Fast Algorithm for Transient Analysis of Distributed Interconnects Including Driver and Load Circuits

by

Lidija Filipovic

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Applied Sciences

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Ontario, Canada
September 2007

© Lidija Filipovic 2007
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.
Abstract

Rapid advances in integrated circuit technology, such as, the ever-increasing signal speed, complex functionality, larger circuits and shrinking device sizes, have drastically increased the need for fast and accurate interconnect analysis. At higher frequencies, the conventional lumped models become inadequate in describing the interconnect performance and distributed transmission line models become necessary. In this thesis, an efficient algorithm is introduced for time-domain simulation of interconnects including driver and load. Presented approach extracts transfer function definition of output signals in terms of pole-residue based time-domain macromodels. The new algorithm is suitable for a repetitive type of environment, where the transient responses need to be estimated for varying loads and drivers, quickly. The proposed method yields significant speed-up compared to the MNA type analysis of distributed networks with load and driver. Several numerical examples are presented to demonstrate its validity and efficiency.
Acknowledgements

I would like to express my most sincere gratitude to my mentors and supervisors Prof. Ram Achar and Prof. Michel Nakhla. Their tireless support and encouragement have made this work an enriching and enjoyable experience. It has been my pleasure to be advised and guided by you.

Furthermore, I would like to thank and acknowledge my colleagues in the CAD group at Carleton University. A special thanks are due to Dharmendra Saraswat and Andrew Charest for many helpful discussions.
## Contents

1 **Introduction** ................................................................. 1
   1.1 Background and Motivation ...................................... 1
   1.2 Contributions .......................................................... 4
   1.3 Organization of the Thesis ........................................ 4

2 **Interconnects** ............................................................... 6
   2.1 Interconnect Modelling .............................................. 6
      2.1.1 Lumped Models ................................................. 8
      2.1.2 Distributed Models ........................................... 9
      2.1.3 Full-Wave Models ........................................... 10
   2.2 Interconnect Simulation ........................................... 11
      2.2.1 Modified Nodal Analysis (MNA) ......................... 11
      2.2.2 Time-Domain Simulation .................................. 16

3 **Transmission Line Macromodeling** .............................. 18
3.1 Method of Characteristics (MoC) ............................................................ 18
3.2 Macromodeling Issues ............................................................................. 21
  3.2.1 Stability ........................................................................................... 22
  3.2.2 Causality ......................................................................................... 22
  3.2.3 Passivity .......................................................................................... 23
  3.2.4 Efficiency ......................................................................................... 24
3.3 Matrix Rational Approximation (MRA) ............................................... 24
  3.3.1 Time-Domain Simulation Using MRA Macromodels ................ 29
  3.3.2 MRA Macromodel Properties ....................................................... 32
3.4 DEPACT: Delay-Extraction Based Passive Compact Transmission-Line
  Macromodeling Algorithm ......................................................................... 33
  3.4.1 Time-Domain Simulation Using DEPACT Macromodel ........... 35
  3.4.2 DEPACT Macromodel Properties ............................................... 38
3.5 Simulation Issues in the Presence of Driver and Load ......................... 38

4 Proposed Closed-form Model for Transient Analysis of MTL with
  Driver and Load Circuits ........................................................................ 40
  4.1 Interconnect Macromodel ..................................................................... 41
  4.2 Output Evaluation with Load ............................................................... 43
  4.3 Output Evaluation with Both Driver and Load .................................. 48
  4.4 Transient Response Computation ....................................................... 50
4.5 Analysis of Multiconductor Transmission Lines ...................................... 51

4.6 Numerical Examples ................................................................................... 53

4.6.1 Example 1: Single Transmission Line ................................... 53

4.6.2 Example 2: Single Transmission Line with Complex Driver and Load ............................................................................ 54

4.6.3 Example 3: Multiconductor Transmission Line Network . . . . 61

5 Performance Improvements and the Special Cases of the Proposed Algorithm 65

5.1 Fast Pole Computation for Low-Order Macromodels ............................ 66

5.1.1 Cubic Polynomials ....................................................................... 67

5.1.2 Quartic Polynomials .................................................................. 68

5.1.3 Quintic Polynomials .................................................................... 70

5.2 Broadband and Long Delay Interconnect Macromodeling Issues ...... 72

5.3 Distributed RC Interconnects ............................................................. 73

5.3.1 RC Distributed Transmission Line Macromodel ..................... 74

5.3.2 RC Network Properties ............................................................. 75

5.3.3 Output Evaluation with Capacitive Load ..................................... 76

5.3.4 Output Evaluation with Capacitive Load and Resistive Driver 80

5.3.5 Numerical Example .................................................................. 85

5.4 Distributed Lossless (LC) Interconnect .............................................. 91
5.4.1 LC Distributed Transmission Line Macromodel ........................................ 91
5.4.2 LC Network Properties ........................................................................... 93
5.4.3 Output Evaluation with Capacitive Load .............................................. 93
5.4.4 Output Evaluation with Load and Driver .............................................. 98
5.4.5 Numerical Example ............................................................................ 99

6 Conclusion .................................................................................................. 109

6.1 Summary ................................................................................................. 109

6.2 Suggestions for Future Research ............................................................ 111

A MRA approximation when L = 0 ............................................................... 113
List of Figures

1.1 High-speed interconnect effects ................................................................. 2

2.1 Lumped model section ................................................................................ 8
2.2 Illustrative comparison of transmission line response with lumped model for high-frequency inputs ......................................................... 9
2.3 Distributed model section ........................................................................... 10
2.4 Sample circuit to demonstrate MNA formulation ................................. 13
2.5 Sample circuit to demonstrate MNA formulation with distributed interconnects ............................................................... 16

3.1 Poles and zeros of scalar rational approximation in (3.16) .............. 27
3.2 DEPACT macromodel realization of (3.40) ............................................. 35

4.1 Transmission line ......................................................................................... 41
4.2 Transmission line with load ........................................................................ 44

vii
4.3 Poles of (4.7) for varying $C_L$ .................................................. 46
4.4 Poles of (4.7) for varying $G_L$ .................................................. 47
4.5 Transmission line with driver and load ............................................ 48
4.6 Multiconductor transmission line network ........................................ 51
4.7 Example 1: Pole movement in the expression for output transfer func-
    tion with varying load capacitance ............................................... 55
4.8 Example 1: Output response of interconnect with load = 1pF ............ 56
4.9 Example 1: Pole movement in the expression for output transfer func-
    tion with varying driver resistance ............................................... 57
4.10 Example 1: Transient response at the near end .............................. 58
4.11 Example 1: Transient response at the far end ............................... 58
4.12 Circuit of Example 2 ................................................................. 59
4.13 Example 2: Output response of interconnect with load = 1pF ............ 59
4.14 Example 2: Transient response at the near end .............................. 60
4.15 Example 2: Transient response at the far end ............................... 60
4.16 Coupled transmission line with driver and load for Example 3 .......... 61
4.17 Example 3: Transient response at near end of signal line, ($V_2$) ........ 62
4.18 Example 3: Transient response at near end of victim line, ($V_4$) ........ 63
4.19 Example 3: Transient response at far end of signal line, ($V_3$) ............ 63
4.20 Example 3: Transient response at far end of victim line, ($V_4$) ............ 64
5.1 Polynomial addition with real negative roots ............................................. 78
5.2 Illustration of polynomials representing the poles given in Table 5.1 ........ 79
5.3 \textit{Polybound}$_2$ boundaries and roots ...................................................... 83
5.4 \textit{Polybound}$_1$ boundaries and roots ...................................................... 83
5.5 Illustration of \textit{Polybound}$_1$ and \textit{Polybound}$_2$ with boundaries .......... 84
5.6 Denominator polynomial of (5.36) as a sum of bounding polynomials
\textit{Polybound}$_1$ and \textit{Polybound}$_2$ ................................................................. 84
5.7 Roots of \textit{Polybound}$_1$, in (5.40) (Iteration required = 1) .............. 86
5.8 Roots of \textit{Polybound}$_2$, in (5.41) (Iteration required = 1) .............. 87
5.9 Roots of the denominator of the output expression, in (5.39) (Iteration
required = 1) ....................................................................................................... 88
5.10 Comparison of transient response at far end (\(V_{\text{out}}\)) ................. 89
5.11 Comparison of transient response at near end (\(V_{\text{in}}\)) .................... 90
5.12 Poles of (4.8) for varying driver inductance \(L_D\) ......................... 100
5.13 Poles of (4.8) for varying driver resistance \(R_D\) ............................... 101
5.14 Roots of \textit{Polybound}$_1$ .......................................................... 103
5.15 Roots of \textit{Polybound}$_2$ .......................................................... 104
5.16 Roots of the denominator of the output expression ....................... 106
5.17 Transient response at far end (\(V_{\text{out}}\)) ................. 107
5.18 Transient response at near end (\(V_{\text{in}}\)) ............................... 108
List of Tables

4.1 Example 1: Comparison of computational cost (seconds) .......................... 54
4.2 Example 2: Comparison of computational cost (seconds) ......................... 59
4.3 Example 3: Comparison of computational cost (seconds) ......................... 62

5.1 Roots of Adding Polynomials with Alternating Roots .......................... 77
5.2 Comparative locations of boundaries and roots of \( Poly_{bound_1}(10^9) \) ........ 86
5.3 Comparative locations of boundaries and roots of \( Poly_{bound_2}(10^9) \) ........ 87
5.4 Comparative locations of roots of boundaries and poles of the output expression \( (10^9) \) ................................................................. 88
5.5 Poles and residues of output expressions of (5.36) and (5.37) \( (10^9) \) ........ 89
5.6 Comparison of computational cost (seconds) ........................................ 90
5.7 Imaginary roots of \( Poly_1 \) and \( Poly_2 \) .......................................... 96
5.8 Polynomial coefficients of \( Poly_1 \) and \( Poly_2 \) and \( Poly_{Sum} \) ........ 96
5.9 Coefficients of modified polynomials of Table 5.8 .............................. 97
5.10 Real roots of the modified polynomials of Table 5.9 .......... 97
5.11 Comparison of polynomial roots ........................................ 97
5.12 Comparative locations of boundaries and roots of $Poly_{bound1}(10^9)$ .. 102
5.13 Comparative locations of boundaries and roots of $Poly_{bound2}(10^9)$ .. 105
5.14 Output expression for pole boundaries and final pole locations ($10^9$) ... 105
5.15 Poles and residues of output expressions ($10^9$) ..................... 107
5.16 Comparison of computational cost (seconds) ......................... 108
Abbreviations

VLSI       Very Large Scale Integrated Circuit
RLCG       Resistance-Inductance-Capacitance-Conductance
MoC        Method of Characteristic
MRA        Matrix Rational Approximation
DEPACT      Delay-Extraction Based Passive Compact Transmission-Line Macro-modeling Algorithm
MNA        Modified Nodal Analysis
MTL        Multiconductor Transmission Line
Chapter 1

Introduction

1.1 Background and Motivation

Today's VLSI circuits are faced with rapidly increasing signal speeds, complex functionality, large circuit size and shrinking device sizes. Such rapid advances have increased the need for accurate, fast and economical methods for circuit simulation and design. With the evolving capabilities of VLSI circuits, the once negligible interconnects have become the major bottleneck for high-speed design and analysis.

Interconnect effects greatly impact the signal integrity in high-speed designs. Considerable signal degradation may be experienced due to effects such as propagation delay, attenuation, signal reflection, ringing, crosstalk, skin effect, etc. (Figure 1.1).
The varying physical properties of interconnects and their locations at all hierarchical design levels presents further challenges. For example, the interconnects can assume various forms, such as on-chip interconnects, packaging structures, printed circuit boards and vias, etc. Predicting interconnect behaviour early in the circuit development cycle has become a critical part of VLSI design success. Accurately capturing signal integrity issues at early stages of the design cycle, results in fewer product re-designs and re-fabrications, which otherwise can be time consuming and costly.

At very low operating frequencies, interconnects can be considered a short circuit without any significant problems. As the complexity of circuits increases so does the importance of anticipating interconnect behaviour. At medium frequencies, the interconnects can be modelled as lumped RC, RLC or RLGC models, where R, C, L and G represent the resistance, capacitance, inductance and conductance, respec-
tively. However, the lumped representation may lead to large circuit sizes, leading to excessive computational times in circuit simulations. As the frequency further increases, the interconnect behaves more like a transmission line, whose characteristics are governed by Telegrapher's equations. This poses significant problems for circuit simulators due to the mixed frequency/time analysis requirement. SPICE type simulators rely on generating the transient response by marching in time, while solving the circuit equations (ordinary differential equations) at each time point. However, the distributed interconnects are best described in the frequency-domain, leading to the mixed frequency/time analysis problem. Consequent to these issues, significant research has been reported in the literature to achieve transient analysis of distributed interconnect circuits via compact, passive time-domain macromodeling [1] - [2].

In order to address the above difficulties, an efficient algorithm for transient analysis of distributed transmission line models is presented in this thesis. Closed-form time-domain output expressions are developed for the desired output signals in pole-residue form including the load, driver and the interconnect macromodel. The new algorithm overcomes the need for using traditional circuit simulators for repetitive interconnect analysis.
1.2 Contributions

The objective of this thesis is to develop efficient algorithms for time-domain simulation of interconnect circuits, including the load and driver. The main contributions made in this thesis are:

1. Development of a generalized algorithm for fast transient analysis of distributed interconnects without performing SPICE-like circuit analysis. For this purpose, closed-form expressions for output functions including load, driver and interconnect macromodel are developed.

2. Application of innovative polynomial root finding methods for fast evaluation of output expressions in terms of poles and residues.

3. Special cases of the proposed algorithm are developed for analysis of RC type ($L = 0$) and for interconnects with no losses ($R = 0$, $G = 0$).

1.3 Organization of the Thesis

In Chapter 2, currently available methods to perform time-domain simulation, as well as their benefits and drawbacks are discussed. Current macromodeling techniques, MoC, MRA and DEPACT are described in Chapter 3. Chapter 4 introduces the proposed general algorithm for transient analysis of interconnect circuits with both
driver and load. Output expressions are obtained by combining the distributed interconnect macromodel, the driver and the load. The proposed algorithm is further explored for performance improvements in Chapter 5, where low order interconnect approximations, RC type transmission lines and lossless transmission lines are considered. Conclusions and suggestions for further research are given in Chapter 6.
Chapter 2

Interconnects

2.1 Interconnect Modelling

Due to the increased operating speed and complexity of today’s VLSI circuits, interconnect effects are becoming the major bottlenecks in the modelling and analyzing of high-speed designs. Interconnect effects such as ringing, reflection, attenuation, crosstalk, etc. greatly impact signal integrity. Interconnects can exist at any level of hierarchy, such as on chip, packaging structures and printed circuit boards and come in different physical shapes and structures, such as vias, wire bonds, solder balls. This, in turn, greatly increases the complexity of interconnect modelling and simulation.
An essential factor to consider when analyzing interconnects and generating models is its *electrical length*. Interconnects are considered to be *electrically short* if at highest operating frequency of interest, the interconnect is physically shorter than one-tenth of the wavelength [3]. That is:

$$\frac{l}{\lambda} < 0.1; \quad \lambda = \frac{v}{f}$$ \hspace{1cm} (2.1)

where $l$ is the interconnect length, $\lambda$ is the signal wavelength, $v$ is the propagation velocity and $f$ is the frequency. Otherwise, the interconnect is considered, *electrically long*. A practical relationship between maximum frequency and rise time of a signal is expressed as [4], [3]

$$f_{\text{max}} \approx \frac{0.35}{t_r}$$ \hspace{1cm} (2.2)

Depending on the operating frequency, signal rise and fall times, length of the interconnect and its physical properties, an appropriate interconnect model is selected. In recent history, extensive research and development has focused on deriving methods to incorporate the effects and complexities introduced by interconnects. Some models that are very accurate become computationally costly and slow for large circuits, while simpler models rely on various approximation methods that result in faster and cheaper simulation. This chapter provides an overview of current modelling techniques of interconnect networks.
2.1.1 Lumped Models

Lumped interconnect models can consist of small circuit segments, as one shown in Figure 2.1, where $R$, $L$, $C$ and $G$ are the resistance, inductance, capacitance and conductance of the interconnect, respectively. At low operating frequencies, the lumped model may consist of a simple RC tree. As the frequency increases, RC trees become insufficient and are replaced by more complex sections, as the one shown in Figure 2.1. In order to achieve a higher degree of accuracy, large number of sections may be required. Typically, each subsection must be related to the wavelength of the highest frequency of interest:

$$\Delta x = \frac{d}{n} << \lambda$$  \hspace{1cm} (2.3)

where $\Delta x$ is the length of the subsection, $d$ is the length of the entire interconnect, $n$ is the number of sections and $\lambda$ is the wavelength as described in equation (2.1). For example, using fewer lumped sections can lead to high error in the transient response.
of an interconnect circuit, as shown in Figure 2.2. For electrically long interconnects lumped circuit representation satisfying (2.3) can result in a large number of circuit nodes, therefore making the simulation highly CPU intensive.

