set of poles is used. Each of the terms in admittance Equation (50) is interpreted in terms of an equivalent circuit.

## 5.1 Self Admittance Term

For the self-admittance term for the  $m^{\text{th}}$  real pole,

$$Y_r[s] = \frac{\gamma_m}{s - p_{mr}}$$

$$I_r^i = \frac{\gamma_m}{s - p_{mr}} v^i = \frac{1}{\frac{s}{\gamma_m} - \frac{p_{mr}}{\gamma_m}} v^i = \frac{1}{LD_m + RD_m} v^i, \quad (51)$$

where inductance value  $LD_m = \frac{1}{\omega_o \gamma_m}$  and resistance value  $RD_m = \frac{-p_{mr}}{\gamma_m}$  at port  $i$ . The

equivalent circuit realization is given in Figure 15.

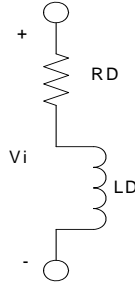


Figure 15. Equivalent circuit realization of a real pole in the self-admittance term.

For the self-admittance term for the  $n^{\text{th}}$  complex-conjugate pairs,

$$Y_n(s) = \frac{\alpha_n + j\beta_n}{s - p_{nr} - jp_{ni}} + \frac{\alpha_n - j\beta_n}{s - p_{nr} + jp_{ni}}$$

$$= \frac{2\alpha_n s - 2(\alpha_n p_{nr} + \beta_n p_{ni})}{s^2 - 2p_{nr}s + (p_{nr}^2 + p_{ni}^2)} \quad (52)$$

The above equation can be written in current-voltage form as

$$I_n^i = \frac{2\alpha_n s - 2(\alpha_n p_{nr} + \beta_n p_{ni})}{s^2 - 2p_{nr}s + (p_{nr}^2 + p_{ni}^2)} v^i \quad (53)$$

Equivalently, the above equation can be expressed as

$$I_n^i = \frac{\frac{1}{L^n} s + \frac{1}{R_p^n L^n C^n}}{s^2 + \left(\frac{R_s^n}{L^n} + \frac{1}{R_p^n C^n}\right)s + \frac{R_p^n + R_s^n}{R_p^n L^n C^n}} v^i, \quad (54)$$

where

$$R_s^n = \frac{-\alpha_n p_{nr} + \beta_n p_{ni}}{2\alpha_n^2}$$

$$L^n = \frac{1}{2\omega_o \alpha_n}$$

$$R_p^n = \frac{p_{ni}^2 (\alpha_n^2 + \beta_n^2)}{2\alpha_n^2 (-\alpha_n p_{nr} - \beta_n p_{ni})}$$

$$C^n = \frac{2\alpha_n^3}{\omega_o p_{ni}^2 (\alpha_n^2 + \beta_n^2)}.$$

The equivalent circuit realization is given in Figure 16.

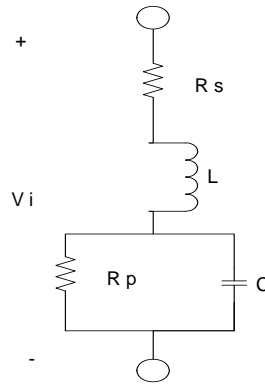


Figure 16. Equivalent circuit realization for a complex-conjugate pair in the self-admittance term.

For the self-admittance term, the direct constant value and the proportional constant value are represented as a series resistor and a capacitor:

$$\begin{aligned} R_{dc} &= \frac{1}{d} \\ C_{prop} &= \frac{h}{\omega_o}. \end{aligned} \tag{55}$$

The equivalent circuit realization is given in Figure 17.

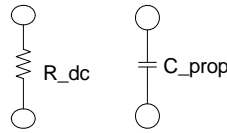


Figure 17. Equivalent circuit realization for direct constant and proportional constant in the self-admittance term.

## 5.2 Transadmittance Term

The equivalent circuit realization of transadmittance between ports can be realized in such a way that the contribution of the  $j^{\text{th}}$ -port voltage to the current of the  $i^{\text{th}}$ -port through the real poles, complex-conjugate pair poles, and constant terms that relates between these ports. Therefore, it is possible to create a voltage controlled current source (VCCS) in the independent side (port  $j$ , in this case), which in turn control the current controlled voltage source (CCVS) in the dependent side (port  $i$ , in this case).

