

**ON THE SIMULATION  
OF  
OVERHEAD TRANSMISSION LINES**

by

SHAWN F. SILVERMAN

A Thesis Submitted to  
the Faculty of Graduate Studies  
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering  
University of Manitoba  
Winnipeg, Manitoba, Canada

Advisor: Dr. A. Gole

(xviii + 123 + R3 + A24 =) 168 pages

© Shawn F. Silverman, August 2005

## Abstract

This thesis explores and implements techniques for frequency domain modelling and time domain simulation of overhead transmission lines. The popular Vector Fitting algorithm is employed to approximate the frequency domain model using rational functions, and the recursive convolution technique is applied to the rational approximation to generate a time domain form.

The frequency domain model is translated into the time domain using delay extraction, modal decomposition, passivity enforcement, and rational approximation. Several approaches to each of these procedures are investigated.

The thesis also discusses several choices for the integration method used within the recursive convolution procedure.

In order to make the transmission line modeller and simulator easy to use, a Java-based library and partial graphical interface were developed. Specifically, the goal was to develop a platform-independent program that can run either stand-alone or as an applet inside a web page.

## Acknowledgements

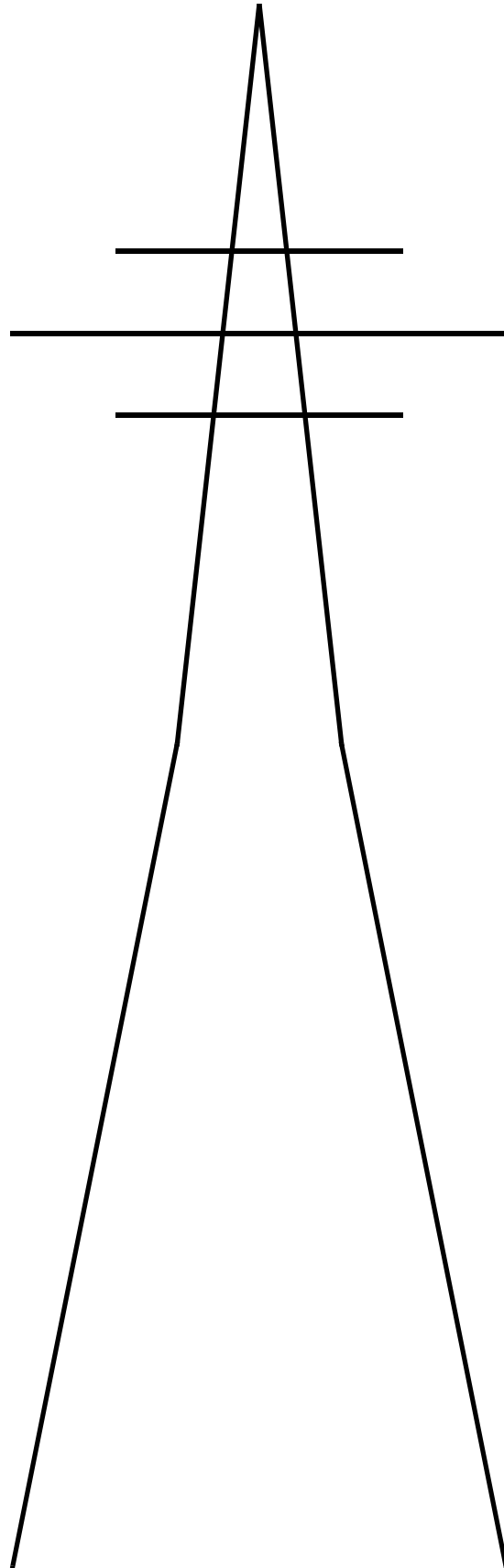
I would like to acknowledge all the people who have provided their support and encouragement throughout this entire process. I could not have finished this without you.

The following people and organizations deserve mention (in no particular order). I figured I'd be an Engineer and use a list instead of nicely crafted paragraphs.

Thank you:

- To all the people who came to my oral presentation, all 46 of you. Your support meant a lot to me.
- Garth Irwin of Electranix Corporation for agreeing to read my thesis and for his time spent with me discussing transmission line simulation problems.
- Michael Potter and Aram Faghfouri, my esteemed office mates, for generously letting me bounce mathematical ideas and problems off of them, and for every animated discussion that ensued. Also, Michael listened to and gave many excellent suggestions for my oral presentation.
- Michelle Saper for helping me refine my oral presentation. The advice was invaluable.
- Ian Jeffrey for agreeing to peruse my thesis and for his support.
- Dr. D. Kelly of the math department and Dr. G. Bridges of the Engineering department for being on my examining committee.
- Dr. W. Kinsner of the Engineering department for his undying support and encouragement, and for office space over the last year.
- Dr. Bob McLeod for his support and encouragement.
- Manitoba Hydro for funding.
- The HVDC Centre for office space.
- Isabelle Harouche for helping to teach me how to finish things.
- Dr. J. Lovetri for making me realize that I could only count on myself to finish this thing.
- Erwin Dirks and Grace McCaskill for providing excellent support services in the Power Tower.
- Dr. A. Gole for being my advisor and always providing his expert viewpoints on all this material.
- Dr. J. Williams, Dr. T. Berry, Dr. G. Krause, and Dr. D. Trim in the math department for our mathematical discussions.
- Dom Kwok for showing me how to do Wu style Tai Chi. It helped. :)
- My brother Chad and his wife Neely for my nephew Asher.
- My Mother and Father for showing support in their own ways.

I hope that I did not forget anyone, but if I did, you're certainly covered in the first paragraph!



# Table of Contents

	Page
Abstract.....	ii
Acknowledgements.....	iii
Table of Contents .....	v
List of Tables.....	xiii
List of Figures.....	xv
CHAPTER I: INTRODUCTION .....	1
CHAPTER II: A FREQUENCY DEPENDENT MODEL FOR OVERHEAD TRANSMISSION LINES.....	3
2.1 Unit-Length Matrices.....	3
2.1.1 Unit-Length Impedance, $Z$ .....	3
2.1.2 Unit-Length Admittance, $Y$ .....	5
2.1.3 Things of Which to Be Aware.....	5
2.1.4 Eliminating the Ground Wires .....	6
2.2 The Full Model .....	7
2.2.1 Characteristic Admittance Matrix.....	7
2.2.2 Propagation Matrix .....	7
2.3 Chapter Summary .....	8
CHAPTER III: FREQUENCY DOMAIN MODELS AND RATIONAL FUNCTIONS .9	
3.1 Convolving with Rational Functions .....	10
3.1.1 Direct Trapezoidal Integration.....	11
3.1.2 Assume Input is Linear .....	12

3.2	A Note on the Stability of Trapezoidal Integration.....	12
3.3	Terms with Conjugate Pairs.....	14
3.3.1	Terms with Conjugate Pairs, a State Variable Approach .....	15
3.4	Terms with Higher Order Poles, a Future Direction .....	17
3.4.1	An Alternative Formulation.....	19
3.5	Chapter Summary .....	19
CHAPTER IV: MODELLING A TRANSMISSION LINE USING RATIONAL FUNCTIONS.....		20
4.1	Characteristic Admittance Matrix.....	20
4.2	Propagation Matrix .....	22
4.2.1	Fitting the Propagation Matrix, H, in the Modal Domain .....	23
4.2.2	A Thought Experiment with Modal Time Delays .....	24
4.2.3	Finding the Modes of the Propagation Matrix, H.....	25
4.2.4	Finding the Delay for Each Mode of H .....	25
4.2.5	Refining the Calculated Delay .....	28
4.2.6	Fitting the Propagation Matrix, H, in the Phase Domain.....	29
4.2.7	Mode Grouping.....	30
4.3	Future Directions .....	31
4.4	Chapter Summary .....	31
CHAPTER V: VECTOR FITTING THEORY .....		33
5.1	A First Approach to Curve Fitting.....	33
5.1.1	Ill-Conditioning of this Formulation.....	34
5.2	Pole-Residue Form .....	35
5.2.1	The Pole-Residue Form is Nonlinear.....	35

5.3	The Theory of Vector Fitting .....	35
5.3.1	Initial Reasoning .....	36
5.4	Vector Fitting Methodology .....	37
5.4.1	Finding the Poles .....	37
5.4.2	Finding the Residues .....	39
5.4.3	Iteration .....	39
5.5	Summary So Far .....	39
5.7	Calculating the Zeros of $D(s)$ .....	41
5.8	Weighting .....	42
5.9	The “Vector” in Vector Fitting .....	42
5.10	Future Directions .....	43
5.10.1	Regularization .....	43
5.10.2	DC value .....	44
5.10.3	Poles at Zero .....	45
5.10.4	Convergence Properties .....	45
5.11	Poles of Order Greater Than One, a Future Direction .....	45
5.11.1	Detection .....	45
5.11.2	Implementing This Technique .....	47
5.11.3	Retrofitting Vector Fitting to Detect Higher-Order Poles .....	48
5.12	Chapter Summary .....	49
CHAPTER VI: VECTOR FITTING RESULTS .....		50
6.1	Fitting with Complex Poles .....	50
6.2	Fitting with Real Poles .....	51

6.3	Fitting with a reduced number of poles .....	53
6.4	Poles Outside the Considered Frequency Range .....	54
6.5	Fitting a Smooth Curve Using Real Poles .....	55
6.6	Fitting a Smooth Curve Using Complex Poles .....	57
6.7	Unstable Poles.....	58
6.8	Sensitivity to Eigenvalues.....	58
6.9	Chapter Summary .....	58
CHAPTER VII: ENFORCING PASSIVITY .....		60
7.1	Approach Based on Linearization and Quadratic Programming .....	60
7.2	Approach based on Passivity Compensation .....	61
7.3	Negative Eigenvalue Deletion .....	61
7.4	Enforcing the Positive Definiteness of E .....	62
7.5	Chapter Summary .....	62
CHAPTER VIII:AUTOMATIC POLE DETERMINATION .....		63
8.1	Distribute Poles.....	63
8.2	Intelligently Guess the Poles.....	64
8.2.1	Character of the Peaks .....	64
8.2.2	Smooth Curves with Complex Poles .....	65
8.3	Pole Choices .....	66
8.4	Future Directions .....	66
8.4.1	Distribution Scheme Based on the Propagation Matrix.....	66
8.5	Chapter Summary .....	66
CHAPTER IX: VECTOR FITTING RESULTS FOR THE TRANSMISSION LINE MODEL .....		67



9.1 Terminology .....	67
9.2 Two-Conductor Transmission Line .....	67
9.3 Three-Conductor Transmission Line .....	72
9.4 Chapter Summary .....	76
CHAPTER X: SIMULATION IN THE TIME DOMAIN.....	77
10.1 The “Black Box” .....	77
10.1.1 Simulation Steps .....	77
10.2 Transmission Line Update Equations .....	78
10.3 Convolution with the Characteristic Admittance, $Y_0$ .....	79
10.4 Convolution with the Propagation Function, $H$ .....	80
10.5 Eliminating High-Frequency Poles.....	81
10.6 Simulation in the Frequency Domain .....	82
10.7 Chapter Summary .....	83
CHAPTER XI: SIMULATION RESULTS .....	84
11.1 Two-Conductor Transmission Line .....	84
11.1.1 Applied 60Hz Voltage, Characteristic Termination .....	85
11.1.2 Applied 60Hz Voltage, Open Circuit Termination .....	89
11.1.3 Applied 60Hz Voltage, Short Circuit Termination .....	92
11.1.4 Applied Step Voltage, $5\mu\text{s}$ Time Step .....	95
11.1.5 Applied Step Voltage, $0.5\mu\text{s}$ Time Step .....	97
11.2 Three-Conductor Transmission Line .....	100
11.2.1 Applied 60Hz Balanced Voltage, Characteristic Termination .....	101
11.2.2 Applied 60Hz Balanced Voltage, Open Circuit Termination.....	104

11.2.3 Applied 150kHz Zero-Sequence Voltage, Characteristic Termination .....	107
11.2.4 Applied 150kHz Zero-Sequence Voltage, Open Circuit Termination	110
11.3 Chapter Summary .....	113
CHAPTER XII: NOTABLE IMPLEMENTATION DETAILS .....	114
12.1 Program Layout .....	114
12.2 Why Matlab Wasn't Used .....	114
12.3 Matrix Library.....	115
12.3.1 Overview of the Matrix Library.....	115
12.3.2 Basic Layout .....	116
12.4 Numerical Stability .....	117
12.5 The Eigenvalue-Finding Routine.....	118
12.5.1 Test Matrices for the Eigenvalue Routine.....	119
12.6 Chapter Summary .....	119
CHAPTER XIII: CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIONS.....	121
13.1 Vector Fitting .....	121
13.2 Passivity Enforcement .....	122
13.3 Time Domain Simulation.....	122
13.4 Direct Z-Domain Formulation.....	122
13.5 A Nice Looking Simulator.....	123
REFERENCES .....	124
APPENDIX A: TRANSMISSION LINE EQUATIONS .....	127
A.1 Direct Solution for Voltage .....	128

A.2	Direct Solution for Current .....	128
A.3	Two-Port Equations .....	129
A.4	The Z and Y Parameters .....	130
A.5	Useful Proofs .....	130
APPENDIX B:	THE LEAST SQUARES PROBLEM .....	132
APPENDIX C:	RESIDUE CALCULATION .....	133
C.1	Complex Residues .....	133
C.2	More on Complex Residues.....	134
APPENDIX D:	ZEROS OF A RATIONAL FUNCTION .....	136
D.1	Expansion.....	136
D.2	Eigenvalues .....	136
D.3	Proof.....	137
D.4	Calculation with Reals .....	139
D.5	Additional Notes on Eigenvalue Calculation.....	140
APPENDIX E:	RELATIONSHIP BETWEEN PHASE AND ATTENUATION FOR A MINIMUM PHASE SHIFT FUNCTION .....	141
APPENDIX F:	STATE EQUATION FORM OF AN NTH ORDER DIFFERENTIAL EQUATION.....	144
APPENDIX G:	COMPLEX IDENTITIES AND IMPLEMENTATION TECHNIQUES .. 146	
G.1	Why Is This Chapter Here? .....	146
G.2	Notation .....	146
G.3	Basic Identities.....	146
G.3.1	Rudimentary.....	146
G.3.2	Additive .....	147

G.3.3	Multiplicative.....	147
G.3.4	Trigonometric .....	147
G.3.5	Hyperbolic .....	148
G.4	Implementation Techniques.....	148
G.4.1	Hyperbolic Functions.....	148
G.4.2	Magnitude Functions .....	148
APPENDIX H:	DON'T FALL FOR THESE.....	150
H.1	Exponentials and Phase .....	150

## List of Tables

	Page
Table 6-1: Poles and Residues of the First Example Polynomial, in Hz.....	50
Table 6-2: Extra Poles and Residues, in Hz .....	50
Table 6-3: Extra Poles and Residues, in Hz .....	51
Table 6-4: Found Poles for the Reduced Order Fitting Case, in Hz .....	53
Table 6-5: Found Poles for the Extra Range Case, in Hz.....	55
Table 6-6: Poles and Residues of the Second Example Polynomial, in Hz .....	55
Table 6-7: Found Poles and Residues of the Second Example Polynomial, in Hz .....	57
Table 8-1: Ten Linearly Distributed Poles over 1 to 1000Hz .....	63
Table 9-1: Two-Conductor Transmission Line .....	67
Table 9-2: Fitting Parameters .....	68
Table 9-3: Three-Conductor Transmission Line .....	72
Table 9-4: Fitting Parameters .....	72
Table 11-1:Two-Conductor Transmission Line .....	84
Table 11-2:Fitting Parameters for Two-Conductor Line.....	85
Table 11-3:Simulation Parameters, 60Hz, Characteristic Termination.....	85
Table 11-4:Simulation Parameters, 60Hz, Open Circuit Termination .....	89
Table 11-5:Simulation Parameters, 60Hz, Short Circuit Termination .....	92
Table 11-6:Three-Conductor Transmission Line .....	100
Table 11-7:Fitting Parameters for Three-Conductor Line .....	100
Table 11-8:Simulation Parameters, 60Hz, Characteristic Termination.....	101
Table 11-9:Simulation Parameters, 60Hz, Open Circuit Termination .....	104

Table 11-10:Simulation Parameters, 150kHz, Characteristic Termination.....	108
Table 11-11:Simulation Parameters, 150kHz, Open Circuit Termination .....	110

## List of Figures

	Page
Fig. 3-1. Approximation of present input using previous time step.....	13
Fig. 3-2. No approximation of present input. ....	13
Fig. 3-3. Pseudocode for calculating each $a_i$ .....	18
Fig. 3-4. Pseudocode for calculating each $b_i$ .....	19
Fig. 4-1. Three-phase transmission line with a perfect delay. ....	24
Fig. 6-1. Starting with complex poles.....	51
Fig. 6-2. Starting with real poles. ....	52
Fig. 6-3. Starting with a reduced number of poles. ....	53
Fig. 6-4. There are poles outside the considered frequency range. ....	54
Fig. 6-5. Fitting a smooth curve with real poles.....	56
Fig. 6-6. Fitting a smooth curve with complex poles. ....	57
Fig. 7-1. Circuit with negative capacitance. ....	62
Fig. 9-1. Fitted (1,1) element of $Y_0$ .....	68
Fig. 9-2. Fitted (1,2) element of $Y_0$ .....	69
Fig. 9-3. Fitted (1,1) element of $H$ .....	69
Fig. 9-4. Fitted (1,2) element of $H$ .....	70
Fig. 9-5. The modes of $H$ .....	70
Fig. 9-6. Modal contributions for the fitted (1,1) element of $H$ . ....	71
Fig. 9-7. Modal contributions for the fitted (1,2) element of $H$ . ....	71
Fig. 9-8. Fitted (1,1) element of $Y_0$ .....	73
Fig. 9-9. Fitted (1,2) element of $Y_0$ .....	73

Fig. 9-10. Fitted (1,1) element of H.....	74
Fig. 9-11. Fitted (1,2) element of H.....	74
Fig. 9-12. The modes of H.....	75
Fig. 9-13. Modal contributions for the fitted (1,1) element of H. ....	75
Fig. 9-14. Modal contributions for the fitted (1,2) element of H. ....	76
Fig. 10-1. The transmission line as a black box. ....	77
Fig. 10-2. Conventions for current. ....	78
Fig. 10-3. Test configuration. ....	82
Fig. 11-1. Two conductors, 60Hz voltage, characteristic load, 1.667ms step. ....	86
Fig. 11-2. Two conductors, 60Hz voltage, characteristic load, 1.667ms step, exact.....	87
Fig. 11-3. Two conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step. ....	87
Fig. 11-4. Two conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step, exact. ....	88
Fig. 11-5. Two conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step, PSCAD. ....	88
Fig. 11-6. Two conductors, 60Hz voltage, open circuit load, 1.667ms step. ....	89
Fig. 11-7. Two conductors, 60Hz voltage, open circuit load, 1.667ms step, exact. ....	90
Fig. 11-8. Two conductors, 60Hz voltage, open circuit load, 50 $\mu$ s step. ....	90
Fig. 11-9. Two conductors, 60Hz voltage, open circuit load, 50 $\mu$ s step, exact. ....	91
Fig. 11-10. Two conductors, 60Hz voltage, open circuit load, 50 $\mu$ s step, PSCAD. ....	91
Fig. 11-11. Two conductors, 60Hz voltage, short circuit load, 1.667ms step. ....	92
Fig. 11-12. Two conductors, 60Hz voltage, short circuit load, 1.667ms step, exact. ....	93
Fig. 11-13. Two conductors, 60Hz voltage, short circuit load, 50 $\mu$ s step. ....	93
Fig. 11-14. Two conductors, 60Hz voltage, short circuit load, 50 $\mu$ s step, exact.....	94
Fig. 11-15. Two conductors, 60Hz voltage, short circuit load, 50 $\mu$ s step, PSCAD. ....	94



Fig. 11-16. Two conductors, step voltage, characteristic load, 5 $\mu$ s step. ....	95
Fig. 11-17. Two conductors, step voltage, open circuit load, 5 $\mu$ s step. ....	96
Fig. 11-18. Two conductors, step voltage, short circuit load, 5 $\mu$ s step. ....	96
Fig. 11-19. Two conductors, step voltage, characteristic load, 0.5 $\mu$ s step. ....	97
Fig. 11-20. Two conductors, step voltage, open circuit load, 0.5 $\mu$ s step. ....	98
Fig. 11-21. Two conductors, step voltage, short circuit load, 0.5 $\mu$ s step. ....	98
Fig. 11-22. Two conductors, step voltage, characteristic load, 0.5 $\mu$ s step, PSCAD. ....	99
Fig. 11-23. Two conductors, step voltage, open circuit load, 0.5 $\mu$ s step, PSCAD.....	99
Fig. 11-24. Two conductors, step voltage, short circuit load, 0.5 $\mu$ s step, PSCAD. ....	100
Fig. 11-25. Three conductors, 60Hz voltage, characteristic load, 1.667ms step.....	102
Fig. 11-26. Three conductors, 60Hz voltage, characteristic load, 1.667ms step, exact. ..	102
Fig. 11-27. Three conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step.....	103
Fig. 11-28. Three conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step, exact. ....	103
Fig. 11-29. Three conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step, PSCAD.....	104
Fig. 11-30. Three conductors, 60Hz voltage, open circuit load, 1.667ms step. ....	105
Fig. 11-31. Three conductors, 60Hz voltage, open circuit load, 1.667ms step, exact.....	105
Fig. 11-32. Three conductors, 60Hz voltage, open circuit load, 50 $\mu$ s step. ....	106
Fig. 11-33. Three conductors, 60Hz voltage, open circuit load, 50 $\mu$ s step, exact.....	106
Fig. 11-34. Three conductors, 60Hz voltage, open circuit load, 50 $\mu$ s, PSCAD.....	107
Fig. 11-35. Three conductors, 150kHz voltage, characteristic load, 0.667 $\mu$ s step.....	108
Fig. 11-36. Three conductors, 150kHz voltage, characteristic load, 0.667 $\mu$ s step, exact. ....	109
Fig. 11-37. Three conductors, 150kHz voltage, characteristic load, 0.667 $\mu$ s step, PSCAD..	110

Fig. 11-38. Three conductors, 150kHz voltage, open circuit load, 0.667 $\mu$ s step. ....	111
Fig. 11-39. Three conductors, 150kHz voltage, open circuit load, 0.667 $\mu$ s step, exact...	112
Fig. 11-40. Three conductors, 150kHz voltage, open circuit load, 0.667 $\mu$ s step, PSCAD. ...	112
Fig. 12-1. Program components. ....	114
Fig. 12-2. Difference between row-oriented and column-oriented. ....	116
Fig. 13-1. Screenshot of the graphical transmission line designer. ....	123
Fig. A-1. ‘Small’ transmission line segment. ....	127

# CHAPTER I

## INTRODUCTION

The goal of this thesis is to develop and simulate a model for overhead transmission lines, and is composed of three parts.

First, the popular technique of frequency domain fitting is studied for overhead transmission line models. These models are only expressed in the frequency domain, and the fitting process creates a new model that has a computationally efficient form in the time domain. An algorithm called Vector Fitting is used for this step.

Second, the fitted frequency domain function is expressed in the time domain using a technique called recursive convolution. This permits using the transmission line in both linear and nonlinear studies. Certain devices, such as diodes, switches, and transformer cores, are nonlinear and only have a time domain representation.

The third is to create a platform independent program to facilitate the study of overhead transmission lines. Using Java, it is possible to create an applet that runs on many platforms, including web browsers. This would make it possible for users worldwide to use such a program.

It is the author's belief that in order to competently expound on a topic, it is necessary to explore several more levels of difficulty. The original aim of this thesis was to explore the Vector Fitting algorithm as applied to overhead transmission lines. However, an accurately fitted curve, as we shall see, tells us nothing about the quality of the technique itself.

This is the reason that this thesis went further and developed a time domain model. It was hoped that utilizing the results of the Vector Fitting algorithm in a time domain implementation would expose any advantages or shortcomings of the method. Vector Fitting is, after all, just a means to an end.

### 1.1 Chapter Summaries

This section will describe a summary of each chapter.

Chapter 2 shows how to construct the frequency dependent model for an overhead transmission line. There are two main components of this model: the propagation matrix, and the characteristic admittance matrix.

Chapter 3 discusses how to represent, in the time domain, a frequency domain transfer function that is approximated with rational functions. A technique called recursive convolution is used to calculate an output given an input applied to the approximated model.

Chapter 4 shows how to express the transmission line model in terms of rational functions. The propagation component also contains a time delay, and the method for extracting this is also discussed.

Chapter 5 shows how to create a rational function approximation using a technique called Vector Fitting. This process first finds the poles, and then finds the residues.

Chapter 6 shows the results of applying the Vector Fitting algorithm to several sample polynomials. This verifies that the algorithm works.

Chapter 7 discusses why it is important to do further analysis on a fitted curve. Specifically, the results of Vector Fitting may not produce a passive system, even though the device being modelled is itself passive. It is shown how to make the system passive using one of several techniques.

Chapter 8 shows how to intelligently choose starting guess poles to bootstrap the Vector Fitting algorithm.

Chapter 9 contains some results of applying Vector Fitting to a few transmission line models.

Chapter 10 takes the results of the previous chapters and shows how to utilize them in a time domain simulation. Specifically, it is shown how to utilize the model of a transmission line in a generic network simulator. Also discussed is how to generate an exact steady state solution at a specific frequency to verify the time domain results.

Chapter 11 shows a number of simulation results for two transmission line geometries. Various terminations and applied voltages are used.

Chapter 12 discusses some notable implementation details.

Finally, Chapter 13 contains conclusions and recommendations.

There are also several appendices that provide supplementary information.

## CHAPTER II

# A FREQUENCY DEPENDENT MODEL FOR OVERHEAD TRANSMISSION LINES

Let's begin with a description of how to model an overhead transmission line. It will be seen that the model developed here is specified in the frequency domain. This chapter, therefore, is a first step towards our goal of a time domain representation.

### 2.1 Unit-Length Matrices

Before the full model of an overhead transmission line can be presented, special per-unit matrices must be defined. Please refer to Appendix A for a visual description.

The theory presented there is general and shows how to utilize the unit-length impedance and admittance matrices,  $\mathbf{Z}$  and  $\mathbf{Y}$ , of a transmission line to derive the voltage and current relationships. However, these matrices must be known, and are specific to the type of line.

This section shows how to construct the  $\mathbf{Z}$  and  $\mathbf{Y}$  matrices specifically for an overhead transmission line. Please refer to [1], [2], and [4] for more details about this formulation. Only a summary is presented here.

#### 2.1.1 Unit-Length Impedance, $\mathbf{Z}$

All the elements of this matrix are composed of an aerial and ground return impedance. The diagonal elements also include a term due to the internal impedance.

The aerial and ground return impedance is given by:

$$Z_{ij} = \frac{j\omega\mu}{2\pi} \left\{ \ln \frac{D_{ij}}{d_{ij}} + 2 \int_0^{\infty} \frac{e^{-\alpha \cos \theta_{ij}} \cos(\alpha \sin \theta_{ij})}{\alpha + \sqrt{\alpha^2 + jr_{ij}^2}} d\alpha \right\} \quad (2-1)$$

The first term is the aerial reactance of the conductor if the ground is a perfect conductor. The second term contributes additional impedance due to the finite ground conductivity, and is known as Carson's integral.

$D_{ij}$  is the distance between conductor  $i$  and the image of conductor  $j$ , and  $d_{ij}$  is the distance between conductors  $i$  and  $j$ . If  $i$  equals  $j$ , then  $d_{ii}$  is the radius of conductor  $i$ .

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i + y_j)^2}$$

$$d_{ij} = \begin{cases} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, & i \neq j \\ r_i \text{ (radius of conductor } i), & i = j \end{cases} \quad (2-2)$$

Although the integral is not used in the calculation as we shall see shortly, it contains the following terms which are defined as:

$$\theta_{ij} = \text{atan} \frac{(x_i - x_j)}{(y_i + y_j)} \quad (2-3)$$

$$r_{ij} = \sqrt{\omega \mu \sigma} D_{ij} = \frac{D_{ij}}{|d_e|}$$

It has been shown by Gary, Dubanton, Deri, and Semlyen that the integral can be approximated by

$$2 \int_0^{\infty} \frac{e^{-\alpha \cos \theta_{ij}} \cos(\alpha \sin \theta_{ij})}{\alpha + \sqrt{\alpha^2 + jr_{ij}^2}} d\alpha \approx \ln \frac{D'_{ij}}{D_{ij}}, \quad (2-4)$$

where  $D'_{ij}$  is similar to  $D_{ij}$  but considers each conductor to be raised off the ground by an additional complex depth,  $d_e$ .  $d_e$  is the classical complex depth of penetration and is defined as

$$d_e = \sqrt{\frac{\rho}{j\omega\mu}}, \quad (2-5)$$

where  $\rho$  is the ground resistivity.

Combining these concepts,

$$Z_{ij} = \frac{j\omega\mu}{2\pi} \left\{ \ln \frac{D_{ij}}{d_{ij}} + \ln \frac{D'_{ij}}{D_{ij}} \right\} = \frac{j\omega\mu}{2\pi} \ln \frac{D'_{ij}}{d_{ij}} \quad (2-6)$$

The diagonal elements also include a term due to the internal impedance. This is given by:

$$Z_{iic} = \frac{\rho}{2\pi r d_e} \coth \frac{0.777r}{d_e} + k \quad (2-7)$$

The  $d_e$  variable is defined the same way as above, except that it uses the conductor resistivity for  $\rho$  and not the ground resistivity. The constant,  $k$ , is derived by equating the expression at zero frequency to the DC resistance of the conductor:

$$\begin{aligned} \frac{\rho}{2\pi r d_e} + k &= \frac{\rho}{\pi r^2} \\ \therefore k &= \frac{\rho}{\pi r^2} \left(1 - \frac{1}{2(0.777)}\right) \end{aligned} \quad (2-8)$$

Recall that  $\coth z \rightarrow 1/z$  as  $z \rightarrow 0$ .

Finally, the diagonal elements equal  $Z_{ii} + Z_{iic}$ .

### 2.1.2 Unit-Length Admittance, $Y$

The following is defined:

$$[Y^{-1}]_{ij} = \frac{1}{j2\pi\epsilon_0\omega} \ln \frac{D_{ij}}{d_{ij}} \quad (2-9)$$

It is assumed that the air is virtually lossless and that the ground is at virtually zero potential. Assuming that the air is lossless means that the admittance will be purely imaginary. Some assume that there is a very small real part,  $G$ , because the air is not actually 100% lossless, however this paper does not include this value. In other words,  $Y = jB$  and not  $Y = G + jB$ . There is some discussion of shunt conductance in [5].

Note that the square root in the calculation of  $D_{ij}$  and  $d_{ij}$  can be eliminated by multiplying the logarithm by 0.5.

### 2.1.3 Things of Which to Be Aware

First, the accuracy of the approximation to the integral depends on the ratio of the distance between conductors to their height above ground [6]. This means that it is not a useful approximation for problems such as crosstalk between lines having a large distance between them.

Second, the approximation uses assumptions that are only valid to about 1 or 2 MHz. Specifically, it only works up to the point where the transmission mode changes to a surface wave.

### 2.1.4 Eliminating the Ground Wires

Normally, ground wires are grounded at every tower. Up to about 350kHz, it is possible to assume that the voltage on the ground wires is zero at all points. This allows a simplification of the  $\mathbf{Z}$  and  $\mathbf{Y}$  matrices, as follows.

For  $\mathbf{Z}$ , write

$$\frac{d}{dx} \begin{bmatrix} \mathbf{V}_c \\ \mathbf{V}_g \end{bmatrix} = - \begin{bmatrix} \mathbf{Z}_{cc} & \mathbf{Z}_{cg} \\ \mathbf{Z}_{gc} & \mathbf{Z}_{gg} \end{bmatrix} \begin{bmatrix} \mathbf{I}_c \\ \mathbf{I}_g \end{bmatrix}, \quad (2-10)$$

where the  $c$  and  $g$  subscripts represent the “conductor” and “ground” wire groups, respectively. If  $\mathbf{V}_g = 0$ , then

$$\frac{d\mathbf{V}_c}{dx} = -\mathbf{Z}'\mathbf{I}_c, \quad (2-11)$$

where

$$\mathbf{Z}' = \mathbf{Z}_{cc} - \mathbf{Z}_{cg}\mathbf{Z}_{gg}^{-1}\mathbf{Z}_{gc}. \quad (2-12)$$

$\mathbf{Z}'$  is a reduced order impedance matrix with the ground wires removed under the above assumption.

For  $\mathbf{Y}$ , write:

$$\frac{d}{dx} \begin{bmatrix} \mathbf{I}_c \\ \mathbf{I}_g \end{bmatrix} = - \begin{bmatrix} \mathbf{Y}_{cc} & \mathbf{Y}_{cg} \\ \mathbf{Y}_{gc} & \mathbf{Y}_{gg} \end{bmatrix} \begin{bmatrix} \mathbf{V}_c \\ \mathbf{V}_g \end{bmatrix} \quad (2-13)$$

Thus,

$$\frac{d\mathbf{I}_c}{dx} = -\mathbf{Y}'\mathbf{V}_c, \quad (2-14)$$

where

$$\mathbf{Y}' = \mathbf{Y}_{cc}. \quad (2-15)$$

$\mathbf{Y}'$  is a reduced order admittance matrix, also with the ground wires removed.



## 2.2 The Full Model

The special unit-length matrices are used to define the *propagation matrix* and the *characteristic admittance matrix*.

The propagation matrix,  $\mathbf{H}$ , describes how current and voltage waves cross the transmission line. This includes attenuation and time delays. The characteristic admittance matrix,  $\mathbf{Y}_0$ , describes the voltage and current relationships at each end of the line.

### 2.2.1 Characteristic Admittance Matrix

Appendix A already defines the characteristic admittance matrix as:

$$\boxed{\mathbf{Y}_0 = \mathbf{Z}^{-1}\Gamma = \mathbf{Z}^{-1}\sqrt{\mathbf{ZY}}} \quad (2-16)$$

Note that this matrix, by construction, approaches a constant at high frequencies.

### 2.2.2 Propagation Matrix

The propagation matrix is derived here.

Take the current waves,  $\mathbf{I}_a$  and  $\mathbf{I}_b$ . The forward-travelling wave,  $\mathbf{I}_a$ , at positions  $x$  and  $x + \Delta x$  is defined as:

$$\frac{\mathbf{I}_x + \mathbf{Y}_0 \mathbf{V}_x}{2} = (e^{-\Gamma x})^T \mathbf{I}_a \equiv \mathbf{I}_{ax} \quad (2-17)$$

and

$$\frac{\mathbf{I}_{x+\Delta x} + \mathbf{Y}_0 \mathbf{V}_{x+\Delta x}}{2} = (e^{-\Gamma x} e^{-\Gamma \Delta x})^T \mathbf{I}_a = (e^{-\Gamma \Delta x})^T (e^{-\Gamma x})^T \mathbf{I}_a = (e^{-\Gamma \Delta x})^T \mathbf{I}_{ax} \quad (2-18)$$

Similarly, the backward-travelling wave,  $\mathbf{I}_b$ , at positions  $x$  and  $x - \Delta x$  is defined as:

$$\frac{\mathbf{I}_x - \mathbf{Y}_0 \mathbf{V}_x}{2} = (e^{\Gamma x})^T \mathbf{I}_b \equiv \mathbf{I}_{bx} \quad (2-19)$$

and

$$\frac{\mathbf{I}_{x-\Delta x} - \mathbf{Y}_0 \mathbf{V}_{x-\Delta x}}{2} = (e^{\Gamma x} e^{-\Gamma \Delta x})^T \mathbf{I}_b = (e^{-\Gamma \Delta x})^T (e^{\Gamma x})^T \mathbf{I}_b = (e^{-\Gamma \Delta x})^T \mathbf{I}_{bx} \quad (2-20)$$

Thus, there is a matrix, call it  $\mathbf{H}$ , that can be used to find the value of one of the travelling waves farther down the line, along the direction of travel. It is defined as:

$$\boxed{\mathbf{H} = (e^{-\Gamma \Delta x})^T = e^{-\sqrt{YZ} \Delta x}} \quad (2-21)$$

Note that this matrix, by construction, approaches zero at high frequencies.

### 2.3 Chapter Summary

This chapter showed the construction of a frequency dependent model for an overhead transmission line. Also shown was how to calculate the unit-length impedance and admittance matrices to be used in the current and voltage formulation of the model. These matrices are used to build the *propagation* and *characteristic* matrices.

The propagation matrix describes how current and voltage waves cross the transmission line, and includes time delays and attenuation. The characteristic admittance matrix describes the voltage and current relationships at each end of the line.

Also discussed was how to reduce the order of the matrices by eliminating the ground wires.

## CHAPTER III

# FREQUENCY DOMAIN MODELS AND RATIONAL FUNCTIONS

The next step in the process of obtaining a time domain model for a transmission line will be to approximate the frequency domain model by rational functions. Rational functions lead to a computationally efficient way of calculating the time domain results. This chapter will show how and why they are even considered in the first place, and Ch. V will show how they are actually derived from a frequency domain model.

First, let's illustrate why rational functions are even considered in the first place.

Given a frequency domain model or *transfer function*,  $F(s)$ , and an input,  $U(s)$ , one finds the output,  $X(s)$ , using the expression:

$$X(s) = F(s)U(s) \quad (3-1)$$

The output in the time domain,  $x(t)$ , is found by convolution instead of multiplication as in the frequency domain:

$$x(t) = f(t) \otimes u(t) = \int_0^t f(t - \tau)u(\tau)d\tau \quad (3-2)$$

If we consider discrete time steps of  $\Delta t$  instead of continuous time, where the  $n^{\text{th}}$  time step,  $x(n\Delta t)$ , is denoted as  $x_n$ , then we can modify the equation slightly:

$$\begin{aligned} x_n &= \int_0^{(n-1)\Delta t} f(t - \tau)u(\tau)d\tau + \int_{(n-1)\Delta t}^{n\Delta t} f(t - \tau)u(\tau)d\tau \\ &= x_{n-1} + \int_{(n-1)\Delta t}^{n\Delta t} f(t - \tau)u(\tau)d\tau \end{aligned} \quad (3-3)$$

This is a recursive relation, and further equations using this form to perform a convolution will be termed *recursive convolution*. In other words, this is one technique to perform convolution in the time domain using discrete time steps.

It is immediately clear that

1. We need the time domain form of  $F(s)$ , and
2. One integration per time step needs to be performed.

Depending on the complexity of the function, computing the output in the time domain may be computationally expensive. Certain functions, however, have a form which leads to a computationally efficient time domain relation. Rational functions are one such function type.

### 3.1 Convolution with Rational Functions

Convolution with rational functions, as we shall see shortly, leads to the computationally efficient time domain recursive relation:

$$x_n = \alpha u_n + \beta u_{n-1} + \gamma x_{n-1} \quad (3-4)$$

The three coefficients,  $\alpha$ ,  $\beta$ , and  $\gamma$ , are calculated using a variety of methods, two of which are derived below.

Say we have a rational function which represents our transfer function:

$$F(s) = \frac{c}{s-a} \quad (3-5)$$

The variables  $c$  and  $a$  are real constants, and  $s$  is the usual frequency variable. Given the input,  $U$ , and output,  $X$ , (dropping the  $s$  parameter for convenience) Eq. (3-1) becomes:

$$X = \frac{c}{s-a} U \quad (3-6)$$

This is a state equation, where:

$$sX = aX + cU \quad (3-7)$$

In the time domain, this is a differential equation of the form

$$\dot{x} = ax + cu, \quad (3-8)$$

with the state equation form

$$\begin{aligned} \dot{x} &= ax + cu \\ y &= x + 0u \end{aligned} \quad (3-9)$$

having the exact solution:

$$x(t) = e^{at} x(0) + c \int_0^t e^{a(t-\tau)} u(\tau) d\tau \quad (3-10)$$

If we consider discrete time steps of size  $\Delta t$  instead of continuous time, and if we again use  $x_n$  to represent  $x(n\Delta t)$ , then this time domain solution can be rewritten through the following derivation.

$$\begin{aligned}
x_n &= e^{an\Delta t} x_0 + c \int_0^{n\Delta t} e^{a(n\Delta t - \tau)} u(\tau) d\tau \\
&= e^{an\Delta t} x_0 + \left[ c \int_0^{(n-1)\Delta t} e^{a(n\Delta t - \tau)} u(\tau) d\tau \right] + \left[ c \int_{(n-1)\Delta t}^{n\Delta t} e^{a(n\Delta t - \tau)} u(\tau) d\tau \right] \\
&= e^{a\Delta t} e^{a(n-1)\Delta t} x_0 + e^{a\Delta t} \left[ c \int_0^{(n-1)\Delta t} e^{a((n-1)\Delta t - \tau)} u(\tau) d\tau \right] \\
&\quad + \left[ c \int_{(n-1)\Delta t}^{n\Delta t} e^{a(n\Delta t - \tau)} u(\tau) d\tau \right] \\
&= e^{a\Delta t} x_{n-1} + c \int_{(n-1)\Delta t}^{n\Delta t} e^{a(n\Delta t - \tau)} u(\tau) d\tau
\end{aligned} \tag{3-11}$$

Now, depending on the form of the input,  $u(t)$ , the integration can be performed in a number of different ways. Two integration techniques will be considered in this project, both assuming that  $u(t)$  varies linearly over each time step.

### 3.1.1 Direct Trapezoidal Integration

This technique leads to the first method of calculating the coefficients of Eq. (3-4).

