Enhancing Signal Integrity Design Methodologies Utilizing Discrete Frequency Domain Techniques

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE
UNIVERSITY OF MINNESOTA
BY

Paul Eric Dahlen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Hal H. Ottesen, Adviser

April 2014

Acknowledgements

I would like to acknowledge the efforts of my Ph.D. advisor, Hal H. Ottesen, of the University of Minnesota - Rochester, and also retired from IBM, for his patience, assistance, and mentorship on this project. Our technical discussions and advising sessions have enabled these results. Similarly, I would like to acknowledge my previous academic advisors, Ronald D. Moe at the University of North Dakota, and Keith S. Champlin at the University of Minnesota - Twin Cities, without whose encouragement I would have not have pursued further graduate degrees in electrical engineering.

I also gratefully acknowledge the support of my employer, the IBM Corporation, which has generously funded this project through the IBM Degree Work Study Program and evaluation of intellectual property items related to this project for patenting or publication. I would like to particularly thank my electrical packaging technical mentors, IBM Distinguished Engineers Gerald K. Bartley and John M. Ryba, as well as all of my direct first-line IBM managers over my career, namely, Lowell Avery, Dennis Foster, Wayne Vlasak, Scott Bancroft, Linda Van Grinsven, Michael Gruver, and Jim Barnhart.

Finally, I must thank the following individuals, my IBM electronic packaging and signal integrity engineering colleagues, for their input and peer review of the dissertation.

Gerald K. Bartley, *Distinguished Engineer*, IBM Rochester
Mark O. Maxson, *Senior Technical Staff Member*, IBM Rochester
Darryl J. Becker, *Senior Electronic Packaging Engineer*, IBM Rochester
Gregory R. Edlund, *Senior Signal Integrity Engineer*, IBM Rochester
Trevor J. Timpane, *Senior Signal Integrity Engineer*, IBM Rochester
Matthew S. Doyle, *Advisory Signal Integrity Engineer*, IBM Rochester
Benjamin A. Fox, *Advisory Signal Integrity Engineer*, IBM Rochester
Wesley D. Martin, *Advisory Signal Integrity Engineer*, IBM Rochester
Thomas W. Liang, *Advisory Signal Integrity Engineer*, IBM Rochester
Jesse M. Hefner, *Staff Signal Integrity Engineer*, IBM Rochester
George R. Zettles, *Staff Signal Integrity Engineer*, IBM Rochester
Layne A. Berge, *Signal Integrity Engineer*, IBM Rochester

Dedication

This dissertation is dedicated to my immediate family. My parents, Howard and Gloria, have always encouraged, but never pressured, me. My wife, Jane, and my sons Peter, Philip, and Andrew, are my greatest supporters and cheerleaders. They are my most priceless treasures.

Abstract

Signal integrity engineering involves the use of electrical models and time-domain simulation to predict signal waveform degradation as the signal propagates across interconnects. It is employed most prevalently in the design of large digital systems, such as computers. Typically, the analysis and design techniques are concentrated in the continuous time domain, with the evaluation of time-domain waveform signal attributes being the primary tool for quantification of the degradation effects. Consistent with this continuous time-domain approach, the system models are often identified and expressed in the analog frequency domain, since this is the most natural domain for model identification, either by electromagnetic field simulation or empirical measurement.

This research investigation focuses on the use of digital signal processing techniques in the discrete time domain and associated discrete frequency domains to augment typical signal integrity engineering techniques. Specifically, it explores in detail the use of Laplace-domain (s-domain) to z-domain transform methods to convert system interconnect models identified in the analog frequency domains to models in the discrete frequency domains. The models are first converted from the analog frequency domain, using known vector fitting algorithms, to form a rational function approximation for the system in the s-domain. They are then converted from the s-domain to the z-domain using methods generally applied in the fields of control theory and digital filter design, but which are less familiar in the field of signal integrity engineering. Two new s- to z-domain transformation techniques are developed that are particularly well-suited for signal integrity applications.

The z-domain models are then assessed thoroughly in the z-plane using a variety of pole-zero analysis techniques to gain further insight into the nature of the system, and a new enhanced graphical method is introduced for the efficient assessment of such models in the z-plane. The overall results of this project are targeted toward enhancing signal integrity design methodologies in an industrial setting.

Table of Contents

List of Tables

List of Figures

List of Acronyms

2D	two-dimensional
3D	three-dimensional
AC	alternating current
ARX	auto regressive exogeneous
ARMA	auto regressive moving average
ARMAX	auto regressive moving average exogeneous
ASTAP	advanced statistical analysis program
AS/X	advanced statistical analysis program extended
ATCA	Advanced Tele-Communication Architecture
BER	bit error rate
BLT	bilinear transform
BLT0	bilinear transform with no-pre-warping applied
BLT1	bilinear transform using pre-warping at one critical frequency
BLT2	bilinear transform using pre-warping of entire discrete frequency vector
BSM	bottom side metal
C4	controlled collapse chip connect
CAD	computer-aided design
CAE	computer-aided engineering
CCGA	ceramic column grid array
CDR	clock data recovery
CEC	central electronics complex
CMOS	complementary metal oxide semiconductor
CPU	central processing unit
CT	continuous time
CTFT	continuous-time Fourier transform
DC	direct current
DDR	double date rate
DFE	decision-feedback equalization

DIMM	dual in-line memory module
DSA	digital sampling analyzer
DSP	digital signal processing, or discrete signal processing
DT	discrete time
DTFT	discrete-time Fourier transform
DFT	discrete Fourier transform
EDA	electronic design automation
ETFE	empirical transfer function estimate
FFE	feed-forward equalization
FFT	fast Fourier transform
FIR	finite impulse response
HDD	hard disk drive
HSS	high-speed SERDES
HSSCDR	IBM's high-speed SERDES clock data recovery statistical simulator
HSPICE	commercial version of SPICE developed by Synopsys, Inc.
I^2C	inter-integrated circuit
IB	Infiniband
IBIS-AMI	I/O Buffer Information Specification Algorithmic Modeling Interface
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
IFFT	inverse fast Fourier transform
IIT	impulse invariant (method) transform
IIR	infinite impulse response
IO	input / output
JTAG	Joint Test Action Group
LLSE	linear least square error
LSE	least squares error
LSEE	least squares error estimator
LTI	linear time-invariant
MBLT	modified bilinear transform
MIMO	multiple-input, multiple-output

MLE	maximum-likelihood estimator
MMZT	modified matched-Z transform
MMZT1	modified matched-Z transform with gains matched at DC
MMZT2	matched-Z transform with gains matched at analog pole frequencies
MZT	matched-Z transform
PCIe	Peripheral Component Interface Express
PDF	probability density function
PEM	prediction error method
PSPICE	commercial version of SPICE now owned by Cadence Design Systems
PWB	printed wiring board
PWL	piecewise linear
RF	radio frequency
RLC	resistor inductor capacitor
RLGC	R, L, G, C transmission line parameters
ROC	region of convergence
Rx	receiver
SAS	serial attach SCSI
SATA	serial advanced technology attachment
SCSI	small computer system interface
SERDES	serializer / deserializer
SI	signal integrity
SID	system identification
SISO	single-input, single-output
SPICE	simulation program with integrated circuit emphasis
SSA	statistical simulation analysis
SSD	solid state device
TSM	top side metal
Tx	transmitter
USB	Universal Serial Bus
USPTO	United States Patent and Trademark Office
VNA	vector network analyzer

List of Symbols and Notation

t	continuous time	[sec]
n	discrete time sampling index	[n is an integer]
k	discrete frequency sampling index	[k is an integer]
f	analog frequency	[Hz]
$\Omega = 2\pi f$	analog radian frequency	[radians/sec]
$s = \sigma + j\Omega$	complex analog frequency (s-domain)	[radians/sec]
$\sigma = Re[s]$	real part of complex variable s	[radians/sec]
T_s	sampling interval	[sec]
$f_s = 1/T_s$	sampling frequency	[samples/sec or Hz]
$\omega = \Omega T_s$	digital frequency	[radians]
$z = re^{j\omega}$	complex digital frequency (z-domain)	[radians]
$\omega_o = 2\pi/N$	discrete frequency resolution	[radians]
$f_o = f_s/N$	equivalent analog frequency resolution	[Hz]
$f_N = f_s/2$	Nyquist frequency	[samples/sec or Hz]
$x(t)$	continuous time-domain function	
$h_c(t)$	continuous-time impulse response	
$\delta(t)$	Dirac delta function	
$X(j\Omega)$	continuous-time Fourier transform of $x(t)$	
$X(s)$	Laplace transform of $x(t)$	
$H_a(j\Omega)$	analog frequency transfer function	
$H_a(s)$	analog transfer function; Laplace transform of $h_c(t)$	
$x(n)$	discrete time-domain sequence, a uniformly-sampled version of $x(t)$	
$h_d(n)$	discrete-time impulse response	
$\delta(n)$	Kronecker delta function	
$X(e^{j\omega})$	discrete-time Fourier transform of $x(n)$	
$X(k)$	discrete Fourier transform of $x(n)$	
$X(z)$	z-transform of $x(n)$	

$H_d(e^{j\omega})$	discrete frequency transfer function	
$H_d(z)$	digital transfer function; z-transform of $h_d(n)$	
$a, b, c, \ldots$	scalar parameters	
$\hat{a}, \hat{b}, \hat{c}, \ldots$	estimated scalar parameters	
$\underline{\theta}$	matrix of scalar parameters $\underline{\theta}$	
$\underline{\hat{\theta}}$	estimated matrix of scalar parameters	
$\underline{I}$	identity matrix	
R	transmission line parameter, conductor loss	[ohms/meter]
L	transmission line parameter, line inductance	[H/meter]
G	transmission line parameter, dielectric loss	[mhos/meter]
C	transmission line parameter, line capacitance	[F/meter]
l	transmission line length parameter	[meters]
Z_o	characteristic impedance	[ohms]
$\gamma = \alpha + j\beta$	transmission line propagation factor	[1/meter]
$\alpha = Re[\gamma]$	transmission line attenuation factor	[neper/meter]
$\beta = Im[\gamma]$	transmission line phase factor	[radians/meter]
$\underline{S}$	Scattering matrix	
S_{ij}	Scattering parameters, matrix elements of $\underline{S}$	
$\mathcal{F}\{\ldots\}$	continuous-time Fourier transform operator	
$\mathcal{P}\{\ldots\}$	(Cauchy) principal value	
$\mathcal{L}\{\ldots\}$	Laplace transform operator	
$\mathcal{Z}\{\ldots\}$	z-transform operator	
k_f	circuit frequency scaling factor	

Chapter 1: Introduction and Thesis Organization

1.1 Introduction

Signal integrity engineering consists of the analysis and design of chip driver and receiver circuits, and electronic packaging interconnect structures, such that signals propagating between chips undergo minimal distortion. This is done through the prediction and analysis of time-domain voltage waveforms, using computer-aided design tools such as electromagnetic field solvers and circuit simulation tools. As inter-chip interface speeds have approached and surpassed a few tens and hundreds of MHz, signal integrity engineering has become an integral part of the design process for digital systems. This is especially true for physically large, complex systems such as computers and their associated subsystems, which today are comprised of interfaces which commonly operate in the range of a few GHz or more. Every subsystem of a modern computer, including the main processor-memory central electronics complex (CEC), the various input-output (IO) subsystems, and the data storage subsystems comprised of hard disk drive (HDD) and solid-state device (SSD) components, now requires signal integrity engineering. Even commonly-known, industry-standard externalized connections to these subsystems operate in the hundreds of MHz or GHz ranges, and thus require extensive signal integrity analysis to guarantee proper electrical functionality.

The goal of this research project is to successfully apply known methods and theory from the fields of digital signal processing (DSP) and system identification (SID) to analysis and design problems in the field of signal integrity (SI) engineering. The overall objective is to significantly enhance known state-of-the-art signal integrity analysis and design methodologies, especially those methodologies geared toward the development of complex digital electronic systems such as computers.

Extensive experience as a practicing signal integrity design engineer, and comprehensive academic background in signal integrity engineering theory, classic microwave/radio-frequency (RF) theory, and DSP theory, have all led to the observation that much of the method and theory of DSP can be methodically applied to the field of digital signal integrity. The fundamental reason for this is that, even though the signal waveforms considered in signal integrity engineering are generally of a continuous-time (CT) nature, they are actually acquired or generated, and thus represented, as discrete-time (DT) sequences of numbers. These waveform representations are constructed either by empirical acquisition, using sampling instrumentation, or by prediction as simulated waveforms, using SPICE or similar tools on digital computers. Even so, known DSP methods are not often systematically applied to signal integrity problems, at least not on a large scale, primarily because most practicing signal integrity engineers do not have significant backgrounds in DSP and the direct applicability of DSP methods is not widely recognized.

This leads to the following observation, which serves as the primary motivation for this research project:

> *Known processes and methodologies for performing signal integrity engineering could be significantly enhanced by borrowing and applying known theory and techniques from digital signal processing engineering, for the purposes of system analysis and system identification. Specifically, comprehensive time-domain analysis strategies, based mainly upon SPICE simulation and similar methods, could be enhanced by utilizing z-domain methods to improve the model development and evaluation process. In particular, acquiring the system transfer function* $H_d(z)$ *using s-domain to- z-domain transforms, and utilizing pole-zero analysis methods for model assessment and simplification could lead to significant improvements in these processes and methodologies.*

1.2 Thesis Organization

This thesis is organized into nine additional chapters:

Chapter 2: Background: An overview of the motivating applications, problems, and prevailing considerations in signal integrity engineering, as encountered in an industrial design environment, is presented.

Chapter 3: Theoretical Foundations: A review of various theoretical aspects from the fields of signal integrity engineering, radio-frequency (RF) and microwave engineering, system identification engineering, and digital signal processing engineering are presented. These multifaceted disciplines do not significantly overlap in their general practice. As such, most engineers practicing in these disciplines are not likely to have background in all of these areas.

Chapter 4: Identification of the System Model: An overview of the process of model extraction is presented, starting with the use of scattering parameters for the interconnect system identification method. Once the scattering matrix is obtained, the subsequent development of the system analog frequency transfer function $H_a(j\Omega)$ is presented. A comparison of other system identification techniques, as possible alternatives to scattering parameter analysis, is provided for completeness.

Chapter 5: Development of the Transfer Function Model $\boldsymbol{H_a(s)}$: A continuation of the process begun in Chapter 4 is presented, using the system identification data acquired in Chapter 4 to derive a more complete description of the system. The system transfer function $H_a(s)$, expressed in the s-plane of the complex analog frequency domain, is obtained from the analog frequency transfer function $H_a(j\Omega)$ found previously using previously known vector fitting algorithms.

Chapter 6: Development of the Transfer Function Model $\boldsymbol{H_d(z)}$ from $\boldsymbol{H_a(s)}$: A comparison of various methods to obtain the system transfer function $H_d(z)$ in the complex digital frequency domain from the system transfer function $H_a(s)$ in the complex analog frequency domain is presented, using commonly-known and prevalently-used s-to-z transforms from DSP theory.

Chapter 7: Development of the Transfer Function Model $\boldsymbol{H_d(z)}$ Directly from the Partial Fraction Expansion Terms of $\boldsymbol{H_a(s)}$: Demonstration of new methods for obtaining the system discrete transfer function $H_d(z)$ directly from the partial fraction expansion form of the system analog transfer function $H_a(s)$, which is the natural form of the output obtained directly from the vector fitting algorithms used in finding $H_a(s)$, are presented.

Chapter 8: Examination and Optimization of $\boldsymbol{H_d(z)}$ Utilizing z-Plane Techniques: The results and insight of Chapter 6 and Chapter 7 are used to subsequently suggest novel methods to: 1) look for opportunities to simplify the model mathematically (which enables more efficiency in the analysis process); 2) appropriately select a frequency range for effective system identification of the model, and a sampling rate for effective transformation of the model from the s- to the z-domains; and 3) utilize a new graphical technique for assessing the relationship between the pole-zero map of $H_d(z)$ and the frequency response $H_d\left(e^{j\omega}\right)$, as well as their individual attributes simultaneously.

Chapter 9: Signal Integrity Methodology Enhancements Resulting from the Utilization of the Discrete Transfer Function Model $\boldsymbol{H_d(z)}$: New criteria for comparing or selecting s- to z- transformation methods are introduced, and applied to the eight such transforms utilized in this project. A discussion of proposed enhancements to an industrial signal integrity methodology flow, utilizing the techniques developed in the preceding chapters, is also presented.

Chapter 10: Conclusions and Future Work: A summary of the project's major results and resulting publications, along with suggestions for possible future extensions, are presented.

Chapter 2: Background

2.1 Introduction

In the design of complex electronic systems, robust signal integrity performance requires significant investment in analysis and design. This investment comes in the form of highly-trained and specialized personnel resources, expensive engineering computer-automated design (CAD) or computer-automated engineering (CAE) tools, and in the form of expensive test and measurement equipment for design validation. Signal integrity engineers tend to have backgrounds in radio frequency (RF), microwave, or electromagnetic engineering or, alternatively, in analog circuit design. These skill sets do not necessarily overlap easily with digital designers, logic designers, or computer system architects, making the signal integrity role fairly specialized in many industrial design environments [1], [2].

A further complication is that signal integrity design and analysis tasks must, by definition, be performed concurrently with other system design tasks, and thus must be integrated into the overall chip-, package-, or system-level design flows [3]. This can be relatively complex to implement in a product design environment, especially if the engineering design teams are large, which is typical for the design of complex systems. Finally, the signal integrity engineering problem itself is technically complex, with clock speeds in advanced digital applications such as computers already in the GHz range. In such applications, even the miscellaneous peripheral interfaces typically operate in the hundreds of MHz [4]. Signal integrity engineering inherently requires the application of distributed circuit theory in lieu of lumped element theory [5], [6], which in itself requires even more specialized training. All of these factors require expensive investments to address signal integrity considerations.

This chapter will describe the typical aspects of modern signal integrity engineering, from the perspective of the methodologies and tasks used in an industrial setting, as applied to the design of large-scale, complex electronic systems. Included in this description will be the major attributes of a typical system design, including the functional block diagram for the system, its typical physical layout characteristics, and the translation of those physical attributes into an electrical model. Finally, an overview of the simulation methods used to generate time-domain data and frequency-domain data will be given, setting the context in which the rest of this project will be presented, and establishing the motivation for seeking new techniques to augment the existing methods.

2.2 System Description

Signal integrity engineering is generally required when the system designs become complex enough that multiple high-power, high-speed modules are required to implement the system. In this context, a module is defined as an integrated component consisting of one or more integrated circuit chips and the associated electronic package [7]. Some example system designs would be a typical computer workstation, a commercial server, or a storage system, each of which would contain sophisticated microprocessors and memory control modules, high-speed IO hub modules and bridges, high data rate and high-capacity memory devices, and high-performance hard disk and flash storage devices [8]. In addition, multiple high data rate system signaling interfaces, both custom and industry-standard, would typically be found interconnecting the various modules comprising the system. In a modern system, almost all of these interfaces typically require some type of signal integrity intervention to ensure their proper function with the required margin [1], and it may also be necessary to re-assess them over the system's entire life cycle if some of the components in the system need to be updated [9].

This section will describe a typical computer system, starting with an architectural block diagram and a physical description of the system packaging. System packaging generally includes the system enclosure, and also the manner in which the system card and board hierarchy is designed. The translation from the architectural and physical descriptions to an electrical representation is followed by a model extraction or identification step and its subsequent formatting into a SPICE netlist or similar model format. This will ultimately result in an electrical simulation run in signal integrity CAE software to generate time-domain waveforms, for interpretation by the signal integrity engineer.

2.2.1 System Block Diagram

Figure 2-1 shows an architectural block diagram for a typical UNIX and Linux[1] commercial server [8]. Note that the design has multiple high-speed, inter-chip signaling interfaces, most of which will require signal integrity analysis to achieve a functional design. The highest-performance interfaces, requiring the most signal integrity engineering, would be the processor-to-processor paths, the processor-to-memory data paths, the high-speed serializer/deserializer (SERDES) PCIe[2] and SAS[3] interfaces to the IO complex, and the double data rate (DDR) memory interfaces to the DIMMs.[4]

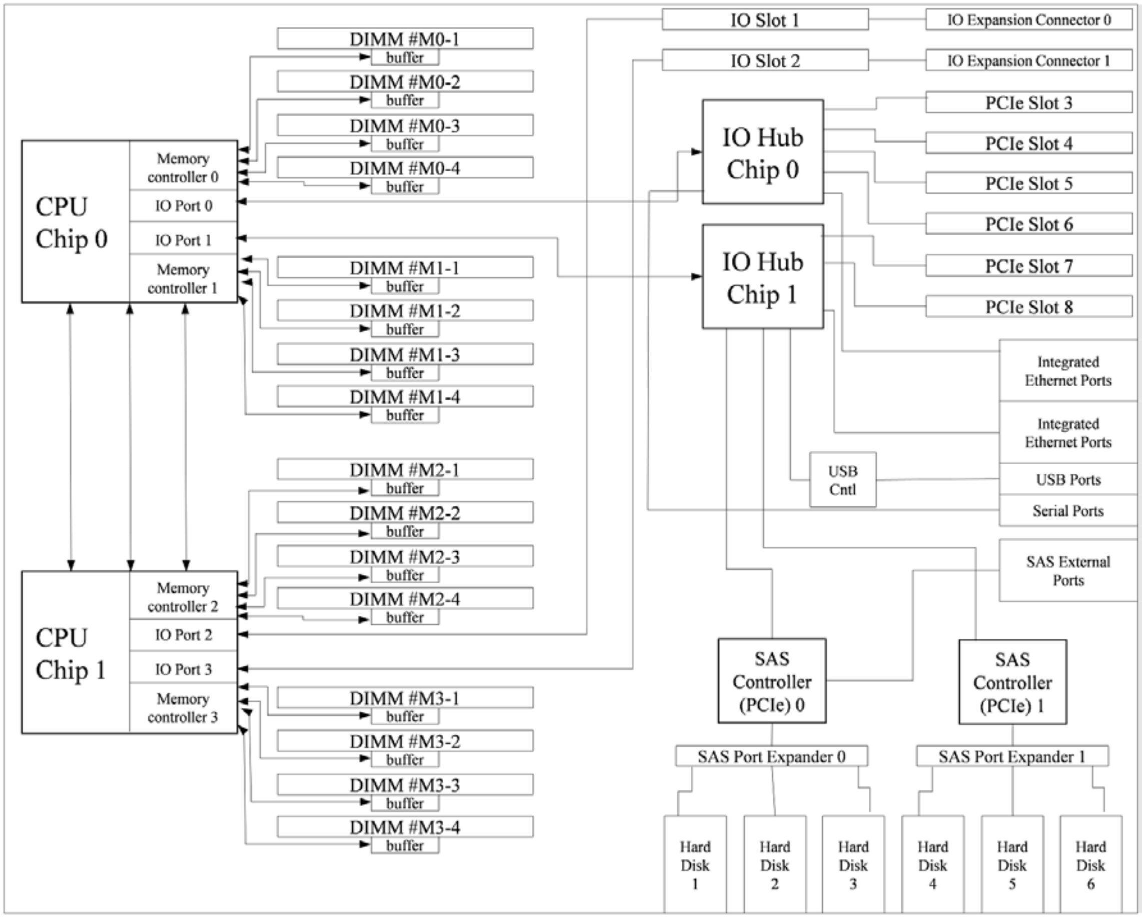


Figure 2-1: Architectural Block Diagram for a Typical Computer System

---

[1] UNIX and Linux are widely used multi-user operating systems used for desktop workstatations, commercial servers, and high-performance computing systems.

[2] PCIe is an industry-standard acronym for Peripheral Component Interface Express.

[3] SAS is an industry-standard acronym for Serial Attach SCSI (Small Computer System Interface).

[4] DIMM is an industry-standard acronym for dual in-line memory module.

It should be noted that, for the system of Figure 2-1, even the peripheral IO interfaces, such as JTAG[5], USB[6], and I^2C[7], may require some attention. Other examples of complex systems, including computer workstations, flash memory storage systems, and high-performance computing systems show similar characteristics, with a wide variety of high-speed, high-performance interfaces requiring signal integrity analysis [10], [11], [12], [13].

From the system's architectural block diagram of Figure 2-1, and after imposing relevant physical constraints due to system packaging characteristics, a conceptual physical layout of the system would be generated. The layout would be optimized to support any electrical, mechanical, thermal, and product or application requirements [7]. Examples of such constraints would be signaling and timing budgets, noise budgets, physical size, shape, or cost requirements, air flow and cooling requirements, and similarity to both the customer's existing physical system footprint and data center characteristics [2], [7] .

2.2.2 Physical Representation of an Interconnect Structure

A typical signaling channel ultimately consists of a chip-to-chip interconnect structure connecting two or more modules. Figure 2-2 shows an illustration of a typical processor-to-controller chip structure in a modern computer workstation. The physical representation of this inter-chip signaling path consists of an electrical path starting on the sending chip at the IO driver circuit, and continuing through the 1st-level package, which in this case is depicted as having ceramic column grid array (CCGA) technology for the bottom side metal (BSM) and a controlled collapse chip connect (C4) technology for the top side metal (TSM). The electrical path then continues through the printed wiring board (PWB), through a connector interface between the PWB and a daughter card PWB, and then finally back through the daughter card and onto another 1st-level CCGA package, then finally onto the receiving chip to the IO receiver circuit. Models are

---

[5] JTAG is an industry-standard acronym for Joint Test Action Group.
[6] USB is an industry-standard acronym for Universal Serial Bus.
[7] I^2C is an industry-standard acronym for Inter-Integrated Circuit.

added for the IO transmitter and receiver circuits using transistor-level or behavioral-level models. [2], [7], [14].

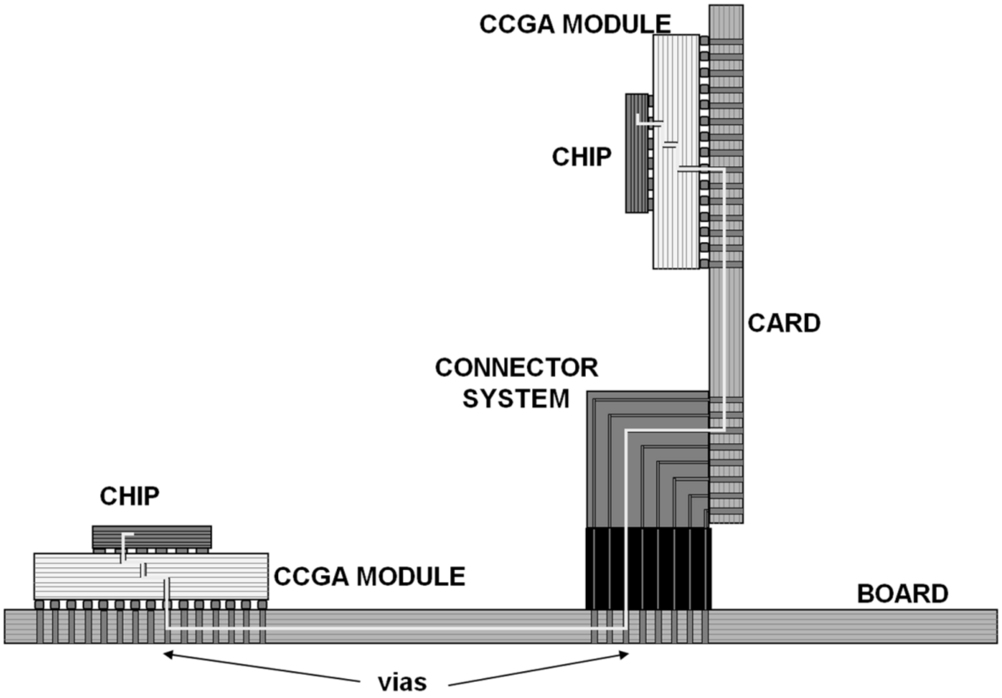


Figure 2-2: Processor-to-Controller Physical Path in a Typical Computer Application

2.2.3 Electrical Representation in Electrical Schematic Form in SPICE

Historically, lumped-element circuits have been used to represent undesirable electrical parasitic resistive, inductive, and capacitive effects, such as those due to the presence of circuit board vias, pads for solder attach, or connectors. Lumped elements are also used to represent passive components such as damping resistors, termination resistors, shunt capacitors, or ferrite bead inductors, which have been intentionally added to the path to condition the signal in a particular way [7]. Distributed-element circuit models, on the other hand, have been used to represent transmission lines, since distributed element circuit theory is based upon partial differential equation representations of the circuit, and can thus include the effects of propagation delay [6]. Indeed, the importance of the interconnect model characteristics has lead to extensive research into design and measurement techniques for transmission lines of the type used in modules, cables, and printed wiring boards (PWBs) [15]–[18].

As mentioned previously, a time-domain SPICE simulation is often run to predict the transient waveform at the receiver. Figure 2-3 shows an electrical model representation of the interconnect structure of Figure 2-2, now in electrical schematic form. Notice that connectors and other package-to-package or package-to-chip boundaries are modeled with RLC lumped-element parasitic elements if the physical dimensions are small enough relative to the wavelength of the system's signal waveforms, while PWB signal traces are modeled as transmission lines. Also labeled on the figure are typical waveforms at various points in the network. Notice the indicated distortion in the receiver's waveform as the signal propagates over the interconnect structure.

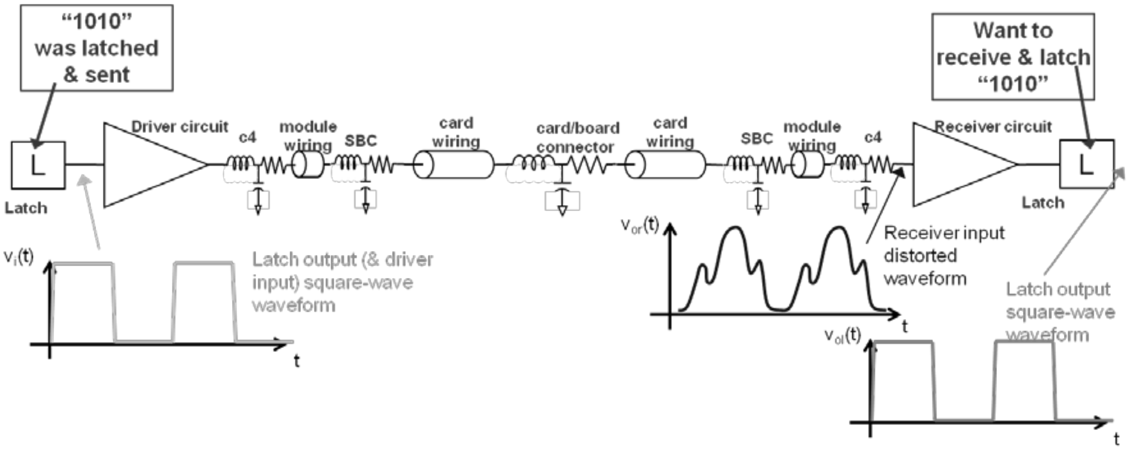


Figure 2-3: Electrical Schematic Representation of a Typical Point-to-Point Interconnection in SPICE Language Format

2.2.4 Electrical Representation as Two-Port Networks

Alternatively, for faster systems with higher frequencies of operation, various two-port network representations, such as scattering parameters, can be developed that modern SPICE programs can interpret [19]. One advantage of the scattering parameter format, which is shown in Figure 2-4, is that it is a natural format for characterizing or identifying the interconnect model, either by empirical measurement or by simulation in electromagnetic field simulators. Regardless of whether the scattering parameter format or traditional SPICE netlist format is used, the physical structure has now been translated to an electrical circuit structure for subsequent time-domain simulation.

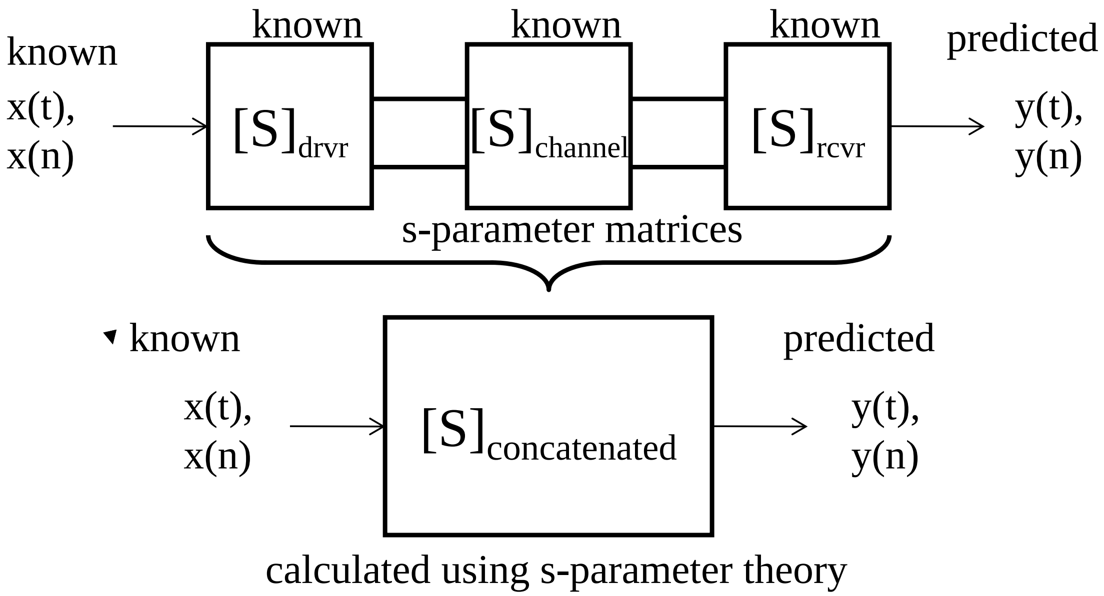


Figure 2-4: Electrical Schematic Representation of an Interconnect in Scattering-Parameter Format

2.3 Use of Simulation Techniques in Signal Integrity Engineering

Once the electrical representation of the system is obtained, in the form of Figure 2-3 or Figure 2-4, simulations can be run to predict system performance. Simulation is generally used in multiple ways. Simulations can be run in the frequency domain, using full-wave three-dimensional electromagnetic field solver programs, to calculate the desired lumped- or distributed-element circuit models. These models are then used in subsequent time-domain simulations to predict the time-response waveforms at the receiver circuit. Alternatively, the models extracted from the 3D field solvers can be further simulated in the frequency domain to further assess their frequency response.

2.3.1 Model Extraction

In order to perform a signal integrity time-domain simulation, an electrical model must first be created for the physical representation of the interconnect system depicted in Figure 2-2. This model needs to be accurate enough to adequately represent the electrical performance of the system to the required degree of accuracy. Beyond that, however, the level of complexity of the model is determined by the signal integrity engineer, and the resulting model that is obtained is certainly not unique.

As described in the previous section, there are two approaches to developing the models. The first approach, used historically, has been to use a combination of lumped circuit elements and distributed transmission line elements to model the overall interconnect path in the native SPICE language of the simulation engine [20], [21]. This approach has the disadvantage of being less accurate at higher frequencies, due to the limitations of lumped-element circuit theory for modeling distributed effects, but also has the significant advantage of being more intuitively familiar to most electrical engineers. It is also easier to isolate or correlate certain characteristics visible in the simulated waveforms to certain circuit elements.

Using this first approach, the lumped element representations can be extracted by a variety of commercially-available electromagnetic field solver tools that work directly on the three-dimensional physical structure of the system [22], [23], [24]. Similarly, the distributed element transmission line models can be extracted in the same or similar manner. One advantage for transmission line structures is that the distributed-element circuit representation can also be obtained using only a two-dimensional field solver tool [25], applied to the transmission line cross section, which requires considerably less effort.

The second approach, used more commonly at the present time, is to use a scattering parameter matrix [19] or macromodeling approach [26]. Such an approach can be used to represent either the entire interconnect system as a single model, or to concatenate multiple scattering parameter matrix models, each of which represents a particular component of the interconnect system. The scattering parameter approach has the advantage of being more numerically sound at high-frequencies, since the resulting model is no longer based on lumped-element circuit theory, but also has the disadvantage of being far less intuitive for the engineer to interpret. The models are extracted using the same types of CAE tools used in the first approach above, but the output format of the models is different, in this case being either scattering parameter matrices or black box macromodels, not SPICE language format [4].

2.3.2 Simulation Using SPICE Engines to Calculate the Transient Response

The signal integrity analysis of a typical high-speed interface in a digital system usually involves prediction of the waveform shapes for the signals propagating across the interface. This is usually, but not always, done in the time-domain, using a circuit simulation tool such as SPICE. This simulation activity is expensive to undertake, because: (1) the CPU time and computing resource can be significant; (2) organizing and developing the matrix of simulations to be run requires significant insight and experience; (3) analyzing the vast amounts of data generated can be difficult; and (4) a typical computer system design can have multiple high-speed interfaces, requiring these tasks to be replicated multiple times. From a design flow and schedule standpoint, these high-speed interface designs must usually be done in parallel, requiring teams of several or more signal integrity engineers.

Traditionally and historically, for computer and similar digital system applications, time-domain analysis has dominated the field of signal integrity engineering. Much of the reason for this lies in the nature of the most prevalent signaling methods typically used in computer and other digital systems, namely, sending packets of digital ‘1s’ and ‘0s’ down a transmission line, as opposed to communication-theory modulation techniques used in RF and other systems. As a result, typical “bit strings” or “pulse trains” have significant frequency content, due to their non-periodic and square-wave “pulse-like” natures, making frequency-based analysis formulations rather cumbersome [27].

Fortunately, SPICE simulators have been available since the 1970s [20], [21]. Such tools have made it possible to analyze pulse train propagation directly in the time domain, especially for systems with frequencies in the tens of MHz, for which attenuation effects in interconnects can often be treated as negligible, and thus allowing the use of lossless transmission lines. With the advent of faster digital systems, however, which operate at frequencies in the hundreds of MHz or a few GHz, it has become necessary to model frequency-dependent losses in transmission line structures [2]. This has greatly complicated the SPICE time-domain analysis problem, stressing the ability of the simulators to accurately handle attenuation effects, and increasing the solution time of many simulations. Table 2-1 shows a summary of well-known SPICE simulation tools [20], [28]–[31].

Tool Name	**Company**	**Simulation Language**	**Availability**
PowerSPICE	IBM Corp.	SPICE and ASTAP	IBM Internal Use
HSPICE	Synopsis	SPICE	Commercially Available
PSPICE	Cadence	SPICE	Commercially Available
Spectre	Cadence	SPICE	Commercially Available

Table 2-1: Listing of Popular SPICE Simulation Tools

2.3.3 Simulation Using Statistical Simulation Analysis Engines to Calculate the Transient Response

Table 2-2 shows a summary of well-known statistical simulation analysis (SSA) simulation tools [32]–[37]. These tools are relatively new, have common basis of operation [36], and have become popular for the design of modern high-speed SERDES interfaces. Examples of such tools are the industry-standard StatEye, which is available as an open-source application, IBM’s HSSCDR, a proprietary tool available only within IBM to its developers and select customers, and the commercially-available simulation engines from CAE software vendors ANSYS, Agilent, SiSoft, and Cadence.

Tool Name	**Company**	**Availability**
HSSCDR	IBM Corp.	IBM Internal Use / Select External Customers
DesignerSI	ANSYS	Commercially Available
ADS	Agilent	Commercially Available
Quantum Channel Designer	SiSoft	Commercially Available
StatEye	stateye.org	Opensource / Freeware
Allegro Sigrity Serial Link Analysis Option	Cadence	Commercially Available

Table 2-2: Listing of Popular Statistical Simulation Analysis Tools

These SSA tools have the advantages of being able to simulate many thousands of bits in just a few seconds, and can also incorporate directly random- and stochastic phenomena from their probability density functions (PDFs), making it possible to account for random noise and channel effects. These tools make extensive use of digital signal processing theory, including both deterministic and stochastic effects [38].

2.3.4 Assessment of Transient Response Waveforms

Figure 2-5 shows a typical time-domain waveform for a 1-Gb/s SERDES interface, as generated from a SPICE simulation. Using basic circuit structures of the type depicted in Figure 2-3 or Figure 2-4, SPICE models are developed using standard SPICE element building blocks or scattering parameter model call routines. The subsequent simulations are performed and the resulting waveforms are evaluated in the continuous time domain. Simplistic models may be comprised simply of lumped resistance, inductance, and capacitance (RLC) elements, while more comprehensive models also make use of distributed-element representations, such as transmission lines. In addition, sophisticated models usually try to account for more numerically complex effects such as frequency-dependent signal attenuation and dispersion.

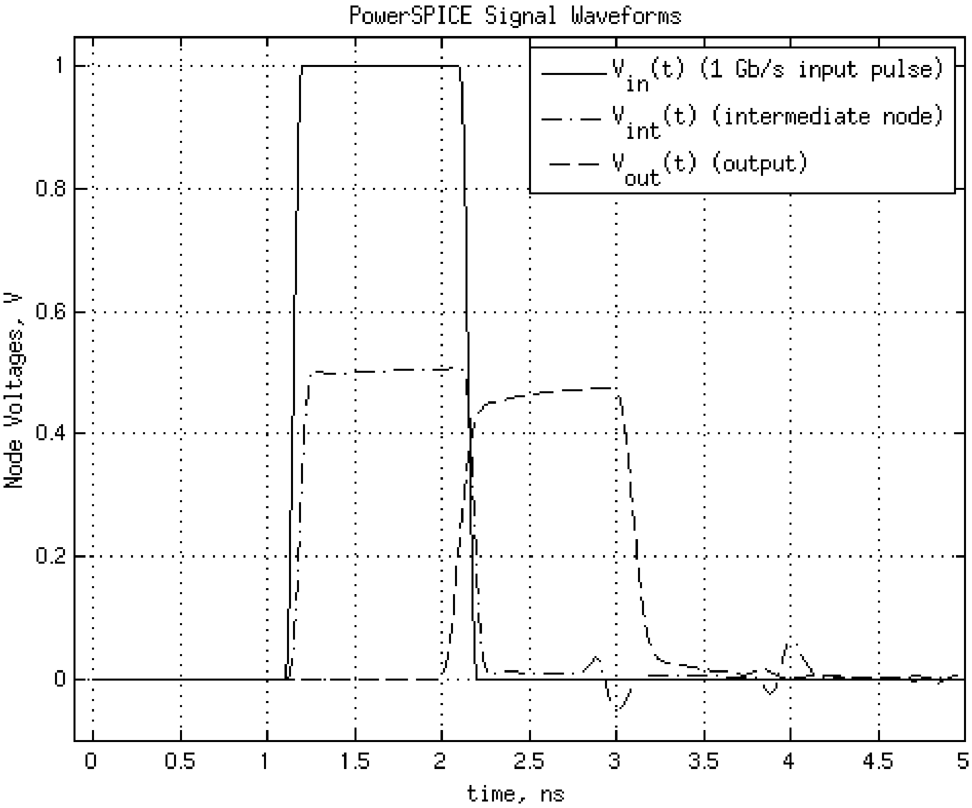


Figure 2-5: SPICE-Generated Signal Waveform for a 1 Gb/s SERDES Interface

A simple SPICE model may assume ideal V_{dd} and AC ground rails, where V_{dd} is defined as the operating voltage applied to the circuit in question. More complex representations may attempt to incorporate noise characteristics on the rails. These additional characteristics can add greatly to the models' sophistication and, usually, increased simulation run times result. Despite the potential for complex models and long simulation times, SPICE remains one of the most useful signal integrity design tools, because of the great flexibility and simplicity of its input language, its widespread use, and its general familiarity with the electrical engineering community. SPICE is a general-purpose tool that allows for a great variety of input assumptions, its input language is in the format of widely understood lumped-element circuit theory, and the correlation between the physical representation and the electrical representation in the time domain is straightforward for most electrical engineers. In contrast, many frequency-domain tools require more interpretation, and another layer of theory superimposed into the model formulation.

2.3.5 Using Simulation to Calculate the Frequency Response

Just as both SPICE-based and SSA-based simulation tools are used to calculate the continuous time response of an interconnect system, they can also be used to calculate the analog frequency response of the system. One advantage of this is that the signal integrity engineer can gain further insight into the nature of the system. Examples of this insight would be getting a estimate of the system's overall bandwidth characteristics, or a sense of its dominant poles and zeros, by looking at both magnitude response and phase response characteristics using a Bode plot [39], [40].

To obtain the frequency response, SPICE can be run in AC analysis mode, which essentially performs a series of sinusoidal steady-state analyses for a system like that shown in Figure 2-3, and expresses the results in the analog frequency domain [28], [29]. The frequency range over which the analysis is performed is determined and specified by the signal integrity engineer, and based upon the assumed frequency range of the system's input and output signals. It should be noted that, since the signals are generally of a pulse nature as shown in Figure 2-5, the frequency content can be considerable due to the fact that multiple harmonic frequencies must be present to comprise the signal.

Similarly, the SSA simulators can be used to calculated the analog frequency response of the interconnect model. These tools often are able to calculate the scattering parameter matrices of the overall concatenated model of Figure 2-4, which can be very useful in examining the attenuation and reflection characteristics of the entire network. Even though the individual component models used to create the overall channel models may have been initially extracted separately in the frequency domain, the simulator may have the ability to synthesize the overall response as a natural output of the process of performing the time-domain simulation, and thus is another set of output data available for evaluation [32].

2.4 Signal Integrity Engineering in an Industrial Setting

For the development of complex electronic systems, such as computer servers and large storage systems, signal integrity engineering becomes an important part of the design flow. It is not unusual to employ a team of several or more signal integrity engineers on system projects involving multiple high-speed interfaces [4]. For such projects, it is commonplace for the signal integrity team to share workload and break down the activity into various subtasks, including model development, time-domain simulation, and lab measurement activity. In such circumstances, is not unusual for the team to share model libraries, software tool licenses, and laboratory equipment.

2.4.1 Block Diagram of a Signal Integrity Engineering Design Flow

For large organizations, it is common for the signal integrity teams to interact with the other system design teams, including the card/board development teams, the chip/package development teams, and the system architecture teams. The cross-functional nature of an industrial signal integrity design flow is shown in Figure 2-6. It is not unusual for the signal integrity teams to be involved very early in the system floor planning process, to ensure that the physical arrangement of the system can support the required signaling speeds and other performance requirements.

Later in the design process, the signal integrity teams support the 1st-level package, card, and board design processes by providing specific design rules and wiring constraints to drive the wiring and physical design process for these components. These design rules and constraints are determined, either wholly or partially, by the circuit simulations described in the previous section. They may also be determined by standard interface design specifications, past experience, or from information provided by the module design teams for the case where a functional module is purchased from a 3rd-party vendor as a commercially-available component. Examples of such commonly-available modules are microprocessors, IO hub chips, or bus expander modules.

Finally, the signal integrity teams are involved in the verification of the designs once they have been completed. This post-design verification activity can consist of laboratory empirical measurements on prototype system hardware using high-speed signal analyzers, or it can be performed as another simulation set, this time using models extracted directly from the system CAD design data. Sophisticated design flows involving complex engineering tradeoffs generally emphasize early intervention by the signal integrity team, thereby eliminating the need for extensive post-design verification.

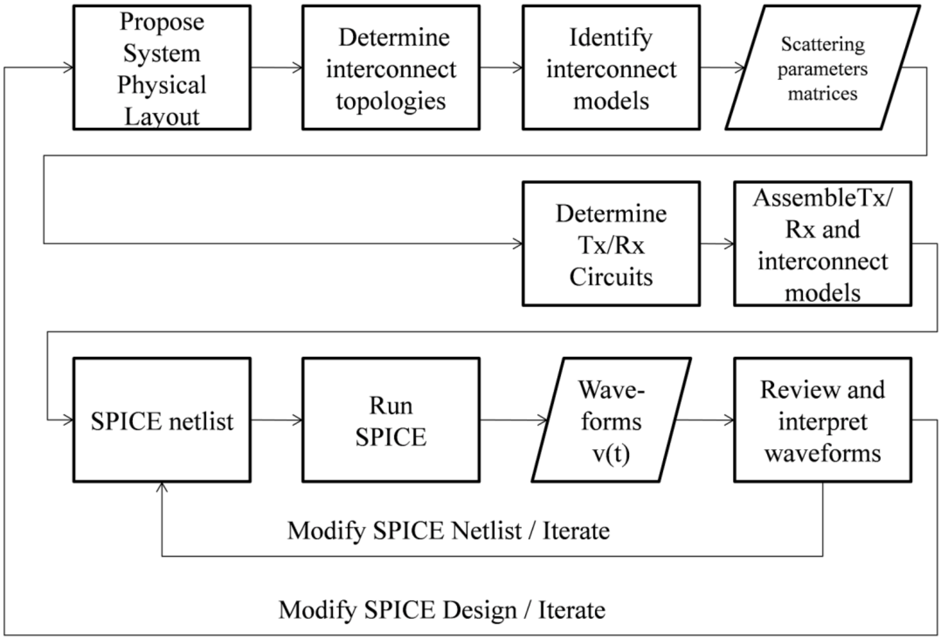


Figure 2-6: Block Diagram of an Industrial Signal Integrity Design Process

2.4.2 Problems and Challenges

In principle, an industrial signal integrity engineering process follows the flow depicted in Figure 2-6. In reality, however, these activities involve a series of engineering tradeoffs involving cost, effort, available resources, and return on investment. The model extraction and simulation process is difficult and time-consuming, and it is not always possible to execute it on all of the inter-module interconnection circuits in the system. In addition, in some cases the simulation tools can produce answers with numerical precision that is greater than what is appropriate, given the limitations of the resolution of the models or the level of knowledge of the system.

Several common limitations to the effectiveness of the signal integrity engineering of the system are prohibitive cost, prohibitive complexity of the models and resulting simulation, limitations in the analysis resolution of the methodologies or techniques, and limitations on assumptions behind the input to the simulation tools, or with the simulation tools themselves. Several examples will be given in the following sections.

2.4.2.1 Cost Limitations of Signal Integrity Engineering

Cost considerations can set the level of signal integrity engineering done for a particular system, just as for any other engineering design activity. The following examples demonstrate the reality of such considerations.

As a first example, suppose a system contains five high-speed interfaces, and that each high-speed interface in the system requires 100 simulations to effectively analyze. Suppose further that each simulation takes two hours to run, as may be the case when using a SPICE simulator with transistor-level driver and receiver models. Assuming unlimited computer resources are available, suppose further it takes a signal integrity engineer one week of effort (i.e., a "person-week") to submit 100 simulations and analyze the results, thereby completing the design of one interface. This translates into five person-weeks of effort to complete the analysis of the entire system's high-speed interface design. Assuming 42 hours per person-week, and an overall burden rate of $\$200$/hour, this translates to $(5\ PW)\ x\ (42\ hours\ /\ PW)\ x\ (\$200\ /\ hour)\ =\ \$42{,}000$ for engineering resource to analyze the system for signal integrity using SPICE. This effort assumes the models were already available for subsequent use in time-domain SPICE simulation. Clearly, reducing the overall effort, or at least improving the quality of the information extracted, or better utilizing the information extracted, would have value.

As a second example, consider the aforementioned 500 simulations which are required to analyze the design's five high-speed interfaces. Again assuming that a single

SPICE simulation requires two hours to run, this translates to a total of 1000 hours of CPU time. Given the likelihood that the interfaces must all be designed in parallel, it may be reasonable to assume that five 1-CPU workstations are required to execute the project, or equivalently, one 5-CPU multi-user machine. Assume the latter case requires the purchase of a large 6-CPU machine at a cost of $12,000.

Additional factors could also arise, such as delayed time-to-market resulting from analysis “re-spins” and the associated costs. The economic effects could be even more significant than those mentioned above if signal integrity design issues adversely affect the revenue stream associated with the project, due to initial time-to-market delays and product quality issues.

2.4.2.2 Model Complexity Limitations in Signal Integrity Engineering

Another commonly-encountered problem in signal integrity engineering is the temptation to resort to intensive computational analysis to overcome excessive model complexities, rather than to use models of appropriate complexity, which can be difficult to ascertain. This occurs when the methodology becomes over-reliant on automated simulation tools and techniques, thereby generating excessive, often marginally-useful data as the default output of the simulation tools. Such approaches are characterized by a lack of engineering insight that reduces or otherwise limits the usefulness of the generated data. This problem has become more common as more powerful and easier-to-operate software tools, and larger and faster CPUs, have become more readily available and less expensive.

Thus there is significant motivation to reduce the amount of data generated, and to make better use of the data that is generated. One key to this problem is to acquire and gain engineering insight by reviewing intermediately data as they are generated, so that better decisions could be made about what data to generate next, and thereby improving the overall efficiency of the analysis processes.

A related problem is the natural tendency to make incremental extensions to the base signal integrity model structures, with more parasitic elements, more frequency-dependent interconnect models, models with increased bandwidth, longer and more sophisticated bit stimulus patterns, more complex IO driver and receiver models, and faster signal edge rates with inherently more frequency content. This leads to ever increasing model size, complexity, and simulation run times. This can lead to another aspect of computationally-intensive analysis, in which it becomes a challenge to simply get the models to run at all, because of their size and complexity.

2.4.2.3 Limitations of the Prevalent Methodology and Techniques

As previously mentioned, many signal integrity engineers were trained in other disciplines. They may have started out as digital or analog circuit designers, and became more specialized, or they may have been trained as microwave, RF, or transmission line engineers [1], [2]. More specifically, it is uncommon for signal integrity engineers to have significant backgrounds in discrete signal processing (DSP), system identification (SID), control theory, or communication theory. As a result, techniques common to those disciplines, such as impulse response or transfer function representations, discrete Fourier transforms, and pole-zero analysis techniques, tend to be relatively unfamiliar. The observation that many known techniques from DSP and related disciplines could be very useful in signal integrity engineering is largely unrecognized, and relatively little work has been done to advance technology transfer in this area, at least in terms of overall signal integrity methodology development.

2.4.2.4 Inherent Limitations within Design Tools or Underlying Assumptions

This lack of technology transfer between the areas of DSP and signal integrity is unfortunate, given the direct applicability of many DSP methods. In reality, the inter-chip signals in digital systems are continuous-time (CT) analog waveforms [7]. Ironically, however, when working with these signals, signal integrity engineers often unknowingly implement them as discrete-time (DT) sequences. SPICE simulators, for example, generate signal waveforms using discrete sequences of numbers at small (but generally non-uniform) “time step” intervals. Thus the waveforms are actually DT sequences, even though the signal integrity engineer thinks of the waveforms as CT in nature. Similarly, when acquiring waveforms in the laboratory, SI engineers usually use digital sampling analyzers (DSAs), which simply acquire discrete-time samples of the analog waveforms at an appropriately high sampling rate [1], [2]. Once again, data which are interpreted by the signal integrity engineers as CT waveforms are actually DT sequences. As a consequence, for both empirical and predicted signal integrity waveforms, DSP techniques can usually be almost directly applied.

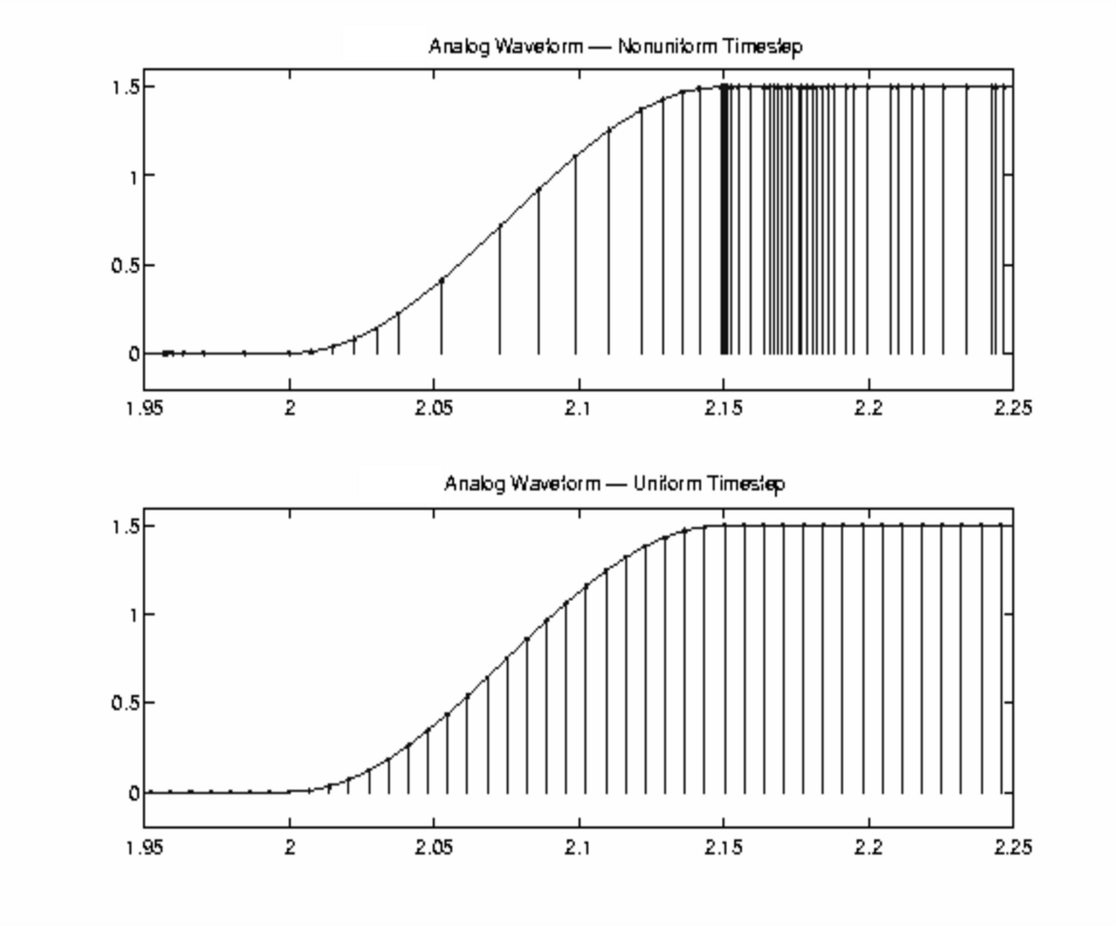


Figure 2-7: SPICE-Generated Waveforms Comprised of Uniform and Non-Uniform Timesteps

The upper waveform in Figure 2-7 shows a predicted discrete-time waveform as produced directly by the SPICE simulator, including "stem" representations of the data points to highlight the discrete samples and sampling intervals. Notice the signal exhibits a non-uniform time step between signal samples. This is typical in SPICE, as the simulator adjusts the time steps based upon the slopes of all the signals being studied at the various circuit nodes in the system. At any point in time where any signal has a fast di/dt or dv/dt, SPICE reduces the time step to improve resolution. At points in time where none of the signals are transitioning, SPICE increases the time step to save CPU time and storage space, with no significant loss in resolution. Since one of the assumptions for most common DSP methods is a uniform sampling rate, traditional DSP methods are not directly applicable to this waveform.

The lower waveform in Figure 2-7 shows the same SPICE waveform, re-interpolated using a very simple MATLAB routine such that the resulting DT sequence exhibits a uniform sampling rate. Note that the CT interpolated representation is indistinguishable from the CT interpolated representation of the original SPICE signal, at least to the human eye. The advantage with this new DT sequence, however, is that DSP methods are directly applicable since the sequence exhibits a uniform sampling interval! To the signal integrity engineer, both waveforms in Figure 2-7 are equivalent CT waveforms, as long as the interpolated CT waveforms overlay each other closely.

2.5 Conclusion

This chapter has described the typical aspects of modern signal integrity engineering, from the perspective of the methodologies and tasks used in an industrial setting, as applied to the design of large-scale, complex electronic systems. Descriptions of a typical system design, including the functional block diagram for the system, its typical physical characteristics, and the translation of those physical attributes into an electrical model, were shown. An overview of the simulation methods used to generate time-domain data and frequency-domain data was given, setting the context in which signal integrity engineering is performed today.

An obvious conclusion is that today’s signal integrity analysis activities are almost completely reliant on computationally-intensive simulation methods, and that these methods are primarily rooted in the continuous time domain for their interpretation, with analog frequency domain methods mostly limited to those activities required for effective extraction of models to feed the time-domain simulators subsequently.

The rest of this project will investigate ways to utilize various frequency-domain analysis techniques, primarily in the discrete frequency domains, to further examine, develop, and optimize the models prior to invoking the time-domain simulation process. The primary motivation for this approach is the desire to obtain additional insight into the nature of the system. This will provide an opportunity to both leverage that increased insight into better system designs and, possibly, to further simplify or optimize the models to enable more productive and effective time-domain simulation analysis.