- In the transadmittance term, the  $m^{\text{th}}$  real pole between ports  $i$  and  $j$  can be realized as in Figure 18.

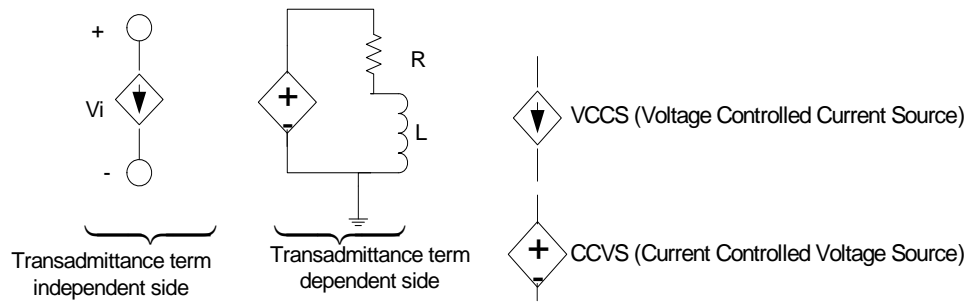


Figure 18. Equivalent circuit realization in transadmittance term of a real pole between ports  $i$  and  $j$ .

- In the transadmittance term, the  $n^{\text{th}}$  complex-conjugate poles between ports  $i$  and  $j$  can be realized as in Figure 19.

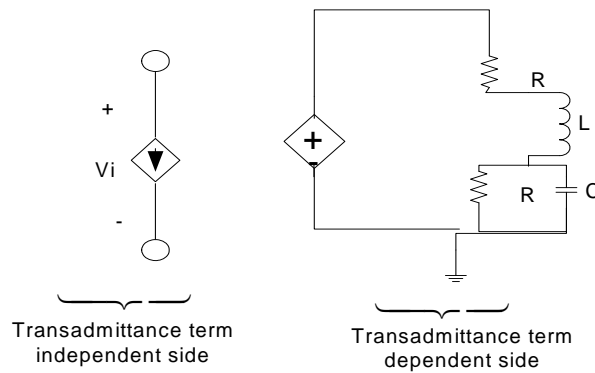


Figure 19. Equivalent circuit realization in transadmittance term of a complex-conjugate pair poles between ports  $i$  and  $j$ .

The direct and proportional constant also contributes to the transadmittance realization between ports as a resistor and capacitor, respectively. Therefore, the equivalent circuit for the  $i^{\text{th}}$  port resulting from the combination of all the equivalent subcircuit representations of the self-admittance and transadmittance terms is as in Figure 20.

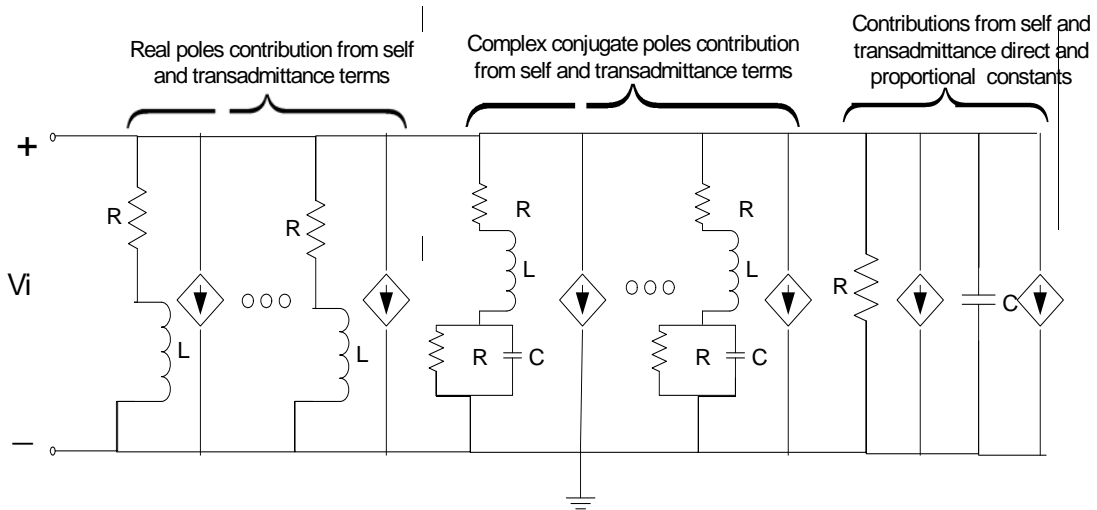


Figure 20. Equivalent circuit realization for the  $i^{\text{th}}$  port.