The relation in Eq. (3-8) can be integrated directly:

$$\int_0^t \frac{d}{d\tau} x(\tau) d\tau = \int_0^t (ax(\tau) + cu(\tau)) d\tau \tag{3-12}$$

Thus,

$$\begin{aligned}
x_n &= x_0 + \int_0^{n\Delta t} (ax(\tau) + cu(\tau)) d\tau \\
&= x_0 + \int_0^{(n-1)\Delta t} (ax(\tau) + cu(\tau)) d\tau + \int_{(n-1)\Delta t}^{n\Delta t} (ax(\tau) + cu(\tau)) d\tau \\
&= x_{n-1} + \int_{(n-1)\Delta t}^{n\Delta t} (ax(\tau) + cu(\tau)) d\tau
\end{aligned} \tag{3-13}$$

If trapezoidal integration is used to compute the integral, then the three recursive coefficients become:

$$\begin{aligned}
\alpha &= \left(1 - \frac{a\Delta t}{2}\right)^{-1} \frac{c\Delta t}{2} \\
\beta &= \left(1 - \frac{a\Delta t}{2}\right)^{-1} \frac{c\Delta t}{2} \\
\gamma &= \left(1 - \frac{a\Delta t}{2}\right)^{-1} \left(1 + \frac{a\Delta t}{2}\right)
\end{aligned} \tag{3-14}$$

### 3.1.2 Assume Input is Linear

This technique leads to the second method of calculating the coefficients of Eq. (3-4). It is based on the work of Semlyen and Dabuleanu [7].

The input,  $u(t)$ , is assumed to vary linearly over each time step, and trapezoidal integration is then applied to Eq. (3-11) to produce these coefficient values:

$$\begin{aligned}
\alpha &= -\frac{c}{a} \left(1 + \frac{1 - e^{a\Delta t}}{a\Delta t}\right) \\
\beta &= \frac{c}{a} \left(e^{a\Delta t} + \frac{1 - e^{a\Delta t}}{a\Delta t}\right) \\
\gamma &= e^{a\Delta t}
\end{aligned} \tag{3-15}$$

Note that if the bilinear transform,  $e^x = e^{x/2}/e^{-x/2} \approx (1 + x/2)/(1 - x/2)$ , is used to approximate the exponential terms in Eq. (3-15), then Eq. (3-14) can be shown to be equivalent.

## 3.2 A Note on the Stability of Trapezoidal Integration

It can be shown that trapezoidal integration is always stable [8]. However, there is a subtle instability that can creep into the above recursive calculations.

The standard conductance-plus-current-source representation of an inductor will be used to explain this. We will start with the mathematical description, and then use trapezoidal integration to produce a recursive relation for the current.

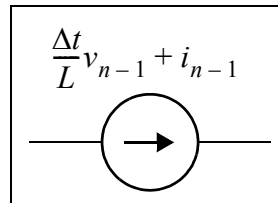
$$i(t) = \frac{1}{L} \int_{t-\Delta t}^t v(t) dt + i(t - \Delta t) \tag{3-16}$$

turns into the recursive relation:

$$i_n = \frac{\Delta t}{2L}v_n + \left[ \frac{\Delta t}{2L}v_{n-1} + i_{n-1} \right] \quad (3-17)$$

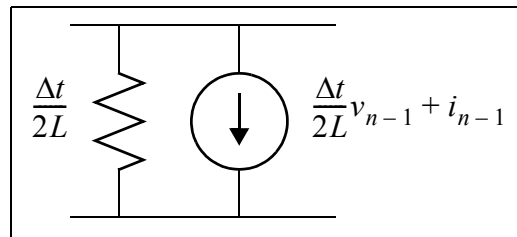
The present value of the current,  $i_n$ , depends on the present value of the input voltage,  $v_n$ . This presents a problem since the dependency may also be the other way around, with the voltage ultimately depending on the current. Said in another way, the present state depends on the present input.

One solution is to assume that the voltage does not change very much between time steps, so that  $v_n \approx v_{n-1}$ . This obviously changes the entire equation, and thus the relation is no longer precisely trapezoidal integration. One way of understanding this approximation is to look at it as just a current source that only depends on previous values of current and voltage, and that varies with each time step. Fig. 3-1 shows what this looks like in circuit form. This solution does not necessarily maintain the stability of the integration.



**Fig. 3-1.** Approximation of present input using previous time step.

Another solution is to view the factor in front of the  $v_n$  term as a conductance that participates in the admittance network of the entire system. This will produce the correct value of  $v_n$  in the present time step. The only disadvantage is that the admittance changes if the time step changes, and so the network may need to be reanalyzed. Fig. 3-2 shows this in circuit form.



**Fig. 3-2.** No approximation of present input.

This is how the recursive equations for a transmission line will be utilized. The present input will be incorporated using a conductance. This is the technique developed by Dommel [9].

### 3.3 Terms with Conjugate Pairs

It is possible for a rational function to have complex values for  $c$  and  $a$ . One fortunate constraint is that all complex values must exist in complex conjugate pairs. In other words, if  $c$  and  $a$  exist in the rational function, then  $\bar{c}$  and  $\bar{a}$  must exist as well.

The reason for this constraint is that we are working strictly with a real-world system. These can never be represented by complex numbers, only real numbers. Rational functions, in order to produce real results, must have real-valued coefficients, and complex numbers existing only in complex conjugate pairs ensure this condition. Please refer to Appendix C for a proof and further discussion of this fact. Also ensured is the causality of the transfer function.

This section shows how to utilize any complex pairs, but stay in the real domain.

Take, then, this rational function:

$$F(s) = \frac{c}{s-a} + \frac{\bar{c}}{s-\bar{a}} \quad (3-18)$$

When an input is applied to this transfer function,

$$X = \left( \frac{c}{s-a} + \frac{\bar{c}}{s-\bar{a}} \right) U = \frac{c}{s-a} U + \frac{\bar{c}}{s-\bar{a}} U, \quad (3-19)$$

the output is calculated using the recursive relation:

$$x_n = x_{1_n} + x_{2_n} = [\alpha_1 u_n + \beta_1 u_{n-1} + \gamma_1 x_{1_{n-1}}] + [\alpha_2 u_n + \beta_2 u_{n-1} + \gamma_2 x_{2_{n-1}}] \quad (3-20)$$

The subscript of each coefficient indicates to which of the two partial fractions it belongs.

Without much effort, it can be shown that  $\alpha_2 = \bar{\alpha}_1$ ,  $\beta_2 = \bar{\beta}_1$ , and  $\gamma_2 x_{2_{n-1}} = \overline{\gamma_1 x_{1_{n-1}}}$ .

Therefore, rewrite Eq. (3-20) as:

$$x_n = 2Re\{\alpha_1\}u_n + 2Re\{\beta_1\}u_{n-1} + 2Re\{\gamma_1 x_{1_{n-1}}\} \quad (3-21)$$

The “new” coefficients and final term are now:

$$\begin{aligned} 2Re\{\alpha_1\} &= 2Re\left\{-\frac{c}{a}\left(1 + \frac{1-e^{a\Delta t}}{a\Delta t}\right)\right\} \\ &= -\frac{2}{|a|^2}\left[X + \frac{1}{\Delta t|a|^2}(X(a'A - a''B) + Y(a'B + a''A))\right] \end{aligned} \quad (3-22)$$



$$\begin{aligned}
 2Re\{\beta_1\} &= 2Re\left\{\frac{c}{a}\left(e^{a\Delta t} + \frac{1 - e^{a\Delta t}}{a\Delta t}\right)\right\} \\
 &= \frac{2}{|a|^2}\left[X\left(1 - A + \frac{(a'A - a''B)}{\Delta t|a|^2}\right) - Y\left(B - \frac{(a'B + a''A)}{\Delta t|a|^2}\right)\right],
 \end{aligned} \tag{3-23}$$

and

$$\begin{aligned}
 2Re\{\gamma_1 x_{1_{n-1}}\} &= 2Re\left\{e^{a\Delta t} x_{1_{n-1}}\right\} \\
 &= 2e^{a'\Delta t}(\cos(a''\Delta t)Re\{x_{1_{n-1}}\} - \sin(a''\Delta t)Im\{x_{1_{n-1}}\})
 \end{aligned} \tag{3-24}$$

The auxiliary variables,  $X$ ,  $Y$ ,  $A$ , and  $B$ , are defined as

$$\begin{aligned}
 X &= (c'a' + c''a'') & A &= 1 - e^{a'\Delta t} \cos(a''\Delta t) \\
 Y &= (c''a' - c'a'') & B &= e^{a'\Delta t} \sin(a''\Delta t)
 \end{aligned} \tag{3-25}$$

where the complex variables,  $c$  and  $a$ , are defined as  $c = c' + jc''$  and  $a = a' + ja''$ . It is convenient to write these expressions in this way when creating a computer implementation.

For completeness, the imaginary parts of the  $\alpha_1$  and  $\beta_1$  coefficients are derived:

$$Im\{\alpha_1\} = -\frac{1}{|a|^2}\left[Y + \frac{1}{\Delta t|a|^2}(Y(a'A - a''B) - X(a'B + a''A))\right] \tag{3-26}$$

$$Im\{\beta_1\} = \frac{1}{|a|^2}\left[Y\left(1 - A + \frac{(a'A - a''B)}{\Delta t|a|^2}\right) + X\left(B - \frac{(a'B + a''A)}{\Delta t|a|^2}\right)\right] \tag{3-27}$$

### 3.3.1 Terms with Conjugate Pairs, a State Variable Approach

Leading into the topic of higher order poles, this subsection shows how to use the state variable approach from Appendix F to derive an alternative approach to finding a recursive formula for the terms with conjugate pairs.

Let  $a = a' + ja''$  and  $c = c' + jc''$ . We now consider the relation:

$$Y = \left(\frac{c}{s - a} + \frac{\bar{c}}{s - \bar{a}}\right)U \tag{3-28}$$

Rewrite this as:

$$(s^2 - 2a's + |a|^2)Y = (s - \bar{a})cU + (s - a)\bar{c}U = s(2c')U - 2(a'c' + a''c'')U \quad (3-29)$$

In the time domain, this has the form:

$$\ddot{y} - 2a'\dot{y} + |a|^2y = 2c'\dot{u} - 2(a'c' + a''c'')u \quad (3-30)$$

The method of Appendix F can turn this into a first-order form using matrices:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -|a|^2 & 2a' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2c' \\ -2(a'c' + a''c'') + 2a'2c' \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 0u \end{aligned} \quad (3-31)$$

This is in the standard state equation form:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du \end{aligned} \quad (3-32)$$

Therefore, the update equation is

$$\begin{aligned} \mathbf{x}_n &= \alpha u_n + \beta u_{n-1} + \gamma \mathbf{x}_{n-1} \\ y_n &= x_{1_n} \end{aligned}, \quad (3-33)$$

where:

$$\begin{aligned} \alpha &= -A^{-1} \left( I + A^{-1} \frac{I - e^{A\Delta t}}{\Delta t} \right) B \\ \beta &= A^{-1} \left( e^{A\Delta t} + A^{-1} \frac{I - e^{A\Delta t}}{\Delta t} \right) B \\ \gamma &= e^{A\Delta t} \end{aligned} \quad (3-34)$$

For convenience, the following is derived analytically:

$$e^{A\Delta t} = \frac{e^{a'\Delta t}}{a''} \begin{bmatrix} a'' \cos(a''\Delta t) - a' \sin(a''\Delta t) & \sin(a''\Delta t) \\ -|a|^2 \sin(a''\Delta t) & a'' \cos(a''\Delta t) + a' \sin(a''\Delta t) \end{bmatrix} \quad (3-35)$$

Trapezoidal integration gives:

$$\begin{aligned}\alpha &= \beta = \left(I - \frac{A\Delta t}{2}\right)^{-1} B \frac{\Delta t}{2} \\ \gamma &= \left(I - \frac{A\Delta t}{2}\right)^{-1} \left(I + \frac{A\Delta t}{2}\right)\end{aligned}\tag{3-36}$$

The method that does not use this state equation form is the preferred way to implement the recursive convolution on a computer.

### 3.4 Terms with Higher Order Poles, a Future Direction

This section shows how to represent higher order poles in the time domain, but using a state variable approach. This subject will also be discussed in a later section (Sec. 5.11 in the chapter on Vector Fitting).

The following discussion uses the variable  $p$  instead of  $a$  to represent the poles in order to avoid confusion with the use of  $a_i$  as coefficients of the derivatives of  $y$ .

If  $p$  is real, then the output corresponding to this pole is

$$Y = \frac{c}{(s-p)^n} U,\tag{3-37}$$

which, in the time domain, becomes

$$\left( \sum_{k=0}^{n-1} \binom{n-1}{k} D^{n-1-k} (-p)^k \right) y = cu,\tag{3-38}$$

where  $D$  is the differential operator.

This is an  $n^{\text{th}}$  order system with

$$a_i = \binom{n-1}{i} (-p)^i \text{ and } b_i = \begin{cases} c & \text{if } i = n-1 \\ 0 & \text{otherwise} \end{cases},\tag{3-39}$$

if we use the state equation formulation described in Appendix F.

For complex poles, construct a state equation using conjugate pairs to numerically guarantee real results. We proceed as follows.

$$\begin{aligned}
 Y &= \left( \frac{c}{(s-p)^n} + \frac{\bar{c}}{(s-\bar{p})^n} \right) U \\
 &= \frac{c(s-\bar{p})^n + \bar{c}(s-p)^n}{((s-p)(s-\bar{p}))^n} U \\
 &= \frac{c(s-\bar{p})^n + \bar{c}(s-p)^n}{(s(s-2p') + |p|^2)^n} U
 \end{aligned} \tag{3-40}$$

Using the binomial expansion theorem:

$$\begin{aligned}
 &\left( \sum_{k=0}^{\infty} \binom{n}{k} s^{n-k} |p|^{2k} \left[ \sum_{r=0}^{\infty} \binom{n-k}{r} s^{n-k-r} (-2p')^r \right] \right) Y \\
 &= \left( \sum_{k=0}^n \binom{n}{k} s^{n-k} (c\bar{p}^k + \bar{c}p^k) \right) U
 \end{aligned} \tag{3-41}$$

In the time domain, this is a differential equation of order  $2n$ :

$$\left( \sum_{k=0}^n \binom{n}{k} |p|^{2k} \left[ \sum_{r=0}^{n-k} \binom{n-k}{r} D^{2(n-k)-r} (-2p')^r \right] \right) y = \left( \sum_{k=0}^n \binom{n}{k} D^{n-k} 2\operatorname{Re}\{c\bar{p}^k\} \right) u \tag{3-42}$$

If this expression is written out in table form, a method to calculate each coefficient,  $a_i$ , of the derivatives of  $y$  becomes evident. Fig. 3-3 shows pseudocode that accomplishes this.

```

if (i <= n) {
    k = 0
    r = i
    j = n
} else {
    k = i - n
    r = 2n - i
    j = r
}
a_i = 0
while (r >= 0) {
    a_i = a_i + choose(n,k) * |p|2k * choose(j,r) * (-2p')
    k = k + 1
    r = r - 2
    j = j - 1
}
    
```

**Fig. 3-3.** Pseudocode for calculating each  $a_i$ .

Each coefficient,  $b_i$ , of the derivatives of  $u$  is easier to calculate, and can be read directly from Eq. (3-39). For completeness, Fig. 3-4 shows pseudocode that can do this.

$$b_i = 2\text{Re}\{c\bar{p}^i\} \cdot \text{choose}(n, i)$$

**Fig. 3-4.** Pseudocode for calculating each  $b_i$ .

This section was intended to shed some light on how to simulate more elaborate models, however, the procedure presented here was not implemented for this project.

### 3.4.1 An Alternative Formulation

It is possible, in a way similar to the technique in Sec. 3.3, to perform the calculation without matrices. However, this is left as an exercise to the reader.

## 3.5 Chapter Summary

This chapter has shown why it is useful to represent a frequency domain model using rational functions. Specifically, an input can be applied to the time domain representation of a rational function in a computationally efficient way.

Also shown was how to stay in the real domain, so that a computer does not have to utilize complex arithmetic.

## CHAPTER IV

# MODELLING A TRANSMISSION LINE USING RATIONAL FUNCTIONS

In the previous two chapters, we developed a frequency domain model for an overhead transmission line, and showed how to utilize rational functions to produce a computationally efficient time domain simulation. This chapter shows specifically how to apply a rational function approximation to the transmission line model.

The next chapter will show how to find a rational approximation to a curve by first finding the poles, and then finding the residues.

### 4.1 Characteristic Admittance Matrix

For convenience, the definition of the characteristic admittance matrix,  $\mathbf{Y}_0$ , from Eq. (2-16) will be repeated here:

$$\mathbf{Y}_0 = \mathbf{Z}^{-1} \Gamma = \mathbf{Z}^{-1} \sqrt{\mathbf{ZY}} \quad (4-1)$$

It was stated that, by construction, this matrix approaches a constant at high frequencies. A suitable form for the rational approximation, therefore, is a biproper function:

$$Y_{ij}(s) \approx \sum_m \frac{(c_m)_{ij}}{s - a_m} + d_{ij} \quad (4-2)$$

Notice that the same set of poles, but a different set of residues, are used for each element of this matrix. This is a useful restriction since it will make the time domain computation more efficient. The following derivation, which is based on the work in [10] and [11], explains why. It is based on the modes of the matrix.

Let  $\mathbf{T}$  be the eigenvector matrix, and let  $\lambda$  be the eigenvalue matrix – these are the modes. Then:

$$\mathbf{Y}_0 = \mathbf{Z}^{-1} \sqrt{\mathbf{ZY}} = \mathbf{T} \lambda \mathbf{T}^{-1} \quad (4-3)$$

Right-multiplying a matrix by a diagonal matrix multiplies each column by the corresponding term in the diagonal matrix, so rewrite this as

$$\mathbf{Y}_0 = \begin{bmatrix} \mathbf{T}_1 \lambda_1 & \mathbf{T}_2 \lambda_2 & \dots & \mathbf{T}_n \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{T}_1^{inv} \\ \mathbf{T}_2^{inv} \\ \vdots \\ \mathbf{T}_n^{inv} \end{bmatrix}, \quad (4-4)$$

where  $\mathbf{T}_k$  is the  $k^{\text{th}}$  column of  $\mathbf{T}$ , and  $\mathbf{T}_k^{inv}$  is the  $k^{\text{th}}$  row of  $\mathbf{T}^{-1}$ . Note that  $\mathbf{T}_k^{inv}$  does not refer to the inverse of  $\mathbf{T}_k$ .

Next, write:

$$\mathbf{Y}_0 = \sum_{k=1}^n \mathbf{T}_k \lambda_k \mathbf{T}_k^{inv} = \sum_{k=1}^n (\mathbf{T}_k \mathbf{T}_k^{inv}) \lambda_k \quad (4-5)$$

For visual purposes, the frequency dependence is added to this equation:

$$\mathbf{Y}_0(j\omega) = \sum_{k=1}^n (\mathbf{T}_k(j\omega) \mathbf{T}_k^{inv}(j\omega)) \lambda_k(j\omega) \quad (4-6)$$

The authors of [10] hypothesize that it is only necessary to find the poles of each mode, as opposed to the whole expression in the summation, and that the  $\mathbf{T}_k \mathbf{T}_k^{inv}$  factor can be neglected. The reason given is that the frequency variation in  $\mathbf{T}_k \mathbf{T}_k^{inv}$  is always accompanied by a frequency variation in the mode, and that the final calculation to find the residues will account for this approximation.

Another perspective for neglecting this factor is that the modes are basis functions, and any function, in some linear space, is a linear combination of its basis functions. The linear multiplier, therefore, will appear “automagically” when finding the residues.

Therefore, if we neglect this factor, we can write:

$$\mathbf{Y}_0 = \sum_{k=1}^n \lambda_k = \sum_{k=1}^n Y_{kk} \quad (4-7)$$

It is known from matrix theory that the sum of the eigenvalues equals the trace. This means that it is not necessary to diagonalize  $\mathbf{Y}_0$ , and that only  $n$  diagonal elements need to be utilized to find the poles.

Once the poles are found, all that remains is to calculate the residues, and this is accomplished using a least-squares approach. In other words, solve

$$Y_{ij}(j\omega) \approx \sum_{m=1}^N \frac{(c_m)_{ij}}{j\omega - a_m} + d_{ij} \quad (4-8)$$

for the  $(c_m)_{ij}$  values. This is of the form  $\mathbf{AX} = \mathbf{B}$ , where, for  $M$  frequency samples,

$$\mathbf{A} = \begin{bmatrix} \frac{1}{j\omega_1 - a_1} & \cdots & \frac{1}{j\omega_1 - a_N} & 1 \\ \vdots & & \vdots & \vdots \\ \frac{1}{j\omega_M - a_1} & \cdots & \frac{1}{j\omega_M - a_N} & 1 \end{bmatrix}, \quad (4-9)$$

$$\mathbf{X} = \begin{bmatrix} (c_1)_{11} & \cdots & (c_1)_{nn} \\ \vdots & & \vdots \\ (c_N)_{11} & \cdots & (c_N)_{nn} \\ d_{11} & \cdots & d_{nn} \end{bmatrix}, \text{ and} \quad (4-10)$$

$$\mathbf{B} = \begin{bmatrix} Y_{11}(j\omega_1) & \cdots & Y_{nn}(j\omega_1) \\ \vdots & & \vdots \\ Y_{11}(j\omega_M) & \cdots & Y_{nn}(j\omega_M) \end{bmatrix}. \quad (4-11)$$

Solving this equation using a standard technique, such as QR or SVD decomposition will give a least-squares solution. It is also possible to perform some sort of iterative refinement on the solution by choosing a suitable search procedure.

## 4.2 Propagation Matrix

The definition of the propagation matrix,  $\mathbf{H}$ , from Eq. (2-21) is repeated here:

$$\mathbf{H} = (e^{-\Gamma\Delta x})^T = e^{-\sqrt{YZ}\Delta x} \quad (4-12)$$

By construction, this matrix approaches zero at high frequencies, so a suitable rational approximation might look like:



$$H_{ij}(s) \approx \sum_m \frac{(c_m)_{ij}}{s - a_m} \quad (4-13)$$

At first glance, this approximation may appear to satisfy the approaching-zero criterion. However, this approximation would need a very high order to accurately approximate the exponential. An exponential in the frequency domain, recall, is actually a time delay.

It has been suggested to first extract a time delay for each element of  $\mathbf{H}$ , and then fit the remaining function, but Gustavsen et al. [11] point out that this can cause oscillating behaviour in the frequency domain, especially for lines with high ground resistivity. This would mean that a higher order fitting is required.

One way around this is to extract a time delay for each mode using an appropriate line model. The authors of [11] describe why the ‘‘Universal Line Model’’ [10] is a good approach to use. This model is based on the calculation of unknown residues given known poles and time delays. It also assumes that all poles contribute to all elements of  $\mathbf{H}$ .

#### 4.2.1 Fitting the Propagation Matrix, $\mathbf{H}$ , in the Modal Domain

This is just the first step in the process of fitting  $\mathbf{H}$ ; the phase domain fitting is the final step. An approach similar to that of the previous section is used. First, diagonalize  $\mathbf{H}$ :

$$\mathbf{H} = e^{-\sqrt{\mathbf{Y}\mathbf{Z}}l} = \mathbf{T}e^{-\gamma l}\mathbf{T}^{-1}, \quad (4-14)$$

where  $l$  is the transmission line length.

Next, express this in a manner similar to Eq. (4-5):

$$\mathbf{H} = \sum_{k=1}^n (\mathbf{T}_k \mathbf{T}_k^{inv}) e^{-\gamma_k l} = \sum_{k=1}^n (\mathbf{T}_k \mathbf{T}_k^{inv}) e^{-\gamma_k' l} e^{-j\omega \tau_k} \quad (4-15)$$

Note that we have extracted a constant delay,  $\tau_k$ , for each mode. The  $\mathbf{T}_k \mathbf{T}_k^{inv}$  and  $e^{-\gamma_k' l}$  factors are still frequency dependent.

Finally, for the same reasoning as in the previous section, discard the  $\mathbf{T}_k \mathbf{T}_k^{inv}$  factor and express  $\mathbf{H}$  as:

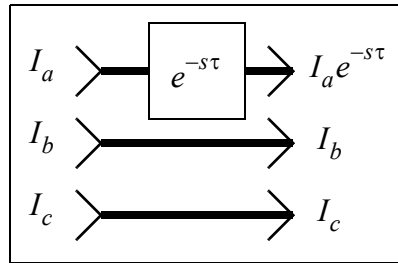
$$\mathbf{H}(j\omega) = \sum_{k=1}^n e^{-\gamma_k'(j\omega)l} e^{-j\omega \tau_k} \quad (4-16)$$

This expression is complicated by the fact that each mode has a different extracted delay. This means that we must perform the diagonalization to find each mode, unlike  $\mathbf{Y}_0$ .

#### 4.2.2 A Thought Experiment with Modal Time Delays

This section will show why it is necessary to diagonalize the matrix and work with each mode, rather than stay in the phase domain, if non-frequency-dependent line delays are to be considered.

Given a 3-phase transmission line, assume that there exists a box with which we can apply a pure time delay on one of the lines. Fig. 4-1 shows what this looks like.



**Fig. 4-1.** Three-phase transmission line with a perfect delay.

Also assume that this is a very short segment of the line, so that the characteristic admittance has no effect.

Our task is to find what a transformation matrix,  $\mathbf{M}$ , looks like so that

$$\mathbf{M} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} I_a e^{-s\tau} \\ I_b \\ I_c \end{bmatrix} \quad (4-17)$$

It can be shown, for any arbitrary relationship between the  $I_x$  components, that the only solution for  $\mathbf{M}$  is the diagonal matrix:

$$\begin{bmatrix} e^{-s\tau} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-18)$$

In other words, the only way for a delay to affect only one phase is if the transformation matrix is diagonal. Hence, finding the  $n$  delays for an  $n$ -phase transmission line only makes sense in the modal domain.

### 4.2.3 Finding the Modes of the Propagation Matrix, $H$

The modes of  $H$  are determined by using a frequency-dependent transformation matrix,  $T$ :

$$H = T e^{-\gamma l} T^{-1} \quad (4-19)$$

There are several techniques to find these matrices. The first is just a simple eigenvector finding routine. However, it is often the case that “corresponding” eigenvalues switch places at adjacent frequencies. For example, mode 1 may occupy position 1 for some first set of frequencies, and then occupy position 3.

One technique that proposes to get around this problem is discussed by Wedepohl et al. [12]. It uses the Newton-Raphson (NR) method to find the eigenvectors and eigenvalues for the next frequency point using the values from the current frequency point as a starting guess.

Another technique, based on sequential quadratic programming (SQP) is presented by Nguyen and Chan [13].

The Newton-Raphson method, as well as the “check” method of Sec. 6 in [12], were implemented for this project, however, neither technique worked properly. The NR method sometimes needed frequency samples that were closer together than what was generated, and the implementation was not improved due to time constraints. The “check” method, which works by correlating the current frequency sample with the previous frequency sample, sometimes could not correctly determine the correct values. This was also probably due to the frequency sample spacing issue.

The method that was actually used was to utilize a constant transformation matrix, calculated at some high frequency, for every frequency point. This may be less accurate than finding the frequency dependent transformation matrices, however, it is a reasonable solution. Furthermore, continuously-transposed lines have something close to a constant transformation matrix, so we can assume that the simulator works for this case.

### 4.2.4 Finding the Delay for Each Mode of $H$

Once the modes are found, a constant time delay inherent to each mode is found. While this ignores the fact that the time delay for a given mode may actually be frequency dependent, it nevertheless simplifies the model.

This has several advantages:

- Calculations for each end of the line are decoupled. In other words, the results on either side will not appear instantaneously on the other end.
- A lower order fitting can be used if the delay is factored out.

Because a time delay is an exponential in the frequency domain, it may be expressed as a Taylor expansion with poles of ever-increasing order. Removing this factor will remove these higher order terms, thus decreasing the number of poles required. This is the reason why a lower order fitting is sufficient in this case.

Gustavsen and Semlyen [14] outline a procedure for extracting the delay and a minimum phase shift function, given a scalar transfer function. Let this transfer function be one of the modes, and express it as

$$M(j\omega) \approx m(j\omega)e^{-j\omega\tau}, \quad (4-20)$$

where  $m(j\omega)$  is the minimum phase shift function, and  $\tau$  is the time delay. This relation is approximate because  $\tau$  is only accurate for a single selected frequency, as we shall soon see.

Clearly,

$$|m(j\omega)| = |M(j\omega)|, \quad (4-21)$$

so it is trivial to find this component of  $m(j\omega)$ . The phase is trickier to find, and relies on a formula developed by Bode [15]. This is further explained in Appendix E, but a brief description will be given here:

If the magnitude of the function is known, then the phase at a specific frequency,  $\omega_c$ , of the *minimum phase shift* function with the same magnitude, can be found using the expression

$$B_c = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{dA}{du} \ln \coth \frac{|u|}{2} du, \quad (4-22)$$

where  $B_c$  is the phase at  $\omega_c$ ,  $A$  is the log of the magnitude, and  $u = \ln \frac{\omega}{\omega_c}$ .

An equation for  $\tau$  will now be derived.

First, rewrite  $M(j\omega)$  as

$$M(j\omega) = e^{-(\alpha(\omega) + j\beta(\omega))l} \approx m(j\omega)e^{-j\omega\tau}, \quad (4-23)$$

where  $\alpha(\omega)$  is the attenuation,  $\omega/\beta(\omega)$  is the phase velocity, and  $l$  is the line length.

Next, solve for  $\tau$ , and express it in terms of a single frequency point,  $\Omega$ :

$$-\beta(\omega)l = \angle m(j\omega) - \omega\tau \quad (4-24)$$

$$\tau = \frac{\beta(\Omega)l + \angle m(j\Omega)}{\Omega} \quad (4-25)$$

It should be clearer why the relation in Eqs. (4-20) and (4-23) is only approximate: the value of  $\tau$  is based only on the single frequency point,  $\Omega$ .

Finally, find the phase shift of  $m(j\omega)$  at  $\Omega$ :

$$\angle m(j\Omega) = \frac{\pi}{2} \cdot \frac{d}{d(\ln \omega)} \ln |m(j\omega)| \Big|_{\omega = \Omega} + \Delta(u), \quad (4-26)$$

where

$$\Delta(u) = \frac{1}{\pi} \int_{-\infty}^{\infty} \left( \frac{d}{du} \ln |m(j\omega)| - \frac{d}{du} \ln |m(j\omega)| \Big|_{u=0} \right) \ln \coth \frac{|u|}{2} du \quad (4-27)$$

and

$$u = \ln \frac{\omega}{\Omega}. \quad (4-28)$$

The discussion by Gustavsen and Semlyen in [14] informs us that the first term alone will give a good estimate for the phase, but that including the second term will give an even better result.

The choice of  $\Omega$  is important, as we need to find a representative value for  $\tau$ . Bode shows in Fig. 14.2 in [15], that for a decaying attenuation function,  $A(\omega)$ , the phase approaches the frequency axis, and becomes close near the point where  $A(\omega)$  decays to zero. Gustavsen and Semlyen call this the “toe portion” [14] of  $A(\omega)$ . Hence, a good choice for  $\Omega$  is somewhere in this region because the phase is relatively “constant” and is not a large negative value that would result in a negligible phase shift. An additional fact is that frequencies close to zero will cause the  $e^{-j\omega\tau}$  factor to be close to one, hence the delay has almost no effect.

Instead of choosing this to be the point where  $\frac{|M(j\omega)|}{|M(j0)|} = 0.1$  as in [14], however, our implementation will follow the procedure of [11] and use a value farther along the “toe”.

Next, this integral can be calculated numerically in a straightforward manner using forward, backward or central differences for the derivatives, and a simple summation for the integral.

#### 4.2.5 Refining the Calculated Delay

The calculated delay could be refined. For example, Brent's Method, given an upper and lower bound, could potentially find a better approximation. A good value to use for the lower bound would be the delay due to the speed of light in a vacuum. This is actually the smallest possible value.

Gustavsen [16] discusses a procedure for choosing a good upper bound, and provides a different method to find a larger lower bound based on the phase shift of a minimum phase shift function.

However, this is actually a maximum, and the delay due to the speed of light in a vacuum is a better lower bound. This is proven below. The term "lag", used shortly, is from [17], and refers to the negative of the phase shift in the expression  $e^{j\theta}$ . Thus, the "lag" for this expression is  $-\theta$ .

Say the propagation function,  $H$ , contains an exact time delay,  $\tau$ , where

$$\angle H = \angle H_{nodelay} - \omega\tau \quad (4-29)$$

This phase can also be partitioned into the phase of the corresponding minimum phase shift function at  $\Omega$  plus some other delay,  $\tau_m$ :

$$\angle H = \angle H_{min} - \omega\tau_m \quad (4-30)$$

Using the fact that  $|H| = |H_{min}|$ , we can write:

$$H = |H|e^{\angle H} = |H|e^{j\angle H_{nodelay}}e^{-j\omega\tau} = |H_{min}|e^{j\angle H_{min}}e^{-j\omega\tau_m} \quad (4-31)$$

Gathering the phase values:

$$\tau_m + \frac{\angle H_{nodelay} - \angle H_{min}}{\omega} = \tau \quad (4-32)$$

If  $H$ , when multiplied by  $e^{j\omega\tau_m}$ , becomes a minimum phase shift function, then multiplying by  $e^{j\omega\tau}$  will not produce a minimum phase shift function. Hence, since the lag of a

minimum phase shift function is always less than the lag of any other function with the same magnitude [17],

$$-\angle H_{min} < -\angle H_{nodelay} \text{ or equivalently } \angle H_{min} > \angle H_{nodelay}. \quad (4-33)$$

Therefore:

$$\tau_m > \tau \quad (4-34)$$

This means that the delay,  $\tau_m$ , calculated via Bode's method cannot be used as the lower bound in Brent's Method. Observe that this disagrees with point 4 in the conclusions of [16].

#### 4.2.6 Fitting the Propagation Matrix, $H$ , in the Phase Domain

The modes are found, and a constant time delay is extracted for each mode. Finally, we find the poles of each mode,  $k$ , by first extracting the delay:

$$\mathbf{H}_k e^{j\omega\tau_k} = e^{-\gamma_k' l} \quad (4-35)$$

Then we solve for the residues from:

$$H_{ij}(j\omega) \approx \sum_{k=1}^n \left( \sum_{m=1}^{N_k} \frac{(c_{m_k})_{ij}}{j\omega - a_{m_k}} \right) e^{-j\omega\tau_k} \quad (4-36)$$

This, like Eq. (4-8) for  $\mathbf{Y}_0$ , is of the form  $\mathbf{AX} = \mathbf{B}$ , where, for  $M$  frequency samples,

$$\mathbf{A} = \begin{bmatrix} \frac{e^{-j\omega_1\tau_1}}{j\omega_1 - a_{1_1}} & \cdots & \frac{e^{-j\omega_1\tau_1}}{j\omega_1 - a_{N_1}} & \cdots & \cdots & \frac{e^{-j\omega_1\tau_n}}{j\omega_1 - a_{1_n}} & \cdots & \frac{e^{-j\omega_1\tau_n}}{j\omega_1 - a_{N_n}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{e^{-j\omega_M\tau_1}}{j\omega_M - a_{1_1}} & \cdots & \frac{e^{-j\omega_M\tau_1}}{j\omega_M - a_{N_1}} & \cdots & \cdots & \frac{e^{-j\omega_M\tau_n}}{j\omega_M - a_{1_n}} & \cdots & \frac{e^{-j\omega_M\tau_n}}{j\omega_M - a_{N_n}} \end{bmatrix}, \quad (4-37)$$

$$\mathbf{X} = \begin{bmatrix} (c_{1_1})_{11} & \cdots & (c_{1_1})_{nn} \\ \vdots & \vdots & \vdots \\ (c_{N_{k_1}})_{11} & \cdots & (c_{N_{k_1}})_{nn} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ (c_{1_n})_{11} & \cdots & (c_{1_n})_{nn} \\ \vdots & \vdots & \vdots \\ (c_{N_{k_n}})_{11} & \cdots & (c_{N_{k_n}})_{nn} \end{bmatrix}, \text{ and} \quad (4-38)$$

$$\mathbf{B} = \begin{bmatrix} H_{11}(j\omega_1) & \cdots & H_{nn}(j\omega_1) \\ \vdots & \vdots & \vdots \\ H_{11}(j\omega_M) & \cdots & H_{nn}(j\omega_M) \end{bmatrix}. \quad (4-39)$$

This is also solved using a least-squares fitting approach, with some possible iterative refinement.

#### 4.2.7 Mode Grouping

It can be seen that the computation time depends on the number of modes,  $n$ , instead of the number of elements in the matrix. In fact, computational efficiency was one of the main reasons for performing modal decomposition in the first place. What if this could be reduced even further? This subsection shows how. It is based on the work by Morched et al. [10].

First assume that two or more modes share the same time delay,  $\tau$ , and say there are  $A$  modes in this group. The delay can be factored out from these terms, and their partial sum becomes

$$\left( \sum_{k=1}^A (\mathbf{T}_k \mathbf{T}_k^{inv}) e^{-\gamma_k' l} \right) e^{-j\omega\tau} \quad (4-40)$$

for this selection of modes. We assumed that only this factor produces the poles of interest, so instead of finding the poles for each term,  $e^{-\gamma_k' l}$ , let us find the poles for the sum:

$$\sum_{k=1}^A e^{-\gamma_k' l} \quad (4-41)$$



In fact, we may as well find the poles for the average of this group of modes. The least-squares solution will force the factor of  $A$  to appear inside the residues.

Thus, the computational efficiency can be increased by grouping modes that have the same time delay, taking their average, and treating this as a new “supermode”. However, it is unlikely that any mode will have the same exact delay. Instead, they may have delays that are close.

Suppose two modes,  $M_1 e^{-j\omega\tau_1}$  and  $M_2 e^{-j\omega\tau_2}$ , have a delay that differs from the other by  $\Delta\tau$ , so that  $\tau_1 - \tau_2 = \Delta\tau$ . Therefore, multiplying these elements by  $e^{j\omega\tau_1}$  will result in an uncompensated factor with a phase angle of

$$\omega(\tau_1 - \tau_2) = \omega\Delta\tau, \quad (4-42)$$

and a maximum uncompensated phase of  $\Omega\Delta\tau$  for some frequency  $\Omega$ .

Practically, then, if some maximum allowed tolerance, say  $\theta$ , is specified, then modes that have a maximum uncompensated phase shift of

$$\Omega\Delta\tau < \theta \quad (4-43)$$

can be considered to have “close” time delays, and can be lumped together into a “supermode”. The time delay that is used for this supermode is the smallest of the set of delays from each mode.

### 4.3 Future Directions

Brent’s Method, or some other form of calculated delay refinement should be implemented. As well, the delay extraction techniques presented by Achar et al. [19] could be explored.

As well, a robust technique that can be used to find the frequency dependent modal transformation matrices needs to be implemented.

For lossless lines, the propagation function will be flat, and Eq. (4-13) is no longer a valid starting point. A special case of just a phase shift, without performing the fitting, would be a useful addition.

### 4.4 Chapter Summary

This chapter showed how to construct a rational approximation for the frequency domain transmission line model. Both parts of the model, the characteristic admittance and propagation matrices, were addressed.

The propagation matrix approximation contained parts that the characteristic admittance approximation did not. These included a time delay for each mode and the necessity to calculate a frequency dependent transformation matrix at each frequency point. The time delay portion takes into account the exponential in the analytic formula, and reduces the required order of the rational approximation.

Also shown was how to improve computational efficiency by grouping “close” modes.

This chapter assumed that there exists a technique for finding the poles of a rational approximation before finding the residues. The following chapter will show how to find these poles.

# CHAPTER V

## VECTOR FITTING THEORY

The previous chapters developed a rational function approximation for an overhead transmission line model. Assumed was the ability to find the poles of this approximation. This chapter shows how to find these poles using a technique called *Vector Fitting*.

This technique was introduced by Gustavsen and Semlyen [20]. Before discussing this technique, however, a brief introduction will be given so as to provide a context.

### 5.1 A First Approach to Curve Fitting

The coefficients of a rational function can be determined as follows.

Let  $f(s)$  be defined as:

$$f(s) = \frac{a_0 + a_1s + a_2s^2 + \dots + a_Ps^P}{b_0 + b_1s + b_2s^2 + \dots + b_Qs^Q} = \frac{N(s)}{D(s)} \quad (5-1)$$

It can be assumed that  $b_0 = 1$  without loss of generality.

If  $f(s)$  is known for several frequency points,  $s_k = j\omega_k$ , and split into real and imaginary parts [18], then we can write

$$f_{Re}(\omega_k) + jf_{Im}(\omega_k) = \frac{N_{Re}(\omega_k) + jN_{Im}(\omega_k)}{1 + D_{Re}(\omega_k) + jD_{Im}(\omega_k)} \quad (5-2)$$

where

$$\begin{aligned} N_{Re}(\omega_k) &= a_0 - a_2\omega_k^2 + a_4\omega_k^4 - \dots \\ N_{Im}(\omega_k) &= a_1\omega_k - a_3\omega_k^3 + a_5\omega_k^5 - \dots \\ D_{Re}(\omega_k) &= -b_2\omega_k^2 + b_4\omega_k^4 - b_6\omega_k^6 + \dots \\ D_{Im}(\omega_k) &= b_1\omega_k - b_3\omega_k^3 + b_5\omega_k^5 - \dots \end{aligned} \quad (5-3)$$

Multiplying both sides of Eq. (5-2) by the denominator and equating real and imaginary parts, we obtain:

$$\begin{aligned}
N_{Re}(\omega_k) - f_{Re}(\omega_k)D_{Re}(\omega_k) + f_{Im}(\omega_k)D_{Im}(\omega_k) &= f_{Re}(\omega_k) \\
N_{Im}(\omega_k) - f_{Im}(\omega_k)D_{Re}(\omega_k) - f_{Re}(\omega_k)D_{Im}(\omega_k) &= f_{Im}(\omega_k)
\end{aligned} \tag{5-4}$$

In other words, we are solving an equation that looks like:

$$D(s)f(s) = N(s) \tag{5-5}$$

This can be written, using  $M$  samples of  $f(\omega)$ , in  $Ax = b$  form to produce solutions for the  $a_n$  and  $b_n$  coefficients. This is illustrated in Eq. (5-6).

$$\begin{aligned}
\mathbf{x}^T &= [a_0 \ a_1 \ \dots \ a_P \ b_1 \ \dots \ b_Q] \\
\mathbf{b}^T &= [f_{Re}(\omega_1) \ f_{Im}(\omega_1) \ f_{Re}(\omega_2) \ f_{Im}(\omega_2) \ \dots \ \dots \ f_{Re}(\omega_M) \ f_{Im}(\omega_M)] \\
\mathbf{A} &= \text{stack}(\mathbf{A}_k) \quad k = 1, \dots, M \\
\mathbf{A}_k &= \begin{bmatrix} 1 & 0 & -\omega_k^2 & 0 & \omega_k^4 & \dots & f_{Im}(\omega_k)\omega_k & f_{Re}(\omega_k)\omega_k^2 & -f_{Im}(\omega_k)\omega_k^3 & -f_{Re}(\omega_k)\omega_k^4 & \dots \\ 0 & \omega_k & 0 & -\omega_k^3 & 0 & \dots & -f_{Re}(\omega_k)\omega_k & f_{Im}(\omega_k)\omega_k^2 & f_{Re}(\omega_k)\omega_k^3 & -f_{Im}(\omega_k)\omega_k^4 & \dots \end{bmatrix}
\end{aligned} \tag{5-6}$$

If the number of samples,  $M$ , is larger than the number of unknowns,  $P + Q + 1$ , then this produces an overdetermined system. There are several different methods that can be used to solve this to give a unique solution in the least squares sense. In other words, it is possible to produce a solution that minimizes the 2-norm,  $\|\mathbf{Ax} - \mathbf{b}\|_2$ .