Figure 2.2: Illustrative comparison of transmission line response with lumped model for high-frequency inputs

2.1.2 Distributed Models

As the frequency of operation increases, the interconnect length becomes a significant portion of the wavelength. As a result the interconnect becomes electrically long, as described in equation (2.1). Consequently, interconnects tend to act as transmission lines and are modeled, as shown in Figure 2.3, using per unit length parameters R,
L, G and C, where \( \omega = 2\pi f \) and \( f \) is the frequency. From this model, propagation constant \( \gamma \) and the characteristic impedance \( Z_0 \) are defined as

\[
\gamma = \sqrt{(R + j\omega L)(G + j\omega C)}
\]

\[
Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}
\]  

2.1.3 Full-Wave Models

As the operating speed increases and if the interconnect physical dimensions did not scale accordingly, then the cross-sectional dimensions of the interconnect becomes a significant portion of the wavelength. In such cases, full wave electromagnetic models become necessary. Simulation based on full-wave models is highly involved since it accounts for all field components and boundary conditions. Information obtained through this analysis is in terms of field parameters, such as propagation constant,
characteristic impedance, etc. Since circuit simulators require nodal voltages and circuit impedances, significant work must be performed to link and integrate the model into a circuit simulation.

Full-wave analysis consists of performing a large number of computations at every frequency point. Although highly accurate, this method is extremely CPU intensive and expensive for practical simulation purposes of today’s high-speed VLSI circuits.

2.2 Interconnect Simulation

Section 2.1 examined various interconnect models. This section focuses on simulation of interconnect models. The intent of this thesis is fast transient simulation of interconnect networks and hence this section briefly reviews the basics of time-domain simulation. This section considers the derivation of component stamps followed by formulation of MNA equations for SPICE simulators. It also discusses the interconnect modelling requirements for circuit simulation.

2.2.1 Modified Nodal Analysis (MNA)

Before interconnect simulation is considered, an overview of basic circuit simulation is provided. Note that circuits are represented by equations, in frequency or time-
domain, obtained using Kirchoff’s current law. MNA equations for linear networks are written in the time-domain as [3], [5]:

\[
W \dot{x}(t) + Gx(t) = b(t) \tag{2.5}
\]

\[
y = L^T x(t)
\]

and in the frequency-domain as

\[
(G + sW)X(s) = b(s) \tag{2.6}
\]

\[
Y(s) = L^T X(s)
\]

Frequency-domain expression can be used to obtain solutions at any frequency point. Time-domain simulation, however, consists of differential equations and the corresponding solution is more involved. Integration methods, such as trapezoidal rule, are adopted in order to obtain solutions at incrementing time steps. Depending on the nature of the circuit and the input signal, this simulation may be very slow due to computation at many time steps.

In order to demonstrate how these equations are formulated, a simple example is given. The circuit of Figure 2.4 is represented by the MNA equation (2.7).
As seen from (2.7), stamps of circuit elements are entered into appropriate matrices to formulate the circuit equations for simulation.

**Transmission Line Equations**

In order for the transmission line models to be included into the circuit equations, they must be written in a form that can be included into the MNA.

In the case when lumped interconnect models are used, the task of including the interconnect into the MNA is straightforward. The interconnect element is simply replaced by the lumped sections and RLC elements are stamped into the MNA. Al-
though simple, for a circuit with many interconnects, or interconnect models with many lumped sections, the size of the circuit equations can increase rapidly, making this a slow and inefficient simulation process.

When considering more sophisticated interconnect models, such as distributed transmission lines, may be used, its stamp must first be developed. Transmission line behaviour is governed by Telegrapher’s equations, described below in time (2.8) and frequency (2.9) as

\[
\begin{align*}
\frac{\partial}{\partial x} v(x,t) &= -Ri(x,t) - L \frac{\partial}{\partial t} i(x,t) \quad (2.8) \\
\frac{\partial}{\partial x} i(x,t) &= -Gv(x,t) - C \frac{\partial}{\partial t} v(x,t)
\end{align*}
\]

\[
\begin{align*}
\frac{\partial}{\partial x} V(x,s) &= - (R + sL) I(x,s) \quad (2.9) \\
\frac{\partial}{\partial x} I(x,s) &= - (G + sC) V(x,s)
\end{align*}
\]

where \( R \in \mathbb{R}^{N \times N}, L \in \mathbb{R}^{N \times N}, G \in \mathbb{R}^{N \times N}, C \in \mathbb{R}^{N \times N} \) are per-unit-length parameters of the transmission line, \( i(x,t) \in \mathbb{R}^N \) and \( v(x,t) \in \mathbb{R}^N \) are the current and voltage, respectively as a function of position on the line \( (x) \) and time \( (t) \) and \( I(x,s) \in \mathbb{R}^N \) and \( V(x,s) \in \mathbb{R}^N \) are the current and voltage as a function of position and frequency \( s = j\omega = j2\pi f \).
Equation (2.9) in a hybrid form becomes:

\[
\begin{bmatrix}
V(d, s) \\
I(d, s)
\end{bmatrix} = e^K
\begin{bmatrix}
V(0, s) \\
I(0, s)
\end{bmatrix}
\]

(2.10)

where \( K = (D + sE)d \), \( D = \begin{bmatrix} 0 & -R \\ -G & 0 \end{bmatrix} \) and \( E = \begin{bmatrix} 0 & -L \\ -C & 0 \end{bmatrix} \) and \( d \) is the interconnect length.

A stamp of a circuit element describes the relationship between port currents and voltages. Therefore, with some algebraic manipulations, the stamp is written as:

\[
\begin{bmatrix}
I(0, s) \\
-I(d, s)
\end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}
\begin{bmatrix}
V(0, s) \\
V(d, s)
\end{bmatrix}
\]

(2.11)

\[
= \begin{bmatrix} -T_{12}^{-1}T_{11} & T_{12}^{-1} \\ -T_{21} + T_{22}T_{12}^{-1}T_{11} & -T_{22}T_{12}^{-1} \end{bmatrix}
\begin{bmatrix}
V(0, s) \\
V(d, s)
\end{bmatrix}
\]

where \( e^K = T(s) = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \).

For the purpose of illustration, the simple example of Figure 2.4 is expanded to include a distributed interconnect in Figure 2.5. The MNA equation is given in (2.12).
2.2.2 Time-Domain Simulation

As discussed in previous sections, transient responses are obtained by computing the response at several time points. This process relies heavily on the number of time steps required by the circuit simulator. These factors depend on the nature of input signals and the nonlinearity of the circuit. For an input signal with sharp changes, smaller time steps may be required, in order to obtain the accurate response.
While analyzing high-frequency interconnects, such as distributed elements, transient analysis suffers from the mixed frequency/time difficulty. This is because the MTLs are best described in frequency-domain and do not have a direct representation in time-domain. The next chapter reviews various macromodeling techniques that are used to convert MTLs into time-domain described ordinary differential equations.
Chapter 3

Transmission Line Macromodeling

For the increasingly complex behaviour of interconnects, appropriate models are needed that efficiently and accurately capture their characteristics. This chapter reviews the current macromodeling techniques available for distributed interconnect analysis. In particular, it reviews the popularly used MoC, MRA and DEPACT macromodeling algorithms.

3.1 Method of Characteristics (MoC)

Method of characteristics was developed in the 1960s for the purpose of accurate modelling of distributed interconnects for transient analysis. Initially, MoC was developed for lossless transmission lines, and in the later years, it was expanded to include transmission lines with losses. The overview of this macromodel is provided in this section.
Interconnect behaviour is best described by Telegrapher’s equations, as shown in equation (2.9) in the frequency-domain. Using (2.11), an analytical solution can be derived as [3]:

\[
\begin{bmatrix}
    I_1 \\
    I_2
\end{bmatrix}
= \frac{1}{\frac{Z_0}{1-e^{-2\gamma d}}}
\begin{bmatrix}
    1+e^{-2\gamma d} & -2e^{-\gamma d} \\
    -2e^{-\gamma d} & 1+e^{-2\gamma d}
\end{bmatrix}
\begin{bmatrix}
    V_1 \\
    V_2
\end{bmatrix}
\] (3.1)

where the propagation constant \( \gamma \) and the characteristic impedance \( Z_0 \) are defined as

\[
\gamma = \sqrt{(R + j\omega L)(G + j\omega C)}
\]

\[
Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}
\] (3.2)

\( V_1 \) and \( I_1 \) represent the input voltage and current, respectively, while \( V_2 \) and \( I_2 \) represent the voltage and current leaving the transmission line. The equation (3.1) can be re-arranged to express the nodal voltages:

\[
V_1 = Z_0 I_1 + e^{-\gamma d}[2V_2 - e^{-\gamma d}(Z_0 I_1 + V_1)]
\] (3.3)

\[
V_2 = Z_0 I_2 + e^{-\gamma d}[2V_1 - e^{-\gamma d}(Z_0 I_2 + V_2)]
\]

In order to proceed, define \( W_1 \) and \( W_2 \) as:

\[
W_1 = e^{-\gamma d}[2V_2 - e^{-\gamma d}(Z_0 I_1 + V_1)]
\] (3.4)

\[
W_2 = e^{-\gamma d}[2V_1 - e^{-\gamma d}(Z_0 I_2 + V_2)]
\]

Now, (3.3) is combined with (3.4) to obtain:

\[
V_1 = Z_0 I_1 + W_1
\] (3.5)

\[
V_2 = Z_0 I_2 + W_2
\]
Combining of (3.5) and (3.4) results in:

\[ W_1 = e^{-\gamma d} [2V_2 - e^{-\gamma d} (2V_1 - W_1)] \quad (3.6) \]
\[ W_2 = e^{-\gamma d} [2V_1 - e^{-\gamma d} (2V_2 - W_2)] \]

Further manipulation of equations shown above, results in a recursive relationship between \( W_1 \) and \( W_2 \):

\[ W_1 = e^{-\gamma d} (2V_2 - W_2) \quad (3.7) \]
\[ W_2 = e^{-\gamma d} (2V_1 - W_1) \]

If the transmission line is lossless, then we have \( \gamma (s) = s\sqrt{LC} \) and \( Z_0 = \sqrt{L/C} \). In this case, \( \gamma \) will be purely imaginary while \( Z_0 \) is a real constant. In order to transform the expression into time-domain, the exponential in (3.7) is transformed into a delay element by taking the inverse Laplace transform. Consequently, \( e^{-\gamma d} \) and \( e^{-2\gamma d} \) expressions become time shifts of \((t - \gamma d)\) and \((t - 2\gamma d)\), respectively. Now (3.7) can be expressed in time-domain:

\[ w_1(t + \gamma d) = 2v_2(t) - w_2(t) \quad (3.8) \]
\[ w_2(t + \gamma d) = 2v_1(t) - w_1(t) \]
Next, using (3.8) and performing inverse Laplace transform of the expression in (3.1), a time-domain stamp of a lossless interconnect can be obtained as:

\[
Z_0 \left( \begin{bmatrix}
  i_1(t) \\
  i_2(t)
\end{bmatrix} - \begin{bmatrix}
  i_1(t - 2\gamma d) \\
  i_2(t - 2\gamma d)
\end{bmatrix} \right) = \begin{bmatrix}
  v_1(t) \\
  v_2(t)
\end{bmatrix} + \begin{bmatrix}
  v_1(t - 2\gamma d) - 2v_2(t - \gamma d) \\
  -2v_1(t - \gamma d) + v_2(t - 2\gamma d)
\end{bmatrix}
\]

(3.9)

This method works well for lossless transmission lines. However, if a line is not lossy this does not accurately represent transient behaviour, since the propagation constant is not purely imaginary and cannot be transformed as a simple delay element. As a result, \( w_1 \) and \( w_2 \) cannot be directly translated into time-domain expressions. A key concern of this macromodeling technique is that the passivity of the macromodel is not guaranteed. The concerns surrounding this problem will further be discussed in the following section.

### 3.2 Macromodeling Issues

The goal of macromodeling is to overcome the mixed frequency/time analysis associated with the distributed interconnect analysis. However, there are several macromodeling issues that influence the final transient analysis. They are discussed in detail in the following sections.
3.2.1 Stability

Interconnects are generally stable networks, and as such should have models that retain this property. For a circuit to be stable, all poles must be in the negative half of the s-plane. The transient response of an unstable circuit is characterized by its oscillatory nature. Luckily, it is easy to ensure the stability of a macromodel, as the poles on the right half of the s-plane can simply be discarded in the final macromodel.

3.2.2 Causality

The concept of causality simply states that an effect cannot precede its cause. Therefore, every physical system must be causal. A system is defined to be causal if and only if for all input pairs \( x_1(t) \) and \( x_2(t) \) [6]

\[
x_1(t) = x_2(t), \quad t \leq t_0; \quad \forall t_0
\]

such that the two corresponding outputs satisfy

\[
w_1(t) = w_2(t); \quad t \leq t_0
\]

where \( w(t) = h(t) \ast x(t) = \int_{-\infty}^{\infty} h(t - \tau) x(\tau) d\tau \) and \( h(t) \) is the system impulse response, with each element \( h_{ij} \) being the response at port \( i \) when an ideal impulse (Dirac’s delta function) is applied at port \( j \), with all other inputs set to zero.
The general definition is simplified for the case of a linear, time-invariant (LTI) systems such that, a linear system is causal if and only if for every input $x(t)$ that vanishes for $t \leq t_0$, the corresponding output $w(t)$ vanishes too for $t \leq t_0$.

This condition imposes a constraint on the impulse response of linear, time-invariant (LTI) systems, such that all elements $h_{ij}(t)$ of the impulse response matrix $h(t)$ are vanished for $t < 0$

$$h(t) = 0 \quad t < 0 \quad (3.10)$$

The above definition of causality is general and can be applied to both lumped and distributed systems. For distributed systems, it may be required to develop a strategy for explicitly considering propagation delay. It is also be noted that all passive systems are guaranteed to be causal.