In Figure 20, all the VCCS are from the contribution of the  $j^{\text{th}}$  transadmittance term. All the other components are from the self-admittance terms. The dependent sources, which control the VCCS in the  $i^{\text{th}}$  port, are as in Figure 21.

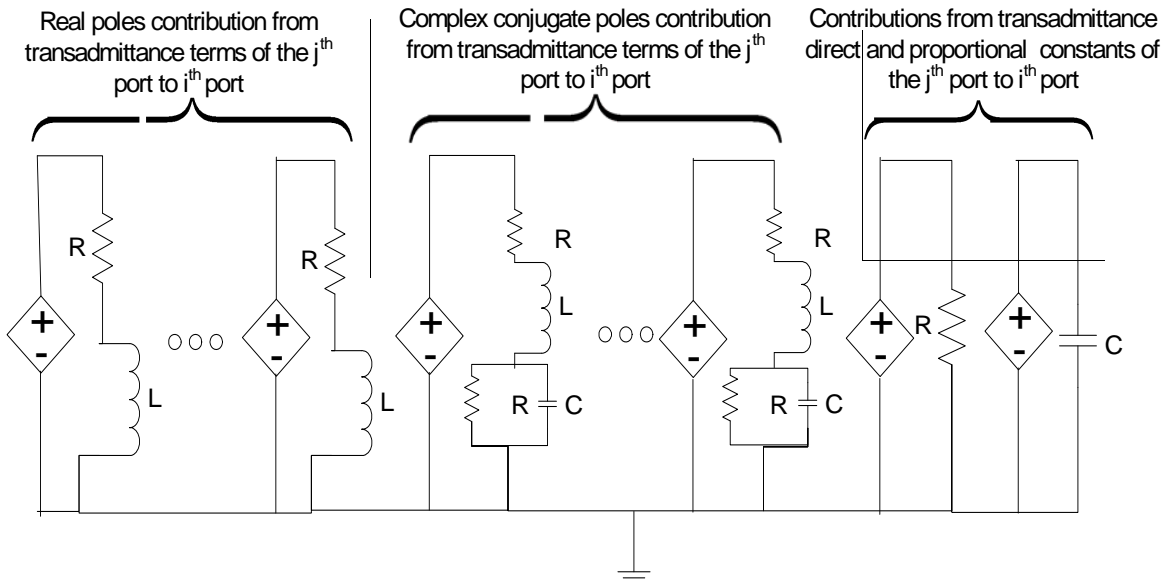


Figure 21. Equivalent circuit realization of the dependent sources for the transadmittance terms.

## 6 CONCLUSIONS

The goal of this work is to construct low-order macromodels of passive systems from frequency data for simulation of multiport networks. The macromodels need to be stable and passive to be used to speed time-domain circuit simulation of large networks. Using the constructed macromodels, the electromagnetic behavior of distributed networks can be successfully integrated into circuit simulators for design and analysis of a more complex system. A combined method that uses rational interpolation is proposed to improve the initial poles and convergence of the vector fitting method. The procedure takes advantage of the robustness of the rational interpolation method in generating stable poles over partitioned frequency range and the numerical stability and fast convergence of the vector fitting method to construct high-order approximation over wide frequency range. The convergences of the standard vector fitting and proposed methods are compared. It is shown that the error decreases very fast and stagnates at high iterations. The movements of poles are also studied to compare the initial poles and the final estimate poles. A SPICE compatible equivalent realization is also studied for a system characterized by admittance matrix.

In this work, the convergence of the hybrid method is illustrated by using various examples and by studying the movement of the poles with iteration. The mathematical proof of the convergence with iteration should be studied. The hybrid method gives good approximation independent of the parameter measured or given. SPICE compatible equivalent circuit realization should also develop for other types of parameters like S-parameter frequency data.