Chapter 3: Theoretical Foundations

3.1 Introduction

The field of signal integrity analysis is based primarily on continuous time-domain representations of signals, and analog frequency-domain representations of models. Two of the greatest challenges are working with models requiring relatively large bandwidth and, similarly, working with signals with wide spectral content [1]. In contrast, the field of digital signal processing (DSP) is based primarily on discrete time-domain representations of signals, and digital frequency-domain representations of models and transfer functions.

The analytical methods most commonly used for signal integrity analysis include time-domain simulation methods such as SPICE, and model extraction methods based on electromagnetic field solvers [4]. Historically, signal integrity channel modeling has consisted of distributed-element transmission line representations of interconnects, and lumped-element parasitic representations of other discontinuities. More recently, as signals have surpassed 2 Gb/s, it has become increasingly common to use purely distributed-element representations of the entire channel in the form of s-parameters [38]. Frequency-based methods, if used at all, have traditionally been borrowed from the fields of microwave or RF engineering, or from analog control theory, and are generally focused in the analog frequency domains using continuous-time Fourier transform theory or, less commonly, using Laplace transform theory.

The analytical methods most commonly used for digital signal processing, in contrast, focus on discrete-time, sampled versions of time-domain signals, and digital frequency domain methods associated with discrete-time systems are more commonly used. These frequency-domain methods, also common to the field of digital control theory, include z-transforms, discrete-time Fourier transforms, and discrete Fourier transforms that are usually implemented as fast Fourier transform (FFT) algorithms.

Because of the widespread use of continuous time-domain and analog frequency-domain tools and methods, and lumped- and distributed-element circuit theory, signal integrity analysis and the associated design methodologies have become very complex. In general, methods common to digital signal processing have not been widely adopted for signal integrity engineering, possibly due to lack a of training and familiarity of these methods within the signal integrity community.

This chapter will summarize both the key theoretical foundations employed in the field of signal integrity and the key theoretical foundations employed in the field of digital signal processing, control theory, and system identification. It will establish a context for the use of the latter methods in the field of signal integrity engineering. The overview will begin with summaries of the familiar continuous and discrete time domains, and their associated frequency domains which are themselves also quite familiar and related to their respective time domains by various Fourier transform pairs. The discussion will then be expanded to include the definitions of the more general, and more powerful, complex frequency domains represented by the Laplace and z- transforms, and will summarize the relation of these two transforms to both their associated Fourier domains as well as to each other. A brief overview of select topics from circuit theory will be included next, including a discussion of transmission lines and scattering parameters, which enable the extension of circuit theory to high-frequency networks. Two specialized topics, those of system identification and vector fitting, are included next, as these methods allow for the extraction and formulation of system models that enable subsequent analysis in the complex frequency domains, which is the ultimate objective. Finally, several mathematical formulations for key system characteristics are presented that allow direct assessment of system characteristics in the complex frequency planes.

3.2 Continuous-Time and Analog-Frequency Domains

The emphasis by signal integrity engineers on continuous-time domain representation of signals, and analog-frequency domain representation of models, is largely a result of the historical ties to analog circuit design methods for signal analysis, and microwave and RF engineering methods for model extraction.

As such, signal integrity engineers almost always treat signals in the continuous-time domain and analog-frequency domain. Figure 3-1 shows a typical transient waveform at the receiver input node. The waveform is generally expressed as a function of continuous-time and represented on the graph as $v_{out}(t)$ vs. t, as shown.

It is understood that the waveform has spectral component content, and this is typically considered in the context of the analog frequency domain. The analog frequency domain's relation to the continuous-time domain is expressed formally via the continuous-time Fourier transform (CTFT) pair, defined as [41], [42]:

$$V(f) = V(j\Omega) = \mathcal{F}\{v(t)\} = \int_{-\infty}^{\infty} v(t)\, e^{-j2\pi tf} dt\ , \tag{3.1}$$

where f is the analog frequency in Hz, Ω is the analog radian frequency in radians/sec, defined as [39]:

$$\Omega = 2\pi f\ , \tag{3.2}$$

and

$$v(t) = \mathcal{F}^{-1}\{V(f)\} = \int_{-\infty}^{\infty} V(f)\, e^{j2\pi tf} df\ . \tag{3.3}$$

Note that this pair is specific to the continuous domains, and will be distinguished from other transform pairs associated with the discrete-time and discrete-frequency domains.

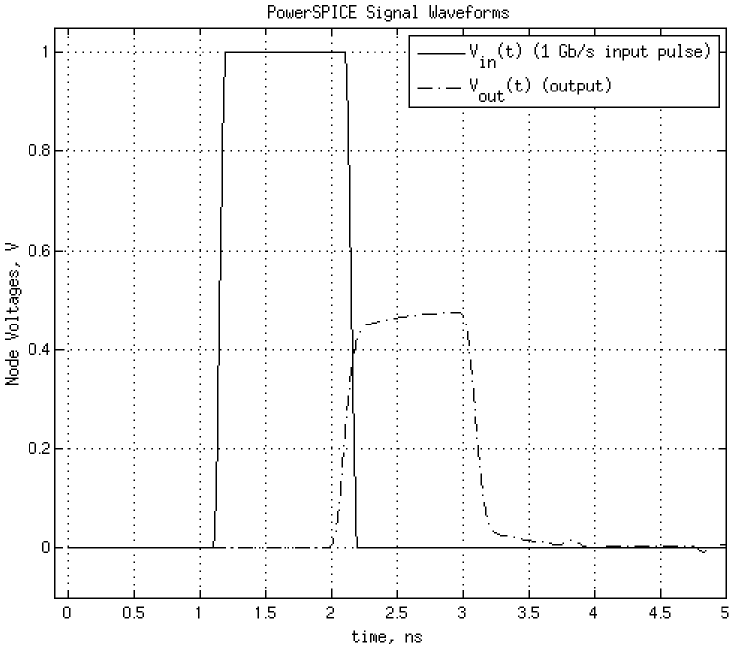


Figure 3-1: Example SPICE Output Waveform in the Continuous Time Domain

Note that it is common in the literature using continuous-time and analog frequency concepts for the analog radian frequency to use the symbol ω instead of the symbol Ω chosen here. The reason for using Ω will become evident when discussing the discrete frequency domains in future sections. From the definition for analog radian frequency, the analog complex frequency can be defined as [39]:

$$s = \sigma + j\Omega . \tag{3.4}$$

Here σ represents the distance from the vertical frequency axis in the complex s-plane, and $j = \sqrt{-1}$. The s-plane will be discussed in more detail in a following section.

The continuous and analog domains defined above are well understood by signal integrity engineers. Ironically, however, many of the simulation tools and measurement instruments used by the engineers often make extensive use of the discrete counterparts of these domains. Thus, the relationships between the continuous and discrete domains are important. These relationships, while well-understood by digital signal processing (DSP) engineers, are often not visible to, or well understood by, signal integrity engineers.

One example of the incorporation of DSP concepts into signal integrity engineering is that modern digital signal analyzer instruments have built-in sampling capabilities, as well as built-in Fast Fourier Transform algorithms. From a simulation standpoint, SPICE and similar transient simulator engines make use of digital sequences to represent continuous-time waveforms.

3.3 Discrete-Time and Discrete-Frequency Domains

Just as the field of signal integrity analysis is based primarily on continuous-time domain and analog frequency domain representation of signals, the field of digital signal processing (DSP) utilizes the discrete time and discrete frequency domains.

The discrete frequency domain’s relation to the discrete time domain is expressed formally for a discrete sequence $v(n)$ via the discrete-time Fourier transform (DTFT) pair, defined as [41], [42]:

$$V\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} v(n)\, e^{-j\omega n} , \tag{3.5}$$

and

$$v(n) \; = \; \frac{1}{2\pi} \int_{-\pi}^{\pi} V(e^{j\omega}) \, e^{j\omega n} d\omega \,, \tag{3.6}$$

where ω is the digital (or discrete) frequency and n is the discrete sampling index.

3.3.1 Discrete-Time Domain Relation to Continuous-Time Domain

For signal integrity analysis, a discrete sequence $v(n)$ can be considered to be a sampled version of the causal continuous-time function $v(t)$. This relationship can be expressed as [43]:

$$v(n) = v(t)|_{t=nT_s} \; = \; v(nT_s) \quad n = 0, 1\,, 2, \dots\,, \tag{3.7}$$

and where T_s is the sampling interval. T_s is inversely proportional to the sampling rate, defined as [41], [42]:

$$f_s = \frac{1}{T_s} \,. \tag{3.8}$$

3.3.2 Discrete-Frequency Domain Relation to Analog-Frequency Domain

It was mentioned previously that f is the analog frequency and has units of Hz. Ω is the analog radian frequency, with units of radians/s. The discrete frequency ω has units of radians and is related to f, Ω, and T_s as [41]:

$$\omega = \Omega T_s = 2\pi f T_S = \frac{2\pi f}{f_s} \,. \tag{3.9}$$

Again, note the distinction in notation here between the discrete frequency ω and the analog radian frequency Ω. It was pointed out earlier that for non-DSP applications, the analog radian frequency is often denoted ω, but Ω is used here to avoid confusion, given the present context.

3.3.3 DTFT, DFT, FFT, and CTFT Relationships

The relationship between the continuous-time and discrete-time Fourier transforms is one of the most fundamental concepts of digital signal processing. The relationship of the CTFT of a continuous time signal $v_a(t)$ and the DTFT of its sampled version $v(n)$ can be expressed as [41]:

$$V(e^{j\omega}) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} V_c(j(\frac{\omega}{T_s} - \frac{2\pi k}{T_s}))\,, \tag{3.10}$$

where $V_c(j\Omega)$ is the CTFT of the continuous time signal $v_a(t)$. Note that $V(e^{j\omega})$ is a continuous function in ω. The relation (3.10) basically states that $V(e^{j\omega})$ is a combination of frequency- and amplitude-scaled, periodically-repeated copies of $V_c(j\Omega)$.

The discrete Fourier transform (DFT) pair is defined as [41], [42]:

$$V(k) = \begin{cases} \sum_{n=0}^{N-1} v(n)e^{-j\frac{2\pi}{N}n}\,, & 0 \le k \le N-1 \\ 0, & otherwise \end{cases}, \tag{3.11}$$

and

$$v(n) = \begin{cases} \frac{1}{N}\sum_{k=0}^{N-1} V(k)e^{j\frac{2\pi}{N}k}\,, & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}. \tag{3.12}$$

where $v(n)$ and $V(k)$ are both finite-length sequences of length N. Note that, in contrast to $V(e^{j\omega})$, $V(k)$ is a discrete function in k, and the sequences of (3.11) and (3.12) are assumed to have values of zero for n and k values outside of the range from 0 to $N-1$.

In terms of the relationship between the DTFT and the DFT, it can be shown [41] that the discrete function $V(k)$ can be obtained by sampling the DTFT of $v(n)$ periodically as:

$$V(k) = V(e^{j\omega})|_{\omega=\frac{2\pi}{N}k} = V(e^{j\frac{2\pi}{N}k}), \qquad 0 \le k \le N-1\ . \tag{3.13}$$

Note that the DFT operates on discrete-time sequences of finite length, and results in a finite sequence in the sampled discrete frequency domain of the same length, while the DTFT yields an infinite-length function in the discrete frequency domain.

The fast Fourier transform (FFT) of a finite, discrete sequence $x(n)$ is a family of algorithms which allow for very efficient computation of the DFT for a sequence [41]. As such, the FFT is not a unique transform, but rather a specific implementation of the DFT.

FFT algorithms make use of symmetry and periodic properties of the DFT to enable the performance increase. Many modern simulation tools and laboratory instruments have built-in FFT computation engines, to allow for easy spectral analysis of continuous time waveforms, which have been sampled analytically or empirically. Often the sampling process is not visible to the signal integrity engineers using the tools or instrumentation. The most well-known and common family of FFT algorithms is the Cooley-Tukey radix-2 decimation-in-time algorithm, although there are many variations even within this particular family [42].

3.4 Complex Frequency Domain

The s-domain and z-domain represent the complex analog and digital frequency domains, respectively, and aid in the analysis of differential equations and difference equations in the continuous time and discrete time domains under certain conditions. The s- and z-domains are particularly suitable for filter design, and are widely used in DSP and control theory. More recently, the field of signal integrity analysis has utilized these domains as linear macromodeling has become more widely utilized [44], [45],[46], [47].

3.4.1 Laplace Transform Domain, or s-Domain

The Laplace transform domain, or s-domain, relates to the continuous-time domain via the unilateral Laplace transform pair, defined as [41], [42]:

$$V(s) = \mathcal{L}\{v(t)\} = \int_{0}^{\infty} v(t)e^{-st}dt\,, \tag{3.14}$$

and

$$v(t) = \mathcal{L}^{-1}\{V(s)\} = \frac{1}{2j\pi}\oint_{C} V(s)e^{st}ds\,. \tag{3.15}$$

It is also known that the bilateral Laplace transform is important for the analysis of some systems. The bilateral Laplace transform is defined as [41]:

$$V(s) = \int_{-\infty}^{\infty} v(t)e^{-st}dt\,. \tag{3.16}$$

For the Laplace transform, the region of convergence (ROC) is defined as the set of points in the s-plane for which the integral of (3.16) converges absolutely. The ROC is important when considering the stability and causality of systems.

3.4.1.1 Definition of the Two-Dimensional s-Plane

Note that s-domain analysis makes significant use of the concept of the complex two-dimensional (2D) s-plane, as shown in Figure 3-2. The relationship of the s-domain to the analog frequency domain Ω can be thought of as the latter mapping to the vertical axis of the complex s-plane where $\sigma = 0$, that is, where $s = j\Omega$. As such, the analog frequency domain can be thought of as a subset of the s-domain, with some information being lost when working in the Ω-domain in lieu of the s-domain [42].

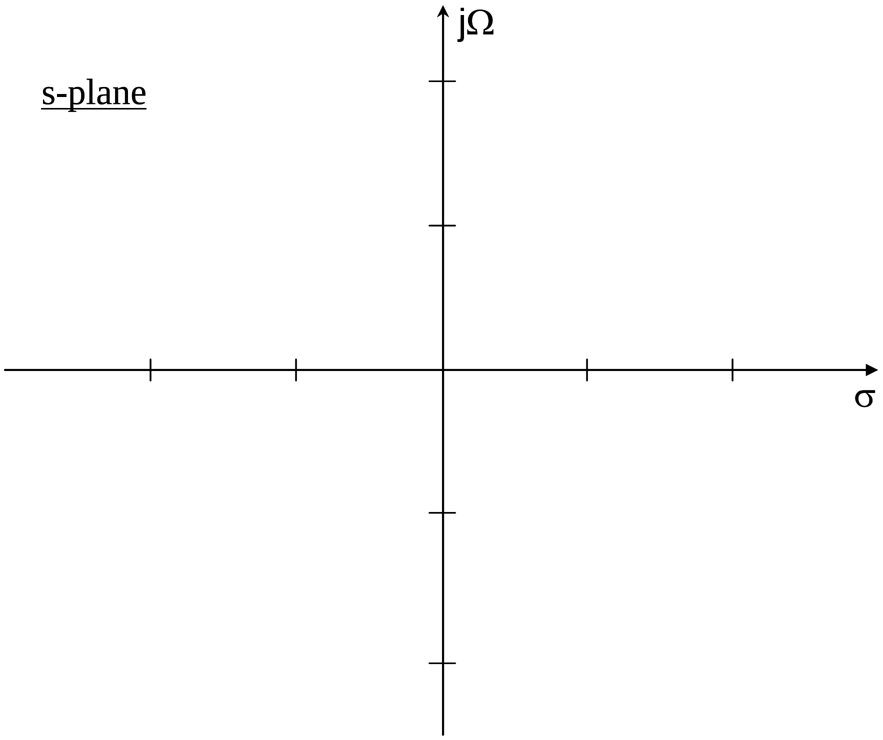


Figure 3-2: Complex 2D s-Plane in the Laplace s-Domain

Accordingly, for continuous-time systems, there are certain aspects of the system which can, in principle, be studied only in the complex analog frequency domain or continuous time domain, and for which the pertinent information becomes lost when working in the analog frequency domain. One obvious example of such a characteristic is the stability of the system, which is known to be indicated by the requirement that all system poles must lie in the left-half plane of the s-plane [40]. System stability can only be validated by examination of responses in the continuous time domain, or in the s-plane. It cannot be validated in the analog frequency or analog radian frequency domains.

3.4.1.2 **Definition of System Transfer Function** $H_a(s)$

For a given circuit, the s-domain voltage transfer function $H_a(s)$ is defined as the ratio of the system's output voltage to its input voltage in the s-domain, i.e.,

$$H_a(s) \ = \frac{V_{out}(s)}{V_{in}(s)} = \frac{B_a(s)}{A_a(s)}\,, \tag{3.17}$$

where the expressions for $V_{out}(s)$ and $V_{in}(s)$ are calculated using the Laplace transform of (3.14). $H_a(s)$ is a useful representation for understanding system characteristics in the complex analog frequency domain.

An important aspect of the transfer function $H_a(s)$ is its relationship to the system continuous time impulse response, $h_c(t)$, defined as the continuous time-domain response to the Dirac delta function $\delta(t)$, which is defined as

$$\delta(t) \ = \begin{cases} +\infty, x = 0 \\ 0, x \neq 0 \end{cases}. \tag{3.18}$$

and for which

$$\int_{-\infty}^{\infty} \delta(t)dt = 1\,. \tag{3.19}$$

The impulse response is known to completely characterize the system [40], [48]. The transfer function $H_a(s)$ is related to $h_c(t)$ via the relationship

$$H_a(s) = \mathcal{L}\{h_c(t)\} = \int_{0}^{\infty} h_c(t)e^{-st}dt\,, \tag{3.20}$$

and thus the transfer function $H_a(s)$ is the Laplace transform of the continuous-time impulse response. It can be seen, therefore, that the transfer function $H_a(s)$ also completely characterizes the system.

3.4.1.3 Relationship of the CTFT Domain Frequency Response $H_a(j\Omega)$ to the Laplace Domain Transfer Function $H_a(s)$

Similar to the definition of a Laplace-domain system transfer function $H_a(s)$ given by (3.17), it is possible to define an analog frequency transfer function $H_a(j\Omega)$, also referred to as the frequency response, in the analog frequency domain as:

$$H_a(j\Omega) \ = \frac{V_{out}(j\Omega)}{V_{in}(j\Omega)} = \frac{B_a(j\Omega)}{A_a(j\Omega)}, \tag{3.21}$$

where the expressions for $V_{out}(j\Omega)$ and $V_{in}(j\Omega)$ are calculated using the continuous-time Fourier transform of (3.1) [41],[42].

Alternatively, the analog frequency response given by (3.21) can be interpreted in terms of the Laplace transform, as the value of $H_a(s)$ evaluated where $s = j\Omega$, which corresponds exactly to the imaginary frequency axis of the s-plane of Figure 3-2 [43]. The relationship between the system's s-domain transfer function $H_a(s)$ and its analog frequency transfer function, $H_a(j\Omega)$ can thus be defined formally as:

$$H_a(j\Omega) \ = \ H_a(s)|_{s=j\Omega}. \tag{3.22}$$

3.4.2 z-Transform Domain, or z-Domain

The z-transform domain, or z-domain, relates to the discrete-time domain via the z-transform pair, defined as [41], [42]:

$$V(z) = \mathcal{Z}\{v(n)\} = \sum_{\mathrm{n}=-\infty}^{\infty} v(n)z^{-n}\,, \tag{3.23}$$

and

$$v(n) = \mathcal{Z}^{-1}\{V(z)\} = \frac{1}{2j\pi}\oint_C \ V(z)z^{n-1}dz\,. \tag{3.24}$$

For the z-transform, the region of convergence (ROC) is defined as the set of points in the z-plane for which the sequence $v(n)z^{-n}$ is absolutely summable, i.e.,

$$ROC = \left\{\sum_{\mathrm{n}=-\infty}^{\infty} |v(n)z^{-n}| < \infty\right\}. \tag{3.25}$$

The ROC is important when considering the stability and causality of systems.

3.4.2.1 Definition of the Two-Dimensional z-Plane

Note that z-domain analysis makes significant use of the concept of the complex two-dimensional (2D) z-plane, as shown in Figure 3-3. The relationship of the z-domain to the discrete frequency domain ω can be thought of as the latter mapping to the unit circle of the z-plane where $|z| = 1$. As such, the discrete frequency domain can be thought of as a subset of the z-domain, with some information being lost when working in the ω-domain in lieu of the z-domain [42].

Accordingly, for discrete systems, there are certain aspects of the system which can, in principle, be studied only in the complex z-domain or the discrete time domain, and for which the pertinent information becomes lost when working in the discrete frequency ω-domain. One obvious example of such a characteristic is the stability of the system, which is known to be indicated by the requirement that all system poles lie inside the unit circle of the z-plane [40]. System stability of discrete systems can thus only be validated by examination of responses in the discrete time domain, or in the z-plane. It cannot be validated in the discrete frequency domain.

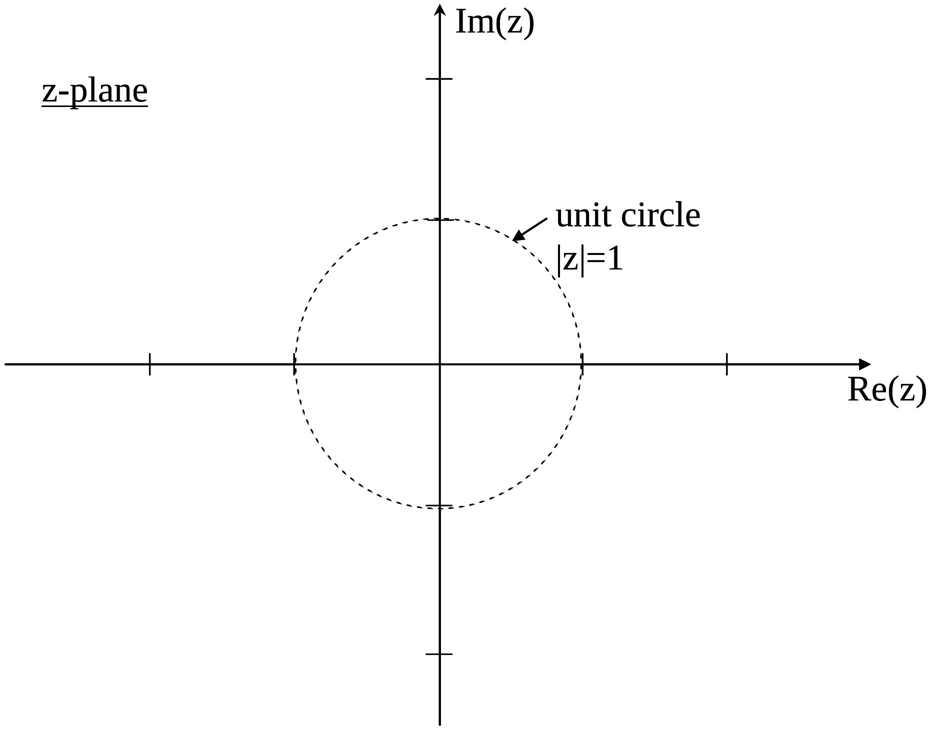


Figure 3-3: Complex 2D z-Plane in the z-Domain, Showing Unit Circle at |*z*| = *1*

3.4.2.2 Definition of System Digital Transfer Function H$_d$(z)

Analogously to the case of the Laplace domain, a system digital transfer function can be defined in the z-domain as:

$$H_d(z) = \frac{V_{out}(z)}{V_{in}(z)} = \frac{B_d(z)}{A_d(z)}, \tag{3.26}$$

where the expressions for $V_{out}(z)$ and $V_{in}(z)$ are calculated using the z-transform of (3.23). Like $H_a(s)$ in the complex analog frequency domain, $H_d(z)$ is a useful representation for understanding system characteristics in the complex discrete frequency domain.

An important aspect of the digital transfer function $H_d(z)$ is its relationship to the system discrete-time impulse response, $h_d(n)$, defined as the discrete time-domain response to the Kronecker delta function $\delta(n)$, which is defined as

$$\delta(n) = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \, . \end{cases} \tag{3.27}$$

The impulse response is known to completely characterize the system [41], [42], [49]. The transfer function $H_d(z)$ is related to $h_d(n)$ via the relationship

$$H_d(z) = \mathcal{Z}\{h_d(n)\} = \sum_{\mathrm{n}=-\infty}^{\infty} h_d(n) z^{-n} \, , \tag{3.28}$$

and thus the digital transfer function $H_d(z)$ is the z-transform of the discrete-time impulse response. It can be seen, therefore, that the digital transfer function $H_d(z)$ also completely characterizes the system.

Under certain conditions and with certain limitations, it is possible to approximate $H_d(z)$ from the knowledge of $H_a(s)$ using a variety of methods, and this process will be a major topic of this dissertation.

3.4.2.3 Relationship of the DTFT Domain Frequency Response *H$_d$(e$^{j\omega}$)* to the z-Transform Domain Digital Transfer Function *H$_d$(z)*

Similar to the definition of a z-domain system digital transfer function $H_d(z)$ given by (3.26), it is possible to define a digital frequency transfer function $H_d\left(e^{j\omega}\right)$, also referred to as the frequency response, in the discrete frequency domain as:

$$H_d\left(e^{j\omega}\right) = \frac{V_{out}(e^{j\omega})}{V_{in}(e^{j\omega})} = \frac{B_a\left(e^{j\omega}\right)}{A_a(e^{j\omega})}, \tag{3.29}$$

where the expressions for $V_{out}(e^{j\omega})$ and $V_{in}(e^{j\omega})$ are calculated using the discrete-time Fourier transform of (3.5) [41],[42].

Alternatively, the discrete frequency response given by (3.29) can be interpreted in terms of the z-transform, as the value of $H_d(z)$ evaluated where $z = e^{j\omega}$, which corresponds exactly to the unit circle of the z-plane of Figure 3-3 [43][8]. The relationship between the system's z-domain transfer function $H_d(z)$ and its ω-domain frequency response, or frequency transfer function, $H_d\left(e^{j\omega}\right)$ can be defined formally as:

$$H_d\left(e^{j\omega}\right) = H_d(z)|_{z=e^{j\omega}}. \tag{3.30}$$

3.4.3 *$H_a(s)$* to *$H_d(z)$* Transformation Methods

Direct conversion from the s-domain to the z-domain is useful for modifying system transfer functions for subsequent examination and analysis. Techniques for this conversion are well-established in filter theory, and these techniques can be applied directly to signal integrity problems as well [41].

An exact mapping from the s-plane to the z-plane is established through the relationship

$$z = e^{sT_s}, \tag{3.31}$$

but a difficulty arises from the fact that the frequency response in the discrete frequency ω-domain is periodic. Essentially, mapping between the continuous- and discrete-frequency domains involves mapping the $j\Omega$ axis from the complex s-plane to the unit circle in the complex z-plane. Note that this mapping would be periodic, and thus non-unique, at continuous frequencies Ω that correspond to discrete frequencies greater than $\omega = 2\pi$.

Three widely-used s-domain to z-domain transformations that can be readily applied to filter design and signal integrity problems are the bilinear transformation, the matched-Z transformation, and the impulse invariant method transformation [41]–[43].

---

[8] The $j\Omega$ axis in the s-plane of Figure 3-2 is warped into a circle with radius equal to 1 in the z-plane of Figure 3-3 using the transformation $z = e^{j\omega} = e^{j\Omega T_s}$.

These are particularly useful if the performance specifications are given in the frequency domain, which is usually the case in filter design problems.

3.4.3.1 Bilinear Transformation

The bilinear transformation (BLT) can be used to map between the s-domain and z-domain, in either direction. For this transformation, the analog poles and zeros are mapped to discrete poles and zeros. Because it was developed for filter applications, the bilinear transformation carefully maintains the magnitude response of the analog filter between $H_d(z)$ and $H_a(s)$, although the frequencies become distorted at high frequencies [41].

The bilinear transformation is based upon a 1st-order Pade approximation [43]. Recall that the exact map from the s- to the z-domain is given by (3.31). The 1st-order Pade approximation yields the inverse bilinear transformation [41],

$$z = e^{sT_s} \approx \frac{1 + \frac{T_s}{2} s}{1 - \frac{T_s}{2} s} \; . \tag{3.32}$$

Solving (3.32) for s yields the bilinear transformation [41]:

$$s = \frac{2}{T_s}\left(\frac{z-1}{z+1}\right), \tag{3.33}$$

and thus the transformation from $H_a(s)$ to $H_d(z)$ using the BLT becomes

$$H_d(z) = H_a(s)\big|_{s=\frac{2}{T_s}\left(\frac{z-1}{z+1}\right)} \; . \tag{3.34}$$

Figure 3-4 represents graphically the relationships established between the s-plane and z-plane by the bilinear transform. The non-linear bilinear transform of (3.33) has the property that the entire $j\Omega$ axis of the s-plane maps onto the unit circle of the z-plane, with frequency distortion at high frequencies but note that, unlike the map $z = e^{sT_s}$, which implies $\omega = \Omega T_s,$ the mapping is unique, with no periodic mapping [41].

Another property of the bilinear transformation is that the left-half plane of the complex s-plane maps to the interior of the unit circle in the complex z-plane. The important implication here is that stability information for a given system is contained in the s- and z-domains. The stability information is lost when working in the analog and discrete frequency domains Ω and ω.

Applying the bilinear transformation to the analog and discrete frequency domains results in the following relationships [43]:

$$\Omega = \frac{2}{T_s} tan(\frac{\omega}{2}) \, , \tag{3.35}$$

and

$$\omega = 2 \, tan^{-1}(\frac{\Omega T_s}{2}) \; . \tag{3.36}$$

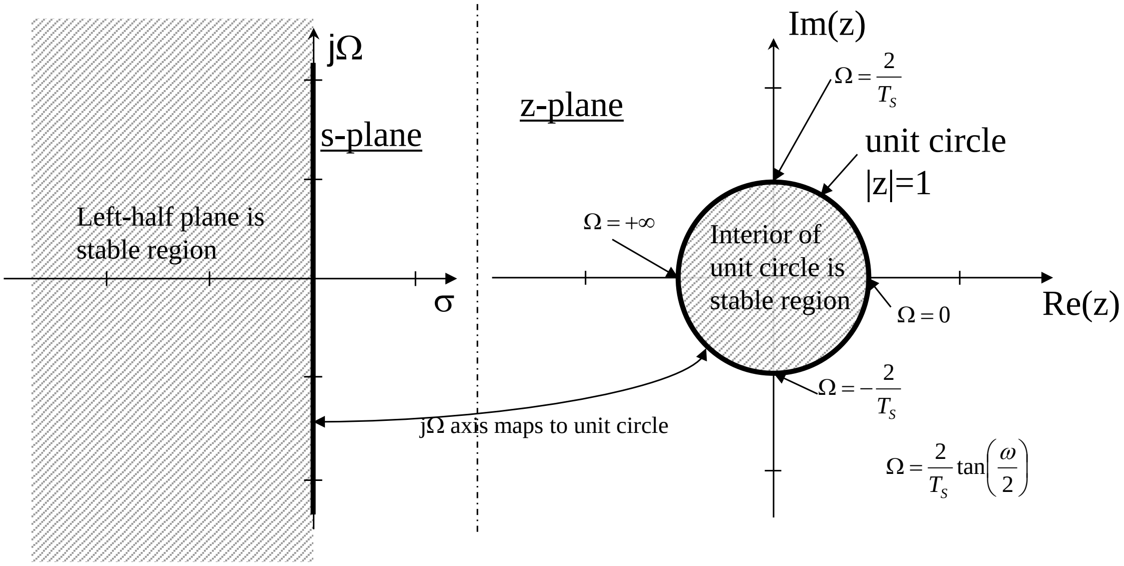


Figure 3-4: Relationship Between Complex s-Plane and Complex z-Plane for the Bilinear Transformation

Note the distinction between the discrete-to-continuous map given by the non-linear bilinear transform and the linear map, namely $\omega = \Omega T_s$. Note from the z-plane diagram in Figure 3-4 that the s-plane frequencies Ω do not translate to the z-plane frequencies ω via the linear map.

For filter design problems, the frequency distortion that occurs at high frequencies can be compensated for at certain critical design frequencies by pre-warping these frequencies prior to applying the bilinear transform, using the formulas above. For filter design problems, these critical design frequencies are usually those which define the pass bands and stop bands for the filter [41].

3.4.3.2 Matched-Z Transformation

The matched-Z transformation can also be used to map between the s-domain and z-domain, in either direction. Like the bilinear transformation, the analog poles and analog zeros are mapped to discrete poles and zeros. Because it was also developed for filter applications, the matched-Z transformation carefully maintains the magnitude response of the analog filter between $H_d(z)$ and $H_a(s)$, but the frequencies become distorted at high frequencies [43].

Unlike the bilinear transform, for the matched-Z transform the mapping between the z- and s-domains does follow the exact map given by (3.31) above. The matched-Z transform is most easily applied with $H_d(z)$ and $H_a(s)$ expressed in factored form, namely, where

$$H_d(z) = K_d \left[\frac{\prod_{k=1}^{M} \left(\mathrm{z} - \mathrm{e}^{\mathrm{b_k T_s}}\right)}{\prod_{k=1}^{N} \left(\mathrm{z} - \mathrm{e}^{\mathrm{a_k T_s}}\right)} \right]; \; N \geq M\,, \tag{3.37}$$

where K_d is a gain constant, and

$$H_a(s) = K_a \left[\frac{\prod_{k=1}^{M} (\mathrm{s} - \mathrm{b_k})}{\prod_{k=1}^{N} (\mathrm{s} - \mathrm{a_k})} \right]; \; N \geq M\,, \tag{3.38}$$

where K_a is also a gain constant. The gain constants K_d and K_a must be matched at some frequency of interest when using the matched-Z transform.

The general procedure for using the matched-Z transformation to move from the s-domain to the z-domain is thus [43]:

1. Express $H_a(s)$ in pole-zero factored form. It is assumed all poles have a frequency of less than the Nyquist frequency $\omega = \pi$. The poles are mapped according to the exact map (3.31).

2. Finite zeros of $H_a(s)$ are mapped according to the exact map (3.31) if they have frequency less than the Nyquist frequency $\omega = \pi$, otherwise they are mapped to $z = -1$.
3. The gain K_d of $H_d(z)$ of (3.37) is adjusted to match the gain K_a of $H_a(s)$ of (3.38) at some frequency of interest. For low-pass filters, this frequency is often chosen at DC, where $s = 0$ and $z = 1$. For high-pass filters, this frequency is often chosen at infinite frequency where $s = \infty$ and, by convention, $z = -1$.

Figure 3-5 shows the relationship between the complex s- and z-planes for the matched-Z transformation. Note that the matched-Z transformation is not a unique map, in that frequencies above the Nyquist frequency are all mapped to the Nyquist frequency. This is a controversial aspect of the matched-Z transform, but in reality differs little from the bilinear transform’s frequency distortion at high frequencies. The matched-Z transform has the advantage that it follows the exact map of (3.31) for much of the frequency spectrum.

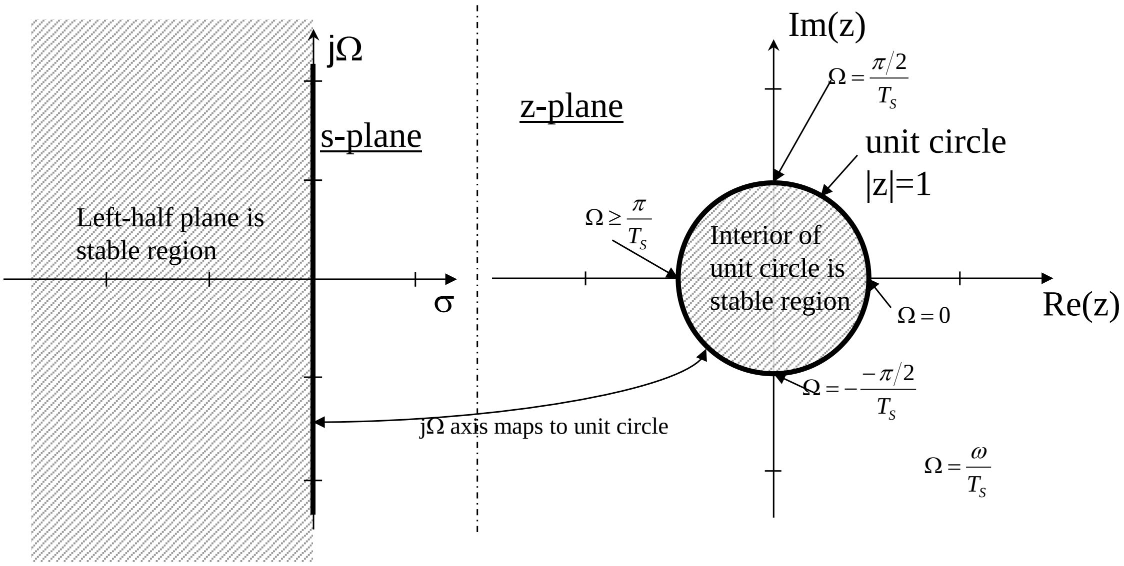


Figure 3-5: Relationship Between Complex s- and z-Planes for the Matched-Z Transformation

3.4.3.3 Impulse Invariant Method Transformation

The impulse invariant transform (IIT) is a third well-known method for mapping from the s- domain and z-domain, in either direction [42], [41], [43]. For this transformation, the analog poles are directly mapped to discrete poles via the relation

(3.31), just as is done for the matched-Z transformation described previously. The zeros, however, are mapped in a less-straightforward fashion, which is one of the main drawbacks of the method for some applications, such as filter design, where matching the frequency responses $H_a(s)$ and $H_d\left(e^{j\omega}\right)$ is important. For certain applications, such as where good time-domain performance is important, for example, the IIT can work very well indeed [42].

The impulse invariant method transform carefully maintains the relationship between the system's continuous-time impulse response $h_c(t)$, and the sampled version of this $h_d(n)$, where the relation between them is expressed as:

$$h_d(n) = T_s h_c(t)|_{t=nT_s} = T_s h_c(nT_s),\ n = 0, 1, 2, \dots, \tag{3.39}$$

and T_s is once again the sampling interval. For a continuous-time system with Laplace-domain transfer function $H_a(s)$ expressed in the pole-residue form

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = \sum_{k=1}^{N} \frac{c_k}{s - a_k}, \tag{3.40}$$

application of the IIT results in [41]

$$H_d(z) = \sum_{k=1}^{N} \frac{T_s c_k z}{z - e^{a_k T_s}}\ . \tag{3.41}$$

In general, the matched-Z transform and the bilinear transform can better handle lower sampling rates than older s- to z-domain transformation methods, such as the impulse invariant method, and have largely replaced the latter in filter design [43]. Because of its prevalence and wide adoption, however, it is useful to utilize the IIT for comparison purposes.

3.4.4 Time-Frequency Uncertainty Principle for Quantities Related by the Fourier Transform

One of the limiting concepts when transforming between the various time and frequency domains, both analog and digital, is the time-frequency uncertainty principle, which is mathematically inherent to pairs of functions related by Fourier transforms [50], [51]. Perhaps the best-known example of such an uncertainty principle is that expressed by Heisenberg in the field of quantum mechanics, but it can be shown to be inherent to all systems which have a function $v(t)$ with a Fourier transform $V(f)$, as expressed in the relation of (3.1).

The uncertainty principle for the case of time-frequency domains, such as those being considered in this project, can be expressed mathematically as [51]:

$$D_o\big(v(t)\big)D_o\big(V(f)\big) \geq \frac{1}{16\pi^2}\,, \tag{3.42}$$

where

$$D_o\big(v(t)\big) = \int_{-\infty}^{\infty} t^2 |v(t)|^2 dt\,. \tag{3.43}$$

The significance is the realization that the resolution in the time and frequency domains is fundamentally limited; this realization may be more important than the quantitative result. The implication is that, while it may be possible to know with great precision the value of a function $v(t)$ in the time domain, doing so will limit the precision for which $V(f)$ can be known in the frequency domain, and vice versa.

The uncertainty principle was expressed in (3.42) in terms of the continuous-time Fourier transform (CTFT), but it should be clear that it applies to the DTFT pair as well. The time-frequency uncertainty principle, in a variety of forms, has pragmatic implications for the expected performance of systems related to signal processing, filter design, image processing, and a variety of engineering and scientific applications, as is evident by mangy patents and publications, e.g., [52], [53], [54], [55], [56].

3.5 Circuit Theory

Signal integrity engineers make significant use of circuit theory for their analyses. The SPICE simulators make use of high level programming languages, such as SPICE or ASTAP, for their input [20], [21]. These languages follow the conventions of standard lumped-element circuit theory as their basis, adding special functions and formulations for transistors, transmission lines, and other special-purpose elements. Distributed-element circuit theory is more generalized, and includes such concepts as transmission line theory and scattering-parameter theory [57]. Distributed-element circuit theory is required when the underlying assumptions required by lumped-element theory break down. An understanding of the fundamental assumptions underlying both lumped- and distributed-element circuit theory is thus required for using signal integrity tools and interpreting the results properly.

3.5.1 Lumped-Element Circuit Theory

Lumped-element circuit theory is familiar to all electrical engineers, both in the continuous time and analog frequency domains. Most signals encountered in signal integrity analysis are considered in the context of the continuous time domain. Lumped-element analysis, however, has fundamental limitations which preclude utilizing it exclusively for signal integrity analysis [57].

Lumped-element theory can be derived directly from Maxwell’s equations, and greatly simplifies the analysis of circuits under the right conditions. In particular, it allows for algebraic analysis of circuits being excited by sinusoidal stimuli in the steady state case, instead of differential equation solutions in the more general, time-domain case. Under these conditions, it is possible to covert easily between the analog radian frequency Ω-domain and the continuous time domain.

The application of lumped-element theory also requires some underlying assumptions [57]. One important assumption that is particularly relevant to signal integrity engineering is that the wavelength of the signals in question must be very long relative to the circuit’s physical dimensions. A widely-used guideline is that the circuit dimensions must be smaller than about 1/10 of a wavelength [4], [6]. This is true for circuits with moderate physical size at very low frequencies, or for circuits with small physical size at moderate speeds. The theory breaks down, however, for circuits with huge physical size at low speeds, and circuits with moderate physical size at high speeds.

As an example, if one considers an electrical power utility’s distribution network, the wavelength of a 60-Hz signal is 5000 km. Thus, for the case of 60 Hz, a transmission network with a length of over 500 km cannot be considered using lumped-element circuit theory, but the wiring network inside of a person’s house can be considered using lumped-element theory.

As a second example, if one considers a printed wiring board (PWB) that is 32 cm by 28 cm (a standard dimension of an ATCA form factor computer blade) a 1-GHz signal has a wavelength of about 14 cm. Thus physical interconnects of several centimeters or larger cannot be treated using lumped theory, but physical interconnects at the chip-level can be considered using lumped-element theory.

3.5.2 Transmission Line Theory

The second example above shows that lumped-element theory is not adequate for signal integrity problems at the PWB level, although it may be adequate at the 1st-level package or chip level, depending upon the frequency. Accordingly, signal integrity engineers have used a combination of lumped-and distributed-element circuit theory in their analyses since computers started to utilize signals in the few tens of MHz [1]. SPICE simulators were enhanced to contain built-in transmission line algorithms. Further, the signals found in computers are typically not purely sinusoidal, but rate pulse trains consisting of pseudo-random patterns of 1's and 0's. These signals by nature have wide spectra, containing not just their fundamental frequency defined by the baud rate, but also having significant harmonic content [4].

The most familiar form of distributed circuit element is the transmission line. Transmission line theory is well-developed mathematically and generally compatible with lumped-element concepts. As such, it is a natural extension of lumped-element theory that is easily understandable to most electrical engineers.

Figure 3-6 shows the basic definition of a transmission line element [6]. The definition of a transmission line consists of the distributed resistance parameter R, distributed inductance parameter L, distributed shunt conductance parameter G, and the distributed capacitance C, respectively, all per unit length. In general, the $RLGC$ parameters are frequency-dependent, and this dependence must be accounted for when developing transmission line models for SPICE simulation [6], [57]. In the figure, an infinitesimally short length Δz is shown.

R represents the conductor loss of the line, while G represents the dielectric loss. Transmission lines allow engineers to mathematically account for three real effects, propagation delay along the interconnect, signal attenuation in the interconnect, and loading effects of the interconnect. Two derived quantities from the $RLGC$ representation are [6]:

$$Z_o = \sqrt{\frac{R + j\Omega L}{G + j\Omega C}}, \tag{3.44}$$

where Z_o is the characteristic impedance of the transmission line, and

$$\gamma = \sqrt{(R + j\Omega L)(G + j\Omega C)} = \alpha + j\beta\ , \tag{3.45}$$

where γ is the transmission line propagation constant. Again, note that Z_o and γ are frequency-dependent [6].

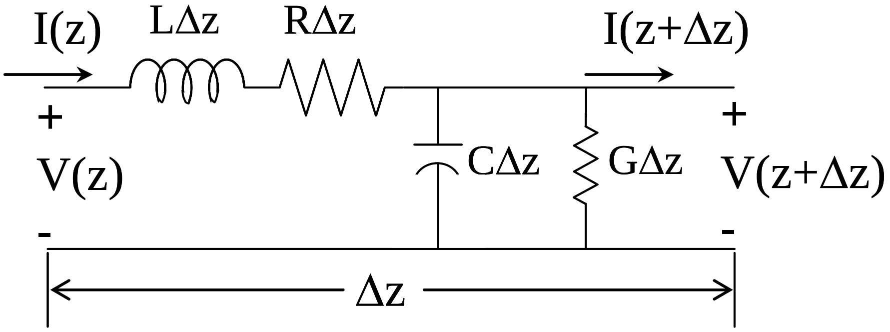


Figure 3-6: Basic Definition of a Transmission Line Element

In the above relations, R, L, G, and C, are real numbers, while Z_o and γ are complex numbers. Thus specifying the $RLGC$ parameters is completely equivalent to specifying Z_o and γ, and both conventions are used depending upon the context. Z_o is an indicator of line loading and impedance mismatch effects, while γ is an indicator of propagation speed and attenuation effects, with the “attenuation factor” α representing attenuation and the “phase constant” β representing propagation velocity.

The $RLGC$ parameters describe the line itself. There are some additional parameters used in transmission line theory which describe the relationship of the line to circuit elements to which the line interfaces. The reflection coefficient is defined as [6]:

$$\Gamma_L = \frac{V_-}{V_+} = \frac{Z_L - Z_o}{Z_L + Z_o}, \tag{3.46}$$

where Z_L is the impedance of the load attached to the end of the line.

Similarly, a transmission coefficient can be defined for the terminated transmission line [19]:

$$T_L = \frac{V_+ + V_-}{V_+} = 1 + \Gamma_L = \frac{2Z_L}{Z_L + Z_o}, \tag{3.47}$$

and can be seen to be the ratio of the total voltage at the load to the incident voltage at the input to the line.

The transmission line's input impedance Z_{input} is also load-dependent and can be expressed as [6]:

$$\frac{Z_{input}}{Z_o} = \frac{\frac{Z_L}{Z_o} + tanh((\alpha + j\beta)l)}{1 + \frac{Z_L}{Z_o} tanh((\alpha + j\beta)l)} = \frac{\frac{Z_L}{Z_o} + tanh(\gamma l)}{1 + \frac{Z_L}{Z_o} tanh(\gamma l)}, \tag{3.48}$$

where l is the length of the transmission line terminated in load impedance Z_L. For equations (3.44) through (3.48), sinusoidal steady-state excitation is assumed, and both the input and output parameters are all frequency-dependent.

3.5.3 Scattering Parameters

Another familiar concept of distributed circuit theory is the scattering parameter, or s-parameter. Scattering parameters are closely related to the concepts of reflection coefficients and transmission coefficients, as described in the previous section for transmission lines. Scattering parameters are particularly useful for circuit element and interconnect characterization using empirical measurements or simulation at high frequencies, where the concepts of voltage and currents become spatially dependent, and open- and short-circuit terminations become difficult to implement over any significant frequency range. In contrast, s-parameter measurements are made with the network terminated in non-reflecting or matched loads, which are easier to implement over practical frequency ranges [19].

Figure 3-7 shows a two-port network with incident and emerging waves at each port. The matrix of scattering parameters, or scattering matrix, is defined by this configuration as [57]:

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \tag{3.49}$$

or

$$\underline{b} = \underline{S}\,\underline{a}\,, \tag{3.50}$$

where $\underline{S}$ is the scattering matrix, $\underline{a}$ is matrix of normalized incident waves, and $\underline{b}$ is the matrix of normalized emerging waves.

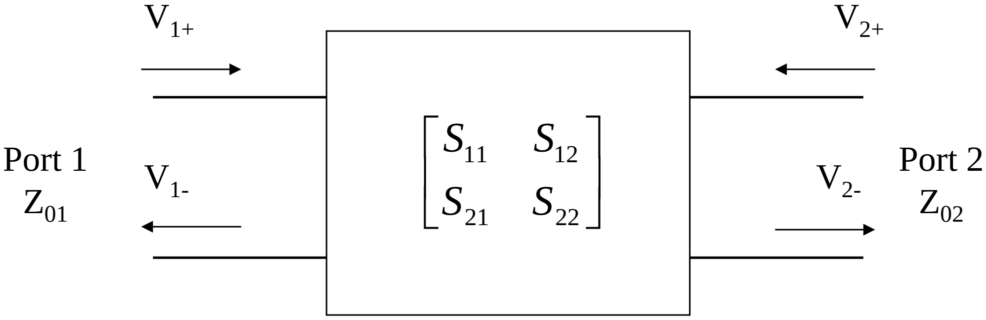


Figure 3-7: Definition of Scattering Parameters for a Two-Port Network

The elements of $\underline{a}$ and $\underline{b}$ are normalized to the impedance of the corresponding port as shown in Figure 3-7, namely [19],

$$a_n = \frac{V_{n+}}{\sqrt{Z_{on}}}, \qquad b_n = \frac{V_{n-}}{\sqrt{Z_{on}}}\ . \tag{3.51}$$

The elements of $\underline{S}$ for a given circuit are frequency-dependent and are determined experimentally by vector network analyzer (VNA) measurements or, if in a pre-hardware phase of the design, by simulation with a full-wave electromagnetic field solver using assumed physical and material parameters for the circuit. As such, for a two-port network, the scattering matrix elements are defined as [19] :

$$S_{11} = \left.\frac{b_1}{a_1}\right|_{a_2=0}, \quad S_{12} = \left.\frac{b_1}{a_2}\right|_{a_1=0}, \quad S_{21} = \left.\frac{b_2}{a_1}\right|_{a_2=0}, \quad S_{22} = \left.\frac{b_2}{a_2}\right|_{a_1=0}. \tag{3.52}$$

As can be seen from the above equations and Figure 3-7, as an example, S_{11} would be measured or calculated by establishing that $a_2 = 0$ and measuring the reflected wave at port 1 when an incident wave a_1 on port 1 is used to excite the network. This process is implemented by removing any source on port 2 and terminating port 2 such that nothing is reflected back into the network at the port 2 interface.

From these definitions, it can be seen that the S_{11} and S_{22} parameters are essentially reflection coefficients similar to that defined for the transmission line previously, while the S_{12} and S_{21} parameters are related to the transmission coefficient concept defined previously. It should also be noted that, conceptually, the two-port network can be generally expanded to an N-port network. This is commonly done for differential pairs (a four-port network), or for expanded channel models where crosstalk is to be evaluated with both victim and aggressor lines present [38]. The resulting channel models can become quite complex, and care must be taken by the signal integrity engineers to balance the sophistication of the model with the computational resource required and available, such that the resulting simulation effort remains manageable.

3.5.4 Relationship of Transfer Function $H_a(s)$ to Scattering Parameters

For a given circuit, the s-domain voltage transfer function $H_a(s)$ can be found from the interconnect's scattering parameters if the circuit's source and load impedances are also known [58]. The conversion is performed as:

$$H_a(j\Omega) = H_a(f) = \frac{S_{21}(1+\Gamma_L)(1-\Gamma_S)}{2(1-S_{22}\Gamma_L)(1-\Gamma_{in}\Gamma_S)}, \tag{3.53}$$

where the s-parameters are as defined previously, the Γ_L and Γ_S are the reflection coefficients looking into the source and load, respectively, and Γ_{in} is the reflection coefficient looking into the two-port network with the known load termination attached, given as:

$$\Gamma_{in} = S_{11} + S_{12}S_{21}\frac{\Gamma_L}{1-S_{22}\Gamma_L}. \tag{3.54}$$

This relationship is important for cases where it is desired to identify the system model using scattering parameters, which is commonly done for signal integrity applications, and subsequently study the system characteristics by analyzing the transfer function or frequency response characteristics in the frequency domain, which is commonly done in DSP applications.

3.5.5 Circuit Frequency Scaling Techniques

Circuit scaling techniques have been used historically in the design of analog filters [59]. These techniques, which can involve either scaling of frequencies or circuit element impedances, allow the modification of a known network having a given frequency response to a different network having a frequency response which more closely matches a desired response. Historically, these modifications were intended to be implemented in the circuit design, but the techniques can also work to improve the ease of numerical calculations or simply the analysis of the network, especially when the calculations become cumbersome.

Of these techniques, those related to frequency scaling are particularly useful because they are easy to apply in the s-domain. As such, they become practical for situations where the system transfer function $H_a(s)$ is known, and are directly applicable when $H_a(s)$ is given in the rational form $B_a(s)/A_a(s)$. Better still, if $H_a(s)$ is also expressed in the pole-residue form given by (3.40), then the scaling become extremely intuitive, and results in simple scaling to the pole, zero, and residue coefficients.

To see this, recall the partial fraction expansion form of $H_a(s)$, expressed as:

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = \sum_{k=1}^{N} \frac{c_k}{s - a_k}. \tag{3.55}$$

From frequency-scaling theory [59], it is known that a transfer function $H_a(s)$ can be scaled down in frequency by a factor of k_f by applying the transform $s \rightarrow sk_f$ on the s-domain variable, as follows:

$$H_{proto}(s) = H_a(s)|_{s \rightarrow sk_f} = \sum_{k=1}^{N} \frac{c_k}{sk_f - a_k} = \sum_{k=1}^{N} \frac{\frac{c_k}{k_f}}{s - \frac{a_k}{k_f}}. \tag{3.56}$$

Here the effect of the scaling factor can be clearly seen; it scales down the residue variables c_k and the pole variables a_k by a factor of k_f. The net effect of this is that $H_{proto}(s)$ has the same frequency response as $H_a(s)$, but over a frequency range that is scaled down by a factor of k_f.

The motivation for doing the scaling operation of (3.56) lies in the inherent problems in dealing with $H_a(s)$ at high frequencies. Specifically, for systems of the type used in signal integrity analysis the poles' and residues' values can become extremely large, often in the range of a few GHz or tens of GHz. This creates problems in making numerical calculation using computer routines, as the numerical precision limitations of the computers become relevant. By scaling the residue and pole values, subsequent calculation of the pole-zero factored form of $H_{proto}(s)$ is significantly easier to handle computationally. This leads to

$$H_{proto}(s) = \frac{B_{proto}(s)}{A_{proto}(s)} = \frac{K_{ap}(s - b_{p1})(s - b_{p2}) \cdots (s - b_{pM})}{(s - a_{p1})(s - a_{p2}) \cdots (s - a_{pN})}, \tag{3.57}$$

where the b_{pk} and a_{pk} terms are the zeros and poles of $H_{proto}(s)$, N is the order of the model, and M is the number of finite zeros. The function is then scaled back up to the original frequency range, by applying the reverse transformation $s \to s/k_f$ on the s-domain variable, as follows [59]:

$$H_a(s) = H_{proto}(s)\Big|_{s \to \frac{s}{k_f}} = \frac{K_{ap}(\frac{s}{k_f} - b_{p1})(\frac{s}{k_f} - b_{p2}) \cdots (\frac{s}{k_f} - b_{pM})}{(\frac{s}{k_f} - a_{p1})(\frac{s}{k_f} - a_{p2}) \cdots (\frac{s}{k_f} - a_{pN})}, \tag{3.58}$$

Comparing (3.57) and (3.58) it is easily seen that the zeros and poles of $H_a(s)$ are the zeros b_{pk} and poles a_{pk} of $H_{proto}(s)$ scaled by the factor k_f, that is,

$$b_{ak} = b_{pk} k_f, \tag{3.59}$$

and

$$a_{ak} = a_{pk} k_f. \tag{3.60}$$

Note that it still may not be computationally easy to calculate the overall function $H_a(s)$, due to complications which may arise from the calculation of K_a, but at least the process above enables the direct identification of the poles and zeros of $H_a(s)$.

3.5.6 State-Space Representations of Systems

As stated previously, the transfer function representation for the system allows its evaluation in the frequency domain. Similarly, the state space representation allows for the system's evaluation in the time domain. This representation expresses the model in terms of its input and output signals, as well as internal variables known as state variables [40], [49]. One advantage of the state space representation is that it lends itself to finding time-domain solutions, and thus is often the basis for many time-domain simulation algorithms [60].

Like transfer function representations, state space representations can be created for both continuous and discrete-time systems. As such, discrete-time representations can often be created to approximate continuous-time systems. This is particularly useful for signal integrity analysis, which is often considered in the time-domain, yet the underlying simulation algorithms operate essentially on sampled versions of the continuous-time signals [60]. As such, the relationship between the discrete-time representation of the system and its continuous-time counterpart may be of interest.

3.5.6.1 State-Space Representation for Continuous-Time Systems

For the continuous-time case, the state space representation of the system consists of the following [48]:

$$\underline{\dot{x}}(t) = \underline{F_{ss}}\,\underline{x}(t) + \underline{G_{ss}}\,\underline{u}(t)\,, \text{ and} \tag{3.61}$$

$$\underline{y}(t) = \underline{H_{ss}}\,\underline{x}(t) + \underline{J_{ss}}\,\underline{u}(t)\,, \tag{3.62}$$

where $\underline{x}(t)$ is a vector of state variables, $\underline{u}(t)$ is a vector of inputs, and $\underline{y}(t)$ is the vector of outputs. $\underline{F_{ss}}$ is the state matrix, $\underline{G_{ss}}$ is the input matrix, $\underline{H_{ss}}$ is the output matrix, and $\underline{J_{ss}}$ is the feed-forward matrix.

The matrices can be formulated by applying the transfer function $H_a(s)$ to generic input and output functions $U(s)$ and $Y(s)$, respectively, and writing the algebraic equations in the s-domain [40]. From the algebraic representation, differential equations can be written in the time domain, using the inverse Laplace transform. These differential equations lead directly to (3.61) and (3.62) above.

3.5.6.2 State-Space Representation for Discrete-Time Systems

Analogously, for the discrete-time case, the state space representation of the system consists of the following [48]:

$$\underline{x}(k+1) = \underline{\Phi_{\mathrm{ss}}}\,\underline{x}(k) + \underline{\Gamma_{\mathrm{ss}}}\,\underline{u}(k)\,, \text{ and} \tag{3.63}$$

$$\underline{y}(k) = \underline{H_{ss}}\,\underline{x}(k) + \underline{J_{ss}}\,\underline{u}(k)\,, \tag{3.64}$$

where again $\underline{x}(k)$ is a vector of state variables, $\underline{u}(k)$ is a vector of inputs, and $\underline{y}(k)$ is the vector of outputs. $\underline{\Phi_{ss}}$ is the state matrix, $\underline{\Gamma_{ss}}$ is the input matrix, $\underline{H_{ss}}$ is the output matrix, and $\underline{J_{ss}}$ is the feed-forward matrix.

The matrices can be formulated by applying the transfer function $H_d(z)$ to generic input and output functions $U(z)$ and $Y(z)$, respectively, and writing the algebraic equations in the z-domain [40]. From the algebraic representation, difference equations can be written in the time domain, using the inverse z-transform. These difference equations lead directly to (3.63) and (3.64) above.

3.5.6.3 Relationship Between the Continuous-Time and Discrete-Time State-Space Representations

The discrete-time state-space representation of (3.63) and (3.64) above can, in principle, be derived directly from the z-Domain transfer function $H_d(z)$, if it is known. It is also possible to create the discrete-time representation from the continuous time representation of (3.61) and (3.62), using the following relationships [49]:

$$\underline{\Phi_{\mathrm{ss}}} = e^{\underline{F_{ss}}\,T_s}, \text{ and} \tag{3.65}$$

$$\underline{\Gamma_{\mathrm{ss}}} = \left[\int_0^{T_s} e^{\underline{F_{ss}}\eta}\, d\eta\right] \underline{G_{ss}}\,, \tag{3.66}$$

where $\underline{F_{ss}}$, $\underline{G_{ss}}$, $\underline{\Phi_{\mathrm{ss}}}$, and $\underline{\Gamma_{\mathrm{ss}}}$ are the matrices defined in (3.61) and (3.63) previously, and T_s is the sampling interval assumed in the conversion from the continuous-time to discrete-time system. This relationship is the basis for the MATLAB algorithm timeresp.m, which will be used extensively in later sections of this project to evaluate the time-domain response of continuous-time systems [61].