### 3.2.3 Passivity

Interconnects do not generate energy, that is, they are passive circuits. Therefore, interconnect model passivity must be maintained. It is not easy to verify and ensure passivity, but is of great importance. If a model does not take passivity into account, serious simulation problems can occur. This is because, when a non-passive interconnect model is combined with a passive termination, the resulting network could become unstable, generating faulty simulation results [1] [3] [7] [8].
A linear n-port network, with admittance matrix $Y(s)$ is passive, if and only if [9]:

- $Y(s^*) = Y^*(s)$, where $'$ indicates the complex conjugate operator
- $Y(s)$ is a positive real matrix, or $z^* [Y^T(s^*) + Y(s)] z \geq 0$, where $\text{Re}(s) > 0$ and for any arbitrary $z$

Ensuring macromodel passivity allows for passive models with passive terminations to create a guaranteed stable overall network.

### 3.2.4 Efficiency

The efficiency of simulation involving macromodels are significantly influenced by the order and sparsity of macromodels. Combining the elements generated from large-order interconnect macromodels with MNA equations may result in an excessively large circuit. Simulation of these circuits would be too CPU intensive and slow.

### 3.3 Matrix Rational Approximation (MRA)

As discussed in section 2.1, distributed interconnects are described by partial differential equations (2.8). The solution to (2.8) can be expressed in a matrix exponential and hybrid form as:
\[
\begin{bmatrix}
V(d, s) \\
-I(d, s)
\end{bmatrix} = e^K 
\begin{bmatrix}
V(0, s) \\
I(0, s)
\end{bmatrix}
\]

(3.11)

where \( K = \begin{bmatrix} 0 & -a \\ -b & 0 \end{bmatrix} \) and \( a = (R + sL)d, \ b = (G + sC)d \), \( d \) is the interconnect length and \( V, I \) are the frequency-domain terminal voltage and current vectors, respectively. The hybrid form of the transmission line stamp described in equation (3.11) does not have a direct representation in time-domain, making it difficult to interface with non-linear simulators. In this section, a macromodel based on matrix rational approximation will be described that approximates the interconnect equation. The important issue of passivity is addressed in this macromodel.

MRA approximates the exponential function (3.11) as a closed-form rational function using pre-determined coefficients. In scalar form, an exponential expression can be approximated by the following rational function:

\[
\sum_{i=0}^{N} \psi_{Ni} k^i 
\]

where the coefficients \( \psi_{Ni} \) and \( \psi_{Mi} \) are known \textit{a priori} via closed-form Padé relations for any order \( N \) and \( M \), as shown:

\[
\psi_{Ni} = \frac{(N + M - i)!N!}{(N + M)!i!(N - i)!}, \quad \psi_{Mi} = \frac{(N + M - i)!M!}{(N + M)!i!(M - i)!}
\]

(3.13)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In order to apply this function to the problem of interconnect modelling, the scalar $k$ is replaced by the matrix $K$ from (3.11), and the scalar equation (3.12) is expanded to a matrix form:

$$\Psi_M(K) \approx e^K \Psi_N(K)$$  \hspace{1cm} (3.14)

Matrices $\Psi_N(K)$ and $\Psi_M(K)$ can then be written as:

$$\Psi_N(K) = \sum_{i=0}^{N} \psi_{Ni}K^i; \quad \Psi_M(K) = \sum_{i=0}^{M} \psi_{Mi}K^i$$  \hspace{1cm} (3.15)

The exponential matrix is now represented by a rational function and subsequently, the interconnect macromodel can be defined. In order to ensure model passivity, further considerations are required.

As described in section 3.2, passivity is one of the key concerns when generating a macromodel. In order to ensure passivity of the model, the polynomials describing the rational function approximation must be strictly Hurwitz [10], which can be ensured by selecting the numerator and denominator to be same, as follows

$$e^k \approx \frac{\Psi_N(k)}{\Psi_N(-k)} = \frac{\sum_{i=0}^{N} \psi_i k^i}{\sum_{i=0}^{N} \psi_i (-k)^i}$$  \hspace{1cm} (3.16)

For a strictly Hurwitz polynomial, $\Psi_N(k)$ all roots must appear on the left half plane. Since zeros of the rational function are defined as the roots of $\Psi_N(-k)$, they are
expected to be all in the right plane, and mirror images of the poles. This condition is shown graphically in Figure 3.1.

![Figure 3.1: Poles and zeros of scalar rational approximation in (3.16)](image)

Figure 3.1: Poles and zeros of scalar rational approximation in (3.16)

Applying this condition to the macromodel equation (3.14), modified expressions are obtained:

\[
\psi_i = \frac{(2N + i)!N!}{(2N)!i!(N - i)!} \tag{3.17}
\]

\[
\Psi_N (-K) \approx e^K \Psi_N (K) \tag{3.18}
\]

\[
\Psi_N (-K) \begin{bmatrix} V(d, s) \\ -I(d, s) \end{bmatrix} = \Psi_N (K) \begin{bmatrix} V(0, s) \\ I(0, s) \end{bmatrix} \tag{3.19}
\]
where

$$\Psi_N(\mathbf{K}) = \begin{bmatrix} \Psi_{N11} & \Psi_{N12} \\ \Psi_{N21} & \Psi_{N22} \end{bmatrix}$$

\begin{align*}
&= \sum_{i=0}^{N} \psi_i \mathbf{K}^i \\
&= \begin{bmatrix}
\sum_{i=0\text{(Even)}}^{N} \psi_i (ab)^{i/2} & - \sum_{i=0\text{(Odd)}}^{N} \psi_i (ab)^{i-1/2} a \\
- \sum_{i=0\text{(Odd)}}^{N} \psi_i (ba)^{i-1/2} b & \sum_{i=0\text{(Even)}}^{N} \psi_i (ba)^{i/2}
\end{bmatrix}
\end{align*}$$

With the obtained expressions, Y parameters of the model are now expressed as:

\begin{align*}
\begin{bmatrix}
\mathbf{I}(0, s) \\
\mathbf{I}(d, s)
\end{bmatrix}
&= \begin{bmatrix}
\mathbf{Y}_{11} & \mathbf{Y}_{12} \\
\mathbf{Y}_{21} & \mathbf{Y}_{22}
\end{bmatrix}
\begin{bmatrix}
\mathbf{V}(0, s) \\
\mathbf{V}(d, s)
\end{bmatrix} \\
\begin{bmatrix}
\mathbf{Y}_{11} & \mathbf{Y}_{12} \\
\mathbf{Y}_{21} & \mathbf{Y}_{22}
\end{bmatrix}
&= \mathbf{X}_1 + \mathbf{X}_2
\end{align*}
where

\[
X_1 = \begin{bmatrix}
\frac{1}{2} [\Psi_{N_{12}}]^{-1} \Psi_{N_{11}} & -\frac{1}{2} [\Psi_{N_{12}}]^{-1} \Psi_{N_{11}} \\
-\frac{1}{2} [\Psi_{N_{12}}]^{-1} \Psi_{N_{11}} & \frac{1}{2} [\Psi_{N_{12}}]^{-1} \Psi_{N_{11}} \\
\end{bmatrix}
\]

\[
X_2 = \begin{bmatrix}
\frac{1}{2} [\Psi_{N_{22}}]^{-1} \Psi_{N_{21}} & \frac{1}{2} [\Psi_{N_{22}}]^{-1} \Psi_{N_{21}} \\
\frac{1}{2} [\Psi_{N_{22}}]^{-1} \Psi_{N_{21}} & \frac{1}{2} [\Psi_{N_{22}}]^{-1} \Psi_{N_{21}} \\
\end{bmatrix}
\]  

(3.23)

In order to perform time-domain simulation using the MRA macromodel, described in frequency-domain, it must be linked with a SPICE-like nonlinear simulator, by realizing it in terms of circuit elements or ordinary differential equations.

### 3.3.1 Time-Domain Simulation Using MRA Macromodels

In order to simulate the interconnect approximated by MRA, time-domain realization is required. Time-domain macromodel is derived by expressing the MRA in terms of products of low-order rational functions. These will then be realized in terms of circuit elements. The macromodel scalar expression in (3.12) is written in terms of poles and zeros:
if $N$ is even
\[
de^x \approx \prod_{i=1}^{N/2} \frac{[(\alpha_i + x)(\alpha_i^* + x)]}{[(\alpha_i - x)(\alpha_i^* - x)]}
\]  
(3.24)
if $N$ is odd
\[
de^x \approx \frac{(\alpha_0 + x) \prod_{i=1}^{(N-1)/2} [(\alpha_i + x)(\alpha_i^* + x)]}{(\alpha_0 - x) \prod_{i=1}^{(N-1)/2} [(\alpha_i - x)(\alpha_i^* - x)]}
\]  
(3.25)
where $\alpha_i = x_i + jy_i$ are complex roots for $i > 0$, $\alpha_0$ is a real root and * represents the complex conjugate operator.

For macromodeling and circuit realization, above expressions (3.25) and (3.19) are combined (to obtain formulation that allows for circuit realization) as follows:

\[
\begin{bmatrix}
V(d,s) \\
-I(d,s)
\end{bmatrix}
\approx [\Psi_N(-K)]^{-1} \Psi_N(K) 
\begin{bmatrix}
V(0,s) \\
I(0,s)
\end{bmatrix}
\]  
(3.26)
if $N$ is even
\[
[\Psi_N(-K)]^{-1} \Psi_N(K) = \prod_{i=1}^{N/2} [((\alpha_i U - K)(\alpha_i^* U - K))^{-1} [((\alpha_i U + K)(\alpha_i^* U + K)]
\]  
(3.27)
if $N$ is odd
\[
[\Psi_N(-K)]^{-1} \Psi_N(K) = [((\alpha_0 U - K)^{-1}(\alpha_0 U + K)]
\times \prod_{i=1}^{(N-1)/2} [((\alpha_i U - K)(\alpha_i^* U - K))^{-1} [((\alpha_i U + K)(\alpha_i^* U + K)]
\]  
(3.28)
where \( \mathbf{U} \) is the identity matrix, \( \alpha_0 = x_i + jy_i \) and \( \alpha_i^* = x_i - jy_i \) are obtained from scalar approximation of (3.25). In order to simulate the model, circuit representation is created as a series of subsections, where each \( i^{th} \) subsection is written as:

\[
\begin{bmatrix}
\mathbf{V}_{i+1} \\
-\mathbf{I}_{i+1}
\end{bmatrix} = [\Psi_{N_i}(-\mathbf{K})]^{-1} [\Psi_{N_i}(\mathbf{K})] \begin{bmatrix}
\mathbf{V}_i \\
\mathbf{I}_i
\end{bmatrix}
\] (3.29)

Each subsection of the MRA macromodel consists of two pole-zero pairs derived from the scalar approximation (3.25):

\[
[\Psi_N(-\mathbf{K})]_i = [(\alpha_i \mathbf{U} - \mathbf{K})(\alpha_i^* \mathbf{U} - \mathbf{K})]
\]

\[
= \begin{bmatrix}
\mathbf{ab} + (x_i^2 + y_i^2) \mathbf{U} & 2x_i \mathbf{a} \\
2x_i \mathbf{b} & \mathbf{ab} + (x_i^2 + y_i^2) \mathbf{U}
\end{bmatrix}
\] (3.30)

\[
[\Psi_N(\mathbf{K})]_i = [(\alpha_i \mathbf{U} + \mathbf{K})(\alpha_i^* \mathbf{U} + \mathbf{Z})]
\]

\[
= \begin{bmatrix}
\mathbf{ab} + (x_i^2 + y_i^2) \mathbf{U} & -2x_i \mathbf{a} \\
-2x_i \mathbf{b} & \mathbf{ab} + (x_i^2 + y_i^2) \mathbf{U}
\end{bmatrix}
\] (3.31)

and one real pole-zero pair:

\[
[\Psi_N(-\mathbf{K})]_0 = [\alpha_0 \mathbf{U} - \mathbf{K}] = \begin{bmatrix}
\alpha_0 \mathbf{U} & \mathbf{a} \\
\mathbf{b} & \alpha_0 \mathbf{U}
\end{bmatrix}
\] (3.32)

\[
[\Psi_N(\mathbf{K})]_0 = [\alpha_0 \mathbf{U} + \mathbf{K}] = \begin{bmatrix}
\alpha_0 \mathbf{U} & -\mathbf{a} \\
-\mathbf{b} & \alpha_0 \mathbf{U}
\end{bmatrix}
\] (3.33)
Each subsection, as defined in (3.29), is converted to Y-parameter formulation:

\[ Y_{1i} = Y_{2i} = \frac{\left(x_i^2 + y_i^2\right)}{4x_i^2} a^{-1} + \frac{1}{4x_i} b + x_i (a + (x_i^2 + y_i^2) b^{-1})^{-1} \]

and for the real pole-zero pair:

\[ Y_{10} = Y_{20} = \frac{a_0}{2} a^{-1} + \frac{1}{2a_0} b \]

\[ Y_{12} = Y_{21} = -\frac{a_0}{2} a^{-1} + \frac{1}{2a_0} b \]  

MRA time-domain macromodel is formulated by first computing the poles and zeros of rational approximation function, then using those values to generate expressions for the Y-parameters of the macromodel. Once Y-parameters are obtained, circuit definition becomes straight-forward, and the circuit can be incorporated into a SPICE type simulator.

### 3.3.2 MRA Macromodel Properties

As mentioned in section 3.2, passivity is essential for reliable macromodeling of high-speed interconnects. The great advantage introduced by MRA is that it is passive by construction, ensuring that when this model is combined with linear terminations, a stable network will result.
A major drawback of the MRA method is that for interconnects with large delays, such as long lossy lines, high-orders of approximation may be required causing the simulation to be too slow and CPU intensive.

### 3.4 DEPACT: Delay-Extraction Based Passive Compact Transmission-Line Macromodeling Algorithm

A further improvement to the MRA macromodel was made by extracting the delay from the exponential matrix [11], [12]. With this development, delay is extracted prior to performing MRA, requiring a much lower order. A lower order of approximation results in a smaller equivalent circuit and it improves the efficiency of the simulation. Although great improvements are made toward reducing the order of interconnect models, therefore minimizing the size of the simulation matrix, the time-domain simulation is still performed at many time steps, in SPICE type simulators.

DEPACT method performs this by approximating the exponential matrix, via mod-
ified Lie approximation, as follows:

\[ e^K = e^{A+sB} \]  \hspace{1cm} (3.36)

where \( A(s) = \begin{bmatrix} 0 & -R \\ -G & 0 \end{bmatrix} \) and \( B(s) = \begin{bmatrix} 0 & -L \\ -C & 0 \end{bmatrix} \)

Lie product is given as \( \prod_{i=1}^{m} \Psi_i \), \( m \) representing the number of sections and

\[ \Psi_i = e^{sB/m} e^{A/m} \]  \hspace{1cm} (3.37)

such that the product converges to \( e^{A+sB} \) as \( m \to \infty \).

This is modified such that:

\[ \Psi_i = e^{A/2m} e^{sB/m} e^{A/2m} \]  \hspace{1cm} (3.38)

where \( e^{A/2m} \) are lossy sections to be modelled with MRA and \( e^{sB/m} \) are the delay components. Another modification to the Lie formula is given:

\[ \Psi_i = e^{sB/2m} e^{A/m} e^{sB/2m} \]  \hspace{1cm} (3.39)

This formulation becomes attractive in cases where long delays exist, such as for long low-loss lines. For frequency dependent interconnect parameters, the model is represented as

\[ e^{A(s)+sB(s)} \approx \prod_{i=1}^{m} \Psi_i = \prod_{i=1}^{m} e^{sB/2m} e^{(A(s)+sB_a(s))/m} e^{sB/2m} \]  \hspace{1cm} (3.40)
3.4.1 Time-Domain Simulation Using DEPACT Macromodel

In order to simulate the interconnect approximated by the DEPACT macromodel in time-domain, circuit realization is required. Time-domain macromodel is derived by approximating the transmission line stamp with a product of exponential matrices. A cascade of lossless and lossy subnetworks is generated to implement the DEPACT macromodel into a SPICE-like circuit simulator, as shown in Figure 3.2.