## APPENDIX – SOURCE CODE

The Matlab® code implementation of the hybrid method is presented. One should modify the code below to plot the preferred data. Due to constraint in space, small functions are taken out.

```

%*****
% FUNCTION NAME:- Start_by_rational(fun,frequency,Order p)
%*****
% This Function accepts data to be approximated and the order
% of approximation. It performs rational interpolation from the
% given data. It returns the approximate poles in the frequency range.
%
% BY: Yidnek Mekonnen , June 2004
%*****

function [prunedpoles] = Start_by_rational(fun,frequency,p)
q=p; %assume the denominator order = numerator order
% Rational approximation of a complex analytical function
f0 = fun(1);
alpha=1;
f=fun(2:length(fun));
f= f';
freq=frequency(2:length(frequency))/alpha;
freal = real(f);
x = (freq).^2;
xmax = max(x);
x=x/xmax;
K = length(x);
A = ones(1,K+1);
for k=1:p
    A = [A:[0,x.^k]];
end
for k=1:q
    A = [A:[0,-(x.^k).*freal]];
end;
soln = (A')\[f0,freal]';
soln = real(soln);
for I=1:q
    soln(1+p+I)=soln(1+p+I)/(xmax^I);
end
denpoly = fliplr([1:soln(2+p:p+q+1)].');
%prunedpoles = sqrt(-roots(denpoly));
poles = sqrt(-roots(denpoly));
prunedpoles = poles(find(real(poles)>0));
prunedpoles_real = real(prunedpoles);
prunedpoles = prunedpoles_real+ j*1*2*imag(prunedpoles);
prunedpoles = prunedpoles';
for k=1: length(prunedpoles)
    if imag(prunedpoles(k)) == 0
        prunedpoles(k) = prunedpoles(k) + j*2*prunedpoles(k);
        prunedpoles(length(prunedpoles)+1) = real(prunedpoles(k)) -
j*imag(prunedpoles(k));
    end
end

```

```

end
prunedpoles = sort(prunedpoles);
%*****%

%*****%
% FUNCTION NAME: - STD_VECTORFIT(fun,frequency,Order p) %
%*****%
%*****%
% This Function accepts the data to be approximated, initial starting %
% poles,the order of approximation,and the vector fitting number of %
% iteration. It returns the residues,the new poles , direct constants and %
% proportional constants and also the root mean error and the normal error %
%
%
%           BY: Yidnek Mekonnen      ,   June 2004
%*****%

function[Residues,New_poles,d_constant,Pro_constant,D,rmserror]=
STD_VECTORFIT(f,frequency,Starting_pole,Order,No_iteration)
freq_length = length(frequency);
s = 2*pi*j*frequency;
f_copy = f;
G_row = []; G = []; H = [];
No_real_pole = 0;
Real_Starting_pole = [];
Complx_Starting_pole = [];
for i =1: length(Starting_pole) % Knowing Only real poles and complex poles
    if imag(Starting_pole(i))==0
        Real_Starting_pole =[Real_Starting_pole, Starting_pole(i)]; %Only
real starting poles
        No_real_pole = No_real_pole +1; %Counter of Real starting poles
    else
        Complx_Starting_pole
=[Complx_Starting_pole,Starting_pole(i)]; %complex starting poles
    end
end
sort(Complx_Starting_pole); %sorting complex starting poles
sort(Real_Starting_pole); %Sorting real starting poles only
Starting_pole=[Real_Starting_pole,Complx_Starting_pole];
f = f_copy;
for k=1: freq_length
    for p=1:No_real_pole %Real poles only
        G_row=[G_row, 1./(s(k)-Starting_pole(p))];
    end
    %The following routine helps to get the conjugate solution of C and
    %C_bars. C_bar will be the poles of the system and it must be in
    %complex-conjugate b/s it represent a real system.
    for p=No_real_pole+1:2:length(Starting_pole) %for complex starting
poles
        real_part_pole=(1/(s(k)-Starting_pole(p)))+(1/(s(k)-
(Starting_pole(p))'));
        imag_part_pole=j*(1/(s(k)-Starting_pole(p))-1/(s(k)-
(Starting_pole(p))'));
        G_row=[G_row,real_part_pole,imag_part_pole];
    end;
    G =[G;G_row];
    H =[H;G_row*(-f(k))];
    G_row =[];
end %End of freq_length loop