These techniques include QR decomposition and Singular Value Decomposition (SVD). Other formulations exist that are equivalent to solving the normal equations,

$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ , but Whitfield [21] showed that solving the system using these may yield unstable transfer functions for stable systems, and that other iterative methods may not converge. QR decomposition and SVD, therefore, are the best candidates.

### 5.1.1 Ill-Conditioning of this Formulation

Note that the highest power of  $\omega$  in  $\mathbf{A}$  will be  $\max(P, Q)$ , the chosen order of the rational function. These large numbers exist in the same matrix as lower powers of  $\omega$ . For example, consider a sample at 1MHz, and say that  $P$  is 20. The matrix will contain both  $2\pi \times 10^{6(120)}$  and 1.

The condition number, therefore, will, by construction, be large. This ill-conditioning is even worse when considering the normal form mentioned above because the condition number of  $A^T A$  is the square of the condition number of  $A$ .

There are preprocessing techniques, such as shifting and scaling, that improve the condition number, however this topic will not be discussed here. More details can be found in [18].

## 5.2 Pole-Residue Form

One way to avoid this problem is to keep the powers of  $\omega$  under control. This can be accomplished by decomposing Eq. (5-1) into its partial fraction form:

$$\sum_n \frac{c_n}{s - p_n} + d + se + \dots = \frac{a_0 + a_1s + a_2s^2 + \dots + a_Ps^P}{b_0 + b_1s + b_2s^2 + \dots + b_Qs^Q} = \frac{N(s)}{D(s)} \quad (5-7)$$

If  $P \leq Q + 1$  then the partial fraction form stops at the first power of  $s$ .

If the solution for the rational function can somehow be formulated so that  $c_n$ ,  $p_n$ ,  $d$ , and  $e$  are found instead of the  $a$  and  $b$  coefficients, then it may be possible to avoid the ill-conditioning problem that is due to the wide disparity in the magnitudes of the elements of the  $A$  matrix.

### 5.2.1 The Pole-Residue Form is Nonlinear

There is a new problem. The formulation becomes nonlinear because there are unknowns in the denominator.

What if the poles *were* known? The problem then becomes linear because only the residues,  $c_n$ ,  $d$ , and  $e$  are unknown. This is the subject of the next few sections.

## 5.3 The Theory of Vector Fitting

The theory of Vector Fitting [20] concerns itself with finding a rational approximation for a complex function,  $f(s)$ , using the partial fraction form of Eq. (5-7). The ill-conditioning problem is avoided by using this form, and the nonlinearity problem is avoided by assuming that the poles are known.

It must be noted that the goal of this fitting procedure is to approximate a transmission line model using rational functions. Transmission lines are physical systems, and hence any analytical representation must be causal and also must ultimately produce real-valued

results in a simulation. Real coefficients for the polynomials in  $N(s)$  and  $D(s)$  ensure these conditions.

Now consider again this form of the approximation:

$$f \approx \frac{N(s)}{D(s)} = \sum_{n=1}^N \frac{c_n}{s - a_n} + d + se \quad (5-8)$$

The residues,  $c_n$ , and poles,  $a_n$ , can be real or complex quantities, while  $d$  must  $e$  be real. If any poles or residues are complex, however, they must exist in complex conjugate pairs to ensure that the polynomials formed by algebraically combining all the terms have real coefficients.

It is worthwhile to repeat that the rational function in Eq. (5-8) is linear in the unknowns  $c_n$ ,  $d$ , and  $e$ , but nonlinear in  $a_n$ . As well, the approximation stops at the first order of  $s$  because passive electrical networks are composed of only first-order elements.

Note that when multiplied out, the approximation has the form

$$\sum_{n=1}^N \frac{c_n}{s - a_n} + d + se = \frac{\prod_{n=1}^N (s - z_n)}{\prod_{n=1}^N (s - a_n)}, \quad (5-9)$$

where the order of the numerator is one greater than the order of the denominator.

### 5.3.1 Initial Reasoning

We have already seen in the previous section that ill-conditioning becomes a problem for higher order fittings when the form of Eq. (5-1) is used. In other words, the system is solved using the form:

$$D(s)f(s) = N(s) \quad (5-10)$$

This can be avoided by using the form of Eq. (5-8). However, the problem is nonlinear because there is an unknown in the denominator of each term. Let's introduce, then a known denominator,  $\tilde{D}(s)$ , and pose the problem in an alternate form:

$$\frac{D(s)}{\tilde{D}(s)}f(s) = \frac{N(s)}{\tilde{D}(s)} \quad (5-11)$$

In other words, change the nonlinear problem into a linear one by assuming the poles are known, *and* use the pole-residue form of the rational functions,  $D(s)/\tilde{D}(s)$  and  $N(s)/\tilde{D}(s)$ , to avoid the ill-conditioning problem.

Note that finding the zeros of the rational function  $D(s)/\tilde{D}(s)$  will produce the actual poles of our original problem. Also, it is important to point out that if  $\tilde{D}(s)$  is chosen to be close to  $D(s)$ , then their ratio will approximately equal one, and the solution will probably present itself much sooner when using iteration.

Now we have a rational function with linear unknowns. The next step is to determine a procedure that solves for these unknowns.

## 5.4 Vector Fitting Methodology

The method of Vector Fitting consists of solving Eq. (5-8) in two stages. The first stage finds the zeros of  $D(s)$ , which are the poles of the rational approximation of  $f(s)$ . The second stage determines the residues, and this is equivalent to solving for  $N(s)$ .

### 5.4.1 Finding the Poles

Start by stating that  $f(s)$  is a rational function, such that  $f(s) = N(s)/D(s)$ , where the order of the numerator is one greater than the order of the denominator. Our task is to find suitable values for the coefficients of  $N(s)$  and  $D(s)$  so that this relation is as accurate as possible.

In actuality, we are working with samples of  $f(s)$ ,  $f(s_k)$ , and these are assumed to lie exactly on the rational function,  $N(s)/D(s)$ . Points between the samples are never considered.

Next, choose known poles,  $\tilde{a}_n$ , to form an approximation,  $f(s) \approx N(s)/\tilde{D}(s)$ , where

$$\tilde{D}(s) = \prod_n (s - \tilde{a}_n).$$

It is possible to add some unknown function,  $\varepsilon(s)$ , to  $f(s)$  such that the two sides are exactly equal:

$$\frac{N(s)}{\tilde{D}(s)} = f(s) + \varepsilon(s) \tag{5-12}$$

Factor out  $f(s)$  to obtain:

$$\frac{\tilde{N}(s)}{\tilde{D}(s)} = (1 + \sigma(s))f(s) , \text{ with } \sigma(s) = \frac{\varepsilon(s)}{f(s)} \quad (5-13)$$

Note that this is really the same thing as Eq. (5-11), but with a slightly different approach to get to the same place. The  $(1 + \sigma(s))$  factor is our  $D(s)/\tilde{D}(s)$  term from before, and finding the zeros of this function will give us the actual poles of  $f(s)$ .

Using Eqs. (5-11) and (5-13), it is possible to state that the function,  $\sigma$ , has a rational approximation with the same known poles:

$$\sigma(s) = \sum_n \frac{r_n}{(s - \tilde{a}_n)} \quad (5-14)$$

As well, the  $(1 + \sigma(s))$  term's numerator and denominator have the same order if the order of  $\tilde{D}(s)$  is chosen to be equal to the order of  $D(s)$ . This can be easily shown to be true.

Lastly, write out the full form of Eq. (5-13):

$$\left( \sum_n \frac{c_n}{(s - \tilde{a}_n)} + d + se \right) = \left( 1 + \sum_n \frac{r_n}{(s - \tilde{a}_n)} \right) f(s) \quad (5-15)$$

Thus:

$$\left( \sum_n \frac{c_n}{(s - \tilde{a}_n)} + d + se \right) - f(s) \sum_n \frac{r_n}{(s - \tilde{a}_n)} = f(s) \quad (5-16)$$

This is a linear equation with unknowns,  $c_n$ ,  $d$ ,  $e$ , and  $r_n$ . It is solved using an equation of the form  $Ax = b$ , where

$$\begin{aligned} \mathbf{x} &= [c_1 \dots c_N \ d \ e \ r_1 \dots r_N]^T \\ \mathbf{b} &= [f(s_1) \dots f(s_K)]^T \\ \mathbf{A} &= \text{stack}(\mathbf{A}_k) \\ \mathbf{A}_k &= \begin{bmatrix} \frac{1}{s_k - \tilde{a}_1} & \dots & \frac{1}{s_k - \tilde{a}_N} & 1 & s_k & \frac{-f(s_k)}{s_k - \tilde{a}_1} & \dots & \frac{-f(s_k)}{s_k - \tilde{a}_N} \end{bmatrix} \end{aligned} \quad (5-17)$$



and  $K$  is the total number of sampled frequencies.  $K$  must be at least as large as the width of  $A$  so that a solution is possible. In other words,  $K \geq 2N + 2$ .

Notice that formulating the solution to our original problem in this way has removed the ill-conditioning problem discussed in the previous chapter, and also preserved a linear formulation.

To summarize, finding  $r_n$  equates to finding the zeros of  $1 + \sigma(s)$ . These are also the zeros of  $D(s)$ . Sec. 5.7 discusses the actual zero-finding process. The  $c_n$  terms are not needed to find these zeros, but are found anyway because they are problem constraints.

This is the end of stage one.

### 5.4.2 Finding the Residues

Once the zeros of  $(1 + \sigma(s))$  are found, these are used as the new set of known poles,  $\tilde{a}_n$ , and then the equation

$$\sum_n \frac{c_n}{(s - \tilde{a}_n)} + d + se = f(s) \quad (5-18)$$

is solved in a similar manner to obtain  $c_n$ ,  $d$ , and  $e$ . This is again an overdetermined problem of the form  $Ax = b$ .

### 5.4.3 Iteration

These equations are usually overconstrained because there are a very large number of samples. Therefore, the solutions are likely not exact. They will be a “best fit” in the least-squares sense. However, the new poles are probably better than the original guessed poles.

Practically, then, the new poles can be used as the starting poles for an additional iteration. This becomes an iterative procedure with the calculated poles becoming the starting poles for each step.

Hopefully, this will converge to some “best” set of poles.

## 5.5 Summary So Far

By incorporating a known denominator,  $\tilde{D}(s)$ , into the  $D(s)f(s) = N(s)$  formulation, it becomes possible, in a straightforward way, to formulate a solution using only unity powers of  $s$ , and therefore  $\omega$ .

Second, the problem of finding zeros and poles is reduced to the simpler problem of only needing to find the zeros of a function representing the relative error in the solution.

Third, iteration can be used to refine the poles.

The fourth point is that the roots of  $\tilde{D}(s)$  are known and chosen such that the order is the same as the order of  $D(s)$ . Now, it may be the case that the order of  $D(s)$  is unknown, in which case a guess will have to be made. This is discussed in more detail in Ch. VIII.

## 5.6 Solving with Complex Values

It may be the case that there are complex poles and residues in the solution. Appendix C discusses some facts regarding these, such as the fact that residues come in perfect conjugate pairs if their corresponding poles are complex conjugates. The reader is referred there for more details and a proof of this fact.

A computer program can only work with one-dimensional, real-valued numbers. It is necessary, then, to overcome this limitation and somehow represent complex numbers using only real values.

In order to solve this system using only real values, the complex numbers are separated into real and imaginary parts as follows.

Assume that the partial fractions at locations  $i$  and  $i + 1$  in a given row of  $A$  constitute a complex conjugate pair:

$$\begin{aligned} a_i &= a' + ja'' & a_{i+1} &= a' - ja'' \\ c_i &= c' + jc'' & c_{i+1} &= c' - jc'' \end{aligned} \quad (5-19)$$

The two corresponding elements in  $A_{k,i}$  and  $A_{k,i+1}$  are modified to become:

$$A_{k,i} = \frac{1}{s_k - a_i} + \frac{1}{s_k - \bar{a}_i} \quad A_{k,i+1} = \frac{j}{s_k - a_i} - \frac{j}{s_k - \bar{a}_i} \quad (5-20)$$

The corresponding residues in the solution vector  $\mathbf{x}$ , therefore, become  $c'$  and  $c''$ .

Formulating Eq. (5-17), then, in terms of only real quantities produces the formulation:

$$\begin{bmatrix} A' \\ A'' \end{bmatrix} \mathbf{x} = \begin{bmatrix} b' \\ b'' \end{bmatrix} \quad (5-21)$$

Formulating the problem in this way also has the effect of enforcing the conjugacy property. The benefit of this is that there will be perfect complex conjugate pairs that are unaffected by numerical round-off.

### 5.7 Calculating the Zeros of $D(s)$

Once the residues,  $r_n$ , of  $\sigma(s)$  are known, it is possible to calculate the zeros of  $(1 + \sigma(s))$  by finding the eigenvalues of the matrix

$$\mathbf{H} = \mathbf{A} - \mathbf{b}\mathbf{r}^T, \quad (5-22)$$

where

$$\mathbf{A} = \begin{bmatrix} \tilde{a}_1 & 0 & \dots & 0 \\ 0 & \tilde{a}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \tilde{a}_N \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \mathbf{r}^T = [r_1 \ r_2 \ \dots \ r_N] \quad (5-23)$$

The matrix  $\mathbf{r}^T$  is a row vector containing the residues of  $\sigma$ ,  $\mathbf{A}$  is a matrix containing the starting poles along the diagonal, and  $\mathbf{b}$  is a column vector of ones.

The discussion in Appendix D shows why this is true.

As a side point, the more “obvious” way of finding the zeros is to multiply all the terms out, and then use some root finding technique. However, this would require more calculations. In fact, eigenvalues are often used to find polynomial roots, so multiplying all the terms out would create lots more unnecessary work since we are finding eigenvalues of a simpler matrix anyway.

It is desirable to again use only real quantities, as any complex eigenvalues will come out as perfect conjugate pairs. The modification necessary to do this is as follows.

Each complex pole,  $a_i$ , in  $\mathbf{A}$  is replaced with the square matrix

$$\begin{bmatrix} a' & a'' \\ -a'' & a' \end{bmatrix}, \quad (5-24)$$

which is just the matrix form of a complex number, and each corresponding element in  $\mathbf{b}$  and in  $\mathbf{r}$  is replaced with

$$\hat{b} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \hat{r} = \begin{bmatrix} r' & r'' \end{bmatrix} \quad (5-25)$$

respectively, where  $a_i = a' + ja''$ , and  $r_i = r' + jr''$ .

Please refer to Appendix D for the derivation of these replacements.

## 5.8 Weighting

It is possible to cause the fitting to be more accurate around a specified set of frequencies. This is accomplished by premultiplying both sides of the  $Ax = b$  equation by a weighting matrix,  $W$ :

$$WAx = Wb \quad (5-26)$$

$W$  is defined as a diagonal matrix with weighting factors positioned at the desired frequencies. For example, if we wish to fit the curve more accurately at 60Hz, and the 60Hz sample is in row three of a set of four samples, then the weighting matrix might look like

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5-27)$$

where  $w$  is the desired weighting factor, a number greater than one.

Practically, the weighting is applied by multiplying the desired rows of  $A$  and  $b$  by the corresponding weighting factor, and leaving the rest unaffected.

## 5.9 The “Vector” in Vector Fitting

The “Vector” in Vector Fitting refers to the fact that this algorithm can be expanded to include a set, or *vector*, of functions having the same set of known poles. This is accomplished by simply stacking Eq. (5-16) as follows. The formulation shown is for a vector of two functions, but the technique can be expanded to an arbitrary number.

$$\begin{bmatrix} \left( \sum_n \frac{c_n^{(1)}}{(s-\tilde{a}_n)} + d^{(1)} + se^{(1)} \right) \\ \left( \sum_n \frac{c_n^{(2)}}{(s-\tilde{a}_n)} + d^{(2)} + se^{(2)} \right) \end{bmatrix} - \begin{bmatrix} f_1 \sum_n \frac{r_n}{(s-\tilde{a}_n)} \\ f_2 \sum_n \frac{r_n}{(s-\tilde{a}_n)} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (5-28)$$

Note that the superscripts denote the curve number, and are not exponents.

The known poles are shared, as is the  $\sigma(s)$  function. As before, the solution for the unknown residues is of the form  $Ax = b$ , but each matrix is modified to accommodate the extra set of variables:

$$\begin{aligned} \mathbf{x} &= \left[ c^1_1 \dots c^1_N d^1 e^1 c^2_1 \dots c^2_N d^2 e^2 r_1 \dots r_N \right]^T \\ \mathbf{b} &= \text{stack}(\mathbf{b}_k) \\ \mathbf{b}_k &= \begin{bmatrix} f_1(s_k) \\ f_2(s_k) \end{bmatrix} \\ \mathbf{A} &= \text{stack}(\mathbf{A}_k) \end{aligned} \quad (5-29)$$

$$\mathbf{A}_k = \begin{bmatrix} \frac{1}{s_k - \tilde{a}_1} & \dots & \frac{1}{s_k - \tilde{a}_N} & 1 & s_k & 0 & 0 & 0 & 0 & 0 & \frac{-f_1(s_k)}{s_k - \tilde{a}_1} & \dots & \frac{-f_1(s_k)}{s_k - \tilde{a}_N} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{s_k - \tilde{a}_1} & \dots & \frac{1}{s_k - \tilde{a}_N} & 1 & s_k & \frac{-f_2(s_k)}{s_k - \tilde{a}_1} & \dots & \frac{-f_2(s_k)}{s_k - \tilde{a}_N} \end{bmatrix}$$

Once the new poles are found from the zeros of  $(1 + \sigma(s))$ , each curve is fitted separately to find the residues.

## 5.10 Future Directions

There are many questions about the Vector Fitting algorithm that remain unanswered. Some questions to be posed include:

- Under what conditions does the algorithm remain stable, and how do these conditions affect convergence?
- How can we improve fitting accuracy?

### 5.10.1 Regularization

An analysis of the singular values of the  $\mathbf{A}$  matrix of Eq. (5-17) indicates that curves with more peaks and valleys are better conditioned. In other words, smoother curves have

smaller singular values. Please see Section 6 of [20] for further details on the subject of singular values and Vector Fitting.

It may be possible to “penalize” smoother curves and to “reward” curves with more peaks and valleys. A potential procedure, in addition to the “weighting” mentioned previously, would be to premultiply the curve by its second derivative or by some function times its second derivative:

$$f_{new}(j\omega) = f(j\omega) \frac{d^2}{d\omega^2} f(j\omega) \quad (5-30)$$

This is called “regularization”.

### 5.10.2 DC value

It is possible to fix the value of the fitted curve at  $s = 0$  to a specified value by modifying the algorithm as follows. [The basic idea is from a discussion with Dr. A. Gole of the University of Manitoba.]

First, refactor each term in the summation:

$$\frac{c_n}{s - a_n} = \frac{c_n' a_n}{s - a_n} + c_n' - c_n' = \frac{s c_n'}{s - a_n} - c_n' \quad (5-31)$$

We now have:

$$\sum_n \frac{c_n}{(s - \tilde{a}_n)} + d + se = \sum_n \frac{s c_n'}{(s - \tilde{a}_n)} + \left( - \sum_n c_n' + d \right) + se = s \sum_n \frac{c_n'}{(s - \tilde{a}_n)} + d' + se \quad (5-32)$$

The effect of this modification is that at  $s = 0$ , the value is fixed at  $d'$ , and this can be set to any value. Of course, appropriate modifications would have to be made to the  $\mathbf{A}$  matrix of Eq. (5-17). As well, since  $d'$  is known, this can be subtracted from  $\mathbf{b}$  in the same equation.

This discussion is in this section because it was not implemented in this project.

This modification may or may not contribute to the accuracy of the results. It has yet to be explored.

### 5.10.3 Poles at Zero

Various numerical problems occur when trying to model a pole at zero frequency because of close-to-division-by-zero errors. Perhaps some combination with the formulation of Eq. (5-32) will yield improvements.

### 5.10.4 Convergence Properties

Work needs to be done to determine if the Vector Fitting algorithm can converge to a global minimum, or even to a local minimum. Perhaps the algorithm does converge, but to a region, and not to a specific point. It is already known that the fitting algorithm converges to a different solution, depending on the starting poles.

In fact, under what conditions does the algorithm converge?

Perhaps a different overdetermined fitting strategy, something other than the least-squares approach would improve the convergence. A procedure such as nonlinear optimization would be a good area to explore.

Also, it needs to be determined how many iterations need to be performed, and what kind of stopping procedure applies.

## 5.11 Poles of Order Greater Than One, a Future Direction

The Vector Fitting approach does not detect poles of order greater than one. Its construction enforces this. This section proposes a way to detect the existence of these poles in a post-processing step.

After a curve is fitted, some of the poles may be very close, and their corresponding residues will be very large. This is probably due to poles of order two or greater. The technique presented here removes these very large numbers from the approximation; this is one of its major advantages.

### 5.11.1 Detection

Let's start with a pole of order two to illustrate the issues. A pole of order two cannot be expressed as:

$$\frac{A}{(s-a)^2} = \frac{B}{s-a} + \frac{C}{s-a} \quad (5-33)$$

To see why, we must first examine the expression:

$$\frac{A}{(s-a)(s-b)} = \frac{\left(\frac{A}{a-b}\right)}{s-a} + \frac{\left(\frac{A}{b-a}\right)}{s-b} \quad (5-34)$$

When  $a$  and  $b$  are very close, where  $b = a + \delta$  for a small  $\delta$ , this becomes:

$$\frac{A}{(s-a)(s-(a+\delta))} = \frac{-\left(\frac{A}{\delta}\right)}{s-a} + \frac{\left(\frac{A}{\delta}\right)}{s-(a+\delta)} \quad (5-35)$$

It can be seen that as  $a$  and  $b$  get closer together, the residues dramatically increase in magnitude.

In general, for  $N$  distinct poles,

$$\frac{A}{\prod_{n=1}^N (s-a_n)} = \sum_{n=1}^N \frac{c_n}{s-a_n}, \quad (5-36)$$

where

$$c_n = \frac{A}{\prod_{j \neq n} (a_n - a_j)} \quad (5-37)$$

Now, if it can be shown, given a set of  $N$  poles, that  $|a_i - a_j| < \delta$  for every  $i \neq j$ , and if the residues,  $c_n$ , are “large”, then there is a good chance that we have encountered a pole of order  $N$ . The next question then becomes how to recover the residue,  $A$ .

Again, let’s examine the  $N = 2$  case, and then progress to higher orders. Eq. (5-35) implies that each term will have similar residues, but with opposite sign. It is stated that they are “similar” instead of “the same” because round-off error may exist in the calculations.

The steps, then, are as follows for discovering a pole of order two.

1. Choose a suitable “closeness” threshold,  $\delta$ , for determining if poles are the same.
2. Find pairs of poles,  $a_i$  and  $a_j$ , such that  $|a_i - a_j| < \delta$ .



3. Multiply the associated residues,  $c_i$  and  $c_j$ , by  $(a_i - a_j)$  and  $(a_j - a_i)$ , respectively, to form  $A_i = c_i(a_i - a_j)$  and  $A_j = c_j(a_j - a_i)$ . Eqs. (5-37) and (5-34) should illustrate why these multipliers were chosen. They theoretically allow the residue,  $A$ , to be recovered.
4. If these values are the same to within some tolerance, then we have found a pole of order two.

For higher-order poles, the steps are as follows.

1. Choose a suitable “closeness” threshold,  $\delta$ .
2. Find a set of  $N$  poles,  $\{a_n\}$ , such that  $\frac{1}{N} \sum_{i \neq j} |a_i - a_j| < \delta$ . Other criteria could be used, such as  $\sqrt[N]{\prod_{i \neq j} |a_i - a_j|} < \delta$ , but the summation probably requires less computational effort and doesn't need to concern itself with any zeros sneaking in.
3. Multiply the associated residues,  $c_n$ , by  $\prod_{i \neq n} (a_n - a_i)$  to form values for each  $A_n$ .
4. If all these values are the same to within some tolerance, then we have found a pole of order  $N$ .

The higher the order, the larger the residues will be.

In practice, the check in the last step does not need to be performed. The reason for this can be seen in the example for two close poles in Eq. (5-34).

### 5.11.2 Implementing This Technique

Recall that when detecting a pole of order  $N$ , sets of  $N$  values must be selected from the entire set of poles. There are good algorithms for selecting unique subsets of a given set, however, it will be the case that each pair of poles selected for comparison will be chosen multiple times. It is worthwhile to investigate how this can be done in a more efficient manner.

Because the total practical number of poles chosen to fit a curve is relatively small — up to hundreds, not thousands or more — the pair comparisons can first be stored before the subset selection procedure. There are  $(N(N - 1))/2$  pairs, and if the result of each pair

comparison to the given tolerance is stored in one bit, then the storage requirements are only  $(N(N-1))/2$  bits, or  $(N(N-1))/16$  bytes. The representation doesn't matter, but in the implementation presented here, the bit is set if the pair is deemed to be "close".

For, say, 200 poles, the storage requirement is only about 2.4 kilobytes. This is a miniscule amount of memory to use in today's systems. Now, if the number of points was thousands or more, then the storage requirements would start to be in the megabyte range.

When searching through each subset, the pair comparison flags are consulted, and if all corresponding flags are set, then these poles are marked as "used" in another array of flags. The residue is then calculated.

For best results, the search order for poles of order  $N$  and below should start at  $N$  because it is possible for poles to be missed if the search is started at two. For example, say there are five poles, and that 3 are "close". If the search starts at two, then two out of the three "close" poles will be marked as "used", and the third of this group will be on its own.

This limitation can be overcome if the search is allowed to analyze poles already marked as "used", but this would require extra bookkeeping, and the same results would appear if the search were done backwards, as originally suggested.

Its ideas weren't used here, but the curious may wish to read [22] for some discussion on searches of this type.

### 5.11.3 Retrofitting Vector Fitting to Detect Higher-Order Poles

The Vector Fitting technique can be modified to detect higher-order poles. This is useful for the more general system identification problem [23].

Simply replace all occurrences of terms of the form  $\frac{x}{s-a}$  with  $\frac{x}{(s-a)^q}$ . The procedure must keep track of only one extra piece of information per pole, and that is the pole order.

The other modification to the algorithm involves the zero finding procedure. Instead of finding the zeros of

$$\sigma(s) = \sum_n \frac{r_n}{(s - \tilde{a}_n)} + 1, \quad (5-38)$$

the zeros of

$$\sigma(s) = \sum_n \frac{r_n}{(s - \tilde{a}_n)^{q_n}} + 1 \quad (5-39)$$

must be found instead.

## 5.12 Chapter Summary

This chapter showed how to use a technique called *Vector Fitting* to find the poles and residues for a rational approximation to some curve. It is necessary to start the procedure with some known set of poles, but a method for finding this initial set will be discussed in Ch. VIII.

## CHAPTER VI

### VECTOR FITTING RESULTS

This chapter shows selected vector fitting results for several example polynomials. The polynomials are the same as the example polynomials in [20] so that we can verify our results.

All fittings use four iterations.

#### 6.1 Fitting with Complex Poles

The first example polynomial has the poles and residues of Table 6-1, and constant and linear terms,  $d = 0.2$  and  $e = 2 \times 10^{-5}$ .

**Table 6-1:** Poles and Residues of the First Example Polynomial, in Hz

Poles	Residues	Found Poles	Found Residues	Found $d$ and $e$
-4500	-3000	-4500	-3000	0.2
-41000	-83000	-41000	-83000	$2 \times 10^{-5}$
$-100 \pm j5000$	$-5 \pm j7000$	$-100 \pm j5000$	$-5 \pm j7000$	
$-120 \pm j15000$	$-20 \pm j18000$	$-120 \pm j15000$	$-20 \pm j18000$	
$-3000 \pm j35000$	$6000 \pm j45000$	$-3000 \pm j35000$	$6000 \pm j45000$	
$-200 \pm j45000$	$40 \pm j60000$	$-200 \pm j45000$	$40 \pm j60000$	
$-1500 \pm j45000$	$90 \pm j10000$	$-1500 \pm j45000$	$90 \pm j10000$	
$-500 \pm j70000$	$50000 \pm j80000$	$-500 \pm j70000$	$50000 \pm j80000$	
$-1000 \pm j73000$	$1000 \pm j45000$	$-1000 \pm j73000$	$1000 \pm j45000$	
$-2000 \pm j90000$	$-5000 \pm j92000$	$-2000 \pm j90000$	$-5000 \pm j92000$	

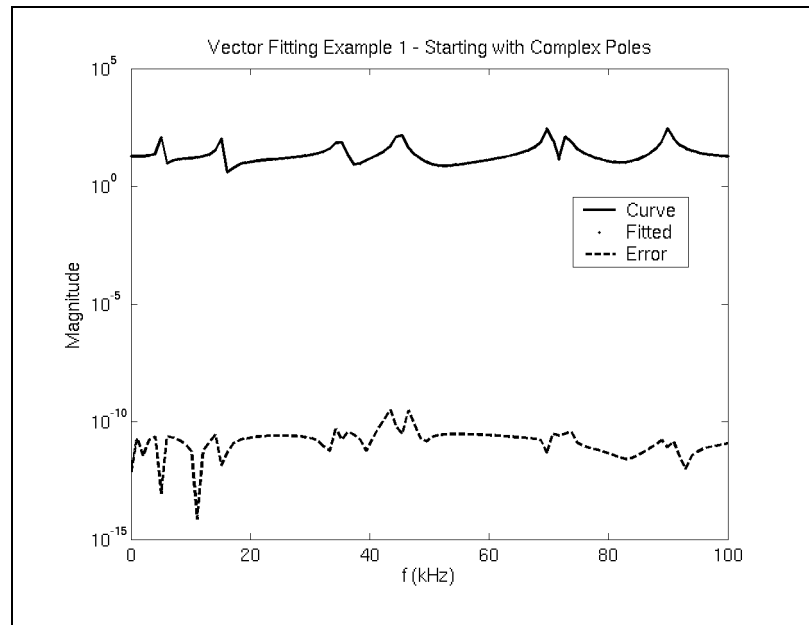
For the first example, 20 complex starting poles were distributed linearly over the frequency range. The poles and residues were found to be very close to the actual values (to within about 8 or 9 significant figures), and Table 6-2 shows the extra two poles and residues. Fig. 6-1 shows a plot of the fitted curve.

Notice that the residue-pole ratios for the extra two poles are very small, and so these poles could be deleted.

**Table 6-2:** Extra Poles and Residues, in Hz

Poles	Residues
618084.66	-0.00039214
-155946.38	-0.00015329

The RMS error after the four iterations was  $2.760 \times 10^{-9}$  and the error after fitting the residues was  $2.904 \times 10^{-9}$ . The maximum deviation was  $4.197 \times 10^{-9}\%$ .



**Fig. 6-1.** Starting with complex poles.

## 6.2 Fitting with Real Poles

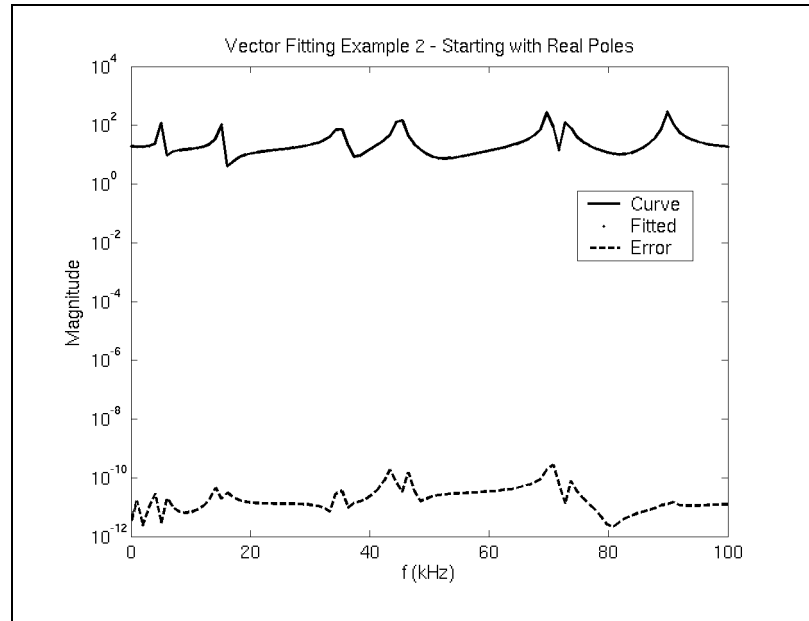
For the second example, 20 real seed poles were distributed over the frequency range. Again, the poles and residues were found to be very close to the actual values, also to within about 8 or 9 significant figures. Table 6-3 shows the extra two poles and residues, and Fig. 6-2 shows a plot of the fitted curve.

The residue-pole ratios for the extra poles are still very small, so these poles could be deleted.

**Table 6-3:** Extra Poles and Residues, in Hz

Poles	Residues
-7489068.4	-2.5053508
-475349.92	-5.0865713

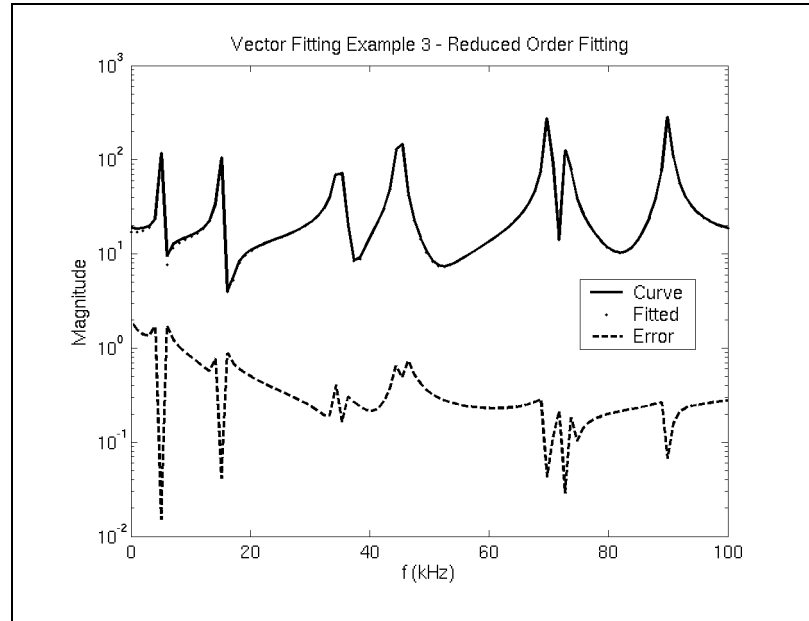
The RMS error started at 22.44 , but quickly reduced to  $2.332 \times 10^{-10}$  after the four iterations. The error after fitting the residues was  $2.685 \times 10^{-10}$  , and the maximum deviation was  $3.659 \times 10^{-10} \%$  .



**Fig. 6-2.** Starting with real poles.

### 6.3 Fitting with a reduced number of poles

The third example distributed 14 seed poles over the frequency range. The idea is to see how a reduced order fitting affects the results. Fig. 6-3 shows a plot of the fitted curve.



**Fig. 6-3.** Starting with a reduced number of poles.

The RMS error was 5.503 after the four iterations and slightly higher after fitting the residues (but the same to 9 significant figures). The maximum deviation was 10.15%.

The fitted poles and residues are shown in Table 6-4.

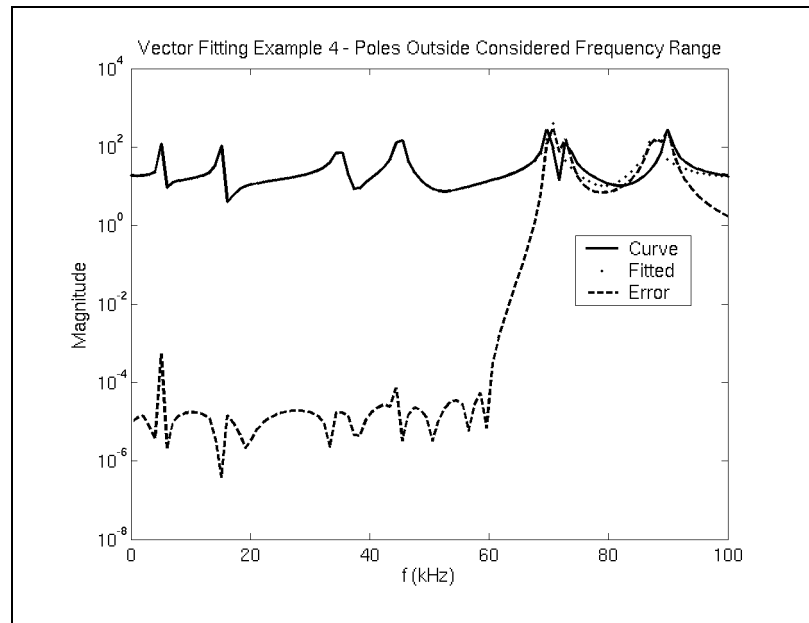
**Table 6-4:** Found Poles for the Reduced Order Fitting Case, in Hz

Found Poles	Found Residues	Found $d$ and $e$
$-1998.1753 \pm j90000.637$	$-3622.4809 \pm j91917.680$	$-0.01594223$
$-998.10169 \pm j73001.576$	$2124.5675 \pm j45111.089$	$2.0557355 \times 10^{-5}$
$-497.43415 \pm j70000.714$	$50908.136 \pm j80042.918$	
$-361.28698 \pm j44998.569$	$1054.6380 \pm j69091.525$	
$-3039.7022 \pm j35001.358$	$7477.3001 \pm j45481.644$	
$-87.088291 \pm j4989.7521$	$-4639.8368 \pm j8170.8925$	
$-138.24888 \pm j14994.700$	$973.17813 \pm j18652.370$	

## 6.4 Poles Outside the Considered Frequency Range

For the next example, 16 seed poles were distributed from 1 to 60kHz, so there are poles that exist in the curve that are outside the fitted frequency range. The fitting appears very accurate in the considered interval, and inaccurate outside of this interval.

Fig. 6-4 shows a plot of the fitted curve.



**Fig. 6-4.** There are poles outside the considered frequency range.

The RMS error for the entire frequency range was  $1.854 \times 10^{-4}$  after the four iterations, and slightly higher after fitting the residues (but the same to 5 significant figures). The maximum deviation over the whole frequency range was 340.0%, but only  $4.615 \times 10^{-4}\%$  up to 60kHz.



The fitted poles and residues are shown in Table 6-5.

**Table 6-5:** Found Poles for the Extra Range Case, in Hz

Found Poles	Found Residues	Found $d$ and $e$
$-3659.6956 \pm j87398.424$	$-8189.2113 \pm j101418.77$	0.16535286
$-840.46423 \pm j70454.255$	$48445.525 \pm j112999.76$	$2.0021016 \times 10^{-5}$
$-1500.0504 \pm j45000.009$	$87.100093 \pm j9999.6112$	
$-200.00190 \pm j45000.001$	$43.030187 \pm j60000.408$	
$-3000.0014 \pm j35000.000$	$5999.9013 \pm j45000.023$	
$-120.00035 \pm j15000.000$	$-19.959276 \pm j17999.995$	
-40997.108	-82991.772	
$-100.00078 \pm j4999.9999$	$-5.0415475 \pm j6999.9961$	
-4499.6471	-2999.5898	

## 6.5 Fitting a Smooth Curve Using Real Poles

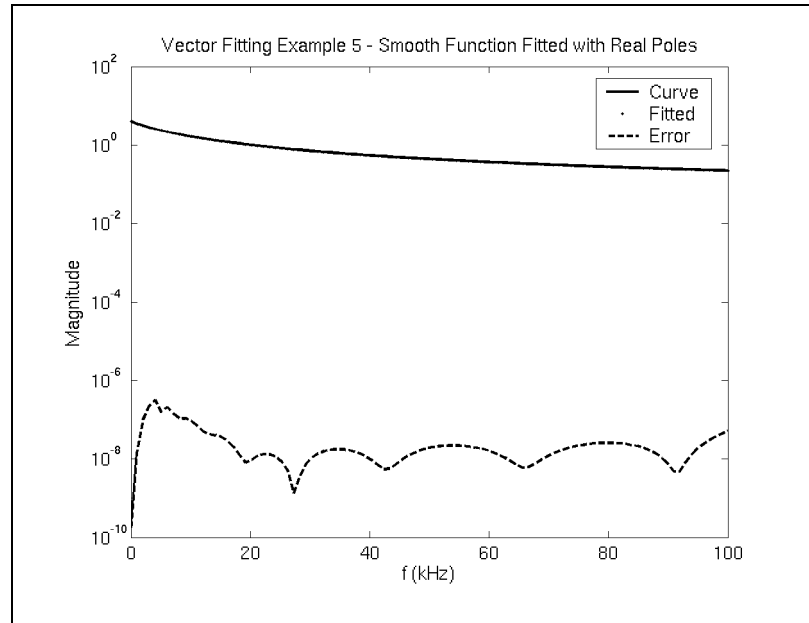
The final two examples fit a curve having the poles and residues of Table 6-6.