![Diagram of DEPACT macromodel realization](Figure 3.2: DEPACT macromodel realization of (3.40))

In order to realize the lossless sections, distributed voltage and currents of the
interconnect can be represented using Telegrapher’s equations

\[
\frac{\partial}{\partial x} v(x, t) = -L_m \frac{\partial}{\partial t} i(x, t) \tag{3.41}
\]

\[
\frac{\partial}{\partial x} i(x, t) = -C_m \frac{\partial}{\partial t} v(x, t)
\]

where

\[
L_m = \frac{L(\infty)}{2m}; \quad C_m = \frac{C(\infty)}{2m} \tag{3.42}
\]

Propagating modal voltages \( \hat{v} \) and currents \( \hat{i} \) are defined:

\[
v(x, t) = E_v \hat{v}(x, t); \quad i(x, t) = E_i \hat{i}(x, t) \tag{3.43}
\]

where \( E_v \) and \( E_i \) are transformation matrices, such that \( C_m \) and \( L_m \) are diagonalized:

\[
E_v^{-1} L_m E_v = \hat{L}; \quad E_i^{-1} C_m E_v = \hat{C} \tag{3.44}
\]

Using the above expressions, the lossless sections may now be described by the modal voltages and currents

\[
\frac{\partial}{\partial x} \hat{v}(x, t) = -\hat{L} \frac{\partial}{\partial t} \hat{i}(x, t) \tag{3.45}
\]

\[
\frac{\partial}{\partial x} \hat{i}(x, t) = -\hat{C} \frac{\partial}{\partial t} \hat{v}(x, t)
\]

where \( \hat{L} \) and \( \hat{C} \) are defined as diagonal matrices

\[
\hat{L} = diag \{ \hat{l}_1, \hat{l}_2, \ldots, \hat{l}_p \}; \quad \hat{C} = diag \{ \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_p \} \tag{3.46}
\]

and \( P \) denotes the number of ports of the system.
Now, modal propagation constant \( \gamma_i \) and characteristic impedance \( \hat{Z}_0 \) are defined as
\[
\gamma_i = s\sqrt{\frac{\mu_i}{\epsilon_i}}; \quad \hat{Z}_0 = \sqrt{\frac{\mu_i}{\epsilon_i}}
\] (3.47)

As described by the MoC algorithm (section 3.1) for lossless interconnects, the solution to (3.45) can be obtained using (3.47)
\[
\dot{V}_1 - \hat{Z}_0 \dot{I}_1 = W_1; \quad \dot{V}_2 - \hat{Z}_0 \dot{I}_2 = W_2
\] (3.48)
\[
W_1 = \gamma \left( 2\dot{V}_2 - W_2 \right); \quad W_2 = \gamma \left( 2\dot{V}_1 - W_1 \right)
\]

where \( \dot{V}_1, \dot{I}_1, \dot{V}_2, \) and \( \dot{I}_2 \) are the near and far end modal voltages and currents of lossless sections, \( \hat{Z}_0 = diag \{ \hat{Z}_{0_1}, \hat{Z}_{0_2}, \ldots, \hat{Z}_{0_p} \} \) and \( \gamma = diag \{ e^{-\gamma_1}, e^{-\gamma_2}, \ldots, e^{-\gamma_p} \} \).

Using the expressions of frequency-domain modal voltages (3.48), time-domain modal voltage expressions are obtained as
\[
\dot{v}_1 (t) - \hat{Z}_0 \dot{i}_1 (t) = w_1 (t)
\] (3.49)
\[
\dot{v}_2 (t) - \hat{Z}_0 \dot{i}_2 (t) = w_2 (t)
\]
\[
w_1 (t) = 2\dot{v}_2 (t - \tau) - w_2 (t - \tau)
\]
\[
w_2 (t) = 2\dot{v}_1 (t - \tau) - w_1 (t - \tau)
\]

The circuit is implemented by representing the obtained time-domain expressions (3.49) as time-domain dependent delay sources and lumped impedances.

Lossy sections are realized by implementing the passive MRA algorithm for the approximation of exponential term \( e^{A/m} \). \( \gamma \)-parameters of the lossy network, describing
the nodal voltages and currents, are obtained as described in section 3.3. Due to the purely lossy nature of this realization, resistive networks are obtained to represent the lossy sections of DEPACT macromodel realization.

3.4.2 DEPACT Macromodel Properties

Section 3.2 described the importance of issues such as passivity, causality and stability. All these interconnect network properties are preserved by the DEPACT macromodel. As noted earlier, one major cause of MRA macromodel inefficiency is the required high-order approximation to model long delay interconnects. By extracting the delay, DEPACT macromodel has reduced the order of the system that is to be modelled, therefore significantly reducing the order of the final macromodel.

3.5 Simulation Issues in the Presence of Driver and Load

Recent improvements in the area of time-domain macromodeling have succeeded in producing more compact interconnect representations. However, once drivers and receivers are included, traditional time-domain simulation is performed. This requires
formulating the circuit equations at each time point with driver, load and macro-model. These are subsequently solved for every time point. As can be expected, the solution of large set of circuit equations at every time point can result in slow and inefficient transient simulation. This problem is significantly pronounced when repetitive simulations are performed, which is typical for the case of interconnect analysis, such as eye diagram simulation.
Chapter 4

Proposed Closed-form Model for Transient Analysis of MTL with Driver and Load Circuits

As discussed in Chapters 1 and 2, interconnect effects are becoming the primary source of concern for time-domain simulation of high-speed VLSI circuits. In addition to the growing need for optimal interconnect designs, repetitive solutions of interconnect networks with varying loads and drivers are becoming increasingly necessary. In order to address these issues, an efficient algorithm is developed in this chapter that generates pole-residue type rational function models of multiconductor distributed interconnects including the driver and load. An appropriate closed-form
distributed transmission line macromodel is selected and combined with driver and load to obtain the output expression.

The development of the algorithm will be presented by first describing the transmission line macromodel, in section 4.1, and then combining it with a load, in section 4.2, and driver, in section 4.3. Properties of the output expression in time-domain are discussed in section 4.4. The proposed method is extended to multiconductor transmission line networks in section 4.5. Numerical examples are presented to demonstrate the accuracy, efficiency and validity of the proposed algorithm in section 4.6.

4.1 Interconnect Macromodel

![Distributed interconnect diagram](image)

Figure 4.1: Transmission line

Distributed interconnect (Figure 4.1) behaviour is described using a set of Telegrapher's equations. Solving these equations in time-domain is highly costly and inefficient. This lead to the development of numerous transmission line models and
methods for simulation, which is briefly reviewed in Chapter 2 and Chapter 3. For the proposed algorithm in this thesis, transmission line behaviour is described by a macromodel based on matrix rational approximation (MRA) [13]. This model guarantees the passivity and stability of the resulting network, as well as resulting in a high degree of accuracy.

As discussed in Chapter 3, the MRA interconnect macromodel is obtained by approximating the exponential matrix of the expression in (3.11) as described in (3.16) - (3.22). Using these expressions, final form of the MTL MRA macromodel in (3.21) - (3.22), for the case of a single transmission line can be written as

\[
\begin{bmatrix}
I_{in} \\
I_{out}
\end{bmatrix}
= \begin{bmatrix}
Y_{11} & Y_{12} \\
Y_{21} & Y_{22}
\end{bmatrix}
\begin{bmatrix}
V_{in} \\
V_{out}
\end{bmatrix}
\]

\[
= \frac{1}{2\Psi_{N_{12}}\Psi_{N_{22}}}
\times
\begin{bmatrix}
\Psi_{N_{11}}\Psi_{N_{22}} + \Psi_{N_{12}}\Psi_{N_{21}} & -\Psi_{N_{11}}\Psi_{N_{22}} + \Psi_{N_{12}}\Psi_{N_{21}} \\
-\Psi_{N_{11}}\Psi_{N_{22}} + \Psi_{N_{12}}\Psi_{N_{21}} & \Psi_{N_{11}}\Psi_{N_{22}} + \Psi_{N_{12}}\Psi_{N_{21}}
\end{bmatrix}
\begin{bmatrix}
V_{in} \\
V_{out}
\end{bmatrix}
\]  

(4.1)

where

\[
\Psi_N(K) = \begin{bmatrix}
\Psi_{N_{11}} & \Psi_{N_{12}} \\
\Psi_{N_{21}} & \Psi_{N_{22}}
\end{bmatrix}
= \begin{bmatrix}
\sum_{i=0(Even)}^{N} \psi_i(ab)^{\frac{i}{2}} - \sum_{i=0(Odd)}^{N} \psi_i(ab)^{\frac{i-1}{2}} a \\
- \sum_{i=0(Odd)}^{N} \psi_i(ba)^{\frac{i-1}{2}} b + \sum_{i=0(Even)}^{N} \psi_i(ba)^{\frac{i}{2}}
\end{bmatrix}
\]  

(4.2)
with \( a = R + sL \) and \( b = G + sC \).

With respect to the above expression, the following observations can be made:

- The poles of the interconnect Y-parameters consist of the poles of \( \Psi_{N12} \) and \( \Psi_{N22} \) and the diagonal zeros are the roots of \( \Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21} \) and off-diagonal zeros are found to be roots of \( -\Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21} \).

- The number of poles of the interconnect model is \( O(\Psi_{N12}) + O(\Psi_{N22}) = 2N - 1 \), as expected.

- From (4.1) and (4.2), it can be noted that for single transmission lines

\[
\Psi_{N11} = \Psi_{N22} \quad (4.3)
\]

And since

\[
\Psi_{N12} = - \sum_{i=0(Odd)}^{N} \psi_i (ab)^{i-1} a; \quad \Psi_{N21} = - \sum_{i=0(Odd)}^{N} \psi_i (ab)^{i-1} b \quad (4.4)
\]

it is noted that

\[
\Psi_{N12} b = \Psi_{N21} a \quad (4.5)
\]

\[
\Psi_{N12}(G + sC) = \Psi_{N21}(R + sL)
\]

4.2 Output Evaluation with Load

In this section, an expression for the output voltage is derived in the presence of both the macromodel (4.1) and the load (Figure 4.2).
Figure 4.2: Transmission line with load

Let the load of the transmission line be described by its admittance, $Y$:

$$-I_{out} = Y_{Load} V_{out} \quad (4.6)$$

Now, equation (4.1) is combined with (4.6) in order to include the transmission line load circuit.

$$\begin{bmatrix} I_{in} \\ -Y_{Load} V_{out} \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_{in} \\ V_{out} \end{bmatrix}$$

$$= \frac{1}{2\Psi_{N_{12}} \Psi_{N_{22}}} \times \begin{bmatrix} \Psi_{N_{11}} \Psi_{N_{22}} + \Psi_{N_{12}} \Psi_{N_{21}} & -\Psi_{N_{11}} \Psi_{N_{22}} + \Psi_{N_{12}} \Psi_{N_{21}} \\ -\Psi_{N_{11}} \Psi_{N_{22}} + \Psi_{N_{12}} \Psi_{N_{21}} & \Psi_{N_{11}} \Psi_{N_{22}} + \Psi_{N_{12}} \Psi_{N_{21}} \end{bmatrix} \begin{bmatrix} V_{in} \\ V_{out} \end{bmatrix} \quad (4.7)$$

Next, (4.7), is re-written to express the output voltage in terms of the input voltage as

$$V_{out} = \frac{\Psi_{N_{11}} \Psi_{N_{22}} - \Psi_{N_{12}} \Psi_{N_{21}}}{\Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}} + 2\Psi_{N_{12}} \Psi_{N_{22}} Y_L} V_{in} \quad (4.8)$$
The above output expression possesses the following characteristics:

- The zeros are identical to zeros of the off-diagonal Y-parameter entries of the transmission line macromodel in (3.21).

- In the absence of a load, the poles of the output expression are identical to the zeros of the diagonal Y-parameter entries (but negative) of the transmission line macromodel (4.1). That is

\[ V_{out} = -\frac{\text{zeros}(Y_{\text{off-diagonal}})}{\text{zeros}(Y_{\text{diagonal}})} V_{in} \]  \hspace{1cm} (4.9)

- As the admittance of the loading circuit increases, the poles move toward the roots of \(2\Psi_{N12}\Psi_{N22} Y_L\) which consists of a combination of the poles of the transmission line Y-parameters and the roots of the load admittance expression. This is shown for capacitive \((C_L)\) and conductive \((G_L)\) load circuits in Figure 4.3 and Figure 4.4, respectively. From this observation, general pole movement patterns can be observed, when load circuit consists of a C or G element. For such circuits, pole finding improvements are made by improving the starting pole guess for the iterative pole search, therefore reducing the number of iterations required to determine the expression poles.
Figure 4.3: Poles of (4.7) for varying $C_L$
Figure 4.4: Poles of (4.7) for varying $G_L$
4.3 Output Evaluation with Both Driver and Load

The method used to obtain the output expression in section 4.2 will be extended to obtain the output expression for an interconnect circuit when both driver and load are present, as in Figure 4.5.

Let the driver for the transmission line circuit be described by its Z-parameter.

\[
I_{in} = Z_{Driver}^{-1} (V_s - V_{in})
\]  

(4.10)

Now, equation (4.7) is modified to include the driver parameters:

\[
\begin{bmatrix}
Z_{Driver}^{-1} (V_s - V_{in}) \\
-Y_{Load} V_{out}
\end{bmatrix}
= 
\begin{bmatrix}
Y_{11} & Y_{12} \\
Y_{21} & Y_{22}
\end{bmatrix}
\begin{bmatrix}
V_{in} \\
V_{out}
\end{bmatrix}
\]

\[
= \frac{1}{2\Psi_{N12} \Psi_{N22}}
\]

\[
\times
\begin{bmatrix}
\Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21} & -\Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21} \\
-\Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21} & \Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21}
\end{bmatrix}
\begin{bmatrix}
V_{in} \\
V_{out}
\end{bmatrix}
\]

Figure 4.5: Transmission line with driver and load

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
This expression, (4.11), is re-arranged to obtain the expression describing the output voltage in terms of the input signal:

\[
V_{out} = \frac{\psi_{N_{11}} \psi_{22} - \psi_{N_{12}} \psi_{N_{21}}}{((\psi_{N_{12}} \psi_{N_{21}} + \psi_{N_{11}} \psi_{N_{22}}) Y_L + 2\psi_{N_{11}} \psi_{N_{21}}) Z_D + \psi_{N_{12}} \psi_{N_{22}} + \psi_{N_{11}} \psi_{N_{22}} + 2\psi_{N_{12}} \psi_{N_{22}} Y_L} V_s \quad (4.12)
\]

Another manipulation produces the near-end response:

\[
V_{in} = \frac{\psi_{N_{11}} \psi_{N_{22}} + \psi_{N_{21}} \psi_{N_{12}} - 2\psi_{N_{12}} \psi_{N_{22}} Y_L}{[(\psi_{N_{11}} \psi_{N_{22}} + \psi_{N_{12}} \psi_{N_{21}}) Y_L + 2\psi_{N_{11}} \psi_{N_{21}}] Z_D + \psi_{N_{11}} \psi_{N_{22}} + \psi_{N_{12}} \psi_{N_{21}} + 2\psi_{N_{12}} \psi_{N_{22}} Y_L} V_s \quad (4.13)
\]

As in section 4.2, the expressions are analyzed in order to improve the method used to compute the obtained expression:

- Zeros of (4.8) are identical to the zeros of (4.12), which are still the zeros of the off-diagonal entries of the transmission line Y-parameter matrix.
- In the absence of a driver admittance, (4.12) reduces to (4.8), as expected.
- Combining (4.3) and (4.5) with the obtained output expression (4.12), an alternative denominator expression may be derived in terms of \( \psi_{N_{11}} \) and \( \psi_{N_{12}} \) only.
• Poles of the near end response (4.13) are same as those of the far end output expression (4.12).