```

```

A = [G,H,(ones(1,freq_length))',s.'];
imag_A = imag(A);
A = real(A);
A = [A;imag_A];
clear imag_A; clear G; clear H;
imag_f = imag(f);
f = real(f);
f = [f;imag_f];
clear imag_f;
for col=1:length(A(1,:)); %normalizing the A matrix with Euclidian norm
    Escale(col)=norm(A(:,col),2); %Euclidian norm
    A(:,col)=A(:,col)./Escale(col);
end
X = A\f;
X = X./Escale. '; % rescale the solution
C = X(1:(length(X)-2)/2); %the residues of the function f
C_bar = X((length(X)-2)/2+1:(length(X)-2)); % the residues of the sigma
function
d_constant = X(length(X)-1); % the direct constant d
Pro_constant = X(length(X)); % the proportional constant in the function
C = C. ';
C_bar = C_bar. ';
for p=No_real_pole+2:2:length(C) %make C complex-conjugate
    r1 = C(p-1);
    r2 = C(p);
    C(p-1)=r1+j*r2;
    C(p)=r1-j*r2;
end
for p=No_real_pole+2:2:length(C) %make C_bar complex-conjugate
    r1 = C_bar(p-1);
    r2 = C_bar(p);
    C_bar(p-1)=r1+j*r2;
    C_bar(p)=r1-j*r2;
end
%Calculating the Zeros of sigma function ,which is the poles of the function
A=[]; B=[];CC=[];
for p=1:2:length(Complx_Starting_pole)
    A(p,p)=real(Complx_Starting_pole(p));
    A(p+1,p+1)=A(p,p);
    A(p,p+1)=imag(Complx_Starting_pole(p));
    A(p+1,p)=-1*A(p,p+1);
    B(p)=2;
    B(p+1)=0;
    CC(p)=real(C_bar(p));
    CC(p+1)=imag(C_bar(p));
end
B = B';
H = A - B*CC;
New_poles = eig(H);
for p=1:length(New_poles)
    if real(New_poles(p))>0 %making unstable poles stable.
        New_poles(p)= -1*real(New_poles(p))+j*2*imag(New_poles(p));
    end
end
New_poles = sort(New_poles);
New_poles = New_poles - 2*j*imag(New_poles);
clear f;
clear C;
clear Escale;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Finding the residues and other constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Real_New_pole = [];
Complx_New_pole = [];
No_real_pole = 0;
for p=1:length(New_poles) % Knowing Only real poles and complex poles
    if imag(New_poles(p))==0
        Real_New_pole =[Real_New_pole, New_poles(p)]; %Only real starting
poles
        No_real_pole = No_real_pole +1; %Counter of Real starting poles
    else
        Complx_New_pole =[Complx_New_pole,New_poles(p)]; %complex starting
poles
    end
end
New_poles=[Real_New_pole,Complx_New_pole];
G_row =[]; G=[];
for k=1: freq_length
    f = f_copy;
    for p=1:No_real_pole %Real poles only
        G_row=[G_row, 1./(s(k)-New_poles(p))];
    end
    for p=No_real_pole+1:2:length(New_poles)
        real_part_pole=(1/(s(k)-New_poles(p)))+(1/(s(k)-(New_poles(p))'));
        imag_part_pole=j*(1/(s(k)-New_poles(p))-1/(s(k)-(New_poles(p))'));
        G_row=[G_row,real_part_pole,imag_part_pole];
    end;
    G =[G;G_row];
    G_row =[];
end %End of freq_length loop
Starting_pole = New_poles; % For next Iteration
A =[G,(ones(1,freq_length))',s.'];
imag_A = imag(A);
A = real(A);
A = [A;imag_A];
clear imag_A; clear G;
imag_f = imag(f);
f = real(f);
f =[f;imag_f];
clear imag_f; clear Escale;
for col=1:length(A(1,:));
    Escale(col)=norm(A(:,col),2); %Euclidian norm
    A(:,col)=A(:,col)./Escale(col);
end
X = A\f;
X = X./Escale. ';
C=[];
C=X(1:length(X)-2);
d_constant = X(length(X)-1);
Pro_constant =X(length(X));
C=C. ';
for p=No_real_pole+2:2:length(C) %make C complex-conjugate
    r1 = C(p-1);
    r2 = C(p);
    C(p-1)=r1+j*r2;
    C(p)=r1-j*r2;
end