3.5.6.4 Calculating the System Transfer Function from the State-Space Representation

Just as the transfer function representation can be used to formulate the state-space representation, it is also possible to reverse the process. The transfer function $H_a(s)$ can be calculated from the state equations of (3.61) and (3.62) for the continuous-time case as follows [49]:

$$H_a(s) = \frac{Y(s)}{U(s)} = \underline{H_{ss}}\left(s\underline{I} - \underline{F_{ss}}\right)\underline{G_{ss}} + \underline{J_{ss}}\ , \tag{3.67}$$

where $\underline{I}$ is the identity matrix and the other matrix variables are as defined previously in (3.61) and (3.62).

Similarly, the transfer function $H_d(z)$ can be calculated from the state equations of (3.63) and (3.64) for the discrete-time case, as follows [49]:

$$H_d(z) = \frac{Y(z)}{U(z)} = \underline{H_{ss}}\left(z\underline{I} - \underline{\Phi_{ss}}\right)\underline{\Gamma_{ss}} + \underline{J_{ss}}\,, \tag{3.68}$$

where $\underline{I}$ is the identity matrix and the other matrix variables are as defined previously in (3.63) and (3.64).

3.6 Time-Domain Simulation Methods

The primary tool of signal integrity analysis is time-domain transient simulation. Historically, these tools have been SPICE simulator tools with enhanced algorithms to account for non-linear devices (usually transistors and diodes) and for distributed circuit elements for transmission line interconnects. SPICE-based tools date back to the early 1970’s and typically use nodal equations of various types and Gaussian elimination methods for their solution [20], [21]. Essentially, this amounts to a time-domain convolution method. The driver and receiver models are preferably non-linear device models that are technology-dependent, and usually consist of hundreds of transistors. This results in overall simulation models of great complexity, resulting in long simulation times.

More recently, the desire to simulate high-speed SERDES links, the need to including the feed-forward equalization (FFE) and decision-feedback equalization (DFE) filters in the driver and receiver models, the desire to invoke a frequency-dependent interconnect model, and a desire to include the associated jitter terms in a single comprehensive model, have all led to the development of a new class of transient simulators known generally as statistical simulation analysis (SSA) programs [38]. Unlike SPICE, these programs rely on fast convolution methods and conversion between the discrete time and discrete frequency domains using FFT methods, as well as superposition methods to formulate the various output waveforms at the end of the simulation.

3.6.1 SPICE Simulation Methods

SPICE simulators have been in use since the early 1970s [20], [21]. Originally, it was not unusual for various semiconductor companies to develop their own SPICE simulators to match the CMOS or bipolar technologies of their manufacturing lines, leading to the development of multiple simulation programs. Today, there are several prevailing commercially-available simulation codes offered by EDA companies, and micro-electronics companies which own foundries formulate their models into one or more generally-accepted formats.

Although differences still exist among various SPICE simulator engines, the underlying SPICE algorithms are based on the general principle depicted in Figure 3-8 [62]. SPICE simulators generally have three analysis modes - DC analysis, AC analysis, and transient analysis. Of these, transient analysis is most import for signal integrity applications, although the DC analysis solution is an inherent part of the AC analysis and transient analysis processes as well.

Referring to Figure 3-8, a DC analysis for a linear circuit is performed using blocks 3 and 4. For a non-linear DC analysis, blocks 1-6 are required. For a transient analysis with linear circuits, the outer loop is executed, ignoring blocks 2 and 5. For a transient analysis with non-linear circuits, the entire flow is required, with blocks 2-6 being completed at each point in time $t(n)$.

Note in particular the adaptive time step of block 7. SPICE dynamically adjusts the time step $h(n)$ to reduce long simulation times and to improve accuracy. The basis for adapting the time step is the rate at which the voltages and currents being calculated in the nodal matrix are changing. This adaptive time step has great implications when applying DSP theory to SPICE-generated waveforms, as DSP theory generally assumes a uniform sampling interval T_s for the application of various DSP methods [41], [42].

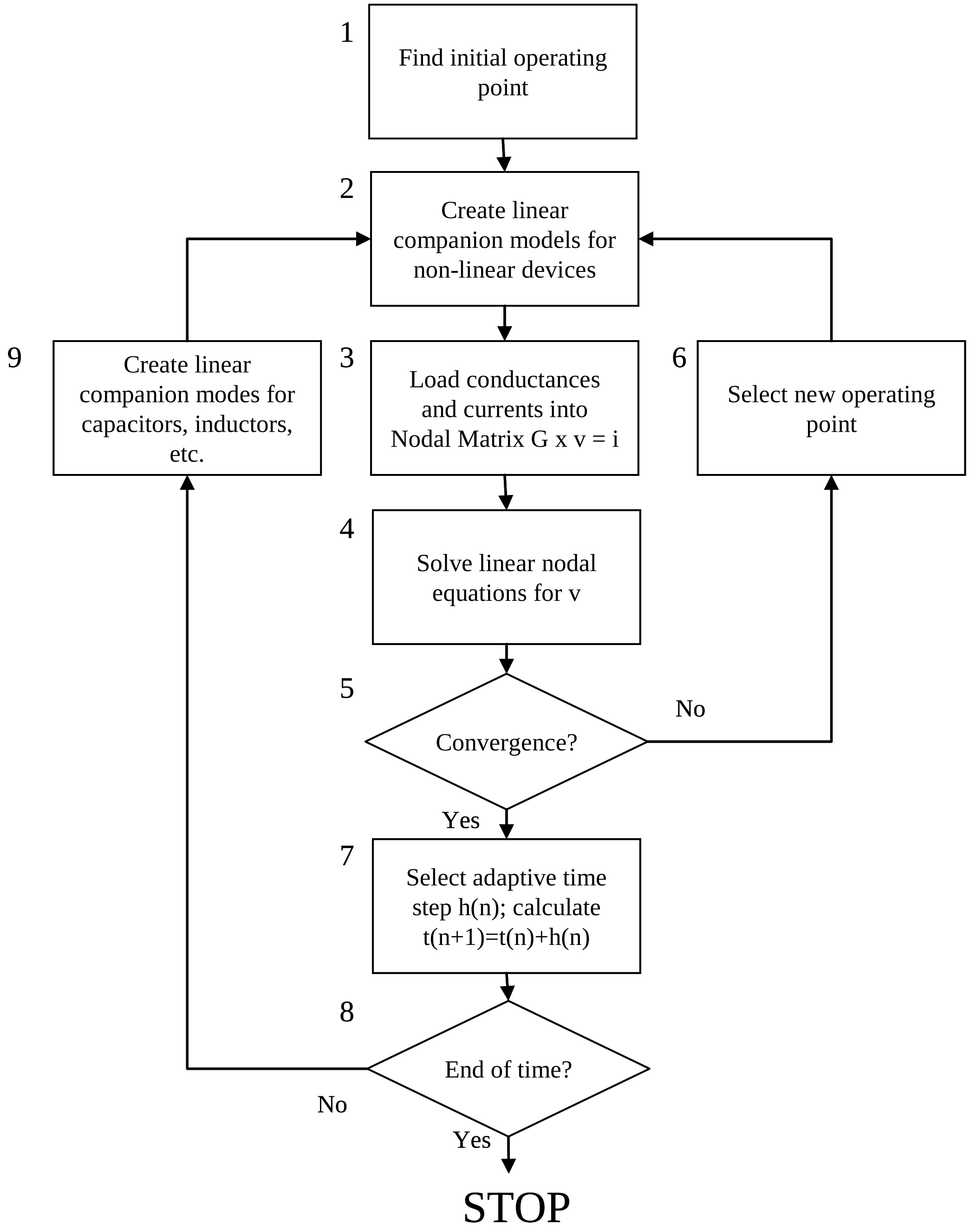


Figure 3-8: General Overview of SPICE Simulator Algorithm

3.6.2 Statistical Signal Analysis Simulation Methods

The use of statistical signal analysis (SSA) methods is a relatively recent development in signal integrity analysis [38]. These methods have been developed in response to the need to account for the ever-increasing number of variables when simulating high-speed SERDES (HSS) links. It is generally accepted that SSA-based methods must be employed at speeds greater than 5 Gb/s, but the methods are commonly employed at lower speeds in the range of 2 to 5 Gb/s [38]. SSA simulators are based on fast convolution and superposition methods and as such can simulate millions of bits, as opposed to SPICE simulators based on incremental time-stepping algorithms.

Figure 3-9 shows a diagram of a typical HSS system [38]. The system consists of three primary components, namely, the transmitter circuitry, the channel, and the receiver circuitry. For a typical HSS link, however, the transmitter circuitry is more than just the IO driver; it also generally includes feed-forward equalization (FFE) circuitry to compensate for anticipated channel losses. Figure 3-10 shows the structure of a typical HSS transmitter [38].

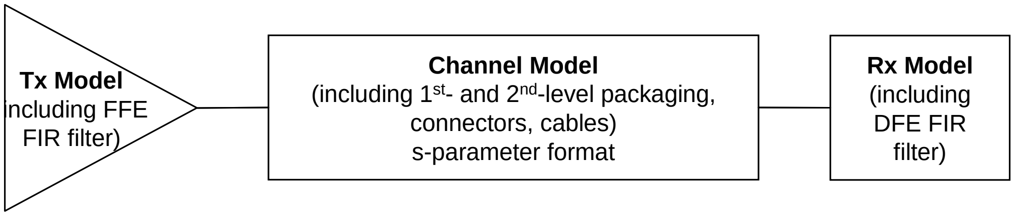


Figure 3-9: Typical Model for an HSS Link

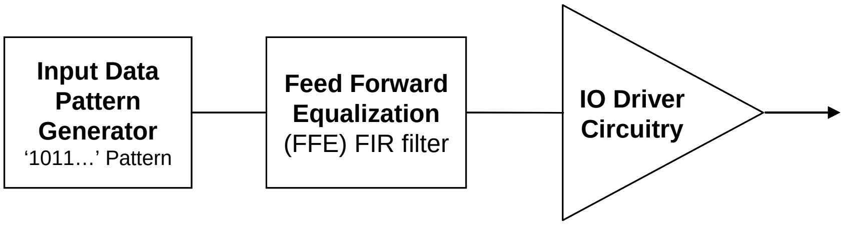


Figure 3-10: Transmitter Model Structure

Similarly, the receiver circuitry also generally contains decision-feedback equalization (DFE) circuitry, in addition to the receive threshold circuitry, to compensate for the fact that the eye at the receiver input might be totally closed. Figure 3-11 shows the structure of a typical HSS receiver [38].

The channel model shown in Figure 3-9 consists of the passive interconnect structure that usually consists of 1st- and 2nd-level packaging, the associated wiring and vias, and any connectors and cables. The channel model is generally developed separately, either by empirical extraction using vector network analyzer (VNA) measurement methods, or by using simulation in a full-wave electromagnetic field solver [63]. The latter case is more useful from a hardware design standpoint, in that it can be done prior to hardware fabrication, as part of the channel design process. The former case is more useful for model-to-hardware correlation and hardware design validation activities.

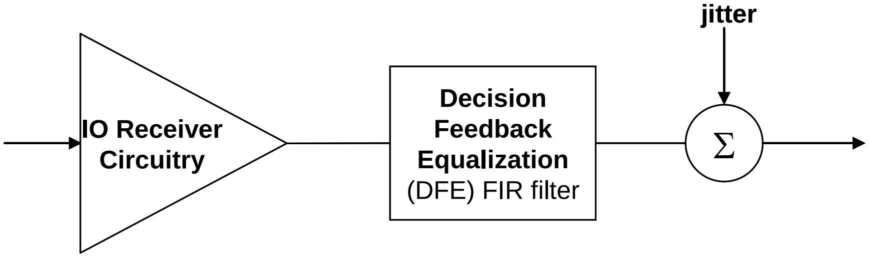


Figure 3-11: Receiver Model Structure

The channel model is generally expressed as an s-parameter matrix, usually using a standard scattering parameter format. Often, the channel is broken down into subcomponents representing the 1st-level package for the transmitter device, the 2nd-level packaging consisting of board-level interconnect and cables/connectors, and the 1st-level package for the receiver device. The s-parameters are measured for each of these subcomponents separately, and then analytically concatenated to form the overall s-parameter matrix for the channel. For the transmitter, this can be expressed as [38]:

$$\underline{T_x}(j\Omega) = \begin{bmatrix} 1 & T_{x21}(j\Omega) \\ 1 & T_{x22}(j\Omega) \end{bmatrix}, \tag{3.69}$$

where $\underline{T_x}(j\Omega)$ is the s-parameter matrix for the transmitter. Here $T_{x21}(j\Omega)$ represents the low-pass filter effect of rolling off the square-wave digital input to the driver stage, and $T_{x22}(j\Omega)$is a representation of the transmitter 1st-level package's return loss. Likewise for the receiver, and s-parameter matrix can be formed as [38]:

$$\underline{R_x}(j\Omega) = \begin{bmatrix} R_{x11}(j\Omega) & 1 \\ 1 & 1 \end{bmatrix}, \tag{3.70}$$

where $\underline{R_x}(j\Omega)$ is the s-parameter matrix for the receiver. Here $R_{x11}(j\Omega)$ represents the receiver 1st-level package's return loss. Finally, for the 2nd-level package, the s-parameter matrix can be expressed as [38]:

$$\underline{S_{cb}}(j\Omega) = \begin{bmatrix} \underline{S_{11}}(j\Omega) & \underline{S_{12}}(j\Omega) \\ \underline{S_{21}}(j\Omega) & \underline{S_{22}}(j\Omega) \end{bmatrix}, \tag{3.71}$$

where $\underline{S_{11}}(j\Omega)$, $\underline{S_{12}}(j\Omega)$, $\underline{S_{21}}(j\Omega)$, and $\underline{S_{22}}(j\Omega)$ are measured 4-port differential data compiled for the channel. To determine an overall transfer function for the channel in the digital frequency domain, the s-parameter matrices given above must each be converted to chain-scattering matrix (t-matrix) parameters and multiplied using matrix algebra to obtain an overall t-matrix representation [57], which can then be converted back to s-parameters. Finally, a Ω-domain transfer function $T_r(j\Omega)$ can be obtained from the channel's overall s-parameter matrix [58].

Once the channel's overall transfer function is obtained, it is used to calculate the response of the channel to a stimulus as defined in Figure 3-12, using fast convolution methods. This is called the channel pulse response and is calculated as follows [38]:

$$t_x(t) = H(t)H(t_{period} - t)\,, \tag{3.72}$$

$$r_x(j\Omega) = t_x(j\Omega)T_r(j\Omega)\,, \tag{3.73}$$

and

$$r_x(t) = IFFT(r_x(j\Omega))\,, \tag{3.74}$$

where $H(t)$ represents a unit-step function at $t = 0$, $t_x(t)$ is the resulting input signal pulse of Figure 3-12, $t_x(j\Omega)$ is the input signal pulse in the ω-domain, and $r_x(t)$ is the channel's pulse response. The general principle of SSA simulation which enables the tremendous increase in simulation speed, and thus allows simulation of millions of bits, is the use of superposition principles to construct the overall channel response using combinations of the above pulse response $r_x(t)$ [38].

The FFE and DFE circuitry for the transmitter and receiver, respectively, must be included in the overall simulation of the link [38]. These FFE and DFE circuits are considered to be digital filters, and are generally implemented as a finite impulse response (FIR) filters. As such, the FFE and DFE filters have filter coefficients which must be set appropriately to compensate for the channel losses. Most SSA simulation programs have built-in algorithms to optimally set these coefficients [38]. The DFE coefficients are generally determined by examining various samples of the pulse response waveform, while the FFE coefficients are generally determined by examining the channel transfer function, finding its inverse, and using an optimization algorithm to find a best-fit to the FFE filter response that matches this inverse [64].

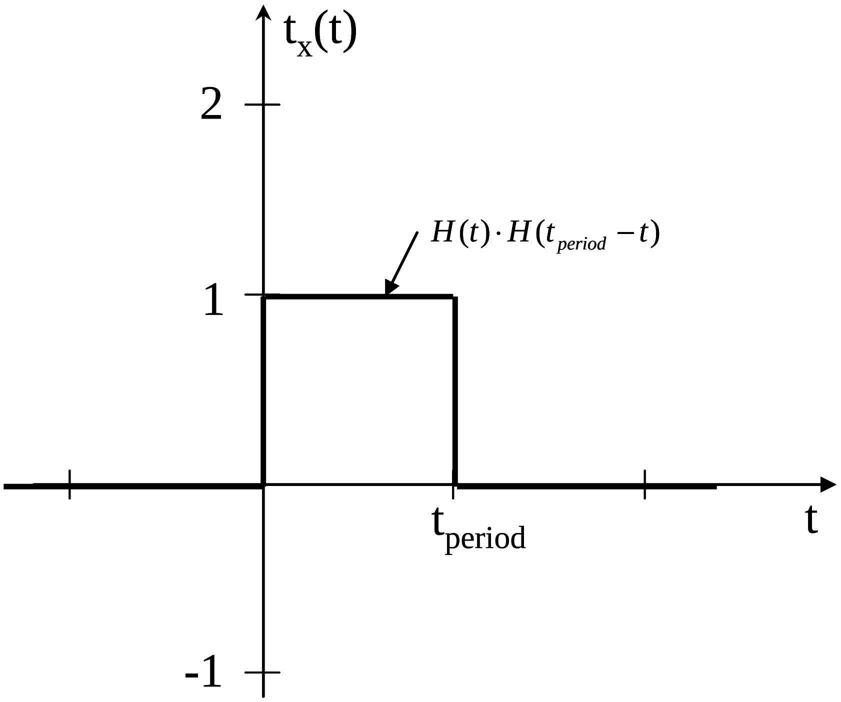


Figure 3-12: Pulse Stimulus for Calculation of Channel Pulse Response

In addition to incorporating the driver, receiver, and channel models, SSA simulators also directly handle the various deterministic and random jitter terms associated with the transceiver circuitry. Deterministic jitter includes data-dependent jitter, duty-cycle distortion, sinusoidal jitter, periodic jitter, and bounded uncorrelated jitter. Random jitter includes uncorrelated unbounded Gaussian jitter, and correlated bounded Gaussian jitter [38], thereby enabling a comprehensive simulation including all physical components of the link, and incorporating the various jitter components.

There are many different SSA simulation engines. Companies that design libraries of HSS transmitter and receiver macros often develop their own proprietary SSA engines so that they can easily provide the HSS transmitter and receiver models and define the associated FFE and DFE structures and associated jitter terms. One example of a proprietary engineer is IBM’s HSSCDR tool [32], [38]. An example of an open-source SSA engine is the StateEye tool [64]. The StatEye tool tends to use industry standard bus specifications to define its transmitter and receiver models and associated equalization and jitter terms. Various computer automated design/engineering (CAD/CAE) software vendors also produce SSA simulation codes for purchase [33]–[35], [37].

3.7 System Identification

System identification involves forming models for systems from observed data [65]. Generally, measured data is available for the system's input and output, from which the model can be created. Both time- and frequency-domain system identification methods are available, and the data can be in either form. Figure 3-13 and Figure 3-14 show the general system identification problem in the time and frequency domains, respectively.

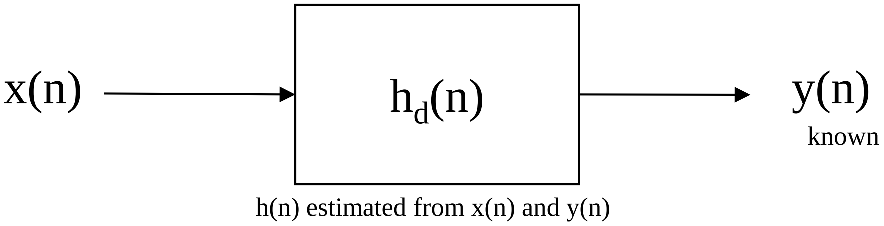


Figure 3-13: System Identification Problem in the Discrete Time Domain

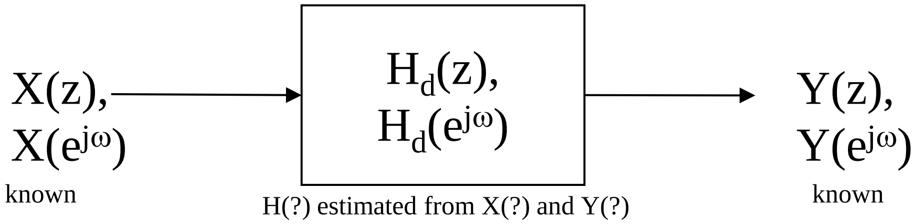


Figure 3-14: System Identification Problem in the Discrete Frequency Domains

For systems about which little is known about the system, or for which the model is inherently very complex, the system can be considered to be a "black box." For black-box systems, the system identification process usually involves non-parametric methods. Examples of such methods in the time domain include impulse-response analysis and step-response analysis, which are common methods in DSP and control theory. In the frequency domain, non-parametric methods include sine-wave testing, correlation methods, and empirical transfer-function estimation [65].

In contrast to black box methods, often there is some knowledge of the system's nature and thus it is possible to make assumptions about the general form of the desired model. The "grey-box" concept leads to the use of parametric system identification methods, for which a general, finite parameter vector can be applied in an effort to bound the identification problem [65]. This generally involves finding an estimate for the parameter vector which minimizes the prediction error subject to some constraint. For system identification problems, it is common to use a prediction-error method (PEM) based on either least-squares estimator (LSE) or maximum-likelihood estimator (MLE) [65]. Because of the deterministic, non-stochastic nature of analog waveforms propagating on computer interconnects, the least-squares error estimator (LSEE) seems to be a reasonable choice for the prediction-error method used for signal integrity analysis applications.

For time-domain parametric system-identification, Figure 3-15 shows a block diagram of a general output-error system, which leads to a general family of model structures [65]:

$$A(z)Y(z) = \frac{B(z)}{F(z)}U(z) + \frac{C(z)}{D(z)}E(z)\,. \tag{3.75}$$

In Figure 3-15, $u(n)$ is the input sequence, $y(n)$ is the output sequence, and $e(n)$ is some error that enters into the system, and correspond to $U(z)$, $Y(z)$and $E(z)$, respectively, in

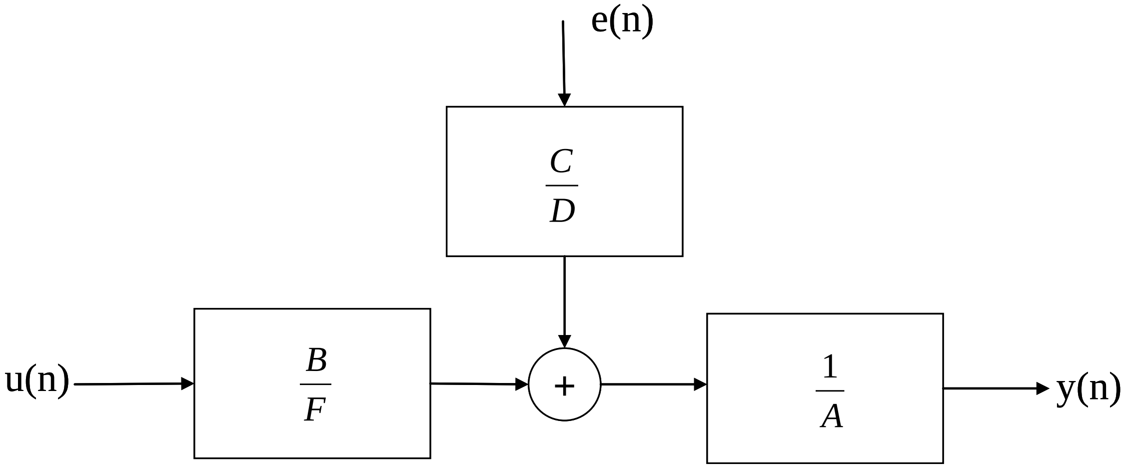


Figure 3-15: Output-Error System Defining a General Family of Model Structures

the equation above. From this general family, specific model structures can be defined which are commonly used in system identification. These include the finite impulse response (FIR), auto-regression-exogeneous (ARX), auto-regressive moving average exogeneous (ARMAX), and auto-regressive moving average (ARMA) models, among others [65]. These system identification routines, like most estimation schemes, have batch-, sequential-, and recursive-mode forms [66].

It is interesting to note a parallel between system identification and the s-parameter concepts discussed earlier. Determination of the s-parameter matrix involves subjecting an unknown system to an input data sequence (in this case an incident wavefront) and observing an output sequence (an emerging wavefront). As such, the s-parameters represent a type of system identification technique, even though s-parameters are usually considered to be an RF engineering characterization technique, and not a system identification technique. Another parallel is that between system identification methods and the pulse-response concept used in SSA simulation. The determination of the channel pulse response to a known input is also essentially a system identification method.

3.7.1 Linear Least Squares Parameter Estimation

The batch-mode and recursive-mode system identification routines manifested in MATLAB, such as the arx.m and rarx.m functions, use least-squares error (LSE) algorithms as their optimization criterion [67]. There are many forms of least-squares error estimation, such as linear least squares (LLSE), weighted least squares (WLSE), with batch, sequential, and recursive methods all available [66].

The LLSE assumes a model for the signal vector s in terms of the parameter vector $\underline{\theta}$, namely [66],

$$\underline{s} = \underline{H} \cdot \underline{\theta}\,. \tag{3.76}$$

Here $\underline{H}$ is an observation matrix formed by using the input and output data sequences for the system, using the system's difference equation description in the discrete time domain. The LLSE is then given by [66]:

$$\underline{\hat{\theta}} = (\underline{H}^T \cdot \underline{H})^{-1} \cdot \ \underline{H}^T \cdot \underline{x}\,, \tag{3.77}$$

where the minimum LSE error criterion is given by:

$$J(\underline{\theta}) = \underline{x}^T \underline{x} - 2\underline{x}^T \underline{H}\, \underline{\theta} + \underline{\theta}^T \underline{H}^T \underline{H}\, \underline{\theta}\,. \tag{3.78}$$

The most difficult part of forming the LSE estimator is the creation of the observation matrix $\underline{H}$. It is formed from past values of the signal vector $\underline{s}$, and it must be carefully constructed.

3.8 Macromodeling Techniques Based Upon Vector Fitting

One important recent development in the modeling of interconnects for signal integrity applications is the concept of developing macromodels in the complex frequency domain to represent distributed-element circuits [44], [45], [46], [47]. These methods use vector fitting of frequency-domain or time-domain input data obtained from a variety of system identification methods [60]. The system identification data is most often given in scattering-parameter or analog frequency response format, to construct a rational function approximation to the transfer function $H_a(s)$ or $H_d(z)$. The formulation of such models offers significant computational advantages over a direct time-domain convolution approach to calculating the transient response [68]. In addition, transfer functions in the complex frequency s- or z-domains can be examined in the complex plane to gain intuition about the nature of the system that is lost in the analog or discrete frequency domains.

For the s-domain, the resulting transfer function for the macromodel has the general rational function form [44]:

$$H_a(s) = \frac{B_a(s)}{A_a(s)}\,. \tag{3.79}$$

The roots of $B_a(s)$ in the equation above are the zeros of the transfer function $H_a(s)$, while the roots of $A_a(s)$ are the poles of $H_a(s)$. When forming the macromodel for the system, the number of desired poles and zeros can be specified to optimize the accuracy of the model with respect to its complexity. The pole-zero map can then be examined in the complex s-plane to examine the characteristics of the model. The function $H_a(s)$ can be approximated as a partial fraction expansion of general form [44]:

$$H_a(s) \approx \sum_{n=1}^{N} \frac{c_n}{s - a_n} + d + sh\,, \tag{3.80}$$

where c_n and a_n are conjugate pairs and d and h are real. The vector fitting algorithm and associated process finds a least-squares approximation of $H_a(s)$ by finding estimates to the parameters c_n, a_n, d, and h. The process is implemented in two stages, pole identification and residue identification. The poles are identified by creating a rational function approximation for an unknown function $\sigma(s)$ and $\sigma(s)H_a(s)$, namely [44]:

$$\begin{bmatrix} \sigma(s)H_a(s) \\ \sigma(s) \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{n=1}^{N} \frac{c_n}{s - \bar{a}_n} + d + sh \\ \displaystyle\sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \bar{a}_n} + 1 \end{bmatrix}, \tag{3.81}$$

where the poles $\bar{a}_n$ are specified by examining the frequency range, picking a set of multiple pairs of complex conjugate poles that span the range, and creating a rational function approximation for both $\sigma(s)$ and $\sigma(s)H_a(s)$. The equation

$$\left[\sum_{n=1}^{N} \frac{c_n}{s - \bar{a}_n} + d + sh\right] \approx \left[\sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \bar{a}_n} + 1\right] \cdot H_a(s)\,, \tag{3.82}$$

is linear in c_n, d, h, and $\tilde{c}_n$ and can be solved as a least-squares problem [44], [66], [65]. The zeros of $\sigma(s)$ are then found and used as the poles of the estimate for $H_a(s)$. The vector fitting process is implemented in a variety of commercial software packages and is readily available to signal integrity engineers [44], [69].

Linear macromodeling can be applied directly in the z-domain as well [46], [47]. For the z-domain, the resulting transfer function for the macromodel has the general rational function form:

$$H_d(z) = \frac{B_d(z)}{A_d(z)}\,. \tag{3.83}$$

The roots of $B_d(z)$ in the equation above are the zeros of the transfer function $H_d(z)$, while the roots of $A_d(z)$ are the poles of $H_d(z)$. When forming the macromodel for the system, the number of desired poles and zeros can be specified to optimize the accuracy of the model with respect to its complexity. The pole-zero map can then be examined in the complex z-plane to examine the characteristics of the model. The function $H_d(z)$ can be approximated as a partial fraction expansion of general form [46]:

$$H_d(z) \approx \sum_{n=1}^{N} \frac{c_n}{s - z^{-1}a_n}, \tag{3.84}$$

where c_n and a_n are, in general, complex conjugate pairs. The vector fitting algorithm and associated process finds a least-squares approximation of $H_d(z)$ by finding estimates to the complex parameters c_n and a_n, following a similar thought process to the s-domain vector fitting outlined above.

3.9 Mathematical Characteristics of Interconnect Models

Stability, passivity, and causality are three important properties of the models utilized in signal integrity analysis. These characteristics are often assumed blindly by signal integrity engineers, but in fact the stability, passivity, and causality conditions can be violated, either in the measurement or simulation of the raw frequency-domain data used to extract the model, or in the model extraction process itself [63]. Even worse, these violations are not evident by looking at the transient response waveforms, but must rather be examined in the complex frequency domain. Further, since electrical interconnects and passive lumped-element components, two of the most commonly modeled elements in signal integrity analysis, are inherently passive, stable and causal, it is unreasonable to accept models for these elements which don’t share these properties. Formal mathematical definitions exist for each of these properties, in both the time- and frequency-domains, for both continuous-time and discrete-time systems. These definitions will be reviewed here.

3.9.1 Continuous-Time Systems

The system is assumed to be a continuous-time, linear time invariant (LTI) system. A linear system is defined to be that for which the system response to a linear combination of inputs [63]

$$v_i(t) = c_1 v_{i1}(t) + c_2 v_{i2}(t)\,, \tag{3.85}$$

is

$$v_o(t) = c_1 v_{o1}(t) + c_2 v_{o2}(t)\,, \tag{3.86}$$

where $v_{o1}(t)$ and $v_{o2}(t)$ are the system's output responses to the input stimuli $v_{i1}(t)$ and $v_{i2}(t)$, respectively. A time-invariant system is defined to be that for which the behavior does not change with time. Formally, if $v_o(t)$ is the output response due to input $v_i(t)$, then $v_o(t-\tau)$ is the response to $v_i(t-\tau)$. For an LTI system, the system input-output relationship can be expressed as a time-domain convolution [63]

$$v_o(t) = h_c(t) * v_i(t) = \int_{-\infty}^{\infty} h(t-\tau) v_i(\tau) d\tau\,, \tag{3.87}$$

where $h_c(t)$ is the system's continuous-time impulse response.

3.9.1.1 Causality for Continuous-Time Systems

The fundamental principle of causality is that an effect cannot occur before its cause [63]. Mathematically, this can be defined in the continuous time domain by stating that a system is causal if and only if, for all input pairs $v_{i1}(t)$ and $v_{i2}(t)$ such that

$$v_{i1}(t) = v_{i2}(t)\,, \quad t \le t_o \qquad \forall t_{o\,,} \tag{3.88}$$

then the corresponding outputs satisfy

$$v_{o1}(t) = v_{o2}(t)\,, \quad t \le t_o\,. \tag{3.89}$$

With this definition, an LTI system is causal if and only if all the elements of its impulse response $h_c(t)$ vanish for $t < 0$, i.e.,

$$h_c(t) = 0, \quad t < 0\,. \tag{3.90}$$

For distributed-element systems, the causality condition is extended to account for finite propagation delay [70]. For distributed systems, the causality condition becomes:

$$h_c(t) = 0, \quad t < T_{ij},\ T_{ij} \ge 0 \quad \forall i,j\,. \tag{3.91}$$

In the complex frequency s-domain, it is useful to use Laplace transform theory and the s-plane to assess causality. Given the focus on causality aspects of the system, it is necessary to note that we must use the bilateral Laplace transform,

$$V(s) = \int_{-\infty}^{\infty} v(t)e^{-st}dt\,, \tag{3.92}$$

in lieu of the more common unilateral Laplace transform. The unilateral transform assumes the system is causal by its very construction [43]. Using the bilateral Laplace transform, and its associated region of convergence (ROC), we can say a system with impulse response $h_c(t)$ is causal if and only if its bilateral Laplace transform is 1) defined and analytic in the half-plane $Re\{s\} > \sigma_o$, and 2) grows not faster than a polynomial for $Re\{s\} > \sigma_o$ [63].

3.9.1.2 Stability for Continuous-Time Systems

The fundamental concept of stability for an LTI system is that the system output response is bounded and does not grow out of control, assuming that the input stimuli to the system are bounded [63]. Thus stability of an LTI system, either lumped-element or distributed-element, is guaranteed if and only if $h_c(t)$ is such that

$$\int_{-\infty}^{\infty} |h_c(t)|dt < +\infty\,. \tag{3.93}$$

In the s-domain, the region of convergence is also important to the concept of stability. The stability condition expressed above in the time domain only holds if, in the s-domain, the ROC of the transfer function $H_a(s)$ corresponding to $h_c(t)$: 1) contains the imaginary $j\Omega$ axis, and 2) the transfer function itself $H_a(s = \infty)$ is also bounded [63].

3.9.1.3 Passivity for Continuous-Time Systems

A system is considered passive if it is unable to produce energy, hence the notion of passive lumped RLC elements and passive distributed interconnect elements like transmission lines. Clearly an interconnect system is a passive system. The mathematical definition of passivity depends on what type of n-port parameters are chosen to represent the network, i.e,, impedance parameters, admittance parameters, or scattering parameters, etc. For the s-parameter case, the definition of passivity in the time domain becomes

$$\int_{-\infty}^{t} \left[\underline{a}^T(\tau)\underline{a}(\tau) - \underline{b}^T(\tau)\underline{b}(\tau)\right] d\tau \geq 0\,,. \tag{3.94}$$

where $\underline{\text{a}}$ and $\underline{\text{b}}$ are the incident and emerging waves of the n-port network, respectively, as depicted in Figure 3-7. It turns out that if an LTI system is passive, it is also causal [63], [42]. Since all passive systems are causal, any non-causal system cannot be passive. Also, any macromodeling process which enforces passivity will also enforce causality.

For the s-domain, and assuming a scattering matrix representation of the system, the passivity condition can be expressed by stating that a scattering matrix $\underline{S}(s)$ represents a passive linear system if and only if 1) each element of $\underline{S}(s)$ is analytic in $Re\{s\} > 0$; 2) $\underline{I} - \underline{S}^H(s)\underline{S}(s)$ is a non-negative-definite matrix for all s such that $Re\{s\} > 0$; and 3) $\underline{S}(s^*) = \underline{S}^*(s)$. Here $\underline{I}$ is the identity matrix, $\underline{S}^*$ represents the complex conjugate of $\underline{S}$, and $\underline{S}^H$ is the transpose conjugate of $\underline{S}$ [63].

3.9.2 Discrete-Time Systems

For discrete-time, linear time invariant (LTI) system, a linear system is defined to be that for which the system response to a linear combination of inputs [63]:

$$v_i(n) = c_1 v_{i1}(n) + c_2 v_{i2}(n)\,, \tag{3.95}$$

is

$$v_o(n) = c_1 v_{o1}(n) + c_2 v_{o2}(n)\,, \tag{3.96}$$

where $v_{o1}(n)$ and $v_{o2}(n)$ are the system's output responses to the input stimuli $v_{i1}(n)$ and $v_{i2}(n)$, respectively. A time-invariant system is defined to be that for which the behavior does not change with time. Formally, if $v_o(n)$ is the output response due to input $v_i(n)$, then $v_o(n - n_o)$ is the response to $v_i(n - n_o)$. For a discrete-time LTI system, the system input-output relationship can be expressed as a time-domain convolution [63]

$$v_o(n) = h(n) * v_i(n) = \sum_{k=-\infty}^{\infty} h_d(n-k)v_i(k)\,, \tag{3.97}$$

where $h_d(n)$ is the system's impulse response.

3.9.2.1 Causality for Discrete-Time Systems

Just as in the continuous-time domain, the fundamental principle of causality in the discrete-time domain is that an effect cannot occur before its cause [63]. Mathematically, this can be defined in the discrete time domain by stating that a system is causal if and only if, for all input pairs $v_{i1}(n)$ and $v_{i2}(n)$ such that

$$v_{i1}(n) = v_{i2}(n)\,, \qquad n \le n_o \qquad \forall n_{o\,,} \tag{3.98}$$

then the corresponding outputs satisfy

$$v_{o1}(n) = v_{o2}(n)\,, \qquad n \le n_o\,. \tag{3.99}$$

With this definition, a discrete-time LTI system is causal if and only if all the elements of its impulse response $h_d(n)$ vanish for $n < 0$, i.e.,

$$h_d(n) = 0, \quad \forall\, n < 0\,. \tag{3.100}$$

In the discrete complex frequency z-domain, it is possible to use z-transform theory and the z-plane to assess causality. In the z-plane, for a general sequence the region of convergence (ROC), defined in (3.25), generally has the shape of an annulus, that is,

$$r_1 < |z| < r_2\,. \tag{3.101}$$

For a causal system, however, the ROC can be shown to be the exterior of a circle whose radius intersects the outermost pole in the z-plane [42]. For such a function, the ROC of $H_d(z)$ will include $|z| = \infty$ [41]. Thus, a discrete-time system with impulse response function $h_d(n)$ is causal if and only if its ROC is the exterior of a circle of radius $r_1 < \infty$ and which includes $z = \infty$, i.e.,

$$ROC = \{r_1 < |z| \le \infty\}, \tag{3.102}$$

where r_1 is the radius of a circle intersecting the outermost pole of the system.

Consistent with the above discussion, an observation can be made regarding the pole-zero map in the z-plane. A given pole-zero map for $H_d(z)$ represents a causal system if and only if there are as many finite poles as finite zeros., and the ROC for $H_d(z)$ is the exterior is as expressed in (3.102).

3.9.2.2 Causality Condition for Discrete-Time Systems in the ω-Domain

While the condition expressed as (3.102) is a valid test for causality, it is difficult to assess because it must be tested for all values of z. Fortunately, for systems which are real, causal, and stable, causality can also be checked in the discrete frequency ω-domain using the discrete Hilbert transform relationships [42], [71]:

$$H_{dI}\left(e^{j\omega}\right) = -\frac{1}{2\pi}\mathcal{P}\int_{-\pi}^{\pi} H_{dR}\left(e^{j\omega}\right)\cot\frac{\omega-\theta}{2}\,d\theta\,, \tag{3.103}$$

and

$$H_{dR}\left(e^{j\omega}\right) = x(0) + \frac{1}{2\pi}\mathcal{P}\int_{-\pi}^{\pi} H_{dI}\left(e^{j\omega}\right)\cot\frac{\omega-\theta}{2}\,d\theta\,, \tag{3.104}$$

Where $H_d\left(e^{j\omega}\right) = H_{dR}\left(e^{j\omega}\right) + jH_{dI}\left(e^{j\omega}\right)$ and $\mathcal{P}\{\dots\}$ indicates that the integral must be evaluated using Cauchy principal values.

3.9.2.3 Stability for Discrete-Time Systems

The fundamental concept of stability for a discrete-time LTI system is that the system output response is bounded and does not grow out of control, assuming that the input stimuli to the system are bounded [41], [42]. Thus stability of a discrete-time LTI system is guaranteed if and only if $h_d(n)$ is such that

$$\sum_{n=-\infty}^{\infty} |h_d(n)| < +\infty\,. \tag{3.105}$$

In the z-domain, the region of convergence is also important to the concept of stability. The stability condition expressed above in the discrete time domain only holds if, in the z-domain, the ROC of the transfer function $H_d(z)$ corresponding to $h_d(n)$ contains the unit circle.

For a discrete-time causal system, the condition for stability in the z-plane can be further narrowed. Since, for a causal system, the ROC is as prescribed in (3.102), and the poles cannot reside in the ROC, a causal LTI system is stable if and only if all the poles of its system are inside the unit circle [41], [42].

Chapter 4: Identification of the System Model

4.1 Introduction

The first step in performing signal integrity analysis is to develop a comprehensive simulation model for use in predicting waveform degradation in the interconnection channel, commonly referred to as the interconnect model. The analysis is usually conducted in the time domain, using a SPICE simulator [72] or, alternatively, a statistical simulation analysis (SSA) simulator such as IBM's HSSCDR [32]. Historically, the interconnect portion of the models have been created in SPICE language utilizing lumped elements such as inductors, capacitors, and resistors, along with distributed elements such as transmission lines to handle the signal propagation delay [29]. More recently, as frequencies have increased, it has become common to represent the interconnect models using scattering parameters [73]. This leads to some advantages associated with model extraction, and to some disadvantages associated with the ability to parameterize some aspects of the model, such as the transmission lines' lengths.

The overall objective of this research project is to examine the interconnect model's characteristics using techniques borrowed from other disciplines in electrical engineering, prior to invoking the simulator in the time domain. The motivation is to obtain additional insight into the nature of the interconnect system that can be leveraged into the subsequent time-domain analysis, thereby minimizing the computationally-intensive simulation cases that must be run, and saving effort, iterations, and time in the product design cycle.

As a first step toward this goal, an initial model must be obtained for the system interconnect. This process is essentially a system identification (SID) problem, and can be approached using a variety of methods [65]. In this chapter, the identification of the model will be done in various frequency domains, both discrete and analog, using the models' frequency transfer functions $H_a(j\Omega)$, $H_d(e^{j\omega})$, and $H_d(k)$. Examination of the resulting analog transfer function $H_a(s)$ and the discrete transfer function $H_d(z)$ that are ultimately developed from these frequency transfer functions will be done in subsequent chapters. The advantages and disadvantages of each of the system identification methods will be compared and contrasted. At the end of the chapter, a single system identification method will be selected for use in the remainder of the project.

4.2 SPICE Circuit Representation for a 4th-Order Test Circuit with Delay

The first step in analyzing the system characteristics is to obtain the channel models used to describe the system interconnects, using a system identification (SID) technique [63]. The model identification will be performed using several known system identification methods from the fields of radio-frequency (RF) and digital signal processing (DSP) engineering. The advantages and disadvantages of the various methods will be compared and contrasted, and a single system identification method will be selected as the one to be utilized from this point forward.

Figure 4-1 shows the test circuit that will be used to compare the various system identification methods. The circuit consists of lumped circuit elements to represent packaging parasitics, and a lossy transmission line distributed circuit element to represent what would most commonly be stripline or microstrip transmission line structures in a printed circuit board, package substrate, or other form of electronic packaging for an integrated circuit chip [1].

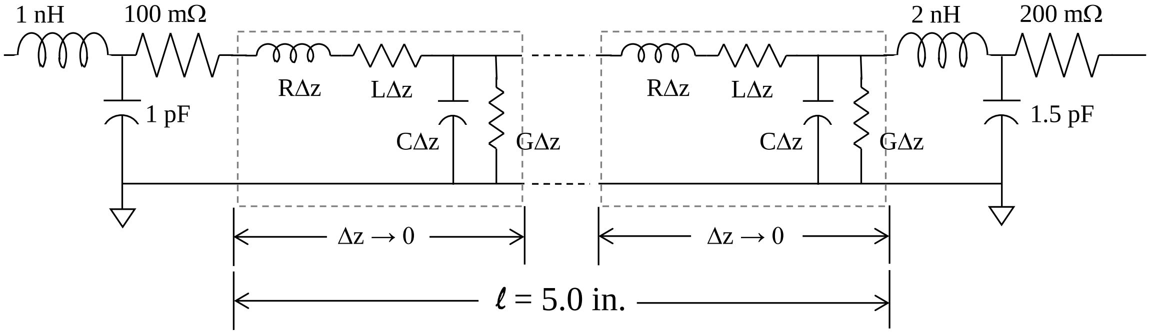


Figure 4-1: Test Circuit with Lumped Elements and Transmission Line with Precise Distributed Element Representation

Note that the ladder network representation of the transmission line in Figure 4-1 is comprised of an infinite number of sections that are infinitesimally small, with $\Delta z \rightarrow 0$, which leads to a description of the transmission line with partial differential equations, in accordance with distributed circuit theory of transmission lines [6]. As such, it must be emphasized that the transmission line's $RLGC$ parameters are distributed elements, not lumped elements, as indicated by the fact they are expressed in per unit length quantities which must be multiplied by Δz to have normal resistive, capacitive, and inductive units.

The depiction of the test circuit of Figure 4-1, while mathematically precise, is not a convenient circuit schematic representation for repetitive use. Figure 4-2 shows a more convenient and typical representation of the test circuit, with the ladder network representation of the transmission line element of Figure 4-1 replaced with the cylindrical symbol representing the transmission line [1], [7]. This symbol will be used for distributed transmission line elements throughout the remainder of this project.

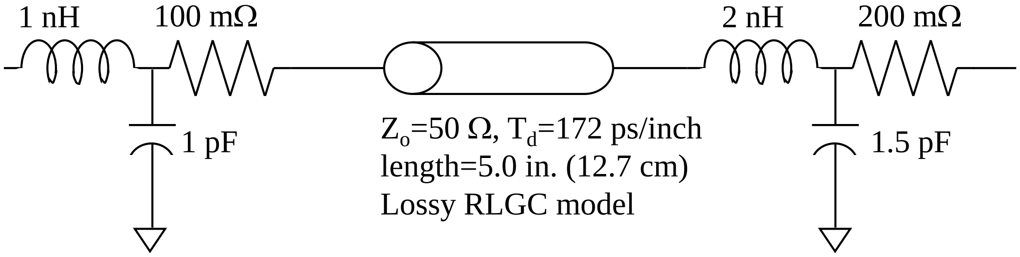


Figure 4-2: Test Circuit with Lumped Elements and Transmission Line with Convenient Electrical Schematic Representation

It should be emphasized that the test circuit as defined in Figure 4-1 and Figure 4-2 is intentionally arbitrary, and is thus somewhat oversimplified as compared to a real channel model encompassing the entire signal path in a typical electronic system. Such a model often includes the overall path from a controlled-collapse chip connect (C4) on the sending module, through 1st- and 2nd-level packaging, and finally terminating at the C4 on the receiving module.[9] The values of the lumped elements were chosen as a reasonable representation of parasitic effects of the electronic packaging, such as connectors, solder connections, and terminal leads. The presence of the transmission line is important for studying the effects of attenuation, dispersion, and propagation delay in the system, as these effects are frequency-dependent and thus non-intuitive in nature [16], [17], [74].

The simplified model shown includes four reactive lumped elements. Without the transmission line element, it is easily seen that the system's transfer function in the

---

[9] The term "controlled collapse chip connect" and associated C4 acronym are most commonly used within IBM Corporation. The more common term in the industry at large is "flip chip."

complex frequency domain, $H_a(s)$, would consist of a 4th-order rational polynomial [40]. The addition of the transmission line element introduces attenuation, dispersion, and delay into the system [6], and significantly complicates the form and order of $H_a(s)$, leading to a rational function of higher order to account for the propagation delay [74]. Nevertheless, the test circuit of Figure 4-2 will be referred to as the "4th-order test circuit with delay" or, often, as simply the "4th-order test circuit."

For the test circuit of Figure 4-2, a typical signal integrity analysis would consist of a time-domain simulation performed in SPICE or a similar tool [2], [72]. The simulation would be performed by assuming the test circuit of Figure 4-2 as the channel interconnect model, and by applying a driver circuit model at the input and a receiver circuit model at the output. Figure 4-3 shows the overall circuit including the driver, interconnect, and receiver load circuit models, with circuit nodes labeled as shown.

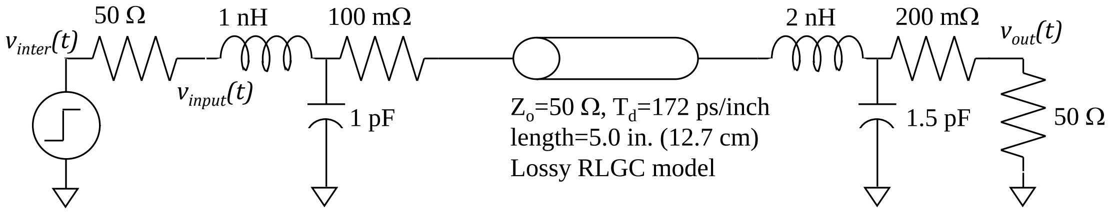


Figure 4-3: 4th-Order Test Circuit (with Delay) with Driver Circuit and Receiver Circuits Attached

The resulting time-domain output waveforms obtained from SPICE simulation of the circuit depicted in Figure 4-3 are displayed in Figure 4-4. Here the signal waveform $v_{inter}(t)$ represents the driver input waveform, which is a perfect square- and pulse-type signal. The node $v_{input}(t)$ represents the driver output waveform, which is distorted due to the loading of the interconnect plus its own load and the associated reflections. The node labeled $v_{out}(t)$ is the voltage at the output of the interconnect, at the input to the receiver load circuit. Note that the driver output waveform $v_{input}(t)$ can also be considered to be the input waveform to the interconnect circuit, and that it is significantly different than the ideal pulse input of $v_{inter}(t)$. In subsequent chapters, at times it will be convenient to excite the various interconnect models obtained using different methods with the $v_{input}(t)$ signal, so that the resulting output can be compared directly with that obtained originally from SPICE.

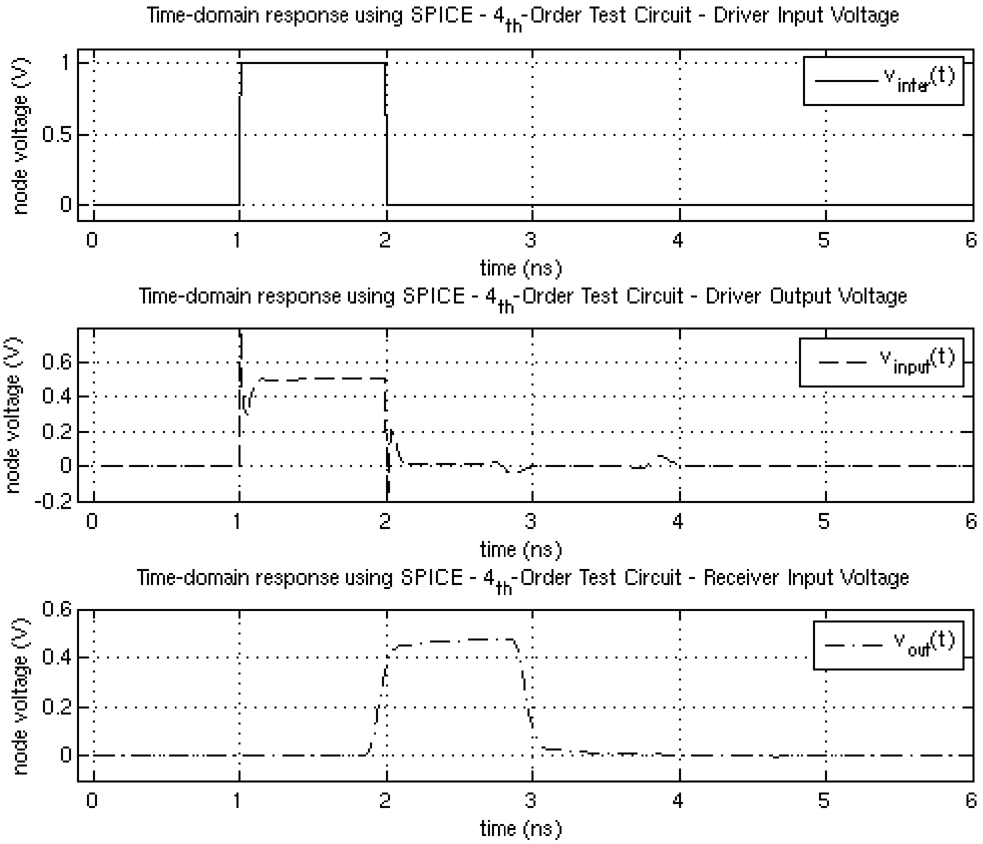


Figure 4-4: Time-Domain Output Waveforms for the 4th-Order Test Circuit (with Delay)

4.3 Creating a Frequency Transfer Function Model for the 4th-Order Test Circuit (with Delay) Using Various System Identification Methods

There are various ways to represent a system model for subsequent analysis. The most obvious method is to express the model in the native language of the SPICE simulator. As stated previously, these languages most commonly rely on lumped elements and distributed elements for their input. They also can generally understand some other input formats, such as scattering parameter matrices, other types of two-port network parameters, and various transfer function representations. For this reason, there are various options when selecting model format for a signal integrity simulation.

Since SPICE and SSA simulators natively understand circuit element and scattering parameter representations, these are the most commonly-used representations for signal integrity analysis. For this research project, however, the representation of choice will generally be in voltage transfer function format. There are three reasons for this. First, these transfer function formats are defined in the complex frequency domains, i.e., the s-domain for continuous time systems, and the z-domain for discrete-time systems. Since one of the stated objectives of this research project is the study of the interconnect models in these domains, it seems natural to work with the transfer function. Second, the entire fields of digital signal processing and control theory work largely in these domains. Since another stated objective of the research project is to borrow methods from those fields, it also makes sense for that reason. Third, vector fitting methods are available which make conversion of system identification data into the s- and z-domains very straightforward. For all of these reasons, the general strategy will be to obtain voltage transfer function models $H_a(s)$ and $H_d(z)$ in the complex frequency domains.

4.3.1 System Identification Using Scattering Parameter Identification Method

As stated previously, the test circuit of Figure 4-2 can be simulated directly in SPICE, using an assumed input waveform, to predict the time-domain output waveform for a given termination. The motivation here is different, however, and is to study the test circuit in the frequency domain by calculating an analytical form of the system transfer function. One popular method for model identification is that of scattering parameter characterization, in which the parameters are calculated using an AC sweep of sinusoidal voltage frequencies over the desired frequency range [19]. The scattering parameters for the test circuit can be obtained empirically for a fabricated system using vector network analyzer (VNA) measurements, but in a typical industrial design process they are usually obtained using electromagnetic field simulation prior to hardware fabrication [63].

4.3.1.1 Calculation of Scattering Parameters for the Test Circuit

For the case of the 4th-order test circuit of Figure 4-2, the scattering matrix was calculated for the test circuit using HSPICE [29] in AC analysis mode. Note that, although scattering parameter characterization can be considered to be a frequency-domain system identification technique, it is not commonly described in this way in RF and microwave literature. Likewise, the system identification literature often does not recognize scattering parameter characterization as a system identification technique, either [65]. Figure 4-5 shows the relationship of the scattering matrix to the test circuit. Note the scattering-parameter port definitions as they are defined in the figure.

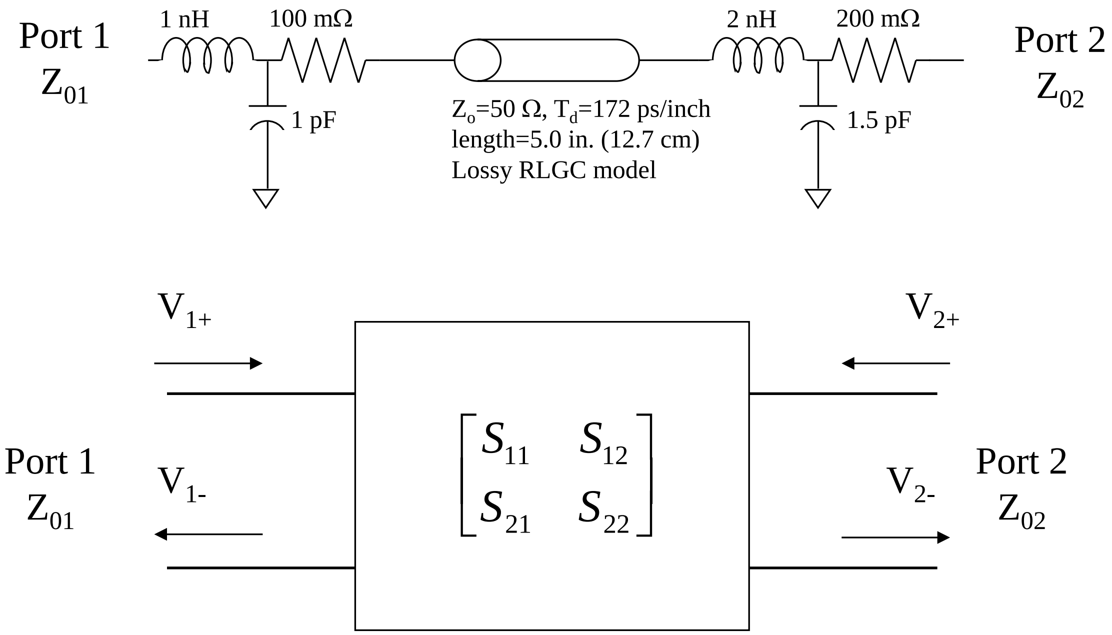


Figure 4-5: Application of Scattering Parameter Concepts to the Test Circuit

Figure 4-6 through Figure 4-9 show Bode plots of test circuit's scattering parameters' magnitude and phase, as calculated by HSPICE [29] over a frequency range of 0 to 10 GHz. Note the ripple effects exhibited in the magnitude responses in all four figures. These result primarily from the lumped circuit elements present on the ends of the 4th-order test circuit. Note the phase responses for the pass-through parameters S_{12} and S_{21} exhibit linear phase characteristics, which are characteristic of systems with propagation delay [19], [75], in this case due primarily to the presence of the transmission line element.