• Zeros of the near end expression are the poles of the output expression when only load is considered (4.8).

• In the absence of a driver, the poles of the output expression are identical to the poles of a transmission line with load only, discussed in section 4.2 and expressed in (4.7).

\[ V_{out} = -\frac{\text{zeros} \left( Y_{\text{off-diagonal}} \right)}{\text{poles} \left( TL_{\text{WithLoad}} \right)} V_s \]  

(4.14)

• As the impedance of the driver circuit increases, the poles move toward the roots of \( \left( (\Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}}) Y_L + 2\Psi_{N_{11}} \Psi_{N_{21}} \right) Z_D \)

### 4.4 Transient Response Computation

The rational expressions obtained for the output signal (4.8), (4.12) and (4.13) can be written in a pole residue form, with poles \( p_i \) and residues \( r_i \)

\[ V_{out}(s) = \left[ \sum_{i=1}^{N_p} \frac{r_i}{s - p_i} \right] V_s(s) \]  

(4.15)

The input signal \( V_s \) can be a step, \( V_s = 1/s \), or a ramp, \( V_s = 1/s^2 \) or a combination
of these functions that represent the input digital signal.

Inverse Laplace transformation can be applied to (4.15) to obtain the output expression in time-domain

\[ v_{\text{out}}(t) = c + \sum_{i=1}^{N_P} \tilde{r}_i e^{\text{Pt}} \]  

(4.16)

where \( c \) is a constant and \( \tilde{r}_i \) are the new residues, including the source waveform. The transient response can now be computed using the pole-residue expression of (4.16).

### 4.5 Analysis of Multiconductor Transmission Lines

![Multiconductor transmission line network](image)

Figure 4.6: Multiconductor transmission line network

In this section, the algorithm defined in sections 4.2 to 4.4 are extended for appli-
cation to multiconductor transmission line networks, as shown in Figure 4.6.

Y-parameter expression (3.22) is expanded for the matrix definitions of distributed interconnect parameters $R \in \mathbb{R}^{N \times N}$, $L \in \mathbb{R}^{N \times N}$, $G \in \mathbb{R}^{N \times N}$, $C \in \mathbb{R}^{N \times N}$.

$$Y = \frac{1}{2} \left[ \Psi_{N12} \right]^{-1} \Psi_{N11} \begin{bmatrix} -I & I \\ I & -I \end{bmatrix} - \frac{1}{2} \left[ \Psi_{N22} \right]^{-1} \Psi_{N21} \begin{bmatrix} I & I \\ I & I \end{bmatrix}$$

(4.17)

Where $\Psi_{N}$ matrix can be evaluated in a closed-form and a priori [13], based on parameter matrices and the approximation of (3.18).

The output expressions can now be written. For a transmission line network when only a loading circuits are present the output can be defined by

$$V_{out} = \left[ \Psi_{21} \Psi_{12} + \Psi_{22} Y_{L} \Psi_{12} + \Psi_{22} \Psi_{11} + \Psi_{22} \Psi_{12} Y_{L} \right]^{-1}$$

$$\times \left[ \Psi_{22} \Psi_{11} - \Psi_{21} \Psi_{12} \right] V_{in}$$

(4.18)

Once both driver and load circuits are considered, the output expression becomes

$$V_{out} = \left[ \left( \Psi_{11} + \Psi_{12} Y_{L} \right) \left( \Psi_{21} + \Psi_{22} Y_{D} \right) + \left( \Psi_{11} + \Psi_{12} Y_{D} \right) \left( \Psi_{21} + \Psi_{22} Y_{L} \right) \right]^{-1}$$

$$\times \left[ \left( \Psi_{11} + \Psi_{12} Y_{D} \right) \Psi_{22} Y_{D} - \Psi_{12} Y_{D} \left( \Psi_{21} + \Psi_{22} Y_{D} \right) \right] V_{S}$$

(4.19)

And the near end response is defined:

$$V_{in} = \left[ \left( \Psi_{21} + \Psi_{22} Y_{L} \right) \left( \Psi_{11} + \Psi_{12} Y_{D} \right) + \left( \Psi_{21} + \Psi_{22} Y_{D} \right) \left( \Psi_{11} + \Psi_{12} Y_{L} \right) \right]^{-1}$$

$$\times \left[ \Psi_{22} Y_{D} \left( \Psi_{11} + \Psi_{12} Y_{L} \right) + \left( \Psi_{21} + \Psi_{22} Y_{D} \right) \Psi_{12} Y_{D} \right] V_{S}$$

(4.20)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
From the obtained expressions, individual responses are still defined as a pole-residue rational functions, allowing for the transient response computation, similar to that of a single interconnect.

4.6 Numerical Examples

4.6.1 Example 1: Single Transmission Line

Consider the single distributed line network with driver resistance $R_D = 15\Omega$, load capacitance is $C_L = 1pF$. The interconnect is described by distributed (per unit length) parameters: $R = 3.448\Omega/cm$, $L = 4.976nH/cm$, $G = 0.01S/cm$ and $C = 1.082pF/cm$. The interconnect length is 6cm.

The interconnect is macromodelled using the MRA algorithm with order $N = 3$, as described in section 3.3. The poles of the output response when only load is added to the interconnect is analyzed in Figure 4.7. In order to obtain the desired transient response residues are computed for the corresponding poles. The result is compared with HSPICE simulation in Figure 4.8. As seen, they match reasonably well.

Similarly, poles of the output expression are considered for varying driver values
and are shown in Figure 4.9. From the graphs in Figures 4.10 and 4.11, it is seen that both the proposed method and HSPICE yield comparable results. In order to compare the performance of the algorithm, the computational cost of the above numerical example using the proposed method is compared with HSPICE simulation, in Table 4.1. As seen from the table, the proposed method yields significant speed-up compared to traditional circuit simulations methods.

Table 4.1: Example 1: Comparison of computational cost (seconds)

<table>
<thead>
<tr>
<th></th>
<th>HSPICE</th>
<th>Proposed</th>
<th>Speed-Up(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.78</td>
<td>0.16</td>
<td>393.75</td>
</tr>
</tbody>
</table>

4.6.2 Example 2: Single Transmission Line with Complex Driver and Load

The interconnect considered for this example is the one described in Example 1, with distributed (per unit length) parameters: \( R = 3.448 \Omega/cm, L = 4.976nH/cm, \)
\( G = 0.01S/cm \) and \( C = 1.082pF/cm \) and length is 6cm. In this example, complex load admittance and driver impedance are considered. The driver is defined by a resistance, \( R_D = 15\Omega \), and inductance, \( L_D = 1.3nH \). The load is represented by a capacitance, \( C_L = 1pF \) and conductance, \( G_L = 0.02S \), as given in Figure 4.12.
Figure 4.7: Example 1: Pole movement in the expression for output transfer function with varying load capacitance
The interconnect is macromodelled using the MRA algorithm with order $N = 3$. The output response, using both the proposed method and HSPICE, when only the load is considered is shown in Figure 4.13. Transient simulation results with both driver and load are shown in Figure 4.14 (for the near end response), and in Figure 4.15 (for the far end response). It can be seen that both responses match accurately. The efficiency of the result is measured by comparing the computation time using the proposed algorithm with the simulation time required by HSPICE simulation. This is shown in Table 4.2. As seen from the table, significant speed-up is achieved by the proposed algorithm compared to traditional transient HSPICE type simulation.
Figure 4.9: Example 1: Pole movement in the expression for output transfer function with varying driver resistance
Figure 4.10: Example 1: Transient response at the near end

Figure 4.11: Example 1: Transient response at the far end
Figure 4.12: Circuit of Example 2

Figure 4.13: Example 2: Output response of interconnect with load = 1pF

Table 4.2: Example 2: Comparison of computational cost (seconds)

<table>
<thead>
<tr>
<th></th>
<th>HSPICE</th>
<th>Proposed</th>
<th>Speed-Up (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.19</td>
<td>0.09</td>
<td>111</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 4.14: Example 2: Transient response at the near end

Figure 4.15: Example 2: Transient response at the far end
4.6.3 Example 3: Multiconductor Transmission Line Network

Consider the coupled distributed line network described in Figure 4.16. The loading is represented by a capacitance, \( C_L = 1.5\, \text{pF} \), and the driver by a resistance, \( R_D = 30\, \Omega \) and the length of the transmission line is 10cm. The multiconductor transmission line is described by per unit length parameters, where

\[
\begin{bmatrix}
9.3519 & 0.092593 \\
0.092593 & 9.3519
\end{bmatrix}
\frac{\Omega}{\text{m}}, \quad
\begin{bmatrix}
846.12 & 306.83 \\
306.83 & 846.12
\end{bmatrix}
\frac{nH}{\text{m}}, \quad \text{and} \quad
\begin{bmatrix}
60.386 & -7.9044 \\
-7.9044 & 60.386
\end{bmatrix}
\frac{\mu F}{\text{m}}.
\]

In the analysis, a MRA approximation of order 6 is considered. The transient responses are shown for the signal line near end in Figure 4.17 and for the victim line near end in 4.18. The transient response for the signal line far end in Figure 4.19 and for the victim line far end response in 4.20. As seen from the plotted responses,
Table 4.3: Example 3: Comparison of computational cost (seconds)

<table>
<thead>
<tr>
<th>HSPICE</th>
<th>Proposed</th>
<th>Speed-Up (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>0.25</td>
<td>304.00</td>
</tr>
</tbody>
</table>

Simulation results are identical to those generated by HSPICE. Computational cost comparison for the above numerical example, using both the proposed method and HSPICE simulation is given in Table 4.3. As seen from the table, the proposed method yields significant speed-up compared to HSPICE.

Figure 4.17: Example 3: Transient response at near end of signal line, \( V_2 \)
Figure 4.18: Example 3: Transient response at near end of victim line, \((V_4)\)

Figure 4.19: Example 3: Transient response at far end of signal line, \((V_3)\)
Figure 4.20: Example 3: Transient response at far end of victim line, \( (V_4) \)
Chapter 5

Performance Improvements and the Special Cases of the Proposed Algorithm

In Chapter 4, a new algorithm was developed for fast transient analysis of interconnect networks, including driver and load circuits. The proposed method consists of defining load and driver circuits by their inductance/admittance and combining the expressions with a distributed transmission line macromodel of the interconnect. The desired output signal is written in pole-residue form, in frequency-domain. Transient responses are obtained by writing the pole-residue expressions in time-domain via inverse Laplace transform.
In the proposed algorithm, the step of finding the poles of the output expressions, relies on polynomial root finding algorithms that may limit the speed and accuracy of the proposed method. To address this issue, special cases are studied in this chapter, leading to further performance improvements while deriving the closed-form expressions. First, the macromodels with low-orders are investigated. Next, special cases of RC distributed transmission line (while neglecting the inductance) and lossless distributed transmission lines ($R = 0$, $G = 0$) are investigated.

5.1 Fast Pole Computation for Low-Order Macromodels

General polynomial root finding consists of iteratively searching for poles until desired accuracy is achieved. For the purpose of interconnect macromodelling proposed in Chapter 4, the iterative search may become inefficient and slow. If the macromodels are of lower-ordering (order 5 or less), the corresponding output expressions can be computed in a closed-form method [14], [15], [16], [17], [18]. Each of the orders of interest will be discussed in this section. By incorporating this simplification into the root finding algorithm, systems of low-order are solved in closed-form expressions without requiring any iterations.
5.1.1 Cubic Polynomials

A solution for the general cubic equations can be found using Cardan’s formula as follows [15], [16], [17].

The general cubic equation

\[ a_3x^3 + a_2x^2 + a_1x + a_0 = 0 \]  \hspace{1cm} (5.1)

is solved by performing a transformation into a reduced cubic equation by replacing \( x \) as

\[ x = z - \frac{a_2}{3a_3} \]  \hspace{1cm} (5.2)

such that (5.1) becomes

\[ z^3 + Hz + G = 0 \]  \hspace{1cm} (5.3)

where

\[ H = \frac{3a_3a_1 - a_2^2}{9a_3^2} \quad G = \frac{2a_2^3 - 9a_3a_2a_1 + 27a_3^2a_0}{27a_3^3} \]  \hspace{1cm} (5.4)

The cubic variable \( z \) is replaced by two variables, \( u \) and \( v \), increasing the degree of freedom of the solution

\[ z = u + v \]  \hspace{1cm} (5.5)

Substituting the expression for \( z \) from (5.5) into (5.3) we obtain

\[ u^3 + v^3 + 3(uv + H)(u + v) + G = 0 \]  \hspace{1cm} (5.6)
In order for the equation in (5.6) to be satisfied, the following conditions must hold

\[ u^3 + v^3 = -G \]  \hspace{1cm} (5.7)
\[ uv = -H \]  \hspace{1cm} (5.8)

Now, equation (5.7) can be written in terms of a single variable, \( u \), which can be found by solving the quadratic equation described in (5.10) as

\[ u^3 - \frac{H^3}{u^3} = -G; \quad u^6 + Gu^3 - H^3 = 0 \]  \hspace{1cm} (5.9)
\[ u^3 = \frac{-G + \sqrt{G^2 + 4H^3}}{2} \]  \hspace{1cm} (5.10)

Now, \( v \) can be determined by substituting the solution of \( u \) into (5.8) and therefore \( z \) can be computed as

\[ z = u + v, \quad \frac{-(u+v) \pm (u-v) \sqrt{3i}}{2} \]  \hspace{1cm} (5.11)

The roots of the initial cubic expression (5.1) are found by transforming \( z \) back to \( x \) via (5.5).

### 5.1.2 Quartic Polynomials

Immediately after Cardan produced the cubic solution of 5.1.1, Ferrari showed how the cubic algorithm can be combined with a quartic to generate the roots of a general quartic polynomial [17]. Ferrari’s solution of the quartic equation is presented [15], [16].
The quartic equation

\[ x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = 0 \]  

(5.12)

is first re-written as a perfect square

\[ x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 + (b_1 x + b_0) = \left( x^2 + \frac{a_3}{2} x + k \right)^2 \]  

(5.13)

The newly introduced variables \( b_1, b_0 \) and \( k \) are found by expanding (5.13) and equating the coefficients of like powers of \( x \)

\[
\begin{align*}
    b_1^2 + a_2 &= 2k + \frac{a_3^2}{4} \\
    2b_1b_0 + a_1 &= ka_3 \\
    b_0^2 + a_0 &= k^2
\end{align*}
\]  

(5.14)

\[
    k^3 - \frac{a_2}{2} k^2 + \frac{1}{4} (a_3a_1 - 4a_0) k + \frac{1}{8} (4a_2a_0 - a_3^2a_0 - a_1^2) = 0
\]  

(5.15)

\( k \) is obtained by solving the cubic in (5.15). It is then substituted into (5.13) to obtain remaining variables.