```

```

end
Residues = C;
D = zeros(1,freq_length);
for q = 1:length(New_poles) %Order ...modified 08.19.2003
    D = D + (Residues(q)/New_poles(q))./(-1+ s./New_poles(q));
end
D = D + Pro_constant*s + d_constant;
diff= D - f_copy.';
rmerror=sqrt(sum(sum(abs(diff.^2))))/sqrt(length(frequency));
%END
%*****%
%*****%

%*****%
% MAIN FILE NAME: - HYBRID_METHOD %
%*****%
%*****%
% This File is the main file. User must give a file name of the data, the %
% order of approximation and the number of iteration. User also must give %
% the parameter to be approximated. This file call S_TO_Y(not included here)%
% Start_by_rational and STD_VECTORFIT. After running, it returns the %
% residues the poles , direct constants and proportional constants and %
% also the root % mean error and the normal error. It plots the parameter %
% user approximated. %
% %
% BY: Yidnek Mekonnen , June 2004 %
%*****%

clear all;
fid=fopen('V60.txt','r'); % Data file
Amat=fscanf(fid,'%e',[9,801]); %read a file which contains a data
fclose(fid);
Amat = Amat';
[Y11,Y21,Y12,Y22,frequency] = S_to_Y(Amat,3); %Type 1 rectangular form or
Type 2 polar form Type 3 only S parameter
f = Y12; % This is chosen to find the system poles...It is possible to
choose other parameter.
f = f.';
f_copy = f; %Retain copy of the parameter
Each_order =6;
[Starting_pole_1] = Start_by_rational(f(1:40),frequency(1:40),Each_order);
[Starting_pole_2] = Start_by_rational(f(41:80),frequency(41:80),Each_order);
[Starting_pole_3] = Start_by_rational(f(81:120),frequency(81:120),Each_order);
[Starting_pole_4] =
Start_by_rational(f(121:160),frequency(121:160),Each_order);
[Starting_pole_5] =
Start_by_rational(f(161:200),frequency(161:200),Each_order);

[Starting_pole]=[Starting_pole_1,Starting_pole_2,Starting_pole_3,Starting_pole_4,Starting_pole_5]; %Collecting all the starting poles from each partition
freq_length = length(frequency);
Order = length(Starting_pole);
No_iteration =8; % Number of Iteration for vector fitting
Error_array = [];
Pole_store = [];
clear figure(4);
for q=1:No_iteration
    Starting_pole = sort(Starting_pole);

```

```

[Residues,New_pole,d_constant,Pro_constant,D,Error]=
STD_VECTORFIT(f,frequency,Starting_pole,Order,No_iteration);
Pole_store = [Pole_store;New_pole]; %only used to display the pole
convergence
if q ~= No_iteration % Except for the last iteration..make the real pole
resulted complex-conjugate..
    for p=1:length(New_pole)
        if imag(New_pole(p))==0
            New_pole(p) = real(New_pole(p)) + j*2*real(New_pole(p));
        end
    end
end
end
Error_array = [Error_array,Error];
Starting_pole = New_pole;
Order = length(Starting_pole);
Error %Displaying the root mean error
end %End of Iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diff= D - f_copy.';
figure(15),plot(frequency,abs(f_copy),'-b',frequency,abs(D),'-
.r','MarkerSize',12);
xlabel('Frequency (GHz)');
ylabel('Response');
legend('Measurement','Approximation',2);
title('Magnitude ');
aa=180/pi;
figure(8),plot(frequency,aa*angle(f_copy),'-b',frequency,aa*angle(D),'-.r');
xlabel('Frequency (GHz)');
ylabel('Phase(-)');
legend('Measurement','Approximation',3);
title('Phase');
New_pole = sort(New_pole);
Iter = [1:No_iteration];
figure(77),plot(Iter,Error_array);
xlabel('No of iteration');
ylabel('Root mean Squared Error');
title('Root mean square vs No of iteration');

%END
%*****
%*****