**Table 6-6:** Poles and Residues of the Second Example Polynomial, in Hz

Poles	Residues	Found Poles	Found Residues
-2000	1000	-1858.2859	759.69950
-4000	-1000	$-5369.0537 \pm j578.37922$	$254.42874 \pm j13.214402$
-9000	7000	-11995.956	14562.277
-15000	12000	$-38789.745 \pm j629.22129$	$25292.379 \pm j736.59872$
-18000	5000	$-62364.938 \pm j1364.1943$	$10959.376 \mp j4948.7231$
-21000	-12000	-207962.73	-1197.7288
-23000	-2000	-269651.74	10645.706
-29500	1500	$-328258.41 \pm j938.96894$	$-11357.045 \pm j5523.4888$
-33000	31000	-385585.65	31780.508
-34000	-12000	-490556.07	-47578.445
-44000	20000	-537855.72	-32135.859
-48000	41000	539212.01	0.07337512
-56000	8000	$-598770.51 \pm j1321.2537$	$24133.899 \mp j50963.349$
-64000	15600	-711936.14	12965.130
-72000	-10000		
-79000	-12000		
-88000	50000		
-93000	-2000		

Note that the pole-residue ratio for the unstable pole is very small.

Fig. 6-5 shows the results of the fitting using 20 real poles having their magnitudes distributed over the frequency range.



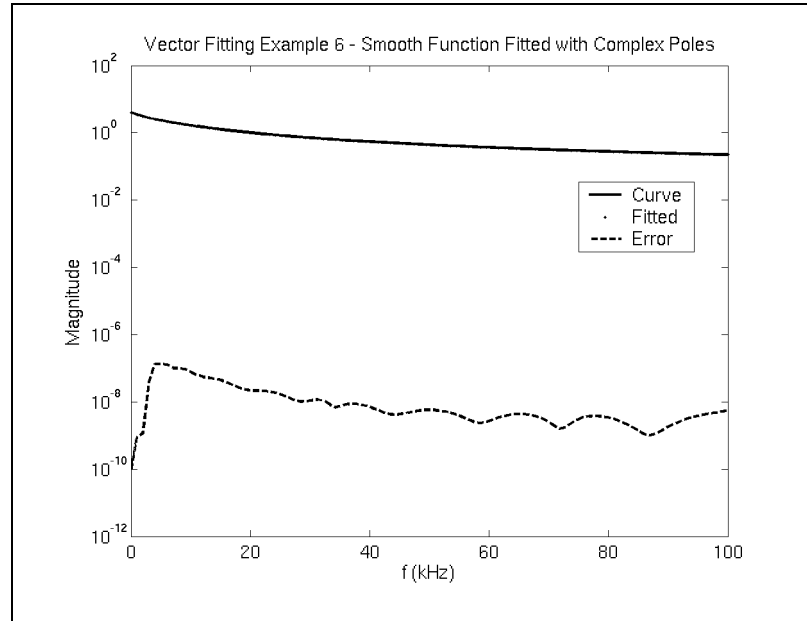
**Fig. 6-5.** Fitting a smooth curve with real poles.

The RMS error after the four fitting iterations was  $1.117 \times 10^{-6}$ , and  $5.479 \times 10^{-7}$  after fitting the residues. The maximum deviation was  $1.258 \times 10^{-5}\%$ .

The poles that were found included several complex poles, which indicates that the algorithm can converge to complex poles even if there are only real starting guesses.

## 6.6 Fitting a Smooth Curve Using Complex Poles

The smooth curve from the previous example was fitted using 20 complex poles distributed over the frequency range. Fig. 6-6 shows the fitting results.



**Fig. 6-6.** Fitting a smooth curve with complex poles.

The RMS error after the four fitting iterations was  $1.078 \times 10^{-6}$ , and  $3.331 \times 10^{-7}$  after fitting the residues. The maximum deviation was  $5.682 \times 10^{-6}\%$ .

Complex poles as well as real poles appeared in the fitted curve. The fitted poles and residues are shown in Table 6-7.

**Table 6-7:** Found Poles and Residues of the Second Example Polynomial, in Hz

Found Poles	Found Residues
-1907.5691	825.11130
-5812.7209±j356.83845	643.49731±j98.478219
-12338.902	13938.162
16774.903±j33120.179	-0.692×10 <sup>-4</sup> ∓j0.384×10 <sup>-5</sup>
-37196.560±j2295.2429	-1169.5119∓j415.29586
39762.019±j74056.099	0.00012431±j0.2088×10 <sup>-4</sup>
-46331.208	89677.495
-77940.454±j31465.442	-0.02692048∓j0.00500485
-88996.935	37710.276
-89760.346±j58891.207	-0.00185696±j0.00042485
-118535.43±j23619.340	0.18285902±j0.05227653
189820.83±j57832.415	-0.00163025±j0.000135

We can see again that the residue-pole ratios for the unstable poles are very small.

## 6.7 Unstable Poles

As we have seen in some of the fitting examples, unstable poles – poles with a positive real part – can appear when fitting a curve using the Vector Fitting algorithm. This situation must be dealt with, especially when modelling what is supposed to be a passive electrical network.

There are several approaches to handling these. First, the unstable poles could be deleted after each iteration. This would reduce the fitting order, but no unstable poles would appear at the end.

The other approach would be to flip each unstable pole into the left-half plane after each iteration. This would also result in a stable fitting.

Additionally, either technique could be applied after all the iterations, instead of after each iteration. Further study needs to be done on this subject.

One other point that deserves mention is that just because we have a stable system, there is no guarantee that the result will be *passive*. Please refer to [24], [25], [26], and [27] for more details on this subject. The next chapter will also address how to ensure passivity.

## 6.8 Sensitivity to Eigenvalues

Recall that eigenvalues are used as the zero-finding procedure to determine the unknown poles for the next iteration. The eigenvalue finding routine used for this project exposed the fact that Vector Fitting can converge to a local minimum.

Using the code developed for this project, it is possible to get similar but varying results each time the fitting is performed. The only source of randomness in the entire sequence of events is the eigenvalue finding routine. The randomness is necessary because it is used to break up any cycles that appear while iterating. (In actuality, it's the Schur decomposition routine, but this is used to find the eigenvalues.)

Now, the eigenvalues that are found each time are correct to within machine tolerance, though there is some variation below this threshold. This indicates that the Vector Fitting algorithm is very sensitive to the zero-finding procedure used to find the next set of poles, and may cause the fitting to converge to a different “local minimum”. This is expressed in quotes because this project has not shown that the Vector Fitting algorithm converges at all to a specific point, let alone a local or global minimum.

## 6.9 Chapter Summary

This chapter showed some Vector Fitting results for several example polynomials. The fitting always seems to converge to an accurate approximation. If there are lots peaks and

valleys, then the fitting seems accurate over the frequency range under consideration. If the curve is smooth, then the fitting is still accurate, although the fitted results may not exactly match the actual poles.

It was also possible for unstable poles to appear in the approximation, however, the corresponding residues were always very small, so these poles could probably be safely deleted.

As well, depending on the location of the starting poles, the Vector Fitting algorithm can converge to different local minima.

## CHAPTER VII

# ENFORCING PASSIVITY

Once a reasonable solution is found to the rational function fitting problem, there is one other thing to consider. The resulting function may not represent a passive system. In other words, the system may emit power for some set of applied voltages at some frequencies, and this is not desirable. Said another way, stability does not imply passivity.

The discussion here is based on the work in [24], [25], and [26]. Another technique is presented by Saraswat, Achar, and Nakhla [27], but will not be discussed here.

First, we must define what it means, mathematically, for a function to be passive. Take some component defined by an admittance matrix,  $\mathbf{Y}$ , in the frequency domain:

$$\mathbf{i} = \mathbf{Y}\mathbf{v} \tag{7-1}$$

In the case of an overhead transmission line,  $\mathbf{Y}$  is symmetric. It is also positive real, and a good definition of this and how it relates to the concept of a passive system is found in [28], [29], and [30].

For any applied voltage,  $\mathbf{v}$ , the absorbed power is given by:

$$\mathbf{P} = \text{Re}\{\mathbf{v}^*\mathbf{Y}\mathbf{v}\} = \text{Re}\{\mathbf{v}^*(\mathbf{G} + j\mathbf{B})\mathbf{v}\} = \text{Re}\{\mathbf{v}^*\mathbf{G}\mathbf{v}\} \tag{7-2}$$

Now,  $\mathbf{G}$ , a real matrix, is symmetric for overhead transmission lines. Therefore,  $\mathbf{G} = \text{Re}\{\mathbf{Y}\}$  is positive definite. In other words, all of its eigenvalues must be real.

Note that we didn't start with the usual definition for complex power, where  $\mathbf{S} = \mathbf{v}\mathbf{i}^*$ . This is because  $\text{Re}\{\mathbf{v}\mathbf{i}^*\} = \text{Re}\{\mathbf{v}^*\mathbf{i}\}$ , so it doesn't matter which one we use.

The question now is once we have a rational approximation to a function,  $Y$ , how is it possible to force all the eigenvalues to be real?

### 7.1 Approach Based on Linearization and Quadratic Programming

Section 3 of [24] describes a way to solve this problem by using a technique called linearization. Briefly, it defines an incremental relationship between the new value of  $Y$  and its parameters,  $x$ :

$$\Delta Y = M\Delta x \tag{7-3}$$

The paper then goes on to formulate a way to minimize  $\Delta x$  using a technique called Quadratic Programming. The reader is referred to this paper for more details.

## 7.2 Approach based on Passivity Compensation

Achar et al. [27] propose a method to compensate only those regions having a negative eigenvalue. It avoids having to sweep over the entire frequency range.

## 7.3 Negative Eigenvalue Deletion

Another, simpler, way to force passivity is to simply delete the negative eigenvalues altogether. This is the technique implemented for this project.

The following procedure must be applied for *each frequency* of the curve. Start with the fitted admittance matrix,  $Y$ , and take the real part of each element.

$$G_{i,j}(s) = \text{Re}\{Y_{i,j}(s)\} = \text{Re}\left\{\sum_{n=1}^N \frac{c_n}{s-a_n} + d + se\right\} = \text{Re}\left\{\sum_{n=1}^N \frac{c_n}{s-a_n}\right\} + d \quad (7-4)$$

Remember that this is a causal system, so the  $d$  term must be real.

The full matrix is therefore:

$$\mathbf{G}(s) = \mathbf{P}(s) + \mathbf{D} \quad (7-5)$$

Next, for each frequency of interest, diagonalize  $\mathbf{G}$  and note that the eigenvalues can be split into positive and negative elements.

$$\mathbf{T}(\Lambda_{pos} + \Lambda_{neg})\mathbf{T}^{-1} = \mathbf{P} + \mathbf{D} \quad (7-6)$$

For example, the matrix  $\Lambda = \begin{bmatrix} 2 & 0 \\ 0 & -3 \end{bmatrix}$  can be split into a positive  $\Lambda_{pos} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$  and a negative  $\Lambda_{neg} = \begin{bmatrix} 0 & 0 \\ 0 & -3 \end{bmatrix}$ .

Last, form a new, corrected, conductance matrix,  $\mathbf{G}_{corr}$ :

$$\mathbf{G}_{corr} = \mathbf{T}\Lambda_{pos}\mathbf{T}^{-1} = \mathbf{P} + \mathbf{D} - \mathbf{T}\Lambda_{neg}\mathbf{T}^{-1} \quad (7-7)$$

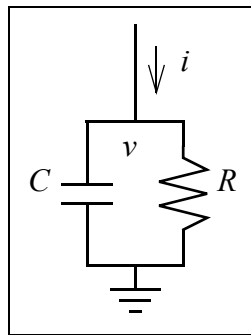
Thus, for each frequency for which we require passivity, a correction is added to  $\mathbf{D}$ .

## 7.4 Enforcing the Positive Definiteness of $\mathbf{E}$

It is also useful to enforce the positive definiteness of  $\mathbf{E}$ . While it does not contribute to the passivity enforcement of  $\mathbf{G}$ , it still helps as far as stability is concerned.

For example, consider a system with a single capacitor having a negative capacitance. Connecting a resistor across its terminals will result in a system with a positive eigenvalue.

To illustrate:



**Fig. 7-1.** Circuit with negative capacitance.

The solution for the voltage of Fig. 7-1 is

$$\begin{aligned}
 Cv' + \frac{v}{R} &= i \\
 v' &= -\frac{v}{RC} + \frac{i}{C} \quad , \\
 \therefore v &= Ke^{\left(-\frac{1}{RC}\right)t} + \frac{1}{C} \int e^{-\frac{1}{RC}(t-\tau)} i d\tau
 \end{aligned} \tag{7-8}$$

where  $K$  is a constant of integration. This is clearly an unstable system when  $C$  is negative.

## 7.5 Chapter Summary

This section discussed three ways to enforce a passive system. The simpler technique of negative eigenvalue deletion is implemented in this project.

Stability does not necessarily imply passivity, so the results of the Vector Fitting procedure must be made passive.



# CHAPTER VIII

## AUTOMATIC POLE DETERMINATION

So far, we have shown how to model a transmission line using a frequency dependent model and how to fit this model using a rational function approximation. What has not been discussed thus far is how to bootstrap the Vector Fitting technique with a set of known poles. This chapter shows several ways to do this.

Once we know how to completely apply Vector Fitting to our transmission line model, fitting results can be calculated.

### 8.1 Distribute Poles

The first method is to distribute the poles such that the imaginary parts are distributed in some fashion over the frequency range of interest. One could choose, say, a linear or logarithmic distribution.

For example, if the frequency range is 1 to 1000Hz, then 10 poles distributed linearly would look something like:

**Table 8-1:** Ten Linearly Distributed Poles over 1 to 1000Hz

Poles (Hz)
$-0.01 \pm j1$
$-1.12 \pm j112$
$-2.23 \pm j223$
$-3.34 \pm j334$
$-4.45 \pm j445$
$-5.56 \pm j556$
$-6.67 \pm j667$
$-7.78 \pm j778$
$-8.89 \pm j889$
$-10 \pm j1000$

There are several conditions that must be met when choosing poles in this way:

- The real parts must be negative to help ensure stability. Note that it is possible for the fitting procedure to produce unstable poles, however, if the system being fitted is stable and the initial guesses are stable, then the solution is probably more likely to converge to a stable one.
- Each complex pole must exist in a complex conjugate pair to ensure a real and causal system.
- For reasons discussed below, the real parts must be some small fraction of the imaginary parts.

Recall that Vector Fitting works better with curves that contain peaks, and it is these peaks which will anchor the solution. Smooth functions, on the other hand, do not contain peaks, however it is still useful to utilize the above procedure for creating a starting set of poles.

## 8.2 Intelligently Guess the Poles

A second way to choose a starting set of poles is to guess where they might be. This can be done by analyzing the behaviour of a curve having one or more real or complex poles.

### 8.2.1 Character of the Peaks

This section gives some motivation for choosing poles at the peaks and valleys. It will show how the character of first-order poles will determine an appropriate guess.

Start with real poles. Say we are given:

$$f(s) = \frac{c}{s-a} \quad (8-1)$$

Find the location of any maxima or minima:

$$|f(j\omega)|^2 = \frac{c^2}{|(j\omega) - a|^2} = \frac{c^2}{\omega^2 + a^2} \quad (8-2)$$

$$\frac{d|f(j\omega)|^2}{d\omega} = \frac{-c^2 2\omega}{(\omega^2 + a^2)^2} \quad (8-3)$$

The only solution, therefore, for  $\frac{d|f(j\omega)|^2}{d\omega} = 0$  is  $\omega = 0$ . This reveals that curves with real poles are smooth and don't have any peaks or valleys, so distributing the poles in some manner is probably a good way to choose the starting guesses.

Let's examine curves with complex poles. Say we have:

$$g(s) = \frac{c}{s-a} + \frac{\bar{c}}{s-\bar{a}} \quad (8-4)$$

Find the location of any maxima or minima:

$$|g(j\omega)|^2 = \frac{|2c'j\omega - (c\bar{a} + \bar{c}a)|^2}{|(j\omega)^2 - 2a'j\omega + |a|^2|^2} = \frac{(2(c'a' + c''a''))^2 + (2c'\omega)^2}{(-\omega^2 + |a|^2)^2 + (2a'\omega)^2} \quad (8-5)$$

$$\frac{d|g(j\omega)|^2}{d\omega} = \frac{8(c')^2\omega((-\omega^2 + |a|^2)^2 + (2a'\omega)^2) - ((2(c'a' + c''a''))^2 + (2c'\omega)^2)(4\omega^3 + 2(4(a')^2 - 2|a|^2)\omega)}{((-\omega^2 + |a|^2)^2 + (2a'\omega)^2)^2} \quad (8-6)$$

Solving for  $\frac{d|g(j\omega)|^2}{d\omega} = 0$ , we arrive at:

$$\omega^2 = \frac{-(c'a' + c''a'')^2 \pm a''|c|\sqrt{(2a'c' + a''c'')^2 + (c'a'')^2}}{(c')^2} \quad (8-7)$$

The second term is positive, so in order to obtain a real value for  $\omega$ , choose the “plus” solution. Thus, we arrive at an expression for the frequency at any peak of  $|g(j\omega)|$ :

$$\omega = \frac{\sqrt{-(c'a' + c''a'')^2 + a''|c|\sqrt{(2a'c' + a''c'')^2 + (c'a'')^2}}}{|c'|} \quad (8-8)$$

When  $a'$  is small compared to  $a''$ , the expression for  $\omega$  simplifies to

$$\omega \approx a'' \quad (8-9)$$

Note that there may not be a peak in the curve at all. To guarantee a peak, the relation

$$\sqrt{(2a'c' + a''c'')^2 + (c'a'')^2} \geq \frac{(c'a' + c''a'')^2}{a''|c|} \quad (8-10)$$

must be satisfied. If  $a'$  is again small compared to  $a''$ , this relation is always satisfied, as it reduces to

$$|c|^2 \geq (c'')^2, \quad (8-11)$$

which is always true.

Lastly, we need to solve Eq. (8-6) for  $\left|\frac{d|g(j\omega)|^2}{d\omega}\right| = \infty$ . It is possible to show that there is no solution for  $\omega$  unless we make the same approximation as before, that  $a' \ll a''$ . In this case, Eq. (8-9) holds true. In fact, an infinite slope implies that  $a' = 0$ .

### 8.2.2 Smooth Curves with Complex Poles

If the imaginary part of  $a$  is small with respect to the real part, then there won't be any peaks since there is no solution to Eqs. (8-8) and (8-10). The curve will be “smooth”.

This is similar to the case with all real poles, and choosing poles that are distributed in some fashion over the frequency range of interest is probably the best way to go.

### 8.3 Pole Choices

It is reasonable to assume, then, that the location of each peak lies near the imaginary part of a conjugate pair of poles. The Vector Fitting procedure is, after all, designed to take as input “good guesses” for the poles. The guesses will then be refined to the actual values. It was surmised in Ch. V that if the guesses are close to the actual values, then the procedure may converge faster.

### 8.4 Future Directions

It may be better to distribute poles in a non-linear or non-logarithmic fashion. For example, it may be better to distribute the poles more densely around the parts of the curves with more “wiggles”.

One suggested technique is to base the sample density on the absolute value of the second derivative of the magnitude or phase of the curve. Since the magnitude of the second derivative is higher near these portions of the curve, the sample density will be higher. This was also discussed in Sec. 5.10.1.

#### 8.4.1 Distribution Scheme Based on the Propagation Matrix

Ch. IV has a section on deriving the modal transformation matrix for each frequency sample of the propagation matrix. It is mentioned that both the Newton-Raphson technique and the correlation technique fail if successive frequency samples are not spaced closely enough.

It may be advantageous to distribute the frequency samples according to where there are “close” eigenvectors, or where these algorithms have trouble converging.

### 8.5 Chapter Summary

The Vector Fitting algorithm needs to be bootstrapped with a set of known starting poles. This chapter explored several ways to intelligently guess and/or distribute poles so that we have a good starting set.

# CHAPTER IX

## VECTOR FITTING RESULTS FOR THE TRANSMISSION LINE MODEL

The last step before a time domain simulation can be performed is to apply Vector Fitting to the overhead transmission line model. This chapter shows some selected fitting results for two line geometries.

Specifically, the propagation and characteristic admittance matrices are fitted for each geometry.

### 9.1 Terminology

It was shown that when each element of the propagation matrix,  $\mathbf{H}$  was fitted, it used an equation of the form

$$H_{ij}(j\omega) = \sum_{k=1}^n h_{m_{ij}}(j\omega) e^{-j\omega\tau_k}, \quad (9-1)$$

where  $h_{m_{ij}}(j\omega)$  is the *modal contribution* for element  $(i, j)$ . Please refer to Eq. (4-36) for the exact definition.

### 9.2 Two-Conductor Transmission Line

Table 9-1 shows the transmission line parameters.

**Table 9-1:** Two-Conductor Transmission Line

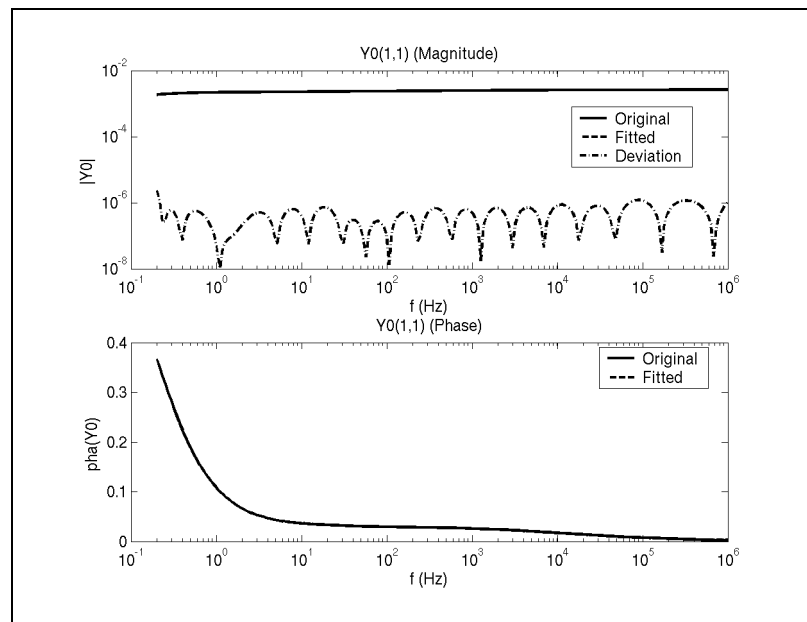
$x$ (m)	$y$ (m)	Radius (mm)	$\rho$ ( $\Omega \cdot \text{m}$ )
-12.5	20	68.2	$2.8 \times 10^{-8}$
12.5	12	107.7	$2.8 \times 10^{-8}$

The ground resistivity is set to  $10\Omega$ , and the line length is 300km. Table 9-2 shows the fitting parameters.

**Table 9-2: Fitting Parameters**

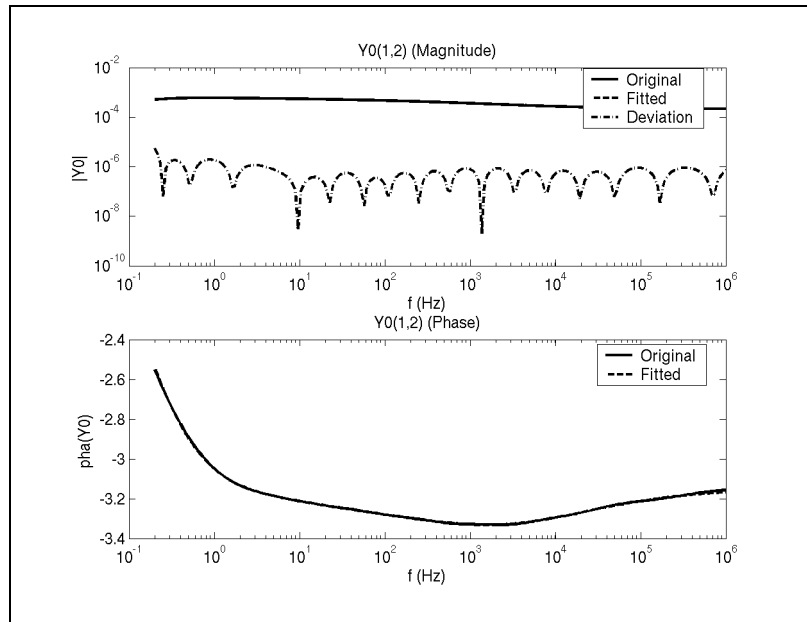
Parameter	Value
Frequency range	0.2 - $10^6$ Hz, 200 samples
Sample distribution	Logarithmic
Sample count	200
Fitting iterations	4
Unstable poles	Delete
Min. admittance poles	20
Min. propagation poles	20

The plots in Figs. 9-1 and 9-2 show two of the fitted elements of  $Y_0$ , the (1,1) and (1,2) elements respectively. The fitting is excellent. The simulation is very accurate when only the characteristic admittance is considered and the propagation matrix is neglected. i.e. when the simulation considers a very long line, such that the propagation effects are negligible.



**Fig. 9-1. Fitted (1,1) element of  $Y_0$ .**

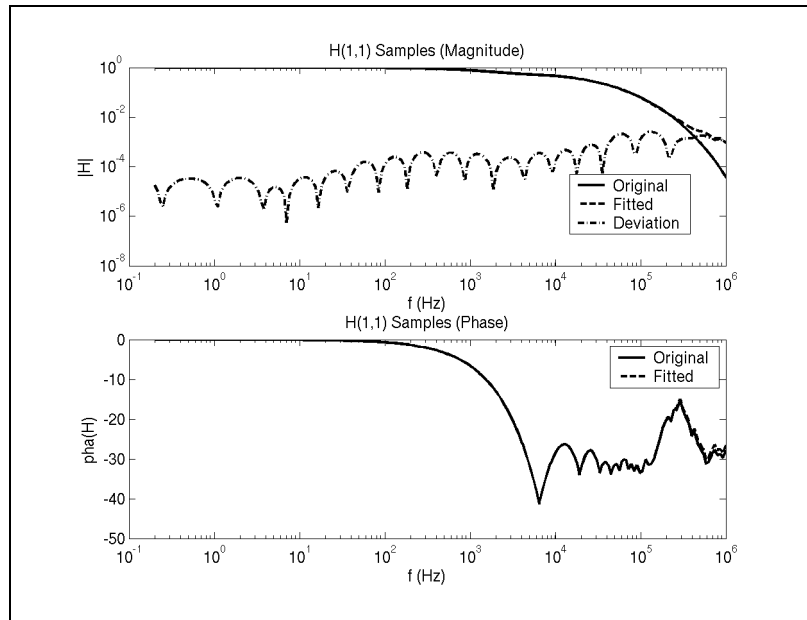
The maximum relative deviation in the magnitude was 2.78659% at 1MHz.



**Fig. 9-2.** Fitted (1,2) element of  $Y_0$ .

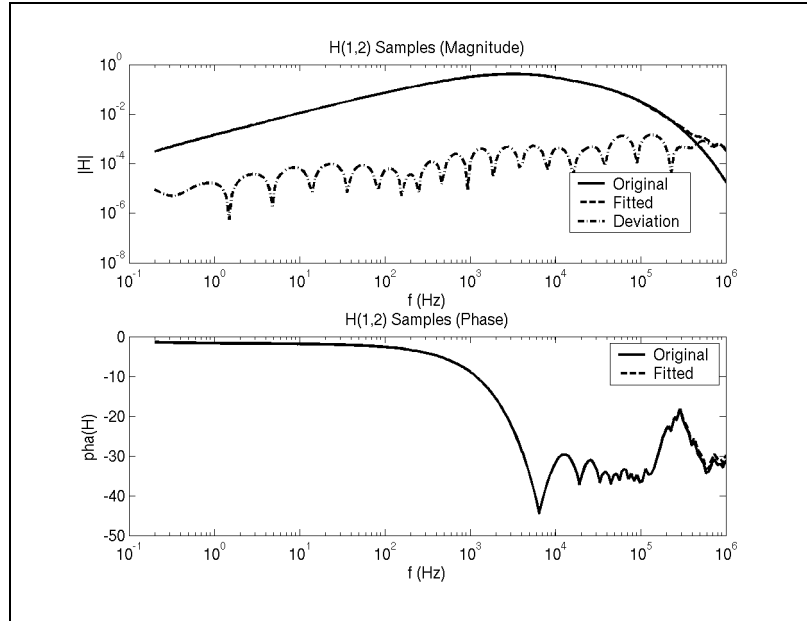
The maximum relative deviation in the magnitude was 4.02941% at 1MHz.

The plots in Figs. 9-3 and 9-4 show two of the fitted elements of  $H$ , the (1,1) and (1,2) elements, respectively. The fitting is very good, but starts to deviate at higher frequencies.



**Fig. 9-3.** Fitted (1,1) element of  $H$ .

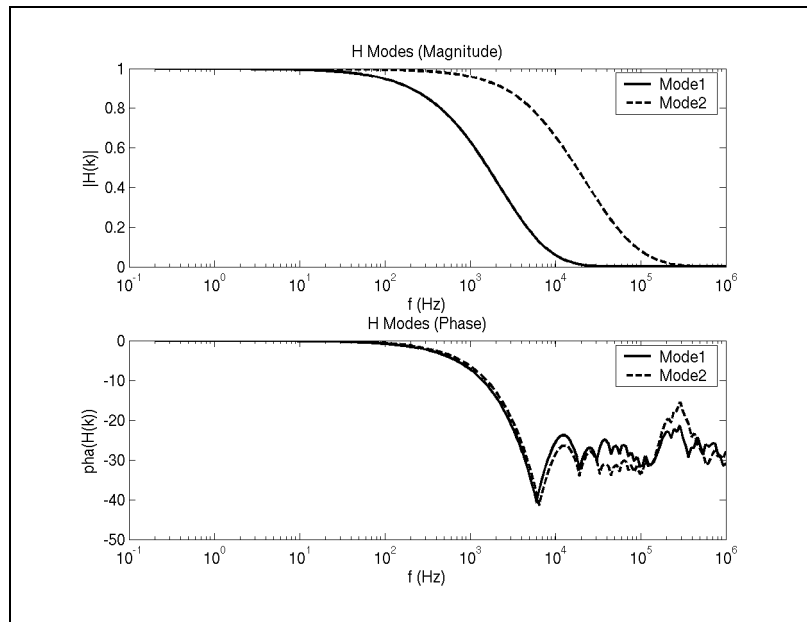
The maximum relative deviation in the magnitude was 2418.31% at 1MHz.



**Fig. 9-4.** Fitted (1,2) element of  $H$ .

The maximum relative deviation in the magnitude was 1820.95% at 1MHz.

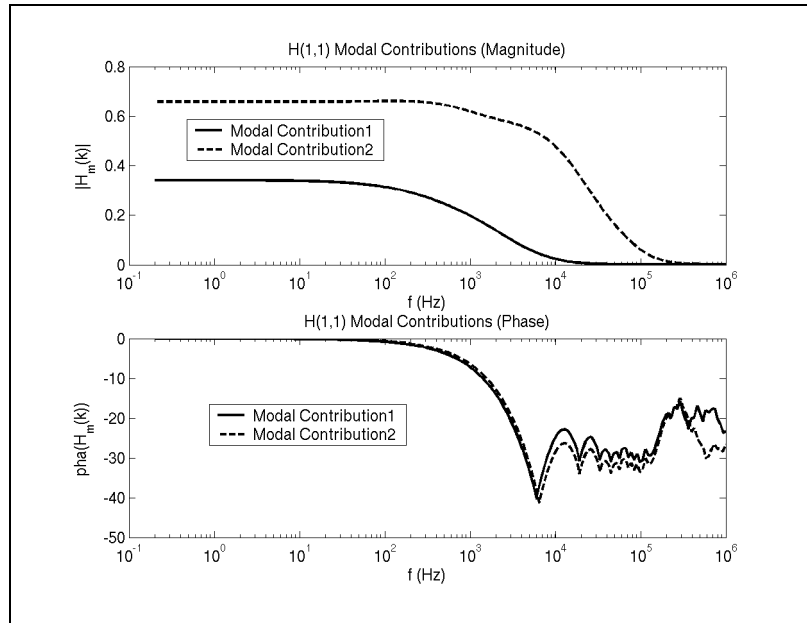
Fig. 9-5 shows a plot of the modes. Note that these were obtained using a constant transformation matrix.



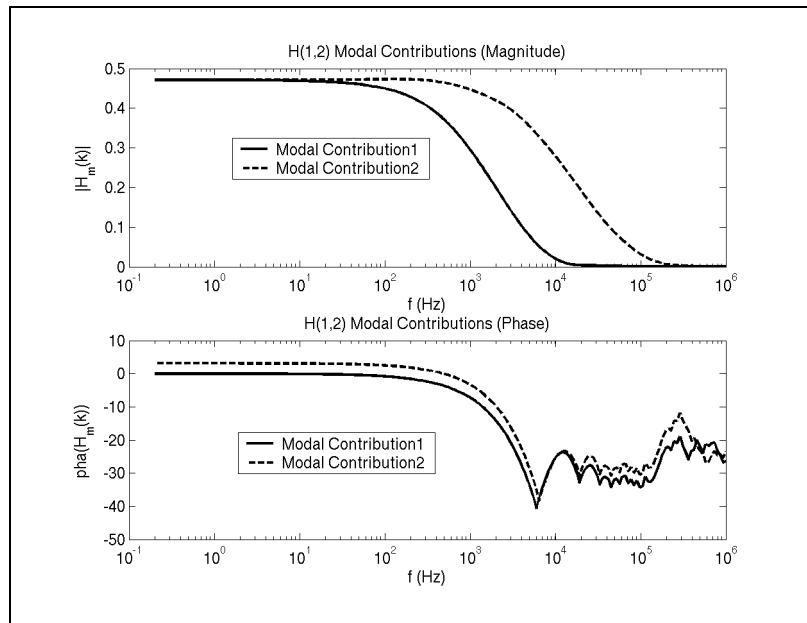
**Fig. 9-5.** The modes of  $H$ .



Finally, Figs. 9-6 and 9-7 show plots of the modal contributions for the (1,1) and the (1,2) elements, respectively.



**Fig. 9-6.** Modal contributions for the fitted (1,1) element of  $H$ .



**Fig. 9-7.** Modal contributions for the fitted (1,2) element of  $H$ .

### 9.3 Three-Conductor Transmission Line

Table 9-3 shows the parameters of the transmission line.

**Table 9-3:** Three-Conductor Transmission Line

$x$ (m)	$y$ (m)	Radius (mm)	$\rho$ ( $\Omega \cdot \text{m}$ )
-10	20	100	$2.8 \times 10^{-8}$
0	20	100	$2.8 \times 10^{-8}$
10	20	100	$2.8 \times 10^{-8}$

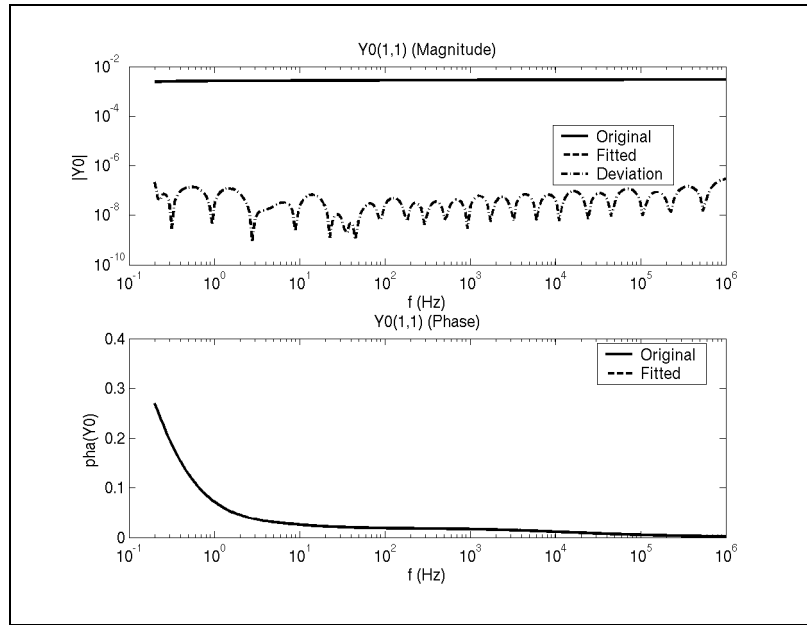
The ground resistivity is set to  $10\Omega$ , and the line length is 200km. Table 9-4 shows the fitting parameters.

**Table 9-4:** Fitting Parameters

Parameter	Value
Frequency range	$0.2 - 10^6$ Hz, 200 samples
Sample distribution	Logarithmic
Sample count	200
Fitting iterations	4
Unstable poles	Delete
Min. admittance poles	20
Min. propagation poles	20

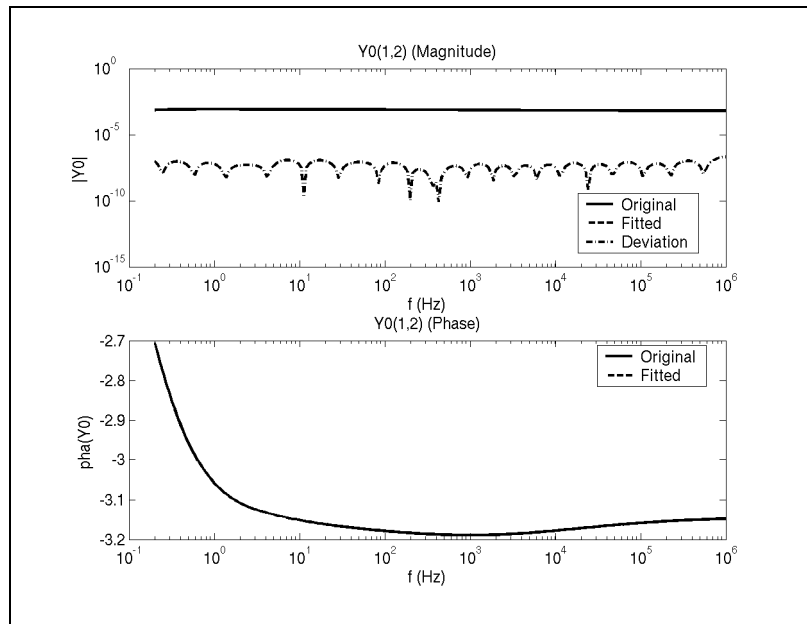
The plots in Figs. 9-8 and 9-9 show two of the fitted elements of  $\mathbf{Y}_0$ , the (1,1) and (1,2) elements respectively. The fitting is excellent. The simulation is very accurate when only the characteristic admittance is considered, and the propagation matrix is neglected. i.e.

when the simulation considers a very long line, such that the propagation effects are negligible.



**Fig. 9-8.** Fitted (1,1) element of  $Y_0$ .

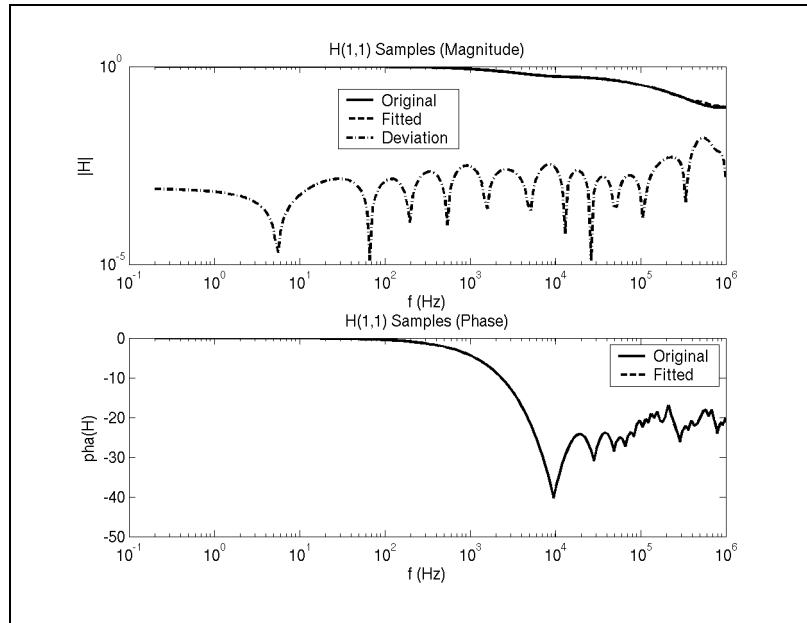
The maximum relative deviation in the magnitude was 0.000310565% at 1MHz.



**Fig. 9-9.** Fitted (1,2) element of  $Y_0$ .

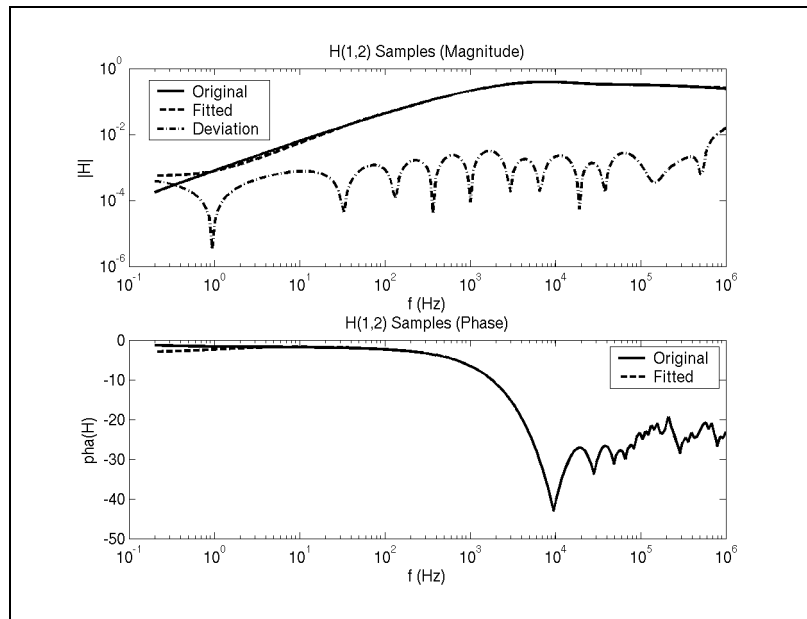
The maximum relative deviation in the magnitude was 0.0532089% at 0.2Hz.

The plots in Figs. 9-10 and 9-11 show two of the fitted elements of  $\mathbf{H}$ , the (1,1) and (1,2) elements, respectively. The fitting is very good, and does not deviate as much at higher frequencies, when compared to the two-conductor case.