Now, from (5.13) a simplification is written

\[
    \left( x^2 + \frac{a_3}{2} x + k \right)^2 = (b_1 x + b_0)^2
\]  

(5.16)
Therefore, the resulting quartic polynomial roots are obtained by solving two quadratic expressions

\[ x^2 + \frac{a_3}{2}x + k = b_1x + b_0 \]  
\[ x^2 + \frac{a_3}{2}x + k = -b_1x - b_0 \]  

(5.17) (5.18)

5.1.3 Quintic Polynomials

A solution to the general quintic equation was discovered by Kiepert in 1878 by describing the relationship between the icosahedron (a polyhedron having 20 faces) and the solution of a quintic equation. This method requires extensive understanding of group theory, symmetry, Galois theory of equations, elliptic functions and associated theta functions [17]. An overview of the procedure required for the solution of a general quintic is given in this section.

First, the general quintic equation

\[ x^5 + Ax^4 + Bx^3 + Cx^2 + Dx + E = 0 \]  

(5.19)

is transformed into the principal quintic equations via Tschirnhausen transformation as

\[ z^5 + 5az^2 + 5bz + c = 0 \]  

(5.20)
where

$$z = x^2 - ux + v$$

(5.21)

and $a, b$ and $c$ are defined as functions of $u,v$ and the general quintic coefficients.

Next step consists of transforming the principal quintic expression (5.20) to the Brioschi Quintic via a second Tschirnhausen transformation of the form

$$y^5 - 10Zy^3 + 45Z^2 y - Z^2 = 0$$

(5.22)

where the new variable is defined by the expression

$$z_k = \frac{\lambda + \mu y_k}{(y_k^2/Z) - 3}$$

(5.23)

and $Z$ represents a composite icosahedral function. This step is performed by relating the above transformation to the partition of an object of icosahedral symmetry into five equal objects of tetrahedral symmetry.

The Brioschi Quintic (5.22) is now transformed to the Jacobi sextic

$$s^6 + \frac{10}{\Delta} s^3 - \frac{12g_2}{\Delta^2} s + \frac{5}{\Delta^2} = 0$$

(5.24)

One of the roots of the sextic in (5.24) is designated as $s_\infty$ and the five other roots, $s_k$ where $0 \leq k \leq 4$, have the form

$$s_k = \left(\frac{s_\infty}{5}\right) \left(1 + r e^k - \frac{1}{r e^k}\right)^2$$

(5.25)
where \( \varepsilon \) represents any fifth root of unity.

The Jacobi sextic is defined by solving the Weierstrass elliptic functions using genus 1 theta functions. The evaluation of periods of the theta functions requires a solution of a cubic equation with coefficients simply calculable from \( g_2 \) and \( \Delta \) in (5.24). Substituting obtained periods into theta series gives the roots \( s_\infty \) and \( s_k \) (\( 0 \leq k \leq 4 \)) of (5.24). The roots of Brioschi quintic (5.22) are then obtained

\[
y_k = \sqrt[5]{\frac{1}{5} (s_\infty - s_k) (s_{k+2} - s_{k+3}) (s_{k+4} - s_{k+1})}
\]  

(5.26)

Now, undoing the Tschirnhausen transformations, the roots of the general quintic (5.19) can be computed.

### 5.2 Braodband and Long Delay Interconnect Macromodeling Issues

It is to be noted that depending on the interconnect behaviour under consideration, corresponding macromodels may require a large number of poles due to a high-order of approximation. This may occur for interconnect networks with signals consisting of high-frequency components due to sharp excitations in the circuit. Another cause of high-order approximation is due to long delays. Signals experiencing long delays consist of flat or nearly-flat response sections. In order to represent these regions, large
number of poles is required. Consequently, high-order approximations can present challenges for computation of poles to be evaluated as the roots of high-order polynomials while using the proposed algorithm. This can result in slow and CPU costly simulation. In order to address this difficulty, further performance improvements are proposed in the following sections.

5.3 Distributed RC Interconnects

The special case of distributed RC interconnects (neglecting the inductance such that \( L = 0 \)) is considered in this section. Properties of RC networks will be analyzed and used to optimize the root finding algorithm presented in this thesis.

First, the transmission line model is considered and reformulated for RC distributed interconnects, in section 5.3.1. In order to further study this special case, RC network properties are discussed in 5.3.2. Development of the output expression with the load included is given in section 5.3.3, and with both driver and load is given in section 5.3.4. Detailed analysis of a numerical example is provided using the proposed methods is described in section 5.3.5.
5.3.1 RC Distributed Transmission Line Macromodel

The matrix rational approximation based transmission line model, described in the previous chapter (in section 4.1), is used to represent the interconnect behaviour. The MRA macromodel may be modified to take into consideration of the special case, when \( L = 0 \). Consequently, we have \( a = Rd \) and \( b = (G + sC)d \). Substituting these, (3.20) can be modified as

\[
\Psi_N(K) = \left[ \begin{array}{c}
\sum_{i=0, (Even)}^{N} \psi_i(Rd)^{i/2} b^{i/2} \\
- \sum_{i=1, (Odd)}^{N} \psi_i(Rd)^{i/2} b^{i/2}
\end{array} \right] - \frac{N}{2} \psi_1(Rd)^{1/2} b^{1/2} (5.27)
\]

From the expression in (5.27) further simplifications can be noted:

\[
\Psi_{N11} = \Psi_{N22} (5.28)
\]

\[
\Psi_{N12} = -Rd \sum_{i=1, (Odd)}^{N} \psi_i(Rd)^{i/2} b^{i/2} (5.29)
\]

\[
\Psi_{N21} = -b \sum_{i=1, (Odd)}^{N} \psi_i(Rd)^{i/2} b^{i/2} (5.30)
\]

Therefore,

\[
\Psi_{N12} b = \Psi_{N21} Rd (5.31)
\]

or

\[
\Psi_{N21} = \Psi_{N12} \frac{b}{Rd} = \Psi_{N12} \frac{G + sC}{R} (5.32)
\]
Also to be noted from (5.32), is that roots of polynomial defined by $\Psi_{N_2}$, consist of the roots of polynomial defined by $\Psi_{N_2}$ and one additional pole: $pole_{12,extra} = \frac{-G}{C}$.

From the above observations, the Y-parameter matrix of the interconnect model given by (3.22) can be modified as

$$
\begin{bmatrix}
Y_{11} & Y_{12} \\
Y_{21} & Y_{22}
\end{bmatrix}
\approx
\frac{1}{2\Psi_{N_2}\Psi_{N_1}}
\begin{bmatrix}
\Psi_{N_1}^2 + \Psi_{N_2}^2 \frac{G+sC}{R} & -\Psi_{N_1}^2 + \Psi_{N_2}^2 \frac{G+sC}{R} \\
-\Psi_{N_1}^2 + \Psi_{N_2}^2 \frac{G+sC}{R} & \Psi_{N_1}^2 + \Psi_{N_2}^2 \frac{G+sC}{R}
\end{bmatrix}
\tag{5.33}
$$

5.3.2 RC Network Properties

As proven in Appendix A, neglecting the inductance in an interconnect, results in an RC circuit representation. Following observations can be found regarding the diagonal elements of the RC network admittance matrices [19], [20]:

- The poles and zeros are simple and lie on the negative real axis of the complex $s$ plane.
- The poles and the zeros alternate.
5.3.3 Output Evaluation with Capacitive Load

The previously obtained output expression of interconnect with a load (4.8) can be slightly modified to only account for the capacitive load condition.

\[ V_{\text{out}} = \frac{\Psi_{N_{22}} \Psi_{N_{11}} - \Psi_{N_{12}} \Psi_{N_{21}}}{\Psi_{N_{21}} \Psi_{N_{12}} + \Psi_{N_{11}} \Psi_{N_{22}} + 2\Psi_{N_{12}} \Psi_{N_{22}} (C_{L}s)} V_{\text{in}} \]  

(5.34)

The expression in (5.34) can be interpreted in terms of poles and zeros of the \( Y \)-parameter of the transmission line, from (5.33), and can be re-written as

\[ V_{\text{out}} = \frac{-\text{Zeros (\( Y_{\text{off-diagonal}} \))}}{\text{Zeros (\( Y_{\text{diagonal}} \)) + Poles (\( Y \)) \times C_{L}s}} V_{\text{in}} \]  

(5.35)

The obtained output expression can be analyzed using the RC network properties described in section 5.3.2, for efficient evaluation of the roots of the denominator expression. According to the properties described in section 5.3.2, poles and zeros of the RC interconnect \( Y \)-parameter matrix are real, negative and alternating. It is to be noted that an important property of the denominator of the output expression in (5.35) is that it consists of a sum of two polynomials: one having poles of the \( Y \)-parameters as its roots, and other having zeros of the \( Y \)-diagonal as its roots. This results in the denominator expression consisting of a sum of two polynomials whose roots are known, negative, real and alternating. This property can be exploited to
locate the possible pole boundaries, for the expression of $V_{out}$ in (5.35), thereby significantly aiding the convergence of the pole computing process [21]. To illustrate this property further, let $Poly_1$ and $Poly_2$ represent polynomials whose roots are real and alternating. The sum of the two polynomials will be referred to as $Poly_{sum}$. An example of such polynomials is shown in Figure 5.1. From the graph, it can be observed that, $Poly_{sum}$ contains roots that are bounded by the real, alternating poles of $Poly_1$ and $Poly_2$.

This above property is also demonstrated through the following numerical example. Consider two adding polynomials $Poly_1$ and $Poly_2$ with real and alternating roots in Table 5.1. The sum of the two polynomials results in a polynomial whose roots exist between the two sets of alternating roots. This result is also shown graphically in Figure 5.2.

<table>
<thead>
<tr>
<th>$Poly_1$</th>
<th>$Poly_2$</th>
<th>$Poly_{sum}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0.69722</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.3028</td>
</tr>
</tbody>
</table>

Since the poles of the output expression in (5.35), are obtained by adding two polyno-
Figure 5.1: Polynomial addition with real negative roots
Figure 5.2: Illustration of polynomials representing the poles given in Table 5.1

mials with alternating roots, it is concluded that these poles must exist between the zeros of the Y-diagonal and the poles of the Y-matrix of the corresponding interconnect MRA macromodel. Using this property, for each pole of the output expression (5.35) a location band can be identified, which are known a priori [13]. This allows for a root finding algorithm to limit the pole search to each known band, therefore significantly reducing the number of iterations that are required to find these poles, leading to faster convergence.
5.3.4 Output Evaluation with Capacitive Load and Resistive Driver

Previously derived output expressions (4.12) and (4.13) are modified to accommodate the new condition for both the far and near end response:

\[ V_{out} = \frac{\Psi_{N_{11}} \Psi_{N_{22}} - \Psi_{N_{12}} \Psi_{N_{21}}}{[(\Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}}) \ sC_L + 2\Psi_{N_{12}} \Psi_{N_{22}} \frac{G+sC}{R}]} \ R_D \]

\[ + \Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}} + 2\Psi_{N_{12}} \Psi_{N_{22}} sC_L \]

\[ V_{in} = \frac{\Psi_{N_{11}} \Psi_{N_{22}} + \Psi_{N_{12}} \Psi_{N_{21}} + 2\Psi_{N_{12}} \Psi_{N_{22}} sC_L}{[(\Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}}) \ sC_L + 2\Psi_{N_{12}} \Psi_{N_{22}} \frac{G+sC}{R}]} \ R_D \]

\[ + \Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}} + 2\Psi_{N_{12}} \Psi_{N_{22}} sC_L \]

In order to better understand where the poles of the above expressions are located, the denominator can be re-written in terms of known entities as

\[ \begin{bmatrix} \text{Zeros (Y Diagonal)} \ sC_L + \text{Poles (Y)} \frac{G+sC}{R} \end{bmatrix} \ R_D + \text{Zeros (Y Diagonal)} + \text{Poles (Y)} \ sC_L \]

(5.38)

This expression can be divided into two polynomials:

\[ \text{(Polybound}_1 \ R_D + \text{(Polybound}_2 \] (5.39)

where

\[ \text{Polybound}_1 = \text{Zeros (Y Diagonal)} \ sC_L + \text{Poles (Y)} \frac{G+sC}{R} \] (5.40)
Both boundary polynomials have roots that are bounded by zeros of the Y-diagonal and poles of the Y-matrix of the interconnect MRA macromodel. Computing the poles of \( \text{Polybound}_2 \) has been discussed in section 5.3.3. Roots of \( \text{Polybound}_1 \) can be determined following a similar procedure. In order to determine the poles of the output expression when both driver and load are present (5.36), properties of the two bounding polynomials are further examined below.

The roots of \( \text{Polybound}_2 \) are found to be between the zeros of Y-diagonal and poles of the Y-matrix, as described in section 5.3.3. This is shown graphically in Figure 5.3. Similarly, the roots of \( \text{Polybound}_1 \) can be obtained, as shown in Figure 5.4. In order to examine the relationship between the two polynomials, they are combined in Figure 5.5. From here, it is observed that the boundaries of the two expressions are the same.

From the polynomial expressions (5.40), it is found that \( \text{Polybound}_1 \) is right-bounded by zeros of TL Y-diagonal while left-bounded by the poles of TL Y-matrix. Similarly, from (5.41), it is noted that \( \text{Polybound}_2 \) is right-bounded by poles of TL Y-matrix and left-bounded by the zeros of TL Y-diagonal. This is shown graphically in Figure 5.5. As a result of this opposite bounding, the two polynomials have roots that are found in alternating bounding sections.
A small difference of a single pole in $\text{Polybound}_1$, at $-G/C$ is noted. From the properties of the MRA approximation it is noted that smallest zero of $Y$-diagonal will be larger than the observed pole at $-G/C$ [1], [10], [13]. Therefore resulting in preservation of alternating roots since

$$|\text{pole}_1 (Y - \text{matrix})| > |\text{zero}_1 (Y - \text{diagonal})| > \left| \frac{G}{C} \right|$$

Since their roots appear in alternating sections, it is concluded that the polynomials $\text{Polybound}_1$ and $\text{Polybound}_2$ have real, alternating roots. The last step in obtaining the poles of the output expressions of (5.36) is to combine $\text{Polybound}_1$ and $\text{Polybound}_2$ as described in (5.39). This problem is, once again, a simple polynomial addition with real alternating roots (the problem solved in section 5.3.3). Graphical representation of the solution is shown in Figure 5.6.

In summary, the new method locates a region for output expression poles, allowing for faster and guaranteed pole convergence. The simplified expressions are optimally utilized by minimizing the iterations during root-finding. The pole boundaries ensure that each pole converges quickly, by restricting the search to the found region.
Figure 5.3: $Poly_{bound_2}$ boundaries and roots

Figure 5.4: $Poly_{bound_1}$ boundaries and roots
Figure 5.5: Illustration of $\text{Poly}_{\text{bound}_1}$ and $\text{Poly}_{\text{bound}_2}$ with boundaries

Figure 5.6: Denominator polynomial of (5.36) as a sum of bounding polynomials $\text{Poly}_{\text{bound}_1}$ and $\text{Poly}_{\text{bound}_2}$
5.3.5 Numerical Example

In this experiment a single transmission line is considered with driver resistance $R_d = 50\Omega$ and a load capacitance of $C_L = 2pF$. The interconnect is 4cm long and is described by per unit length parameters: $R = 550\Omega/m$, $L = 0nH/m$, $G = 0.24S/m$ and $C = 420pF/m$. The interconnect is described by a 6\textsuperscript{th} order MRA macromodel.

Roots of $Poly_{\text{bound}_1}$ in (5.40) are obtained by a bounded pole search, where bounding pole locations are set between the zeros of Y-diagonal and poles of the Y-matrix. These are shown in Table 5.2 and Figure 5.7. The roots of $Poly_{\text{bound}_2}$ in (5.41) are found by performing another bounded pole search with zeros of Y-diagonal and poles of the Y-matrix as boundaries. Resulting poles are shown in Table 5.3 and Figure 5.8. The poles of the output expressions in (5.36) and (5.37) are now found by setting the above computed boundaries. Resulting pole boundaries and locations are shown in Table 5.4 and Figure 5.9. A single iteration was required for each root finding step, resulting in a total of 3 iterations performed while finding all poles of the output response.