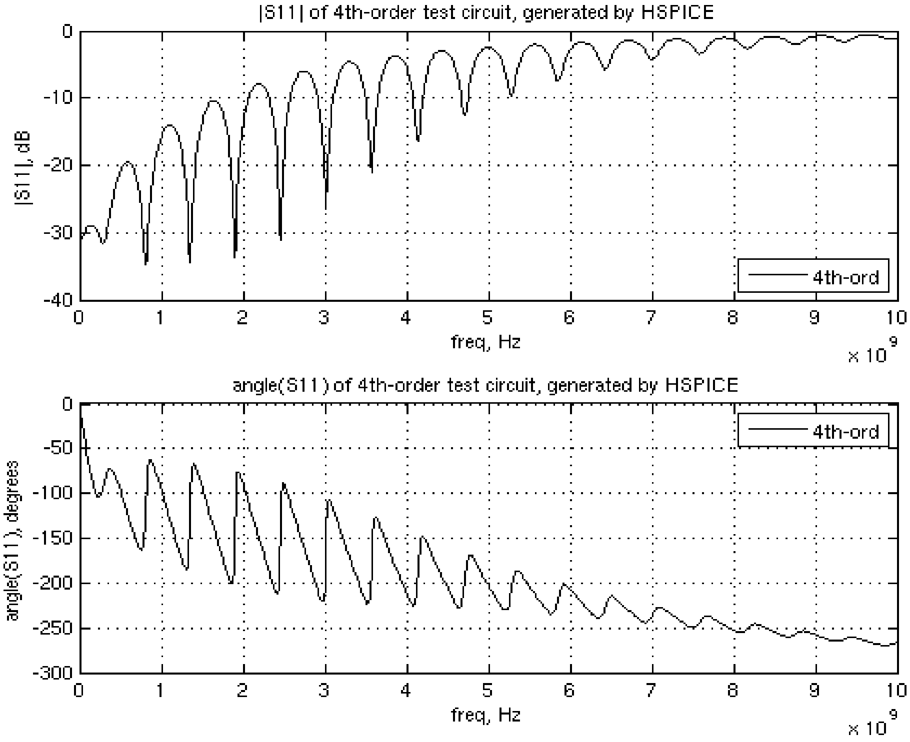


Figure 4-6: ***S_{11}*** **of the 4th-Order Test Circuit with Delay, as Calculated in HSPICE**

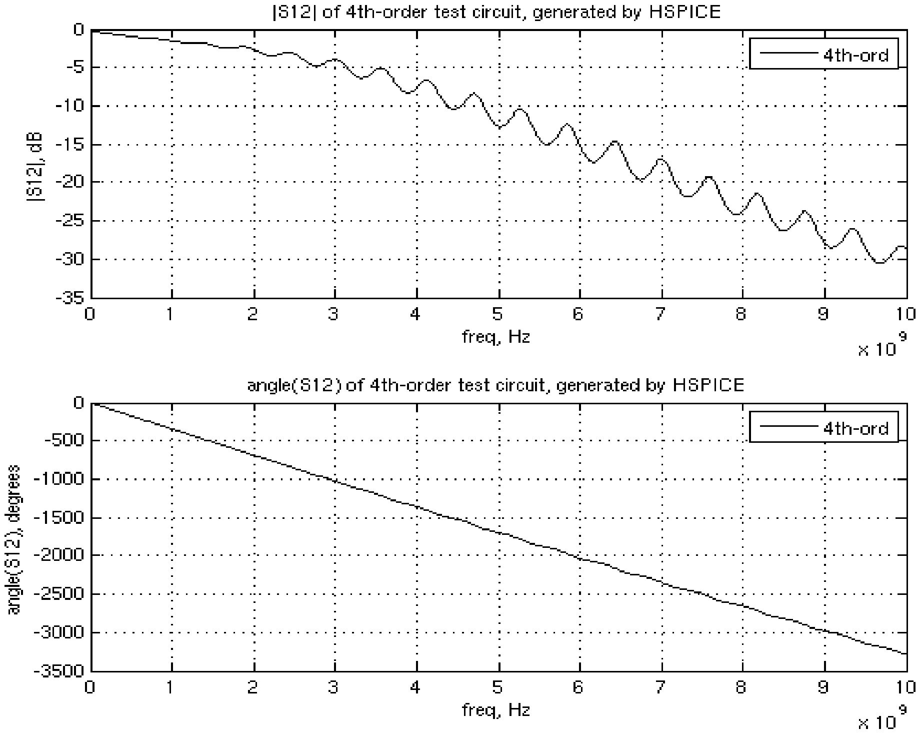


Figure 4-7: ***S_{12}*** **of the 4th-Order Test Circuit with Delay, as Calculated in HSPICE**

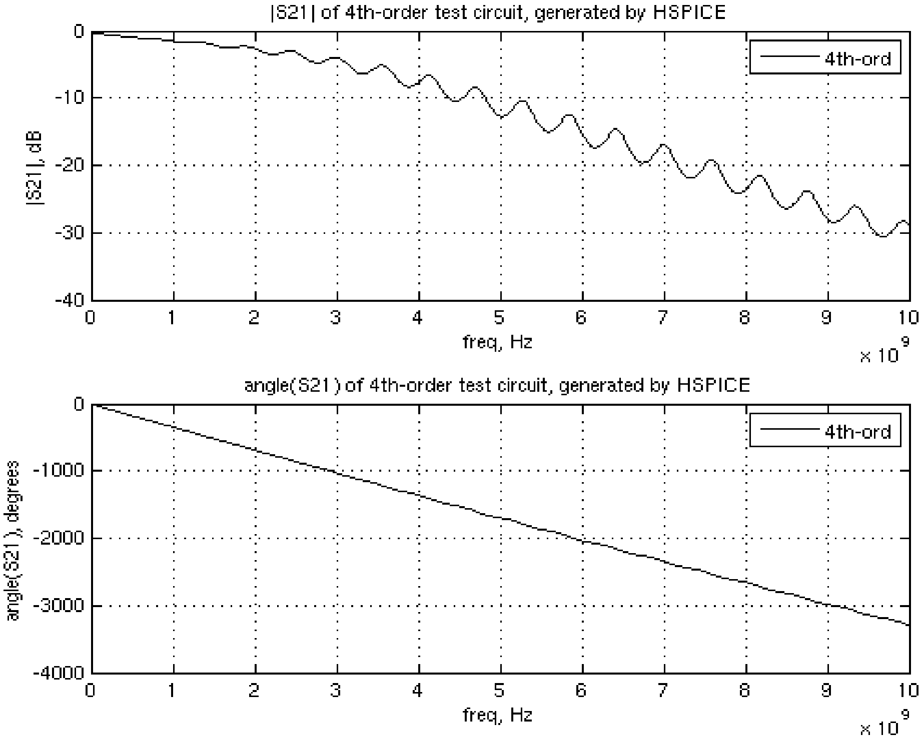


Figure 4-8: S_{21} of the 4th-Order Test Circuit with Delay, as Calculated in HSPICE

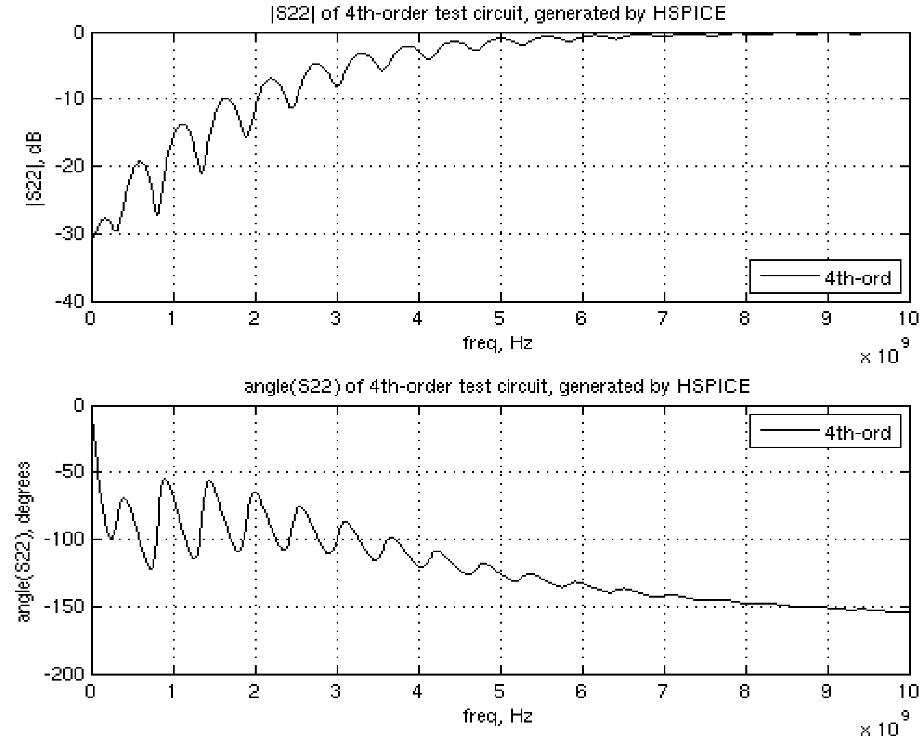


Figure 4-9: S_{22} of the 4th-Order Test Circuit with Delay, as Calculated in HSPICE

4.3.1.2 Independent Validation of the Scattering Parameter Matrix

Because of the ripple features of the scattering parameters' magnitude responses, primarily due to the presence of the lumped elements in the test circuit of Figure 4-2, the scattering parameters were re-extracted over a broader frequency range out to 20 GHz in HSPICE, and also validated independently by calculating them independently in ANSYS Designer [33]. The results are shown to be virtually equivalent between the two tools and are shown in Figure 4-10 through Figure 4-13.

Ideally, in an industrial design flow, the scattering parameters would be verified by fabricating the hardware and performing empirical measurements using a vector network analyzer (VNA) to validate the simulated scattering parameters are valid. Often, however, there are practical limitations regarding how effectively this can be done. In such a case, duplicating the simulation on a different tool platform can provide some additional confidence that the results are valid. This is a common technique when the design processes have a significant pre-fabrication emphasis, which is often the case.

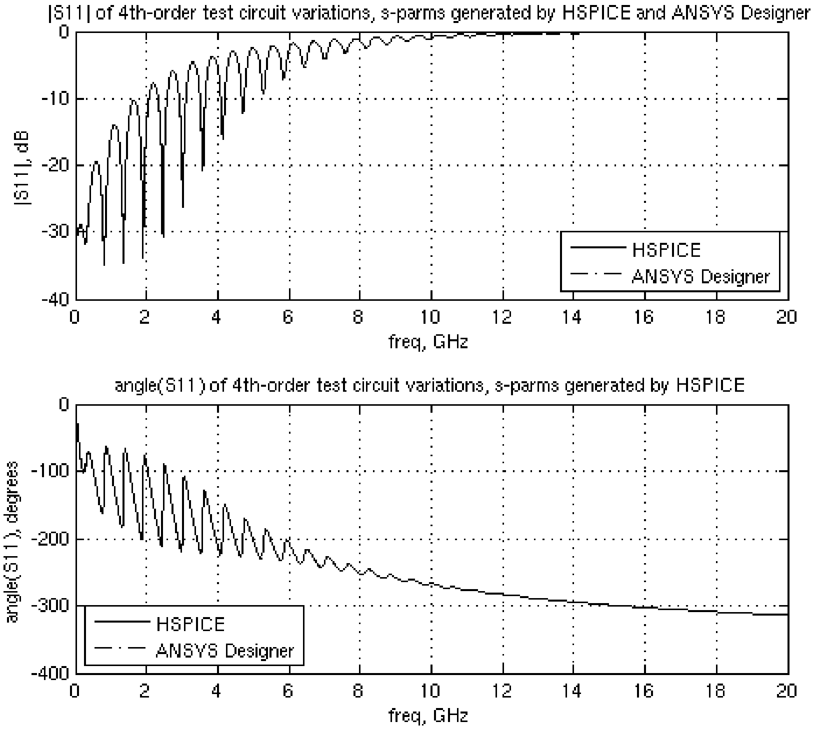


Figure 4-10: S_{11} of the 4th-Order Test Circuit, as Calculated in HSPICE and ANSYS Designer

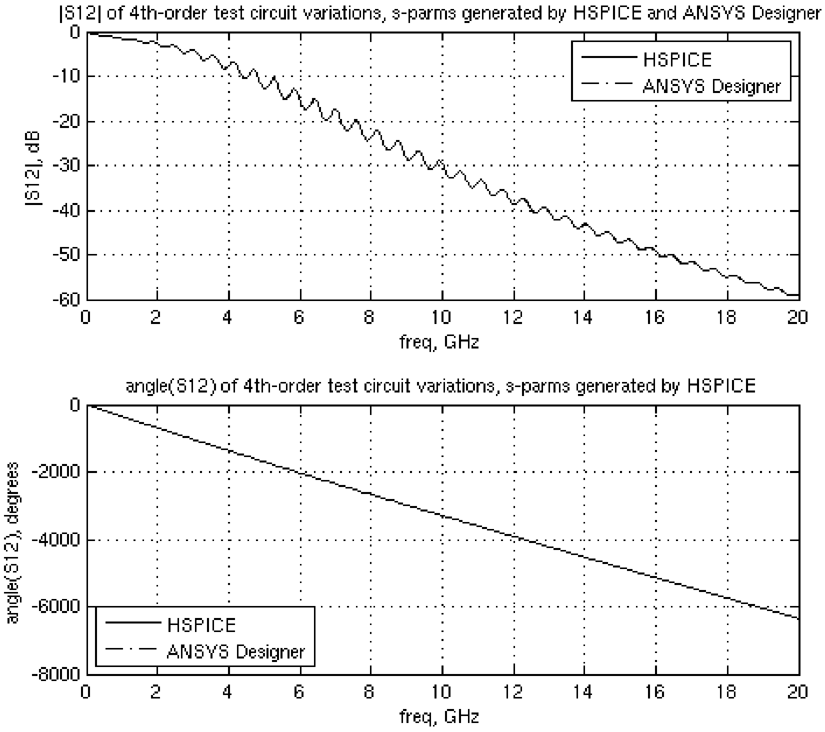


Figure 4-11: *S_{12}* of the 4th-Order Test Circuit, as Calculated in HSPICE and ANSYS Designer

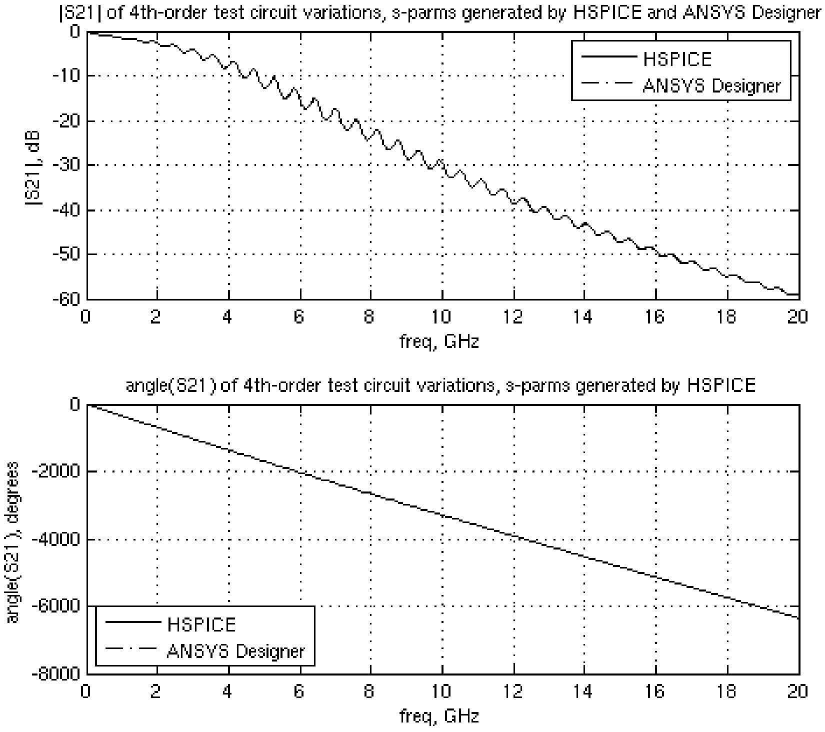


Figure 4-12: *S_{21}* of the 4th-Order Test Circuit, as Calculated in HSPICE and ANSYS Designer

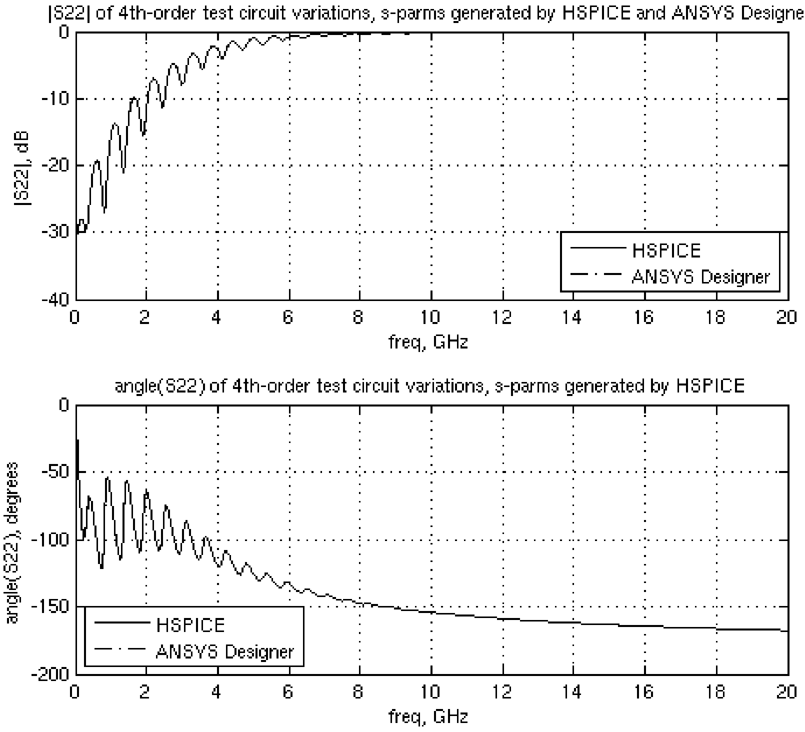


Figure 4-13: *S_{22}* of the 4th-Order Test Circuit, as Calculated in HSPICE and ANSYS Designer

4.3.1.3 Calculation of the System Analog Frequency Transfer Function *$H_a(f)$* from the Scattering Parameter Matrix

After the initial characterization step, an analog domain frequency transfer function, $H_a(j\Omega) = H_a(f)$, can now be derived from the test circuit's scattering parameters using the relation [76]

$$\frac{V_{out}(j\Omega)}{V_S(j\Omega)} = \frac{S_{21}(1+\Gamma_L)(1-\Gamma_S)}{2(1-S_{22}\Gamma_L)(1-\Gamma_{in}\Gamma_S)}, \tag{4.1}$$

where

$$\Gamma_{in} = S_{11} + S_{12}S_{21}\frac{\Gamma_L}{1-S_{22}\Gamma_L}, \tag{4.2}$$

and where the general configuration is shown in Figure 4-14. Here Γ_L and Γ_S are the reflection coefficients looking into the load and source, respectively, and thus Γ_{in} is the reflection coefficient looking into the network terminated in the load impedance corresponding to Γ_L. The general configuration is shown in Figure 4-14.

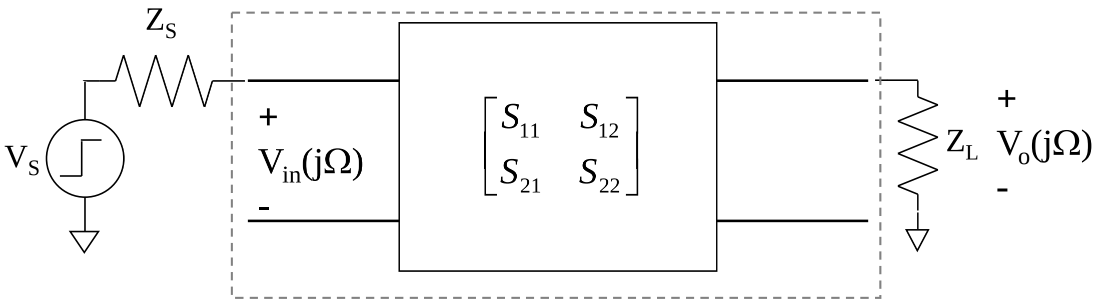


Figure 4-14: Definition of Parameters for Frequency Transfer Function Calculation

It should be noted that the definition of the input voltage associated with the transfer function $H_a(s)$ here is as $V_{in}(j\Omega)$ is as shown in the figure, and that

$$V_{out}(j\Omega) = H_a(j\Omega)V_{in}(j\Omega) \quad . \tag{4.3}$$

Here $H_a(s)$ was calculated using the commercially-available MATLAB function s2tf.m [58] to implement equations (4.1) and (4.2) above, by assuming $Z_S = 0$ and $Z_L = 50\ \Omega$, and resulting in $H_a(j\Omega) = V_{out}(j\Omega)/V_{in}(j\Omega) = V_{out}(j\Omega)/V_S(j\Omega)$, which is only true for $Z_S = 0$. It should be noted that the use of $Z_S = 0$ is not meant to represent the true output impedance of the driver circuit, but rather is a method to compel the s2tf.m function to calculate the transfer function per the definition in (4.3) and Figure 4-14. By defining the transfer function in this way, the source impedance Z_S is separated from the transfer function $H_a(s)$, and thus must be accounted for as part of the driver circuit.

4.3.1.4 Relationship Between the Frequency Transfer Function *H_a(f)* and the Scattering Parameter *S₂₁*

For the 4th-order test circuit, $H_a(j\Omega)$ was calculated using the commercially-available MATLAB function s2tf.m [58] to implement equations (4.1) and (4.2) above, by assuming $Z_S = 0$ and $Z_L = 50\ \Omega$, resulting in $H_a(j\Omega) = V_{out}(j\Omega)/V_S(j\Omega)$. Note that the calculation of the frequency transfer function requires knowledge of the source's output impedance Z_S and the load impedance Z_L. As a result, it is clear that the frequency transfer functions and transfer functions for the interconnect are application-specific, as should be expected [40]. Figure 4-15 shows the magnitude and phase responses for the frequency transfer function $H_a(j\Omega) = H_a(f)$ for the 4th-order test circuit, plotted with the S_{21} scattering parameter for comparison.

Note that the frequency transfer function and S_{21} parameter have similar shapes, but that they are not identical. This has lead to S_{21} being commonly cited as an approximation for $H_a(j\Omega)$ [77]. In fact, for the case of an ideal transmission line with characteristic impedance $Z_o = Z_L$, where $S_{11} = S_{22} = 0$, and where $\Gamma_S = -1$ $(Z_s = 0)$ and $\Gamma_L = 0$, (4.2) reduces to

$$\Gamma_{in}|_{\,S_{11}=S_{22}=\Gamma_L=0} = 0 + S_{12}S_{21}\frac{0}{1-0\cdot 0} = 0, \tag{4.4}$$

and (4.1) becomes

$$\left.\frac{V_{out}(j\Omega)}{V_S(j\Omega)}\right|_{\Gamma_S=-1,\,\Gamma_L=0} = \frac{S_{21}(1+0)(1-(-1))}{2(1-0\cdot 0)(1-0\cdot(-1))} = S_{21}\,. \tag{4.5}$$

This is clearly not true in general, but rather only under the stated conditions for the transmission line and load and source impedances.

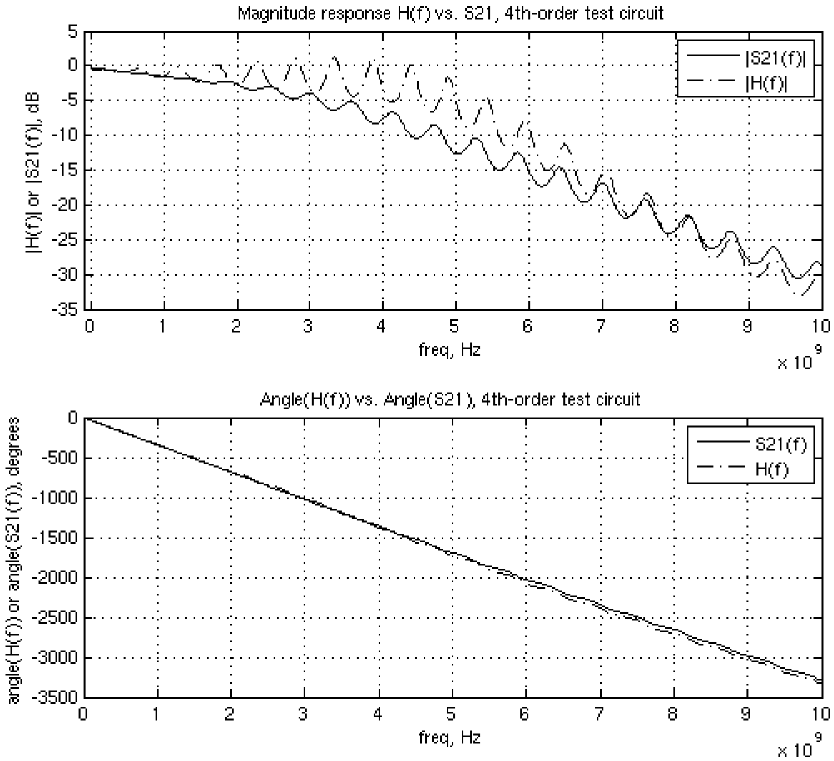


Figure 4-15: Frequency Transfer Function *Ha(jΩ)* for the 4th-Order Test Circuit as Calculated by MATLAB's s2tf.m Function

After deriving the analog frequency transfer function $H_a(j\Omega)$, the analog complex frequency s-domain transfer function $H_a(s)$ can be determined as a rational function approximation using a vector fitting process, such as that developed by Gustavsen [44]. Obtaining a model in the complex frequency domain has several advantages, including the ability to study the model in the complex s-plane using pole-zero analysis. It is also possible to convert $H_a(s)$ into the complex digital transfer function $H_d(z)$ using a bilinear transform [41], a matched-Z transform [78], an impulse invariant method transform [79], or other methods. This transformation enables the direct use of digital signal processing (DSP) techniques to study the model in the complex z-plane, again using pole-zero analysis, and enabling the application of yet another well-developed body of analytical theory and methods. These methods will be addressed in subsequent chapters.

4.3.2 System Identification Using Linear Least Square Error (LLSE) Method

Another widely-recognized method for identification of system transfer functions, in this case taken from the field of traditional system identification, is that based on the minimization of linear least square error (LLSE). This method is based on an assumed model structure and utilizes time-domain waveform data. One common model structure that suits our case is the auto-regressive with exogenous input (ARX) structure [65]. This method is implemented in the arx.m function in the MATLAB system identification toolbox [67], which is utilized here for convenience. The development of models based on this method will be compared to the analog frequency domain's scattering parameter system identification method of the last section, and the discrete frequency-domain's system identification method of the next section.

Figure 4-16 shows the transient input and output waveforms for the 4th-order test circuit, for the case with source output impedance $Z_S = 50\ \Omega$. The sampled form of these waveforms can be used, along with the sampling rate, to generate an ARX model for the test circuit. Application of the function, assuming a sampling interval $T_s = 5$ ps, results in the following transfer function model in the complex discrete frequency domain for the 4th-order test circuit:

$$H_d(z) = \frac{(2.958z^4 + 1.56z^3 - 6.979z^2 + 9.7700z - 4.887)(10^{-4})}{z^4 - 2.662z^3 + 2.102z^2 - 0.2087z - 0.2314}\ z^{-172} \qquad (4.6)$$

which can be seen to be a 4th-order system with delay, as expected. Note that one advantage of this method is that the output of the process comes out naturally in the desired form of the z-domain transfer function $H_d(z)$. The LLSE-based method as implemented in the arx.m function results in a transfer function $H_d(z)$ expressed as a ratio of two polynomials, $B_d(z)/A_d(z)$, which lends itself naturally to studying poles and zeros in the complex frequency domain. This is an advantage, because from this form the pole-zero map can be created, which is one of the major objectives of this research project.

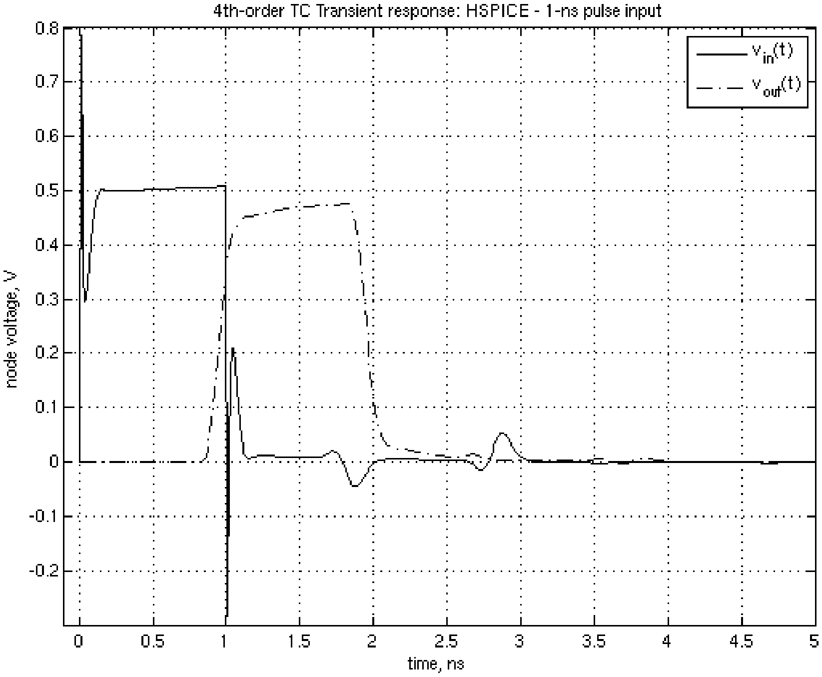


Figure 4-16: Transient Waveforms Calculated by SPICE for the 4th-Order Test Circuit for the Case When $Z_s = 50\ \Omega$ and a 1-ns Pulse Input

Figure 4-17 shows the pole-zero plot in the z-plane for the 4^{th}-order test circuit. Finally, note also the presence of the 172 multiple poles at $z = 0$, also shown on the plot of Figure 4-17. From the shifting property of the z-transform, which states that

$$\mathcal{Z}\{v(n - n_o)\} = V(z) \cdot z^{-n_o}, \tag{4.7}$$

where

$$\mathcal{Z}\{v(n)\} = V(z), \tag{4.8}$$

it can be seen that the z^{-172} term corresponds to a pure delay of $172 \times 5\, ps = 860$ ps, and recall that the original transmission line in the 4th-order test circuit carried a propagation delay of 860 ps. It might be reasonable to expect, based on this observation, that the model of (4.6) might have very good phase performance, since the phase response is an important characteristic of a system with propagation delay.

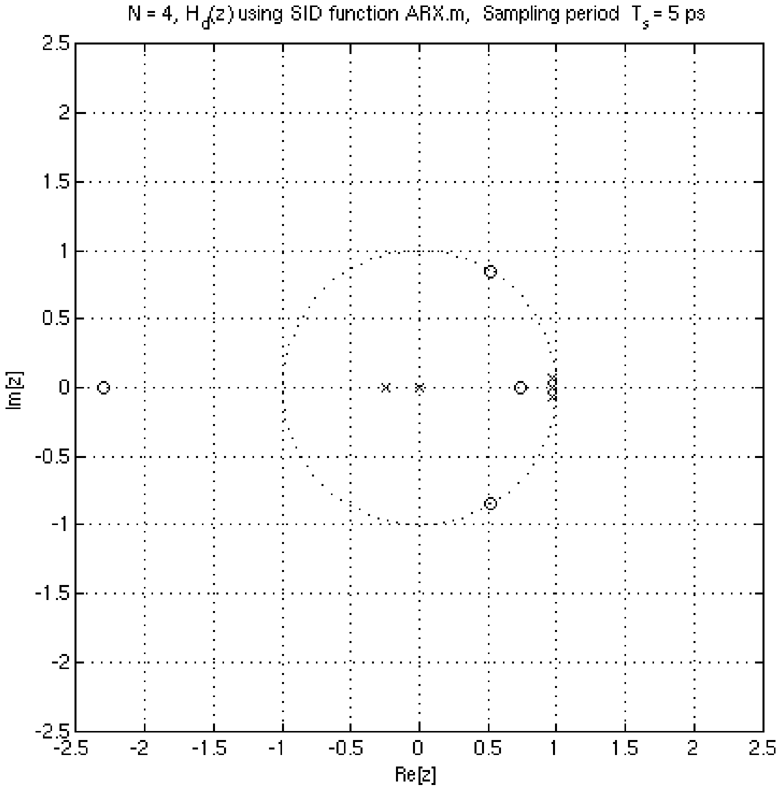
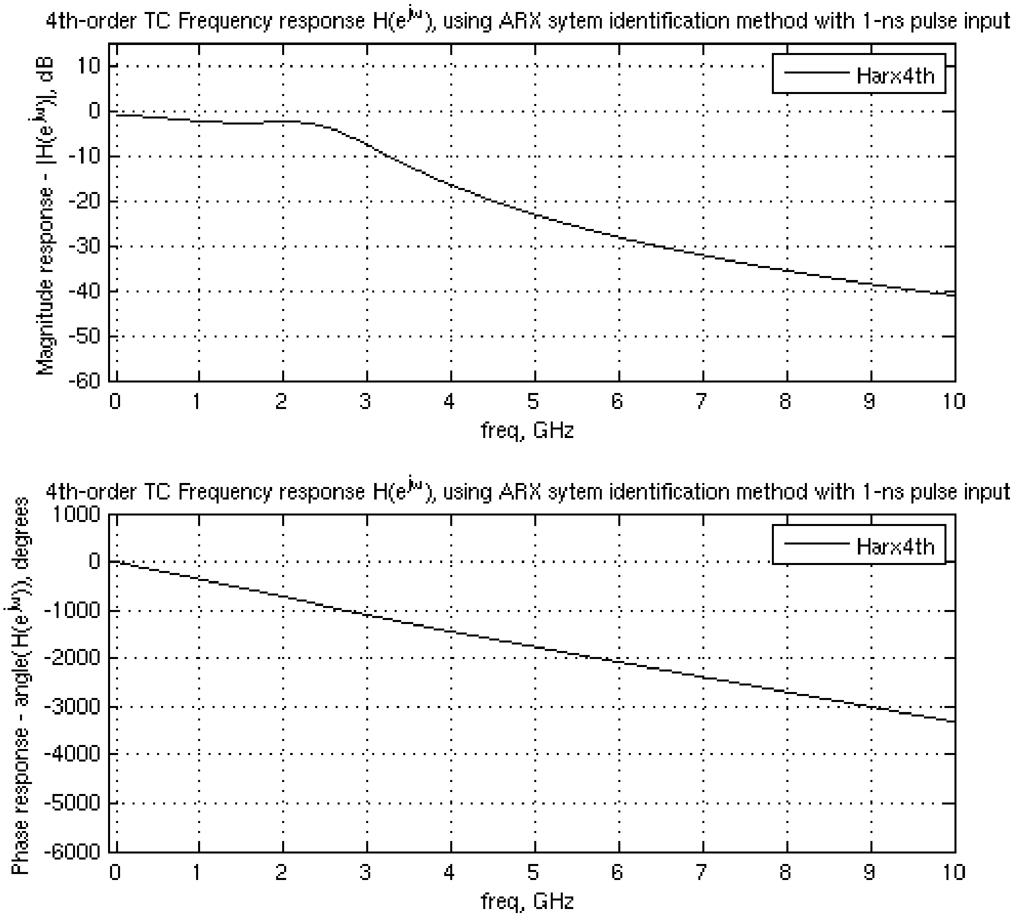


Figure 4-17: Pole-Zero Map of the 4th-Order Test Circuit in the z-Plane Using SID arx.m Function

Figure 4-18 shows the frequency response plot for the model $H_a(z)$ expressed in (4.6). Note that the phase response is indeed very good, corresponding well with the phase results for the analog frequency transfer function $H_a(f)$, as shown in Figure 4-15. The correlation is not as good in the magnitude response, however, as can been seen by noting that the significant rolloff or “knee” in the magnitude curve does not occur at the same frequency, nor does the ripple effect that is so evident in Figure 4-15 show up in Figure 4-17. Nevertheless, the fact that the ARX method suggests a 4th-order model for a system with four reactive elements, and with a separated delay term approximately equal to the known propagation delay of the system, is satisfying. It may be reasonable, based on these results, to surmise that accurately modeling the ripple effects may require a model for $H_a(s)$ or $H_d(z)$ of significantly higher order than four.


4th-order TC Frequency response H(e^{jw}), using ARX sytem identification method with 1-ns pulse input

4th-order TC Frequency response H(e^{jw}), using ARX sytem identification method with 1-ns pulse input

Figure 4-18: Frequency Response of the 4^{th}-Order Test Circuit Transfer Function $H_d(z)$ Obtained by LLSE ARX System Identification Method

Finally, the transfer function $H_d(z)$ can be assessed by comparing its time response to that given by originally by SPICE, since knowledge of the discrete transfer function $H_d(z)$ is equivalent to being able to express the system's difference equations in the discrete time domain [49]. This is analogous to the fact that knowledge of the analog transfer function $H_a(s)$ is equivalent to being able to express the system's differential equations in the continuous time domain [40]. Figure 4-19 shows the output waveform generated using MATLAB's lsim.m function [80] and its comparison to the original

SPICE waveforms. It can be seen that the system propagation delay, edge rate, and attenuation characteristics are all slightly off the original SPICE waveforms, although the system propagation delay is captured fairly well.

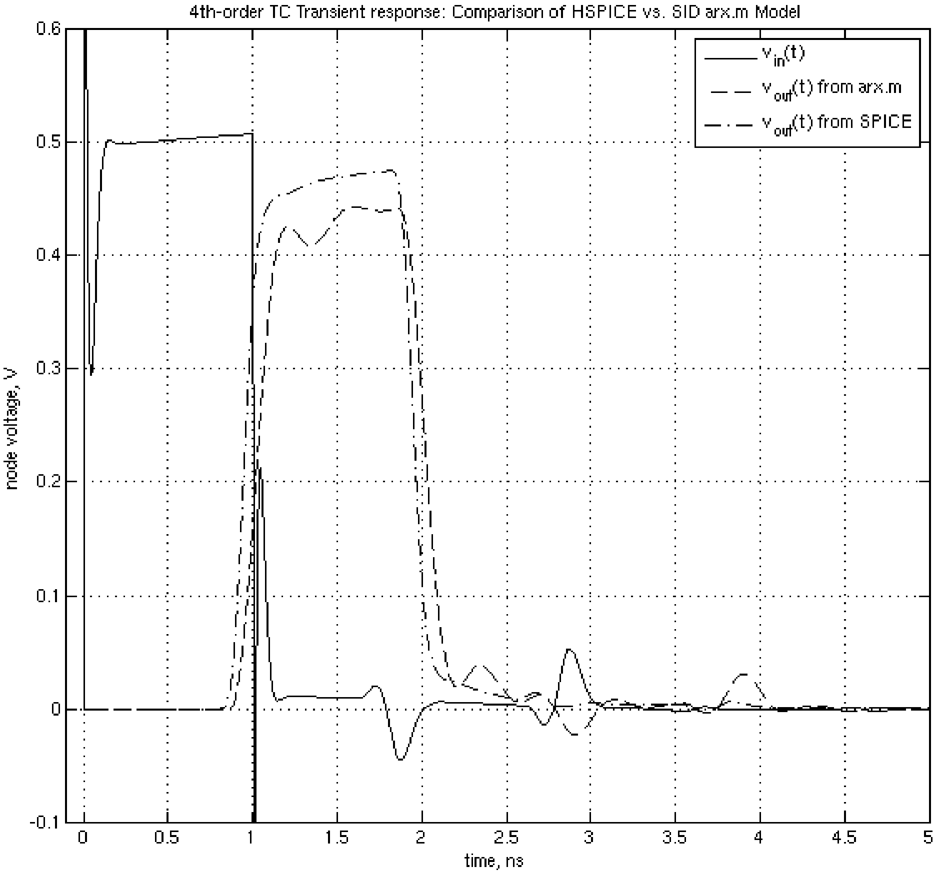


Figure 4-19: Transient Response of the 4th-Order Test Circuit Transfer Function *$H_d(z)$* Obtained by LLSE ARX System Identification Method, vs. SPICE

4.3.3 System Identification Using a Method to Estimate the Frequency Transfer Function from Time- and/or Frequency-Domain Waveform Data

As stated previously, the motivation is to study the test circuit of Figure 4-2 in the complex frequency domain by calculating a system transfer function $H_a(s)$ or $H_d(z)$. The previous section used system identification techniques based on input and output waveforms in the sampled time domain, assumed an ARX model structure, and thus represented a parametric method. Another technique for deriving the transfer function is to use known system identification methods that utilize frequency-domain input and output waveforms. One such method that is well-known in the system identification literature is the empirical transfer function estimate (ETFE), which represents a non-parametric system identification method, in that the resulting estimate has no assumed underlying model structure [65]. The ETFE is given by the following:

$$H_d(e^{j\omega}) = \frac{V_{out}(e^{j\omega})}{V_{in}(e^{j\omega})}\ , \tag{4.9}$$

where $V_{out}(e^{j\omega})$ and $V_{in}(e^{j\omega})$ are the output and input waveforms, respectively, in the discrete frequency domain and N is the length of the output and input data sequences.

Here, the input transient waveform was again assumed to be known as an ideal 1-ns pulse of known risetime, and the output transient response waveform was again derived using the HSPICE simulator directly in transient analysis mode, just as in the previous section. These waveforms are represented in Figure 4-20 for the case where the source has output impedance $Z_S = 0$.

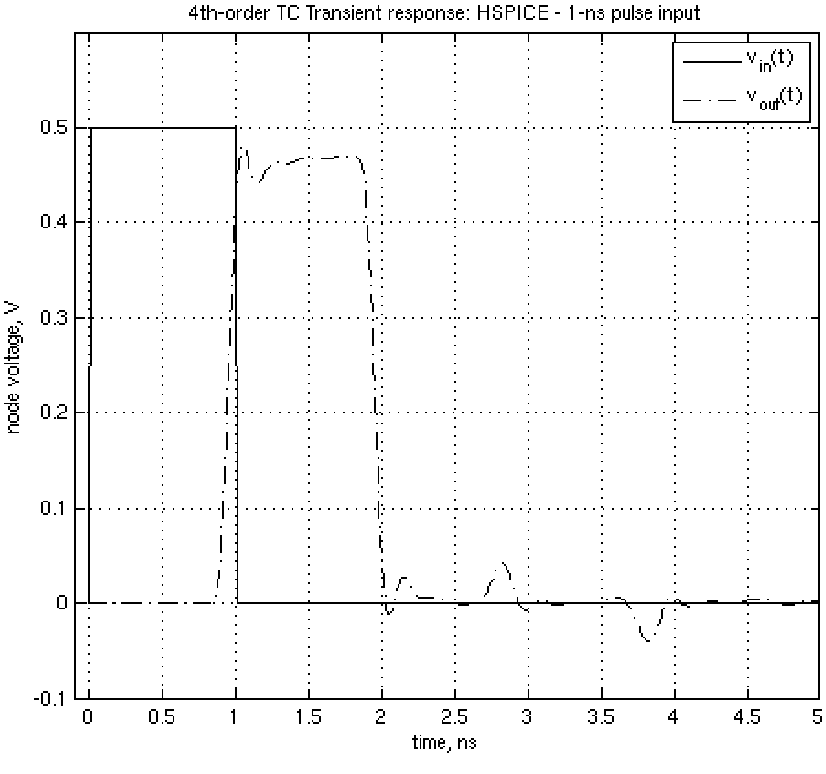


Figure 4-20: Transient Waveforms for the 4th-Order Test Circuit for Case with $Z_S = 0$ and a 1-ns Pulse Input

Note again that, in a normal signal integrity design flow, the objective of the signal integrity analysis would be met here, since the typical goal of predicting the system's transient output waveform would have been achieved. In this case, however, there is an additional objective of obtaining and examining the system's transfer function model in the complex frequency domain, so the SID methods are now employed. Once the transient waveforms have been obtained from the SPICE simulator, they are re-sampled to have a uniform time step and transformed into the sampled digital frequency domain using an Fast Fourier transform (FFT) implementation of the discrete Fourier transform (DFT) [41]. The DFT representations are depicted in Bode plots for the transformed waveforms, as shown in Figure 4-21.

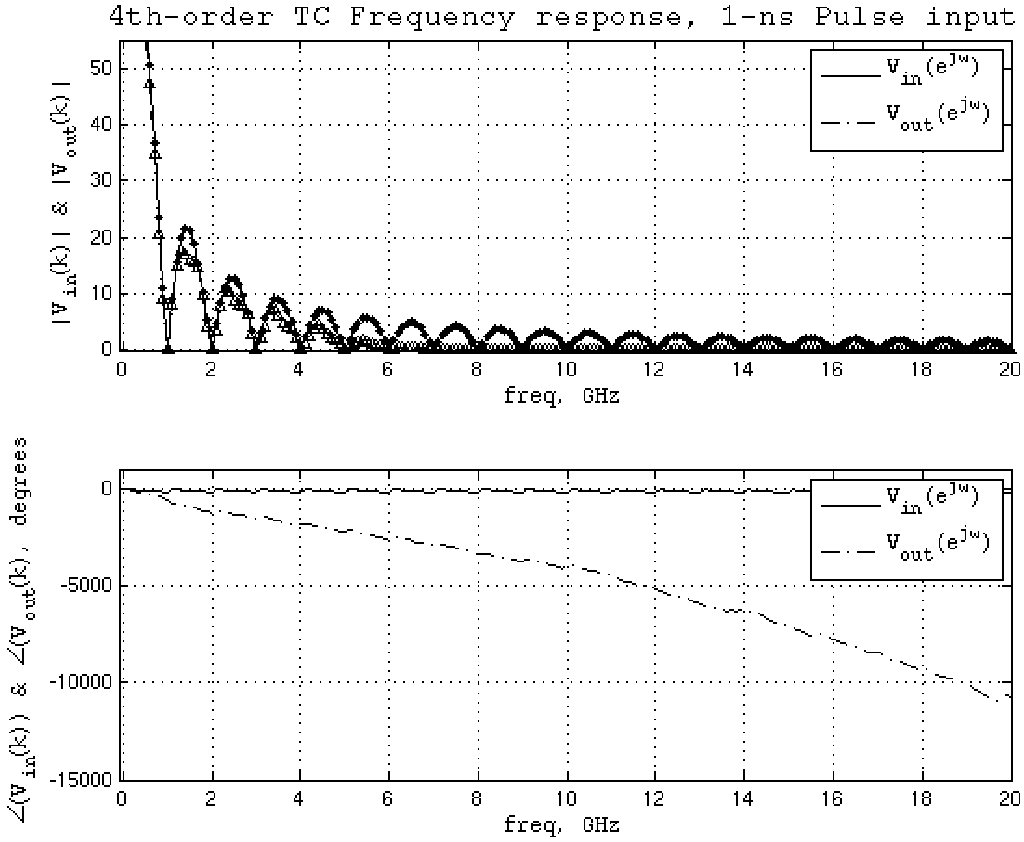


Figure 4-21: Frequency-Domain Input and Output Signals for the 4th-Order Test Circuit for the Case Where $Z_s = 0$ and for a 1-ns Pulse Stimulus

Applying (4.9), an estimate can be calculated for the test circuit's frequency transfer function in the discrete frequency domain. This estimate for $H_d(e^{j\omega})$ is shown as the solid curve in Figure 4-22. For comparison, note the shape of the frequency response for the function $H_a(j\Omega)$, derived using (4.1) and (4.2), and depicted in Figure 4-15. It can be seen when examining the $H_d(e^{j\omega})$ frequency response curve in Figure 4-22 that there are unnatural spikes in the magnitude response due to the zero values present in the 1-ns pulse input waveform's DFT representation of the DTFT function $V_{in}(e^{j\omega})$, as depicted in Figure 4-21, and which end up in the denominator of $H_d(e^{j\omega})$ in (4.9). This is a common occurrence when using the ETFE, and the general procedure is to ignore such values when forming the estimate [65].

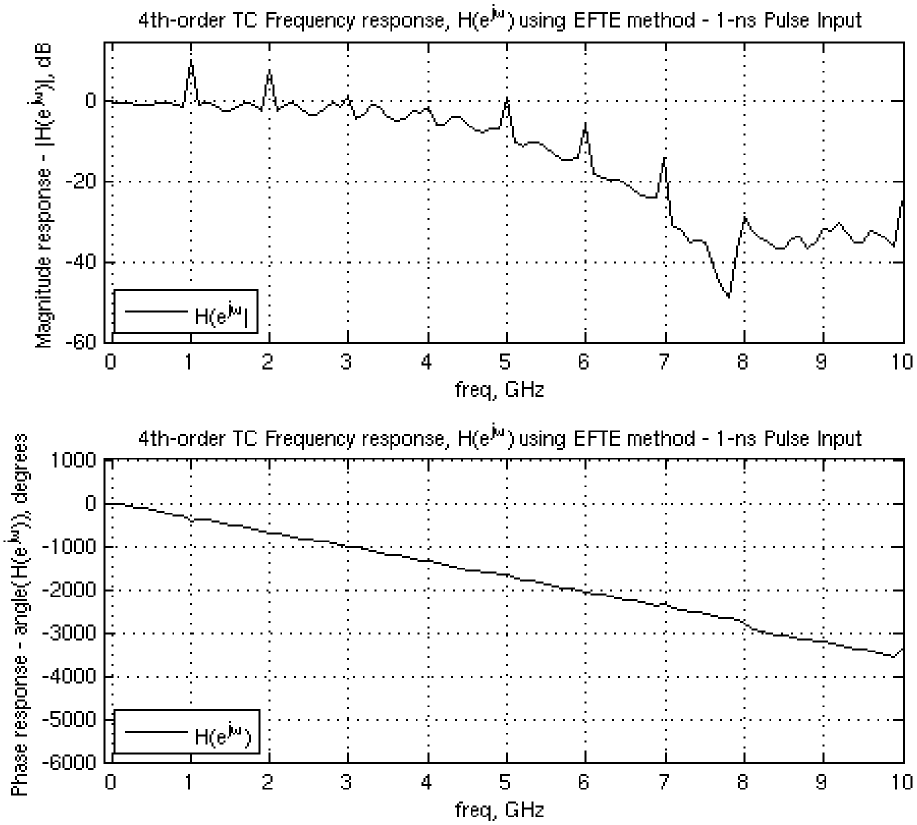


Figure 4-22: Frequency Response of 4th-order Test Circuit Frequency Transfer Functions $H_d(e^{j\omega})$ Obtained by the Empirical Transfer Function Estimate System Identification Method

Another option would be to pick an input excitation waveform that would not contain any zeros in its DFT representation of the DTFT function $V_{in}(e^{j\omega})$. Indeed, it can be seen by examination of the magnitude responses for $V_{out}(e^{j\omega})$ and $V_{in}(e^{j\omega})$ in Figure 4-21 that the frequency content is actually rather sporadic, with widely-varying amplitude and periodic nulls in both the input and output waveforms. It would be desirable to utilize a time-domain input waveform that would have more robust and consistent frequency-domain content over the frequency interval being considered.

From this standpoint, consider an impulse stimulus function, given by the N-point sequence $v_{in}(n) = \delta(n)$, where $\delta(n)$ is the Dirac delta function, defined as

$$v_{in}(n) = A\delta(n) = \begin{Bmatrix} A, & n = 0 \\ 0, & n \neq 0 \end{Bmatrix}. \tag{4.10}$$

The Dirac delta function is known to have extremely robust frequency content, as can be seen from its N-point discrete Fourier transform (DFT) representation $V_{in}(k)$, given by

$$V_{in}(k) = \begin{Bmatrix} A, & 0 \leq k \leq N-1 \\ 0, & otherwise \end{Bmatrix} [41]. \tag{4.11}$$

Thus it is seen that the function $V_{in}(k)$ has uniform magnitude response frequency content in the sampled digital frequency domain. Note the similarity here to the uniform frequency content obtained for the scattering-parameter identification method used in the analog frequency domain earlier, obtained in that case by a sinusoidal frequency sweep process over the desired frequency range. Because of the more robust frequency content, it is reasonable to expect that the input function given in (4.10) might lead to a better ETFE result, if for no other reason than the zero-valued data points are no longer present, and due to the fact that all frequency values over the range will get represented equally.

Figure 4-23 shows the transient waveforms for an HSPICE simulation using an impulse input. HSPICE was used with an impulse at $t = 1$ ns, implemented as a single sample of $0.5\,V$ at that time instant using HSPICE's piecewise linear (PWL) function [29]. Figure 4-24 shows $V_{out}(e^{j\omega})$ and $V_{in}(e^{j\omega})$ for the case of the Dirac delta function input. From these two figures, it can be seen that the impulse response stimulus has the disadvantage of reduced amplitude of the time-domain signal $v_{out}(t)$, or $v_{out}(n)$, at the system output. This is evident in the time-domain waveforms of Figure 4-23, and could easily lead to possible inaccuracies in the time-domain if the waveforms are used directly for time-domain system identification. The advantage of the impulse stimulus is fully realized, however, when transforming to the frequency domain, given that the output waveform $V_{out}(e^{j\omega})$ has better amplitude characteristics than its frequency-domain counterpart derived with the 1-ns pulse input stimulus, and as can be clearly seen by comparing the frequency-domain waveforms for $V_{out}(e^{j\omega})$ and $V_{in}(e^{j\omega})$ in Figure 4-21 and Figure 4-24.

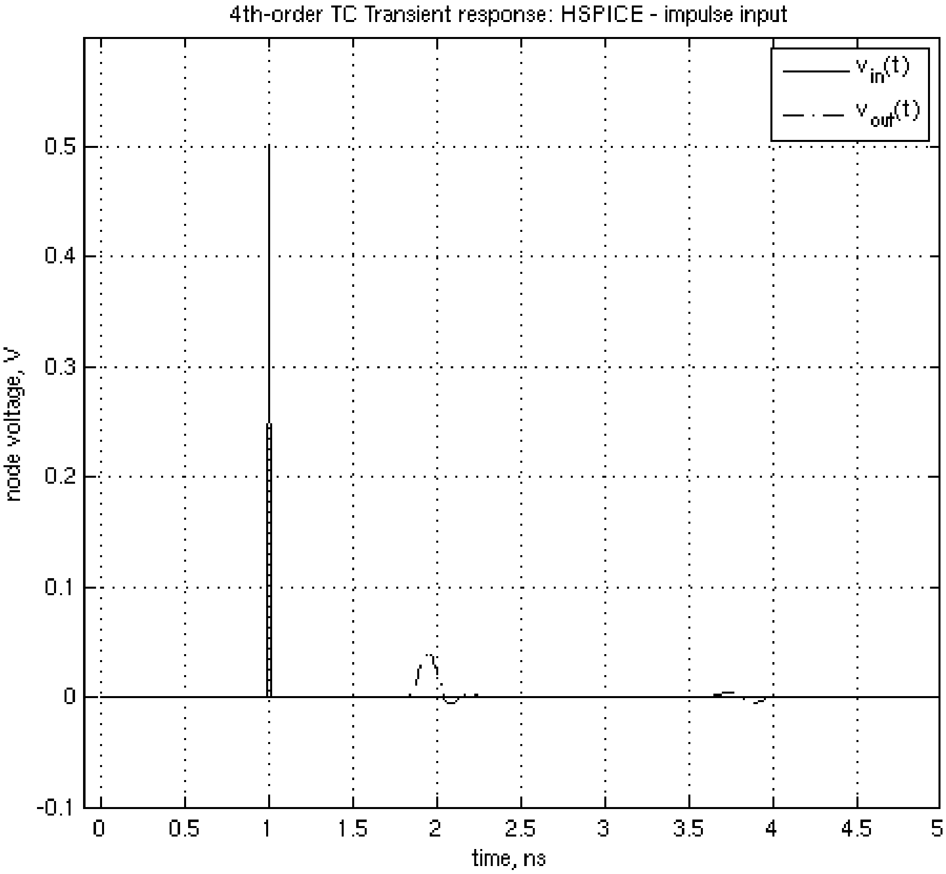


Figure 4-23: Transient Waveforms for the 4th-Order Test Circuit for Source with $Z_S = 0$ and an Impulse Input Stimulus

Figure 4-25 compares the derived frequency transfer functions $H_d(e^{j\omega})$ obtained using the Dirac delta impulse and 1-ns pulse stimuli. Note that the response obtained using the impulse response does not exhibit the spikes due to the zero-valued points in the frequency domain representation of the 1-ns pulse input waveform, as expected. It appears that the use of the impulse stimulus is a better approach for obtaining the frequency transfer function when using the ETFE method.

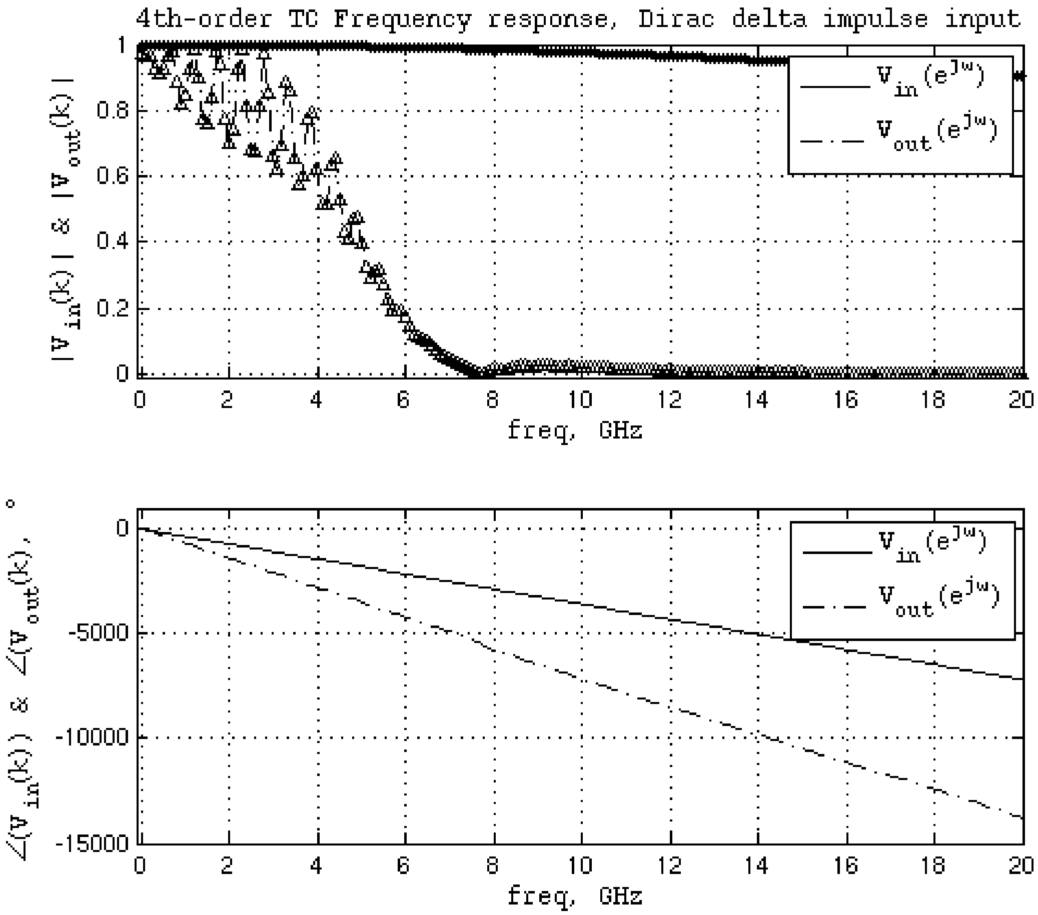


Figure 4-24: Frequency-Domain Waveforms for the 4th-Order Test Circuit for Source $Z_S = 0$ and an Impulse Input Stimulus

Note that frequency and/or transfer function models for the 4th-order test circuit have been obtained in three ways. The analog frequency transfer function $H_a(j\Omega)$ was obtained using scattering parameter characterization to identify the model in the continuous analog frequency domain. The discrete frequency transfer function $H_d(e^{j\omega})$ was obtained using both a time-domain SID method assuming an ARX model structure, and by using the time-domain input and output waveforms' DFT representations in the discrete frequency domain to calculate the empirical transform function estimate, with no assumed model structure. The ARX method gives the transfer function representation $H_d(z)$, and subsequently $H_d(e^{j\omega})$, while the ETFE only gives the frequency transfer function $H_d(e^{j\omega})$. It will now be possible to examine the system in more detail using common DSP methods, both by examining the frequency transfer functions themselves further, and also by deriving and examining the associated system transfer functions in the z- and s-planes for the complex digital and analog frequency domains, respectively. This latter analysis in the complex frequency domains will be undertaken in later chapters.