**Fig. 9-10.** Fitted (1,1) element of  $\mathbf{H}$ .

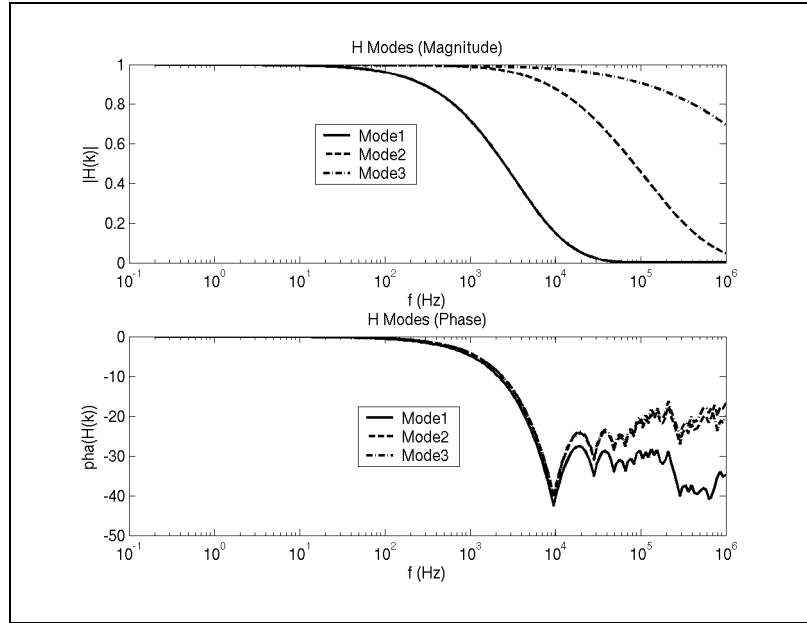
The maximum relative deviation in the magnitude was 14.8691% at 537.891kHz.



**Fig. 9-11.** Fitted (1,2) element of  $\mathbf{H}$ .

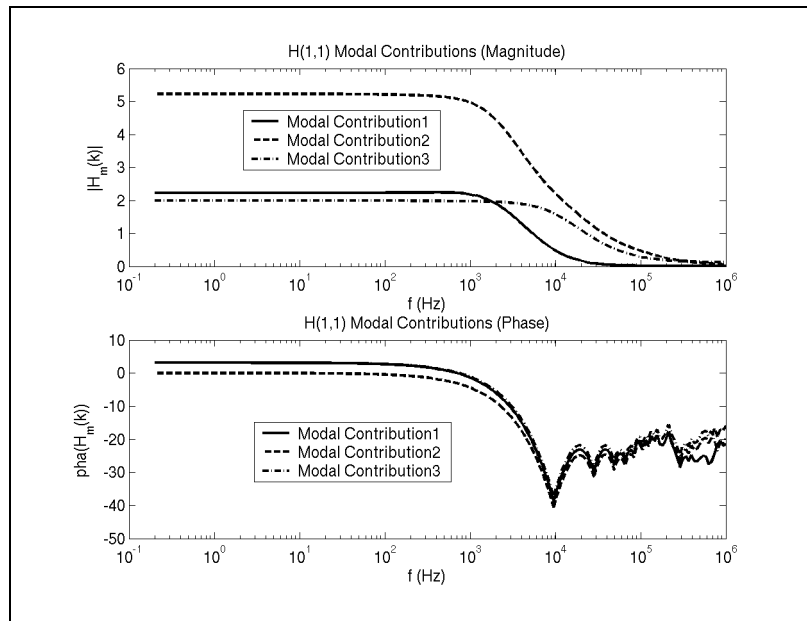
The maximum relative deviation in the magnitude was 221.709% at 0.2Hz.

Fig. 9-12 shows a plot of the modes. Note that these were obtained using a constant transformation matrix.



**Fig. 9-12.** The modes of  $H$ .

Finally, Figs. 9-13 and 9-14 show plots of the modal contributions for the (1,1) and the (1,2) elements, respectively.



**Fig. 9-13.** Modal contributions for the fitted (1,1) element of  $H$ .

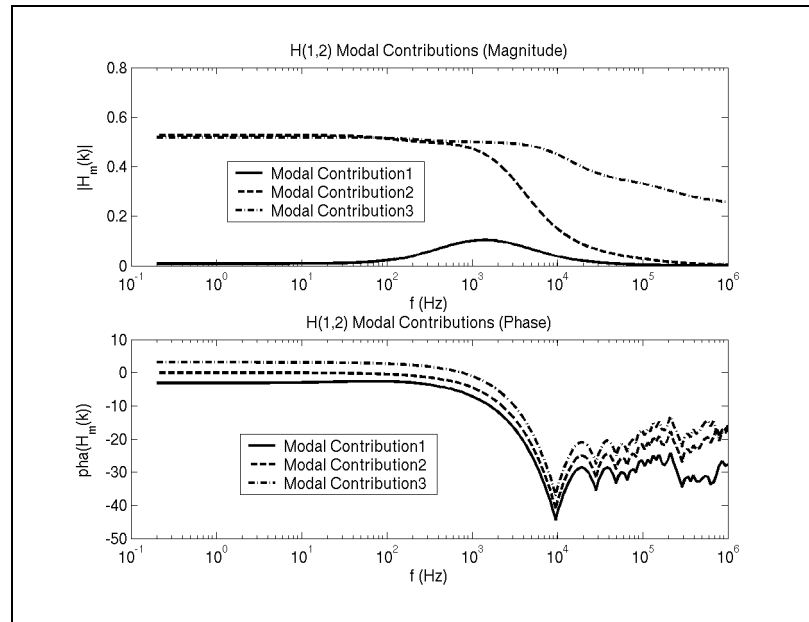


Fig. 9-14. Modal contributions for the fitted (1,2) element of  $H$ .

## 9.4 Chapter Summary

This chapter presented selected fitting results for the characteristic matrices of overhead transmission lines with two different geometries. The fitting results appeared to be better for the characteristic admittance matrix than for the propagation matrix.

# CHAPTER X

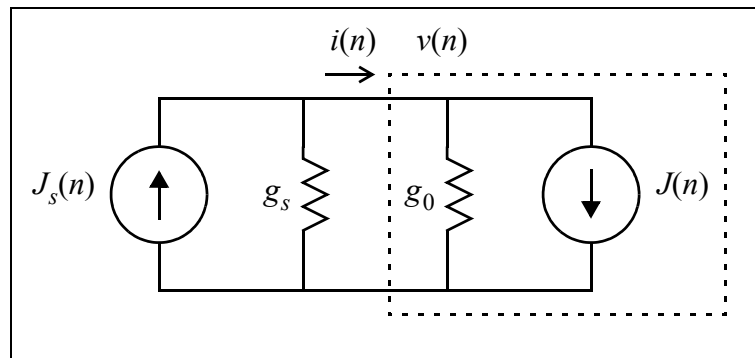
## SIMULATION IN THE TIME DOMAIN

This chapter shows how to take the results of the transmission line model fitting process and use them to produce a simulation in the time domain. The next chapter will show some selected simulation results.

In a simulator, we need to treat the transmission line as a “black box”. The standard way to represent any black box component is with an admittance and history current, and so our goal is to use the recursive convolution techniques of Ch. III to produce this form. The first topic of this chapter, therefore, will be a discussion of how exactly to simulate a component represented in this way so that we have an idea of where we’re headed.

### 10.1 The “Black Box”

The area inside the dotted line in Fig. 10-1 represents one side of the transmission line. The other side has the identical setup. Each of the quantities in the figure is a matrix, where the variables with a subscript  $s$  represent the source as seen by the transmission line, and the argument  $n$  is the  $n^{\text{th}}$  time step.



**Fig. 10-1.** The transmission line as a black box.

#### 10.1.1 Simulation Steps

Fig. 10-1 illustrates how to create a procedure for finding the present voltage and current flowing into the component. First, establish the following relationships.

$$v(n) = (g_s + g_0)^{-1} (J_s(n) - J(n)) \quad (10-1)$$

$$i(n) = J_s(n) - g_s v(n) = J(n) + g_0 v(n) \quad (10-2)$$

Clearly,  $J(n)$  cannot depend on  $v(n)$  since that would result in a circular definition. It is assumed, then, that this value is only ever dependent on previous values of  $v$ . A later section will show how this is true for the transmission line model developed in this project.

The sequence of steps, then, to obtain the present values of  $v$  and  $i$  are as follows.

For time step number  $n$  :

1. Obtain  $J(n)$  from the black box. For the 0<sup>th</sup> time step, this is really any initial state that the black box wishes to apply to the system. It is usually zero.
2. The applied source,  $J_s(n)$ , is known, so calculate  $v(n)$  from Eq. (10-1).
3. Let the black box know about  $v(n)$  so that it can calculate its current for the next time step.
4. Calculate  $i(n)$  from either part of Eq. (10-2).

It's that simple.

It is important to note that the component's history current must not be dependent on any present values.

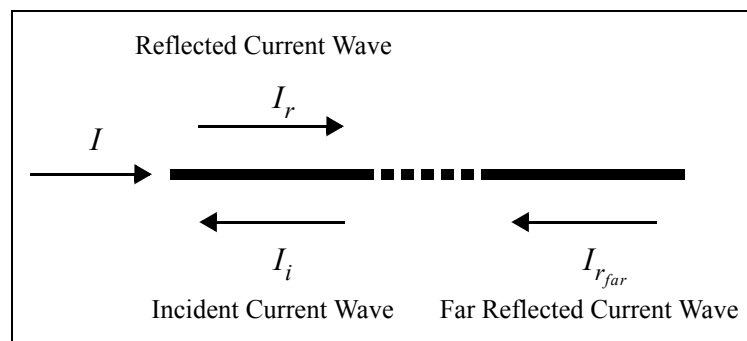
## 10.2 Transmission Line Update Equations

Now that we have a picture of how the transmission line model should fit into a simulator, we can continue with the derivation.

Start with the equation relating the backwards-travelling current wave with the current and voltage at one end of the transmission line. Please see Appendix A for the derivation.

Note, however, that the terminology used here is slightly different than in the appendix.

Fig. 10-2 shows the naming convention that will be used for this section.



**Fig. 10-2.** Conventions for current.



Use Eq. (A-6) from the appendix and the fact that  $I_i = HI_{r_{far}}$  to produce the relation:

$$Y_0 V - I = 2HI_{r_{far}} \quad (10-3)$$

As well, it is evident that:

$$I = I_r - I_i \quad (10-4)$$

These equations are all that is needed to arrive at a set of “update equations” for the complete simulation of a transmission line in the time domain:

$$i(n) = y_0 \otimes v - 2 \sum_k h_k \otimes i_{r_{far}}(n - \tau_k) \quad (10-5)$$

$$i_r(n) = i(n) + i_i(n) \quad (10-6)$$

The notation,  $i(n)$ , denotes the value of  $i(t)$  at  $t = n\Delta t$ . Also, recall that  $H$  was expressed as the sum of  $k$  components, each with a time delay,  $\tau_k$ . Eq. (10-5) is just the time domain form of Eq. (4-16).

Strictly speaking, the notation for  $i_{r_{far}}(n - \tau_k)$  isn't quite rigorous, but it is a convenient way to express the fact that we need the value for  $i_{r_{far}}$  at a specific previous time step. Specifically, it means whichever time step occurred  $\tau_k$  seconds ago. Interpolation is used to find these values because  $\tau_k$  may not be an exact multiple of our time step.

### 10.3 Convolution with the Characteristic Admittance, $Y_0$

Recall that each element of  $Y_0$  is represented by the expression  $\sum_j \frac{c_j}{s - a_j} + d$ . In state equation form, write the update equation for the convolution  $y_0 \otimes v$  as

$$\begin{aligned} y(n) &= \left[ \sum_j x_j(n) \right] + dv(n) \\ &= \left[ \left( \sum_j \beta_j \right) v(n-1) + \sum_j \gamma_j x_j(n-1) \right] + \left[ \left( \sum_j \alpha_j \right) + d \right] v(n) \end{aligned} \quad (10-7)$$

where the coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$  are computed as in Ch. III.

The factor multiplying  $v(n)$  is the “conductance” we need for inserting the transmission line into a circuit solver. Its value is

$$G = \left[ \sum_j \alpha_j \right] + d, \quad (10-8)$$

and it can be precalculated.

Note that it can't really be separated out in the expression for  $x_k(n)$  because each term of  $G$  contributes to  $x_k(n)$  separately.

## 10.4 Convolution with the Propagation Function, $H$

Convolutions performed using the propagation function are done in a similar way. However, each input has a time delay for each mode.

Recall the earlier derivation that:

$$HI_{r_{far}} = \sum_k H_k e^{-j\omega\tau_k} I_{r_{far}} \quad (10-9)$$

Thus:

$$h \otimes i_{r_{far}} = \sum_k h_k \otimes i_{r_{far}}(t - \tau_k) \quad (10-10)$$

The convolution operator is commutative, and so the time delay can be applied to the far-side current first.

Since each mode of  $H$  has the form  $\sum_i \frac{c_j}{s - a_j}$ , the recursive relation for  $h \otimes i_{r_{far}}$  is:

$$\begin{aligned} y(n) &= \sum_k \sum_j x_{j_k}(n) \\ &= \sum_k \left[ \left( \sum_i \alpha_{j_k} \right) i_{r_{far}}(n - \tau_k) + \left( \sum_i \beta_{j_k} \right) i_{r_{far}}(n - 1 - \tau_k) + \sum_i \gamma_{j_k} x_{j_k}(n - 1) \right] \end{aligned} \quad (10-11)$$

As for the delay, the case where  $\tau_k$  is not an exact multiple of the time step is handled via linear interpolation.

This entire expression is a “history term”, so no equivalent “conductance” needs to be constructed, unless high-frequency poles are removed for efficiency, but this is discussed in the next section.

## 10.5 Eliminating High-Frequency Poles

There may be high frequency poles that decay too quickly to be “noticed” by the time step in the simulation. Recall that a pole in the frequency domain is equivalent to a decaying exponential, with the rate of decay determined by the real part of the pole.

$$\frac{c}{s-a} \Leftrightarrow ce^{at} \quad \text{and} \quad \frac{c}{s-a} + \frac{\bar{c}}{s-\bar{a}} \Leftrightarrow 2e^{a't}(c' \cos(a''t) - c'' \sin(a''t)) \quad (10-12)$$

It is possible to ignore these terms. First, determine how much decay is acceptable in one time step:

$$e^{a'\Delta t} < \varepsilon \quad (10-13)$$

A good value for the tolerance,  $\varepsilon$ , is the unit round-off for the machine precision, about  $10^{-16}$  for double precision. Note that  $a'$ , the real part of the pole, is usually stable, so it has a negative value.

Continue by taking the log of each side, and then only work with positive quantities:

$$\frac{|\ln \varepsilon|}{\Delta t} < |a'| \quad (10-14)$$

There is a section in [11] that has an equivalent relation, except that it uses  $10\pi$  instead of  $|\ln \varepsilon|$ , however since  $|\ln \varepsilon| \approx 10\pi$  for the double precision unit round-off, the authors may have done the same calculation.

If a term with a pole is neglected, then the final formulation must compensate for its disappearance. What we are really doing is subtracting a  $c/(s-a)$  term. A good way to compensate for this is to add back the value at DC. In other words, add  $c/-a$  for every removed term.

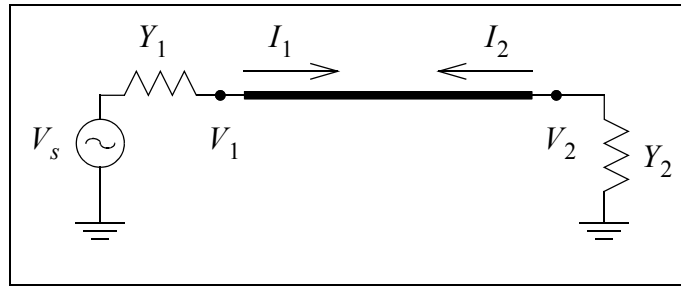
For the propagation function,  $H$ , this actually introduces a conductance. This is handled in exactly the same way that the conductance is handled for the characteristic admittance function,  $Y_0$ .

Lastly, if any poles are removed, then the conductance should probably be checked. Specifically, the conductance matrix for either the propagation or characteristic admittance

functions must have positive values along the diagonal, and negative values everywhere else. If the new conductance matrix does not have this form, then there may be a problem, although not necessarily in the case of the propagation matrix. This needs to be explored further.

## 10.6 Simulation in the Frequency Domain

Simulation in the frequency domain is used in this project to verify the results of the time domain simulation. Presented below is a derivation for the exact solution of the transmission line voltages and currents. Fig. 10-3 shows the conventions that will be used during the derivation.



**Fig. 10-3.** Test configuration.

It is known that:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_0 \coth(\Gamma l) & -Y_0 \operatorname{csch}(\Gamma l) \\ -Y_0 \operatorname{csch}(\Gamma l) & Y_0 \coth(\Gamma l) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (10-15)$$

and it can be seen from Fig. 10-3 that:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_1(V_s - V_1) \\ -Y_2 V_2 \end{bmatrix} \quad (10-16)$$

Using the constraint from Eq. (10-16), it is true that

$$\begin{bmatrix} Y_0 \coth(\Gamma l) + Y_1 & -Y_0 \operatorname{csch}(\Gamma l) \\ -Y_0 \operatorname{csch}(\Gamma l) & Y_0 \coth(\Gamma l) + Y_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} Y_1 V_s \\ \mathbf{0} \end{bmatrix}, \quad (10-17)$$

and the voltages and currents on both sides can be calculated in the frequency domain.

The results from this frequency domain calculation will be plotted with the results of the time domain simulation for verification of the simulated values.

## 10.7 Chapter Summary

This chapter showed how to apply recursive convolution to the fitted transmission line model. The resulting form includes a conductance and current history. This is required for the standard “black box” simulation technique.

Also shown was how to improve the efficiency of the simulation by removing poles having a very high frequency.

Lastly, it was shown how to calculate exact results in the frequency domain. This is useful because these results can be used to verify the time domain calculations.

# CHAPTER XI

## SIMULATION RESULTS

Our toolbox is now complete. We can apply the recursive convolution techniques to a vector fitted transmission line model using the simulation method presented in Ch. X. If we also plot these results and compare them with the results from exact frequency domain calculations, then we can verify the correctness of our model.

The two transmission lines of Ch. IX are simulated in various configurations. The terminology of Fig. 10-3 is used to describe the external parameters.

In all cases, realistic examples will be shown first, and then more “extreme” cases shown after. For example, a balanced 60Hz input with a characteristic admittance termination is considered “realistic”, while the same input or a step input applied to a line with a short circuit termination is considered “more extreme”.

When plotting the comparisons with the exact results, only the voltage on side two will be considered.

As well, all simulations use the “assume linear” integration type of Ch. III, and also weight the fitted curve by 100 at the frequencies of interest.

PSCAD results are shown for comparison, and it will be seen that for the most part, the results from our simulator are just as accurate. Note that the PSCAD values are shown in kilovolts and not volts.

### 11.1 Two-Conductor Transmission Line

Table 11-1 shows the transmission line parameters.

**Table 11-1:** Two-Conductor Transmission Line

$x$ (m)	$y$ (m)	Radius (mm)	$\rho$ ( $\Omega \cdot \text{m}$ )
-12.5	20	68.2	$2.8 \times 10^{-8}$
12.5	12	107.7	$2.8 \times 10^{-8}$

The ground resistivity is set to  $10\Omega$ , and the line length is 300km. Table 11-2 shows the fitting parameters.

**Table 11-2:** Fitting Parameters for Two-Conductor Line

Parameter	Value
Frequency range	$0.2 - 10^6$ Hz, 200 samples
Sample distribution	Logarithmic
Sample count	200
Fitting iterations	10
Unstable poles	Delete
Min. admittance poles	20
Min. propagation poles	10

It was found that more iterations resulted in a simulation that did not change every time it was run. To explain this statement, it is useful to reiterate the point from Ch. V about convergence to a local minimum in the Vector Fitting algorithm. The local minimum to which Vector Fitting converges is dependent on the starting poles. Furthermore, each iteration starts with a new, calculated, set of starting poles ultimately derived from an eigenvalue calculation. Because the eigenvalue routine from the matrix library does not produce identical values every time, the results of the fitting are not exactly the same each time.

Hence, a higher iteration count gives a better chance for the Vector Fitting to converge to a stable point, and by “stable” we mean “stays in the same general area”. Much more investigation needs to be done on this topic, however.

### 11.1.1 Applied 60Hz Voltage, Characteristic Termination

A 60Hz voltage was applied to side one, and side two was terminated with the real part of the characteristic admittance calculated at 60Hz. Table 11-3 shows the simulation parameters for this case.

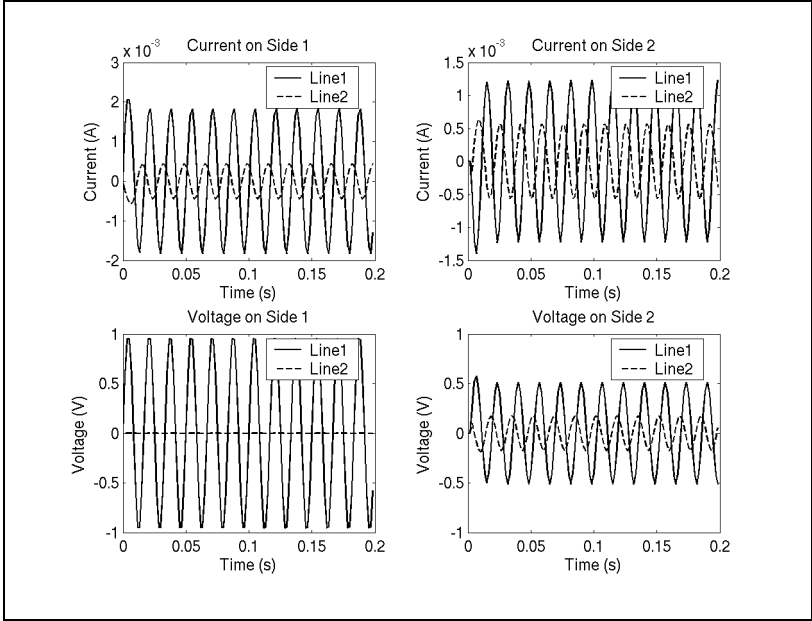
**Table 11-3:** Simulation Parameters, 60Hz, Characteristic Termination

Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 60t) \\ 0 \end{bmatrix} \text{V}$

**Table 11-3:** Simulation Parameters, 60Hz, Characteristic Termination

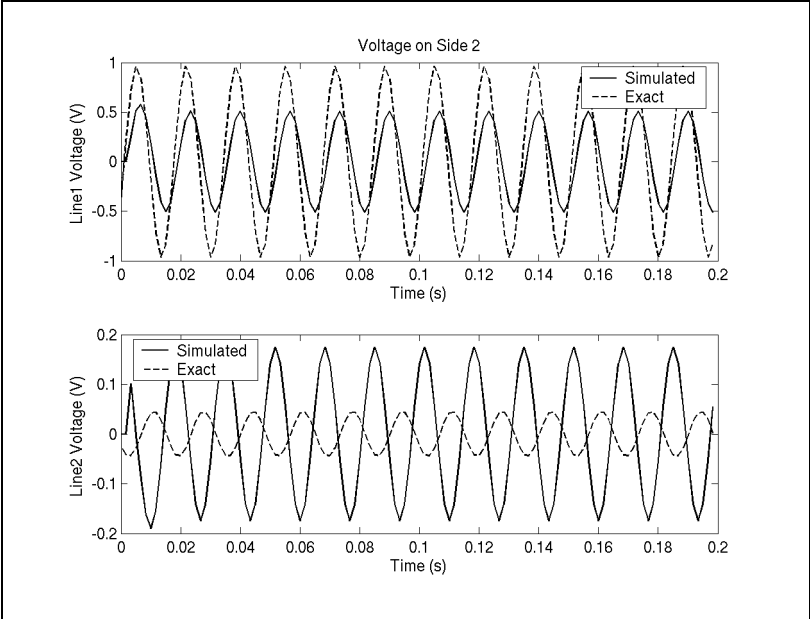
Parameter	Value
$Y_1$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \times 10^6 \end{bmatrix} S$
$Y_2$	$\begin{bmatrix} 0.0023563 & -4.8153 \times 10^{-4} \\ -4.8153 \times 10^{-4} & 0.0026371 \end{bmatrix} S$

Figs. 11-1 and 11-2 show the simulation with a time step of 1.667ms.



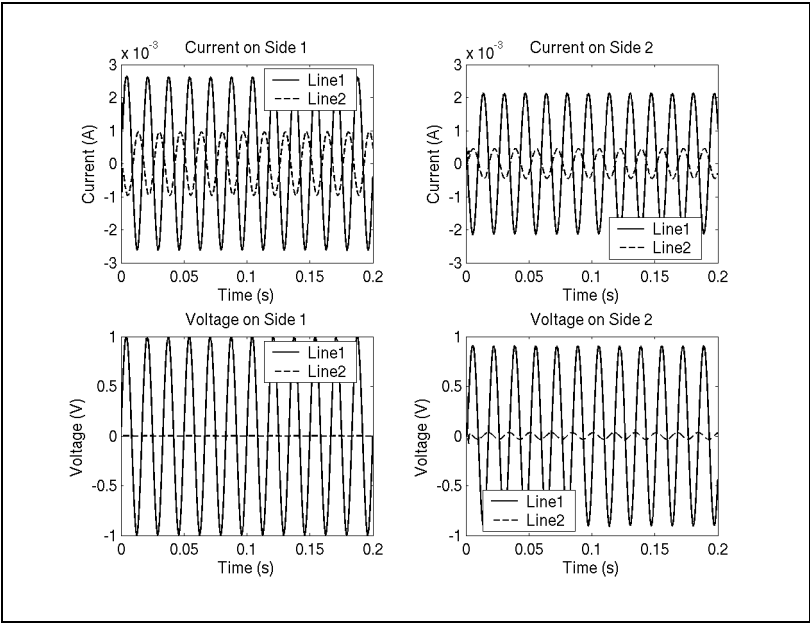
**Fig. 11-1.** Two conductors, 60Hz voltage, characteristic load, 1.667ms step.





**Fig. 11-2.** Two conductors, 60Hz voltage, characteristic load, 1.667ms step, exact.

Figs. 11-3 and 11-4 show the simulation with a time step of 50μs.



**Fig. 11-3.** Two conductors, 60Hz voltage, characteristic load, 50μs step.

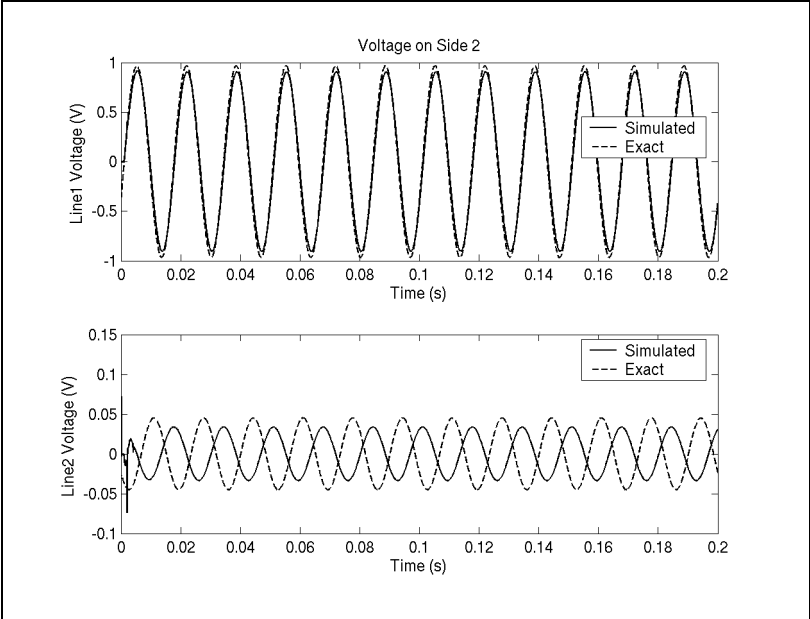


Fig. 11-4. Two conductors, 60Hz voltage, characteristic load, 50μs step, exact.

The results are more accurate with the smaller time step, however, the phase is incorrect on line two. The PSCAD simulation shown in Fig. 11-5 is very accurate, and closely matches the exact results, however the magnitude is too large.

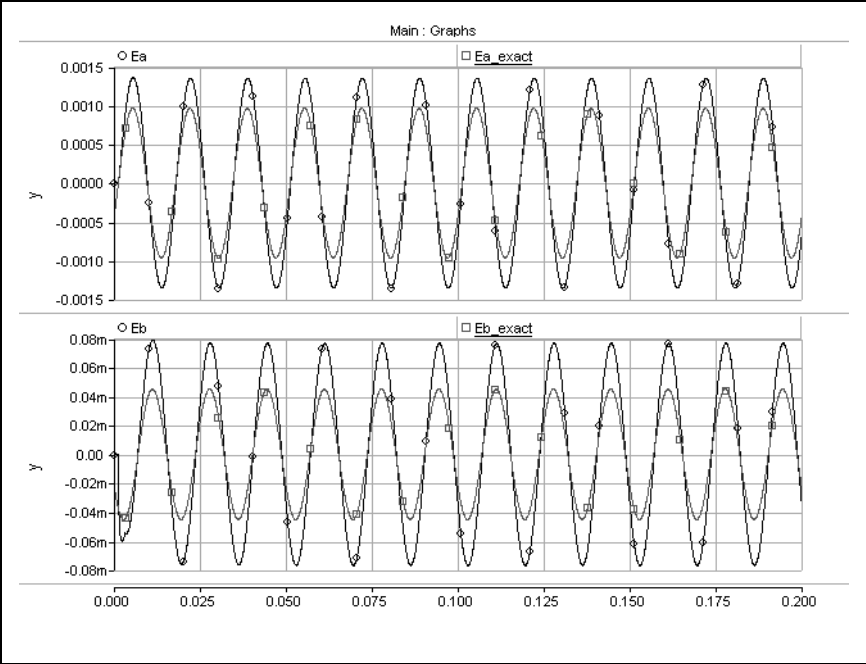


Fig. 11-5. Two conductors, 60Hz voltage, characteristic load, 50μs step, PSCAD.

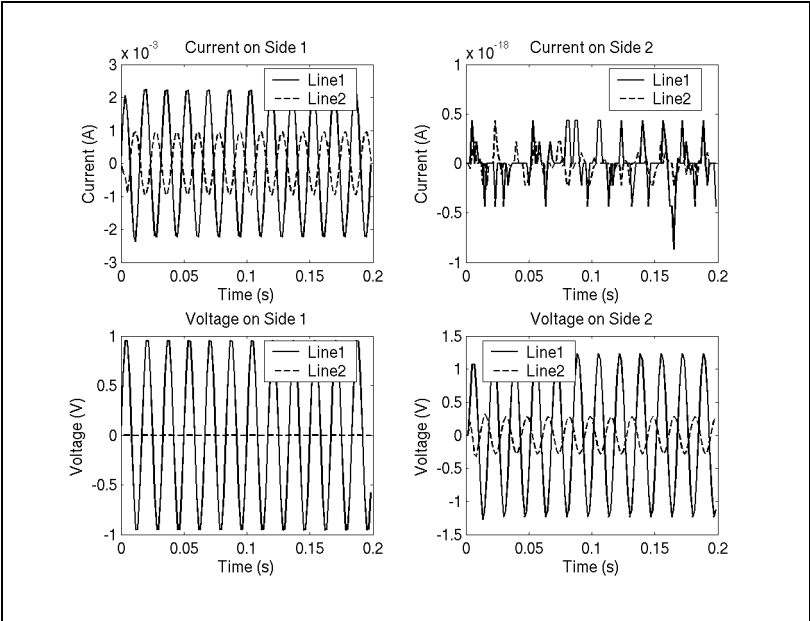
**11.1.2 Applied 60Hz Voltage, Open Circuit Termination**

A 60Hz voltage was applied to side one, and side two was terminated with an open circuit. Table 11-4 shows the simulation parameters for this case.

**Table 11-4:** Simulation Parameters, 60Hz, Open Circuit Termination

Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 60t) \\ 0 \end{bmatrix} \text{ V}$
$Y_1$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \times 10^6 \end{bmatrix} \text{ S}$
$Y_2$	Open circuit

Figs. 11-6 and 11-7 show the simulation with a time step of 1.667ms.



**Fig. 11-6.** Two conductors, 60Hz voltage, open circuit load, 1.667ms step.

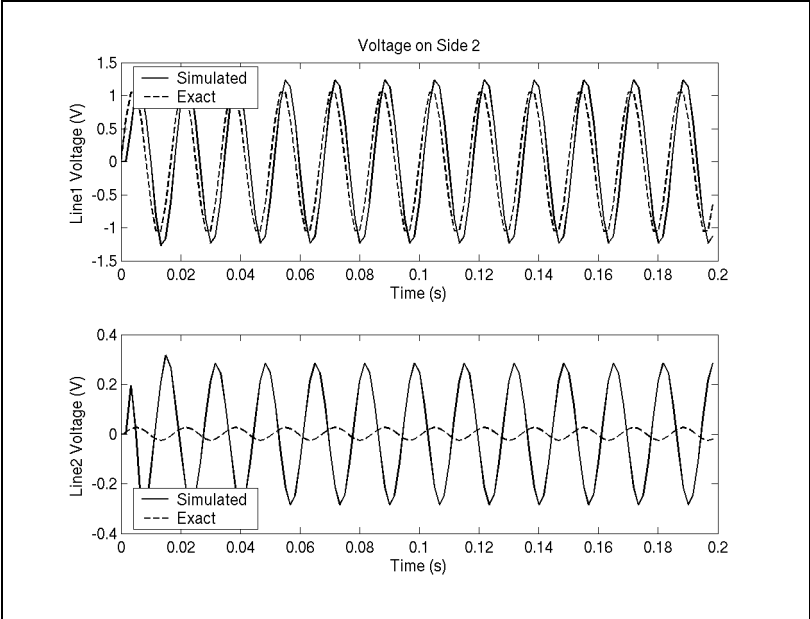


Fig. 11-7. Two conductors, 60Hz voltage, open circuit load, 1.667ms step, exact.

Figs. 11-8 and 11-9 show the simulation with a time step of 50μs.

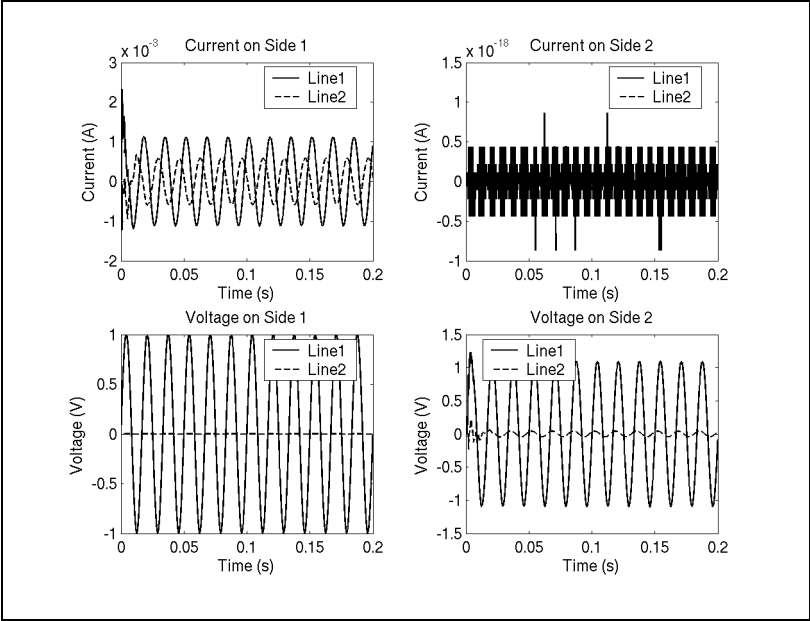


Fig. 11-8. Two conductors, 60Hz voltage, open circuit load, 50μs step.

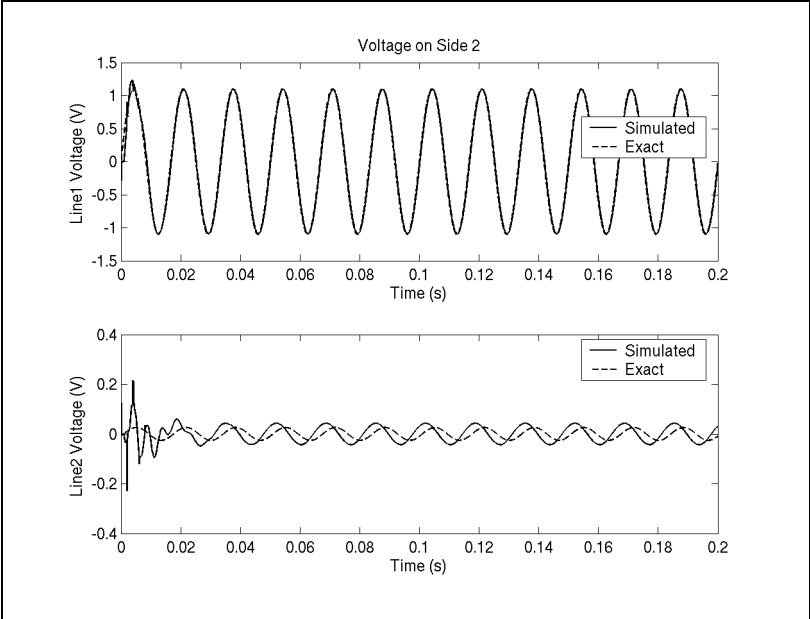


Fig. 11-9. Two conductors, 60Hz voltage, open circuit load, 50μs step, exact.

The results are more accurate with the smaller time step, however, the phase is again incorrect on line two. The PSCAD simulation shown in Fig. 11-10 was also very accurate, and closely matched the exact results, but stabilized much more slowly on line two. The magnitude is again slightly too large.

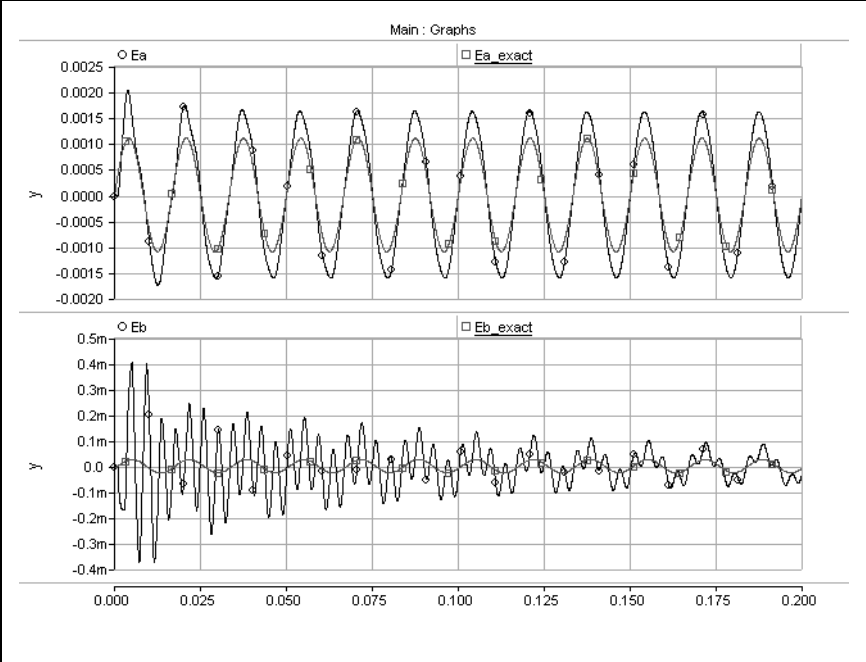


Fig. 11-10. Two conductors, 60Hz voltage, open circuit load, 50μs step, PSCAD.

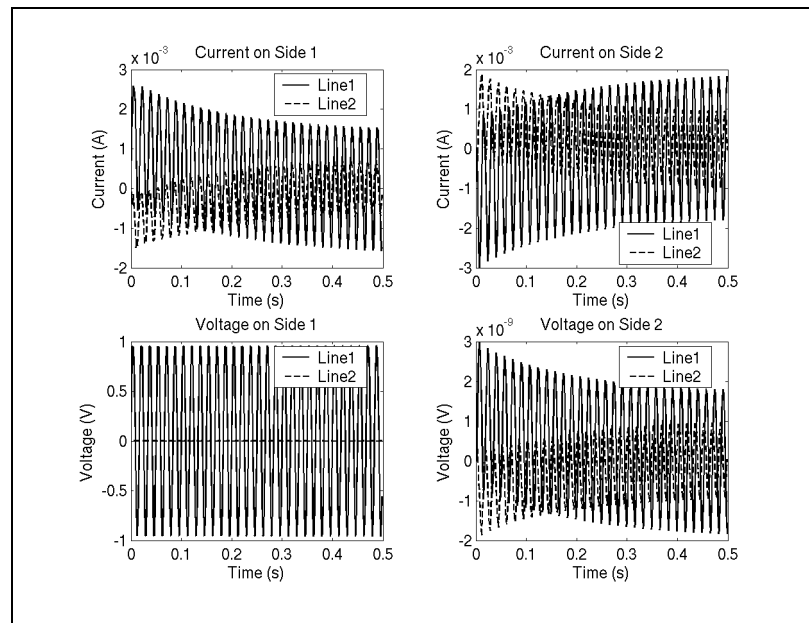
### 11.1.3 Applied 60Hz Voltage, Short Circuit Termination

A 60Hz voltage was applied to side one, and side two was terminated with a short circuit. Table 11-5 shows the simulation parameters for this case.

**Table 11-5:** Simulation Parameters, 60Hz, Short Circuit Termination

Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 60t) \\ 0 \end{bmatrix} \text{ V}$
$Y_1$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \times 10^6 \end{bmatrix} \text{ S}$
$Y_2$	$\begin{bmatrix} 1 \times 10^6 & 0 \\ 0 & 1 \times 10^6 \end{bmatrix} \text{ S}$

Figs. 11-11 and 11-12 show the simulation with a time step of 1.667ms.