In order to obtain the desired transient response corresponding residues are computed (shown in Table 5.5) and the resulting simulations are compared with the HSPICE simulation in Figure 5.10 and Figure 5.11. A CPU comparison of the transient anlysis
using the proposed method and HSPICE is given in Table 5.6. As seen, the proposed
method yields significant speed-up compared to the HSPICE.

Figure 5.7: Roots of \( \text{Poly}_{\text{bound}_1} \), in (5.40) (Iteration required = 1)

Table 5.2: Comparative locations of boundaries and roots of \( \text{Poly}_{\text{bound}_1}(10^9) \)

<table>
<thead>
<tr>
<th>( \text{poles}(Y_{TL}) )</th>
<th>( \text{zeros}(Y_{TL}) )</th>
<th>( \text{Poly}_{\text{bound}_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1999.8</td>
<td>-7802.8</td>
<td>-2389.5</td>
</tr>
<tr>
<td>-543.05</td>
<td>-923.81</td>
<td>-632.69</td>
</tr>
<tr>
<td>-247.39</td>
<td>-358.72</td>
<td>-290.21</td>
</tr>
<tr>
<td>-107.45</td>
<td>-168.35</td>
<td>-139.05</td>
</tr>
<tr>
<td>-27.275</td>
<td>-60.656</td>
<td>-49.217</td>
</tr>
<tr>
<td>-0</td>
<td>-7.2473</td>
<td>-5.7939</td>
</tr>
</tbody>
</table>
Figure 5.8: Roots of $\text{Poly}_{\text{bound}_2}$, in (5.41) (Iteration required = 1)

Table 5.3: Comparative locations of boundaries and roots of $\text{Poly}_{\text{bound}_2}(10^9)$

<table>
<thead>
<tr>
<th>zeros($Y_{TL}$)</th>
<th>poles($Y_{TL}$)</th>
<th>$\text{Poly}_{\text{bound}_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7802.8</td>
<td>$\infty$</td>
<td>-9410.3</td>
</tr>
<tr>
<td>-923.81</td>
<td>-1999.8</td>
<td>-1089.3</td>
</tr>
<tr>
<td>-358.72</td>
<td>-543.05</td>
<td>-416.18</td>
</tr>
<tr>
<td>-168.35</td>
<td>-247.39</td>
<td>-204.2</td>
</tr>
<tr>
<td>-60.656</td>
<td>-107.45</td>
<td>-88.037</td>
</tr>
<tr>
<td>-7.2473</td>
<td>-27.275</td>
<td>-21.946</td>
</tr>
<tr>
<td>0</td>
<td>-0.57143</td>
<td>-0.51023</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 5.9: Roots of the denominator of the output expression, in (5.39) (Iteration required = 1)

Table 5.4: Comparative locations of roots of boundaries and poles of the output expression ($10^9$)

<table>
<thead>
<tr>
<th>$Poly_{bound_1}$</th>
<th>$Poly_{bound_2}$</th>
<th>Output Poles</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$</td>
<td>-9410.3</td>
<td>-9499.5</td>
</tr>
<tr>
<td>-2389.5</td>
<td>-1089.5</td>
<td>-1099.2</td>
</tr>
<tr>
<td>-632.69</td>
<td>-416.18</td>
<td>-419.76</td>
</tr>
<tr>
<td>-290.21</td>
<td>-204.2</td>
<td>-206.58</td>
</tr>
<tr>
<td>-139.05</td>
<td>-88.037</td>
<td>-90.225</td>
</tr>
<tr>
<td>-49.217</td>
<td>-21.946</td>
<td>-24.039</td>
</tr>
<tr>
<td>-5.7939</td>
<td>-0.51023</td>
<td>-1.4118</td>
</tr>
</tbody>
</table>
Table 5.5: Poles and residues of output expressions of (5.36) and (5.37) ($10^9$)

<table>
<thead>
<tr>
<th>Poles</th>
<th>Residues ($V_{out}$)</th>
<th>Residues ($V_{in}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-9499.5</td>
<td>-0.0013203</td>
<td>-0.0094064</td>
</tr>
<tr>
<td>-1099.2</td>
<td>0.0034058</td>
<td>-0.0088535</td>
</tr>
<tr>
<td>-419.76</td>
<td>-0.0047786</td>
<td>-0.0085452</td>
</tr>
<tr>
<td>-206.58</td>
<td>0.0079531</td>
<td>-0.01147</td>
</tr>
<tr>
<td>-90.225</td>
<td>-0.019836</td>
<td>-0.024025</td>
</tr>
<tr>
<td>-24.039</td>
<td>0.079994</td>
<td>-0.084063</td>
</tr>
<tr>
<td>-1.4118</td>
<td>-0.68865</td>
<td>-0.54385</td>
</tr>
<tr>
<td>0</td>
<td>0.62323</td>
<td>0.69021</td>
</tr>
</tbody>
</table>

Figure 5.10: Comparison of transient response at far end ($V_{out}$)
Figure 5.11: Comparison of transient response at near end ($V_{in}$)

Table 5.6: Comparison of computational cost (seconds)

<table>
<thead>
<tr>
<th></th>
<th>HSPICE</th>
<th>Proposed</th>
<th>Speed-Up (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.09</td>
<td>0.016</td>
<td>462.5</td>
</tr>
</tbody>
</table>
5.4 Distributed Lossless (LC) Interconnect

In this section, another case of interest, lossless interconnects \((R = 0, G = 0)\), is considered. The properties of such networks are examined and utilized to speed-up the pole convergence.

The transmission line macromodel is considered in section 5.4.1. Expression for output signal with the load included is developed in section 5.4.3 and with both driver and load is section 5.4.4. Numerical example is presented to demonstrate the validity of the proposed algorithm in section 5.4.5.

5.4.1 LC Distributed Transmission Line Macromodel

In this section, the matrix rational approximation based transmission line model, described in section 4.1, is used to describe interconnect behaviour. The expression obtained in (3.20), is modified to take into consideration the new condition: \(G = 0, R = 0\) and \(a = sLd\) and \(b = sCd\):

\[
\Psi_N(K) = \left[ \sum_{i=0(Even)}^{N} \psi_i \left( s^2d^2LC \right)^{i/2} - \sum_{i=1(Odd)}^{N} \psi_i \left( s^2d^2LC \right)^{i-1/2} sLd \right]

- \sum_{i=1(Odd)}^{N} \psi_i \left( s^2d^2CL \right)^{i-1/2} sCd

+ \sum_{i=0(Even)}^{N} \psi_i \left( s^2d^2CL \right)^{i/2} \right] \quad (5.42)
\]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
From the expression in (5.42) further simplifications are noted as follows

\[ \Psi_{N_{11}} = \Psi_{N_{22}} \]  \hspace{1cm} (5.43)

\[ \Psi_{N_{12}} = - \sum_{i=1(\text{Odd})}^{N} \psi_{i} \left( s^{2}d^{2}LC \right)^{\frac{i-1}{2}} sLd \]  \hspace{1cm} (5.44)

\[ \Psi_{N_{21}} = - \sum_{i=1(\text{Odd})}^{N} \psi_{i} \left( s^{2}d^{2}CL \right)^{\frac{i-1}{2}} sCd \]  \hspace{1cm} (5.45)

Therefore,

\[ \Psi_{N_{12}} sCd = \Psi_{N_{21}} sLd \]  \hspace{1cm} (5.46)

or

\[ \Psi_{N_{21}} = \Psi_{N_{12}} \frac{C}{L} \]  \hspace{1cm} (5.47)

Therefore, from (5.47), roots of polynomial defined by \( \Psi_{N_{21}} \) are same as those defined by \( \Psi_{N_{12}} \). Y-parameter matrix of the interconnect model is given by (3.22). The modifications outlined in (5.43) and (5.47) result in:

\[
\begin{bmatrix}
Y_{11} & Y_{12} \\
Y_{21} & Y_{22}
\end{bmatrix}
= \frac{1}{2\Psi_{N_{12}} \Psi_{N_{11}}}
\begin{bmatrix}
\Psi_{N_{11}}^{2} + \Psi_{N_{12}}^{2} \frac{C}{L} & -\Psi_{N_{11}}^{2} + \Psi_{N_{12}}^{2} \frac{C}{L} \\
-\Psi_{N_{11}}^{2} + \Psi_{N_{12}}^{2} \frac{C}{L} & \Psi_{N_{11}}^{2} + \Psi_{N_{12}}^{2} \frac{C}{L}
\end{bmatrix}
\]  \hspace{1cm} (5.48)
5.4.2 LC Network Properties

MRA macromodel of a lossless transmission line results in an LC network. Following observation can be found regarding LC network admittance matrices [19], [20]:

- Poles and zeros lie on the imaginary axis and are simple.
- The poles and the zeros alternate.
- $Y_{LC}$ is expressed as a ratio of an even and an odd polynomial.
- All residues of $Y_{LC}$ are positive.

5.4.3 Output Evaluation with Capacitive Load

The previously obtained output expression of interconnect with a load (4.8) is slightly modified to only account for the capacitive load condition.

$$V_{out} = \frac{\Psi_{N_2} \Psi_{N_1} - \Psi_{N_{12}} \Psi_{N_{21}}}{\Psi_{N_{21}} \Psi_{N_{12}} + \Psi_{N_{11}} \Psi_{N_{22}} + 2 \Psi_{N_{12}} \Psi_{N_{22}} (C_Ls)} V_{in}$$ \hspace{1cm} (5.49)

The obtained expression is the same as in (5.35), resulting, once again in the output expression in terms of $Y$-parameter poles and zeros

$$V_{out} = \frac{-\text{Zeros} (Y_{\text{off-diagonal}})}{\text{Zeros} (Y_{\text{diagonal}}) + \text{Poles} (Y) \times C_Ls} V_{in}$$ \hspace{1cm} (5.50)
The output expression defined in (5.50) can be analyzed using the LC network properties described in section 5.4.2, for efficient evaluation of the roots of the denominator expression. According to the properties of Y-parameters of LC networks (section 5.4.2), poles and zeros of the LC interconnect are purely imaginary and alternating. In this case the denominator consists of a sum of two polynomials: one having poles of the Y-parameter matrix as its roots, and the other having the zeros of the Y-diagonal as its roots. This property can be exploited to locate pole boundaries for the output expression of (5.50), and aiding the convergence of pole computation.

First, the polynomials obtained from LC interconnect MRA macromodeling will be examined. Since it is known that for LC network Y-parameter matrix has purely imaginary poles and residues (existing in complex conjugate pairs), the polynomials describing the boundaries of the output poles will be even functions. These contain only even exponents as

\[ a_0 + a_2x^2 + a_4x^4 + \cdots + a_{2n}x^{2n} = 0 \quad (5.51) \]

For the discussion to follow, let \( Poly_1 \) and \( Poly_2 \) represent the polynomials whose roots are purely imaginary and alternating. The sum of the two polynomials will be referred to as \( Poly_{sum} \). Since each of polynomial has purely imaginary roots, they
will both be even. A sum of even functions will be an even function (since only even powers are available). Now, roots of an even function $Poly_{Sum}$ can be computed as follows. Since roots are purely imaginary, let

$$x = jy$$  \hspace{1cm} (5.52)

where $y$ denotes the magnitudes of the complex roots, $x$. Next, since

$$j^n = \begin{cases} 
-1 ; & \frac{n}{2} = \text{integer} \\
1 ; & \frac{n}{4} = \text{integer}
\end{cases}$$ \hspace{1cm} (5.53)

the general even function (5.51) can be re-written as a polynomial with real roots equivalent to the magnitude of the purely complex $x$

$$a_0 - a_2y^2 + a_4y^4 - a_6y^6 \cdots + a_{2n}y^{2n} = 0$$ \hspace{1cm} (5.54)

All polynomials being discussed, $Poly_1$, $Poly_2$ and $Poly_{Sum}$ can be re-written such that the modified polynomials have real roots. Now, a similar treatment described for RC interconnects in section 5.3.3 can be adopted here. $Poly_{Sum}(modified)$ is a sum of two polynomials, $Poly_1(modified)$ and $Poly_2(modified)$, that have known, real, alternating roots. As was shown, $Poly_{Sum}(modified)$ will have roots that are bounded by the roots of $Poly_1(modified)$ and $Poly_2(modified)$. Once the magnitude of the imaginary pole is found, the purely complex pole can be computed.

The above method is demonstrated through a numerical example. Two polynomials
with purely imaginary roots, $Poly_1$ and $Poly_2$, are considered, Table 5.7, and the corresponding coefficients are shown in Table 5.8. Polynomials are transformed, as described above, and real roots are considered Table 5.9 and Table 5.10. The real roots are then used to generate the real solution to a modified $Polysum$ which are then converted to purely imaginary values. A comparison of these values in Table 5.11 proves the validity of the above approach.

As a result, the purely imaginary roots of $Polysum$ can be found by determining its magnitude from the magnitude values of alternating roots of $Poly_1$ and $Poly_2$.

Table 5.7: Imaginary roots of $Poly_1$ and $Poly_2$

<table>
<thead>
<tr>
<th>$Poly_1$</th>
<th>$Poly_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>±1j</td>
<td>±2j</td>
</tr>
<tr>
<td>±3j</td>
<td>±4j</td>
</tr>
<tr>
<td>±5j</td>
<td>±6j</td>
</tr>
</tbody>
</table>

Table 5.8: Polynomial coefficients of $Poly_1$ and $Poly_2$ and $Polysum$

<table>
<thead>
<tr>
<th></th>
<th>$Poly_1$</th>
<th></th>
<th>$Poly_2$</th>
<th></th>
<th>$Polysum$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>784</td>
</tr>
<tr>
<td>$Poly_2$</td>
<td>1</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>259</td>
</tr>
<tr>
<td>$Polysum$</td>
<td>1</td>
<td>0</td>
<td>91</td>
<td>0</td>
<td>1043</td>
</tr>
</tbody>
</table>
Table 5.9: Coefficients of modified polynomials of Table 5.8

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Poly_1 (modified) )</td>
<td>1 0 -56 0 784 0 -2304</td>
</tr>
<tr>
<td>( Poly_2 (modified) )</td>
<td>1 0 -35 0 259 0 -225</td>
</tr>
<tr>
<td>( Poly_{sum} (modified) )</td>
<td>1 0 -91 0 1043 0 -2529</td>
</tr>
</tbody>
</table>

Table 5.10: Real roots of the modified polynomials of Table 5.9

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>( \pm 1 )</th>
<th>( \pm 2 )</th>
<th>( \pm 3 )</th>
<th>( \pm 4 )</th>
<th>( \pm 5 )</th>
<th>( \pm 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Poly_1 )</td>
<td>( \pm 1 )</td>
<td>( \pm 2 )</td>
<td>( \pm 1.82 )</td>
<td>( \pm 3.6256 )</td>
<td>( \pm 5.3892 )</td>
<td></td>
</tr>
<tr>
<td>( Poly_2 )</td>
<td>( \pm 2 )</td>
<td>( \pm 4 )</td>
<td>( \pm 1.82 )</td>
<td>( \pm 3.6256 )</td>
<td>( \pm 5.3892 )</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.11: Comparison of polynomial roots

<table>
<thead>
<tr>
<th>Polynomial ( (modified) )</th>
<th>Roots of Polynomial ( Poly_{sum} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Poly_{sum} ) ( (modified) ) ( \times ) ( j )</td>
<td>( \pm 1.82j )</td>
</tr>
<tr>
<td>( \pm 1.82j )</td>
<td>( \pm 1.82j )</td>
</tr>
<tr>
<td>( \pm 3.6256j )</td>
<td>( \pm 3.6256j )</td>
</tr>
<tr>
<td>( \pm 5.3892j )</td>
<td>( \pm 5.3892j )</td>
</tr>
</tbody>
</table>
5.4.4 Output Evaluation with Load and Driver

General output expressions are previously derived to be (4.12). Since the special case of an LC interconnect is considered, the pole boundaries exist purely on the imaginary axis. That is, when no driver is present and a very large (infinitely large) driver impedance is considered, the starting and ending poles of the output expression will be purely imaginary. These are shown as starting and ending pole locations in Figure 5.12 and 5.13.