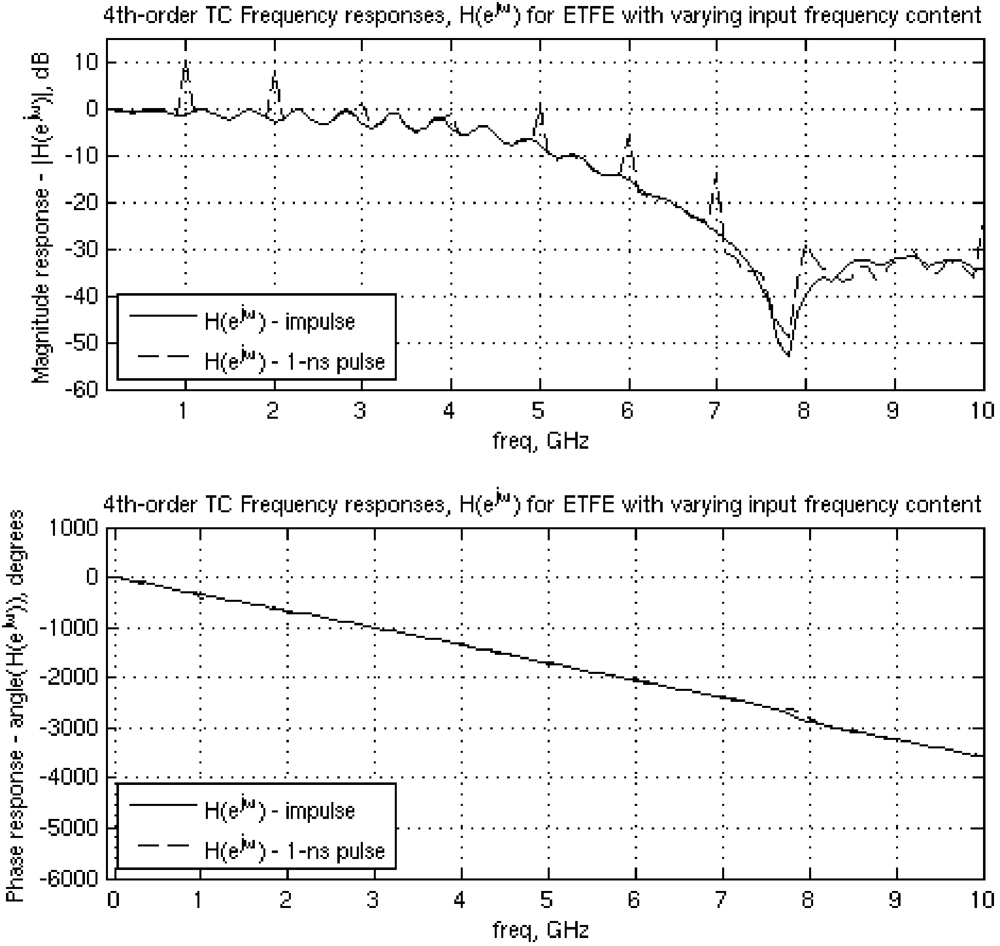


Figure 4-25: Frequency Responses of 4th-Order Test Circuit Frequency Transfer Functions $H_d(e^{j\omega})$ Obtained Using Empirical Transfer Function Estimate with Varying Input Frequency Content

4.4 Comparison of Frequency Responses Generated Using the Various Transfer Function Models

To this point, frequency responses have been calculated for the 4^{th}-order test circuit using three different methods. First, the analog frequency response $H_a(j\Omega)$ was formulated using scattering parameters as the system identification method, with subsequent conversion of the scattering parameters to the system frequency transfer function $H_a(j\Omega)$ using (4.1) and (4.2). Second, the discrete frequency response $H_d(e^{j\omega})$ was generated from $H_d(z)$ using the ARX system identification function. Finally, the frequency response $H_d(e^{j\omega})$ was generated using the empirical transfer function estimate. The three discrete frequency responses are shown, for both magnitude and phase, in Figure 4-26, along with the analog frequency response $H_a(j\Omega)$ for the 4th-order test circuit.

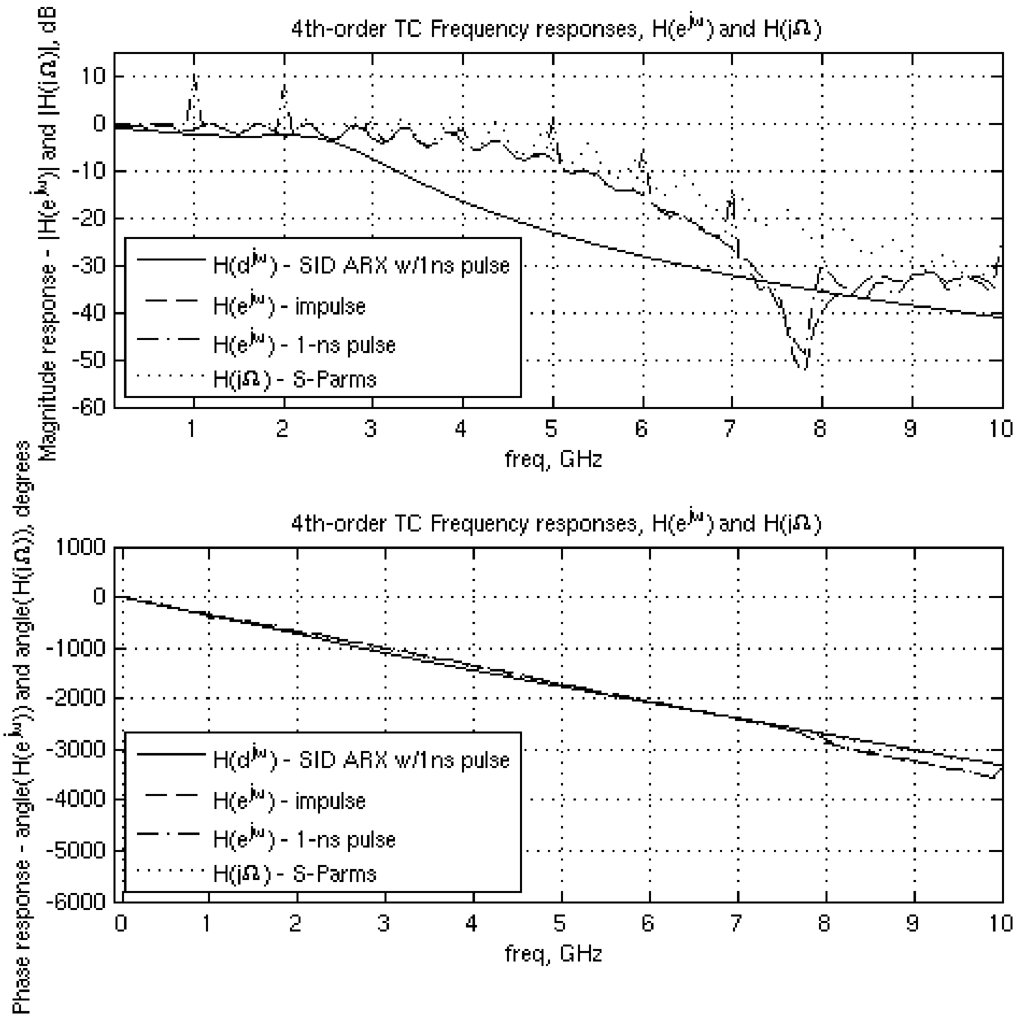


Figure 4-26: Comparison of Frequency Responses $H_d(e^{j\omega})$ and $H_a(j\Omega)$ Obtained Via LLSE ARX-Based, ETFE-Based, and Scattering Parameter Based System Identification Methods

4.5 Conclusions

In this chapter, three different system identification methods were used to identify a model for the 4th-order test circuit, with the objective of comparing and contrasting the approaches for the purpose of choosing a primary method to utilize for the remainder of this research project. The first method identified the model in the analog frequency domain (f-domain). The second method identified the model in the complex digital frequency domain (z-domain). The third method identified the model in the digital frequency domain (ω-domain).

The first method, utilizing scattering parameters in the analog frequency domain, has the advantages of working well at extremely high frequencies, being implementable on commonly-available VNA measurement equipment or commonly-available software simulation packages, and leveraging well-known theory from the field of RF engineering. This method also has the advantage of being the most commonly practiced in the field of signal integrity engineering. The disadvantages are that the resulting frequency transfer function cannot be directly examined in the digital frequency domains, or easily compared to functions expressed in those domains.

The second method, utilizing an LLSE-based system identification method in the time domain, has the advantage of giving output directly in the complex z-domain, which lends itself directly to analysis in that domain. It also has the advantage of easily separating out the delay terms as a z^{-n_o} term. It is characterized by a known model structure (ARX) with assumed order, but this can be a disadvantage if the assumption is not a good one.

The third method, using the empirical transfer function estimate, has the advantage of working well with experimental data, but has no assumed underlying model structure. It is expressed in the digital frequency domain (ω -domain), and cannot be easily compared to functions expressed in the analog frequency domains.

Because of the advantages associated with the scattering parameters method, namely its applicability to high frequencies and the high degree of familiarity with it in the field of signal integrity engineering, it will be the primary system identification method used in this study from this point forward. The advantages of examining the system model in the complex frequency domains are evident from Figure 4-17. It would be advantageous to find a method for examining the 4th-order test circuit in both the complex analog frequency domain (s-domain) and the complex digital frequency domain (z-domain) each time a frequency response is obtained.

Chapter 5: Development of the Transfer Function Model $H_a(s)$

5.1 Introduction

As seen in the previous chapter, examination of the system transfer function model in the complex frequency domain, either analog or digital, has the potential to be useful for the analysis of interconnect models for use in signal integrity applications. It is known that there are computational advantages to expressing the system transfer function model as a rational function approximation $H_a(s)$ in the complex analog frequency domain [44], [72], namely,

$$H_a(s) = \frac{B_a(s)}{A_a(s)}, \tag{5.1}$$

and insight can also be gained by examining the model $H_a(s)$ in the complex s-domain plane by studying the pole-zero maps [40], [48]. For both of these reasons, it can be advantageous to formulate the system model in the complex analog frequency domain.

Similarly, it may be reasonable to expect that the same advantages would be gained by formulating the system transfer function model as a rational function approximation $H_a(z)$ in the complex digital frequency domain. In such a case, the expression would take the form

$$H_d(z) = \frac{B_d(z)}{A_d(z)}, \tag{5.2}$$

and the pole-zero map would be formulated in the complex z-plane [41], [42].

In this chapter, the analog transfer function $H_a(s)$ will be generated for the 4th-order test circuit, introduced in the previous chapter. First, the advantages of generating the rational function approximation of (5.1) will be discussed, including the nature of the relationship of the transfer function $H_a(s)$ to the frequency transfer function $H_a(j\Omega)$, also introduced in the last chapter. The concept of a known and widely-used vector fitting method will then be introduced, with a focus on why the resulting transfer function model is most naturally expressed in partial fraction expansion form. The vector fitting process will then be applied to the 4th-order test circuit, and the frequency response performance of the fitted model will be compared to the original frequency transfer function $H_a(j\Omega)$.

Some practical considerations of applying the vector fitting process to high-speed interconnect systems will be discussed, as well as some problematic aspects of converting the partial fraction expansion form to pole-zero factored form. This difficulty is an unfortunate and somewhat unexpected result, and it can hamper attempts to calculate the desired pole-zero map in the s-plane for interconnect systems.

A convenient process borrowed from frequency scaling concepts used in filter theory will be introduced to solve the problem of converting from partial fraction expansion form to pole-zero factored form. This process will then be applied to the transfer function $H_a(s)$ for the 4th-order test circuit, previously calculated using vector fitting. The pole zero map of the 4th-order test circuit will then be plotted in the s-plane for further analysis. Finally, the chapter will conclude with some attempts to lower the order of the model $H_a(s)$ by separating out the principal delay T_d of the system due to the transmission line distributed element, and with a discussion of the time-domain responses calculated with the new s-domain models.

5.2 Motivation for Generating a System Model $H_a(s)$ from $H_a(j\Omega)$

In addition to the computational advantages and the additional insight to be gained from studying the system model in the complex frequency s-plane, another reason for studying the transfer function in this domain is that there is simply more information present in the function [43]. For example, the frequency transfer function $H_a(f) = H_a(j\Omega)$ is simply the function $H_a(s)$ evaluated on the imaginary axis of the complex s-plane, i.e., where $s = j\Omega$, as shown previously in Chapter 3. One problem with this is that the imaginary axis is exactly the dividing line between stable and unstable regions of the s-plane, so operating here is ambiguous from a system stability perspective.

In addition to the system's stability characteristics, it is also known that the system's passivity and causality characteristics are also assessable in the complex frequency domain [63]. These characteristics are important to the accuracy of the model when it is utilized in subsequent time-domain simulations. Figure 5-1 shows a Venn diagram depicting the relationship between the analog system transfer function $H_a(s)$ and the system frequency transfer function $H_a(j\Omega)$, and the fact that the frequency transfer function contains a subset of the information contained in the transfer function [43], [81].

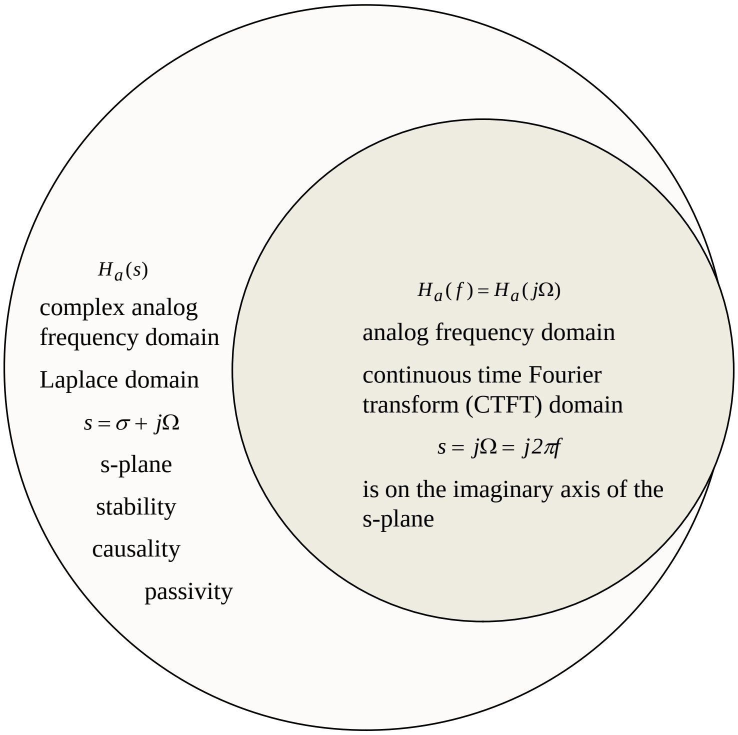


Figure 5-1: Venn Diagram Showing $H_a(j\Omega)$ as a Subset of $H_a(s)$

5.3 Development of System Model from Scattering-Parameter Identification Data

After deriving the analog frequency transfer function $H_a(f) = H_a(j\Omega)$, the analog complex frequency s-domain transfer function $H_a(s)$ can be determined as a rational function approximation using a vector fitting process, such as that developed by Gustavsen [44]. This method uses 1^{st}-order sections of the partial fraction expansion of $H_a(s)$ as basis functions, and works by minimizing the least-squares error (LSE) when performing the fit to the $H_a(j\Omega)$ frequency data. Obtaining a model in the complex frequency domain has several advantages, including the ability to study the model in the complex s-plane using pole-zero analysis. It must be noted, however, that the process of fitting $H_a(j\Omega)$ data to $H_a(s)$ involves adding information artificially back into the function, given that $H_a(j\Omega)$ is a subset of $H_a(s)$, per Figure 5-1. The procedure for determining the pole placement is a key attribute of the overall fitting algorithm.

It should be noted that there are multiple vector fitting algorithms available [26], [45], [82]–[84] which are also least-squares error based. Some of these alternate vector fitting methods utilize time-domain data, some use different basis functions, and some actually provide discrete frequency transfer functions as output. For this research project, however, the method by Gustavsen has been chosen for the vector fitting process for three reasons. First, the method works directly on analog frequency domain system identification data, which is the data type provided naturally in the scattering parameters. Recall that scattering parameter characterization was chosen in Chapter 4 as the system identification method of choice for this research project. Second, the method provides as output the complex analog frequency domain transfer function $H_a(s)$, which is the desired output. Third, the Gustavsen method is the orginal groundbreaking vector fitting method, and as such has the widest acceptance, popularity, and usage.

In addition to studying the characteristics of $H_a(s)$ in the s-plane, it is also possible to convert $H_a(s)$ into the complex digital frequency transfer function $H_d(z)$ using the bilinear transform, matched-Z transform, or impulse invariant transform [41], [42], [48]. This transformation to the z-domain enables the direct use of digital signal processing (DSP) techniques to also study the model in the complex z-plane, again using pole-zero analysis, and enabling the application of yet another well-developed body of analysis theory and methods. The process for converting the analog transfer function $H_a(s)$ to the discrete transfer function $H_d(z)$ for the 4th-order test circuit will be addressed starting in the next chapter.

5.3.1 Development of High-Order Rational Function Approximation for the Transfer Function

For the 4th-order test circuit (with delay) of Figure 5-2 below, the vector fitting method [44] was applied to the frequency transfer function $H_a(j\Omega)$ found previously in Chapter 4, and reproduced in Figure 5-3 for convenience. The vector fitting process [44] was implemented using the function rationalfit.m from the MATLAB RF Toolbox [85]. The function calculates the analog transfer function $H_a(s)$ in the partial fraction expansion form, given by

$$H_a(s) = H_{fit}(s) = \frac{B_a(s)}{A_a(s)} = \left(\sum_{k=1}^{N} \frac{c_k}{s - a_k} + D \right) e^{-sT_d} \,, \tag{5.3}$$

where c_k are the residues, a_k are the poles, D is a direct term, and T_d is the principal delay in the system.

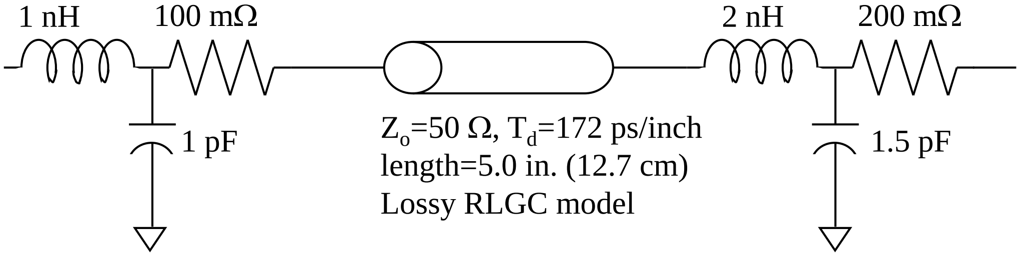


Figure 5-2: 4th-Order Test Circuit with Delay, with Lumped Elements and Transmission Line Distributed Elements Exhibiting Attenuation and Delay

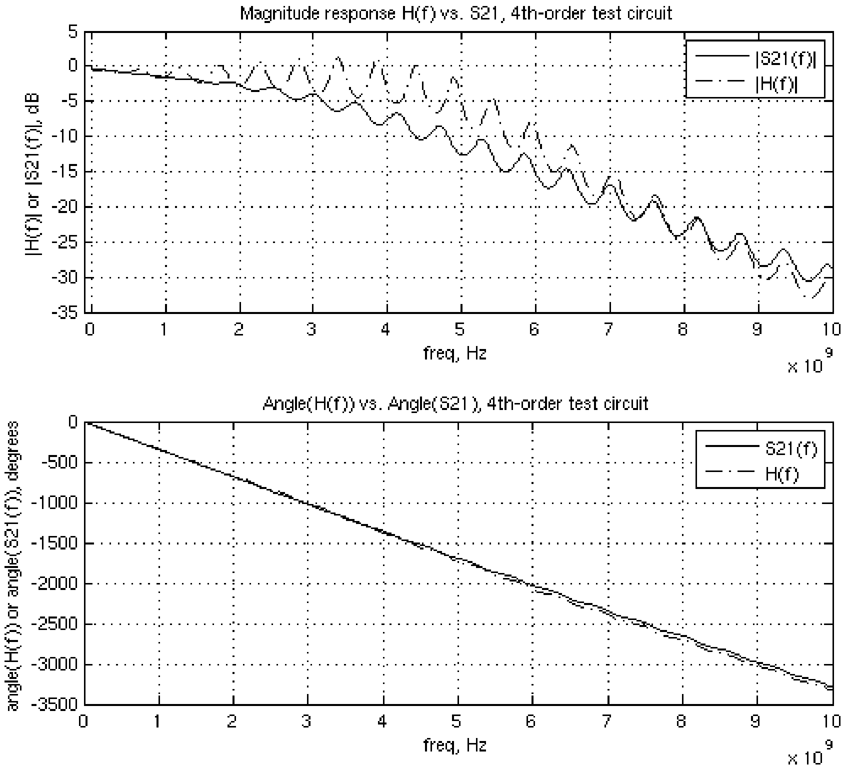


Figure 5-3: Frequency Transfer Function *$H_a(j\Omega)$* for the 4th-Order Test Circuit as Calculated by MATLAB's s2tf.m

The partial fraction expansion form of (5.3) is particularly well suited for computational efficiency and certain forms of macromodeling, namely, for those cases when one wants to use SPICE to implement the transfer function [69]. It is less useful for the generation of pole-zero plots, in that the pole-residue form does not explicitly contain the zero locations.

5.3.1.1 Application of Vector Fitting to Test Circuit with Both Lumped and Distributed Circuit Elements

For the test circuit of Figure 5-2, application of the vector fitting method results in a rational function approximation of order $N = 38$, with the resulting residue and poles:

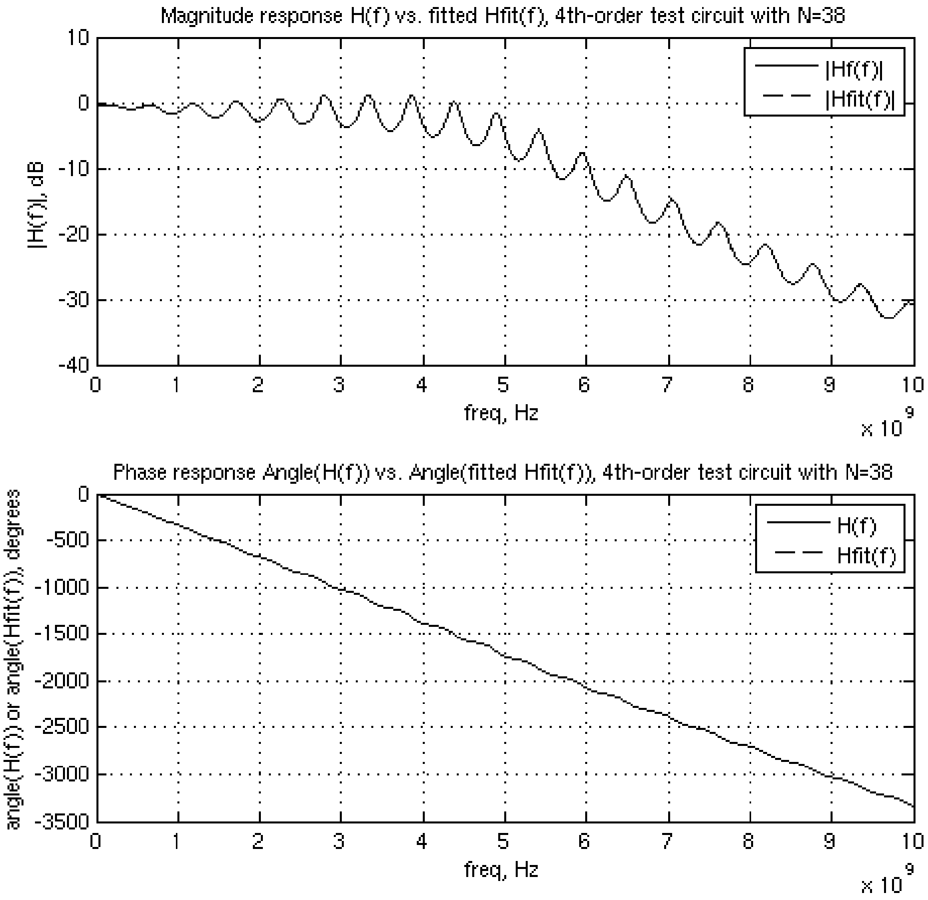
```
Hfit =      Name: 'Rational Function'
            A: [38x1 double]
            C: [38x1 double]
            D: 0.00000000000000e+000
            Delay: 0.00000000000000e+000

Hfit.A (poles):

   -811.301747649996e+006 + 62.6383961389511e+009i
   -811.301747649996e+006 - 62.6383961389511e+009i
   -922.553331967392e+006 + 58.8645791261448e+009i
   -922.553331967392e+006 - 58.8645791261448e+009i
   -840.000334044065e+006 + 55.1859997694428e+009i
   -840.000334044065e+006 - 55.1859997694428e+009i
   -801.893040759086e+006 + 51.5363788366618e+009i
   -801.893040759086e+006 - 51.5363788366618e+009i
   -752.548927442329e+006 + 47.9148916581749e+009i
   -752.548927442329e+006 - 47.9148916581749e+009i
   -707.906808921076e+006 + 44.3411771832699e+009i
   -707.906808921076e+006 - 44.3411771832699e+009i
   -664.354109842137e+006 + 40.8226824020884e+009i
   -664.354109842137e+006 - 40.8226824020884e+009i
   -622.942571713608e+006 + 37.3823622414724e+009i
   -622.942571713608e+006 - 37.3823622414724e+009i
   -588.741481284223e+006 + 34.0308172637667e+009i
   -588.741481284223e+006 - 34.0308172637667e+009i
```

```
   -568.618933856333e+006 + 30.7617627236480e+009i
   -568.618933856333e+006 - 30.7617627236480e+009i
   -572.431728818528e+006 + 27.5132535650082e+009i
   -572.431728818528e+006 - 27.5132535650082e+009i
   -593.267078547925e+006 + 24.2377882060778e+009i
   -593.267078547925e+006 - 24.2377882060778e+009i
   -640.881500974578e+006 + 20.9248376337208e+009i
   -640.881500974578e+006 - 20.9248376337208e+009i
   -718.950383029931e+006 + 17.5832353370390e+009i
   -718.950383029931e+006 - 17.5832353370390e+009i
   -836.675956859968e+006 + 14.2297393689657e+009i
   -836.675956859968e+006 - 14.2297393689657e+009i
   -1.00055187509432e+009 + 10.8626355351864e+009i
   -1.00055187509432e+009 - 10.8626355351864e+009i
   -1.20232183821542e+009 + 7.52716435006931e+009i
   -1.20232183821542e+009 - 7.52716435006931e+009i
   -1.89594172946475e+009 + 1.37169754112862e+009i
   -1.89594172946475e+009 - 1.37169754112862e+009i
   -1.48679092164604e+009 + 4.32243039821584e+009i
   -1.48679092164604e+009 - 4.32243039821584e+009i

Hfit.C (residues):

   -1.73852934794800e+006 - 26.2708411050488e+006i
   -1.73852934794800e+006 + 26.2708411050488e+006i
   -8.24303825217920e+006 + 41.4050159010097e+006i
   -8.24303825217920e+006 - 41.4050159010097e+006i
    10.6496644695613e+006 - 51.4515929111338e+006i
    10.6496644695613e+006 + 51.4515929111338e+006i
   -15.6571524241655e+006 + 70.2649722725269e+006i
   -15.6571524241655e+006 - 70.2649722725269e+006i
    23.8639194084585e+006 - 95.2955868474050e+006i
    23.8639194084585e+006 + 95.2955868474050e+006i
   -36.4588811070515e+006 + 132.627262930310e+006i
   -36.4588811070515e+006 - 132.627262930310e+006i
    58.1053232347736e+006 - 186.249684023465e+006i
    58.1053232347736e+006 + 186.249684023465e+006i
   -92.2082884384281e+006 + 260.644045288611e+006i
```

```
-92.2082884384281e+006 - 260.644045288611e+006i
 145.752355494734e+006 - 354.215536527997e+006i
 145.752355494734e+006 + 354.215536527997e+006i
-216.926351522417e+006 + 453.193718680551e+006i
-216.926351522417e+006 - 453.193718680551e+006i
 303.617987794913e+006 - 540.228969060364e+006i
 303.617987794913e+006 + 540.228969060364e+006i
-388.596953686796e+006 + 597.792392665498e+006i
-388.596953686796e+006 - 597.792392665498e+006i
 474.006576266303e+006 - 640.027033540279e+006i
 474.006576266303e+006 + 640.027033540279e+006i
-562.528834982324e+006 + 683.509659954728e+006i
-562.528834982324e+006 - 683.509659954728e+006i
 664.153806393791e+006 - 750.242170671481e+006i
 664.153806393791e+006 + 750.242170671481e+006i
-779.791856803139e+006 + 880.441163055487e+006i
-779.791856803139e+006 - 880.441163055487e+006i
 909.064697800649e+006 - 1.08586528432533e+009i
 909.064697800649e+006 + 1.08586528432533e+009i
 315.633604159649e+006 - 2.55454508038771e+009i
 315.633604159649e+006 + 2.55454508038771e+009i
-815.147449572848e+006 + 1.58808034659710e+009i
-815.147449572848e+006 - 1.58808034659710e+009i .
```

Figure 5-4 shows a comparison of the original frequency response $H_a(j\Omega)$, determined in Chapter 4, and the frequency response obtained from the fitted function $H_a(s)$ with the order $N = 38$. Note that there are no visually discernible differences in the two frequency responses.

The system order of $N = 38$ is much greater than $N = 4$, which is the expected order of the system, given the presence of the four lumped reactive circuit elements. This difference can be considered and justified from several perspectives. One consideration is that of the delay present in the system due to the transmission line, which inherently exhibits signal propagation delay, and this artificially raises the order of the rational function approximation due to the need to incorporate this delay into the rational function approximation. This importance of the delay characteristic of the system is very apparent in the linear nature of the phase response depicted in Figure 5-4. Alternatively, the delay can be modeled by using a Pade or other approximation to calculate the delay term e^{-sT_d} [86], such as by using an all-pass filter as suggested by Laasko [87], or by similar means [74], [85].


Magnitude response H(f) vs. fitted Hfit(f), 4th-order test circuit with N=38

Phase response Angle(H(f)) vs. Angle(fitted Hfit(f)), 4th-order test circuit with N=38

Figure 5-4: Comparison of $H_a(j\Omega)$ to the Frequency Response of the 4th-Order Test Circuit (with Delay) Rational Function Approximation of Order $N = 38$

The higher order can also be predicted by examining the shape of the magnitude response in Figure 5-4, in which 19 local maxima can be seen in the curve for $H_a(j\Omega)$. These maxima correspond to the transfer function's pole locations. Mirroring the response around the s-plane's horizontal axis into the negative frequency plane would lead to twice that number, or 38, poles. Similarly, it can also be surmised from the magnitude response of $H_a(j\Omega)$ in Figure 5-4 that the rational function approximation would have approximately 38 zeros, corresponding to the 19 local minima seen in the frequency response plot.

5.3.1.2 Application of Vector Fitting to Test Circuit with Transmission Line Element Only

It should be re-emphasized that the high order of the rational function approximation required for the transfer function $H_a(s)$ of the test circuit of Figure 5-2 is primarily driven by the propagation delay present in the system, and not by the presence of the ripple in the magnitude response of Figure 5-4. Even though the pole locations correspond to the relative maxima in the magnitude response, it must be noted that the linear nature of the phase response also drives the requirement for high order, as the vector fitting algorithm fits $H_a(s)$ to both the magnitude and phase responses.

To highlight this, the vector fitting procedure was repeated for another test circuit, which consists of only the lossy transmission line element of the original test circuit of Figure 5-2, minus the lumped element parasitics. For this “tline-only” test circuit, application of the method results in a rational function approximation of order $N = 28$, with the resulting residue and poles:

```
Hfittline = Name: 'Rational Function'
               A: [28x1 double]
               C: [28x1 double]
               D: 0.00000000000000e+000
               Delay: 0.00000000000000e+000

Hfittline.A (poles):

    -75.2845752436545e+009 +                     i
    -4.14236419397871e+009 + 63.1684476212423e+009i
    -4.14236419397871e+009 - 63.1684476212423e+009i
    -5.84301077012365e+009 + 57.5139430689521e+009i
    -5.84301077012365e+009 - 57.5139430689521e+009i
    -6.58297410735073e+009 + 51.4939786928858e+009i
    -6.58297410735073e+009 - 51.4939786928858e+009i
    -6.91469887696315e+009 + 45.2550447752978e+009i
    -6.91469887696315e+009 - 45.2550447752978e+009i
```

```
   -6.99106711608555e+009 + 38.8877998994609e+009i
   -6.99106711608555e+009 - 38.8877998994609e+009i
   -6.85744370420003e+009 + 32.4081736257760e+009i
   -6.85744370420003e+009 - 32.4081736257760e+009i
   -6.76709883329316e+009 + 25.9765726906887e+009i
   -6.76709883329316e+009 - 25.9765726906887e+009i
   -5.74925570085032e+009 + 19.6989186624169e+009i
   -5.74925570085032e+009 - 19.6989186624169e+009i
   -1.63913809064564e+009 + 16.4431642588000e+009i
   -1.63913809064564e+009 - 16.4431642588000e+009i
   -4.34966156292034e+009 + 12.5348700210011e+009i
   -4.34966156292034e+009 - 12.5348700210011e+009i
   -2.17946999811561e+009 + 9.62334723883865e+009i
   -2.17946999811561e+009 - 9.62334723883865e+009i
   -3.10504096615936e+009 + 5.13075077163452e+009i
   -3.10504096615936e+009 - 5.13075077163452e+009i
   -2.69544824929447e+009 + 2.64845896089899e+009i
   -2.69544824929447e+009 - 2.64845896089899e+009i
   -511.126271044357e+006 +                      i

Hfittline.C (residues):

    371.495917369539e+009 +                      i
   -13.3212253037825e+009 + 3.29701731415621e+009i
   -13.3212253037825e+009 - 3.29701731415621e+009i
   -300.350530981887e+006 + 59.3166054984497e+009i
   -300.350530981887e+006 - 59.3166054984497e+009i
    113.070715111657e+009 + 47.4462286597139e+009i
    113.070715111657e+009 - 47.4462286597139e+009i
    144.973260083148e+009 - 105.464086815018e+009i
    144.973260083148e+009 + 105.464086815018e+009i
    5.58326181683103e+009 - 210.504680477202e+009i
    5.58326181683103e+009 + 210.504680477202e+009i
   -148.352096294877e+009 - 147.709935880510e+009i
   -148.352096294877e+009 + 147.709935880510e+009i
   -189.110262351593e+009 + 13.1950525581950e+009i
   -189.110262351593e+009 - 13.1950525581950e+009i
   -59.2685409081308e+009 + 78.3794157106934e+009i
```

```
-59.2685409081308e+009 - 78.3794157106934e+009i
 71.3035569389376e+006 - 196.406674760020e+006i
 71.3035569389376e+006 + 196.406674760020e+006i
-14.7898179409498e+009 + 30.9477744246598e+009i
-14.7898179409498e+009 - 30.9477744246598e+009i
-985.215435394210e+006 - 720.511369711315e+006i
-985.215435394210e+006 + 720.511369711315e+006i
-6.99741396263038e+009 + 7.20779366610419e+009i
-6.99741396263038e+009 - 7.20779366610419e+009i
-6.34209554895659e+009 - 1.63310597824881e+009i
-6.34209554895659e+009 + 1.63310597824881e+009i
 27.7658657285994e+006 +                      i .
```

Figure 5-5 shows a comparison of the original frequency response $H_a(j\Omega)$, determined from the scattering parameters for the “tline-only” test circuit, and the frequency response obtained from the fitted function $H_a(s)$. In this case the fit resulted in order $N = 28$. Note that there are only very minor differences in the two responses.

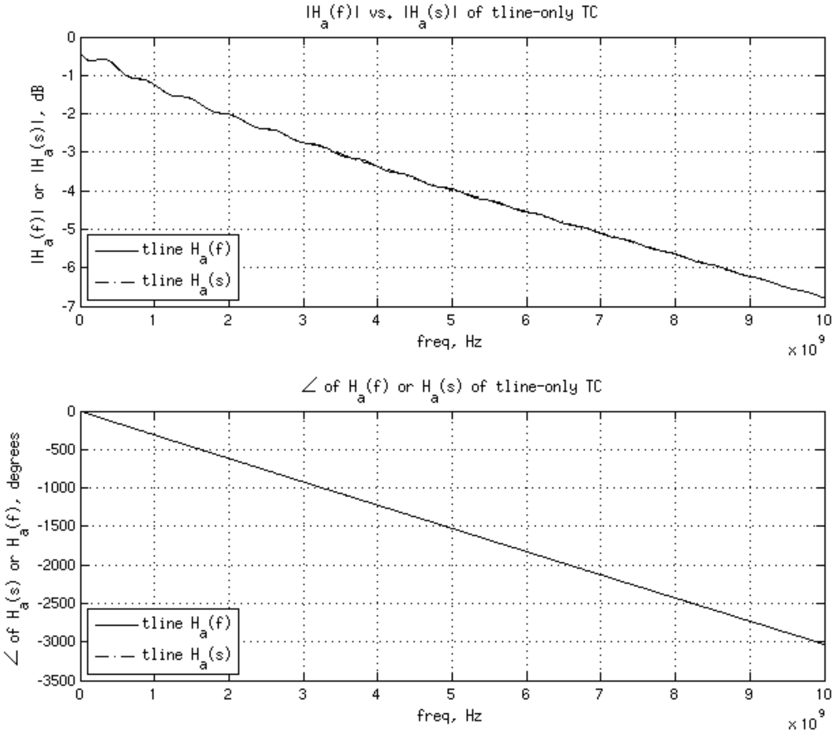


Figure 5-5: Comparison of $H_a(j\Omega)$ to the Frequency Response of the Tline-Only Test Circuit’s Rational Function Approximation of Order $N = 28$

5.3.2 Process for Converting from Partial Fraction Expansion Form to Pole-Zero Factored Form for Transfer Functions with Poles and Residues at High Frequencies

Generally, it is possible to convert from the partial fraction expansion form of $H_a(s)$ in (5.3) to the pole-zero factored form,

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = \frac{K_a(s-b_{a1})(s-b_{a2})\cdots(s-b_{aM})}{(s-a_{a1})(s-a_{a2})\cdots(s-a_{aN})}, \tag{5.4}$$

where the b_{ak} and a_{ak} terms are the zeros and poles of $H_a(s)$, K_a is a gain term, N is the order of the model and the number of poles, and M is the number of finite zeros. This can be done using computer routines such as the residue.m function in MATLAB [88].The fully factored form of $H_a(s)$ shown in (5.4) has two distinct advantages: 1) the poles and zeros are directly evident in the s-plane; and 2) the matched-Z transformation can be directly applied when $H_a(s)$ is expressed in this form [43]. The process for applying the matched-Z and other s- to z-domain transforms will be discussed in detail in the following chapters. Now that the frequency transfer function $H_a(j\Omega)$ has been used to derive the transfer function $H_a(s)$ using vector fitting, it is possible to study the pole-zero map of $H_a(s)$ in the s-plane to gain further insight into the nature of the system.

With some difficulty, the pole-zero factored form of $H_a(s)$ was calculated for the 4th-order test circuit's 38th-order partial fraction expansion. The difficulty arises in the calculation of the zeros in the polynomial $B_a(s)$ when converting from the pole-residue form of (5.3), due to large pole and residue values. Frequency scaling techniques were used to transform the function into a prototype filter in a much lower frequency band prior to calculating $B_a(s)$, using a lowpass to lowpass transformation [59], as follows:

$$H_{proto}(s) = H_a(s)|_{s\to sk_f} = \sum_{k=1}^{N} \frac{c_k}{sk_f - a_k} = \sum_{k=1}^{N} \frac{\frac{c_k}{k_f}}{s - \frac{a_k}{k_f}}. \tag{5.5}$$

Once $H_{proto}(s)$ was obtained in partial fraction expansion form, as shown in (5.5), it was converted to a ratio of two polynomials and reduced to factored form, as follows [59]:

$$H_{proto}(s) = \frac{B_{proto}(s)}{A_{proto}(s)} = K_{ap}\frac{(s-b_{p1})(s-b_{p2})\cdots(s-b_{pM})}{(s-a_{p1})(s-a_{p2})\cdots(s-a_{pN})}, \tag{5.6}$$

where the b_{pk} and a_{pk} terms are the zeros and poles of $H_{proto}(s)$, N is the order of the model, and M is the number of finite zeros. The function is then converted back to the desired frequency range, using frequency scaling as follows [59]:

$$H_a(s) = H_{proto}(s)\Big|_{s\to\frac{s}{k_f}} = \frac{K_{ap}\left(\frac{s}{k_f} - b_{p1}\right)\left(\frac{s}{k_f} - b_{p2}\right)\cdots\left(\frac{s}{k_f} - b_{pM}\right)}{\left(\frac{s}{k_f} - a_{p1}\right)\left(\frac{s}{k_f} - a_{p2}\right)\cdots\left(\frac{s}{k_f} - a_{pN}\right)}. \tag{5.7}$$

Comparing (5.6) and (5.7) it is easily seen that the zeros b_{ak} and poles a_{ak} of $H_a(s)$ are the zeros b_{pk} and poles a_{pk} of $H_{proto}(s)$ scaled by the factor k_f, that is,

$$b_{ak} = b_{pk}k_f\,, \tag{5.8}$$

and

$$a_{ak} = a_{pk}k_f\,. \tag{5.9}$$

The overall process outlined above is summarized in the block diagram of Figure 5-6. Note the use of the frequency scaling processes as indicated in the figure.

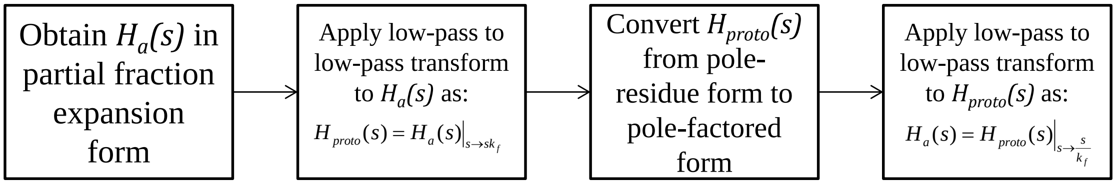


Figure 5-6: Process for Converting *$H_a(s)$* from Pole-Residue to Pole-Zero Factored Form

The results of the vector fitting process are shown in the pole-zero map of Figure 5-7. These results, and specifically the factored form of (5.4) above, will be utilized in Chapter 6 for the application of the matched-Z transform to create a transfer function $H_d(z)$ in the complex digital frequency domain. Note that the function exhibits 38 poles, as indicated in the figure, and 37 zeros. As such, the system can be considered to have one zero at infinity, so that the system is physically realizable [43], [81]. Note also that all of the system poles are located in the left-half plane, so that the system is stable, but that not all the zeros are located in the left-half plane. The implication of this last observation is that the system is not a minimum-phase system, but this should be expected given the phase characteristics exhibited in the frequency response and the fact that the system has significant propagation delay [40], [49].

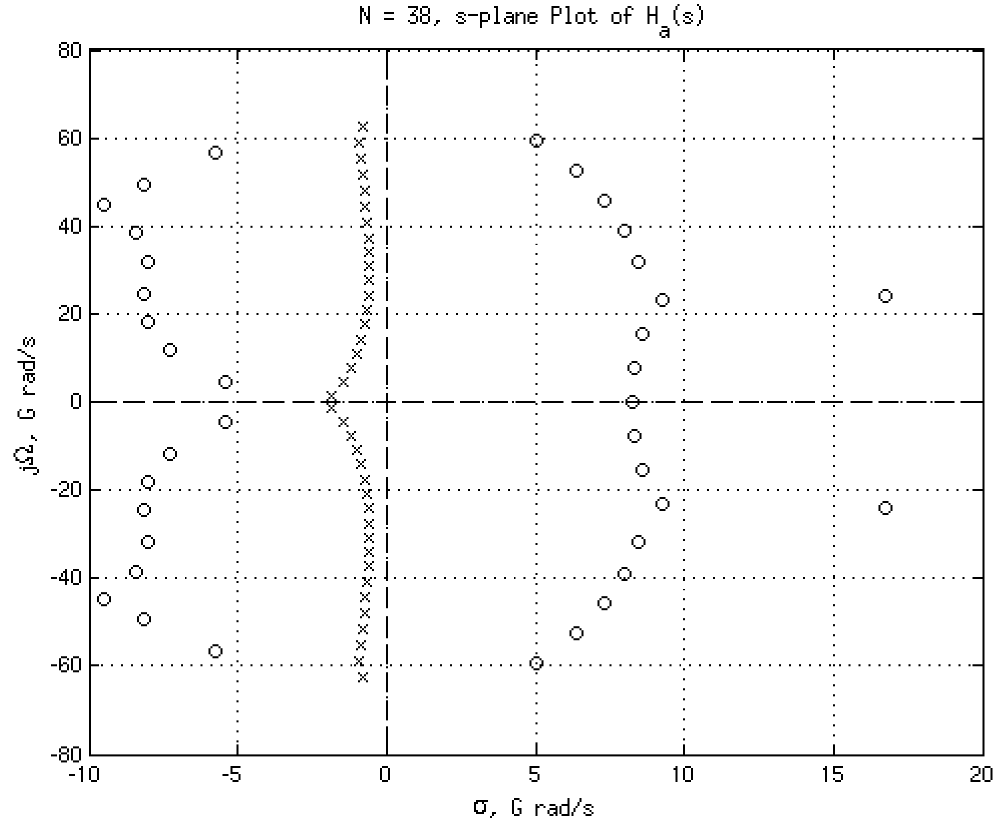


Figure 5-7: Pole-Zero Plot in s-Plane for *$H_a(s)$* of the 4th-Order Test Circuit with Order N=38

Further study of the system in the complex s- and z-planes will be undertaken in later chapters. **It should be noted that the frequency scaling process outlined in Figure 5-6 for converting $H_a(s)$ from partial fraction expansion form to pole-zero factored form has been externally published in the public domain by IBM as a technical disclosure publication** [89]**, as a result of this research project.**

5.3.3 Development of Reduced-Order Rational Function Approximation Model with Separated Delay

A tradeoff can be made between the model's order and the model's accuracy. This accuracy can be quantified in either the frequency domain response or in the time domain response. For the 4^{th}-order test circuit of Figure 5-2, the frequency response $H_a(j\Omega)$ shown in Figure 5-3 is compared in Figure 5-8 to that of a rational function approximation where N is forced to be 4, and a principal constant delay T_d is specified as defined in (5.3). Application of the vector fitting method in this fashion results in a rational function approximation with the following values:

```
Hfit4th =   Name: 'Rational Function'
            A: [4x1 double]
            C: [4x1 double]
            D: 0.00000000000000e+000
            Delay: 838.713532249092e-012

Hfit4th.A(poles):

      -6.79632293858445e+009 + 29.0023263230531e+009i
      -6.79632293858445e+009 - 29.0023263230531e+009i
      -13.2081587156437e+009 + 10.7603223014198e+009i
      -13.2081587156437e+009 - 10.7603223014198e+009i

Hfit4th.C(residues):

      -4.00787915096367e+009 + 3.52211244818399e+009i
      -4.00787915096367e+009 - 3.52211244818399e+009i
       4.08025510536357e+009 - 10.9990579501479e+009i
       4.08025510536357e+009 + 10.9990579501479e+009i .
```

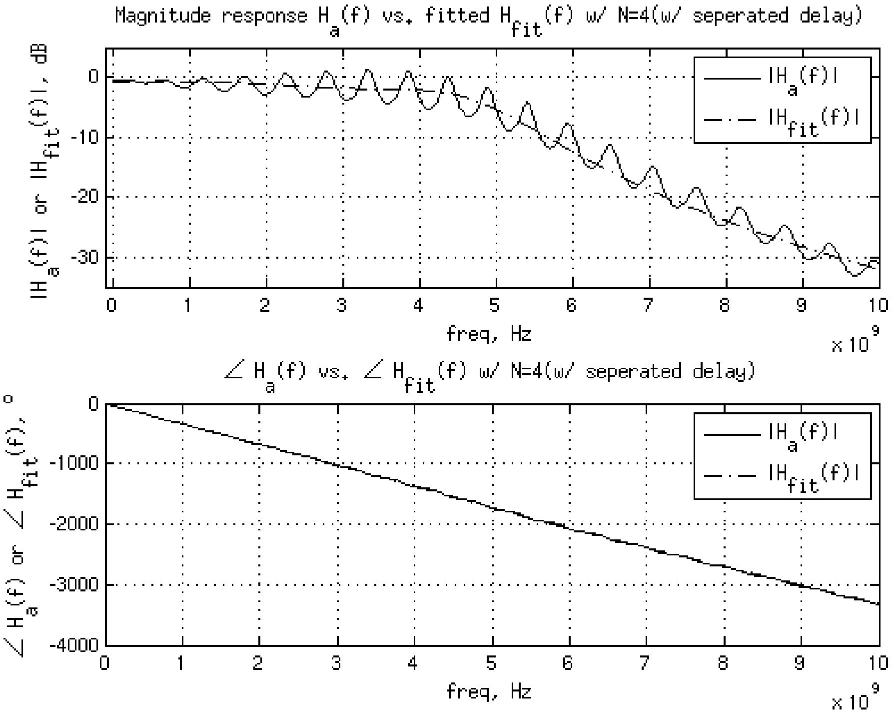


Figure 5-8: Comparison of $H_a(j\Omega)$ to a 4th-Order Rational Function Approximation of Order $N = 4$

It is seen in Figure 5-8 that the magnitude response of the reduced-order model generally follows that of $H_a(j\Omega)$, but that it certainly does not capture all the local maxima and minima, unlike the case for $N = 38$ depicted in Figure 5-4. Note also that the phase response is modeled very accurately by the reduced-order model.

5.4 Assessment of Model Generation Results

A complete assessment and comparison of the s-domain system transfer function, generated by vector fitting and expressed in the pole-residue form of (5.3), requires a close examination of both the frequency and time-domain responses, in addition to their assessment in the s-domain using the pole-zero maps of Figure 5-7. A comparison of the frequency responses for the transfer functions $H_a(s)$ of orders $N = 38$ and $N = 4$, respectively, with that of the original frequency transfer function $H_a(j\Omega)$ was performed in the previous section.

Figure 5-9 shows this comparison of the transient response to a single 1-ns pulse stimulus, for the cases of using HSPICE to generate the response directly from the circuit description in the native SPICE language, and of using the model $H_a(s)$ generated by the Gustavsen vector fitting method set to optimize for low fitting error, resulting in the model of $N = 38$ described previously. In the latter case, the time-domain response is calculated using MATLAB’s timeresp.m function, which assumes as input $H_a(s)$ in the form of (5.3), and generates a continuous time state-space formulation [61]. The continuous-time state-space formulation is then transformed into a discrete-time state-space formulation using known methods for numerical solution by computer [48], [49]. Note that the results display almost perfect correlation from the standpoint of a visual examination of the waveforms. The remarkable correlation suggests that the timeresp.m function’s underlying algorithms may closely mimic those used in HSPICE.

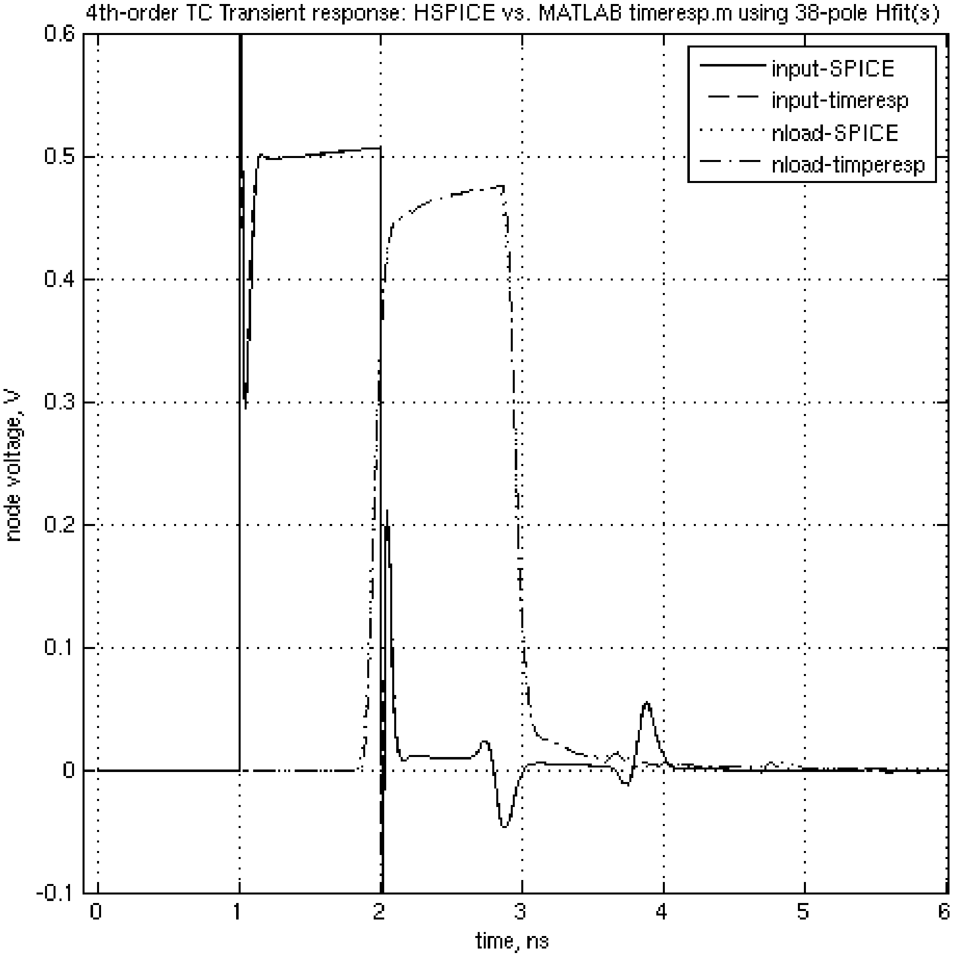


Figure 5-9: Transient Response Comparison of Models Derived Using HSPICE and N =38

Figure 5-10 shows a comparison of the transient response to a single 1-ns pulse stimulus, for the cases of using HSPICE to generate the response directly from the circuit description in the native SPICE language, and of using the model $H_a(s)$ generated by vector fitting with the order restricted to $N = 4$, again using MATLAB's timeresp.m function to calculate the response. Note that the results display good accuracy in the signal's propagation delay, but less accuracy in the waveform's amplitude and peak prediction.

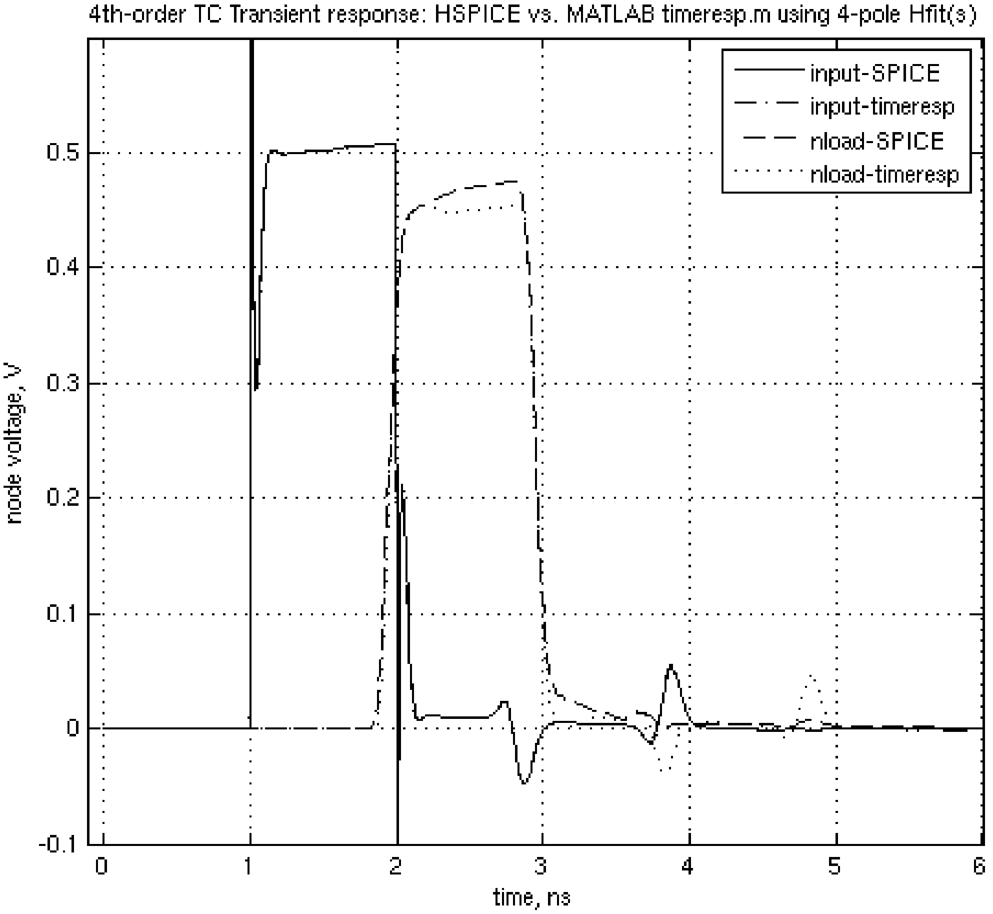


Figure 5-10: Transient Response Comparison of Models Derived Using HSPICE and $N = 4$ with Extracted Principal Delay

It should be noted that the $H_a(s)$ generated forcing $N = 4$ and included the separated delay term, as represented in (5.3), resembles closely the mathematical form of the function identified in Chapter 4 using the ARX method. In that case, the delay term was represented by the z^{-n_o} term, but the system was also 4th-order. Clearly, the vector fitting method used here, utilizing analog frequency domain data in lieu of the time-domain data used for the ARX method, performs well in comparison, and has the advantages of working natively with the scattering parameter data's natural format.

5.5 Conclusions Regarding $H_a(s)$ Model Attributes

The process described in this section for generating a transfer function $H_a(s)$ in the Laplace domain from measured or simulated scattering-parameter data is summarized in the block diagram in Figure 5-11. Note that this process is inherently dependent upon the use of scattering parameter characterization as a system identification technique, upon the relationship between the scattering parameter data and the system's frequency response $H_a(j\Omega)$, and upon the use of a vector fitting method to construct the system's Laplace-domain transfer function $H_a(s)$ in the form of a rational function approximation. Furthermore, if s-plane analysis of the transfer function is desired, then the additional step of converting the function from pole-residue form to pole-zero factored form is required, and the application of frequency scaling techniques to execute this conversion may be required for signal integrity interconnect systems.

There also can be variations on the process by constraining the order of the rational function approximation and including a separable delay term, specified in (5.3), with frequency response results shown in Figure 5-8 and time-domain response results shown in Figure 5-10. The advantage of doing this is model simplification, with the resulting fitted transfer function model potentially being of much lower order, at the expense of some accuracy in reproducing the original frequency response $H_a(j\Omega)$ and time-domain response.

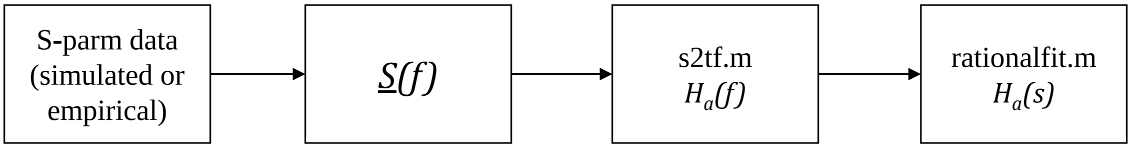


Figure 5-11: Summary for Process for Generating $H_a(s)$ from $S(f)$

For the purposes of comparison, the next chapter will focus on studying the system characteristics in the z-domain, as opposed to the s-domain. To do this, it will be necessary to develop the transfer function $H_d(z)$ in the discrete z-domain, as opposed to the transfer function $H_a(s)$ in the analog s-domain. This will be done by invoking some well-known, widely-adopted s- to z- domain transformations.

Chapter 6: Development of Transfer Function Model $H_d(z)$ from $H_a(s)$

6.1 Introduction

As seen in the previous chapter, examination of the system model in the complex frequency domain, either analog or digital, can be convenient. There are computational advantages to expressing the discrete transfer function $H_d(z)$ as a rational function approximation, namely [44],

$$H_d(z) = \frac{B_d(z)}{A_d(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \ , \qquad M \leq N. \tag{6.1}$$

In addition, insight can be gained by examining the model $H_d(z)$ in the complex z-domain plane by studying the relative positions of the poles and zeros, and assessing their effects on the system's frequency response. For both of these reasons, it can be advantageous to formulate the transfer function $H_d(z)$ in the complex digital frequency domain.

In this chapter, the discrete transfer function $H_d(z)$ will be generated for the 4th-order test circuit. Since scattering parameter characterization has been selected as the primary system identification method, the focus will be on generating $H_d(z)$ from the analog transfer function $H_a(s)$, using several s- to z-domain transformations, even though it should be noted that there are alternative system identification methods that allow for the direct realization of $H_d(z)$. One set of alternative methods uses time-domain identification techniques followed by either parametric methods [67] or non-parametric methods to find the discrete frequency transfer function $H_d(e^{j\omega})$ [90]. Another set of methods uses various forms of vector fitting in the discrete-time or discrete-frequency domain [26], [82]. The approach chosen here to generate $H_d(z)$ from $H_a(s)$ using scattering parameters is based on three practical considerations. First, recall that $H_a(j\Omega)$ is a natural output of the scattering parameter identification procedure once the source and load characteristics of the system are added [58]. The system identification data required for the alternative methods are neither as convenient nor as prevalent. Second, the wide prevalence and acceptance of the original analog frequency-domain vector fitting algorithm [44] gives great confidence in its robustness and viability, and this algorithm works directly on $H_a(j\Omega)$ when finding the transfer function $H_a(s)$. Third, and perhaps most importantly, it is desirable to obtain and retain the analog transfer function $H_a(s)$ to enable examination in the s-plane, and to provide a direct comparison with its digital counterpart $H_d(z)$ in the z-plane.