**Fig. 11-11.** Two conductors, 60Hz voltage, short circuit load, 1.667ms step.

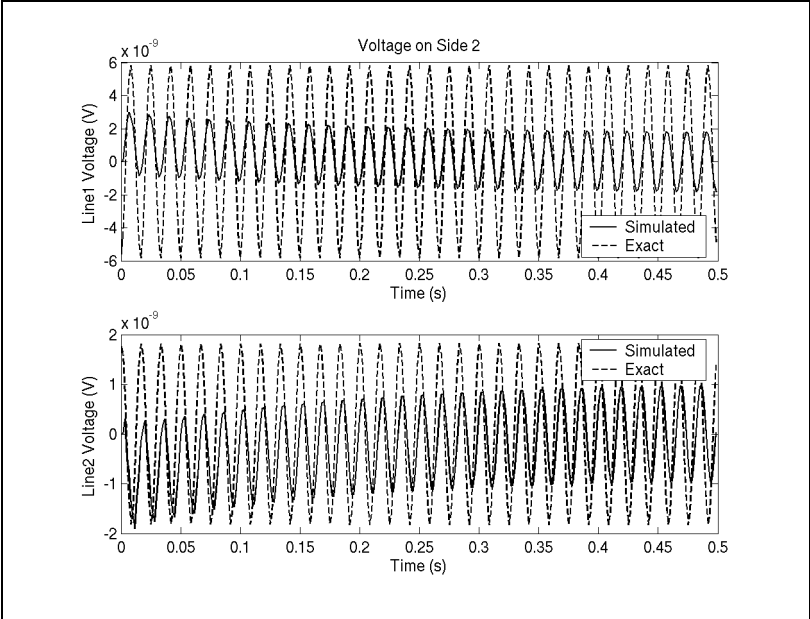


Fig. 11-12. Two conductors, 60Hz voltage, short circuit load, 1.667ms step, exact.

Figs. 11-13 and 11-14 show the simulation with a time step of 50μs.

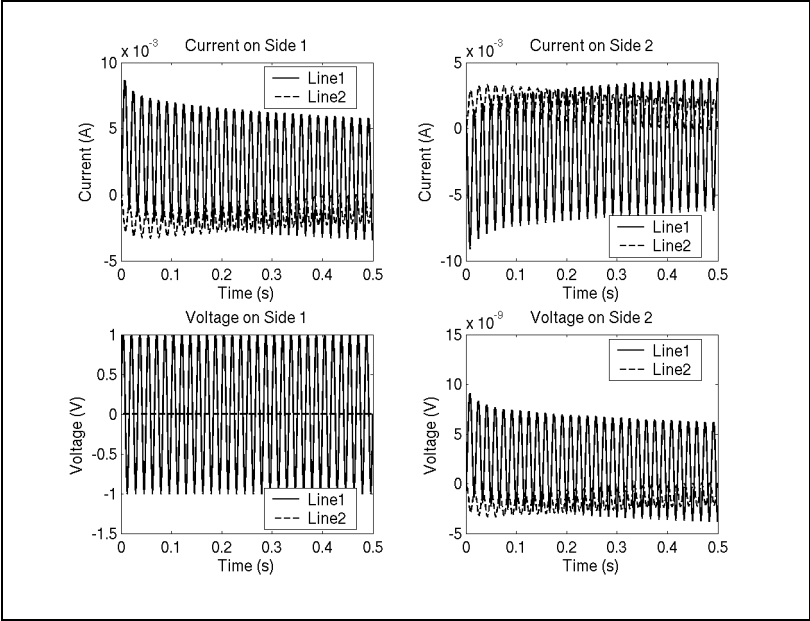


Fig. 11-13. Two conductors, 60Hz voltage, short circuit load, 50μs step.

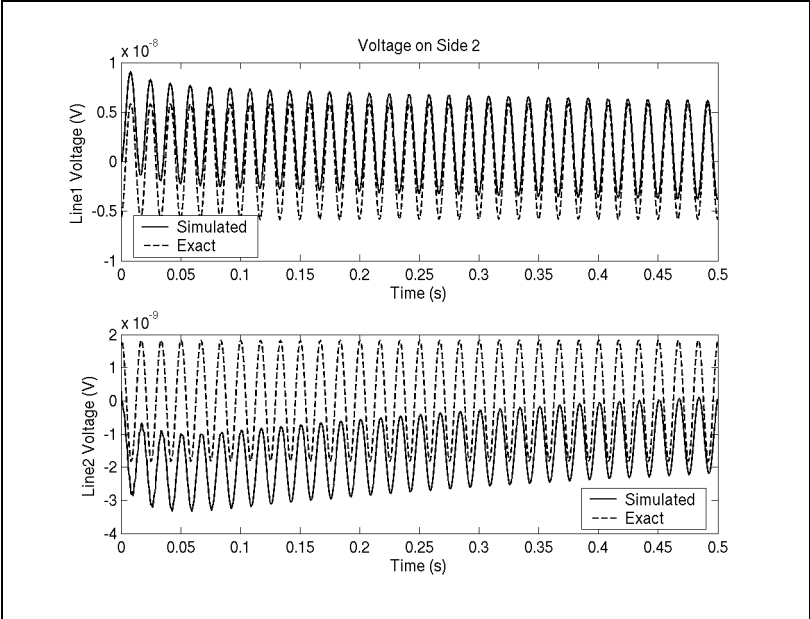


Fig. 11-14. Two conductors, 60Hz voltage, short circuit load, 50μs step, exact.

The results are more accurate with the smaller time step, however, the phase is correct on line two, unlike the previous two cases. The PSCAD simulation shown in Fig. 11-15 was also very accurate, and closely matched the exact results. It stabilized in about the same way.

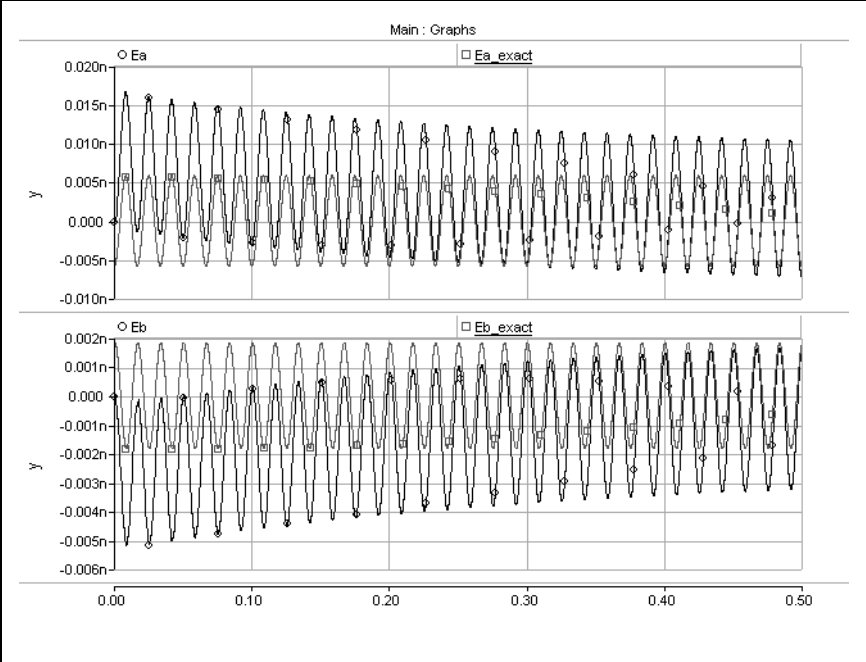


Fig. 11-15. Two conductors, 60Hz voltage, short circuit load, 50μs step, PSCAD.

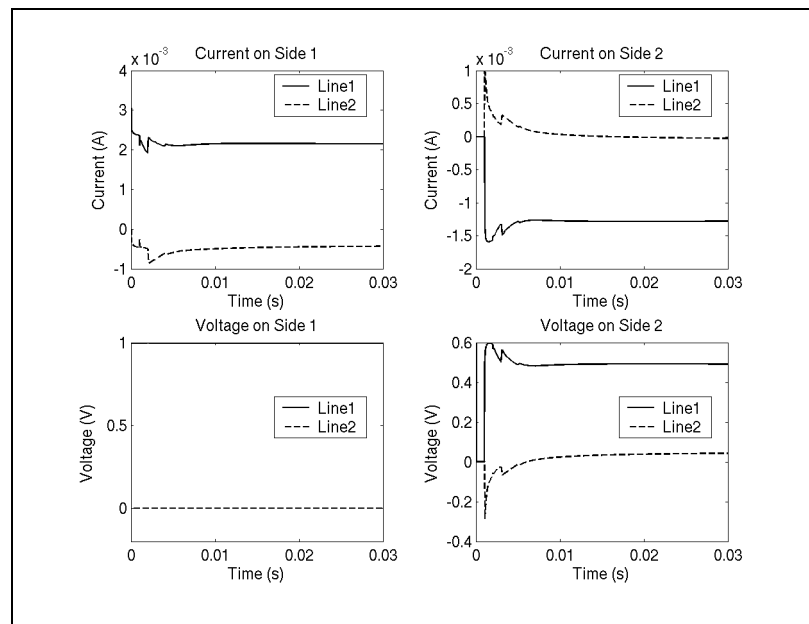


### 11.1.4 Applied Step Voltage, $5\mu\text{s}$ Time Step

So far, we have seen that a smaller time step seems to produce more accurate simulations. This section and the next section will compare a more “extreme” example by applying a step voltage to the three different terminations at different time steps. The same termination on side one was used.

The fitted curves were weighted as  $1/\omega$  during the fitting process because a step has a  $1/s$  distribution of frequencies. This is instead of weighting the matrices at one frequency only, as in the previous examples.

Figs. 11-16, 11-17, and 11-18 show the results with characteristic admittance, open circuit, and short circuit termination, respectively.



**Fig. 11-16.** Two conductors, step voltage, characteristic load,  $5\mu\text{s}$  step.

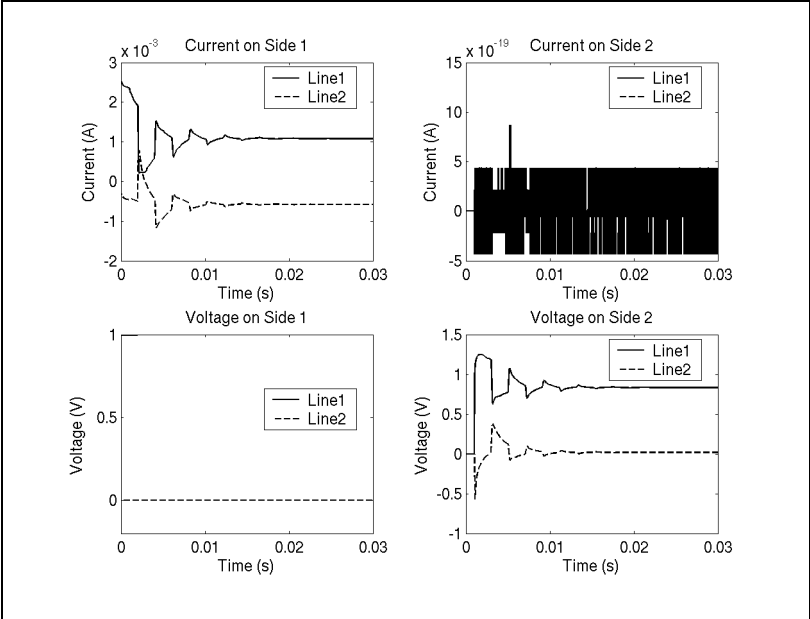


Fig. 11-17. Two conductors, step voltage, open circuit load,  $5\mu\text{s}$  step.

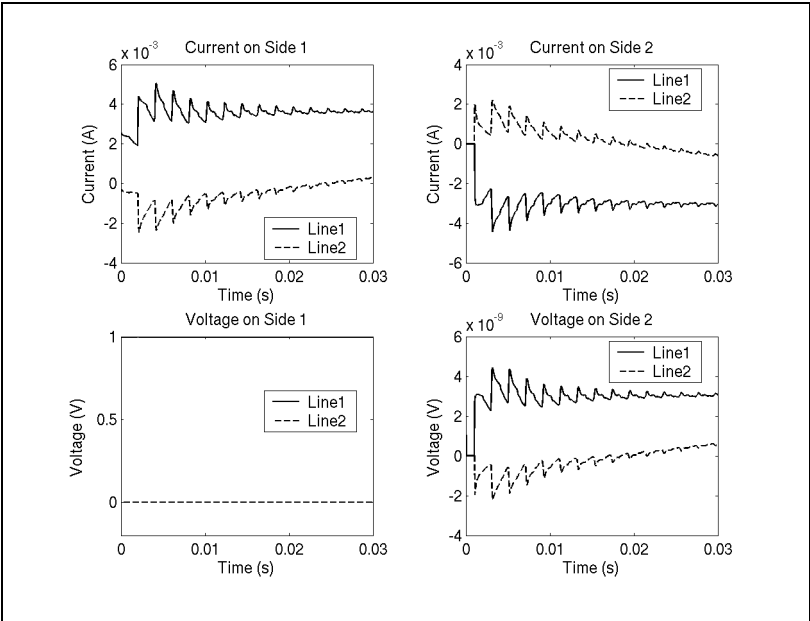


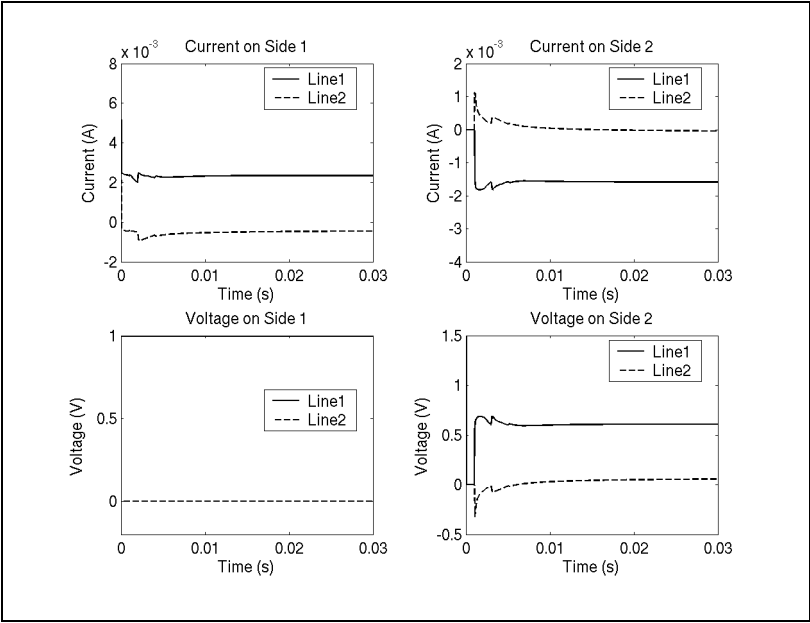
Fig. 11-18. Two conductors, step voltage, short circuit load,  $5\mu\text{s}$  step.

Each result was stable.

### 11.1.5 Applied Step Voltage, 0.5μs Time Step

The 0.5μs time step was chosen because that represents twice the sampling rate of the highest fitted frequency, 1MHz.

Figs. 11-19, 11-20, and 11-21 show the results with characteristic admittance, open circuit, and short circuit termination, respectively.



**Fig. 11-19.** Two conductors, step voltage, characteristic load, 0.5μs step.

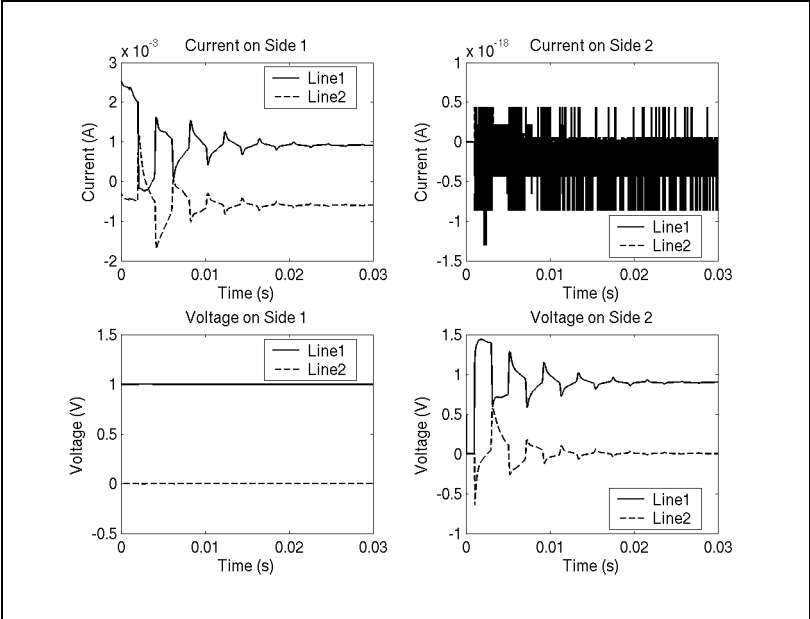


Fig. 11-20. Two conductors, step voltage, open circuit load, 0.5μs step.

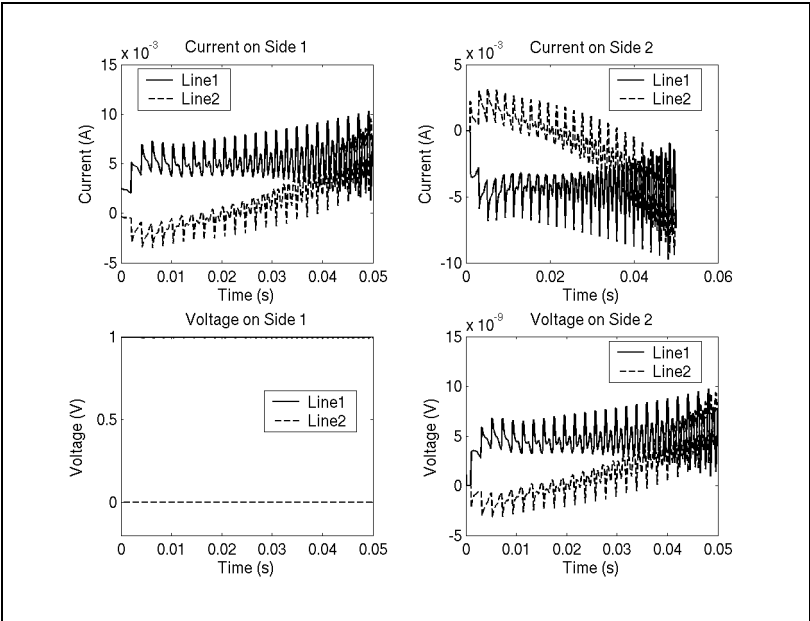
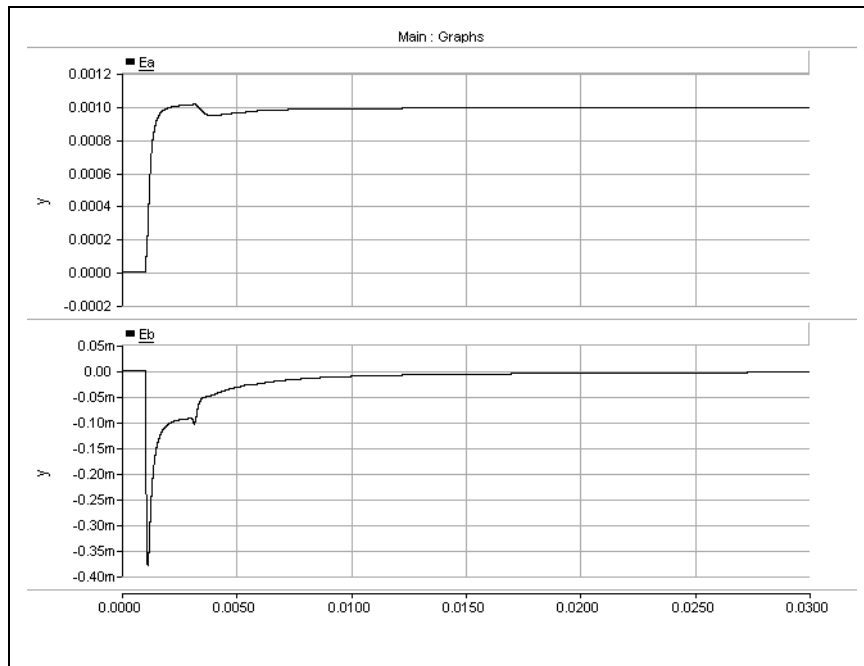


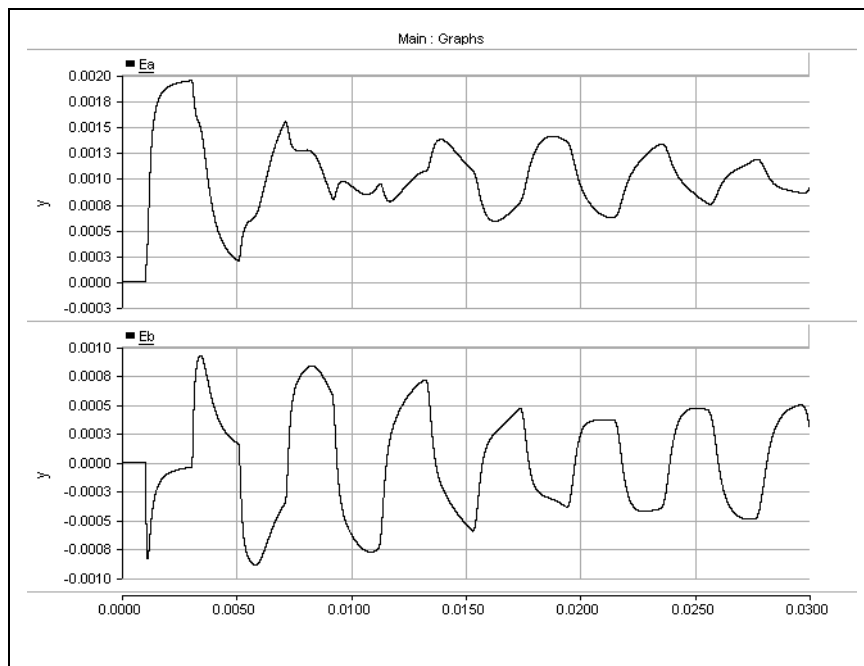
Fig. 11-21. Two conductors, step voltage, short circuit load, 0.5μs step.

Only the last result becomes unstable. A long line with a short circuit can be considered as an inductance plus a small resistance. This should have a response that increases linearly, and then levels off to a constant eventually. Perhaps numerical errors accumulated to create the instability.

Figs. 11-23, 11-23, and 11-24 show the results of the same simulations with PSCAD. The result for the short circuit case in Fig. 11-24 is stable and conforms more with what the result is expected to look like, a linear rise that tapers off eventually to a constant.



**Fig. 11-22.** Two conductors, step voltage, characteristic load,  $0.5\mu\text{s}$  step, PSCAD.



**Fig. 11-23.** Two conductors, step voltage, open circuit load,  $0.5\mu\text{s}$  step, PSCAD

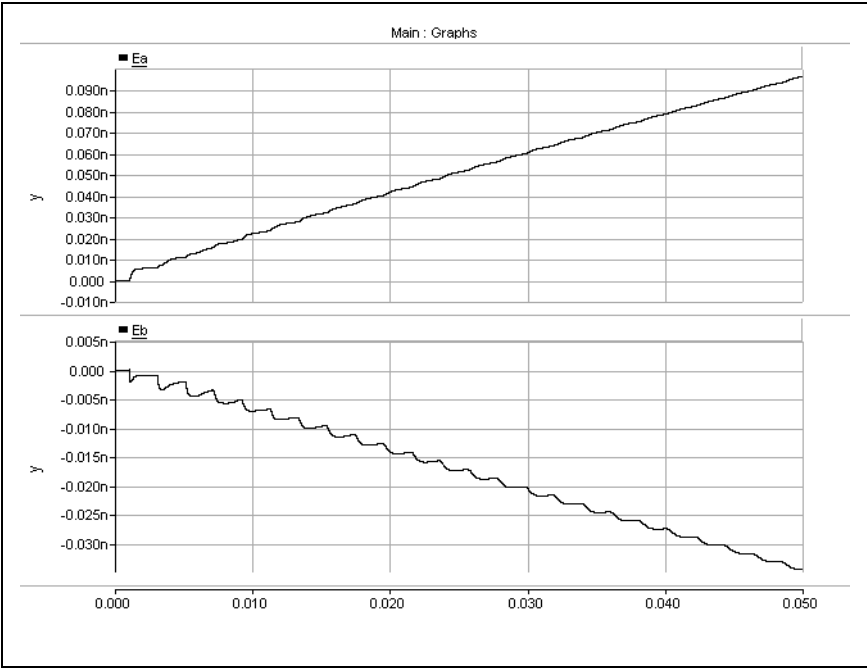


Fig. 11-24. Two conductors, step voltage, short circuit load, 0.5μs step, PSCAD.

11.2 Three-Conductor Transmission Line

Table 11-6 shows the transmission line parameters.

Table 11-6: Three-Conductor Transmission Line

$x$ (m)	$y$ (m)	Radius (mm)	$\rho$ ( $\Omega \cdot m$ )
-10	20	100	$2.8 \times 10^{-8}$
0	20	100	$2.8 \times 10^{-8}$
10	20	100	$2.8 \times 10^{-8}$

The ground resistivity is set to  $10\Omega$ , and the line length is 200km. Table 11-7 shows the fitting parameters.

Table 11-7: Fitting Parameters for Three-Conductor Line

Parameter	Value
Frequency range	$0.2 - 10^6$ Hz, 200 samples

**Table 11-7: Fitting Parameters for Three-Conductor Line**

Parameter	Value
Sample distribution	Logarithmic
Sample count	200
Fitting iterations	10
Unstable poles	Delete
Min. admittance poles	20
Min. propagation poles	10

### 11.2.1 Applied 60Hz Balanced Voltage, Characteristic Termination

A balanced 60Hz voltage was applied to side one, and side two was terminated with the real part of the characteristic admittance calculated at 60Hz. Table 11-8 shows the simulation parameters for this case.

**Table 11-8: Simulation Parameters, 60Hz, Characteristic Termination**

Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 60t) \\ \sin(2\pi 60t - 120^\circ) \\ \sin(2\pi 60t - 240^\circ) \end{bmatrix} \text{ V}$
$Y_1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ S}$
$Y_2$	$\begin{bmatrix} 0.0027630 & -7.9183 \times 10^{-4} & -4.0228 \times 10^{-4} \\ -7.9183 \times 10^{-4} & 0.0029327 & -7.9183 \times 10^{-4} \\ -4.0228 \times 10^{-4} & -7.9183 \times 10^{-4} & 0.0027630 \end{bmatrix} \text{ S}$

Figs. 11-25 and 11-26 show the simulation with a time step of 1.667ms.

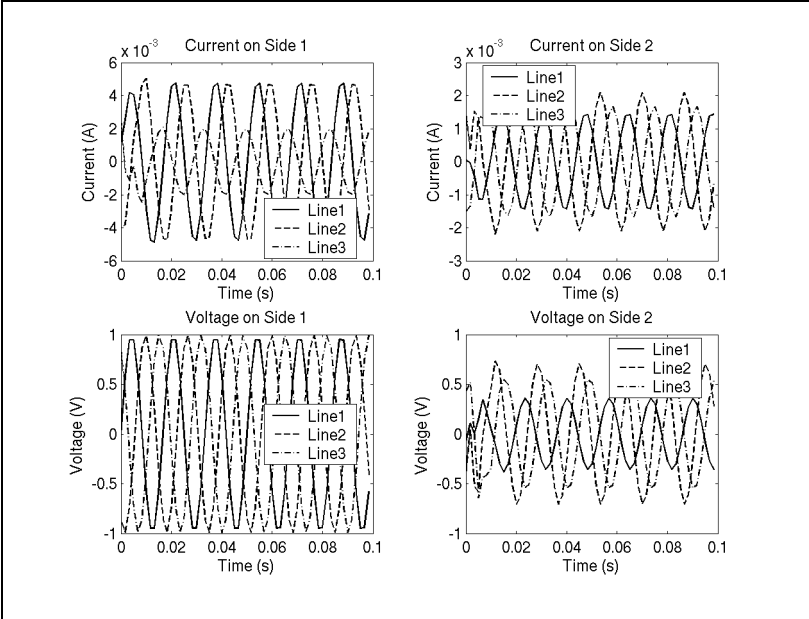


Fig. 11-25. Three conductors, 60Hz voltage, characteristic load, 1.667ms step.

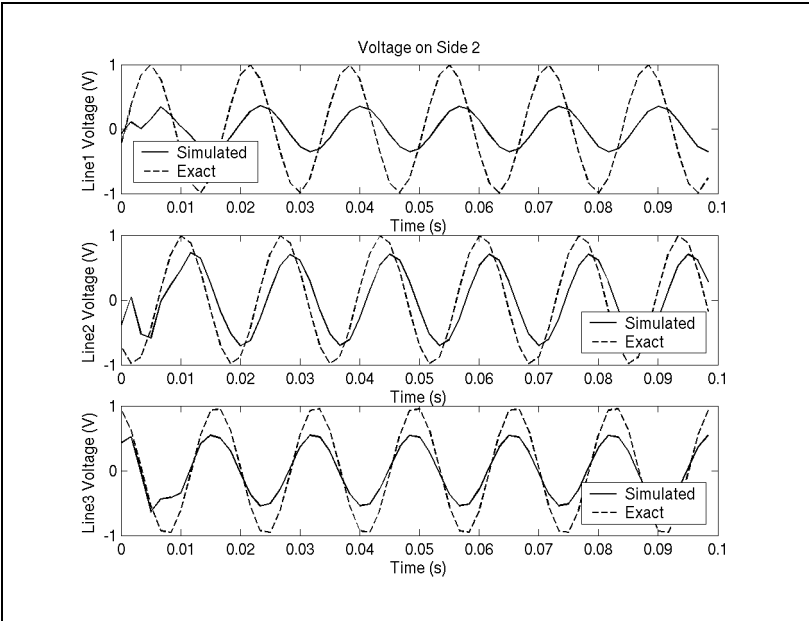
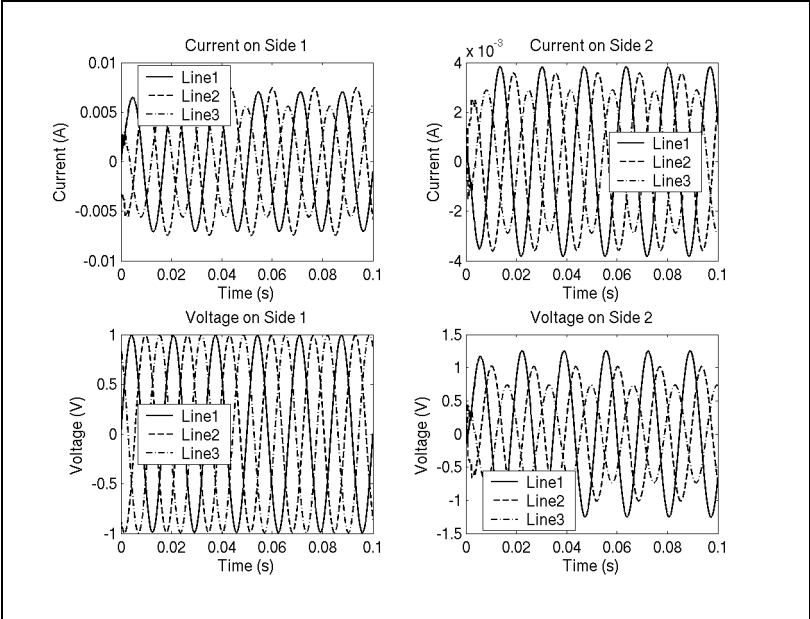


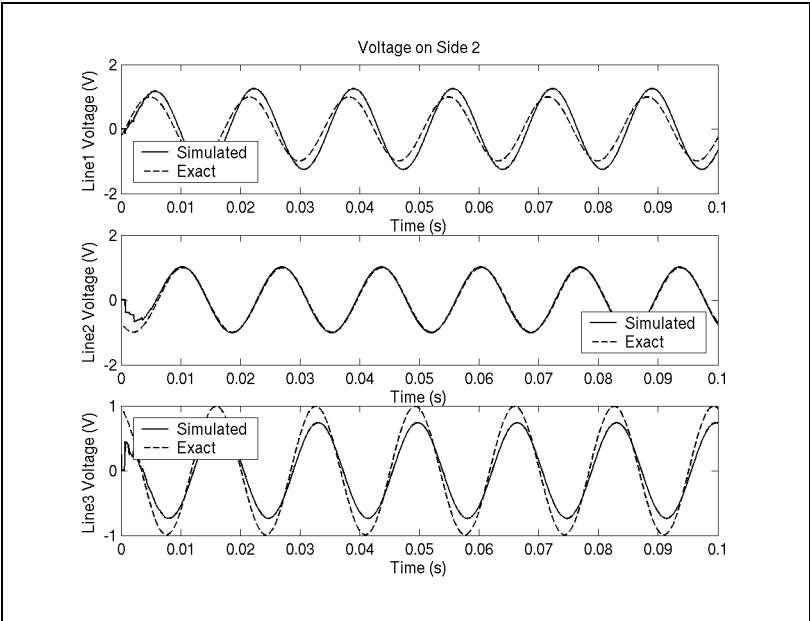
Fig. 11-26. Three conductors, 60Hz voltage, characteristic load, 1.667ms step, exact.



Figs. 11-27 and 11-28 show the simulation with a time step of  $50\mu\text{s}$ .



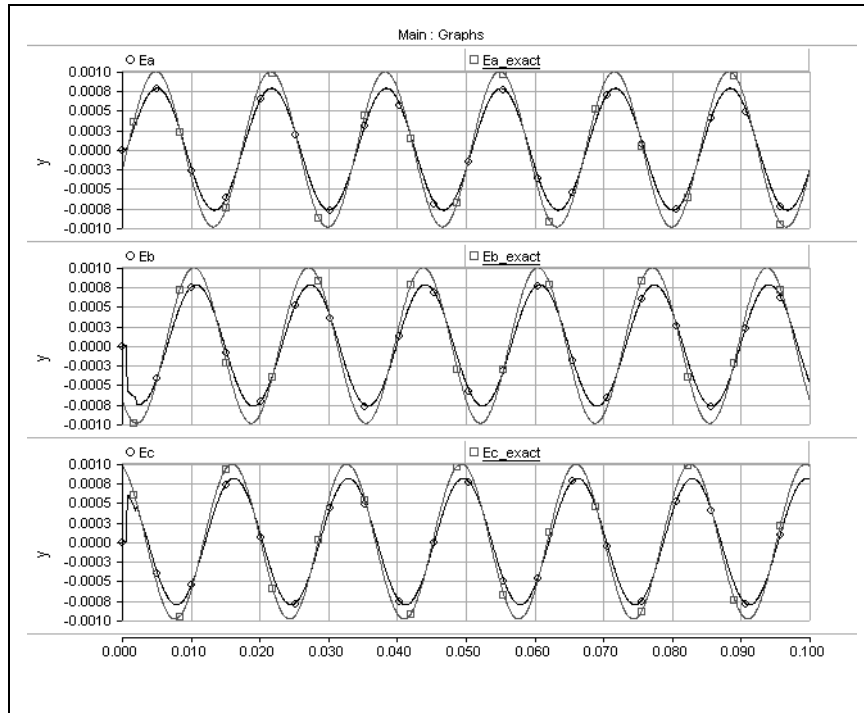
**Fig. 11-27.** Three conductors, 60Hz voltage, characteristic load,  $50\mu\text{s}$  step.



**Fig. 11-28.** Three conductors, 60Hz voltage, characteristic load,  $50\mu\text{s}$  step, exact.

The simulation results for line two match extremely closely to the exact results, but the plots show that lines one and three overshoot and undershoot the magnitude slightly.

Fig. 11-29 shows the PSCAD simulation results. None of the curves matches exactly, but the magnitudes differ less than those of Fig. 11-28.



**Fig. 11-29.** Three conductors, 60Hz voltage, characteristic load, 50 $\mu$ s step, PSCAD.

### 11.2.2 Applied 60Hz Balanced Voltage, Open Circuit Termination

A balanced 60Hz voltage was applied to side one, and side two was terminated with an open circuit. Table 11-9 shows the simulation parameters for this case.

**Table 11-9:** Simulation Parameters, 60Hz, Open Circuit Termination

Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 60t) \\ \sin(2\pi 60t - 120^\circ) \\ \sin(2\pi 60t - 240^\circ) \end{bmatrix} \text{ V}$
$Y_1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ S}$
$Y_2$	Open circuit

Figs. 11-30 and 11-31 show the simulation with a time step of 1.667ms.

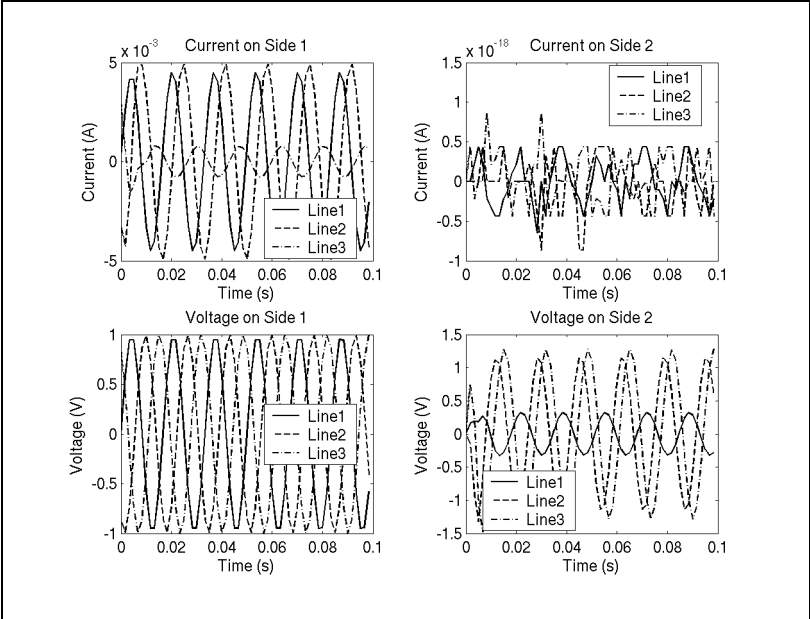


Fig. 11-30. Three conductors, 60Hz voltage, open circuit load, 1.667ms step.

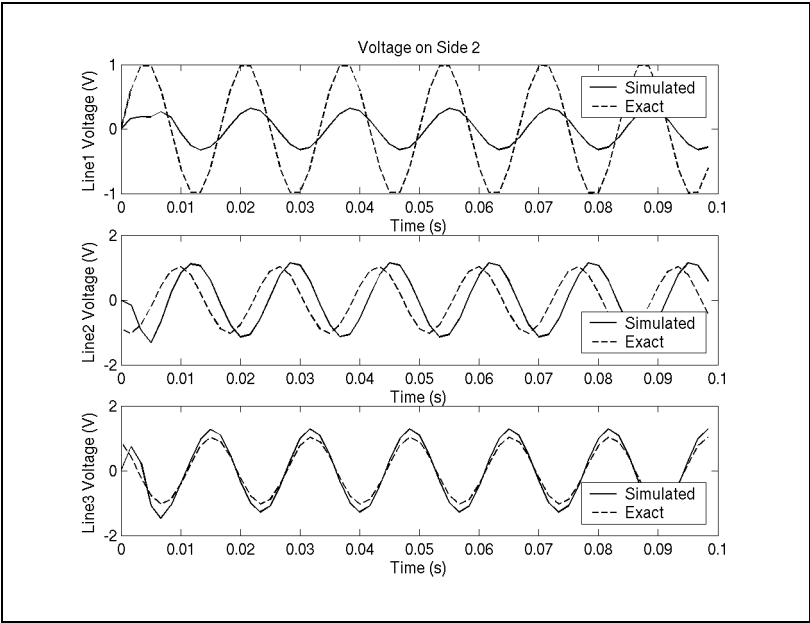
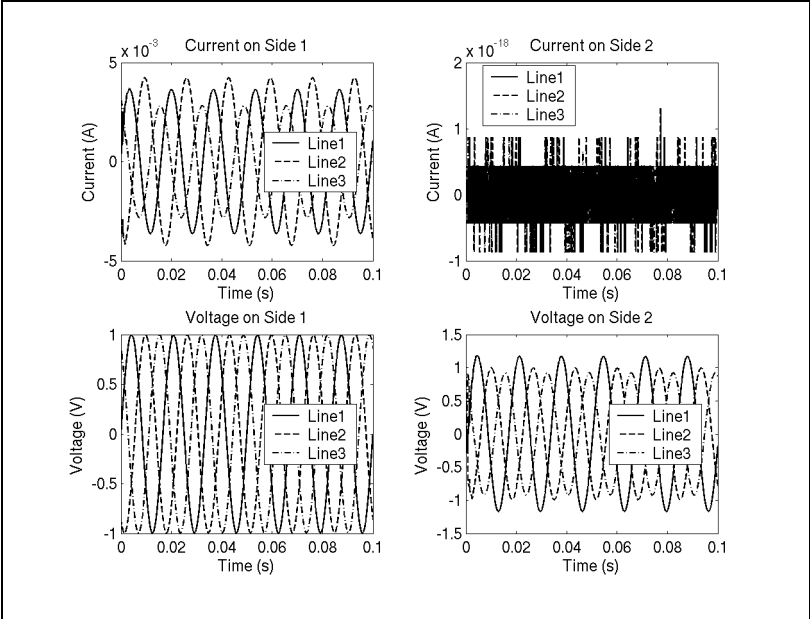
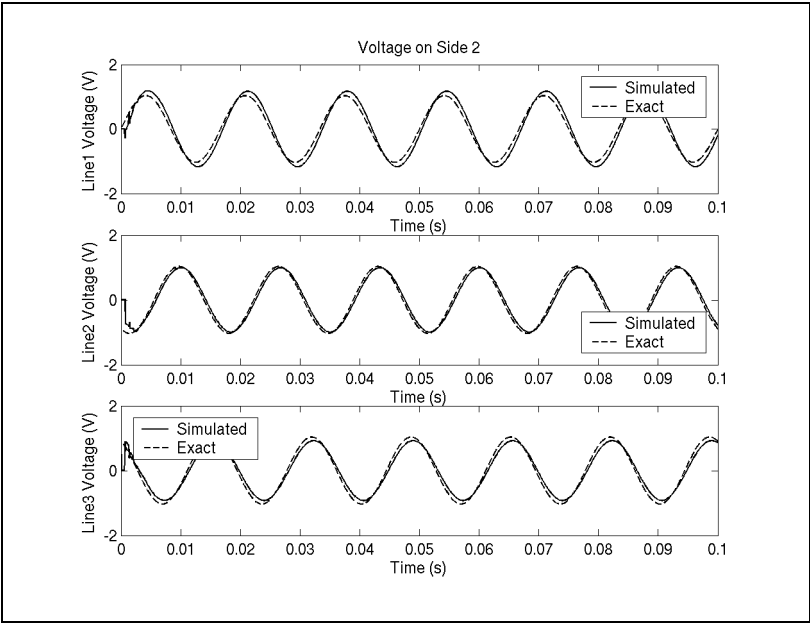


Fig. 11-31. Three conductors, 60Hz voltage, open circuit load, 1.667ms step, exact.