If the nature of the circuit does not change with the addition of the driver, that is, if driver is an inductor, $L$, the output poles would remain on the imaginary axis. For this case, previously derived output expressions (4.12) and (4.13) are modified to accommodate the new condition, $Z_D = sL_D$, for both far and near end response:

$$V_{out} = \frac{\Psi_{N_{11}} \Psi_{N_{22}} - \Psi_{N_{12}} \Psi_{N_{21}}}{\left( (\Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}}) sC_L + 2\Psi_{N_{12}} \Psi_{N_{22}} \frac{C}{L} \right) sL_D} V_S$$  \hspace{1cm} (5.55)

$$V_{in} = \frac{\Psi_{N_{11}} \Psi_{N_{22}} + \Psi_{N_{12}} \Psi_{N_{21}} + 2\Psi_{N_{12}} \Psi_{N_{22}} sC_L}{\left( (\Psi_{N_{12}} \Psi_{N_{21}} + \Psi_{N_{11}} \Psi_{N_{22}}) sC_L + 2\Psi_{N_{12}} \Psi_{N_{22}} \frac{C}{L} \right) sL_D} V_S$$  \hspace{1cm} (5.56)

The pole locations can be determined following the methods described for the RC circuit, as outlined in (5.3.4), since the pole locations move exist only along the purely
imaginary axis (Figure 5.12).

If however, a resistive driver is considered, the LC nature of this circuit is changed. The resulting output expressions become

\[
V_{\text{out}} = \frac{\Psi_{N11} \Psi_{N22} - \Psi_{N12} \Psi_{N21}}{\left[ (\Psi_{N12} \Psi_{N21} + \Psi_{N11} \Psi_{N22}) sC_L + 2\Psi_{N12} \Psi_{N22} \frac{C}{L} \right] R_D + \Psi_{N12} \Psi_{N21} + \Psi_{N11} \Psi_{N22} + 2\Psi_{N12} \Psi_{N22} sC_L} V_S \quad (5.57)
\]

\[
V_{\text{in}} = \frac{\Psi_{N11} \Psi_{N22} + \Psi_{N12} \Psi_{N21} + 2\Psi_{N12} \Psi_{N22} sC_L}{\left[ (\Psi_{N12} \Psi_{N21} + \Psi_{N11} \Psi_{N22}) sC_L + 2\Psi_{N12} \Psi_{N22} \frac{C}{L} \right] R_D + \Psi_{N12} \Psi_{N21} + \Psi_{N11} \Psi_{N22} + 2\Psi_{N12} \Psi_{N22} sC_L} V_S \quad (5.58)
\]

The introduction of a resistance into an LC network results in shifting of the poles from the imaginary axis, as shown in Figure 5.13. Therefore, for this consideration, a lossless (LC) interconnect with a resistive driver, more general methods must be used to determine the pole locations.

### 5.4.5 Numerical Example

In this experiment, a single transmission line is considered with driver resistance \( R_D = 100 \Omega \) and a load capacitance of \( C_L = 0.5pF \). The interconnect is 20cm long and is described by per unit length parameters \( R = 0\Omega/m, \ G = 0S/m, \ L = 279.9nH/m, \)
Figure 5.12: Poles of (4.8) for varying driver inductance $L_D$
Figure 5.13: Poles of (4.8) for varying driver resistance $R_D$
and \( C = 263.7 \, pF/m \). The interconnect is described by a 4\textsuperscript{th} order MRA macromodel.

The output expression pole boundaries are obtained as described in section 5.3.5 by the equation (5.39). The roots of \( Poly_{\text{bound}_1} \) from (5.40) are shown in Table 5.12 and Figure 5.14. Roots of \( Poly_{\text{bound}_2} \) from (5.41) are shown in Table 5.13 and Figure 5.15. The poles of the output expressions are in Table 5.14 and Figure 5.16. In order to obtain the desired transient response, residues are computed (Table 5.15) and transient simulations are compared with HSPICE simulation in Figure 5.17 and Figure 5.18. A CPU comparison of the transient analysis using the proposed method and HSPICE is given in Table 5.16. As seen, the proposed method yields significant speed-up compared to the HSPICE.

Table 5.12: Comparative locations of boundaries and roots of \( Poly_{\text{bound}_1}(10^9) \)

<table>
<thead>
<tr>
<th>poles((Y_{TL}))</th>
<th>zeros((Y_{TL}))</th>
<th>( Poly_{\text{bound}_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pm 7.3581i )</td>
<td>( \pm 14.466i )</td>
<td>( \pm 12.774i )</td>
</tr>
<tr>
<td>( \pm 3.656i )</td>
<td>( \pm 4.9685i )</td>
<td>( \pm 4.884i )</td>
</tr>
<tr>
<td>( \pm 1.7728i )</td>
<td>( \pm 2.6715i )</td>
<td>( \pm 2.6456i )</td>
</tr>
<tr>
<td>0</td>
<td>( \pm 0.88614i )</td>
<td>( \pm 0.87781i )</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 5.14: Roots of $Poly_{\text{bound}_1}$.
Figure 5.15: Roots of $Poly_{\text{bound}_2}$
Table 5.13: Comparative locations of boundaries and roots of $Poly_{bound_2}(10^9)$

<table>
<thead>
<tr>
<th>zeros($Y_{TL}$)</th>
<th>poles($Y_{TL}$)</th>
<th>$Poly_{bound_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>±14.466$i$</td>
<td>±∞</td>
<td>±38.962$i$</td>
</tr>
<tr>
<td>±4.9685$i$</td>
<td>±7.3581$i$</td>
<td>±7.0999$i$</td>
</tr>
<tr>
<td>±2.6715$i$</td>
<td>±3.656$i$</td>
<td>±3.6143$i$</td>
</tr>
<tr>
<td>±0.88614$i$</td>
<td>±1.7728$i$</td>
<td>±1.7561$i$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.14: Output expression for pole boundaries and final pole locations ($10^9$)

<table>
<thead>
<tr>
<th>$Poly_{bound_1}$</th>
<th>$Poly_{bound_2}$</th>
<th>Output Poles</th>
</tr>
</thead>
<tbody>
<tr>
<td>±∞</td>
<td>±38.962$i$</td>
<td>-270.72</td>
</tr>
<tr>
<td>±12.774$i$</td>
<td>±7.0999$i$</td>
<td>-1.9081 ±12.505$i$</td>
</tr>
<tr>
<td>±4.884$i$</td>
<td>±3.6143$i$</td>
<td>-0.3055 ±4.868$i$</td>
</tr>
<tr>
<td>±2.6456$i$</td>
<td>±1.756$i$</td>
<td>-0.17741 ±2.6443</td>
</tr>
<tr>
<td>±0.87781$i$</td>
<td>0</td>
<td>-0.17145 ±0.87781$i$</td>
</tr>
</tbody>
</table>
Figure 5.16: Roots of the denominator of the output expression
Table 5.15: Poles and residues of output expressions ($10^9$)

<table>
<thead>
<tr>
<th>Poles</th>
<th>Residues ($V_{out}$)</th>
<th>Residues ($V_{in}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-270.72</td>
<td>-0.75085</td>
<td>-1.0395</td>
</tr>
<tr>
<td>-1.9081 ±12.505i</td>
<td>0.50249 ±0.062028i</td>
<td>-0.020776 ±0.15981i</td>
</tr>
<tr>
<td>-0.3055 ±4.868i</td>
<td>-0.21248 ±0.00886i</td>
<td>-0.002798 ±0.066436i</td>
</tr>
<tr>
<td>-0.17741 ±2.6443</td>
<td>0.22766 ±0.012029i</td>
<td>0.003743 ±0.071017i</td>
</tr>
<tr>
<td>-0.17145 ±0.87781i</td>
<td>-0.64224 ±0.12546i</td>
<td>0.039093 ±0.20017i</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5.17: Transient response at far end ($V_{out}$)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 5.18: Transient response at near end ($V_{in}$)

Table 5.16: Comparison of computational cost (seconds)

<table>
<thead>
<tr>
<th></th>
<th>HSPICE</th>
<th>Proposed</th>
<th>Speed-Up (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.11</td>
<td>0.03</td>
<td></td>
<td>266.67</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusion

6.1 Summary

This thesis describes a proposed solution for fast transient analysis of interconnect circuits including driver and load. In the new algorithm, frequency-domain output response is written as a transfer function, that is then changed to a pole-residue expression. With inverse Laplace transformation, a frequency-domain pole-residue formulation becomes a time-domain, pole-residue model. This type of transient response is accurate and quickly computed. New permutations of the proposed algorithm of Chapter 4 are further investigated in Chapter 5 to derive the special cases of low order transmission line macromodels, RC and LC interconnect models.
The key issue addressed in this thesis is the time-domain simulation of interconnect networks. Typically transient simulation, when performed in SPICE-like simulators, consists of formulating circuit equations at each time point. These are then solved as the response is marched in time. This simulation may become very slow and inefficient for large circuits and when sharp changes are experienced in the transient response. The proposed method avoids this bottleneck entirely by not requiring SPICE for simulation purposes. This further improves the algorithm since circuit realization of the model is not required or performed.

The proposed algorithm has also eliminated the problem of mixed time-frequency for the described circuits. The proposed algorithm provides expressions as functions of driver and load admittance/impedance and the interconnect MRA macromodel approximation. As such, frequency domain descriptions of these parameters are included into the obtained output response, removing any concern regarding model definition in frequency-domain while obtaining simulations in time-domain.

With the improvements in speed while retaining accuracy, this algorithm may be ideal for applications involving repetitive simulation of a circuit, while modifying one or few parameters. An example of such an application would be interconnect analysis via eye diagram extraction or circuit optimization purposes.
6.2 Suggestions for Future Research

Research and improvements of the proposed algorithm may further advance the modeling and time-domain simulation of interconnects. Some suggestions that may be explored are given:

1. **Further pole analysis**
   Pole locations may be studied for different load, driver and interconnects. Expanding the algorithm to include more conditions, would make the algorithm more practical and likely to be implemented.

2. **Incorporating with delay extraction methods (DEPACT)**
   Proposed algorithm is most efficient for low order systems. Therefore, combining it with a method that ensures low order of approximation may prove beneficial. Delay extraction models, such as DEPACT, may be combined with the algorithm described in this thesis to further improve interconnect time-domain analysis.

3. **Non-linearity**
   Further research the possibility of incorporating non-linearity into the time-domain simulation proposed. This would allow for a wider variety of driver and loading circuits to be included.

4. **Coupling**
The simplified algorithms presented in Chapter 5 are only applicable to single transmission lines. This may be expandable to include MTLs.
Appendix A

MRA approximation when $L = 0$

In this section, a proof will be developed to demonstrate that interconnect MRA approximation when no inductance is considered ($L = 0$) results in an RC circuit. This will be shown by examining sub-sections described by the MRA algorithm [13], [10] and demonstrating that each section can be realized as an RC circuit.

The following conditions on the realizability of RC admittances are given in [19]:

1. Zeros and poles must lie on the negative real axis

2. They must alternate
Proof 1: Zeros and poles must lie on the negative real axis

From MRA [10] it is given that Y-parameter subsections are defined as

\[ Y_{11i} = Y_{22i} = \frac{x_i^2 + y_i^2}{4x_i a} + \frac{b}{4x_i} + \frac{x_i}{a + \frac{x_i^2 + y_i^2}{b}} \]  

(A.1)

where \( x_i \) and \( y_i \) are the real and imaginary parts, respectively, of the roots of the Padé approximation polynomials (3.24) and 3.25.

Expanding the expressions to obtain a transfer function we have

\[ Y_{11i} = Y_{22i} = \frac{(x_i^2 + y_i^2) (ab + x_i^2 + y_i^2) + ba (ab + x_i^2 + y_i^2) + bx_i 4x_i a}{4x_i a (ab + x_i^2 + y_i^2)} \]  

(A.2)

In order to study the poles, denominator expression is analyzed. Since \( a = R \) and \( b = G + sC \) it can be written as \( s (4x_i R^2 C) + 4x_i R (GR + x_i^2 + y_i^2) \) and the pole of the sub-section is defined as

\[ p_{Y_{11i}} = -\frac{GR + x_i^2 + y_i^2}{RC} \]  

(A.3)

Therefore, the pole is shown to be real and negative.

Next, the zeros of the sub-section Y-parameter are examined by solving the numerator expression \( s^2 R^2 C^2 + s 2RC [(x_i^2 + y_i^2) + RG + 2x_i^2] + 2RG (x_i^2 + y_i^2) + (x_i^2 + y_i^2)^2 + R^2 C^2 + x_i^2 4RG \)

By solving the quadratic polynomial, the zeros are found to be

\[ z_{Y_{11i}} = \frac{-3x_i^2 - y_i^2 - RG \pm 2x_i \sqrt{2x_i^2 + y_i^2}}{RC} \]  

(A.4)
From the obtained expression it is shown that the zeros will be real. They will be negative if

\[ 3x_i^2 + y_i^2 + RG > 2x_i\sqrt{2x_i^2 + y_i^2} \]  \hspace{1cm} (A.5)

Expanding and simplifying we have

\[ x_i^4 + 2x_i^2y_i^2 + y_i^4 + 6RGx_i^2 + 2RGy_i^2 + R^2G^2 > 0 \]

Therefore, since the condition is met, the zeros are both real and negative. The first condition is satisfied.

**Proof 2: Zeros and poles must alternate**

In order to see whether the sub-section Y-parameter expression satisfies this property, the magnitudes of poles and zeros are compared. First, it will be shown that \( z_{Y111i} < p_{Y11i} \).

\[ \frac{-3x_i^2 - y_i^2 - RG - 2x_i\sqrt{2x_i^2 + y_i^2}}{RC} < \frac{-GR + x_i^2 + y_i^2}{RC} \]  \hspace{1cm} (A.6)

Simplifying,

\[ -2x_i^2 - 2x_i\sqrt{2x_i^2 + y_i^2} < 0 \]  \hspace{1cm} (A.7)

Therefore, \( z_{Y111i} < p_{Y11i} \). Now, the second zero is shown to be larger than the pole, \( p_{Y11i} < z_{Y11i2} \) by comparing them

\[ \frac{-GR + x_i^2 + y_i^2}{RC} < \frac{-3x_i^2 - y_i^2 - RG + 2x_i\sqrt{2x_i^2 + y_i^2}}{RC} \]  \hspace{1cm} (A.8)
Simplifying

\[ x_i^2 + y_i^2 > 0 \]  
(A.9)

Therefore proving that

\[ z_{Y11i} < p_{Y11i} < z_{Y11z} \]  
(A.10)

And the poles and zeros alternate, as required.

Since the conditions for RC circuit realization are met, it is proven that each MRA sub-section can be composed of only R and C elements. Combining these sections results in interconnect MRA approximation that consists of R and C elements only.
References


no. 6, pp. 504–509, June 1975.