In this chapter, some practical considerations for generating the discrete transfer function $H_d(z)$ will be discussed for the three most widely used s- to z-domain transformation methods. These methods are the bilinear transform (BLT), the matched-Z transform (MZT), and the impulse invariant transform (IIT). The process for deriving $H_d(z)$ from $H_a(s)$ using each method will be presented, as well as some problematic aspects of generating the function in each case.

6.2 Motivation for Generating the Discrete Transfer Function $H_d(z)$ from the Analog Transfer Function $H_a(s)$

In addition to the computational advantages to be gained from studying the transfer function in the discrete complex frequency z-plane, another fundamental reason for working in this domain is that there is simply more information present in the z-domain than in the discrete frequency ω-domain [43]. For example, the discrete frequency transfer function $H_d(e^{j\omega})$ is simply the discrete transfer function $H_d(z)$ evaluated on the unit circle of the complex z-plane, i.e., where $z = e^{j\omega}$, as explained previously in Chapter 3. One problem with working with the frequency transfer function on the unit circle is that the unit circle is exactly the boundary between stable and unstable regions of the z-plane, so operating here is ambiguous from a transfer function stability perspective. In addition to the system's stability characteristics, it is also known that the system's causality and passivity characteristics are also reflected best in the complex frequency domain [63].

Figure 6-1 shows a Venn diagram depicting the general relationship between the system transfer function $H_d(z)$, the system frequency transfer function $H_d(e^{j\omega})$, and the system sampled frequency transfer function $H_d(k)$. As the figure shows, system frequency transfer functions contain subsets of the information contained in system transfer function. Note the similarity between the diagram for the various digital frequency domains and the diagram of Chapter 5 for the various analog frequency domains. Note also, however, that there is now an additional domain, namely, the sampled digital frequency domain, for which there is no corresponding domain in the analog case.

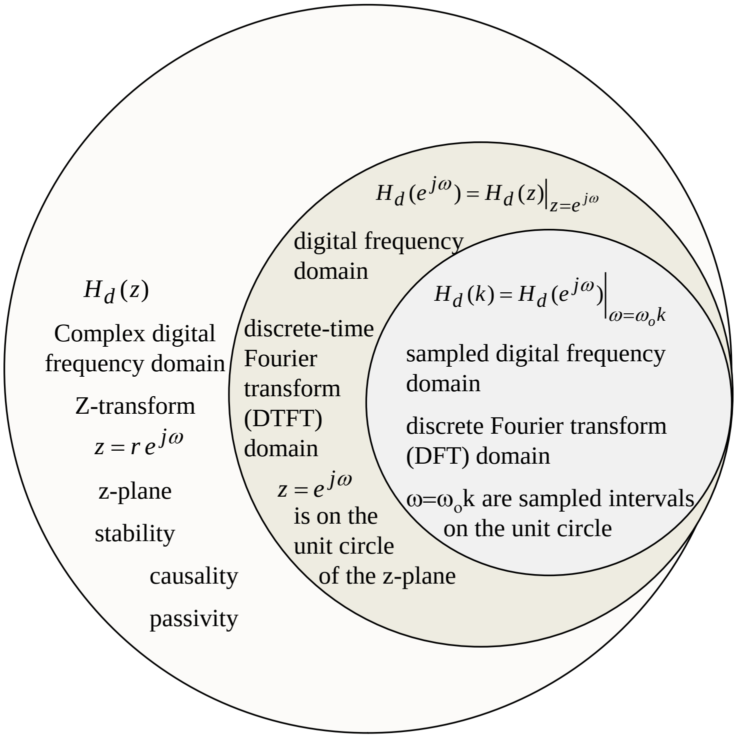


Figure 6-1: Venn Diagram Showing $H_d(k)$ as a Subset of $H_d(e^{j\omega})$ as a Subset of $H_d(z)$

It is clear that the digital complex z-domain and the analog complex s-domain are analogous in nature, in that they both transform from their respective discrete-time and continuous-time domains, and that they both contain stability and causality information. Both the z-domain and the s-domain thus represent alternate but complete descriptions of the time domain, albeit from a different viewpoint. In contrast, the corresponding discrete-time Fourier transform and continuous-time Fourier transform domains do not represent such a complete picture, with the most obvious deficiency being the absence of system stability information [43], [81], [49], [42].

Regardless of their similarities, it should be noted there are also a few significant differences between the s-domain and z-domain that have some practical implications regarding their application. One practical difficulty with using the s-domain at high frequencies has to do with the scales of the real and imaginary axes. In particular, the imaginary component of the complex frequency values can get very large at high frequencies, making it impractical to plot the poles and zeros on a linear scale in such a way that the origin, imaginary axis, and real axis of the complex plane can be easily observed simultaneously on the same plot. Since the imaginary axis of the s-plane represents the stability boundary, not being able to see it adequately in a visual pole-zero plot is a significant limitation.

In contrast, in the z-domain, the frequency variation simply rotates circularly around the origin, as described in Chapter 3. The s-domain is exponentially warped into the z-domain via the relationship $z = e^{sT_s}$. While this may create aliasing problems and ambiguities above the Nyquist frequency, if the sampling rates are carefully and appropriately selected these limitations are avoided, and the visual plotting is much easier to interpret because the axis scaling is more reasonable. In addition, the stability boundary represented by the unit circle is always easily observable in the plot. These advantages, and the fact that there exists a multitude of tools and techniques from the field of digital signal processing (DSP) for the analysis of transfer functions expressed in the z-domain, leads to a desire to evaluate the transfer function $H_d(z)$ in the z-domain for signal integrity applications. This is true even if the system model was originally identified in the analog frequency domain, as is the case when using scattering parameter characterization.

There are several known methods from the field of DSP theory which can be applied to the problem of transforming between the s- and z-domains [42], [41]. In this chapter, three of these methods, namely, the bilinear transform, the matched-Z transform, and the impulse invariant transform, will be applied to the analog transfer function $H_a(s)$ for the 4th-order test circuit. The resulting discrete transfer function $H_d(z)$ will then be available to further analyze the characteristics of this system in the z-domain.

6.3 Development of Discrete Transfer Function $H_d(z)$ Using the Bilinear Transform

One of the most popular and well-known methods for transforming from the s-domain to the z-domain is the bilinear transform (BLT), discussed previously in Chapter

3. It simplifies the exact mapping from the s-plane to the z-plane of $z = e^{sT_s}$ by using a 1st-order Pade approximation,

$$z = e^{sT_s} \approx \frac{1 + \frac{T_s}{2} s}{1 - \frac{T_s}{2} s} \ . \tag{6.2}$$

Solving (6.2) for s results in the bilinear transformation, also found previously in Chapter 3, as [42]:

$$s = \frac{2}{T_s}\left(\frac{z - 1}{z + 1}\right). \tag{6.3}$$

Recall from Chapter 5 that the analog transfer function for the 4th-order test circuit was represented by a rational function approximation with order $N = 38$, and that this function was expressed in terms of a partial fraction expansion of the form:

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = \left(\sum_{k=1}^{N} \frac{c_k}{s - a_k} + D\right) e^{-sT_d}\ , \tag{6.4}$$

For the case where $N = 38$, nearly a perfect fit for the magnitude and phase responses of $H_a(s)$ was obtained using the vector fitting process, with a resulting order of $N = 38$ and no separated principal delay term, i.e., $T_d \ = \ 0$ in (6.4) above. It was also shown that a reasonable approximation to $H_a(s)$ could be obtained in the form of (6.4) above with $N \ = \ 4$ and $T_d \ = \ 839$ ps, if the order were constrained artificially to $N = 4$.

Applying the bilinear transform to the general partial fraction expansion of (6.4) for the case when $T_d \ = \ 0$, results in the following general equation for $H_d(z)$:

$$H_d(z) = H_a(s)\Big|_{s=\frac{2}{T_s}\left(\frac{z-1}{z+1}\right)} = \left(\sum_{k=1}^{N} \frac{\frac{T_s c_k}{2 - a_k T_s}(z + 1)}{z - \frac{2 + a_k T_s}{2 - a_k T_s}} + D\right). \tag{6.5}$$

The derivation of (6.5) is included in Addendum 6A: to this chapter. Note that (6.5) gives an expression for $H_d(z)$ in terms of the poles a_k and residues c_k of the partial fraction expansion form of $H_a(s)$ of (6.4) above, assuming there is no direct term D or delay term e^{-sT_d}. This is advantageous when converting a transfer function in the s-domain that is presented in the pole-residue form.

Applying the relation (6.5) to the 4th-order test circuit with $N = 38$ and $T_s = 40$ ps results in the pole-zero map in the z-plane shown in Figure 6-2. Note that the poles are all within the unit circle, suggesting that the system is stable, while there are zeros both inside and outside the unit circle, suggesting that the system is not a minimum phase system [41]. It should also be explicitly noted that the pole and zero locations depicted in Figure 6-2 were obtained without first performing pre-warping at specified frequencies. The pre-warping process is usually undertaken when utilizing the bilinear transform, to compensate for the inaccuracies and practical difficulties encountered with the utilization of the Pade approximation for the exponential map, as expressed in (6.2) [41]–[43].

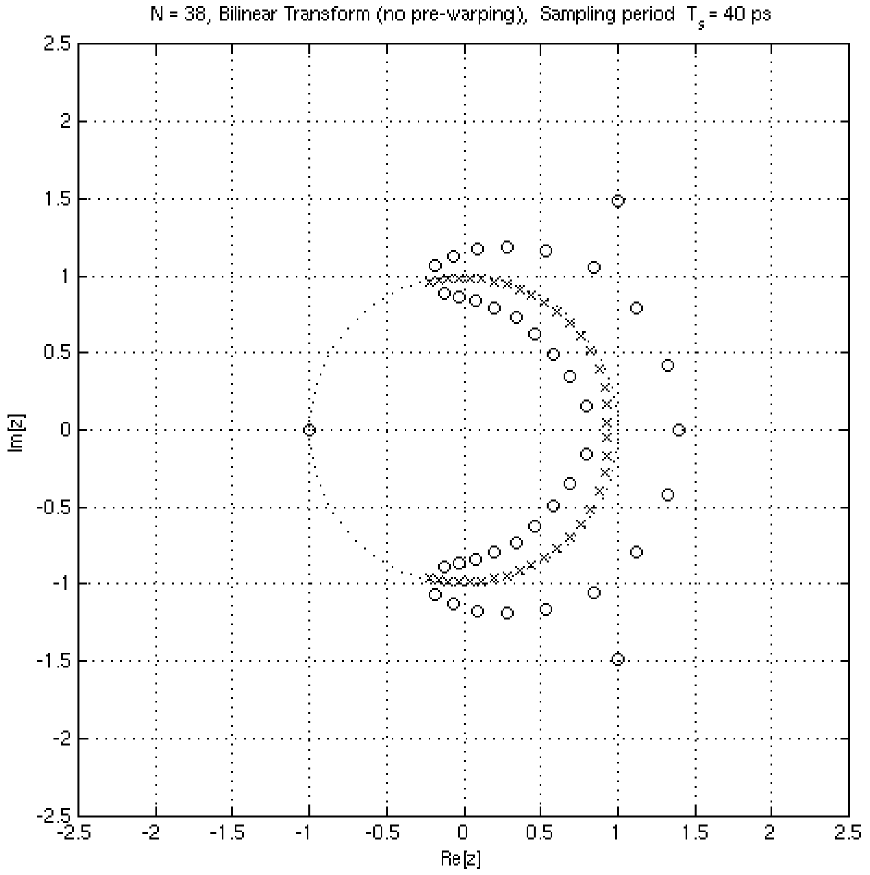


Figure 6-2: Pole-Zero Map in z-Plane for the 4th-Order Test Circuit for $N = 38$ and $T_s = 40$ ps using the Bilinear Transform with No Pre-Warping

6.4 Development of Discrete Transfer Function *Hd(z)* Using the Matched-Z Transform

A second well-known method for transforming from the s-domain to the z-domain is the matched-Z transform (MZT), discussed previously in Chapter 3 [43], [48], [49]. It utilizes the exact map $z = e^{sT_s}$ for all poles, and for all zeros which have imaginary components at less than the Nyquist frequency. Zeros with imaginary components at greater than the Nyquist frequency are explicitly placed at $z = -1$. As described in Chapter 3, application of the matched-Z transform requires that the expression for the analog transfer function $H_a(s)$ be expressed in pole-zero factored form, as follows:

$$H_a(s) = K_a \left[\frac{\prod_{k=1}^{M} (s - b_k)}{\prod_{k=1}^{N} (s - a_k)} \right] ;\ N \geq M \,. \tag{6.6}$$

In this case, the resulting discrete transfer function $H_d(z)$ assumes the form:

$$H_d(z) = K_d \left[\frac{\prod_{k=1}^{M} (z - e^{b_k T_s})}{\prod_{k=1}^{N} (z - e^{a_k T_s})} \right] ;\ N \geq M \,, \tag{6.7}$$

which is also seen to be in pole-zero factored form in the z-domain.

One very significant problem with the application of the matched-Z transformation to functions $H_a(s)$ that have been obtained by vector fitting is that those functions are not obtained in the pole-zero factored form of (6.6), but rather in the partial fraction expansion form of (6.4), which is expressed in terms of poles and residues. The process for converting the function $H_a(s)$ can be problematic for functions of high order with poles and zeros located at high frequencies, due to the limitations in numerical precision for computations involving large numbers [60]. This problem was encountered for the 4th-order test circuit, and ultimately solved using frequency scaling network analysis techniques as described in Chapter 5 and [89], which allowed the large numbers to be avoided in the calculation.

Once $H_a(s)$ for the 4th-order test circuit was obtained in factored form of (6.6), the matched-Z transformation was applied and $H_d(z)$ was obtained in the factored form (6.7). From the factored form of $H_d(z)$ it is straightforward to construct the pole-zero map in the z-plane, as shown in Figure 6-3. Note that the pole and zero locations are different from the bilinear transform case of Figure 6-2 for those locations at higher frequencies of ω. This is due to the fact that no pre-warping of critical frequencies has been undertaken for the bilinear transform case. The advantage of the matched-Z transform over the bilinear transform, at least as applied in Figure 6-2, lies in the simplicity of pole and zero mapping that is so clearly evident in Figure 6-3.

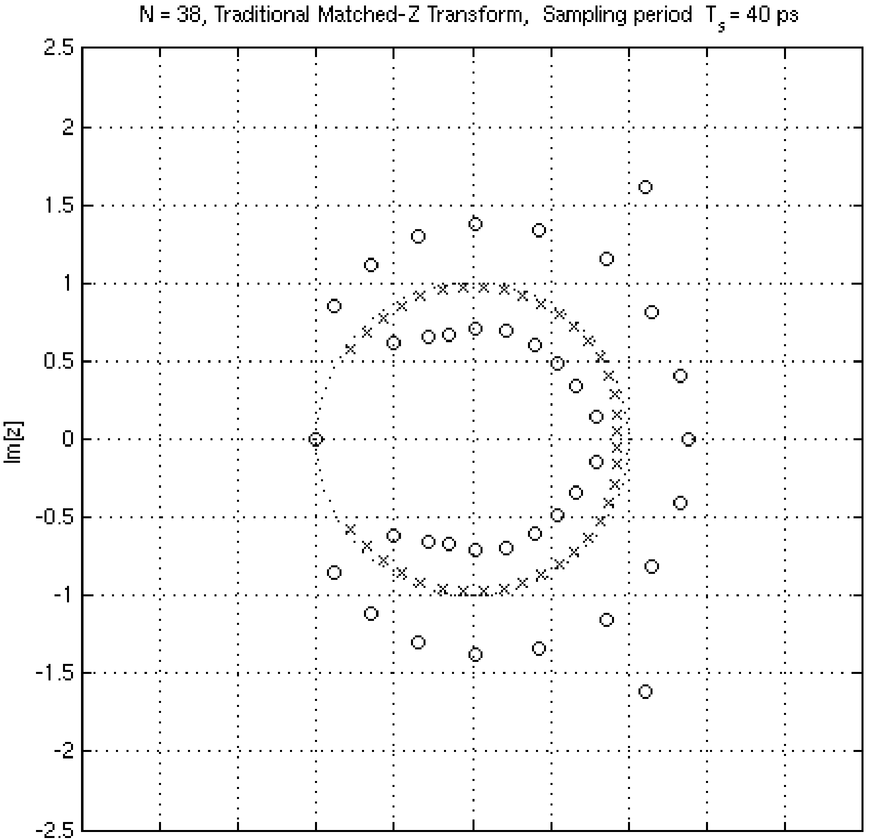


Figure 6-3: Pole-Zero Map in z-Plane for the 4th- Order Test Circuit for $N = 38$ and $T_s = 40$ ps Using the Matched-Z Transform

6.5 Development of Discrete Transfer Function $H_d(z)$ Using Bilinear Transform Variations with Frequency Pre-Warping

The results for the matched-Z transform show that it is possible to get a reasonable discrete frequency transfer function $H_d(z)$ using an s-domain to z-domain transformation. The pole-zero map of Figure 6-3 for the matched-Z transform leads to the conclusion that the pole-zero map of Figure 6-2 obtained using the bilinear transform is unacceptable. In reality, this should not be surprising, given that the BLT is known to warp the frequency variable at higher frequencies [41]–[43].

Because of the extreme popularity of the bilinear transform in digital signal processing applications, it would still be desirable to find a way to apply it for signal integrity applications. Thus, in this section two additional attempts will be made to improve upon the BLT results, by re-applying the BLT using two variations of pre-warping the frequency prior to transforming to the z-domain. The first method, widely used in the design of digital filters, will apply pre-warping at a single critical frequency of interest. This technique, which amounts to a form of frequency scaling, has the effect of matching the magnitude and phase responses of $H_a(s)$ and $H_d(z)$ at that particular frequency [41]. The second method involves pre-warping the entire discrete frequency vector, ω, determining a warped version of the frequency transfer function $H_a(j\Omega)$, fitting an s-domain rational function to the warped frequency transfer function, and then applying the BLT to the warped frequency transfer function to obtain $H_a(s)$ [91].

6.5.1 Development of Discrete Transfer Function $H_d(z)$ Using the Bilinear Transform with Pre-Warping of Critical Frequency

In order to confirm that the discrepancy in the pole-zero mapping between the bilinear transform and matched-Z transform was due to frequency warping, and to improve the overall pole-zero map obtained using the bilinear transform, a second attempt was made to calculate the pole and zero positions using the BLT. For this second attempt, pre-warping of the critical frequency 10 GHz for the low-pass analog filter was applied before invoking the transform of (6.5), with the resulting pole-zero map shown in Figure 6-4. Addendum 6B: contains a derivation for a general pre-warping technique at an arbitrary frequency that shows that pre-warping can be considered as another application and implementation of the concept of frequency scaling [42], [59], [92].

Note that the poles now cover approximately the same overall frequency range around the circle, but that the pole spacing still does not match the pole-zero diagram for the matched-Z transform of Figure 6-3. It can also be seen that the zero positions are significantly off. From this, it can be seen that the BLT's inherent method of matching the frequency response $H_d(e^{j\omega})$ at $\omega = 0$ [92] may be the better approach than trying to match at high frequencies for this high-order system. It could also be suggested that, for a frequency response with a significant number of poles and zeros with critical positions over a broad frequency range, that the bilinear transform's characteristics may not be adequate for this application, even with this additional step of pre-warping the critical frequency value, because frequency distortion cannot be avoided over the entire range.

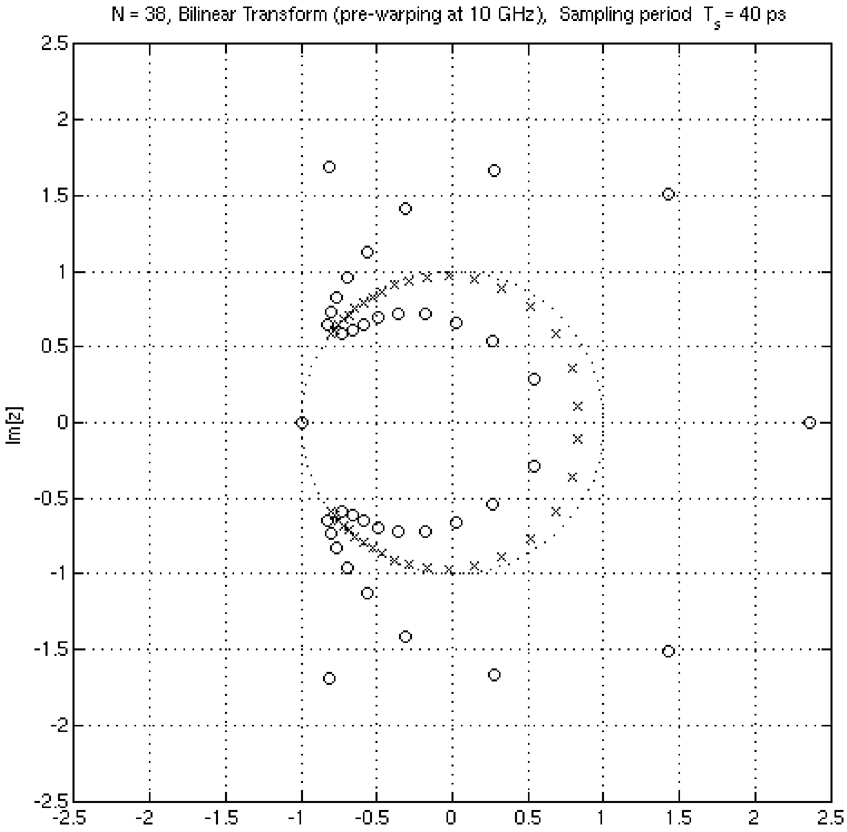


Figure 6-4: Pole-Zero Map in z-Plane for the 4th-Order Test Circuit for $N = 38$ and $Ts = 40\ ps$ Using the Bilinear Transform with Pre-Warping Applied at $10\ GHz$

6.5.2 Development of Discrete Transfer Function *$H_d(z)$* Using the Bilinear Transform with Pre-Warping of Entire Discrete Frequency Vector

In an effort to further improve the overall results of the pole-zero map obtained using the bilinear transformation, a third attempt was made to calculate the pole and zero positions using the BLT. For this third attempt, pre-warping of the entire discrete frequency variable ω was applied, per the process by formulated by Ikai, and patented by Fujitsu [91]. The discrete frequency ω is first pre-warped as follows:

$$\omega_c = 2\tan\frac{\omega}{2}\ , \tag{6.8}$$

where ω_c is the pre-warped discrete frequency. The variable ω_c, is then re-interpolated over its new range so that it has a uniform frequency interval, and re-expressed as ω_c', defined as the re-interpolated version of ω_c. The original frequency transfer function $H_a(f)$ is now re-plotted vs. ω_c', becoming a new frequency transfer function $H_a'(\frac{\omega_c'}{2\pi T_s})$. The vector fitting process [44], [85] is now applied to $H_a'(\frac{\omega_c'}{2\pi T_s})$, resulting in rational function approximation $H_a'(s')$. It is this new function to which the bilinear transform is applied, resulting in:

$$H_d(z) = H_a'(s')|_{s'=\frac{2}{T_s}\left(\frac{z-1}{z+1}\right)}\ . \tag{6.9}$$

The resulting pole-zero map obtained from this process is shown in Figure 6-5. Note the great improvement in the spacing of the poles and zeros, and the favorable comparison of the pole locations to those predicted by the matched-Z transform in Figure 6-3.

It is also apparent when comparing the z-plane pole-zero maps of Figure 6-3 and Figure 6-5 that, while the pole locations are now very close, there is still some significant variation in the zero locations as compared to the results given by the matched-Z transform. In particular, two outlier zeros are present in the right portion of the z-plane. The effect of this variation in the locations of the zeros on the overall performance of the interconnect system will be considered in more detail in subsequent sections.

Figure 6-6 depicts a block diagram summarizing the process for calculating $H_d(z)$ using the bilinear transform with pre-warping of the entire frequency variable ω, as just described.

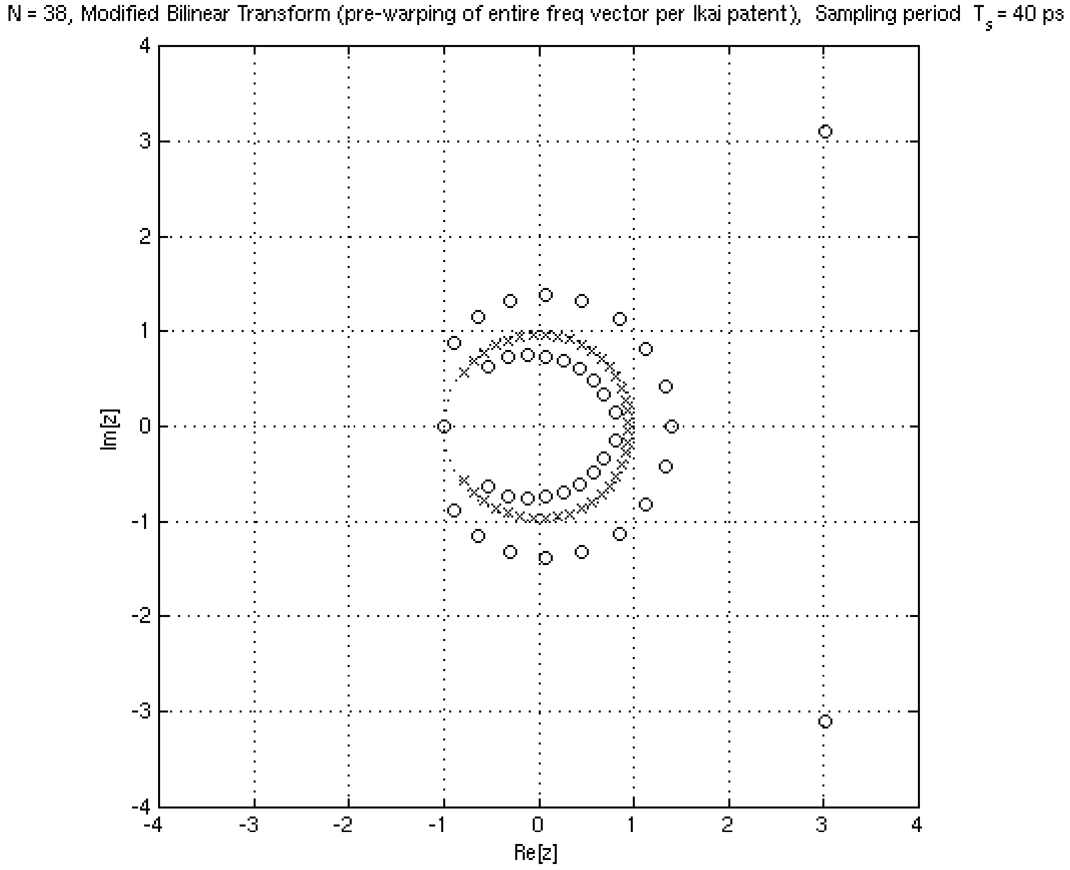


Figure 6-5: Pole-Zero Map in the z-Plane for the 4th-Order Test Circuit for *N=38* and T_s=*40 ps* Using the Bilinear Transform with Pre-Warping Applied to the Entire Frequency Vector ω

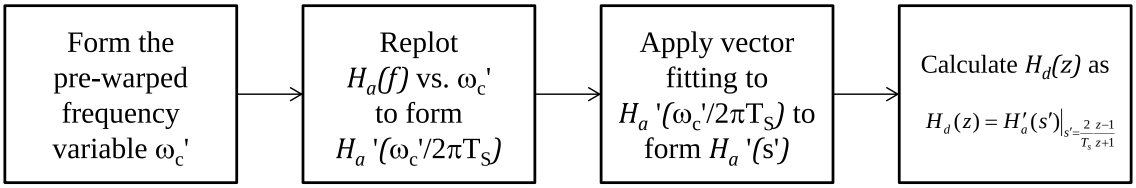


Figure 6-6: Summary of Process for Calculating *$H_d(z)$* Using the Bilinear Transform with Pre-Warping of the Entire Frequency Vector ω

6.6 Development of Discrete Transfer Function *$H_d(z)$* Using Impulse Invariant Transform

In addition to the bilinear transform and the matched-Z transform, a third common method for transforming from the s-domain to the z-domain is the impulse invariant transform (IIT), also discussed previously in Chapter 3 [41]–[43]. It utilizes the exact $z = e^{sT_s}$ map for all poles, but the zeros are not mapped in a such a straightforward manner [81]. Specifically, for an s-domain transfer function $H_a(s)$ expressed in the pole-residue form of (6.4), and with $T_d = 0$, utilizing the impulse invariant transform leads to the z-domain transfer function $H_d(z)$ expressed as:

$$H_d(z) = \sum_{k=1}^{N} \frac{T_s c_k z}{z - e^{a_k T_s}} \;, \tag{6.10}$$

where c_k are the residues and a_k are the poles of the original analog function $H_a(s)$. Note that, like the case for the bilinear transform, the impulse invariant transform lends itself naturally to application to analog transfer functions $H_a(s)$ which are expressed in pole-residue form, which was not the case for the matched-Z method.

Applying the impulse invariant transform as indicated in the relation (6.10) to $H_a(s)$ for the 4th-order test circuit with $N = 38$ and $T_s = 40\,ps$ results in the pole-zero map in the z-plane, as shown in Figure 6-7.

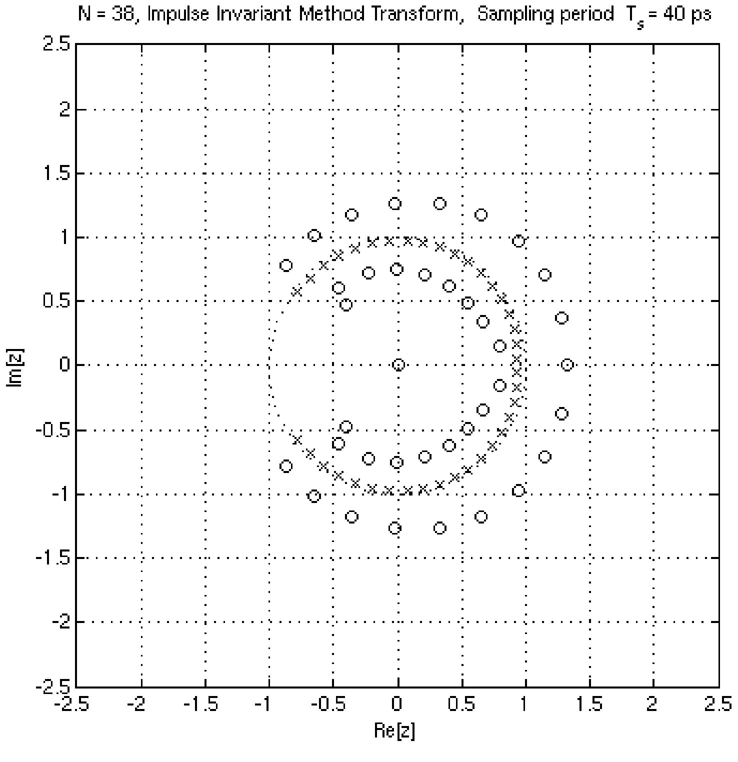


Figure 6-7: Pole-Zero Map in z-Plane for the 4th-Order Test Circuit for $N = 38$ and $T_s = 40$ ps Using the Impulse Invariant Transform

Note that the poles are all within the unit circle, suggesting that the system is stable, and that the poles' locations map in exactly the same manner as those found when using the matched-Z transform, as depicted previously in Figure 6-3. This is expected, since the pole mapping uses the same $z = e^{sT_s}$ algorithm for both the matched-Z and impulse invariant transforms. The zero locations, however, can be seen to map quite differently when using the two methods. Note specifically in Figure 6-7 the zero at the origin of the z-plane. This zero at $z = 0$ does not have a corresponding pole in the s-plane, unless it is considered to be the pole at $s \rightarrow \infty$, which is not a satisfying result, since the zero at a high analog frequency in the s-plane has mapped to a zero at a low discrete frequency in the z-plane. Indeed, it is this very problem with the impulse invariant transform (IIT) that has led to its decreased use as a filter frequency-domain design tool [81].

It should be noted here that this problem with the IIT should not be surprising, given the transformation's origin. Recall from Chapter 3 that the IIT originated by mapping the continuous-time impulse function $h_c(t)$ to the discrete-time impulse function $h_d(n)$, leading to great certainty in the time-domain representation of $h_d(n)$. The time-frequency uncertainty principle then becomes relevant, leading to some uncertainty in the frequency domain or, equivalently in this case, with $H_d(z)$.

6.7 Assessment of Discrete Transfer Function Generation Results via Examination of Frequency-Domain and Time-Domain Responses

A complete assessment and comparison of the z-domain discrete transfer functions given by (6.5), (6.7), and (6.10) using the bilinear transform (BLT), matched-Z transform (MZT), and impulse invariant transform (IIT), respectively, require a close examination of their frequency-domain and time-domain responses, in addition to the pole-zero maps examined above. The following sections will assess these responses for the system of the 4^{th}-order test circuit (with delay).

Note that the time-frequency uncertainty principle introduced in Chapter 3 applies here, and that there are tradeoffs involved when choosing how to assess the model performance. Specifically, the time-frequency uncertainty principle states that it is not possible to have low uncertainty in the time domain and the frequency domain responses simultaneously [50]. Even though uncertainty is present, however, additional knowledge of the system characteristics in both the time and frequency domains still can be expected to provide a more complete understanding than examination of the system in only a single domain.

6.7.1 Frequency Response Comparison of Models

The frequency responses $H_a(j\Omega)$ and $H_d(e^{j\omega})$ associated with the system transfer functions $H_a(s)$ and $H_d(z)$, respectively, are comprised of both the magnitude response and the phase response. Examination of these responses enables a more complete assessment of the system's pole and zero locations, and lends additional insight into how the associated system models will ultimately perform. Frequency domain representations are common in applications where sinusoidal excitation signals are present, where stability is an important consideration, or when frequency band characteristics are an important design constraint [39], [40], [48]. Examples of fields where frequency-domain design methods are widely used are power systems, communication systems utilizing carrier signals, feedback-based control systems, and filter design.

In previous sections, the transfer function $H_d(z)$ was generated in the complex frequency domain using the bilinear transform, the matched-Z transform, and the impulse invariant transform, and the pole-zero maps were plotted. From this pole-zero analysis, it was seen that there was variation in the resulting pole placement for the bilinear transform which was correctable with considerable pre-warping efforts, and also variation in the zero placement for all three of the transforms. In all cases, the system was seen to be stable, as indicated by the fact that all of the poles were located inside the unit circle in the z plane. Beyond these important generalizations, however, it was difficult to assess more exact implications on the system's time and frequency responses using only the pole-zero map assessments.

6.7.1.1 Frequency-Domain Responses for $H_d(z)$ Obtained Using Bilinear Transforms

Figure 6-8 shows a comparison of the analog frequency response $H_a(j\Omega)$ versus the discrete frequency responses $H_d(e^{j\omega})$ calculated using the bilinear transform, for the three cases where there was: 1) no pre-warping; 2) pre-warping at the single critical frequency $f = 10$ GHz; and 3) pre-warping of the entire discrete frequency variable ω. Note that the only discrete frequency response $H_d(e^{j\omega})$ that correlates well with the analog response $H_a(j\Omega)$ is that obtained when the entire frequency variable is pre-warped, and that this response overlays the analog response almost perfectly.

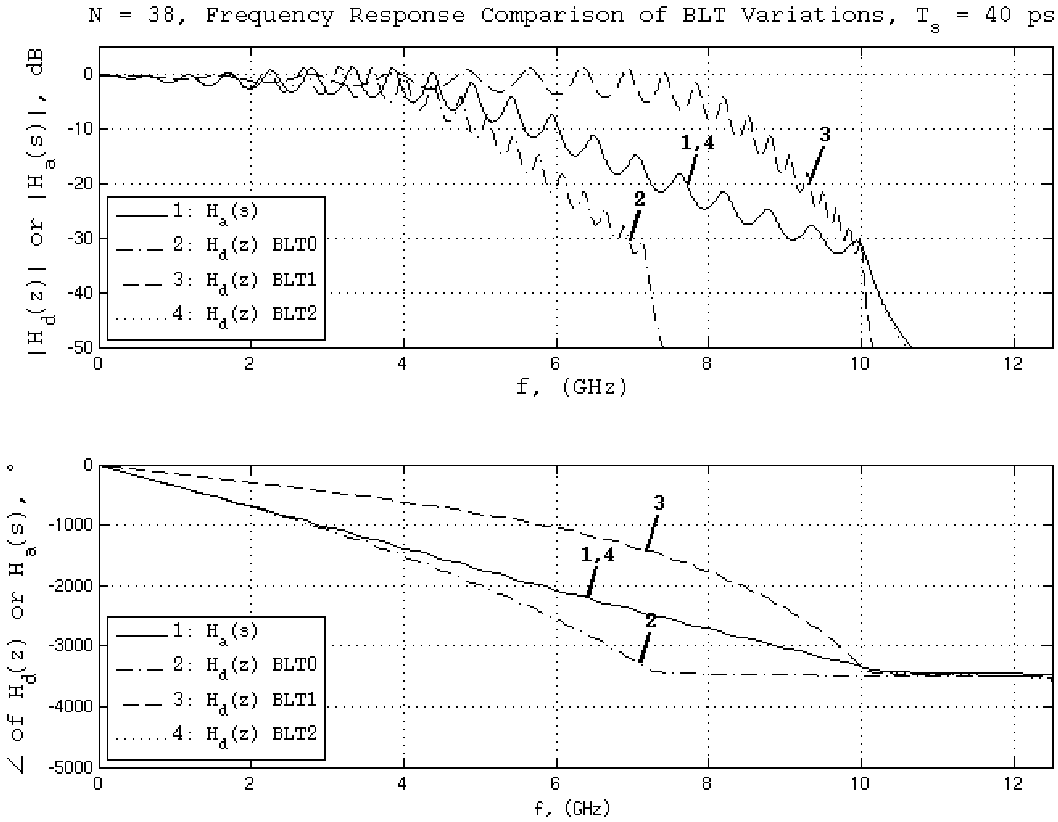


Figure 6-8: Frequency Response *$H_d(z)$* for 4th-Order Test Circuit using BLT0, BLT1, and BLT2

It was stated previously that the pole-zero maps for $H_d(z)$ obtained using the bilinear transform, both without pre-warping, and with pre-warping applied only at $f = 10$ GHz, exhibited significant variation in pole and zero placement, and that it was difficult to get good correlation with the pole-zero map generated using the matched-Z transform over the entire frequency range. The same characteristic is seen in Figure 6-8, in that the response can be made to match at a given frequency, but not over the entire frequency range. This is seen to be true for both the magnitude response and phase response. It is clear that the bilinear transform exhibits serious problems in transforming the 4th-order test circuit's transfer function unless the entire frequency vector is warped and the complex process depicted in Figure 6-6 is applied.[10], [11], [12] A significant disadvantage of this process is that the vector fitting step is not applied directly to the analog frequency response $H_a(j\Omega)$, but rather to a pre-warped version $H_a'(\frac{\omega_c'}{2\pi T_s})$, and thus the resulting macromodel's s-domain transfer function $H_a'(s')$ becomes an intermediate set of data that is ultimately thrown away. In contrast, with the matched-Z transform and impulse invariant transform, the model $H_a(s)$ obtained via vector fitting is useful for both s-domain analysis and also as the analog model on which the s- to z- transformation is directly applied.

6.7.1.2 Frequency-Domain Responses for *$H_d(z)$* Obtained Using Matched-Z and Impulse Invariant Transforms

Figure 6-9 shows a comparison of the frequency response $H_a(j\Omega)$ versus the frequency responses $H_d(e^{j\omega})$ calculated using both the matched-Z transform and the impulse invariant transform. Note that these transform methods give significantly better performance in replicating the analog function's magnitude and phase response than the BLT methods with no pre-warping or pre-warping at only a single critical frequency. The phase response is seen to be particularly good for both the MZT and IIT cases, an important observation given that the propagation delay that is present in the system is one of the most important system characteristics to model accurately. Notice the matched-Z transform's magnitude response starts to droop vs. the analog response at high frequencies. This is an artifact of the process where the analog gain K_a and the digital gain K_d of (6.6) and (6.7), respectively, are being matched at DC.

---

[10] BLT0 represents the bilinear transform without pre-warping.

[11] BLT1 represents the bilinear transform with pre-warping only at 10 GHz.

[12] BLT2 represents the bilinear transform with pre-warping of the entire discrete frequency vector.

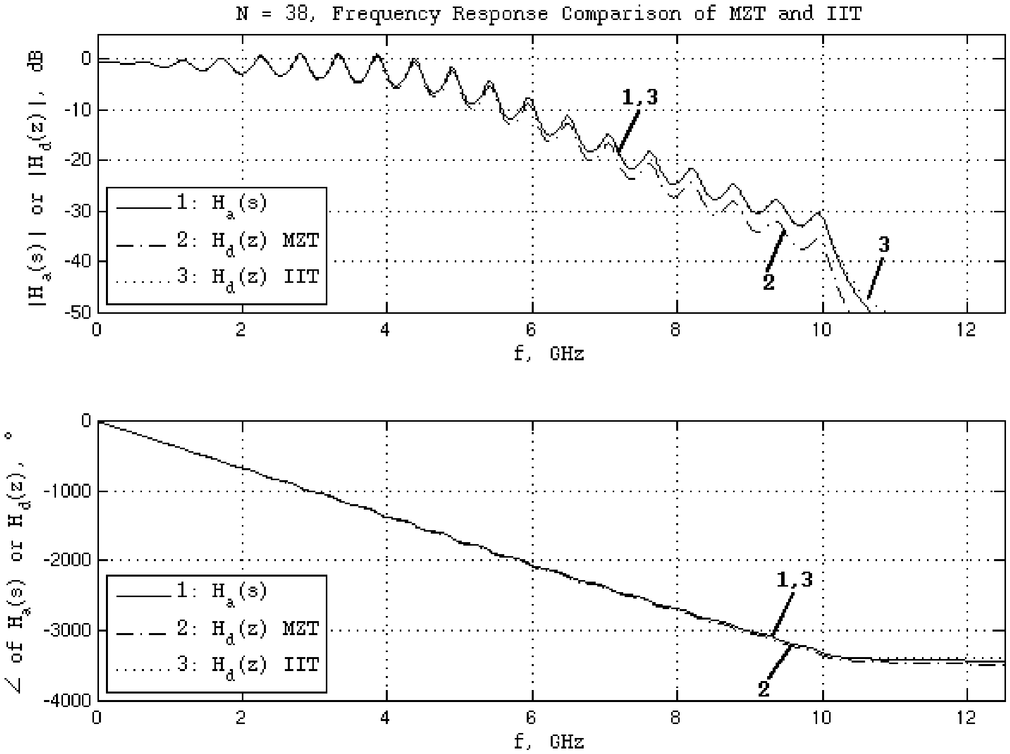


Figure 6-9: Frequency Response *$H_d(z)$* for 4th-Order Test Circuit Using the Matched-Z and Impulse Invariant Transforms

One interesting observation regarding the frequency responses of Figure 6-9 is that the differences in the zero positions resulting from the two transformation methods, which would intuitively seem to be significant based on assessment of the pole-zero maps of Figure 6-3 and Figure 6-7, apparently have minimal effect on the frequency response for frequencies under about 6 GHz, although they do affect the magnitude response at high frequencies. These differences can only be due to the difference in zero positions or the gain, because the matched-Z method and the impulse invariant transformations map the poles in an identical manner.

6.7.2 Time-Domain Response Comparison of Models

In many cases in signal integrity engineering, the ultimate objective with a system transfer function model or frequency transfer function model is to use the model for calculating the time-domain response to a known stimulus [93]. In contrast to the fields mentioned previously, which use primarily frequency-domain analysis, there are several fields which utilize primarily time-domain analysis. Examples of such fields are signal integrity engineering and the simulation of transients on power transmission lines [7], [68]. In this regard, it would be interesting to use the system transfer functions $H_d(z)$ developed in this section from $H_a(s)$ using the bilinear transform (BLT), matched-Z transform (MZT), and impulse invariant transform (IIT) to calculate the time response of the system.

It should be noted that, in fields dominated by time-domain analysis, it is not unusual to immediately move from a transfer function model to a state-space representation, without giving much thought to the frequency characteristics of the system that would be evident by examining the transfer function model in the various frequency domains. Recall that the transfer functions $H_a(s)$ or $H_d(z)$ can be used to calculate the time responses by writing differential equations or difference equations, respectively [40]. From these equations, a state matrix formulation can be formulated allowing direct calculation of the time responses [48], [49]. For the examples to follow, the calculation of the time responses was handled directly using known MATLAB functions timeresp.m [61] for $H_a(s)$ and lsim.m for $H_d(z)$ [80].

6.7.2.1 Time-Domain Responses for $H_d(z)$ Obtained Using Bilinear Transforms

One method for assessing the time responses calculated from $H_d(z)$ is to compare them directly to the time response obtained using $H_a(s)$. Recall that the time response calculated from $H_a(s)$ was assessed previously vs. that obtained from SPICE in Chapter 5, and determined to overlap almost perfectly. This correlation, along with qualitative evaluation of the frequency response, helped to generate confidence that $H_a(s)$ was providing a reasonable representation of the system, even though a system of fairly high order $N = 38$ was required.

Figure 6-10 shows the transient response obtained using the analog transfer function $H_a(s)$ obtained from vector fitting in Chapter 5 and the transfer functions $H_d(z)$ of Figure 6-8 obtained using the bilinear transform without pre-warping, with pre-warping at the single frequency $f = 10$ GHz, and with pre-warping applied to the entire frequency vector[13,14,15]. Note that the transient response for the case where pre-warping was applied only at $f = 10$ GHz correlates poorly from the perspective of propagation delay, which is directly attributable to the curved nature of the phase curve shown in Figure 6-8.

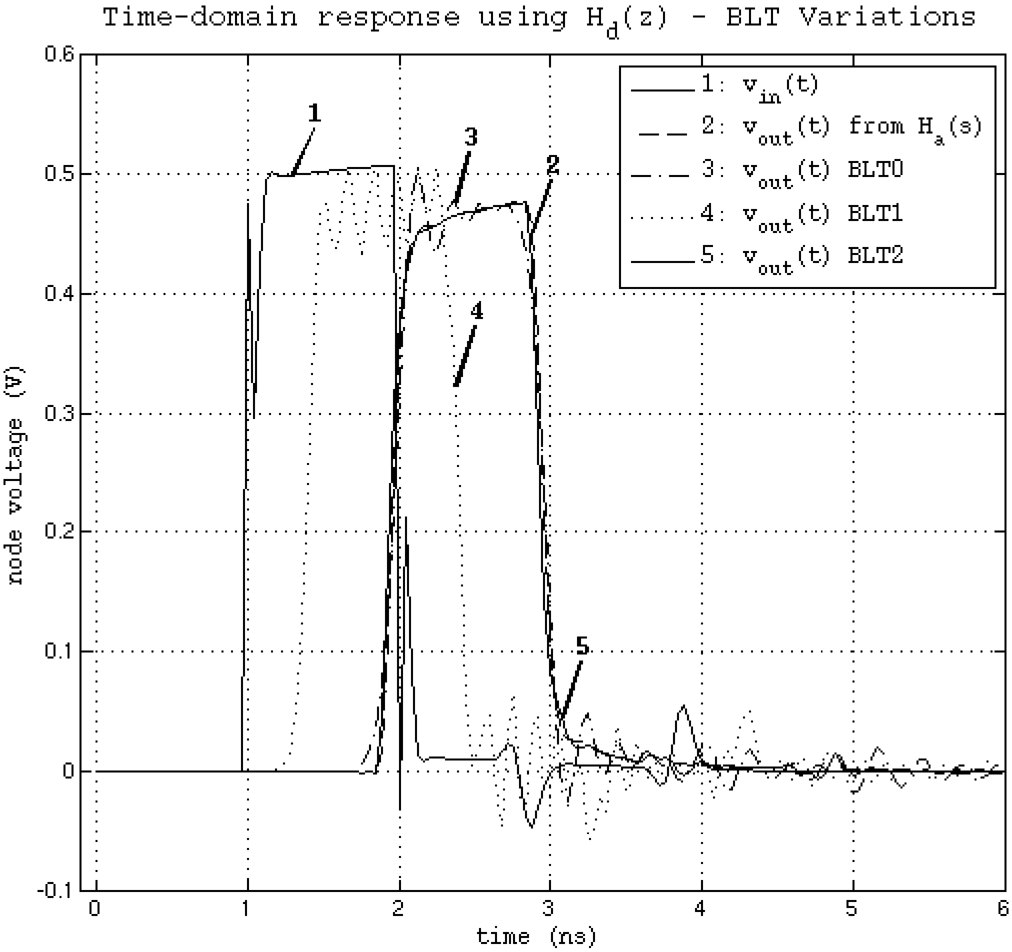


Figure 6-10: Time Response of the 4th-Order Test Circuit Calculated Using *$H_d(z)$* Obtained from BLT0, BLT1, and BLT2

The transient response for the non pre-warped case shows better delay characteristics, although it can be seen that the delay is off in this case as well. Note that both cases exhibit significant ripple near the voltage rails. While neither the delay nor the

---

[13] BLT0 represents the bilinear transform without pre-warping.

[14] BLT1 represents the bilinear transform with pre-warping only at 10 GHz.

[15] BLT2 represents the bilinear transform with pre-warping of the entire discrete frequency vector.

ripple inaccuracies are desirable, for a distributed system with inherent propagation delay the misrepresentation of the system's delay characteristics is particularly troubling.

It should also be noted that the inaccuracies of the delay for both the pre-warped and the non pre-warped cases are non-causal in nature. This type of inaccuracy is considered particularly bad because it is indicating something happens before it is physically possible [63]. In contrast, it can be seen from Figure 6-10 that the BLT implemented using the Ikai method, where the entire discrete frequency variable ω has been pre-warped per the process of Figure 6-6, has excellent correlation in the transient response to that of the original analog transfer function $H_a(s)$. Recall from Figure 6-8 that application of the Ikai method also led to excellent correlation of the discrete frequency response $H_d(e^{j\omega})$ to the analog frequency response $H_a(j\Omega)$. Clearly, the application of the Ikai method at least enables the BLT to be useful for signal integrity applications, despite the complexity and drawbacks of the method.

6.7.2.2 Time-Domain Responses for $H_d(z)$ Obtained Using Matched-Z and Impulse Invariant Transforms

Figure 6-11 shows the transient responses obtained using the transfer function $H_a(s)$ obtained from vector fitting in Chapter 5 and the transfer functions $H_d(z)$ of Figure 6-9 obtained using the matched-Z transformation and the impulse invariant transformation. Note that the transient responses for both cases show very good correlation with the response generated directly from $H_a(s)$. Recall from Figure 6-3 and Figure 6-7 that both transforms map the poles in an identical manner, that both transforms resulted in $H_d(z)$ functions with good phase characteristics, and that only the zeros mapped differently between the two transforms. Any deviations in system delay characteristics are difficult to detect, at least using manual visual comparisons. It is becoming clear from the combined analysis of the pole-zero maps, the frequency responses, and the time-domain responses, that the positions of the zeros may not be that significant, and that the phase characteristics are very significant, to the time-domain response of the system.

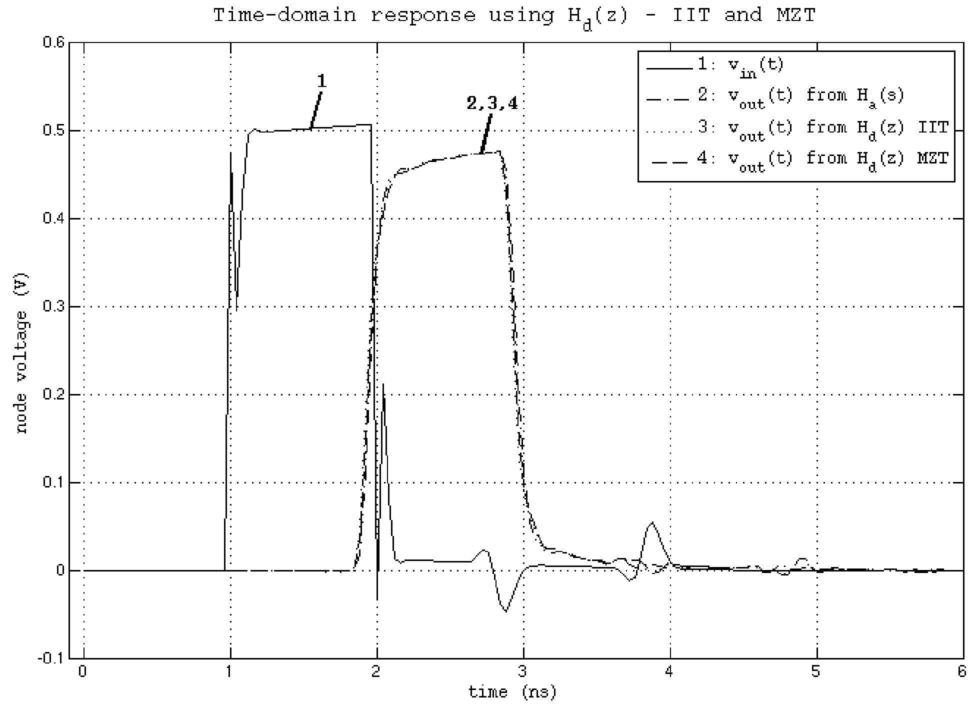


Figure 6-11: Time Response of the 4th-Order Test Circuit Calculated Using $H_d(z)$ Obtained from the Matched-Z and Impulse Invariant Transforms

6.7.3 Additional Comments Regarding Results for $H_d(z)$ Obtained with Bilinear, Matched-Z, and Impulse Invariant Transforms

As mentioned previously, the Laplace domain and z-domain can be considered to be analogous, since both are complex frequency domains and both contain complete information regarding stability, causality, and passivity [42]. From this standpoint, it might seem reasonable to expect that either domain is equally suited to performing pole-zero analysis of transfer functions. Nevertheless, it is clear from the examples studied here that the z-domain has several advantages over the Laplace domain in this regard, some of which are largely pragmatic in nature.

Even though the bilinear transform did not perform well in the transformation of the function $H_a(s)$ in the s-domain to the function $H_d(z)$ in the z-domain, it did have one advantage that proved useful. That advantage was that the transformation, as expressed in (6.5), was directly applicable to the partial fraction expansion form of $H_a(s)$, expressed in (6.4). This characteristic, which is shared by the impulse invariant transformation of

(6.10), allows the function to be transformed without first having to convert to a ratio of factored polynomials in the form of (6.6), a difficult step that is required by the matched-Z transform. A major advantage of this is that any effects of inadequate numerical precision encountered in converting from pole-residue form to rational function form are not encountered until the function is already transformed into z-domain, allowing the application of known techniques to mitigate this problem [59]. Thus the bilinear transformation has an inherent advantage in handling numerical precision issues, but suffers from poor performance due to frequency aliasing. Similarly, the impulse invariant method has this same inherent advantage, but suffers from limitations in its zero-mapping characteristics.

In contrast, the matched-Z transformation has good zero and pole matching characteristics, good frequency response characteristics, and good time response characteristics, at the expense of some exposure to numerical precision issues. Clearly, it would be advantageous to have a form of the matched-Z transform that could be applied to the 1st-order functions of (6.4), giving the dual advantages of avoiding the numerical precision issues seen when converting between pole-residue form and rational function form in the s-domain, and allowing the use of the previously-mentioned, known techniques for mitigating the numerical precision limitations in the z-domain.

At the time just prior to the publication of this dissertation, the application of these methods to signal integrity applications had been under consideration for patent protection. These methods will now be submitted for publication as an Institute of Electrical and Electronic Engineers (IEEE) peer-reviewed journal article [94].**[16]**

6.8 Conclusions Regarding $H_a(s)$ to $H_d(z)$ Transformation Methods

The results presented in this chapter demonstrate that it is possible to successfully transform the analog complex frequency domain transfer function $H_a(s)$ into the digital complex frequency domain transfer functions $H_d(z)$ using common methods. These

---

[16] For information regarding the final disposition of publication status, please contact the author of this dissertation after 31 July 2014. Contact information is listed in Appendix 2.

methods were effective enough to enable pole-zero analysis in the z-plane and useful examination of the frequency responses. In addition, it was shown to be possible to use the $H_d(z)$ transfer functions derived by these methods for accurate calculation of the time response.

Of these methods used for calculating $H_d(z)$, the bilinear transform seemed to have the most difficulties in matching both the frequency and time responses of $H_a(s)$. Most notably, unless the entire discrete frequency ω is pre-warped prior to the application of the BLT, significant distortion can be seen in the pole and zero locations relative to those given by the matched-Z transformation and the impulse invariant transformation. Attempting to match the frequency response at a single critical frequency point was not significantly helpful, due to both the nature and shape of the frequency response magnitude and phase curves. In addition, the bilinear transform demonstrated relatively poor correlation in the phase response, for both the case when no pre-warping was applied, and when pre-warping was applied only to a single frequency point. This was an unfortunate development given the propagation delay inherent in the system. The time response obtained using the bilinear transform for these two cases also did not overlay well with that obtained from $H_a(s)$.

Fortunately, the problems the BLT displayed with poor matching of the phase and time-domain responses disappeared when the discrete frequency ω was pre-warped over its entire range, and the method by Ikai was applied. The process had disadvantages, however, in that it required complex manipulation of the frequency response, the creation of a pre-warped version of the analog frequency response, and a vector fitting step applied to this pre-warped response which resulted in a pre-warped version of the analog transfer function, all before the BLT could be applied. This process is not as straightforward as the matched-Z and impulse invariant transforms, and the pre-warped versions of the frequency and transfer functions were not useful for analog domain analysis, so they were simply discarded.

In contrast, the matched-Z transform resulted in a straightforward mapping of the pole and zero locations from the s-plane to the z-plane. The locations agreed relatively well with results utilizing the impulse invariant transform. The frequency response exhibited good matching to that for $H_a(s)$ in the magnitude response, with some deviation showing up only near the Nyquist frequency f_N due to the expected presence of a zero there at $z = -1$ [81], and even better matching in the phase response. The time response correlation with that given by $H_a(s)$, and also with that given by $H_d(z)$ using the impulse invariant transform, was also excellent.

Similar to the matched-Z transform, the impulse invariant transform also exhibited good correlation between $H_d(z)$ and $H_a(s)$. The frequency response exhibited excellent correlation with $H_a(s)$, both in the magnitude and phase responses. The magnitude response matched that for $H_a(s)$ even better than that of $H_d(z)$ obtained using the matched-Z transformation, especially near the Nyquist frequency f_N since there was no zero there, but rather at the origin $z = 0$. The phase response was almost identical to both that of $H_d(z)$ using the matched-Z transform, and that of $H_a(s)$. The time response also overlaid that from $H_a(s)$ almost perfectly, as expected since the impulse invariant transformation it based on preserving the shape and properties of the impulse response $h_a(t)$ in the sampled domain [42].

In the next chapter, additional attention will be given to the s- to z-domain transformations utilized in this chapter, introducing modified versions of both the bilinear transform and the matched-Z transform that can operate directly on the 1st-order sections of $H_a(s)$.