Figs. 11-32 and 11-33 show the simulation with a time step of  $50\mu\text{s}$ .

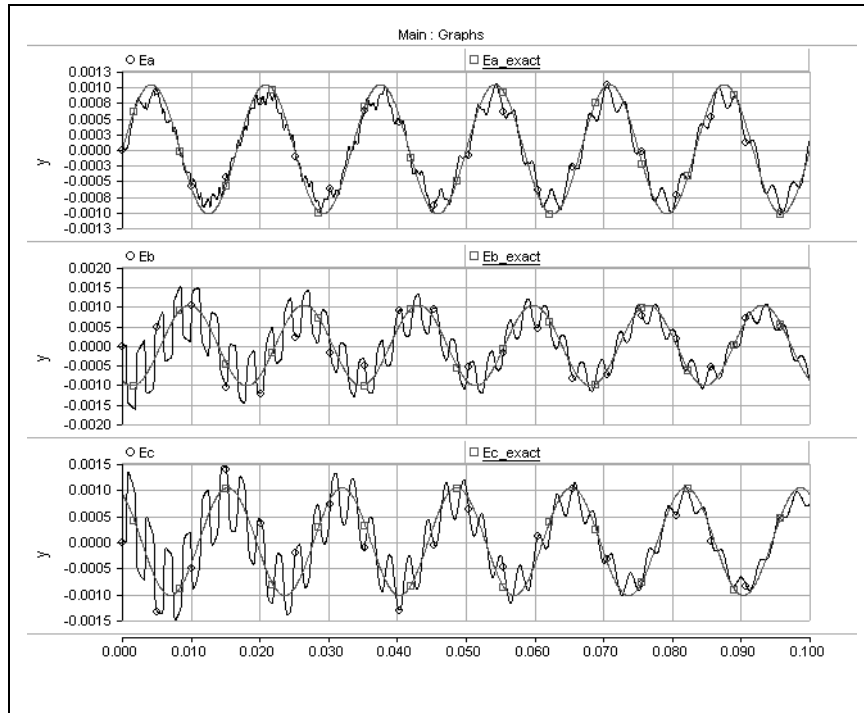


**Fig. 11-32.** Three conductors, 60Hz voltage, open circuit load,  $50\mu\text{s}$  step.



**Fig. 11-33.** Three conductors, 60Hz voltage, open circuit load,  $50\mu\text{s}$  step, exact.

These are excellent results. As before, the curves approach the steady state much more quickly than the PSCAD results of Fig. 11-34.



**Fig. 11-34.** Three conductors, 60Hz voltage, open circuit load, 50 $\mu$ s, PSCAD.

### 11.2.3 Applied 150kHz Zero-Sequence Voltage, Characteristic Termination

A zero-sequence 150kHz voltage was applied to side one, and side two was terminated with the real part of the characteristic admittance calculated at 150kHz. Table 11-10 shows the simulation parameters for this case.

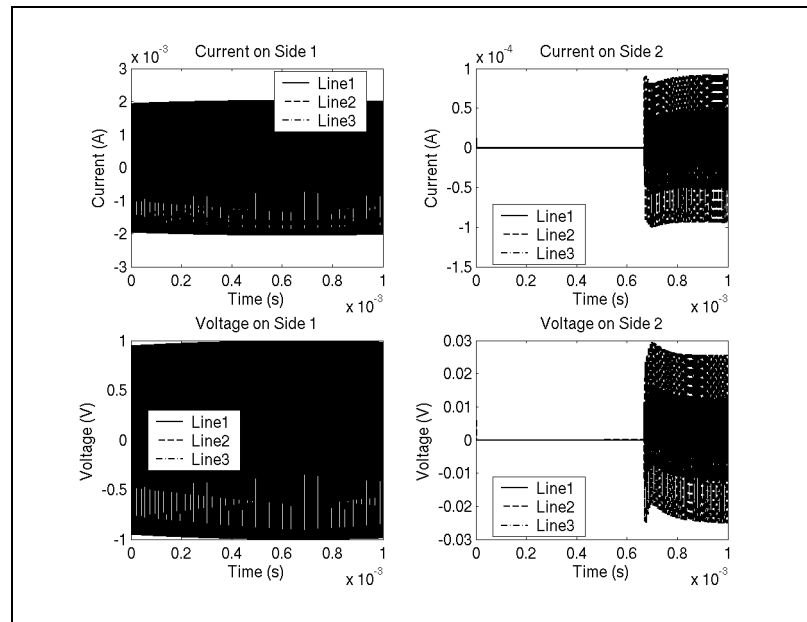
This particular case was chosen to show the effects of a communication frequency on the power lines. We expect the  $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$  mode to appear on side two because this line configuration has modes that look approximately like the Clarke components. This makes sense because this mode, according to Clarke's transformation matrix, has the highest

propagation velocity and least ground return current. Recall that the other two modes have distributions of  $[1 \ 0 \ -1]^T$  and  $[1 \ 1 \ 1]^T$ .

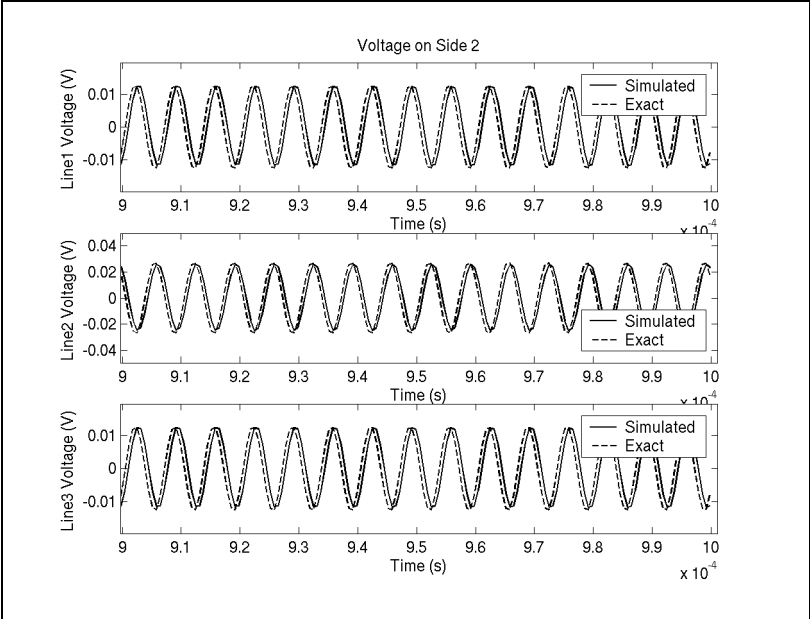
**Table 11-10:** Simulation Parameters, 150kHz, Characteristic Termination

Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 150 \times 10^3 t) \\ \sin(2\pi 150 \times 10^3 t) \\ \sin(2\pi 150 \times 10^3 t) \end{bmatrix} \text{ V}$
$Y_1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ S}$
$Y_2$	$\begin{bmatrix} 0.0029540 & -6.5439 \times 10^{-4} & -2.5500 \times 10^{-4} \\ -6.5439 \times 10^{-4} & 0.0030769 & -6.5439 \times 10^{-4} \\ -2.5500 \times 10^{-4} & -6.5439 \times 10^{-4} & 0.0029540 \end{bmatrix} \text{ S}$

Figs. 11-35 and 11-36 show the simulation with a time step of  $0.667\mu\text{s}$ .



**Fig. 11-35.** Three conductors, 150kHz voltage, characteristic load,  $0.667\mu\text{s}$  step.

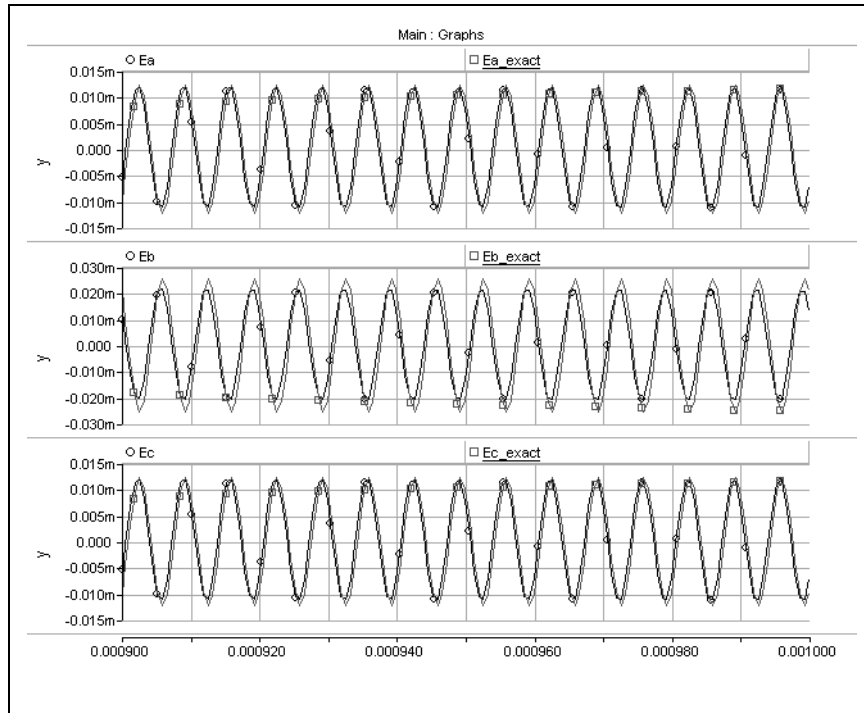


**Fig. 11-36.** Three conductors, 150kHz voltage, characteristic load, 0.667μs step, exact.

The simulated results are lagging slightly, compared with the exact results.

The time delay for the voltage and current on side two is evident in Fig. 11-35. As well, the voltage on side two has the correct  $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$  distribution.

Fig. 11-37 shows the PSCAD results, and these show the simulated results to be leading slightly, although they also match very closely.



**Fig. 11-37.** Three conductors, 150kHz voltage, characteristic load, 0.667 $\mu$ s step, PSCAD.

#### 11.2.4 Applied 150kHz Zero-Sequence Voltage, Open Circuit Termination

A zero-sequence 150kHz voltage was applied to side one, and side two was terminated with an open circuit. Table 11-11 shows the simulation parameters for this case.

We again expect the  $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$  distribution to appear on side two.

**Table 11-11:** Simulation Parameters, 150kHz, Open Circuit Termination

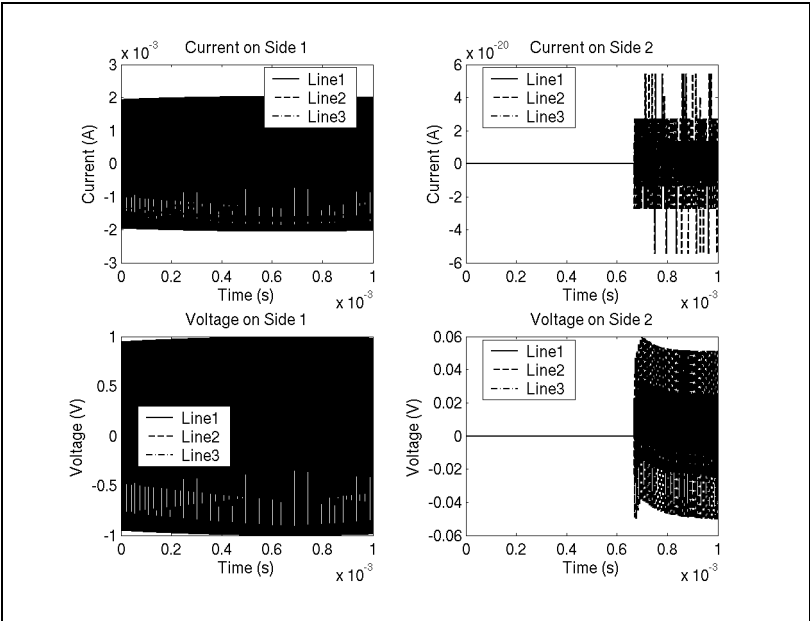
Parameter	Value
$V_s$	$\begin{bmatrix} \sin(2\pi 150 \times 10^3 t) \\ \sin(2\pi 150 \times 10^3 t) \\ \sin(2\pi 150 \times 10^3 t) \end{bmatrix} \text{V}$
$Y_1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{S}$



**Table 11-11:** Simulation Parameters, 150kHz, Open Circuit Termination

Parameter	Value
$Y_2$	Open circuit

Figs. 11-38 and 11-39 show the simulation with a time step of  $0.667\mu\text{s}$ .



**Fig. 11-38.** Three conductors, 150kHz voltage, open circuit load,  $0.667\mu\text{s}$  step.

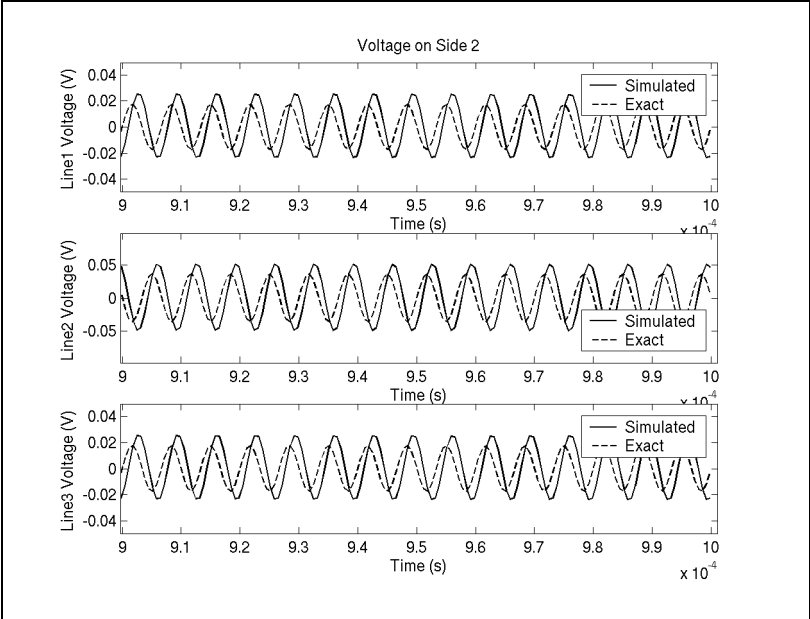


Fig. 11-39. Three conductors, 150kHz voltage, open circuit load, 0.667μs step, exact.

Fig. 11-40 shows the PSCAD results. This time, these look about the same as our results.

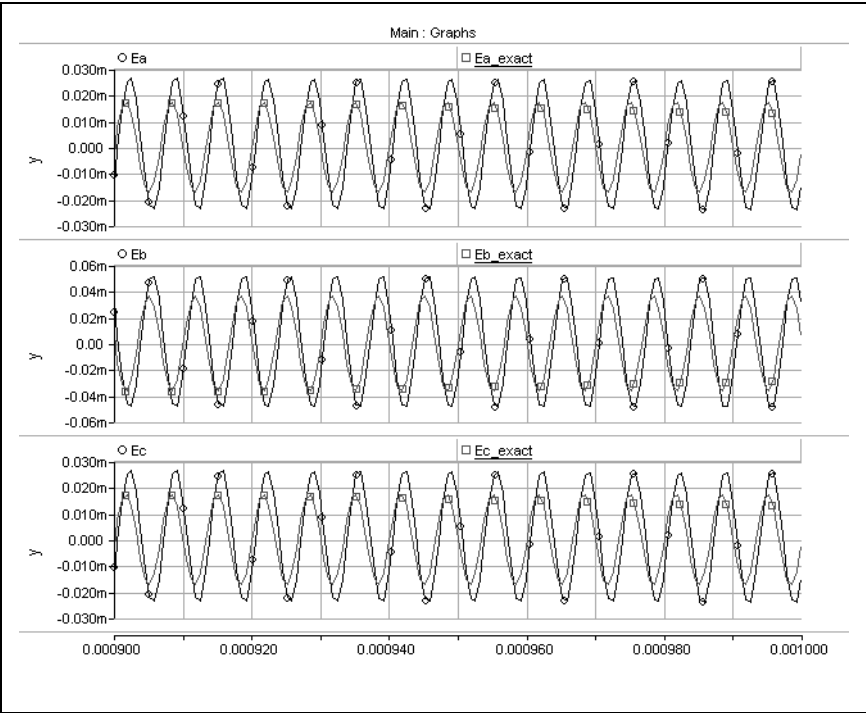


Fig. 11-40. Three conductors, 150kHz voltage, open circuit load, 0.667μs step, PSCAD.

### 11.3 Chapter Summary

Results for selected time domain simulations were shown in this chapter. The results were reasonably accurate for smaller time steps, but a time step that gives a sampling rate of ten times the frequency of interest does not appear sufficient, as the plots generated using the smaller time step show consistently better results.

Second, the phase was not always very accurate, but this could be due to an inaccurate calculated delay for each mode. Sec. 4.2.5 discussed a technique for improving the approximation.

Happily, with the exception of the example with the step voltage applied to the short circuit termination, all the simulations were stable. This unstable example, however, is an “extreme” case, though PSCAD shows a stable result. Therefore, we know that it is possible to simulate this case correctly.

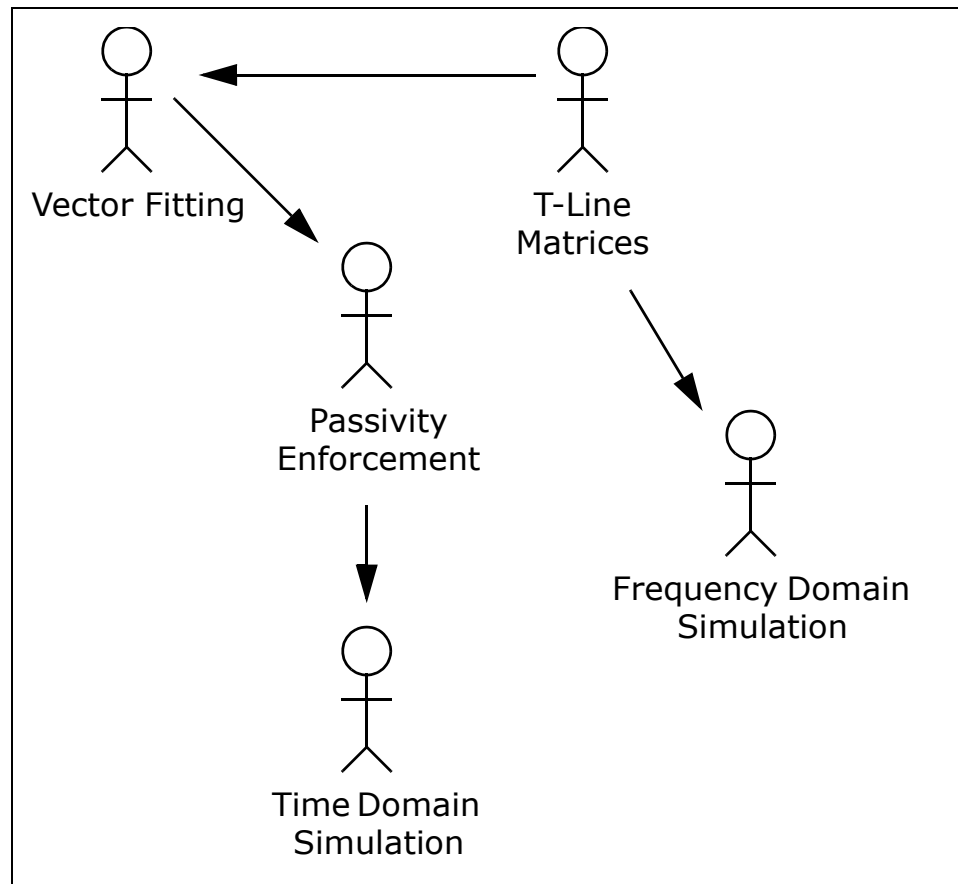
## CHAPTER XII

### NOTABLE IMPLEMENTATION DETAILS

A simulation program that produces the simulation results for this project is not trivial. It is worthwhile, then, to discuss some of the implementation details.

#### 12.1 Program Layout

Fig. 12-1 shows a brief overview of how each component in the simulator works together.



**Fig. 12-1.** Program components.

First, the transmission line matrices are calculated. These are then fitted using Vector Fitting, and the passivity of the characteristic admittance matrix is enforced. Finally, the fitted and passive matrices are passed to the time domain simulator.

#### 12.2 Why Matlab Wasn't Used

Originally, Matlab was going to be used to perform all the calculations for this thesis. It has a robust mathematical library for matrices, and it has excellent plotting capabilities.

However, it does not have the ability to construct very powerful graphical user interfaces (GUI's).

It can, however, interface with Java, a very powerful cross-platform language. At first, then, Java was going to be used to produce the GUI, and Matlab was going to be used for the calculation engine. However, this was not compatible with the longer term plans for this project, which includes the ability to have one program that can run on many different types of machines and that can run remotely, say from a web page. Plus, there is no easy way to redistribute the math engine from Matlab.

Java *is* compatible with all the longer term goals of this project and is an excellent and robust language, however it does not enjoy the availability of many free, yet robust, mathematical libraries.

Producing an independent library for this thesis enabled the author to learn much about the matrix and mathematical theory behind electromagnetic transient simulation, and also to provide a platform to independently produce simulation results.

## 12.3 Matrix Library

This project relies heavily on matrix operations. Since the program developed here is designed to be stand-alone, and since there were no freely available matrix libraries written in Java, one was developed. This section discusses the design and implementation of this library.

The primary source for the algorithms implemented here was [32]. This book not only describes the algorithms and the theory behind them, but also shows how to make them robust and efficient. It was an invaluable resource. [33] was another extremely useful resource.

### 12.3.1 Overview of the Matrix Library

Initially, the library started out as a set of operations on raw arrays for reasons of speed and memory. However, it soon became clear that it is easier to utilize a programming interface that is simpler and easier to read. In addition, the easier a program is to read and understand, the easier it is to maintain, update, and debug.

The design settled on a *Matrix* object that is both a thin wrapper over an array, and a set of operations that work on the array. The operations that were chosen to be implemented are all required for various aspects of this project, including the Vector Fitting algorithm, and for generating transmission line data.

The list of functions includes:

- **Addition**
- **Subtraction**
- **Multiplication**

- **Dot product**
- **Inverse**
- **Transpose**
- **Schur decomposition**
- **QR decomposition**
- **Square root**
- **Upper Hessenberg form**
- **Various norms**
- **Eigenvalues and eigenvectors**
- **Uniform and normal random matrices**
- **Absolute value**
- **Complex angle**
- **Matrix function evaluation**
- **Schur form**
- **Vertical and horizontal concatenation**

Each operation is designed to be used with both real and complex matrices.

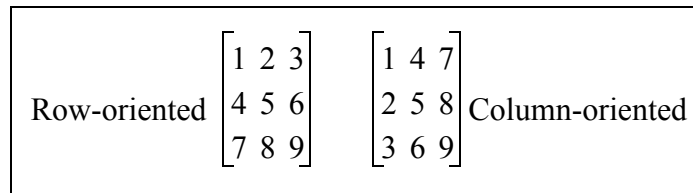
Note that, with a few exceptions, complex matrices are supported by storing the real and imaginary parts of each elements in adjacent positions in the array.

The following subsections discuss how each of these operations were implemented.

### 12.3.2 Basic Layout

As mentioned above, there is a *Matrix* object, designed to work with two-dimensional matrices, that wraps an array of double precision values. Unlike Matlab and FORTRAN, which store multidimensional arrays in column-oriented order, a row-oriented order was used. There were two reasons for this. The first is that Java is similar to C, and C stores multidimensional arrays in row-oriented order. The second reason is that it is easier to think about values row-by-row when programming the algorithms.

What is “row-oriented” and “column-oriented” you ask? Well, *row-oriented* simply means that elements in the same row are adjacent in the array, and *column-oriented* means that elements in the same column are adjacent. For example, in a 3-by-3 matrix, the third value in the array occupies position (1,3) in a row-oriented format, and position (3,1) in column-oriented format. Fig. 12-2 illustrates this.



**Fig. 12-2.** Difference between *row-oriented* and *column-oriented*.

Internally, the *Matrix* object stores the values as a one-dimensional double precision array, hence this concept is important.

Complex values are stored as adjacent real and imaginary values, so this means that a complex matrix with the same number of rows and columns as a real matrix will use twice the storage space.

Lastly, the *Matrix* object stores three other pieces of information: the number of rows, the number of columns, and a flag indicating whether it is complex.

## 12.4 Numerical Stability

Various considerations were made to help ensure the numerical stability of all the algorithms.

Calculations that involved the 2-norm, for instance, were used extensively throughout the matrix library to perform Householder decomposition, and also to find the magnitude of complex numbers.

In general, the 2-norm is calculated using Eq. (12-1):

$$\|\hat{a}\|_2 = \sqrt{\sum_i |a_i|^2} \quad (12-1)$$

This has the potential to overflow numerically even though the result may be within the range of the computer's precision. For example, this could happen if one or more elements are close to the square root of the maximum representable number.

The solution is to first divide every element by the maximum element, perform the calculation, and then multiply the result by this same element. Let  $\hat{a}_i = a_i / \|\hat{a}\|_\infty$ . Since  $|\hat{a}_i| \leq 1$ , this practically eliminates the possibility of overflow. The calculation now looks like:

$$\|\hat{a}\|_2 = \|\hat{a}\|_\infty \sqrt{\sum_i |\hat{a}_i|^2} \quad (12-2)$$

Numerically, this is identical to Eq. (12-1), but it likely won't overflow. As well, underflows are not a problem either if this formulation is used. These can only occur if the magnitude of any element is much smaller than the largest element (which equals 1), but the error incurred by ignoring these terms is below the computer's unit round-off.

Now, the Java language currently does not implement the full IEEE floating-point specification, so it is not possible to even detect overflow and underflow conditions. However, it

is still a good idea to make these considerations to help avoid the problem, which is made even worse in Java by being “silent”.

## 12.5 The Eigenvalue-Finding Routine

It is worth devoting a section to an issue with the eigenvalue-finding routine (actually, it was the Schur decomposition routine, as this is the code that really finds the eigenvalues). This issue actually exposed the fact that the Vector Fitting algorithm converges only to a local minimum.

Note that the QR algorithm is used to find the eigenvalues.

Basically, the issue reared its head when different results appeared every time a simulation was run on an 18-conductor transmission line. Since a computer program is deterministic, it should have the same output for a given input. This wasn’t the case, so naturally, the culprit was the only “random” portion of the program: the “exceptional shift” code in the Schur decomposition routine.

Briefly, this routine uses iteration to converge to a result by shifting certain elements in the matrix by a certain amount for each iteration step. However, it is possible in some cases to encounter symmetries that prevent convergence. A random number of a certain magnitude is added to the shift, and it is hoped that this will break up any of these symmetries.

Now, the random number is chosen to be of about the same magnitude as two elements of the matrix in the region under consideration. At first, a random number was chosen in the range  $[-0.5, 0.5)$ , and then this was multiplied by the minimum magnitude of local matrix elements,  $m_1$  and  $m_4$ :

$$\begin{aligned} p &= [-0.5, 0.5) \\ r &= p \cdot \min(\text{abs}(m_1), \text{abs}(m_4)) \end{aligned} \tag{12-3}$$

Michael Potter, my office mate, suggested another approach, call this “Method 2”. This was to consider a random phase, and use the formula:

$$\begin{aligned} p &= [0, 2\pi) \\ r &= \cos(p)m_1 + \sin(p)m_4 \end{aligned} \tag{12-4}$$

This way, for a given matrix, the bias towards a specific element is removed.

However, “Method 2” did not produce very good results for the following test matrices. Thus, the library settled on using the first approach.



### 12.5.1 Test Matrices for the Eigenvalue Routine

The following test matrices exercise the eigenvalue finding routine. They were taken from this mailing list: <http://www.netlib.org/na-digest/89/v89n49>.

The eigenvalues of  $\begin{bmatrix} 0 & 0 & 0 & -4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ , the characteristic polynomial of  $(x^2 - 2)^2$ , were calculated

as  $\begin{bmatrix} -1.4142135 \\ 1.4142135 \\ 1.4142136 \\ -1.4142136 \end{bmatrix}$ . The actual values are  $\pm\sqrt{2} = \pm 1.4142136$ .

The eigenvalues of  $\begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & -4 \end{bmatrix}$ , the characteristic polynomial of  $(x + 1)^4$ , were calculated as

$\begin{bmatrix} -0.99987278 - 1.2720304 \times 10^{-4}i \\ -0.99987278 + 1.2720304 \times 10^{-4}i \\ -1.0001272 - 1.2723978 \times 10^{-4}i \\ -1.0001272 + 1.2723978 \times 10^{-4}i \end{bmatrix}$ . The actual values are  $-1$ .

The eigenvalues of  $\begin{bmatrix} 0 & 0 & 0 & -16 \\ 1 & 0 & 0 & -32 \\ 0 & 1 & 0 & -24 \\ 0 & 0 & 1 & -8 \end{bmatrix}$ , the characteristic polynomial of  $(x + 2)^4$ , were calculated

as  $\begin{bmatrix} -1.999739 - 2.6092529 \times 10^{-4}i \\ -1.999739 + 2.6092529 \times 10^{-4}i \\ -2.000261 - 2.609851 \times 10^{-4}i \\ -2.000261 + 2.609851 \times 10^{-4}i \end{bmatrix}$ . The actual values are  $-2$ .

Matlab produces comparable results for all three examples.

## 12.6 Chapter Summary

This chapter discussed a few notable implementation details.

First, a complete library for working with matrices was created for this project. It contains excellent implementations for all the project's required algorithms.

Second, stability and numerical precision were taken into account when implementing the functionality.

Last, properly finding eigenvalues is tricky. Due to a problematic symmetry issue, some randomness was introduced to break these cycles. This in turn exposed that the Vector Fitting algorithm converges only to local minima.

## CHAPTER XIII

# CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIONS

The Vector Fitting algorithm and associated frequency and time domain techniques are very useful tools for simulating an overhead transmission line. Using these tools, it has been shown that reasonably accurate simulations can be achieved.

Frequency domain fitting for overhead transmission line models was studied. Vector Fitting produced accurate results that were used in the time domain section, and passivity of the model was enforced.

Next, the fitted frequency domain functions were expressed in the time domain using a technique called recursive convolution, and simulation results were produced. The results were excellent, but they were not identical to PSCAD results. In one case, where a step voltage was applied to a line terminated with a short circuit, the results became unstable. PSCAD showed a stable result for this case.

It was not known why the instability occurred, as the fitted characteristic admittance matrix was known to be passive. Perhaps there are other criteria, having to do with the propagation matrix, that affect passivity.

Third, a platform independent program was created in Java to accomplish these tasks. What was not completed, however, was a graphical interface to this tool. Fig. 13-1 is an actual screenshot of the partial implementation.

Last, a complete description and derivation of how to get from the theory to a practical implementation has been shown. The main development exists in each of the chapters, while the appendices show the necessary proofs. It is hoped that this thesis is a valuable resource for those wishing to explore the theory of transmission line simulation.

The next few sections expand on some specific conclusions and recommendations.

### **13.1 Vector Fitting**

The Vector Fitting algorithm has shown that it can produce very good fitting results. However, it converges to a local minimum that depends on the starting values and the eigenvalue routine.

It would be worthwhile to investigate a technique other than the least-squares approach for use in the algorithm. For example, various optimization methods may be able to find a global minimum or a better local minimum.

Also, different procedures to alter the frequency sample density need to be explored. For example, a linear or logarithmic frequency distribution may not be the best way. A non-linear distribution may achieve superior results.

### **13.2 Passivity Enforcement**

It is possible to force the characteristic admittance matrix to be passive using several techniques. However, there was one simulation that produced unstable results. Perhaps there is an passivity enforcement mechanism for the propagation matrix as well. This should be explored.

Although simulating just the characteristic admittance always gave excellent results, it would be instructive to allow unstable poles during fitting, and then force it to be passive to see the effects on simulation.

### **13.3 Time Domain Simulation**

There are many parameters throughout the whole fitting and modelling process that affect the time domain simulation results. There is probably some optimal set of parameters that produces both accurate and stable results.

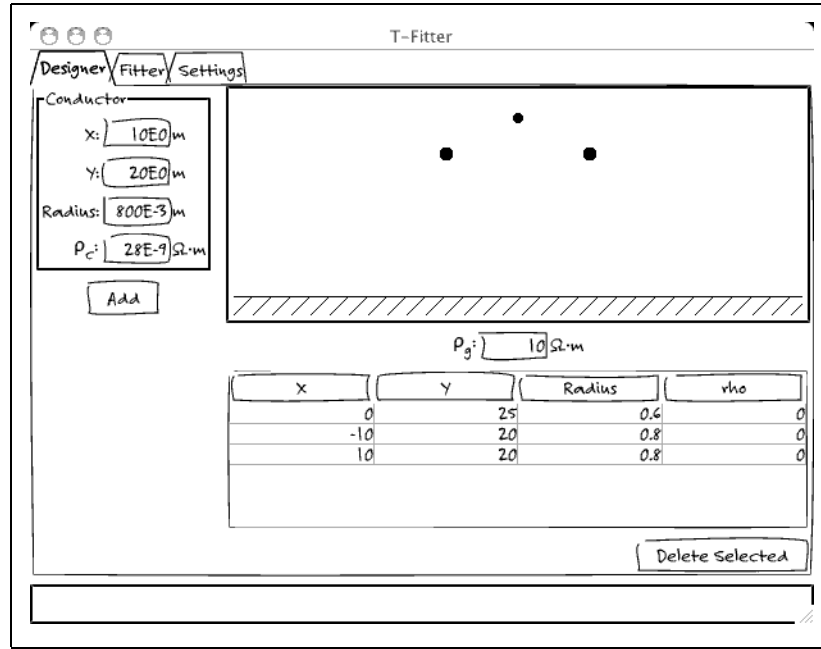
For example, the fitting order, fitting iteration count, and the time delay extraction procedure all affect the accuracy and stability of the simulation.

### **13.4 Direct Z-Domain Formulation**

The final step in the time domain simulation utilizes constants, present samples, and previous samples. It may be possible to formulate the transmission line model directly in the z-domain. One problem with this is that the model would change if the time step changes, and this is not the case when using the s-domain.

### 13.5 A Nice Looking Simulator

Work has been done on creating a nice looking graphical interface for the simulator software developed here. It was not completed, but Fig. 13-1 shows an actual screenshot of the work in progress.



**Fig. 13-1.** Screenshot of the graphical transmission line designer.

## REFERENCES

- [1] C. Gary, “Approche complète de la propagation multifilaire en haute fréquence par utilisation des matrices complexes”, EDF Bulletin de la Direction des Études et Recherches, Série B-Réseaux Électriques Matériels Électriques, 3/4, 1976, pp. 5-20.
- [2] A. Deri, G. Tevan, A. Semlyen, and A. Castanheira, “The complex ground return plane: a simplified model for homogeneous and multilayer earth return”, IEEE Transactions on Power Apparatus and Systems, Vol. 100, No. 8, 1981, pp. 3686-3693.
- [3] L. M. Wedepohl, “The Theory of Natural Modes in Multiconductor Transmission Systems”, Graduate Course, University of Manitoba, September 2002.
- [4] F. A. Uribe, J. L. Naredo, P. Monero, L. Guardado, “Calculating Earth Impedances for Underground Transmission Cables”, Paper 027, International Conference on Power Systems Transients, June 2001, Rio de Janeiro, Brazil.
- [5] A. B. Fernandes, W. L. A. Neves, E. G. Costa, and M. N. Cavalcanti, “The Effect of Shunt Conductance on Transmission Line Models”, Paper 152, International Conference on Power Systems Transients, June 2001, Rio de Janeiro, Brazil.
- [6] Y. Wang and S. Liu, “A Review of Methods for Calculation of Frequency-Dependent Impedance of Overhead Power Transmission Lines”, Proc. Natl. Sci. Counc. ROC(A), Vol. 25, No. 6, 2001, pp. 329-338.
- [7] A. Semlyen and A. Dabuleanu, “Fast and Accurate Switching Transient Calculations on Transmission Lines with Ground Return using Recursive Convolutions”, IEEE Transactions on Power Apparatus and Systems, Vol. 94, March/April 1975, pp. 561-571.
- [8] H. S. Wilf, “A Stability Criterion for Numerical Integration”, Journal of the ACM, Vol. 6, No. 3, July 1959, pp. 363-365.
- [9] H.W. Dommel, “Digital Computer Solution of Electromagnetic Transients in Single and Multiphase Networks”, IEEE Transactions on Power Apparatus and Systems, Vol. 88, No. 4, April 1969.
- [10] A. Morched, B. Gustavsen, and M. Tartibi, “A Universal Model for Accurate Calculation of Electromagnetic Transients on Overhead Lines and Cables”, IEEE Transactions on Power Delivery, Vol. 14, No. 3, July 1999, pp. 1032-1038.
- [11] B. Gustavsen, G. Irwin, R. Mangelrød, D. Brandt, K. Kent, “Transmission Line Models for the Simulation of Interaction Phenomena between Parallel AC and DC Overhead Lines”, Proceedings of the 1999 International Conference on Power Systems Transients, June 20-24, Budapest, Hungary, pp. 61-67.

- [12]L.M. Wedepohl, H. V. Nguyen, and G. D. Irwin, “Frequency-Dependent Transformation Matrices for Untransposed Transmission Lines using Newton-Raphson Method”, IEEE Transactions on Power Systems, Vol. 11, No. 3, August 1996, pp. 1538-1546.
- [13]T.T. Nguyen and H. Y. Chan, “Evaluation of Modal Transformation Matrices for Overhead Transmission Lines and Underground Cables by Optimization Method”, IEEE Transactions on Power Delivery, Vol. 17, No. 1, January 2002, pp. 200-209.
- [14]B. Gustavsen and A. Semlyen, “Simulation of Transmission Line Transients Using Vector Fitting and Modal Decomposition”, IEEE Transactions on Power Delivery, Vol. 13, No. 2, April 1998, pp. 605-614.
- [15]H. W. Bode, *Network Analysis and Feedback Amplifier Design*, D. Van Nostrand & Company, Inc., New York, NY, 1945. (551 pp. 621.385 B631 Ne)
- [16]B. Gustavsen, “Time Delay Identification for Transmission Line Modeling”, 8th IEEE Workshop on Signal Propagation on Interconnects, Proceedings, 2004, pp. 103-106.
- [17]N. Balabanian and W. R. LePage, “What Is a Minimum Phase Network?”, AIEE Basic Sciences Paper, Presented at AIEE Fall General Meeting, Chicago, Illinois, Oct. 3-7, 1955 (publication is January 1956, pp. 785-788).
- [18]A. O. Soysul and A. Semlyen, “Practical Transfer Function Estimation and its Application to Wide Frequency Range Representation of Transformers”, IEEE Transactions on Power Delivery, Vol. 8, No. 3, July 1993, pp. 1627-1637.
- [19]A. Dounavis, N. Nakhla, R. Achar, M. Nakhla, “Delay Extraction and Passive Macro-modeling of Lossy Coupled Transmission Lines”, Electrical Performance of Electronic Packaging, October 2003, pp. 251-254.
- [20]B. Gustavsen and A. Semlyen, “Rational Approximation of Frequency Domain Responses by Vector Fitting”, IEEE Transactions on Power Delivery, Vol. 14, No. 3, July 1999, pp. 1052-1061.
- [21]A. H. Whitfield, “Transfer Function Synthesis Using Frequency Response Data”, Int. Journal of Control, Vol. 43, No. 5, 1986, pp. 1413-1426.
- [22]C. Merkwirth, U. Parlitz, and W. Lauterborn, “Fast Nearest-Neighbor Searching for Nonlinear Signal Processing”, Physical Review E, Vol. 62, No. 2, August 2000, pp. 2089-2097.
- [23]B. Gustavsen and A. Semlyen, “A Robust Approach for System Identification in the Frequency Domain”, IEEE Transactions on Power Delivery, Vol. 19, No. 3, July 2004, pp. 1167-1173.

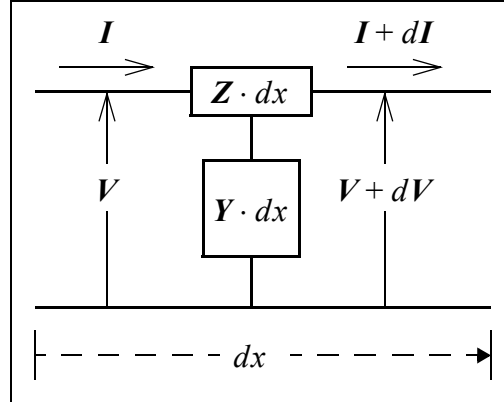
- [24]B. Gustavsen and A. Semlyen, “Enforcing Passivity for Admittance Matrices Approximated by Rational Functions”, IEEE Transactions on Power Delivery, Vol. 16, No. 1, February 2001, pp. 97-104.
- [25]F. A. M. Cipparrone, B. Gustavsen, A. Semlyen, A. Feijoo, and J. Cidras, “Discussion of ‘Enforcing Passivity for Admittance Matrices Approximated by Rational Functions’”, IEEE Transactions on Power Systems, Vol. 16, No. 4, November 2001, pp. 954-955.
- [26]B. Gustavsen and A. Semlyen, “Closure to Discussion of ‘Enforcing Passivity for Admittance Matrices Approximated by Rational Functions’”, IEEE Transactions on Power Systems, Vol. 16, No. 4, November 2001, p. 955.
- [27]D. Saraswat, R. Achar, and M. Nakhla, “Enforcing Passivity for Rational Function Based Macromodels of Tabulated Data”, Electrical Performance of Electronic Packaging, October 2003, pp. 295-298.
- [28]R. W. Freund, “Passive Reduced-Order Modeling via Krylov-Subspace Methods”, Proceedings of the 14th Symposium on Mathematical Theory of Networks and Systems, Perpignan France, June 2000.
- [29]B. D. O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [30]M. R. Wohlers, *Lumped and Distributed Passive Networks*, Academic Press, New York, 1969.
- [31]G. Angelidis and A. Semlyen, “Direct Phase-Domain Calculation of Transmission Line Transients Using Two-Sided Recursions”, IEEE Transactions on Power Delivery, Vol. 10, No. 2, April 1995, pp. 941-949.
- [32]D. S. Watkins, *Fundamentals of Matrix Computations*, 2nd ed., Wiley-Interscience, New York, NY, 2002. (618 pp. ISBN 0-471-21394-2)
- [33]H. Eves, *Elementary Matrix Theory*, Dover, New York, NY, 1980. (325 pp. ISBN 0-486-63946-0)
- [34]K. Ogata, *Modern Control Engineering*, Prentice Hall, 1970.
- [35]G. Irwin, Electranix Corporation, Winnipeg, Manitoba, private communication, 2005.