```

## REFERENCES

- [1] E. Chiprout and M. S. Nakhla, *Asymptotic Waveform Evaluation and Moment Matching of Interconnect Analysis*. Boston, MA: Kluwer, 1994.
- [2] P. Feldmann and R. W. Freud, "Efficient linear circuit analysis by Padè approximation via the Lancos process," *IEEE Trans. On Computer-Aided Design*, vol. 14, pp. 639-649, May 1995.
- [3] W. T. Beyene and J. E. Schutt-Aine, "Efficient transient simulation of high-speed interconnects characterized by sample data," *IEEE Trans. Component, Packaging, and Manufacturing Tech.*, vol. 21, pp. 105-114, Feb 1998.
- [4] R. Gao, Y. S. Mekonnen, W. T. Beyene, and J. E. Schutt-Aine, "Black-box modeling by rational function approximation," in *Proceedings of IEEE Workshop on Signal Propagation in Interconnects*, SPI- 2004, pp. 203-210.
- [5] 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.
- [6] B. Gustavsen, "Computer code for rational approximation of frequency dependent admittance matrices," *IEEE Trans. Power Delivery*, vol. 17, pp. 1093-1098, Oct 2002.
- [7] Y. S. Mekonnen, W. T. Beyene, and J. E. Schutt-Aine, "Improved high-order approximation by combining rational interpolation with the vector fitting method," in *20th Annual Review of Progress in Applied Computational Electromagnetic*, Syracuse, NY, April 19-23, 2004, pp. 104-109.
- [8] J. Tokarzewski, "System zeros analysis via the Moore-Penrose pseudo inverse and the SVD of the first nonzero Markov parameter," *IEEE Trans. on Automatic Control*, vol. 43, no. 9, pp. 1285-1291, Sept. 1998.
- [9] J. Tokarzewski, "Zeros and Kalman canonical form of MIMO LTI systems," in *Proceedings of European Control Conf.*, ECC, Brussels, Belgium, July 1-4, 1997, pp. 114-119.
- [10] J. Tokarzewski, "A note on determination of zeros and zero directions by the Moore-Penrose pseudo inverse of the first nonzero Markov parameter," in *UKACC Int. Conf. on Control*, vol. 1, Sept. 2-5, 1996, pp. 42-47.
- [11] C. T. Chen, *Linear System Theory and Design*. 3<sup>rd</sup> ed., New York: Oxford, 1999.
- [12] G. W. Stewart, *Introduction to Matrix Computations*. New York: Academic Press, 1973.

- [13] B. Gustavsen and A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions," *IEEE Trans. Power Systems*, vol. 16, pp. 97-104, Feb. 2001.
- [14] S. Grivet-Talocia, "Enforcing passivity of macromodels via spectral perturbation of the Hamiltonian matrices," in *Proceedings of IEEE Workshop on Signal Propagation in Interconnects*, May 2003, pp. 287-290.
- [15] S. H. Min and M. Swaminathan, "Efficient construction of two-port passive macromodels for resonant networks," in *Proceedings of IEEE 10<sup>th</sup> Topical Meeting on Electrical Performance of Electrical Packaging*, Oct 2001, pp. 229-232.
- [16] D. Saraswat, R.Achar, and M. Nakhla, "Enforcing passivity for rational function based macromodels of tabulated data," in *Proceedings of IEEE 12<sup>th</sup> Topical Meeting on Electrical Performance of Electrical Packaging*, Oct 2003, pp. 295-298.
- [17] S. Kami, *Network Theory: Analysis and Synthesis*. Boston: Allyn and Bacon, 1966.
- [18] W. T. Beyene, "Pole-clustering and rational interpolation for simplifying distributed systems," *IEEE Trans. Circuits and Systems I*, vol. 46, no. 12, pp. 468-472, Dec. 1999.
- [19] M. J. Choi and A. C. Cangellaris, "A quasi three-dimensional distributed electromagnetic model for complex power distribution networks," in *IEEE 51<sup>st</sup> ECTC*, May 2001, pp. 1111-1116.