Addendum 6A: Derivation of Bilinear Transform Expression for the Partial Fraction Expansion Form of $H_a(s)$ and $H_d(z)$

Included below is the derivation of the expression of $H_d(z)$ using the bilinear transform in (6.5), as applied to the partial fraction expansion form of $H_a(s)$ as expressed in (6.4). It is assumed that the propagation delay $T_d = 0$, i.e., (6.4) becomes:

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = \left(\sum_{k=1}^{N} \frac{c_k}{s - a_k} + D\right). \tag{6.11}$$

Applying the bilinear transform,

$$H_d(z) = H_a(s)\big|_{s=\frac{2}{T_s}\left(\frac{z-1}{z+1}\right)} = \left(\sum_{k=1}^{N} \frac{c_k}{\frac{2}{T_s}\left(\frac{z-1}{z+1}\right) - a_k} + D\right). \tag{6.12}$$

Multiplying the numerator and denominator polynomials by $\frac{T_s}{2}(z+1)$ leads to

$$H_d(z) = \left(\sum_{k=1}^{N} \frac{\frac{T_s c_k}{2}(z+1)}{(z-1) - \frac{T_s a_k}{2}(z+1)} + D\right). \tag{6.13}$$

Grouping like terms in the denominator, this simplifies further to

$$H_d(z) = \left(\sum_{k=1}^{N} \frac{\frac{T_s c_k}{2}(z+1)}{(1 - \frac{T_s a_k}{2})z - (1 + \frac{T_s a_k}{2})} + D\right). \tag{6.14}$$

Finally, isolating the z term in the denominator and the $z+1$ term in the numerator, along with explicitly expressing the $T_s/2$ term, leads to

$$H_d(z) = \left(\sum_{k=1}^{N} \frac{\frac{c_k \frac{T_s}{2}}{1 - a_k \frac{T_s}{2}}(z+1)}{z - \frac{1 + a_k \frac{T_s}{2}}{1 - a_k \frac{T_s}{2}}} + D\right). \tag{6.15}$$

Rearranging terms slightly to simplify the ratio terms above leads to the form

$$H_d(z) = \left(\sum_{k=1}^{N} \frac{\frac{T_s c_k}{2 - a_k T_s}(z+1)}{z - \frac{2 + a_k T_s}{2 - a_k T_s}} + D\right), \tag{6.16}$$

which is exactly the relation expressed in (6.5). Note that this form has the advantages of explicitly showing the pole value, isolating the gain term K_d of (6.7) and showing explicitly that $H_d(z)$ has a zero at $z = -1$.

Addendum 6B: Derivation of Bilinear Transform Expression Pre-warping Factor *H_d(z)*

The bilinear transform of (6.3) automatically matches the frequency response at the DC frequency $\omega = 0$ [92], leading to frequency warping distortion at higher frequencies. To account for this, (6.16) can be modified to match the frequency response at an arbitrary frequency Ω_o, instead of at the DC point. To do so, frequency scaling techniques can again be applied, as was done previously in Chapter 5.

Specifically, the bilinear transform gives the digital transfer function $H_d(z)$ at a particular frequency of interest ω_o, which maps on the $j\Omega$ axis to the distorted frequency

$$\Omega_{warp} = \frac{2}{T_s} \tan \frac{\Omega_o T_s}{2}, \tag{6.17}$$

rather than to the desired frequency of interest Ω_o that was originally observed in the analog transfer function $H_a(s)$. To correct for this, a pre-warping procedure can be applied prior to application of the bilinear transform [42].

Using frequency scaling, the analog transfer function $H_a(s)$ given by (6.11) can be first transformed to a prototype filter with the specified frequency at $\Omega = 1$ Hz, using

$$H_{proto}(s) = H_a(s)|_{s \to sk_{fup}} = H_a(s)|_{s \to s\Omega_o}, \tag{6.18}$$

then scaled back upward in frequency to the pre-warped transfer function $H_{prewarp}(s)$ as follows:

$$H_{prewarp}(s) = H_{proto}(s)\Big|_{s \to \frac{s}{k_{fdown}}} = H_{proto}(s)\Big|_{s \to \frac{s}{\Omega_{warp}}} = H_a(s)|_{s \to \frac{s\Omega_o}{\Omega_{warp}}}. \tag{6.19}$$

The overall effective scaling factor k_{eff} can then be defined as

$$k_{eff} = \frac{\Omega_{warp}}{\Omega_o} = \frac{\frac{2}{T_s} \tan \frac{\Omega_o T_s}{2}}{\Omega_o}, \tag{6.20}$$

where the frequency scaling convention $s \to s/k_{eff}$ has been adopted. This scaling factor can be applied using (6.19) to $H_a(s)$ as a general pre-warping function,

$$H_{prewarp}(s) = H_a(s)|_{s \to \frac{s}{k_{eff}}}, \tag{6.21}$$

prior to the application of the bilinear transform (6.5). This approach is particularly useful when $H_a(s)$ is expressed in the pole-residue form of (6.11) as a partial fraction expansion.

Chapter 7: Development of the Transfer Function Model $H_d(z)$ Directly from the Partial Fraction Expansion Terms of $H_a(s)$

7.1 Introduction

It was shown in Chapter 6 that it is advantageous to be able to convert directly between $H_a(s)$ and $H_d(z)$ partial fraction expansion representations of the transfer function, without the requirement of first converting to pole-zero factored form. The primary motivation is twofold. First, the partial fraction expansion form of $H_a(s)$, a pole-residue form, is the most likely form of the expression for $H_a(s)$ to be encountered in a signal integrity engineering application, since this form is the natural output of the frequency-domain vector fitting process [44]. The pole-residue form, incidentally, also lends itself well to time-domain computations in SPICE [63]. Second, the process for converting $H_a(s)$ from partial fraction expansion form to pole-zero factored form is very difficult for systems with poles and residues in the GHz range, which is common for signal integrity interconnect systems.

In particular, it was shown in Chapter 6 that it is possible to make the conversion from $H_a(s)$ in partial fraction expansion (pole-residue) form to $H_d(z)$ using either the bilinear transform (BLT) or the impulse invariant transform (IIT), with the resulting expression for $H_d(z)$ itself also remaining in pole-residue form. In contrast, with the matched-Z transform (MZT) it is not possible to make the conversion from the s-domain to the z-domain without first converting $H_a(s)$ to a pole-zero factored form. This is because the transform is based on directly mapping poles and zeros from the s-plane to the z-plane, with the resulting expression for $H_d(z)$ itself also remaining in pole-zero factored form. The prerequisite conversion process from a pole-residue representation to a pole-zero representation in the s-domain was shown to be difficult for the 38th-order rational function approximation for $H_a(s)$, due to the effects of limited numerical precision when performing the computations. This was primarily caused by the large numerical values for the poles and residues of $H_a(s)$. This problem was ultimately solved by applying frequency scaling techniques [89] to $H_a(s)$ prior to converting from pole-residue form to pole-zero factored form.

The application of the matched-Z transform (MZT) to the problem of converting $H_a(s)$ to $H_d(z)$ is nevertheless still appealing. The matched-Z transform is straightforward, intuitive, and does not require consideration for pre-warping of frequencies [43]. If the function $H_d(z)$ could be obtained in pole-residue form using the matched-Z transform, with the analog function $H_a(s)$ also in pole-residue form, as was done for the impulse invariant transform (IIT) and bilinear transform (BLT), that would be advantageous. $H_d(z)$ could then be converted to pole-zero factored form in the z-domain, where the poles and residue values would be much smaller. It would then still be possible to study the system characteristics in the z-domain, using a pole-zero map in the z-plane, with its inherent advantages over s-plane analysis discussed previously in Chapter 6.

One point to remember when performing any of these transformations is that all of the widely-used methods for converting from the s-domain to the z-domain, namely, the bilinear transformation, the matched-Z transformation, and the impulse invariant transformation, are themselves approximations with some known limitations [41]–[43]. One obvious idea, then, is to develop new transform methods that may be more practical for high-order rational function approximations for $H_a(s)$. Even with some tradeoffs in accuracy, such functions could be very useful as long as the limitations are understood.

This chapter will expand upon the methods of the past chapter for generating $H_d(z)$ from $H_a(s)$ and for assessing the models' performance in the z-plane, discrete-frequency ω domain, and time domain. First, the general expressions for conversion algorithms for the bilinear transform, the impulse invariant transform, and the matched-Z transforms will be developed, and the transforms will be applied directly to the various 1^{st}-order sections which comprise the pole-residue form of $H_a(s)$. Second, an attempt will be made to determine whether the application of the matched-Z transform to these 1^{st}-order sections of $H_a(s)$ can result in a valid representation of $H_d(z)$, similar to the approach used by the bilinear transform and the impulse invariant transform. Third, an attempt will be made to determine whether the application of bilinear transform to these 1^{st}-order sections of $H_a(s)$ can result in a better representation of $H_d(z)$ than was achieved in Chapter 6, by changing the strategy for pre-warping frequencies to match the gain response. Finally, the results of the application of the new transforms directly to the 1st-order sections will be compared and contrasted in detail, leading to a set of general guidelines for the usage and application of s- to z-domain transformations to system interconnect models in signal integrity engineering.

7.2 Transfer Function Representation as an Expansion of 1st-Order Partial Fraction Terms

Recall that, because of the prevalence of vector fitting macromodeling processes in signal integrity engineering [26], [44], [45], [82], [84], the $H_a(s)$ function is most likely to be encountered in the partial fraction expansion form of its rational function approximation [63]:

$$H_a(s) = \frac{B_a(s)}{A_a(s)} = \left(\sum_{k=1}^{N} \frac{c_k}{s - a_k} + D\right) e^{-sT_d} \,. \tag{7.1}$$

Ignoring the direct term D and delay term e^{-sT_d} for the moment, note that the partial fraction expansion in (7.1) form is a sum of 1st-order partial fraction sections of the form

$$H_{ak}(s) = \frac{c_k}{s - a_k} \,. \tag{7.2}$$

Note also that, in the s-plane, this 1st-order function $H_{ak}(s)$ would have a pole at $s = a_k$ and that

$$\lim_{s\to\infty} H_{ak}(s) = \lim_{s\to\infty} \frac{c_k}{s - a_k} = 0 \,, \tag{7.3}$$

thus $H_{ak}(s)$ can also be said to have a zero at infinity [43]. As mentioned previously, one of the pragmatic limitations of s-plane analysis is the presence of poles and zeros at distances very far from the axes' origin, and this is a good example of such a case.

It is useful to now examine the transformation of these s-domain 1st-order terms using various s-domain to z-domain transform methods. For some high-order transfer functions, even if it is difficult to obtain a pole-zero map of the entire transfer function $H_a(s)$ or $H_d(z)$, it still may be possible to gain some insight by studying a pole-zero map of the various 1st-order sections. In some cases, it may be useful to study both the pole-zero map of the overall function and the pole-zero map of the 1st-order sections. In either case, the ultimate objective is to gain insight into the nature of the system.

7.2.1 Transformation of 1st-Order Partial Fractions Using the Bilinear Transform

Applying the bilinear transform to the 1st-order partial fraction section of (7.2) results in

$$H_{dk_{BLT}}(z) = H_{ak}(s)\Big|_{s=\frac{2}{T_s}\left(\frac{z-1}{z+1}\right)} = \frac{\frac{T_s c_k}{2-a_kT_s}(z+1)}{z-\frac{2+a_kT_s}{2-a_kT_s}} \; . \tag{7.4}$$

In contrast to the s-domain function $H_{ak}(s)$ in (7.2) above, which was found to have a single zero at infinity and a single pole at $s = a_k$, the transformed z-domain function $H_{dk_{BLT}}(z)$ of (7.4) has a single zero at $z = -1$, and a single pole at $z = \frac{2+a_kT_s}{2-a_kT_s}$.

Note in particular that the z-domain version has a zero with a position that is less ambiguous in the z-plane than it had been in the s-plane. It also has the advantage of being clearly visible at a finite location $z = -1$ on the real z axis in the z-plane, as opposed to being located off the plot in an infinite location in the s-plane.

7.2.2 Transformation of 1st-Order Partial Fractions Using the Matched-Z Transform

Applying the matched-Z transform to the 1st-order partial fraction section of (7.2) results in

$$H_{dk_{MZT}}(z) = \frac{-\frac{c_k}{2a_k}\left(1-e^{a_kT_s}\right)(z+1)}{z-e^{a_kT_s}} \; , \tag{7.5}$$

assuming that the digital gain parameter K_{dk} of $H_{dk_{MZT}}(z)$ is matched to the analog gain parameter K_{ak} of $H_{ak}(s)$ at the DC point, namely, where $s = 0$ and $z = 1$. In contrast to the s-domain function $H_{ak}(s)$ in (7.2) above, which was found to have a single zero at infinity and a single pole at $s = a_k$, the transformed z-domain function $H_{dk_{MZT}}(z)$ of (7.5) has a single zero at $z = -1$, and a single pole at $z = e^{a_kT_s}$.

Note in particular that the z-domain version has a zero with a position that is less ambiguous in the z-plane than it had been in the s-plane, just as in the case for the bilinear transform. Note also, however, that the pole position given by the bilinear transform and the matched-Z transform are different, due to the different mapping methods used for the poles.

7.2.3 Transformation of 1st-Order Partial Fractions Using the Impulse Invariant Transform

Applying the impulse invariant transform to the 1st-order partial fraction section of (7.2) results in

$$H_{dk_{IIT}}(z) = \frac{T_s c_k z}{z - e^{a_k T_s}} \ . \tag{7.6}$$

In contrast to the s-domain function $H_{ak}(s)$ in (7.2) above, which was found to have a single zero at infinity and a single pole at $s = a_k$, the transformed z-domain function $H_{dk_{IIM}}(z)$ of (3.39) has a single zero at $z = 0$, and a single pole at $z = e^{a_k T_s}$.

Note in particular that the z-domain version has a zero with a position that is less ambiguous in the z-plane than it had been in the s-plane, just as in the case for the bilinear transform and the matched-Z transform. Note also, however, that the zero position of $z = 0$ given by the impulse invariant transform and the position of $z = -1$ given by the bilinear and matched-Z transforms are considerably different. It could be argued that the impulse invariant transform actually maps the zero position incorrectly [81].

7.2.4 Comparison of Results for Various Transformation Methods for the 1st-Order Partial Fractions

Table 7-1 shows a comparison of the various results for all 38 individual 1st-order sections $H_{dk}(z)$ for the 4th-order test circuit using the bilinear, matched-Z, and impulse invariant transforms, where the 1st-order partial fraction terms have general form:

$$H_{dk}(z) \ = \frac{B_{dk}(z)}{A_{dk}(z)} = \ \frac{b_{1k} z + b_{0k}}{z - p_k}, \tag{7.7}$$

as compared to (7.4), (7.5), and (7.6), respectively.

Parameter	**Bilinear Transform (BLT)**	**Impulse Invariant Transform (IIT)**	**Matched-Z Transform (MZT)**
zeros	-1 (x38)	0 (x38)	-1 (x38)
poles	-0.21893±0.96287i	-0.77876±0.57506i	-0.77876±0.57506i
	-0.15944±0.97166i	-0.68039±0.68259i	-0.68039±0.68259i
	-0.097013±0.98018i	-0.57486±0.77753i	-0.57486±0.77753i
	-0.029911±0.98411i	-0.45633±0.85418i	-0.45633±0.85418i
	0.041792±0.98354i	-0.3289±0.91291i	-0.3289±0.91291i
	0.11755±0.97723i	-0.19584±0.95215i	-0.19584±0.95215i
	0.19679±0.96431i	-0.060443±0.9719i	-0.060443±0.9719i
	0.27832±0.94397i	0.073574±0.97261i	0.073574±0.97261i
	0.36089±0.91547i	0.20319±0.95536i	0.20319±0.95536i
	0.44339±0.87804i	0.32629±0.92145i	0.32629±0.92145i
	0.52577±0.83008i	0.44287±0.87127i	0.44287±0.87127i
	0.60759±0.77015i	0.55244±0.80527i	0.55244±0.80527i
	0.68671±0.69695i	0.65275±0.72384i	0.65275±0.72384i
	0.76011±0.61019i	0.74107±0.62843i	0.74107±0.62843i
	0.82416±0.5106i	0.81461±0.52121i	0.81461±0.52121i
	0.87567±0.3995i	0.87149±0.40445i	0.87149±0.40445i
	0.91172±0.28104i	0.91017±0.28263i	0.91017±0.28263i
	0.92559±0.050897i	0.92557±0.050835i	0.92557±0.050835i
	0.92865±0.16191i	0.92821±0.1621i	0.92821±0.1621i

Table 7-1: Comparison of 1st-Order Sections' Pole/Zero Parameters for the Bilinear, Impulse Invariant, and Matched-Z Transforms

Notice from the table that the zero locations of the 1st-order sections are identical for the bilinear and matched-Z transforms, while the pole locations are identical for the matched-Z and impulse invariant transforms, as expected.

Figure 7-1 shows a pole-zero plot in the z-plane for each of the 38 1st-order functions $H_{dk_{BLT}}(z)$ given by (7.4). Note that each 1st-order function has a zero at $z = -1$, while the poles are same as those shown for the overall function $H_d(z)$ derived by the bilinear transform in Chapter 6. Notice that the pole locations are compressed over a smaller frequency range due to the frequency warping effects observed in Chapter 6, and which are inherent to the bilinear transform [42].

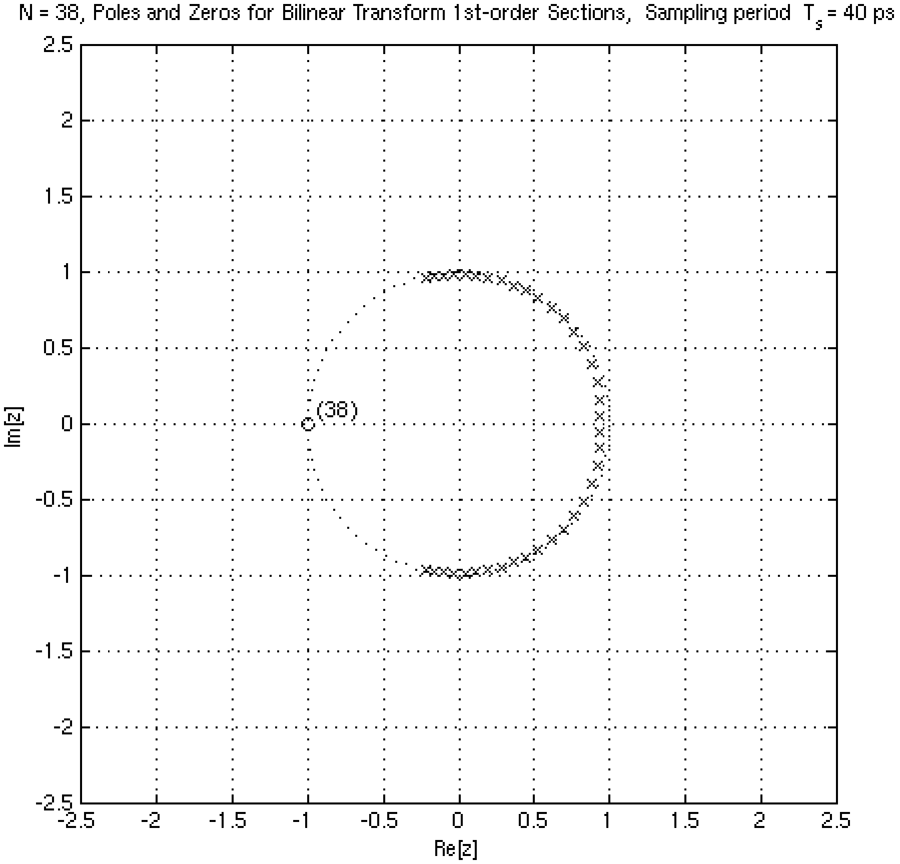


Figure 7-1: Pole-Zero Plot in the z-Plane for 1st-Order Sections of *$H_d(z)$* Derived Using the Bilinear Transform, for the 4th-Order Test Circuit

Figure 7-2 shows a pole-zero plot in the z-plane for each of the 38 1st-order functions $H_{dk_{MZT}}(z)$ given by (7.5). Note that each 1st-order function has a zero at $z = -1$, while the poles are same as those shown for the overall function $H_d(z)$ derived by the matched-Z transform in Chapter 6. Notice that the pole locations are not compressed over the smaller frequency range depicted in Figure 7-1, due to the fact that the matched-Z transform does not exhibit frequency warping like the bilinear transform [43].

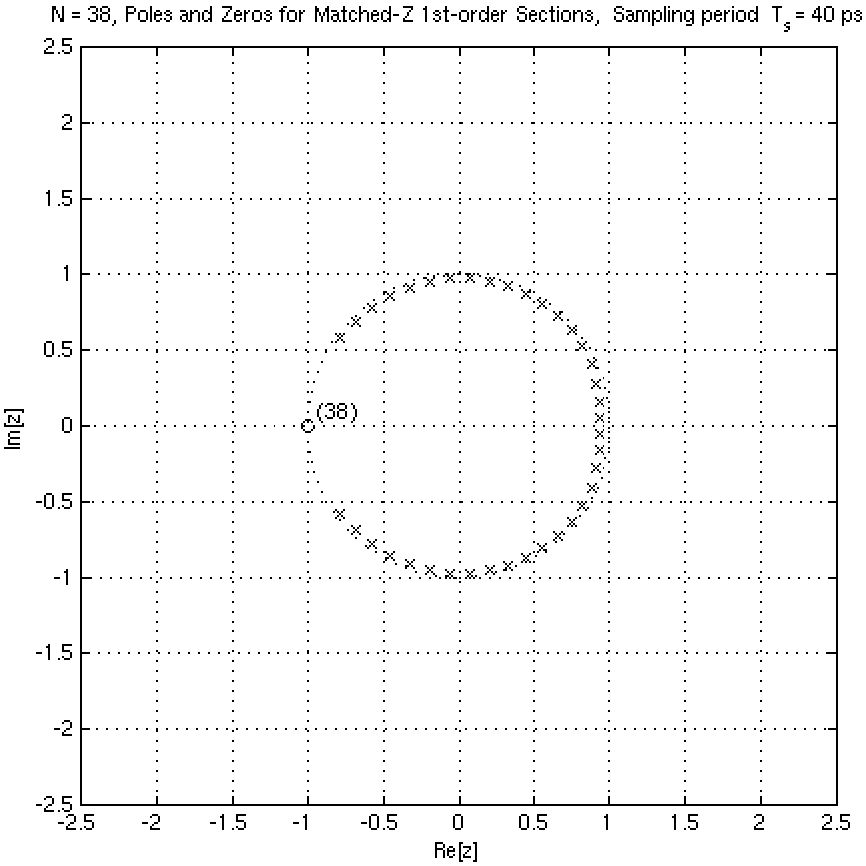


Figure 7-2: Pole-Zero Plot in the z-Plane for 1st-Order Sections of *$H_d(z)$* Derived Using the Matched-Z Transform, for the 4th-Order Test Circuit

Figure 7-3 shows a pole-zero plot in the z-plane for each of the 38 1st-order functions $H_{dk_{IIT}}(z)$ given by (7.6). Note that each 1st-order function has a zero at $z = 0$, as opposed to at $z = -1$ for the bilinear and matched-Z transform cases. The poles, on the other hand, are same as those shown for the overall function $H_d(z)$ derived by the impulse invariant transform in Chapter 6. The poles are also identical to those found using the matched-Z transform, and are not compressed over the smaller frequency range depicted in Figure 7-1, due to the fact that the impulse invariant transform does not exhibit frequency warping like the bilinear transform [43].

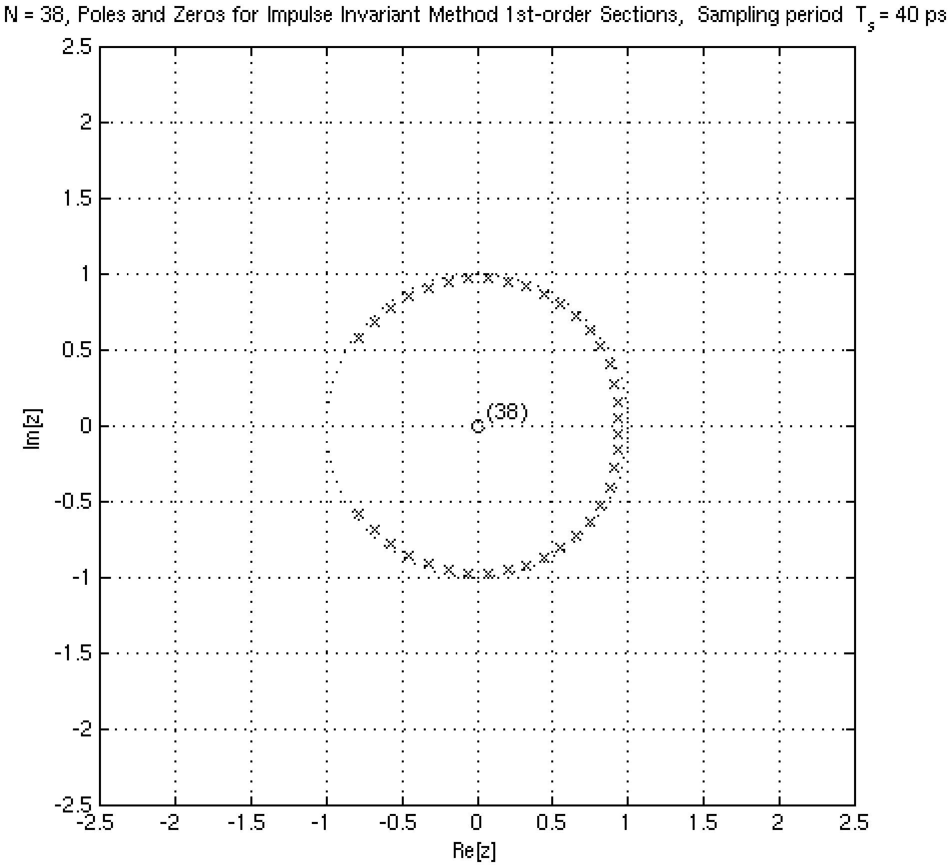


Figure 7-3: Pole-Zero Plot in the z-Plane for 1st-Order Sections of *$H_d(z)$* Derived Using the Impulse Invariant Transform, for the 4th-Order Test Circuit

7.3 Improved Overall $H_a(s)$ to $H_d(z)$ Transformation Functions Based Upon s-to-z-Domain Transformations Applied to 1st-Order Sections $H_{ak}(s)$

The results obtained in Chapter 6 for the development of the digital complex frequency domain transfer functions $H_d(z)$ from $H_a(s)$, using the three most well-known s- to z-domain transformations, are mixed at best. The bilinear transform, when matched at DC or at 10 GHz, demonstrates poor magnitude and phase response characteristics, although it performs very well when the entire discrete frequency ω was pre-warped. The matched-Z transform demonstrates excellent phase response characteristics, but only fair magnitude response characteristics. It was extremely difficult to apply, however, with the primary reason being that the conversion of $H_a(s)$ from partial fraction expansion form to pole-zero factored form was difficult. The impulse invariant transform (IIT), somewhat surprisingly, demonstrated both good phase response and good magnitude response characteristics when compared to that for $H_a(s)$. Unfortunately, the IIT is also known to inherently map the zeros incorrectly from a frequency response perspective, which is bothersome. Finally, it was observed that it was notably easier to work directly with the partial fraction expansion form of $H_a(s)$, regardless of which transform was invoked.

From these results it is clearly desirable to find improved, practical methods for transforming transfer functions for the s-domain to the z-domain. Ideally, these transforms would be directly applicable to the pole-residue form of $H_a(s)$, have excellent frequency response characteristics from both a phase and magnitude perspective, be easily factorable so that the pole-zero map can be studied in the z-plane, and exhibit excellent correlation in the time-domain response. Knowing the limitations imposed by the time-frequency uncertainty principle [50], the approach to finding these transforms will be to use ideas and observations noted in the previous chapter and build upon the results presented there.

7.3.1 Modified Matched-Z Transform for Evaluation of $H_d(z)$ of High Order

It has been observed above that, while the overall function $H_a(s)$ can be difficult to convert from partial fraction expansion form to factored form for high-order functions, working with the 1st-order factors of $H_a(s)$ when doing transformations is quite intuitive. It would be useful to create a method for re-assembling the 1st-order converted functions in the z-domain back into an overall function $H_d(z)$ that could be assessed using pole-zero analysis. Further, it is desirable to use the matched-Z transform to convert the 1st-order factors from the s- to z-domain, in order to take advantage of the straightforward mapping approach, to avoid the pre-warping requirement of the bilinear transform, and to avoid the incorrect mapping of zeros that is inherent to the impulse invariant transform.

The first step in applying such a modified matched-Z method is to create the 1st-order partial fraction terms, in accordance with (7.5). This was implemented with a simple MATLAB function which uses c_k and a_k of (7.5) as the function inputs, and $B_{dk}(z)$ and $A_{dk}(z)$ of (7.5) as outputs. Assuming that the gain parameter of $H_{dk_{MZT}}(z)$ is matched to that of $H_{ak}(s)$ at the DC point, i.e., where $s = 0$ and $z = 1$, the modified matched-Z transform (MMZT) assumes the following form:

$$H_{d_{MMZT}}(z) = \sum_{k=1}^{N} H_{dk_{MZT}}(z) = \sum_{k=1}^{N} \frac{-\frac{c_k}{2a_k}(1 - e^{a_k T_s})(z+1)}{z - e^{a_k T_s}} \ . \tag{7.8}$$

It should be noted that this modified matched-Z transform, while having the same poles as the traditional matched-Z transform, will not have the same zeros. The modified matched-Z transform is thus not identical to the traditional matched-Z transform.

To complete the implementation of the MMZT, a routine is needed to perform the sum of the individual 1st-order $H_{dk_{MZT}}(z)$ factors, such that the function can be assembled into factored form to enable pole-zero analysis in the z-plane. This can be done with an iterative procedure to build up the numerator polynomial using successive concatenations of polynomial multiplication, as follows. Rewriting (7.8) in terms of its numerator and denominator polynomials and introducing a new constant $\tilde{c}_k$, gives

$$H_{d_{MMZT}}(z) = \sum_{k=1}^{N} H_{dk}(z) = \sum_{k=1}^{N} \frac{\tilde{c}_k(z+1)}{z - e^{a_k T_s}} = \sum_{k=1}^{N} \frac{B_{dk}(z)}{A_{dk}(z)}, \tag{7.9}$$

where

$$\tilde{c}_k = -\frac{c_k}{2a_k}(1 - e^{a_k T_s})\ , \tag{7.10}$$

and the MZT subscript on the 1st-order partial fraction terms $H_{dk}(z)$ has been dropped, since the process is understood from this point to be based on the matched-Z transform. It can be seen by examining (7.9) and (7.10) that the coefficients of the polynomials $B_{dk}(z)$ and $A_{dk}(z)$ can easily be calculated from the poles and residues of $H_a(s)$, and thus the 1st-order terms $H_{dk}(z)$ terms can be formed.

The iterative procedure for calculating $H_{d_{MMZT}}(z)$ can now be executed as follows. Starting with

$$H_{d_{MMZT}}(z) = \sum_{k=1}^{N} H_{dk}(z) = H_{d1}(z) + H_{d2}(z) + H_{d3}(z) + \cdots, \tag{7.11}$$

note that a representation based on two 1st-order terms would be:

$$H_{d12}(z) = H_{d1}(z) + H_{d2}(z) = \frac{B_{d1}(z)}{A_{d1}(z)} + \frac{B_{d2}(z)}{A_{d2}(z)} \ . \tag{7.12}$$

Combining the 1st-order partial fraction expansion terms, this becomes

$$H_{d12}(z) = \frac{B_{d1}(z)}{A_{d1}(z)} + \frac{B_{d2}(z)}{A_{d2}(z)} = \frac{B_{d1}(z)A_{d2}(z) + B_{d2}(z)A_{d1}(z)}{A_{d1}(z)A_{d2}(z)} = \frac{B_{d12}(z)}{A_{d12}(z)}, \tag{7.13}$$

where new numerator and denominator polynomials $B_{d_{12}}(z)$ and $A_{d_{12}}(z)$ have been calculated. Continuing the procedure by adding the next 1st-order partial fraction term,

$$H_{d123}(z) = H_{d12} + H_{d3}(z) = \frac{B_{d12}(z)}{A_{d12}(z)} + \frac{B_{d3}(z)}{A_{d3}(z)} \ , \tag{7.14}$$

and again combining the partial fraction terms, this becomes the 2nd-order term

$$H_{d123}(z) = \frac{B_{d12}(z)}{A_{d12}(z)} + \frac{B_{d3}(z)}{A_{d3}(z)} = \frac{B_{d12}(z)A_{d3}(z) + B_{d3}(z)A_{d12}(z)}{A_{d12}(z)A_{d3}(z)} . \tag{7.15}$$

Obviously the process can be continued until the function $H_{d_{MMZ}}(z)$ has been fully calculated and expanded into pole-zero factored form. This recursive, iterative procedure, along with the direct use of the traditional matched-Z transformation to map the 1st-order partial fraction expansion terms, form the basis for a new, modified matched-Z transform (MMZT).

7.3.1.1 Assessment of Modified Matched-Z Transform Results Using Pole-Zero Analysis

Applying the modified matched-Z transform to the 38th-order transfer function $H_a(s)$ for the 4th-order test circuit results in the pole-zero map depicted in Figure 7-4, using $T_s = 40$ ps. Note that the function $H_d(z)$ has 38 poles and 38 zeros, by definition, and that all are finite. As discussed previously, in the s-plane it is likely that there would be one or more zeros at infinity. Note that the poles all fall inside the unit circle, but they are close to the unit circle boundary, suggesting the function could go unstable without some care. Note also the alternating positions of the poles and zeros as the circle is traversed. This corresponds to the 19 local minima and maxima seen in the original frequency response curve $H_a(j\Omega)$ in the preceding chapters.

If the pole-zero map of Figure 7-4 is carefully examined, it can be seen that there are only 37 zeros visible, instead of the expected 38. It turns out that there is another zero located at $z \approx -235.8$, which is somewhat inconvenient, since a stated objective is to be able to view the entire pole-zero map on the same plot. Nevertheless, a method for reliably generating $H_d(z)$ from the partial fraction expansion form of $H_a(s)$ has been established that will yield usable pole-zero plots in the z-plane.

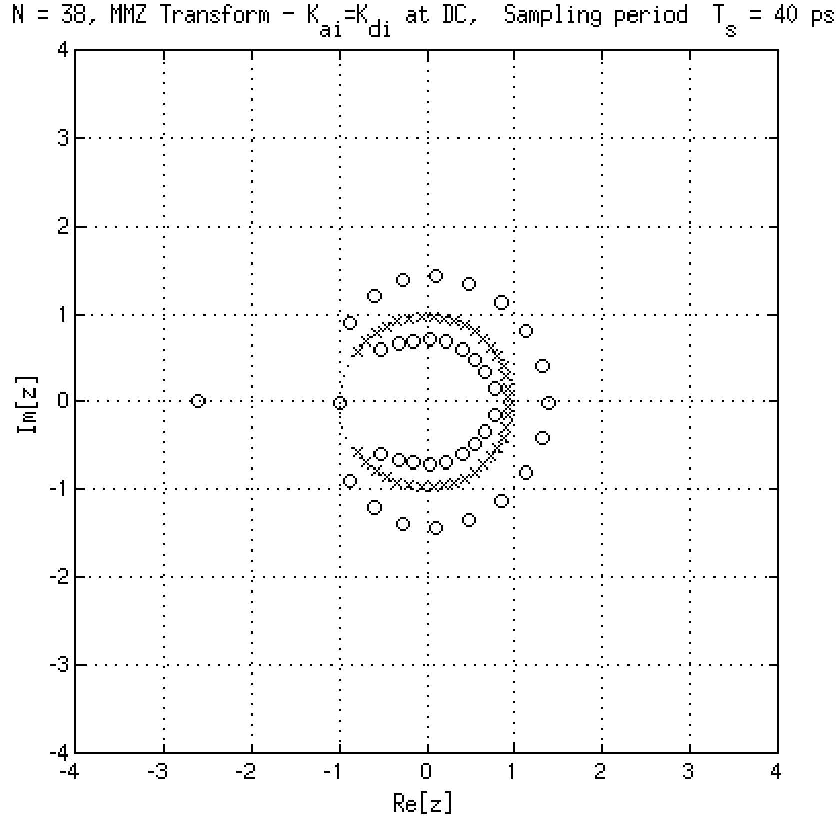


Figure 7-4: Pole-Zero Plot in the z-Plane for *$H_d(z)$* for the 4^{th}-Order Test Circuit Obtained Using the Modifed Matched-Z Transform with *K_d=K_a* at DC and *T_s=40* ps

A complete assessment and comparison of the z-domain system transfer functions given by (7.8) using the new modified matched-Z transform (MMZT) requires a close examination of the frequency and time-domain responses, in addition to the pole-zero maps examined above. The following sections will assess these responses for the case of the modified matched-Z transform, as applied to the system of the 4^{th}-order test circuit. Note that the time-frequency uncertainty principle introduced in Chapter 3 applies here, and that there are tradeoffs involved when choosing how to assess the model performance. Specifically, it is not possible to have low uncertainty in the time domain and the frequency domain responses simultaneously [50].

7.3.1.2 Frequency Response of Traditional Matched-Z and Modified Matched-Z Transforms' Transfer Functions

The frequency responses $H_a(j\Omega)$ and $H_d(e^{j\omega})$ associated with the system transfer functions $H_a(s)$ and $H_d(z)$, respectively, are comprised of both the magnitude response and the phase response. Examination of these responses enables an exact assessment of the system's pole and zero locations and lends additional insight into how the associated system model will ultimately perform when used for waveform prediction purposes [42], [49]. In the previous section, the transfer function $H_d(z)$ was generated in the complex frequency domain using a modified matched-Z transform, and the pole-zero map was plotted in Figure 7-4. From this analysis, it was seen that the there was variation in the resulting zero placement for the modified matched-Z transform (MMZT) when compared to the traditional matched-Z transform (MZT) utilized in Chapter 6, but the system was still seen to be stable, as indicated by the fact that all of the poles were located inside the unit circle in the z plane. The implications of the variations of the zero positions from the new transform are, as of yet, unclear. Beyond these important generalizations, however, it was difficult to assess more exact implications on the system's time and frequency responses using only the pole-zero map assessment.

Figure 7-5 shows a comparison of the frequency response $H_a(j\Omega)$ versus the frequency responses $H_d(e^{j\omega})$ calculated using modified matched-Z, traditional matched-Z, and impulse invariant transforms. Note that neither magnitude response of $H_d(e^{j\omega})$ obtained using the modified or traditional matched-Z transforms correlates perfectly with the response $H_a(j\Omega)$, but that the phase responses of $H_d(e^{j\omega})$ both align very nicely with that for $H_a(j\Omega)$. Interestingly, both the magnitude and phase responses of $H_d(e^{j\omega})$ found using the impulse invariant transform show excellent correlation with $H_a(j\Omega)$.

It is clear from the examination of Figure 7-5 that both the modified and traditional matched-Z transforms have significant deviations from $H_a(s)$ in their magnitude responses, apparently due to the presence of a distinct zero at the Nyquist frequency $f_N = 12.5$ GHz, which corresponds to the point $z = -1$ on the pole-zero plot of Figure 7-4. The excellent correlation in the phase responses $H_d(e^{j\omega})$ to that of $H_a(j\Omega)$ suggests that all three transforms, i.e., the modified matched-Z, the traditional matched-Z, and the impulse invariant, are all well-suited for systems exhibiting propagation delay. Furthermore, all three transforms appear to be superior to the bilinear transformation in their phase response characteristics, unless the complex process involving the pre-warping of the entire frequency variable ω is invoked.

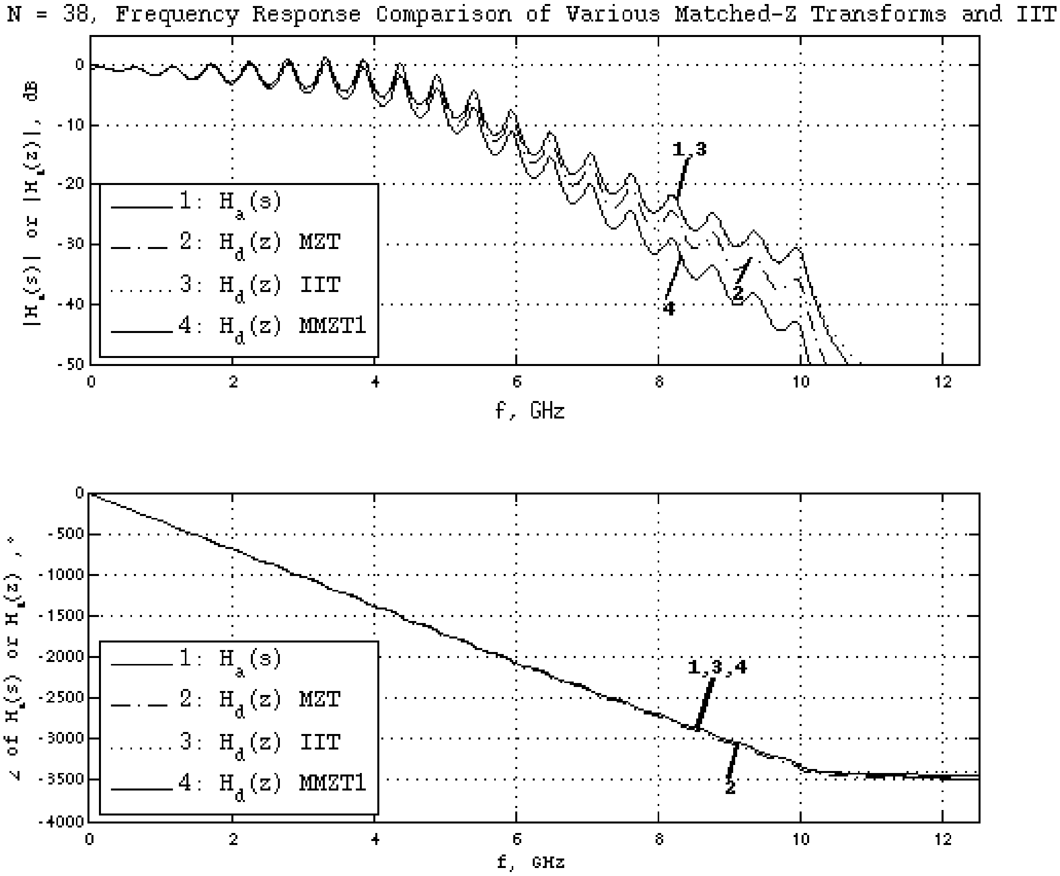


Figure 7-5: Frequency Response *Hd(z)* for the 4th-Order Test Circuit Using the Matched-Z Transform, Impulse Invariant Transform, and MMZT1 Transforms[17]

Finally, there was a clear advantage to using the modified matched-Z transform over the traditional matched-Z transform, in that the MMZT could be applied directly to the pole-residue form of $H_a(s)$, and the troublesome conversion step to pole-zero factored form in the s-domain was avoided entirely, moving this conversion step to the z-domain, where the numerical magnitudes are easier to handle. The matching of the

---

[17] MMZT1 stands for the modified matched-Z transform with matching of the 1st-order sections' analog gain terms K_{ak} to the corresponding digital gain terms K_{dk} at frequency $s = 0$ (DC).

magnitude response of the MMZT was inferior to that of the MZT, however, and the presence of the MMZT's far outlying zero at approximately $z \approx -235.8$ is still troubling.

7.3.1.3 Time Domain Response of Modified Matched-Z Transform's Transfer Function

Recall from Chapter 6 that the ultimate objective with a system transfer function or frequency transfer function model is often to use the model for calculating the time-domain response to a known stimulus [93]. In this regard, it would be interesting to use the system transfer function $H_d(z)$, derived from $H_a(s)$ in the previous sections using the new modified matched-Z transform (MMZT), to calculate the time responses of the system.

Recall again from Chapter 6 that the transfer functions $H_a(s)$ or $H_d(z)$ can be used to calculate the time responses [40]. By writing differential equations or difference equations, respectively, a state matrix representation can be formulated, allowing direct calculation of the time responses [48], [49]. The calculation of the time responses was handled directly using known MATLAB functions timeresp.m [61] for $H_a(s)$ and lsim.m for $H_d(z)$ [80]. Recall also from Chapter 6 the method for assessing the time responses calculated from $H_d(z)$ by comparing them directly to the time response obtained using $H_a(s)$. Recall also that the time response calculated from $H_a(s)$ was assessed previously versus that obtained from SPICE in Chapter 5, and determined to overlap almost perfectly. This correlation, along with qualitative evaluation of the frequency response, helped to generate confidence that $H_a(s)$ was providing a reasonable representation of the system, even though a system of fairly high order $N = 38$ was required.

Figure 7-6 shows the transient response obtained using the transfer function $H_a(s)$ obtained from vector fitting in Chapter 5, compared to the transfer function $H_d(z)$ represented by the pole-zero map in Figure 7-4, and obtained using the new modified matched-Z transformation (MMZT). Note that the transient response calculated using $H_d(z)$ shows very good correlation with the response generated directly from $H_a(s)$, and that this correlation includes signal edge-rate characteristics, amplitude characteristics, and propagation delay characteristics. Again noting from Figure 7-4 that the modified matched-Z transform maps the poles in an identical manner to that of the traditional matched-Z transform, and that the zero positions are mapped differently with outlier zeros, it is apparent that this discrepancy in the zero positions doesn't seem to affect the transient response significantly.

Recalling the frequency response plot from Chapter 6 for the traditional matched-Z transform and from Figure 7-5 for the MMZT, it is seen that both the traditional matched-Z transform and the modified matched-Z transform result in $H_d(z)$ functions with good phase characteristics. Any deviations in system propagation delay characteristics are also difficult to detect, as seen in Figure 7-6, at least using manual visual comparisons. The phase-response correlation to that for $H_a(s)$ overrides the mismatches seen in the magnitude responses for both the MMZT and MZT transforms.

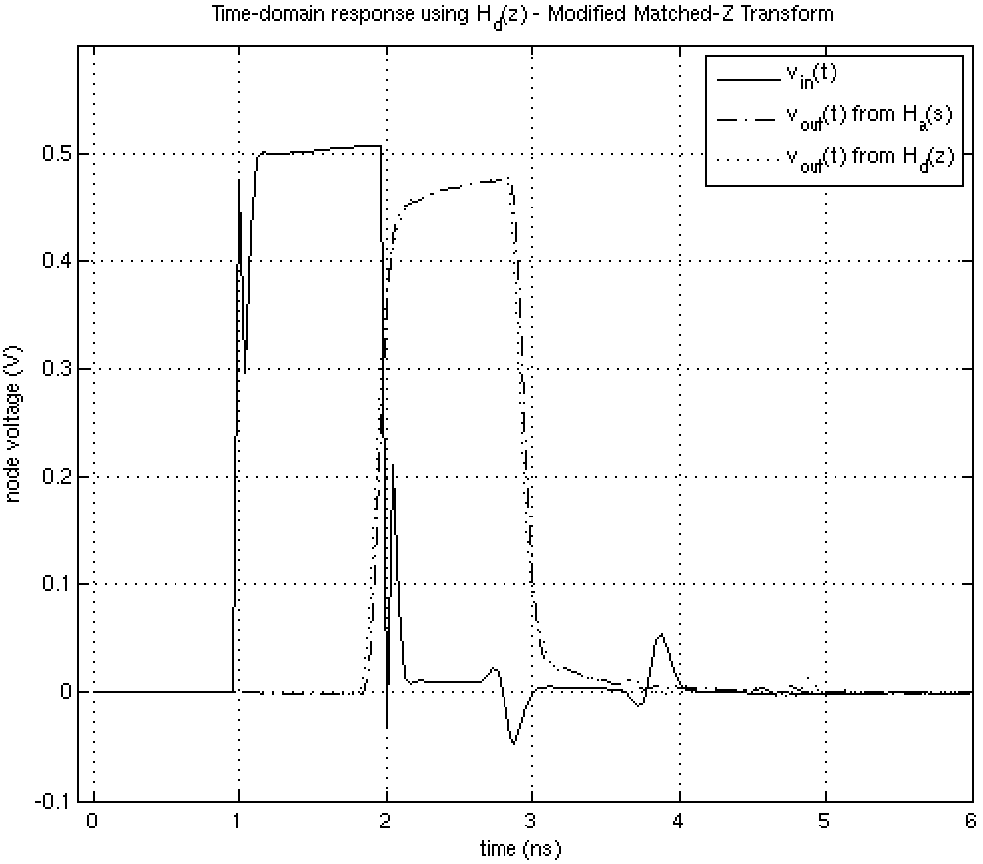


Figure 7-6: Time Response Calculated Using *$H_d(z)$* for the 4th-Order Test Circuit Obtained from the Modified Matched-Z Transform with *K_d=K_a* at DC

7.3.2 Modified Bilinear Transform Method for Evaluation of $H_d(z)$ of High Order

The principal ideas behind the development of the MMZT in the previous section, namely, that it is desirable to work directly with the 1st-order sections of $H_a(s)$, and that it is easier to convert from pole-residue form to pole-zero factored form in the z-domain than it is in the s-domain, lead to the notion that it may be possible to improve upon the bilinear transform results of Chapter 6 as well. One of the important steps in the MMZT was that the gain-matching process was applied to each 1st-order section $H_{ak}(s)$ individually via (7.5), as opposed to matching the gains K_a and K_d of the overall transfer functions $H_a(s)$ and $H_d(z)$, respectively, at DC. Similarly, it is known that the bilinear transform magnitude response can be matched at a particular frequency using the pre-warping process of [42] and demonstrated explicitly in Chapter 6.

When this pre-warping process was applied to the overall transfer function $H_d(z)$ in Chapter 6, both at DC and at $f = 10$ GHz, the results were extremely poor, both for the phase and magnitude responses, and this translated to a poor result for the time response as well. However, the experience of the previous section for matching the gains separately for each individual 1st-order section for the MMZT leads to the notion that a better result may be obtained if the pre-warping process is applied individually to each 1st-order section $H_{ak}(s)$. Then, the resulting 1st-order sections in the z-domain $H_{dk}(z)$ can be subsequently recombined to form $H_d(z)$, just as was done for the MMZT transform. Thus the 1st-order section $H_{dk_{BLT}}(z)$ of (7.4) becomes

$$H_{dk_{MBLT}}(z) = \left[H_{ak}(s)|_{s=\frac{s}{k_{effk}}} \right]_{s=\frac{2\,(z-1)}{T_s(z+1)}}, \tag{7.16}$$

where

$$k_{eff_k} = \frac{\Omega_{warp_k}}{\Omega_{o_k}} = \frac{\frac{2}{T_s} \tan \frac{\Omega_{o_k} T_s}{2}}{\Omega_{o_k}}, \tag{7.17}$$

and where Ω_{o_k} is the analog radian frequency to be matched to for the k^{th} 1st-order section. To select this match frequency, recall the analog 1st-order sections $H_{ak}(s)$ have the form

$$H_{ak}(s) = \frac{c_k}{s - a_k}\,, \tag{7.18}$$

and note that, if the analog pole locations are further defined as

$$a_k = a_{Rk} + j a_{Ik}\,, \tag{7.19}$$

then the pole frequencies Ω_{o_k} to be matched are selected from a_k in the s-plane as

$$\Omega_{o_k} = a_{Ik}\,. \tag{7.20}$$

Applying this modified bilinear transform (MBLT) results in multiple 1st-order sections $H_{dk_{MBLT}}(z)$ that can then be recombined from a partial fraction expansion form to pole-zero factored form, using the same process as that used for the MMZT, as outlined in (7.11) through (7.15) in the previous section.

Figure 7-7 summarizes the process for implementing the modified bilinear transform outlined above. **At the time just prior to the publication of this dissertation, this new modified bilinear transform (MBLT) had been under consideration for patent protection. It will now be submitted for publication as an Institute of Electrical and Electronic Engineers (IEEE) peer-reviewed journal article** [95]**.**[18]

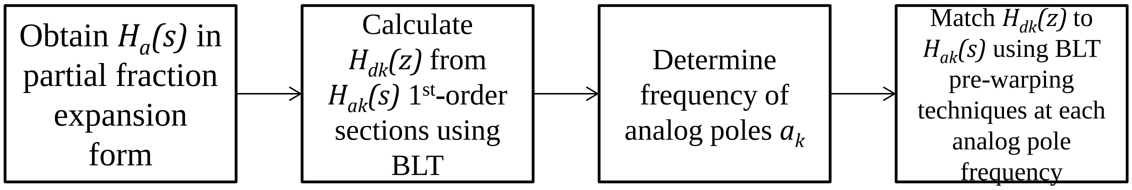


Figure 7-7: Summary of Process for Implementing the Modified Bilinear Transform (MBLT)

7.3.2.1 Assessment of Modified Bilinear Transform Results Using Pole-Zero Analysis

Figure 7-8 shows the pole-zero map for the function $H_d(z)$ for the 4th-order test circuit, as derived using this modified bilinear transform, again for a sampling period $T_s = 40$ ps. Note the considerable improvement of both the spacing of the poles and zeros, as well as their position and magnitude, as compared to the results in Chapter 6 for the bilinear transform when no pre-warping is used, or pre-warping is applied to the

---

[18] For information regarding the final disposition of the modified bilinear transform's (MBLT's) publication status, please contact the author after 31 July 2014. Contact information is listed in Appendix 2.

single critical frequency $f = 10$ GHz. It can be seen that the MBLT's pole and zero locations and spacing also compare favorably to the maps in Chapter 6 for the bilinear transform where the entire discrete frequency variable ω was pre-warped, for the matched-Z transform, and for the impulse invariant transform.

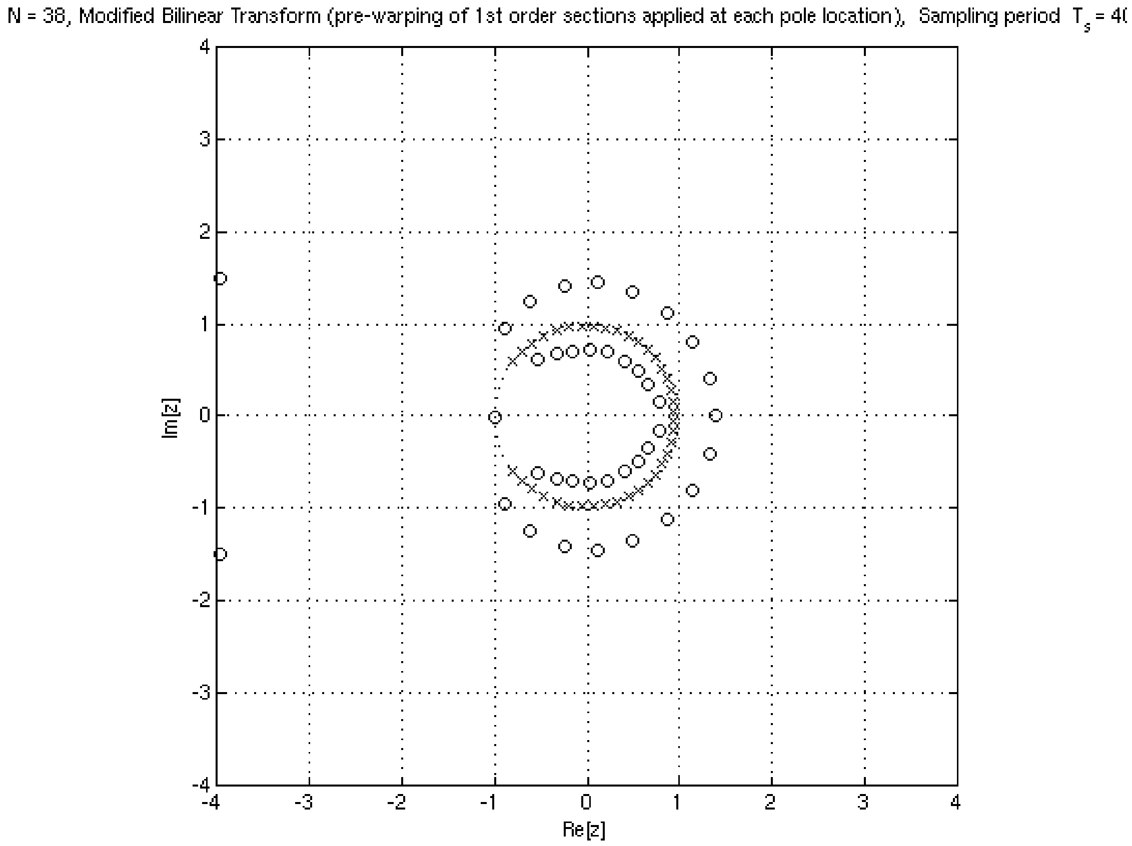


Figure 7-8: Pole-Zero Plot in the z-Plane for *$H_d(z)$* for the 4th-Order Test Circuit Obtained Using the Modified Bilinear Transform with *T_s=40* ps

Note also the comparison to the pole-zero plot of Figure 7-4 for the MMZT. Similar to the MMZT plot, there are a couple of outlying zeros, although for the MBLT the situation is better. The outliers are seen to be located at $z \approx -3.96 \pm 1.49$, and are at least visible on a reasonably-scaled pole-zero map in the z-plane, unlike the case for the MMZT in the previous section.

Finally, as has been the case for all of the s- to z-domain transforms, a complete assessment and comparison of the z-domain system transfer functions obtained using the new modified matched-Z transform requires a close examination of the frequency and transient responses, in addition to the pole-zero maps examined in the previous section.

7.3.2.2 Frequency Response of the Modified Bilinear Transform Transfer Function

In the previous section, the transfer function $H_d(z)$ was generated in the complex frequency domain using the modified bilinear transform (MBLT), and the pole-zero map was plotted in Figure 7-8. From this analysis, it was seen that the there was tremendous improvement in the resulting pole placement over the traditional bilinear transform, which was seen in Chapter 6 to perform poorly when the gain of $H_d(z)$ was matched only at DC or the frequency $f = 10$ GHz. There was variation in the resulting zero placement for the modified bilinear transform when compared to the modified matched-Z transform, but the system was seen to be stable, as indicated by the fact that all of the poles were located inside the unit circle in the z plane. The implications of the variations of the zero positions from the new transform are, as of yet, not totally clear. Beyond these important generalizations, however, it is difficult to assess more exact implications to the system's time and frequency responses using only the pole-zero map assessment.

Figure 7-9 shows a comparison of the original analog frequency response $H_a(j\Omega)$ versus the discrete frequency response $H_d(e^{j\omega})$ calculated using the modified bilinear transform. Note that the magnitude response of $H_d(e^{j\omega})$ obtained using the modified bilinear transform now matches the response $H_a(j\Omega)$ at each pole's frequency location. This is the condition that was forced by matching the gain of each individual 1st-order section $H_{ak}(s)$ in (7.16) and (7.17) at the analog pole frequencies a_k, and this condition seems to carry through when recombining the 1st-order sections in the overall transfer function $H_a(s)$. Perhaps more importantly, the phase response of $H_d(e^{j\omega})$ obtained with the modified bilinear transform now aligns very nicely with that for $H_a(j\Omega)$, something that was not true for the bilinear transform before when matched to only DC or to $f = 10$ GHz in Chapter 6.

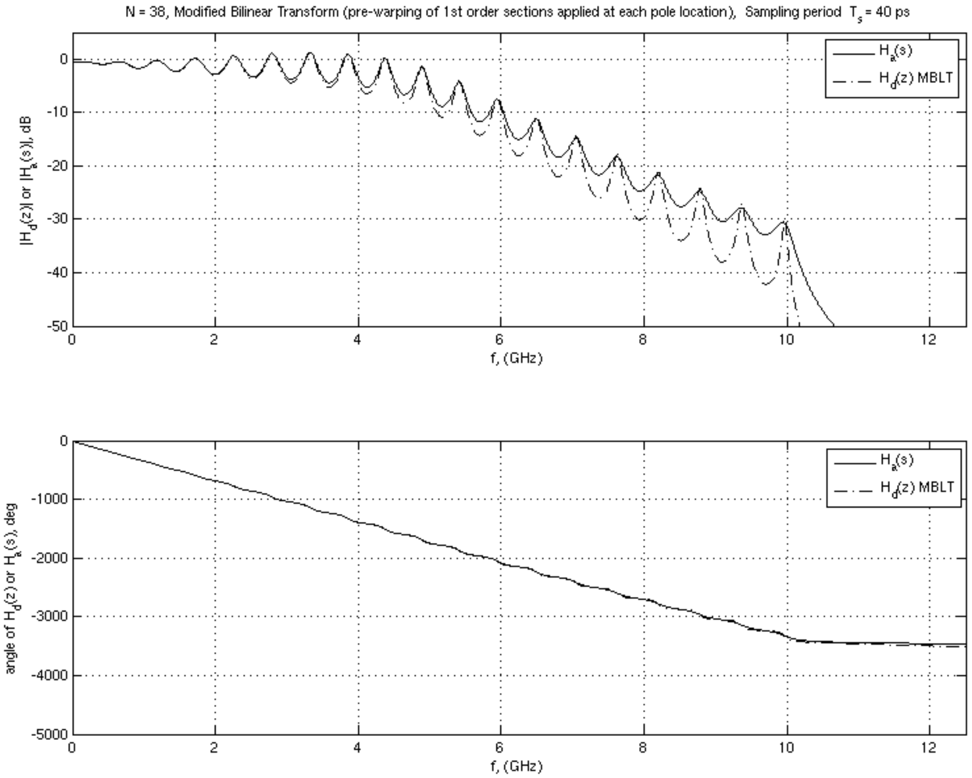


Figure 7-9: Frequency Response of *$H_d(z)$* for the 4th-Order Test Circuit Obtained Using the Modified Bilinear Transform with *T_s=40* ps

The net result is that the modified bilinear transform shows reasonable correlation in the magnitude response, and excellent correlation in the phase response, to the analog frequency domain response $H_a(j\Omega)$. This is a tremendous improvement over application of the bilinear transform utilizing no pre-warping, or when pre-warping at only a single critical frequency. The matching of the magnitude response is perhaps not quite as good as when using the bilinear transform as implemented using the Ikai method, but the MBLT has one advantage over that method in that the MBLT transform operates directly on the vector fitted function $H_a(s)$, and does not require the vector fitting step to be applied to a pre-warped version of $H_a(s)$.