## APPENDIX A

### TRANSMISSION LINE EQUATIONS

This section shows the derivation for the basic multi-conductor transmission line equations. All variables, with the exception of  $x$  are matrices.



**Fig. A-1.** ‘Small’ transmission line segment.

Let us imagine a very small segment of transmission line with length  $dx$ . Since the line has impedance per unit length,  $Z$ , and admittance per unit length,  $Y$ , the segment has a total impedance equal to  $Z \cdot dx$  and a total admittance equal to  $Y \cdot dx$ . Fig. A-1 shows a representation of this scenario.

Let us say that the voltage across the wires is  $V$ , and the current flowing through the segment is  $I$ . We have the two equations:

$$\begin{aligned}
 dV &= -Zdx \cdot I & \frac{dV}{dx} &= -Z \cdot I \\
 dI &= -Ydx \cdot V & \text{OR} & \\
 & & \frac{dI}{dx} &= -Y \cdot V
 \end{aligned}
 \tag{A-1}$$

Substituting  $I = -Z^{-1} \frac{dV}{dx}$  into the second equation and  $V = -Y^{-1} \frac{dI}{dx}$  into the first, we derive:

$$\begin{aligned}\frac{d^2 \mathbf{V}}{dx^2} &= \mathbf{ZY} \cdot \mathbf{V} \\ \frac{d^2 \mathbf{I}}{dx^2} &= \mathbf{YZ} \cdot \mathbf{I}\end{aligned}\tag{A-2}$$

### A.1 Direct Solution for Voltage

These equations have solutions of the form

$$\begin{aligned}\mathbf{V} &= e^{-\sqrt{\mathbf{ZY}}x} \mathbf{V}_a + e^{\sqrt{\mathbf{ZY}}x} \mathbf{V}_b \\ \mathbf{I} &= \mathbf{Z}^{-1} \sqrt{\mathbf{ZY}} (e^{-\sqrt{\mathbf{ZY}}x} \mathbf{V}_a - e^{\sqrt{\mathbf{ZY}}x} \mathbf{V}_b)\end{aligned}\tag{A-3}$$

where  $\mathbf{V}_a$  and  $\mathbf{V}_b$  are forward- and backward-travelling voltage waves, respectively.

Next, we apply the boundary conditions at  $x = 0$ , and solve for  $\mathbf{V}_a$  and  $\mathbf{V}_b$ . Assume that some source voltage,  $\mathbf{V}_s$ , and source current,  $\mathbf{I}_s$ , flowing into the line, are applied at this location:

$$\begin{aligned}\mathbf{V}_a &= \frac{\mathbf{V}_s + \mathbf{Z}_0 \mathbf{I}_s}{2} \\ \mathbf{V}_b &= \frac{\mathbf{V}_s - \mathbf{Z}_0 \mathbf{I}_s}{2}\end{aligned}\tag{A-4}$$

where  $\mathbf{Z}_0 = \Gamma^{-1} \mathbf{Z} = \mathbf{Z}(\Gamma^T)^{-1}$  is known as the *characteristic impedance* of the line, and  $\Gamma = \sqrt{\mathbf{ZY}}$  is called the *propagation constant*. Additionally,  $\mathbf{Y}_0 = \mathbf{Z}^{-1} \Gamma$  is known as the *characteristic admittance* of the line, and is the inverse of  $\mathbf{Z}_0$ .

### A.2 Direct Solution for Current

An alternative solution to the differential equations can be expressed in terms of current waves instead of voltage waves. This has the form

$$\begin{aligned}\mathbf{I} &= (e^{-\sqrt{\mathbf{ZY}}x})^T \mathbf{I}_a - (e^{\sqrt{\mathbf{ZY}}x})^T \mathbf{I}_b \\ \mathbf{V} &= \mathbf{Y}^{-1} (\sqrt{\mathbf{ZY}})^T ((e^{-\sqrt{\mathbf{ZY}}x})^T \mathbf{I}_a + (e^{\sqrt{\mathbf{ZY}}x})^T \mathbf{I}_b)\end{aligned}\tag{A-5}$$

where  $I_a$  and  $I_b$  are forward- and backward-travelling waves, respectively. Notice that the backward-travelling wave,  $I_b$ , is defined in the opposite direction as  $V_b$ . This is because it makes more intuitive sense, as currents are generally considered to have a direction, and  $I_b$  is moving in the opposite direction as  $I_s$ .

Apply the boundary conditions at  $x = 0$  in a manner similar to the steps above to solve for  $I_a$  and  $I_b$ :

$$\begin{aligned} I_a &= \frac{Y_0 V_s + I_s}{2} \\ I_b &= \frac{Y_0 V_s - I_s}{2} \end{aligned} \quad (\text{A-6})$$

where  $Y_0 = (\Gamma^T)^{-1} Y = Y\Gamma^{-1}$  is known as the *characteristic admittance* of the line, and  $\Gamma^T = (\sqrt{ZY})^T$ . It can be shown that this definition of  $Y_0$  is equal to the definition from the previous section. Note that these solutions are consistent with the solutions derived using the voltage waves.

It can also be shown that due to certain properties of the  $Z$  and  $Y$  matrices,

$$(\sqrt{ZY})^T = \sqrt{YZ}.$$

### A.3 Two-Port Equations

Relations between the source quantities and the quantities at some position,  $x = l$ , can be found by examining the boundary conditions at both locations. The values for voltage and current have already been found at the source,  $x = 0$ . Altering our definition of current slightly so that it has a positive value going *into* the line at  $x = l$ , we arrive at:

$$\begin{aligned} V(l) &= e^{-\Gamma l} \frac{V_s}{2} + e^{-\Gamma l} \frac{Z_0 I_s}{2} + e^{\Gamma l} \frac{V_s}{2} - e^{\Gamma l} \frac{Z_0 I_s}{2} \\ &= \cosh(\Gamma l) V_s - \sinh(\Gamma l) Z_0 I_s \\ I(l) &= -Y_0 e^{-\Gamma l} \frac{V_s}{2} - Y_0 e^{-\Gamma l} \frac{Z_0 I_s}{2} + Y_0 e^{\Gamma l} \frac{V_s}{2} - Y_0 e^{\Gamma l} \frac{Z_0 I_s}{2} \\ &= Y_0 \sinh(\Gamma l) V_s - Y_0 \cosh(\Gamma l) Z_0 I_s \end{aligned} \quad (\text{A-7})$$

and finally, the two-port relation:

$$\begin{bmatrix} V_l \\ I_l \end{bmatrix} = \begin{bmatrix} \cosh(\Gamma l) & -\sinh(\Gamma l)\mathbf{Z}_0 \\ \mathbf{Y}_0 \sinh(\Gamma l) & -\mathbf{Y}_0 \cosh(\Gamma l)\mathbf{Z}_0 \end{bmatrix} \begin{bmatrix} V_s \\ I_s \end{bmatrix} \quad (\text{A-8})$$

Because  $\mathbf{Y}_0 \cosh(\Gamma l) = \cosh(\Gamma l)^T \mathbf{Y}_0$ , the above relation can be simplified to:

$$\begin{bmatrix} V_l \\ I_l \end{bmatrix} = \begin{bmatrix} \cosh(\Gamma l) & -\sinh(\Gamma l)\mathbf{Z}_0 \\ \mathbf{Y}_0 \sinh(\Gamma l) & -\cosh(\Gamma l)^T \end{bmatrix} \begin{bmatrix} V_s \\ I_s \end{bmatrix} \quad (\text{A-9})$$

With some reshuffling, this relationship can be reformed to relate all possible pairs of input and output voltage and current. For example:

$$\begin{bmatrix} I_s \\ I_l \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_0 \coth(\Gamma l) & -\mathbf{Y}_0 \operatorname{csch}(\Gamma l) \\ -\mathbf{Y}_0 \operatorname{csch}(\Gamma l) & \mathbf{Y}_0 \coth(\Gamma l) \end{bmatrix} \begin{bmatrix} V_s \\ V_l \end{bmatrix} \quad (\text{A-10})$$

and

$$\begin{bmatrix} I_s \\ I_l \end{bmatrix} = \begin{bmatrix} \coth(\Gamma l)\mathbf{Z}_0 & \operatorname{csch}(\Gamma l)\mathbf{Z}_0 \\ \operatorname{csch}(\Gamma l)\mathbf{Z}_0 & \coth(\Gamma l)\mathbf{Z}_0 \end{bmatrix} \begin{bmatrix} V_s \\ V_l \end{bmatrix} \quad (\text{A-11})$$

#### A.4 The $Z$ and $Y$ Parameters

The  $Z$  and  $Y$  parameters are the impedance and admittance per unit length, respectively. These can be derived using the techniques in [1] and [2].

One important relationship between them is:

$$\mathbf{ZY} = (\mathbf{YZ})^T \quad (\text{A-12})$$

#### A.5 Useful Proofs

This section presents some useful proofs relevant to transmission line equations.

$$\boxed{\mathbf{Y}_0 = \mathbf{Y}_0^T}$$

$$\begin{aligned}
\mathbf{Y}_0 &= \mathbf{Z}^{-1}\Gamma \\
&= \mathbf{E}_I \mathbf{Z}'^{-1} \mathbf{E}_V^{-1} \mathbf{E}_V \gamma \mathbf{E}_V^{-1} \\
&= \mathbf{E}_I \mathbf{Z}'^{-1} \gamma \mathbf{E}_V^{-1} \\
&= \mathbf{E}_I \gamma \mathbf{Z}'^{-1} \mathbf{E}_V^{-1} \\
&= \mathbf{E}_I \gamma \mathbf{E}_I^{-1} \mathbf{E}_I \mathbf{Z}'^{-1} \mathbf{E}_V^{-1} \\
&= \Gamma^T \mathbf{Z}^{-1} \\
&= \Gamma^T (\mathbf{Z}^{-1})^T \\
&= \mathbf{Y}_0^T
\end{aligned} \tag{A-13}$$

$$\boxed{\mathbf{Y}_0 \cosh(\Gamma l) = \cosh(\Gamma l)^T \mathbf{Y}_0}$$

$$\begin{aligned}
\mathbf{Y}_0 \cosh(\Gamma l) &= \mathbf{Z}^{-1} \Gamma \cosh(\Gamma l) \\
&= \mathbf{E}_I \mathbf{Z}'^{-1} \mathbf{E}_V^{-1} \mathbf{E}_V \gamma \mathbf{E}_V^{-1} \mathbf{E}_V \cosh(\gamma l) \mathbf{E}_V^{-1} \\
&= \mathbf{E}_I \mathbf{Z}'^{-1} \gamma \cosh(\gamma l) \mathbf{E}_V^{-1} \\
&= \mathbf{E}_I \cosh(\gamma l) \mathbf{Z}'^{-1} \gamma \mathbf{E}_V^{-1} \\
&= \mathbf{E}_I \cosh(\gamma l) \mathbf{E}_I^{-1} \mathbf{E}_I \mathbf{Z}'^{-1} \mathbf{E}_V^{-1} \mathbf{E}_V \gamma \mathbf{E}_V^{-1} \\
&= \cosh(\Gamma l)^T \mathbf{Z}^{-1} \Gamma \\
&= \cosh(\Gamma l)^T \mathbf{Y}_0
\end{aligned} \tag{A-14}$$

These proofs are based on the fact that multiplication of diagonal matrices is commutative.

## APPENDIX B

# THE LEAST SQUARES PROBLEM

It is often necessary to solve an overdetermined system of equations. It is usually not possible to find an exact solution to this problem, so a solution with some *minimum error* is sought. In the *least squares problem*, the sum of the squares of the errors at each point is minimized.

Let the points  $(t_i, y_i)$ ,  $i = 1, \dots, n$  be a set of samples on some curve,  $y$ . Let  $p(t_i)$  be an approximation to  $y$  at  $t_i$ . The error at  $t_i$ , then, is  $|y - p(t_i)|$ . Our task is to minimize:

$$\|r\|_2^2 = \sum_{i=1}^n |y - p(t_i)|^2 \quad (\text{B-1})$$

While there are more general ways to approach this problem, this discussion will focus on the more specific polynomial solution:

$$p(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_{m-1} t^{m-1} \quad (\text{B-2})$$

If we could manage a perfect match for each  $y_i$ , such that  $p(t_i) = y_i$ , then we have a system of the form:

$$\begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^{m-1} \\ 1 & t_2 & t_2^2 & \dots & t_2^{m-1} \\ 1 & t_3 & t_3^2 & \dots & t_3^{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & t_n & t_n^2 & \dots & t_n^{m-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{bmatrix} \quad (\text{B-3})$$

This can be expressed more compactly as  $\mathbf{Ta} = \mathbf{y}$ .

This is an overconstrained problem, and may not have an exact solution. However, we can consider a solution with the *least error*, and specifically, a solution with the *least sum-of-squares error*.

This means that the two-norm of the difference between the desired result and the actual result,  $\|\mathbf{Ta} - \mathbf{y}\|_2$ , is the smallest possible value for any given  $\mathbf{a}$ .

## APPENDIX C

### RESIDUE CALCULATION

This section presents some facts about how to calculate residues in the partial fraction expansion of a rational function.

First, define the following function, and assume that each pole is of multiplicity ‘1’:

$$f(s) = \frac{N(s)}{D(s)} = \frac{\prod (s - z_n)}{\prod (s - a_n)} \quad (\text{C-1})$$

The partial fraction expansion of  $f(s)$  looks like:

$$f(s) = \sum_n \frac{c_n}{(s - a_n)} + P(s), \quad (\text{C-2})$$

where  $P(s)$  possibly contains a few terms, depending on the order of  $f(s)$ .

Each residue,  $c_n$ , is calculated by “removing” all other terms. This is done by multiplying  $f(s)$  by the corresponding pole,  $(s - a_n)$ , and then setting  $s = a_n$ . This has the dual effect of cancelling out the pole in the  $c_n$  term and setting all other terms to zero. In other words:

$$c_n = [(s - a_n)f(s)]|_{s = a_n} \quad (\text{C-3})$$

#### C.1 Complex Residues

According to the fundamental theorem of algebra, each polynomial has a unique factorization over the complex numbers. It follows that if a polynomial with real coefficients has any complex roots with a non-zero imaginary part, then they must exist in complex conjugate pairs. This can be shown by taking the complex conjugate of both sides.

There is a result, not shown here, that states that the residues of each pole (of multiplicity  $n$ ) is unique. This fact will be used to show that if a pair of poles are complex conjugates of each other, then their corresponding residues are also complex conjugates. However, only poles of multiplicity 1 are considered here because it is one of the assumptions of the vector fitting algorithm.

Begin with the expression

$$f(s) = \frac{N(s)}{\left[ \prod_n (s - a_n)(s - \bar{a}_n) \right] \left[ \prod_k (s - a_k) \right]} = \sum_n \left[ \frac{c_{n_1}}{s - a_n} + \frac{c_{n_2}}{s - \bar{a}_n} \right] + \sum_k \frac{c_k}{s - a_k}, \quad (\text{C-4})$$

where the coefficients of  $f(s)$  are real, and there are  $N$  conjugate pairs of poles and  $K$  real poles.

Take the conjugate of each side of Eq. (C-4):

$$\frac{\overline{N(s)}}{\left[ \prod_n (s - \bar{a}_n)(s - a_n) \right] \left[ \prod_k (s - a_k) \right]} = \sum_n \left[ \frac{\bar{c}_{n_1}}{s - \bar{a}_n} + \frac{\bar{c}_{n_2}}{s - a_n} \right] + \sum_k \frac{c_k}{s - a_k} \quad (\text{C-5})$$

Note that the  $c_k$  values and the coefficients of  $N(s)$  are real, and thus have equivalent conjugate values, preserving the fact that the original function has only real coefficients.

From the uniqueness property,  $\bar{c}_{n_1}$  must be equal to  $c_{n_2}$ , which shows that the residues corresponding to complex conjugate poles must themselves be complex conjugates of each other.

## C.2 More on Complex Residues

There is another approach, useful to this document, that also demonstrates this fact about complex residues. It is a hand-waving argument using the time-domain equivalents of each term.

Define a function in the frequency domain as:

$$F(s) = \frac{c_1}{s - a} + \frac{c_2}{s - \bar{a}} \quad (\text{C-6})$$

Its time-domain equivalent is

$$\begin{aligned} f(t) &= c_1 e^{at} u(t) + c_2 e^{\bar{a}t} u(t) \\ &= e^{\sigma t} u(t) (c_1 \cos(\omega t) + j c_1 \sin(\omega t) + c_2 \cos(\omega t) - j c_2 \sin(\omega t)) \end{aligned} \quad (\text{C-7})$$

if we define  $a = \sigma + j\omega$ .

If this function is to have any meaning in the time domain, its imaginary part must be zero. In other words, if  $Im\{f(t)\} = 0$ , then:



$$e^{\sigma t}u(t)[(Im\{c_1\} + Im\{c_2\})\cos(\omega t) + (Re\{c_1\} - Re\{c_2\})\sin(\omega t)] = 0 \quad (\text{C-8})$$

or

$$\begin{aligned} Im\{c_1\} + Im\{c_2\} &= 0 \\ Re\{c_1\} - Re\{c_2\} &= 0 \end{aligned} \quad (\text{C-9})$$

which implies that  $c_1 = \bar{c}_2$ .

## APPENDIX D

### ZEROS OF A RATIONAL FUNCTION

This section shows how eigenvalues can be used to find the zeros of a rational function given in the partial fraction form:

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

The residues,  $c_n$ , and the poles,  $a_n$ , are known.

Mathematical induction will be used to show that the zeros of this function are equal to the eigenvalues of  $\mathbf{H}$ , where:

$$\mathbf{H}_N = \text{diag}(a_1, a_2, \dots, a_N) - \begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix} \begin{bmatrix} c_1 & c_2 & \dots & c_N \end{bmatrix} \quad (\text{D-2})$$

#### D.1 Expansion

Combining the terms of Eq. (D-1) yields the expression

$$N(s) = \sum_{n=1}^N \left\{ \left( \prod_{i=1}^{n-1} (s-a_i) \right) c_n \left( \prod_{i=n+1}^N (s-a_i) \right) \right\} + \prod_{n=1}^N (s-a_n) \quad (\text{D-3})$$

for the numerator. Phrased another way, our problem is to find the solution of the equation:

$$N(s) = 0 \quad (\text{D-4})$$

#### D.2 Eigenvalues

Finding the eigenvalues of  $\mathbf{H}$  involves solving the expression

$$\begin{vmatrix} a_1 - c_1 - \lambda & -c_2 & -c_3 & -c_4 & \dots & -c_N \\ -c_1 & a_2 - c_2 - \lambda & -c_3 & -c_4 & \dots & -c_N \\ -c_1 & -c_2 & a_3 - c_3 - \lambda & -c_4 & \dots & -c_N \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -c_1 & -c_2 & -c_3 & -c_4 & \dots & a_N - c_N - \lambda \end{vmatrix} = 0 \quad (\text{D-5})$$

for  $\lambda$ .

This can be rearranged and manipulated with basic row and column operations to become:

$$\begin{vmatrix} (\lambda - a_1) + c_1 & c_2 & \dots & c_{N-2} & c_{N-1} & c_N \\ -(\lambda - a_1) & (\lambda - a_2) & 0 & \dots & 0 & 0 \\ 0 & -(\lambda - a_2) & (\lambda - a_3) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -(\lambda - a_{N-2}) & (\lambda - a_{N-1}) & 0 \\ 0 & 0 & \dots & 0 & -(\lambda - a_{N-1}) & (\lambda - a_N) \end{vmatrix} = 0 \quad (\text{D-6})$$

Our task is to show that the expression for this determinant is the same as that for the numerator, replacing  $\lambda$  with  $s$ .

### D.3 Proof

This is easy to show for the  $N = 1$  case, and the  $N = 2$  case is

$$\begin{vmatrix} (\lambda - a_1) & -(\lambda - a_2) \\ c_1 & (\lambda - a_2) + c_2 \end{vmatrix} = (\lambda - a_1)(\lambda - a_2) + c_2(\lambda - a_1) + c_1(\lambda - a_2) = 0, \quad (\text{D-7})$$

which is exactly the expression for the numerator of a second-order polynomial expressed in the form of Eq. (D-1).

Next, assume that it works for the  $N = k$  case, where:

$$|\mathbf{H}_k| = \sum_{n=1}^k \left\{ \left( \prod_{i=1}^{n-1} (\lambda - a_i) \right) c_n \left( \prod_{i=n+1}^k (\lambda - a_i) \right) \right\} + \prod_{n=1}^k (\lambda - a_n) \quad (\text{D-8})$$

We proceed now with the  $N = k + 1$  case. By examination,

$$|\mathbf{H}_{k+1}| = (\lambda - a_{k+1})|\mathbf{A}| + (\lambda - a_k)|\mathbf{B}|, \quad (\text{D-9})$$

where

$$|\mathbf{A}| = \begin{vmatrix} (\lambda - a_1) + c_1 & c_2 & \dots & c_{k-1} & c_k \\ -(\lambda - a_1) & (\lambda - a_2) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -(\lambda - a_{k-2}) & (\lambda - a_{k-1}) & 0 \\ 0 & \dots & 0 & -(\lambda - a_{k-1}) & (\lambda - a_k) \end{vmatrix} \quad (\text{D-10})$$

and

$$|\mathbf{B}| = \begin{vmatrix} (\lambda - a_1) + c_1 & c_2 & \dots & c_{k-1} & c_{k+1} \\ -(\lambda - a_1) & (\lambda - a_2) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -(\lambda - a_{k-2}) & (\lambda - a_{k-1}) & 0 \\ 0 & \dots & 0 & -(\lambda - a_{k-1}) & 0 \end{vmatrix}. \quad (\text{D-11})$$

The value of  $|\mathbf{A}|$  in Eq. (D-10), by observation, is simply  $|\mathbf{H}_k|$ , and the value of  $|\mathbf{B}|$  in Eq. (D-11) is:

$$|\mathbf{B}| = c_{k+1} \prod_{n=1}^{k-1} (\lambda - a_n) \quad (\text{D-12})$$

As well, we observe that the sign is positive, independent of whether  $k$  is even or odd.

It is now possible to form the expression

$$\begin{aligned} |\mathbf{H}_{k+1}| &= (\lambda - a_{k+1})|\mathbf{H}_k| + c_{k+1} \prod_{n=1}^{k-1} (\lambda - a_n) \\ &= \left( \sum_{n=1}^{k+1} \left\{ \left( \prod_{i=1}^{n-1} (\lambda - a_i) \right) c_n \left( \prod_{i=n+1}^{k+1} (\lambda - a_i) \right) \right\} + \prod_{n=1}^{k+1} (\lambda - a_n) \right), \end{aligned} \quad (\text{D-13})$$

observing that the added partial fraction (index “ $k + 1$ ”) now appears.

QED

## D.4 Calculation with Reals

It is possible to do the entire calculation with real values, even though some of the poles and residues may be complex. One reason for this is that it is easier to write a computer program that does not have to worry about complex arithmetic, and another is that it guarantees that any complex eigenvalues come out as perfect conjugate pairs.

One possible way to do this is to place the conjugate poles and residues in the matrices such that each pair is adjacent, and then replace that portion of the matrix with another matrix that has a suitable similarity transformation, via the matrix  $\mathbf{P}$ :

$$\begin{bmatrix} a & 0 \\ 0 & \bar{a} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} r & \bar{r} \end{bmatrix} \Rightarrow \left( \mathbf{P}^{-1} \left( \begin{bmatrix} a & 0 \\ 0 & \bar{a} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} r & \bar{r} \end{bmatrix} \right) \mathbf{P} = \mathbf{P}^{-1} \begin{bmatrix} a & 0 \\ 0 & \bar{a} \end{bmatrix} \mathbf{P} - \mathbf{P}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} r & \bar{r} \end{bmatrix} \mathbf{P} \right) \quad (\text{D-14})$$

The eigenvalues remain the same under these conditions, and this is the motivation for performing this transformation.

The matrix form of a complex number:

$$\begin{bmatrix} a' & a'' \\ -a'' & a' \end{bmatrix} \equiv a' + ja'' \quad (\text{D-15})$$

seems like a good choice for the first term in Eq. (D-14), and yields, via some algebra, some constraints on  $\mathbf{P}$ :

$$\mathbf{P} = \begin{bmatrix} p_1 & -jp_1 \\ p_2 & jp_2 \end{bmatrix} \quad \text{where} \quad \mathbf{P}^{-1} \begin{bmatrix} a & 0 \\ 0 & \bar{a} \end{bmatrix} \mathbf{P} = \begin{bmatrix} a' & a'' \\ -a'' & a' \end{bmatrix} \quad (\text{D-16})$$

Next, we must consider a suitable choice for the second term. Careful observation reveals that  $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} r' & r'' \end{bmatrix}$ , the transformed version where  $r = r' + jr''$ , is not a good choice since the two rows are not linearly independent, and do not produce a solution for  $\mathbf{P}$ . A choice of

$$\begin{bmatrix} 2 \\ 0 \end{bmatrix} \begin{bmatrix} r' & r'' \end{bmatrix} \quad (\text{D-17})$$

instead, gives:

$$\mathbf{P} = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix} \quad (\text{D-18})$$

The constant ‘2’ is there to enforce two things: that the lower element is zero, and that the matrix containing  $r$  and  $\bar{r}$  becomes  $\begin{bmatrix} r' & r'' \end{bmatrix}$ .

Finally, we can conclude the following:

Calculation of the zeros using the eigenvalue technique discussed in this section is possible to do using only real values, even if there are complex poles and residues. This is done by replacing:

1. pairs of complex poles in their 2x2 segment in Eq. (D-2) with the 2x2 matrix from Eq. (D-15),
2. pairs of residues in their 1x2 segment in Eq. (D-2) with the 1x2 real and complex part from Eq. (D-17), and
3. each “one” in the Nx1 matrix of Eq. (D-2) with the 2x1 matrix from Eq. (D-17).

Once these substitutions are made, the eigenvalues are found in the usual manner.

## D.5 Additional Notes on Eigenvalue Calculation

The matrix  $\mathbf{H}$  from Eq. (D-2) can be written in upper Hessenberg form:

$$\mathbf{H} = \begin{bmatrix} -a_1 + c_1 & c_2 & \dots & c_{N-2} & c_{N-1} & c_N \\ a_1 & -a_2 & 0 & \dots & 0 & 0 \\ 0 & a_2 & -a_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{N-2} & -a_{N-1} & 0 \\ 0 & 0 & \dots & 0 & a_{N-1} & -a_N \end{bmatrix} \quad (\text{D-19})$$

The eigenvalues can be found directly using an iterative procedure, such as the QR algorithm.

## APPENDIX E

### RELATIONSHIP BETWEEN PHASE AND ATTENUATION FOR A MINIMUM PHASE SHIFT FUNCTION

The phase shift of a given *minimum phase shift* function can be calculated for a specific frequency if the attenuation is known. First, define

$$H(j\omega) = e^{A(\omega) + jB(\omega)}, \quad (\text{E-1})$$

where  $A(\omega)$  is the attenuation function and  $B(\omega)$  is the phase shift function. Note that  $\ln|H(j\omega)| = A(\omega)$  and  $\angle H(j\omega) = B(\omega)$ .

Bode derived several formulas [15] that relate these two quantities. The formula used in this paper is

$$B(\omega_c) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{d}{du} A(\omega) \ln \coth \frac{|u|}{2} du, \quad (\text{E-2})$$

where  $u = \ln \frac{\omega}{\omega_c}$ .

This particular formula is useful because most of the area under the curve lies around  $\omega_c$  and an approximation to the integral can be derived using only this segment.

Rewrite the equation:

$$\begin{aligned} B_c &= \frac{1}{\pi} \int_{-\infty}^{\infty} \left( \frac{dA}{du} - \frac{dA}{du} \Big|_{u=0} + \frac{dA}{du} \Big|_{u=0} \right) \ln \coth \frac{|u|}{2} du \\ &= \left[ \frac{dA}{du} \Big|_{u=0} \frac{1}{\pi} \int_{-\infty}^{\infty} \ln \coth \frac{|u|}{2} du \right] + \left[ \frac{1}{\pi} \int_{-\infty}^{\infty} \left( \frac{dA}{du} - \frac{dA}{du} \Big|_{u=0} \right) \ln \coth \frac{|u|}{2} du \right] \end{aligned} \quad (\text{E-3})$$

The integral

$$\frac{1}{\pi} \int_{-\infty}^{\infty} \ln \coth \frac{|u|}{2} du = \frac{2}{\pi} \int_0^{\infty} \ln \coth \frac{u}{2} du \quad (\text{E-4})$$

can be expanded using power series as follows.

$$\begin{aligned}
 \ln \coth \frac{u}{2} &= \ln(1 + e^{-u}) - \ln(1 - e^{-u}) \\
 &= \sum_{k=1}^{\infty} \frac{(-1)^{k+1} e^{-ku}}{k} - \sum_{k=1}^{\infty} \frac{-e^{-ku}}{k} \\
 &= 2 \sum_{k=1}^{\infty} \frac{e^{-(2k-1)u}}{2k-1}
 \end{aligned} \tag{E-5}$$

Now,

$$\begin{aligned}
 \frac{2}{\pi} \int_0^{\infty} 2 \left( \sum_{k=1}^{\infty} \frac{e^{-(2k-1)u}}{2k-1} \right) du &= \frac{4}{\pi} \sum_{k=1}^{\infty} \int_0^{\infty} \frac{e^{-(2k-1)u}}{2k-1} du \\
 &= \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{1}{(2k-1)^2} \\
 &= \frac{4\pi^2}{\pi 8} \\
 &= \frac{\pi}{2}
 \end{aligned} \tag{E-6}$$

using a well-known series representation of  $\pi$ . Thus:

$$B_c = \left[ \frac{\pi}{2} \cdot \frac{dA}{du} \Big|_{u=0} \right] + \left[ \frac{1}{\pi} \int_{-\infty}^{\infty} \left( \frac{dA}{du} - \frac{dA}{du} \Big|_{u=0} \right) \ln \coth \frac{|u|}{2} du \right] \tag{E-7}$$

Furthermore, note that

$$du = d \left( \ln \frac{\omega}{\omega_c} \right) = d(\ln \omega - \ln \omega_c) = d(\ln \omega), \tag{E-8}$$

so we can write the equation

$$\frac{dA}{du} = \frac{d}{d(\ln \omega)} \ln |H(j\omega)|, \tag{E-9}$$

observing that this is just the derivative of the magnitude function in the log scale.



This discussion explains the reasoning behind the minimum phase shift calculation that Gustavsen and Semlyen describe in [14].

## APPENDIX F

# STATE EQUATION FORM OF AN N<sup>TH</sup> ORDER DIFFERENTIAL EQUATION

An n<sup>th</sup> order system with constant coefficients can be written as:

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_{n-1} \frac{dy}{dt} + a_n y = b_0 \frac{d^n u}{dt^n} + \dots + b_n u \quad (\text{F-1})$$

This can be expressed in state equation form if we select the state variables [34] as

$$\begin{aligned} x_1 &= y - \beta_0 u \\ x_2 &= \dot{y} - \beta_0 \dot{u} - \beta_1 u = \dot{x}_1 - \beta_1 u \\ x_3 &= \ddot{y} - \beta_0 \ddot{u} - \beta_1 \dot{u} - \beta_2 u = \dot{x}_2 - \beta_2 u \\ &\dots \\ x_n &= y^{(n-1)} - \beta_0 u^{(n-1)} - \dots - \beta_{n-1} u = \dot{x}_{n-1} - \beta_{n-1} u \end{aligned} \quad (\text{F-2})$$

where

$$\begin{aligned} \beta_0 &= b_0 \\ \beta_1 &= b_1 - a_1 \beta_0 \\ \beta_2 &= b_2 - a_1 \beta_1 - a_2 \beta_0 \\ &\dots \\ \beta_n &= b_n - a_1 \beta_{n-1} - \dots - a_{n-1} \beta_1 - a_n \beta_0 \end{aligned} \quad (\text{F-3})$$

As well:

$$\dot{x}_n = y^{(n)} - \beta_0 u^{(n)} - \dots - \beta_{n-1} \dot{u} = \beta_n - a_n x_1 - a_{n-1} x_2 - \dots - a_1 x_n \quad (\text{F-4})$$

Hence, this system can be written as

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix} u \quad (\text{F-5})$$

$$y = [1 \ 0 \ 0 \ \dots \ 0] \mathbf{x} + \beta_0 u$$

or

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du \end{aligned} \quad (\text{F-6})$$

# APPENDIX G

## COMPLEX IDENTITIES AND IMPLEMENTATION TECHNIQUES

This chapter is devoted to various complex identities and implementation techniques.

### G.1 Why Is This Chapter Here?

All of the material here is likely known to the readers of this thesis, or found in most elementary textbooks containing material on complex arithmetic. However, there are many identities and techniques that, when not in front of someone implementing the mathematics of transmission lines in software, are tedious and time-consuming to derive or look up in reference material.

Software that calculates data using complex numbers must ultimately utilize real arithmetic, and so the identities presented here are geared towards such software. Furthermore, most of the software developed here operates almost exclusively on the real and imaginary parts of the data. Hence, the focus in this chapter will be on Cartesian identities.

### G.2 Notation

Let  $z_1 = a + jb$  and  $z_2 = c + jd$ . Many of the identities presented here use two complex numbers, and  $z_1$  and  $z_2$  will represent those two values. Also,  $z$  will be equivalent to  $z_1$ .

### G.3 Basic Identities

#### G.3.1 Rudimentary

$$\overline{z_1} = a - jb \tag{G-1}$$

$$z_1 \overline{z_1} = |z_1|^2 = a^2 + b^2 \tag{G-2}$$

$$e^z = e^a e^{jb} = e^a (\cos(b) + j \sin(b)) \tag{G-3}$$

$$\begin{aligned} \operatorname{Re}\{\ln(z)\} &= \ln\sqrt{a^2 + b^2} \\ \operatorname{Im}\{\ln(z)\} &= \operatorname{atan}\frac{b}{a} \end{aligned} \tag{G-4}$$

**G.3.2 Additive**

$$\begin{aligned} \operatorname{Re}\{z_1 + z_2\} &= a + c \\ \operatorname{Im}\{z_1 + z_2\} &= b + d \end{aligned} \tag{G-5}$$

$$\begin{aligned} \operatorname{Re}\{z_1 - z_2\} &= a - c \\ \operatorname{Im}\{z_1 - z_2\} &= b - d \end{aligned} \tag{G-6}$$

**G.3.3 Multiplicative**

$$\begin{aligned} \operatorname{Re}\{z_1 z_2\} &= ac - bd \\ \operatorname{Im}\{z_1 z_2\} &= bc + ad \end{aligned} \tag{G-7}$$

$$\frac{z_1}{z_2} = \frac{z_1 \bar{z}_2}{z_2 \bar{z}_2} = \frac{z_1 \bar{z}_2}{|z_2|^2} \tag{G-8}$$

$$\operatorname{Re}\left\{\frac{z_1}{z_2}\right\} = \frac{ac + bd}{c^2 + d^2} \tag{G-9}$$

$$\operatorname{Im}\left\{\frac{z_1}{z_2}\right\} = \frac{bc - ad}{c^2 + d^2}$$

**G.3.4 Trigonometric**

$$\begin{aligned} \operatorname{Re}\{e^z\} &= e^a \cos(b) \\ \operatorname{Im}\{e^z\} &= e^a \sin(b) \end{aligned} \tag{G-10}$$

$$\begin{aligned} \operatorname{Re}\{\sin(z)\} &= \frac{\sin(a)}{2}(e^b + e^{-b}) \\ \operatorname{Im}\{\sin(z)\} &= \frac{\cos(a)}{2}(e^b - e^{-b}) \end{aligned} \tag{G-11}$$

$$\begin{aligned} \operatorname{Re}\{\cos(z)\} &= \frac{\cos(a)}{2}(e^b + e^{-b}) \\ \operatorname{Im}\{\cos(z)\} &= -\frac{\sin(a)}{2}(e^b - e^{-b}) \end{aligned} \tag{G-12}$$

### G.3.5 Hyperbolic

$$\operatorname{Re}\{\tanh(z)\} = \frac{e^{2a} - e^{-2a}}{e^{2a} + e^{-2a} + 2\cos(2b)} \quad (\text{G-13})$$

$$\operatorname{Im}\{\tanh(z)\} = \frac{2\sin(2b)}{e^{2a} + e^{-2a} + 2\cos(2b)}$$

$$\operatorname{Re}\{\coth(z)\} = \frac{e^{2a} - e^{-2a}}{e^{2a} + e^{-2a} - 2\cos(2b)} \quad (\text{G-14})$$

$$\operatorname{Im}\{\coth(z)\} = \frac{-2\sin(2b)}{e^{2a} + e^{-2a} - 2\cos(2b)}$$

$$\operatorname{Re}\{\operatorname{csch}(z)\} = \frac{2\cos(b)(e^a - e^{-a})}{e^{2a} + e^{-2a} - 2\cos(2b)} \quad (\text{G-15})$$

$$\operatorname{Im}\{\operatorname{csch}(z)\} = \frac{-2\sin(b)(e^a + e^{-a})}{e^{2a} + e^{-2a} - 2\cos(2b)}$$

## G.4 Implementation Techniques

### G.4.1 Hyperbolic Functions

The hyperbolic functions cannot be implemented as written because of a potential overflow and indeterminate condition. If the magnitude of  $a$  is very large, then  $e^a$  will be calculated as  $\infty$ , and this will result in trying to calculate  $\infty/\infty$ . On systems that use IEEE 754 arithmetic, this calculation will produce a “Not a Number”.

The solution for this problem is to first determine the sign of  $a$ , and then to multiply the top and bottom by either  $e^{-a}$  if  $a$  is positive, or by  $e^a$  if  $a$  is negative. The resulting equation will now contain only exponentials of the form  $e^{-2a}$  for positive  $a$ , or  $e^{2a}$  for negative  $a$ . These approach zero as  $|a|$  gets very large. The result is that a  $\infty/\infty$  condition will never occur.

### G.4.2 Magnitude Functions

The discussion about calculating the magnitude of a complex number is also relevant for calculating the 2-norm of a real or complex vector.

On a computer, a floating-point number is always stored in a finite space, and so there exists a maximum representable value. Usually, any calculations that produce a number

that goes beyond this maximum overflows, and the result is infinite. Note that in IEEE 754 arithmetic,  $\infty$  and  $-\infty$  are valid numbers.

Mathematically, certain calculations should not overflow, but intermediate steps in the implementation just might. For example, the magnitude of a complex number is given by

$$|z| = \sqrt{a^2 + b^2}. \quad (\text{G-16})$$

Let  $M$  be the maximum representable floating-point number. If we let

$$a = b = \sqrt{\frac{M}{2}}, \quad (\text{G-17})$$

then the result of calculating  $|z|$  should be  $M$ . Now, let

$$a = b = \sqrt{\frac{M}{2}} + \Delta, \quad (\text{G-18})$$

where  $\Delta$  is small. Theoretically, the value of  $|z|$  is less than  $M$ . However, performing the calculation from Eq. (G-16) will result in an overflow because the intermediate result before taking the square root is larger than  $M$ .

The solution to this problem is to first choose the larger of  $a$  and  $b$ , divide each term by this value, perform the calculation as usual, and then multiply the answer by that value.

To illustrate, say  $a$  is larger:

$$|z| = a \sqrt{1 + \left(\frac{b}{a}\right)^2} \quad (\text{G-19})$$

Now, the intermediate calculations cannot possibly overflow because each term under the square root is less than or equal to one. The correct answer will now result.

Page 199 of [32] has an excellent discussion of this concept as applied to calculating the 2-norm of a vector.

## APPENDIX H

### DON'T FALL FOR THESE

It is possible to optimize many of the steps in this paper by doing such things as reordering steps or caching values used more than once. This section contains a list of some of the pitfalls one may encounter when implementing a simulator.

A reminder: ***Do not optimize until the very end, if at all.*** It may be tempting to perform many obvious optimizations, but it is more than likely the implementor will insert a bug that will be very difficult to track down and eliminate.

These pitfalls may seem simple and obvious, but it is nice to have them acknowledged and written down somewhere for that one time when the fix to such a problem is not obvious.

#### H.1 Exponentials and Phase

When performing an exponential on a complex number, you lose information if you need the phase again afterwards. For example:

$$e^{a+jb} = e^a \cos b + je^a \sin b \quad (\text{H-1})$$

The imaginary part of the exponent,  $b$ , is the phase, and the real part of the exponent,  $a$ , is the log of the magnitude of the result.  $a$  can be recovered always, but  $b$  cannot be recovered.

For transmission lines, the phase contains important information such as the phase velocity. It is also used to recover unknown phase information for a minimum phase shift function using the Bode formulation of Appendix E.

No procedure that makes use of this information should be given the result of the exponential because it needs to make use of the inverse tangent function. The principal value of this function is usually in the range  $-\pi$  to  $\pi$ , while the range of  $b$  may be anywhere.