It is clear from the examination of Figure 7-9 that $H_d(z)$ obtained with the modified bilinear transform still has significant deviations from $H_a(s)$ in its magnitude responses, although $H_d(z)$ now matches the $H_a(s)$ at every pole location, as desired. The excellent correlation now found in the phase response $H_d(e^{j\omega})$ to that of $H_a(j\Omega)$ is a tremendous improvement, and suggests that the modified bilinear transform may be better suited for systems exhibiting propagation delay. Clearly, the new transform has huge advantages over the bilinear transform matched at only a single frequency, which was basically unusable in the forms demonstrated in Chapter 6. Finally, the modified bilinear transform shares an advantage with both the modified matched-Z transform (MMZT) and the impulse invariant transform (IIT), in that the MBLT can also be applied directly to the pole-residue form of $H_a(s)$. Accordingly, the troublesome conversion step to pole-zero factored form in the s-domain is avoided entirely, moving this conversion step to the z-domain, where the numerical magnitudes are easier to handle.

7.3.2.3 Time Domain Response of Modified Bilinear Transform Transfer Function

Figure 7-10 shows the transient response obtained from the analog transfer function $H_a(s)$ obtained from vector fitting in Chapter 5 and the discrete transfer function $H_d(z)$ represented by the pole-zero map in Figure 7-8, obtained using the new modified bilinear transformation (MBLT). Note that the transient response calculated using $H_d(z)$ shows improved correlation with the response generated directly from $H_a(s)$, particularly in the rising and falling edges, the overall propagation delay characteristics, and the general amplitude characteristics. There is ripple in the time response calculated by the modified bilinear transform's transfer function, however. It can also be seen that the MBLT's time response is likely non-causal, as is indicated by the fact that the rising edge and falling edges of the MBLT response occur slightly ahead of the analog transfer function's time response calculation [63].

The results shown in Figure 7-10 further demonstrate that the bilinear transform modifications incorporated in the MBLT, namely, the matching of the frequency responses $H_{dk}(e^{j\omega})$ of the 1st-order sections of $H_d(z)$ to $H_{ak}(f)$ at the corresponding pole frequencies for each analog section, make the MBLT a viable option for converting analog transfer functions in the s-domain to digital transfer functions in the z-domain. In fact, the tremendous improvement in the performance of the MBLT over the BLT using either no pre-warping or pre-warping at only a single critical frequency suggests that a similar modification to the MMZT's gain matching strategy might yield a similar improvement.

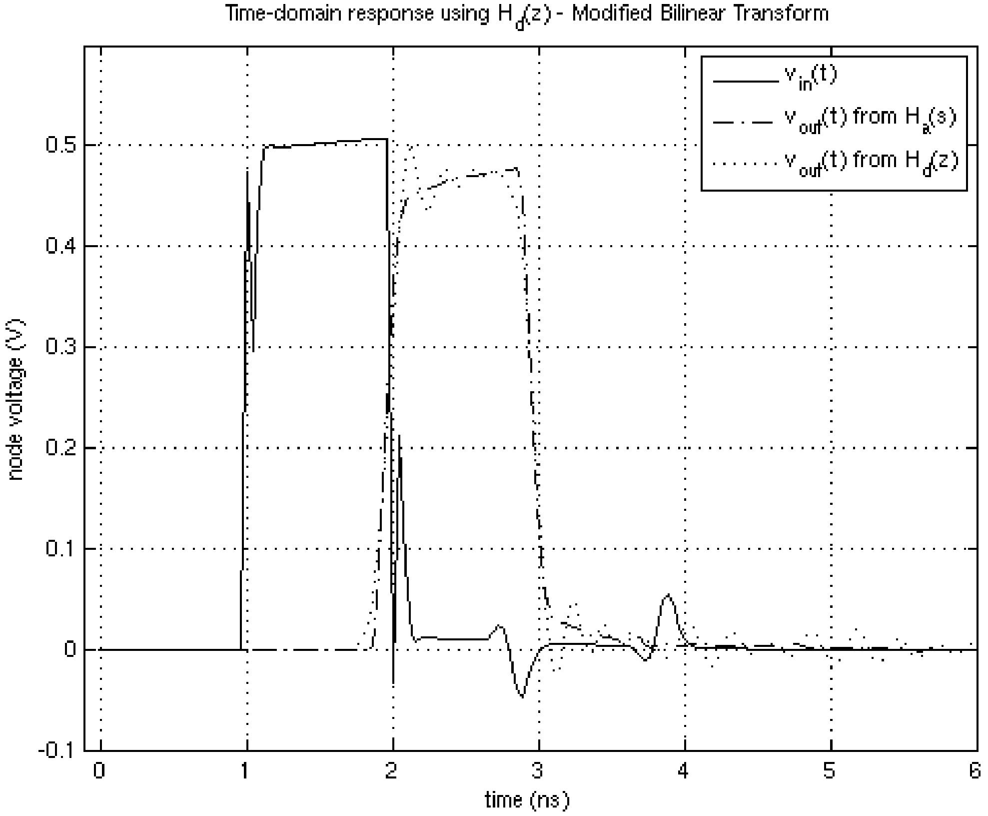


Figure 7-10: Time Response Calculated Using $H_d(z)$ for the 4th-order Test Circuit Obtained from the Modified Bilinear Transform with $T_s=40$ ps

7.3.3 Modified Matched-Z Transform for Evaluation of $H_d(z)$ with Enhancements to Gain Matching of 1st-Order Sections

It was noted in Figure 7-5 that the modified matched-Z transform had excellent matching of the phase response of $H_d(z)$ to the phase response of $H_a(s)$, but that the magnitude response of $H_d(z)$ fell off from that of $H_a(s)$ increasingly rapidly at high frequencies. Recall that the modified matched-Z transform matched the gain K_d of $H_d(z)$ to the gain K_a of $H_a(s)$ at the DC point where $s = 0$ and $z = 1$. There were also significant deviations in the positions of the zeros of $H_d(z)$ for the MMZT, including a bothersome outlier zero very far from the origin on the negative real axis, as compared to the zero positions of $H_d(z)$ found when using the traditional matched-Z transform.

The tremendous improvement in the performance of the MBLT over the BLT suggests that a similar modification to the MMZT algorithm, i.e., matching K_{dk} to K_{ak} of the 1st-order sections $H_{dk}(z)$ and $H_{ak}(s)$, respectively, at each individual pole frequency of $H_{ak}(s)$, might yield a similar improvement. Specifically, it would be desirable to improve the matching of the magnitude response of $H_d(z)$ to that of $H_a(s)$, and to improve the position of the zeros relative the positions found by the traditional matched-Z transform (MZT).

To implement this modification, the 1st-order section $H_{dk}(z)$ with form

$$H_{dk}(z) = \frac{B_{dk}(z)}{A_{dk}(z)} = \frac{b_{dk}(z+1)}{z - e^{a_k T_s}}, \tag{7.21}$$

is again paired with the 1st-order section $H_{ak}(s)$ with form

$$H_{ak}(s) = \frac{c_k}{s - a_k}. \tag{7.22}$$

The condition for gain matching is now changed from the previous derivation of the MMZT, where the gains were matched at DC, i.e., where $s = 0$ and $z = 1$, to a new gain condition where the gains are now matched differently at the radian frequency of the analog pole a_k for each analog 1st-order section. More specifically, if the analog pole location is specified as

$$a_k = a_{Rk} + ja_{Ik}\,, \tag{7.23}$$

then the pole frequency to be matched is selected from a_k in the s-plane as

$$s = jIm[a_k] = ja_{Ik} = j2\pi f_k\,, \tag{7.24}$$

and this is matched in the z-plane to the location

$$z = e^{ja_{Ik}T_s} \tag{7.25}$$

in the z-plane. Thus, the new gain matching condition for each 1st-order section of the MMZT can be expressed mathematically as

$$H_{ak}(s)|_{s=ja_{Ik}} = H_{dk}(z)\big|_{z=e^{ja_{Ik}T_s}}, \tag{7.26}$$

which leads to the final expression for the MMZT:

$$H_{d_{MMZT}}(z) = \sum_{k=1}^{N} H_{dk_{MMZT}}(z) = \sum_{k=1}^{N} \frac{\dfrac{c_k(e^{ja_{Ik}T_s} - e^{a_k T_s})}{(ja_{Ik} - a_k)(e^{ja_{Ik}T_s} + 1)}(z+1)}{z - e^{a_k T_s}}\,. \tag{7.27}$$

The expression of (7.27) can be directly compared to the previous expression for the MMZT in (7.8), for which the MMZT gain matching was performed at DC for each 1st-order section. Applying this revised modified matched-Z transform again results in multiple 1st-order sections $H_{dk_{MMZT}}(z)$, that can also then be recombined from the partial fraction expansion sum form of (7.27) to pole-zero factored form, using the same process as that used for the MMZT and MBLT, as outlined in (7.11) through (7.15) previously. Expression of $H_{d_{MMZT}}(z)$ in pole-zero factored form again enables the creation of the pole-zero map in the z-plane.

Figure 7-11 summarizes the process for implanting the modified matched-Z transform outlined above.

At the time just prior to the publication of this dissertation, this new modified matched-Z transform (MMZT) had been under consideration for patent protection. It will now be submitted for publication as an Institute of Electrical and Electronic Engineers (IEEE) peer-reviewed journal article [95]**.**[19]

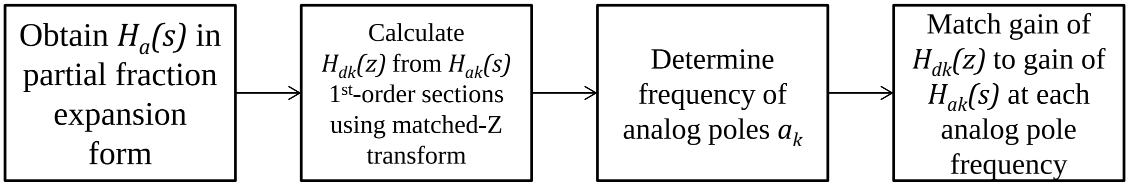


Figure 7-11: Summary of Process for Implementing the Modified Matched-Z Transform (MMZT)

7.3.3.1 Assessment of Modified Matched-Z Transform Results with 1st-Order Gain Matching Transfer Function Using Pole-Zero Analysis

Applying the modified matched-Z transform of (7.27) to the 38th-order transfer function $H_a(s)$ for the 4th-order test circuit, this time with gain matching at the pole frequencies of each 1st-order section $H_{ak}(s)$ instead of at DC, results in the pole-zero map depicted in Figure 7-12, using $T_s = 40$ ps. Note that the function $H_d(z)$ also has 38 poles and 38 clearly-visible zeros, and that all zeros are now easily viewable on the plot.

---

[19] For information regarding the final disposition of the modified matched-Z transform's (MMZT's) publication status, please contact the author after 31 July 2014. Contact information is listed in Appendix 2.

The outlier zeros have been eliminated. As discussed previously, in the s-plane it is likely that there would be at least one zero at infinity.

Note that the poles all fall inside the unit circle, although they are close to the unit circuit boundary suggesting the function could go unstable without some care. Note also the alternating positions of the poles and zeros as the circle is traversed. This corresponds to the 19 local minima and maxima seen in the original magnitude response of $H_a(j\Omega)$.

A close comparison of the pole-zero maps of Figure 7-4, which represented the MMZT with gain matching at DC, and Figure 7-12, which represents the MMZT with gain matching at the individual pole frequencies, shows that the pole locations are identical, as expected, but that the zero positions are different. This should be expected since the residual values have changed between the expressions of (7.8) and (7.27). Aside from these subtle differences in the zero positions, it is difficult to compare the effect of the change in the gain matching condition between these two forms of the MMZT. It will be necessary to continue the comparison by assessing the performance of the function $H_d(z)$ by looking at the frequency response and the time response.

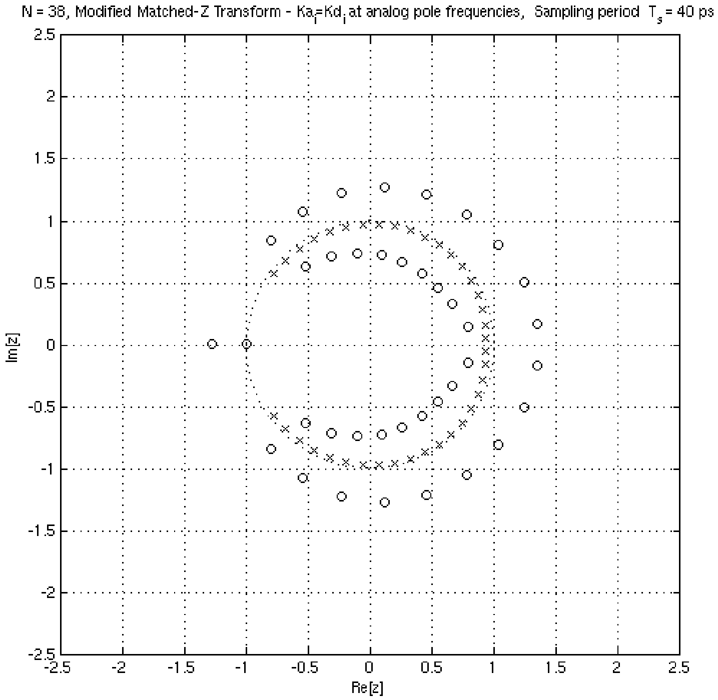


Figure 7-12: Pole-Zero Plot in the z-Plane for *$H_d(z)$* for the 4th-Order Test Circuit Obtained Using the Modified Matched-Z Transform with *K_{dk}=K_{ak}* at Individual Pole Frequencies with *T_s=40* ps

7.3.3.2 Frequency Response of Modified Matched-Z Transform Transfer Function with 1st-Order Gain Matching

Figure 7-13 shows a comparison of the original analog frequency response $H_a(j\Omega)$ versus the discrete frequency response $H_d(e^{j\omega})$ calculated using the modified matched-Z transform of (7.27), where the gains of the 1st-order sections $H_{dk}(z)$ are now matched at the pole frequencies of each 1st-order section $H_{ak}(s)$, instead of at DC. Note that the magnitude response of $H_d(e^{j\omega})$ obtained using the new MMZT now matches the response $H_a(j\Omega)$ extremely well, performing just as well as the impulse invariant transform and bilinear transform using the Ikai method, as shown in Chapter 6. The condition that was forced by matching the gain of each individual 1st-order section $H_{ak}(s)$ seems to carry through remarkably well when recombining the 1st-order sections of the overall transfer function $H_a(s)$.

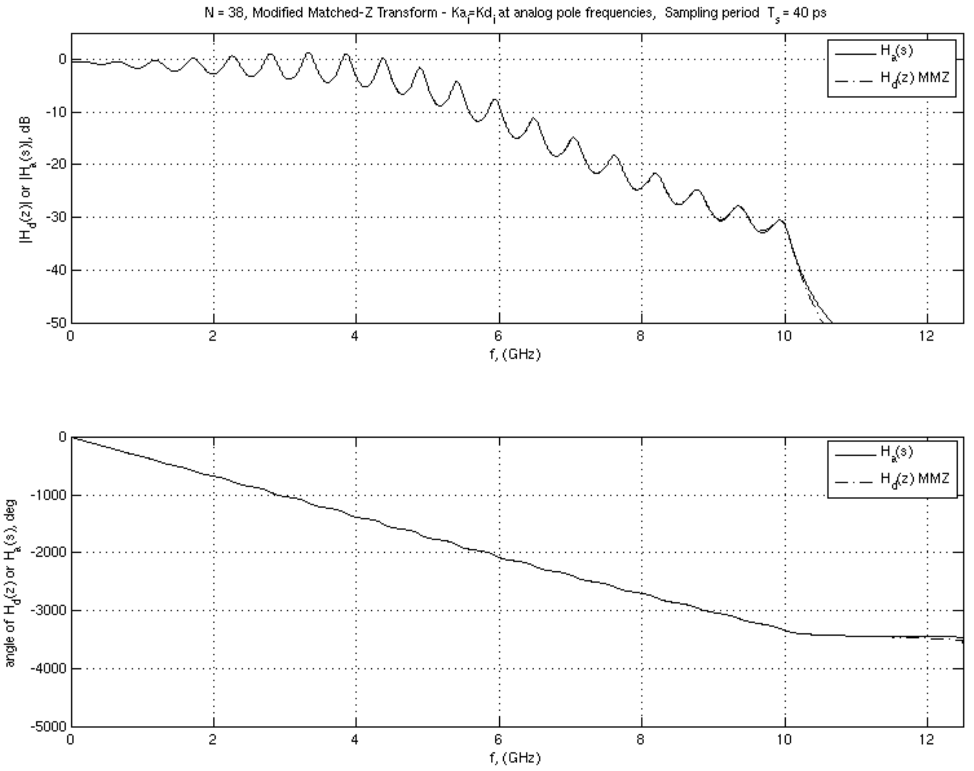


Figure 7-13: Frequency Response of $H_d(z)$ for the 4th-Order Test Circuit Obtained Using the Modified Matched-Z Transform with $K_{dk}=K_{ak}$ at Individual Pole Frequencies

Perhaps just as importantly, the phase response of $H_d(e^{j\omega})$ obtained with this improved version of the MMZT maintains its excellent matching characteristics to that for $H_a(j\Omega)$, something that was also true for the modified matched-Z transform even when gains were matched at DC for each 1st-order section.. The net result is that this new version of the MMZ shows greatly improved correlation in the magnitude response, maintains excellent correlation in the phase response, and now rivals the performance of the IIT and BLT using the Ikai method [91] when matching to the analog frequency domain response $H_a(j\Omega)$.

This revised form of the MMZT is a tremendous improvement over the traditional matched-Z transform, the bilinear transforms applied either with no pre-warping or pre-warping at only a single critical frequency, the modified bilinear transform, and the previous version of the modified matched-Z transform with gain matching of 1st-order sections at DC. In addition, unlike the BLT using the Ikai method [91], the MMZT can be applied directly to the vector fitted version of $H_a(s)$, and not to a pre-warped version of the s-domain transfer function.

7.3.3.3 Time Domain Response for Modified Matched-Z Transform with 1st-Order Gain Matching Model

Figure 7-14 shows the transient response obtained from the analog transfer function $H_a(s)$ obtained from vector fitting in Chapter 5, and the discrete transfer function $H_d(z)$ represented by the pole-zero map in Figure 7-12 obtained using the new modified matched-Z transformation (MMZT), in this case with the gain matching applied at the corresponding pole frequency for each analog 1st-order section $H_{ak}(s)$. Note that the transient response calculated using $H_d(z)$ shows excellent correlation with the response generated directly from $H_a(s)$, just as it did for the case of the MMZT with the gain matching applied at DC. Recall from Figure 7-4 and Figure 7-12 that the modified matched-Z transform with gain-matching at the individual pole frequencies maps the poles in an identical manner to that of the modified matched-Z transform with gain matching at DC, although the zero positions are mapped differently, due to the fact that the residues in (7.8) and (7.27) are different. Both versions of the modified matched-Z transform result in $H_d(z)$ functions with excellent phase characteristics. As such, any deviations in system propagation delay characteristics are also difficult to detect, as seen in Figure 7-14, at least using manual visual comparisons.

When comparing to the MBLT, it is clear that the MMZT with gain matching applied at 1st-order sections is superior at matching the time response characteristics, and it does not display the non-causal behavior of the MBLT, either. The net result is that this new version of the MMZT displays equivalent time-domain performance to the impulse invariant transform, traditional matched-Z, BLT using the Ikai method, and modified matched-Z transform with DC gain matching. It has superior time-domain performance the BLT when using no pre-warping or pre-warping at only a single frequency, and superior time-domain response performance to that of the MBLT.

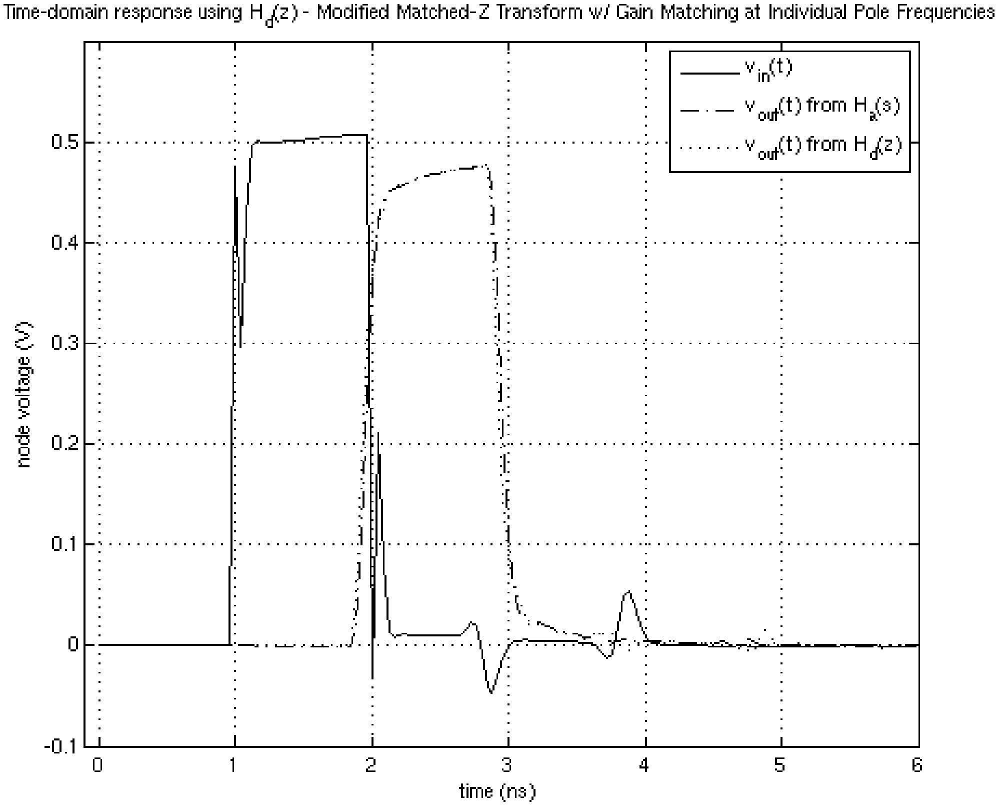


Figure 7-14: Time Response Calculated Using $H_d(z)$ for the 4th-Order Test Circuit Obtained from the Modified Matched-Z Transform with $K_{dk}=K_{ak}$ at Individual Pole Frequencies

7.3.4 Additional Comments Regarding the Transform of 1st-order Sections of $H_a(s)$ to 1st-Order Sections of $H_d(z)$

It should be re-emphasized here that applying the impulse invariant transform to the 1st-order sections of $H_a(s)$ represents no change from the normal application of that method, as there is nothing inherent in the process for applying the transform that precludes it. On the other hand, for the modified bilinear transform, the process of applying the pre-warping process to each individual 1st-order section is a novel technique. Similarly, for the modified matched-Z transforms introduced in this section, both the application of the matched-Z transform to the individual 1st-order sections $H_{ak}(s)$, as opposed to $H_a(s)$ itself, and the technique of matching the gain parameter at the individual pole frequencies of the 1st-order sections $H_{ak}(s)$ are both novel methods.

The remarkable improvement shown in the pole-zero mapping locations, the frequency response matching, and the time-response matching for the MBLT enables the pragmatic use of the very familiar and popular BLT for signal integrity applications, whereas before the BLT was shown to be basically unusable unless the Ikai method was invoked, with its disadvantage that it must be applied to a pre-warped version of $H_a(s)$. Similarly, the remarkable improvement in the magnitude response for the MMZT when imposing the gain matching at the pole frequencies of the analog 1st-order sections, along with its stable time response performance and known advantages over the IIT in transforming the zero locations, make this version of the MMZT the preferred technique for performing $H_a(s)$ to $H_d(z)$ transformations.

Finally, the new MMZT transform introduced in this chapter was seen to handle the 1st-order terms of the partial fraction expansion form of $H_a(s)$ equally as well as the BLT and IIT. It shared the advantages of mapping the infinite zeros encountered in the s-plane to a known location in the z-plane. That zero location of $z = -1$ was the same as that given by the bilinear transform. It mapped the poles in an equivalent manner to the IIT, the BLT using the Ikai method, and the MBLT. The MMZT mapping was seen to be superior to the pole mapping given by the BLT for the cases of no pre-warping and pre-warping at only a single critical frequency. Furthermore, even though the zeros of the resulting, recombined function $H_d(z)$ are different than those given more directly by the traditional matched-Z transform, this ambiguity in zero placement is better than that given inherently by IIT.

7.4 Conclusions Regarding the Practical Considerations of Working with $H_d(z)$ Models and $H_a(s)$ to $H_d(z)$ Transformation Methods

The results of Chapter 6 and Chapter 7 indicate that it is possible to gain significant insight into the nature of the system by looking at the pole-zero map of $H_d(z)$, and by looking at the magnitude and phase response characteristics of $H_d(e^{j\omega})$, while simultaneously maintaining an understanding of the resulting time-domain response characteristics. It was emphasized that all three of these assessments are necessary to maintain a complete understanding of the system.

In addition, it was clearly seen that the new modified matched-Z transform, when applied using the gain matching condition at each 1st-order section's analog pole frequency, has advantages over each of the three most commonly-known methods for transforming between the s- and z-domains: 1) it maps the zeros more precisely than the impulse invariant transform; 2) it transforms from the s- to z-domain more seamlessly than the traditional matched-Z transform and the bilinear transform using the Ikai method, due to the ability of working directly with the partial fraction expansion form of $H_a(s)$, albeit with some resulting differences in the resulting zero locations in $H_d(z)$; and 3) it has superior frequency response matching characteristics to that of the modified bilinear transform.

Even with these successful results, some problems and challenges remain. Frequency response waveforms $H_a(j\Omega)$ with complicated profiles lead to rational function approximations $H_a(s)$ with high orders, such as the 38th-order function obtained for the system of the 4th-order test circuit. Each of the known transformation methods leads to a different set of zero locations in the z-plane. Finally, a full set of methods for making use of z-domain analysis needs to be developed in order to fully realize the advantages associated with the z-plane.

It remains to be seen whether the differences in the zero placement between the various transforms are due mainly to the different applications of the zero mapping approach, or due more to the numerical uncertainty issues associated with the process of converting from pole-residue form to the pole-zero factored form. The next chapter will seek clarification on several of these issues, and introduce some techniques for optimizing the system and gaining further insight by studying the pole-zero maps in the z-plane.

Summary of Modified Bilinear Transform (MBLT) Characteristics

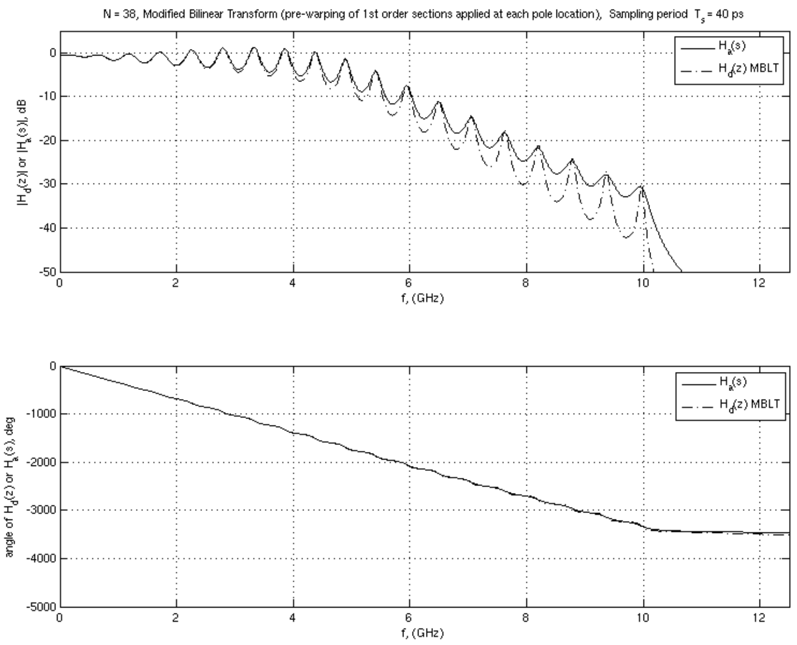


Figure 7-15: MBLT Frequency Response Comparison for the 4th-Order Test Circuit [20,21]

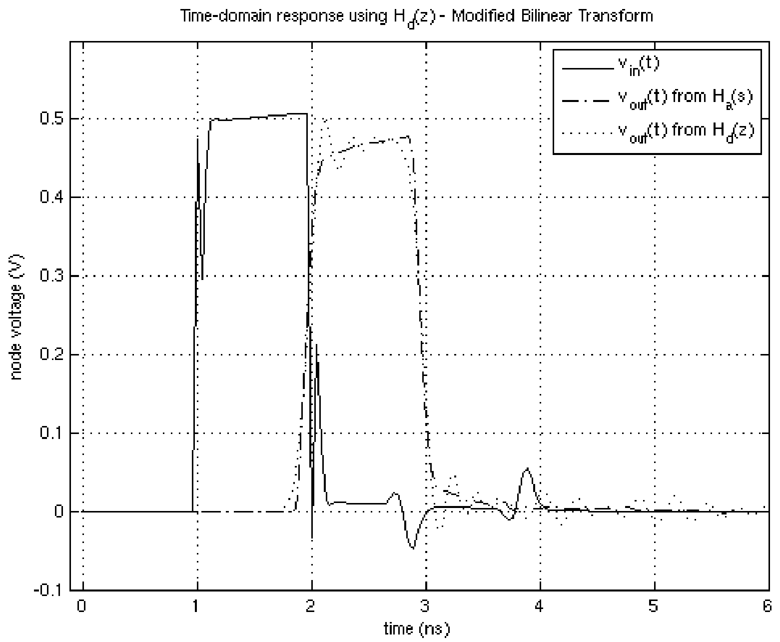


Figure 7-16: MBLT Time Response Comparison for the 4th-Order Test Circuit

---

[20] For the 4th-Order Test Circuit plots on this page, the derived transfer function $H_d(z)$ has order $N = 38$ and a sampling interval of $T_s = 40$ ps.

[21] At the time just prior to the publication of this dissertation, the new modified bilinear transform (MBLT) had been considered for patent protection, and will now be submitted for publication as an IEEE peer-reviewed journal article [95].

Summary of Modified Matched-Z Transform (MMZT) Characteristics

(for the case where the gains of the 1st-Order Sections are Matched at the Analog Pole Frequencies)

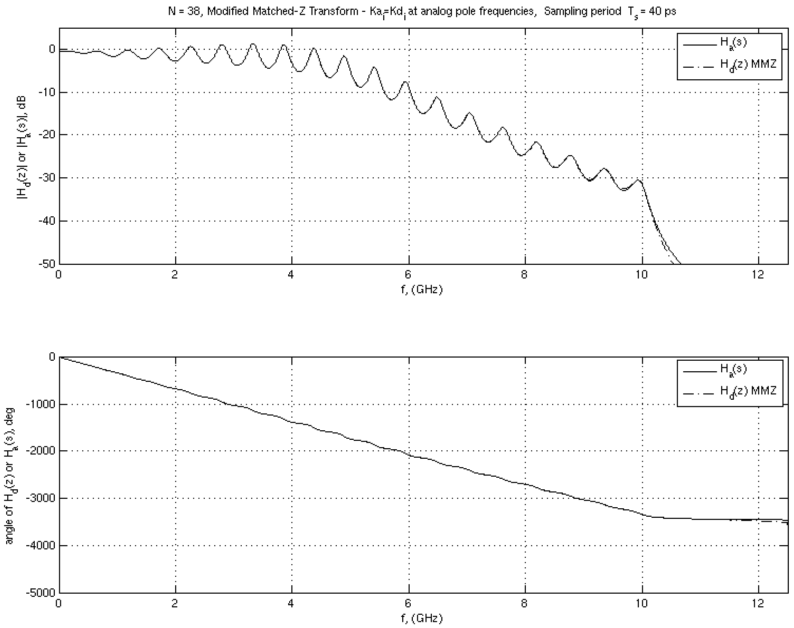


Figure 7-17: MMZT Frequency Response Comparison for the 4th-Order Test Circuit [22][23]

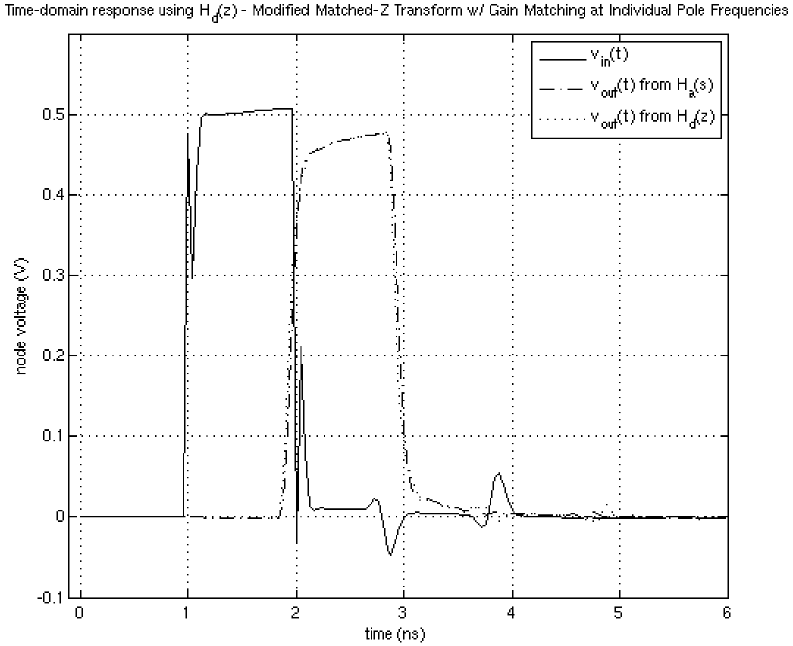


Figure 7-18: MMZT Time Response Comparison for the 4th-Order Test Circuit

---

[22] For the 4th-Order Test Circuit plots on this page, the derived transfer function $H_d(z)$ has order $N = 38$ and a sampling interval of $T_s = 40$ ps.

[23] At the time just prior to the publication of this dissertation, the new modified matched-Z transform (MMZT) had been considered for patent protection, and will now be submitted for publication as an IEEE peer-reviewed journal article [95].

Addendum 7A: MATLAB Computer Program Listing for Modified Matched-Z Transform with Gain Matching at DC (*s = 0* and *z = 1*)

```
Function    BAzk = matchz0(Bsk,Ask,Ts)

%MATCHZ0    Matched z-transformation of 1st-order transfer function
%           Hk(s) = Bsk/(s - Ask) into Hk(z) which is of the form
%
%                   bzk*(z + 1)          Bzk(z)
%           Hk(z) = ----------------  = ---------, where k = 1:N
%                   z - exp(Ask*Ts)      Azk(z)
%
%           where
%           bzk = Bsk*(1-exp(Ask*Ts)/(-2*Ask)
%
%           and two 1st-order polynomial coefficient vectors are
%
%           Bzk = [bzk, bzk], Azk = [1, -exp(Ask*Ts)]
%
%           INPUTS
%           Bsk  =  residue coefficients for Hk(s) (Nx1)
%           Ask  =  pole coefficients for Hk(s) (Nx1)
%           Ts  =  sampling period [seconds]
%
%           OUTPUTS
%           Bzk = numerator: 1st-order polynomial vector (Nx2)
%           Azk = denominator: 1st-order polynomial vector (Nx2)
%           BAzk = [Bzk,Azk]  (Nx4)
%
%           SYNTAX
%           BAzk = matchz0(Bsk,Ask,Ts);
%
%           HHO 7-3-12, latest revision PED 8-03-13

% Gain coefficient  bzk (Nx1)
bzk = -0.5*Bsk.*(1-exp(Ask*Ts))./Ask;
% Numerator: 1st-order polynomial vector Bzk (Nx2)
Bzk = [bzk,bzk];
% Denominator: 1st-order polynomial vector Azk (Nx2)
Azk = ones(length(Ask),2);
Azk(:,2) = -exp(Ask*Ts);
BAzk = [Bzk,Azk];
```

Addendum 7B: MATLAB Computer Program Listing for Modified Bilinear Transform with Gain Matching at Analog Pole Frequencies of $H_{ak}(s)$

```
Function    BAbk = bilinxfrm0(Bsk,Ask,Ts)

%BILINXFRM0 Bilinear transformation of 1st-order transfer function
%           Hk(s) = Bsk/(s - Ask) into Hk(z) which is of the form
%
%                 bzk*(z + 1)              Bzk(z)
%     Hk(z) = -----------------------= ---------, where k = 1:N
%               z - (2+Ask*Ts)/(2-Ask*Ts)    Azk(z)
%
%           where
%           bzk = (Ts*Bsk)/(2-Ask*Ts)
%
%           and two 1st-order polynomial coefficient vectors are
%
%           Bzk = [bzk, bzk], Azk = [1, -(2+Ask*Ts)/(2-Ask*Ts)]
%
%           INPUTS
%           Bsk  =  residue coefficients for Hk(s) (Nx1)
%           Ask  =  pole coefficients for Hk(s) (Nx1)
%           Ts  =  sampling period [seconds]
%
%           OUTPUTS
%           Bzk = numerator: 1st-order polynomial vector (Nx2)
%           Azk = denominator: 1st-order polynomial vector (Nx2)
%           BAbk = [Bzk,Azk]  (Nx4)
%
%           SYNTAX
%           BAzk = blinxfrm0(Bsk,Ask,Ts);
%
%           PED 6-19-13, latest revision  6-19-13

% Gain coefficient  bzk (Nx1)
bzk = (Ts*Bsk)./(2-(Ask*Ts));
% Numerator: 1st-order polynomial vector Bzk (Nx2)
Bzk = [bzk,bzk];
% Denominator: 1st-order polynomial vector Azk (Nx2)
Azk = ones(length(Ask),2);
Azk(:,2) = -(2+Ask*Ts)./(2-Ask*Ts);
BAbk = [Bzk,Azk];
```

Addendum 7C: MATLAB Computer Program Listing for Modified Matched-Z Transform with Gain Matching at Analog Pole Frequencies of $H_{ak}(s)$

```
function    BAzk = matchz0_1st(Bsk,Ask,Ts)

%MATCHZ1ST  Matched z-transformation of 1st-order transfer
%           function
%           Hk(s) = Bsk/(s - Ask) into Hk(z) which is of the form
%
%                      bzk*(z + 1)       Bzk(z)
%           Hk(z) = ---------------  = ---------, where k = 1:N
%                   z - exp(Ask*Ts)      Azk(z)
%
%           where
%                       Bsk*(exp(j*beta_sk*Ts)-exp(Ask*Ts)
%                 bzk = -------------------------------------
%                      (j*beta_sk – Ask)( exp(j*beta_sk*Ts)+1)
%
%           and two 1st-order polynomial coefficient vectors are
%
%                 Bzk = [bzk, bzk], Azk = [1, -exp(Ask*Ts)]
%
%           INPUTS
%              Bsk  =  residue coefficients for Hk(s) (Nx1)
%              Ask  =  pole coefficients for Hk(s) (Nx1)
%              Ts  =  sampling period [seconds]
%
%           OUTPUTS
%              Bzk = numerator: 1st-order polynomial vector (Nx2)
%              Azk = denominator: 1st-order polynomial vector (Nx2)
%              BAzk = [Bzk,Azk]  (Nx4)
%
%           SYNTAX
%              BAzk = matchz0(Bsk,Ask,Ts);
%
%              PED     10-10-13, latest revision 10-10-13

% Gain coefficient  bzk (Nx1)
beta_sk=imag(Ask);
% bzk = -0.5*Bsk.*(1-exp(Ask*Ts))./Ask; % This matches gain at DC.
bzk = (exp(j.*beta_sk.*Ts)-
exp(Ask*Ts))*Bsk./((exp(j.*beta_sk.*Ts)+1).*(j.*beta_sk-Ask)); %
This matched gain at the pole's imaginary frequency.
% Numerator: 1st-order polynomial vector Bzk (Nx2)
Bzk = [bzk,bzk];
% Denominator: 1st-order polynomial vector Azk (Nx2)
Azk = ones(length(Ask),2);
Azk(:,2) = -exp(Ask*Ts);
BAzk = [Bzk,Azk];
```

Addendum 7D: MATLAB Computer Program Listing for Impulse Invariant Transform

```
function    BAik = iimxfrm0(Bsk,Ask,Ts)

% IIMXFRM0  Impulse invariant transformation of 1st-order
%           transfer function Hk(s) = Bsk/(s - Ask) into Hk(z)
which %           is of the form
%
%                    bzk*( z )            Bzk(z)
%           Hk(z) = ----------------  = ---------, where k = 1:N
%                    z - exp(Ask*Ts)      Azk(z)
%
%           where
%                    bzk = Bsk*Ts
%
%           and two 1st-order polynomial coefficient vectors are
%
%                       Bzk = [bzk, 0], Azk = [1, -exp(Ask*Ts)]
%
%            INPUTS
%               Bsk  =  residue coefficient for Hk(s) (Nx1)
%               Ask  =  pole coefficients for Hk(s) (Nx1)
%               Ts  =  sampling period [seconds]
%
%            OUTPUTS
%               Bzk = numerator: 1st-order polynomial vector (Nx2)
%               Azk = denominator: 1st-order polynomial vector (Nx2)
%               BAbk = [Bzk,Azk]  (Nx4)
%
%            SYNTAX
%               BAik = iimxfrm0(Bsk,Ask,Ts);
%
%               PED     6-19-13, latest revision  6-19-13

% Gain coefficient  bzk (Nx1)
bzk = (Ts*Bsk);
% Numerator: 1st-order polynomial vector Bzk (Nx2)
Bzk = [bzk,0];
% Denominator: 1st-order polynomial vector Azk (Nx2)
Azk = ones(length(Ask),2);
Azk(:,2) = -exp(Ask*Ts);
BAik = [Bzk,Azk];
```

Chapter 8: Examination and Optimization of $H_d(z)$ Utilizing z-Plane Techniques

8.1 Introduction

One of the primary advantages of examining the system transfer functions $H_a(s)$ in the s-plane and $H_d(z)$ in the z-plane is that some aspects of the system are readily apparent. System stability, causality, and passivity properties can be determined in both the s-plane and the z-plane, for example, just as they can be determined in the discrete- or continuous-time domains [63], [96]. Opportunities for model order reduction and simplification may follow from examination of the pole-zero map or, at a minimum, it may be possible to see which poles and zeros are likely to dominate the frequency response and which others may have a more limited effect, and whether the model may be close to becoming unstable, non-causal, or non-passive [63].

Thus, examination of the pole-zero map may provide another level of insight beyond what examination in only the time domain can normally provide. This insight may be into the sensitivity of the model, or the system that it represents, to slight variances in the model's construction or to the underlying physical construction of the system. The insight may be into the system's overall complexity, and the opportunities for reducing this complexity, without significant reductions in accuracy. Alternatively, the insight may lead to the realization that the model's resolution is limited in one domain or the other, due to limitations imposed by the time-frequency uncertainty principle [50].

In this chapter, the model for the 4th-order test circuit will be examined in the z-plane in more detail. This examination was started in Chapter 6 by transforming $H_a(s)$ to $H_d(z)$ using the three most widely-known s- to z-domain transformations, namely the bilinear transform (BLT), the impulse invariant transform (IIT), and the matched-Z transform (MZT) [41], [42], [48]. The examination was then continued in Chapter 7, where the model was converted from $H_a(s)$ to $H_d(z)$ using two forms of the new modified matched-Z transformation (MMZT) [95], which invoked two different strategies for gain matching between $H_a(s)$ to $H_d(z)$, and also the new modified bilinear transform (MBLT) [95]. The models were compared for all the transforms using pole-zero analysis in the z-plane, examination of the discrete frequency domain responses $H_d(e^{j\omega})$, and examination of the discrete time domain responses using state-space formulations of the newly-formed rational function approximations $H_d(z)$ to calculate the time domain response.

The analysis of the z-plane maps will now continue in more detail, first by introducing a new, enhanced method for graphical analysis in the z-plane, which allows for simultaneous assessment of the pole-zero map of $H_d(z)$ and the frequency response $H_d(e^{j\omega})$ on the same plot. Second, the assessment of pole-zero locations and their effect on the system characteristics, including the potential for model-order reduction utilizing pole-zero cancellation, will be considered. This will be done using an augmented rational function approximation for the 4th-order test circuit for which the model has been forced to a higher order. Finally, some considerations for optimizing the model's system identification step and s- to z-domain transformation step by selecting an appropriate frequency range, in concert with an appropriate choice of the sampling interval, will be discussed. In the end, the z-plane analysis techniques formalized in this and the preceding chapters will be incorporated into a comprehensive signal integrity methodology, to be formally introduced in Chapter 9.

8.2 Using the Frequency Response $H_d(e^{j\omega})$ to Predict Attributes of the Transfer Function $H_d(z)$

It was suggested in Chapter 5 that a simple, cursory examination of the frequency response $H_a(j\Omega)$ could lead to insight regarding the expected order of the analog transfer function $H_a(s)$ obtained using the vector fitting process. Specifically, it was noted that the presence of 19 valleys suggested 19 zero locations, and 19 peaks suggested the presence of 19 pole locations. The observation of the number of pole locations in itself leads to the conclusion that $H_a(s)$ would be well-represented as a 38th-order system, since the 19 peaks observed would cover half the frequency range, namely, the north half of the s-plane, and the pole-zero map would be symmetric on the south half of the s-plane. It was also seen in the various transformations of $H_a(s)$ to $H_d(z)$ undertaken in Chapters 6 and 7 that the z-domain transfer functions had the same overall system order as the original s-domain transfer function, where the order $H_a(s)$ is determined by the order of the denominator polynomial of the rational function $H_a(s)$ or $H_d(z)$ [48].

Taking this observation further, it was natural to conclude that, in addition to predicting the order of the rational function required to represent $H_a(s)$, it would be possible to predict at least the imaginary parts (i.e., the $j\Omega$ part of $s = \sigma + j\Omega$) of the pole and zero locations in the s-plane based on this examination of $H_a(j\Omega)$. This information, specifically the analog pole frequencies, was used in Chapter 7 to develop two new s-domain to z-domain transforms, the modified matched-Z transform (MMZT) and the modified bilinear transform (MBLT) [95]. Both of these transforms used the analog pole frequencies as points at which to match the gains of the various 1st-order sections comprising the transformed functions of $H_d(z)$, which resulted in great improvement over previous versions of the matched-Z and bilinear transforms. As an example, the pole-zero plot for the 4th-order test circuit, for the case where $N = 38$ and $T_s = 40\ ps$ for $H_d(z)$, as produced utilizing the modified matched-Z transform with gain matching at the individual analog pole frequencies, is reproduced from Chapter 7 and shown in Figure 8-1. The frequency response $H_d(e^{j\omega})$ corresponding to this $H_d(z)$ is also reproduced from Chapter 7 and shown in Figure 8-2.

Careful examination of the pole-zero map of Figure 8-1 and the frequency response of Figure 8-2 reveals the connection between the relative minima and maxima of the frequency response, and the pole and zero locations in the z-plane. For example, note on the pole-zero map that there are poles near the positive real axis, that is, at the point where $\omega = 0$. Notice that this corresponds to the $f = 0$ location on the frequency response plot, where it is seen to be a relative maximum. Following a similar thought process, it is seen that the angle on the pole-zero map where $\omega = \pi\ radians$ has two zeros, and that the corresponding point on the frequency response plot is $f = 12.5$ GHz, which corresponds to the Nyquist frequency f_N when the sampling interval is $T_s = 40$ ps. Obviously, while these relationships are easy to recognize for the cases for $f = 0,\ \omega = 0$ and $f = 12.5$ GHz, $\omega = \pi$ radians, they are more difficult to decipher for the other relative minima and maxima when having to switch back and forth between plots.

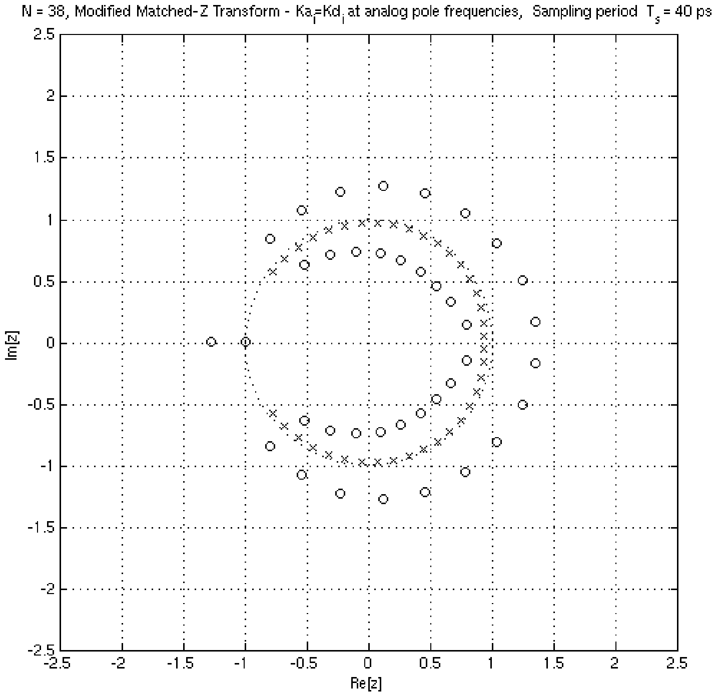


Figure 8-1: Pole-Zero Plot in the z-plane for $H_d(z)$ for the 4th-Order Test Circuit, Derived from the MMZT with Gain Matching at Analog Pole Frequencies, for $N = 38$ and $T_s = 40$ ps

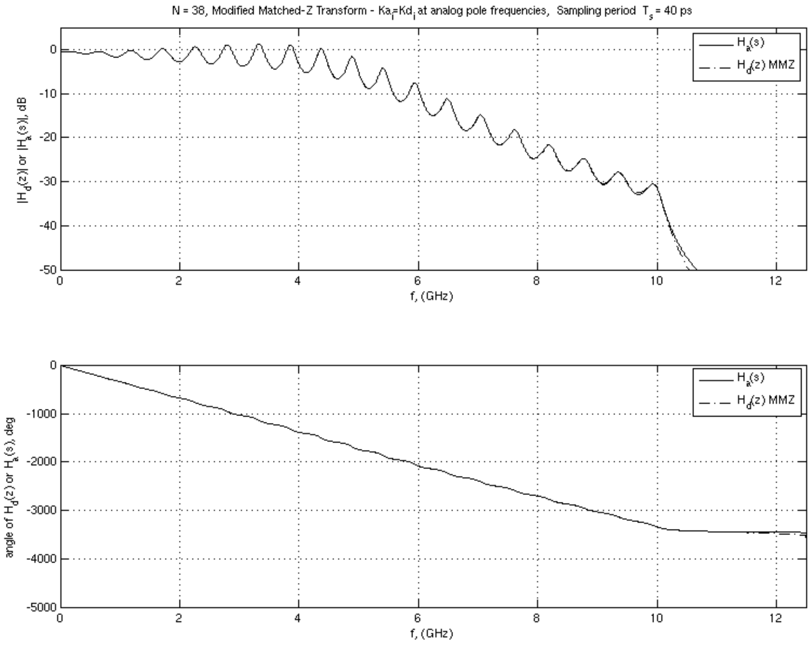


Figure 8-2: Frequency Response for $H_d(e^{j\omega})$ for the 4th-Order Test Circuit, Derived from the MMZT with Gain Matching at Analog Pole Frequencies, for for $N = 38$ and $T_s = 40$ ps

All of these observations suggest that it would be extremely useful to observe the pole-zero map of $H_d(z)$ and the frequency response of $H_d(e^{j\omega})$ simultaneously, to enable a direct assessment of the effects of poles and zeros in the z-plane on the expected frequency response, and to ascertain the relationship of the pole and zero discrete frequencies ω to the pole-zero positions. The converse would also be true, namely, examination of the frequency response of an unknown system in the discrete-frequency ω-domain could suggest the order and nature of the pole zero map. Either way, simultaneous examination of the system from both perspectives could lead to additional insight into the nature of the system.

8.3 Combined Pole-Zero and Frequency Response Graphical Plotting Technique

From the above discussion, it is clear that there may be inherent advantages to looking at the $H_d(z)$ transfer function's pole-zero map in the z-plane and the frequency response $H_d(e^{j\omega})$ simultaneously. Specifically, it would be advantageous to view them both in such a way that the quantitative relationship between them is also evident.

In this regard, it is useful to formally recall the relationship between the digital frequency ω and the coordinates in the z-plane. Specifically, the frequency response $H_d\left(e^{j\omega}\right)$ can be expressed in terms of the discrete transfer function $H_d(z)$ as [42], [43]:

$$H_d\left(e^{j\omega}\right) = H_d(z)|_{z=e^{j\omega}}. \tag{8.1}$$

More generally, z can be related to ω as:

$$z = re^{j\omega} = r\cdot\cos\omega + j\,r\cdot\sin\omega = Re[z] + j\,Im[z], \tag{8.2}$$

so it can be seen that

$$\omega = \tan^{-1}\frac{Im[z]}{Re[z]}. \tag{8.3}$$

Note that (8.3) is the analogous relationship to that noted in previously for the imaginary part of the complex variable s in the Laplace domain, namely:

$$\Omega = Im[s] = Im[\sigma + j\Omega]\ . \tag{8.4}$$

From this standpoint, it can be seen that the angle ω in the z-plane is simply the angle of the vector drawn from the origin to the z coordinate in question measured from the positive horizontal axis. Recalling that ω is also the abscissa coordinate of a frequency response plot, both for the magnitude response and the phase response, it is clear that the frequency response could be drawn on an ω-axis drawn concentrically around the unit circle of the z plane, on the same plot. In this case, ω would represent the angular coordinate, and either the magnitude or phase response of $H_d(e^{j\omega})$, or both, could represent the radial coordinate(s). Doing this for the magnitude response, for example, would establish a clear visual connection between the location of the peaks and valleys of the frequency response in relation to the pole and zero locations in the z-plane.

Figure 8-3 shows such a figure for the case of the pole-zero map of $H_d(z)$ and the magnitude response of $H_d(e^{j\omega})$ for the 4th-order test circuit, obtained using the new modified matched-Z transform introduced previously in Chapter 7, for the case where the gains are matched at the analog pole frequencies of the 1st-order sections of $H_a(s)$. It combines the pole-zero plot of $H_d(z)$ of Figure 8-1 and the magnitude response $|H_d(e^{j\omega})|$ portion of Figure 8-2 into a single graphical plot.

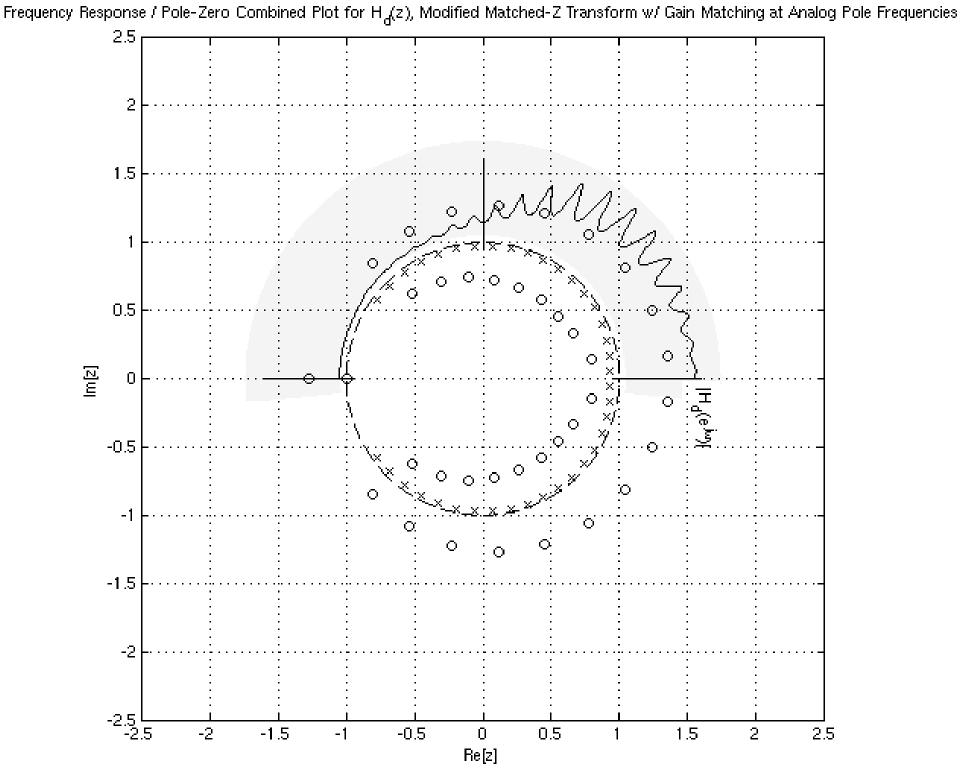


Figure 8-3: Combined Frequency Response and Pole-Zero Plot for *$H_d(z)$* for the MMZT with Gain Matching at Analog Pole Frequencies

The shaded region in the north half of the z-plane represents the area of the plot containing the frequency response information. Recall from Chapter 3 that in the z-plane, the angle where $\omega = 0$ corresponds to DC, and the angle where $\omega = \pi$ corresponds to the Nyquist frequency f_N. It should be emphasized that, even though the frequency response is confined to the north half plane, this is only done for convenience. It is understood that the frequency magnitude response $\left|H_d(e^{j\omega})\right|$ is symmetric about the $\omega = 0$ axis [41], so it could easily be reproduced in the south half of the z-plane with symmetry around the horizontal axis, at the expense of cluttering the plot.

A detailed examination of the combined plot reveals that the relative maxima (peaks) and relative minima (valleys) of the magnitude response curve do indeed correspond to the presence of poles and zeros, respectively. Specifically, it can be seen that the peak or valley in the magnitude response occurs at the same angle ω as the corresponding pole or zero, and thus the direct link between the pole-zero map and the frequency response has been established. It should be pointed out that the plot of Figure 8-3 contains only the magnitude response of $H_d(e^{j\omega})$, not the phase response. It would also be possible to create a combined plot of the pole-zero map and the phase response of $H_d(e^{j\omega})$ or, alternatively, a combined plot containing both responses. Clearly, the choice of which options to include on the pole-zero map would be dictated from a practical standpoint by which specific aspects of $H_a(s)$ and $H_d(e^{j\omega})$ are under consideration.

Several other characteristics of the 4th-order test circuit system can be seen in the combined plot of Figure 8-3. One interesting characteristic is that the zeros alternate between positions outside and inside the unit circle as ω increases. It was already noted in Figure 8-2 that the system exhibits nearly linear phase response characteristics, and that fact can be seen here in that a well-known characteristic of linear phase FIR systems is that they have reciprocal zeros on the inside and outside of the unit circle [81]. While not quite perfectly linear phase, as seen by the phase responses depicted in Figure 8-2, it can be seen that the general trend holds for the IIR 4th-order test circuit system.

To complement Figure 8-3, Figure 8-4 through Figure 8-10 show the combined pole-zero map and frequency magnitude response plot for the system transfer functions $H_d(z)$ and frequency transfer functions $H_d(e^{j\omega})$ obtained from $H_a(s)$ using the modified matched-Z transform with gain matching at DC, traditional matched-Z, impulse invariant, and various bilinear transforms, respectively, of Chapters 6 and 7. The differences in the zero locations obtained using each method are clearly visible, as was noted previously. The pole locations are identical for the IIT and all three variations of the matched-Z transform, whereas the four versions of the bilinear transform shown in Figure 8-7 through Figure 8-10 show different pole locations due to the different pre-warping methods. The next section will focus in more detail on the effects of pole and zero locations on the overall frequency responses of the model.

The computer code which implements the graphical technique used to create the plots in Figure 8-3 through Figure 8-10 is included in Addendum 8A: . **At the time just prior to the publication of this dissertation, this new graphical plotting technique had been under consideration for patent protection. It will now be submitted for publication as an IEEE peer-reviewed publication** [94], [95]**.**[24]

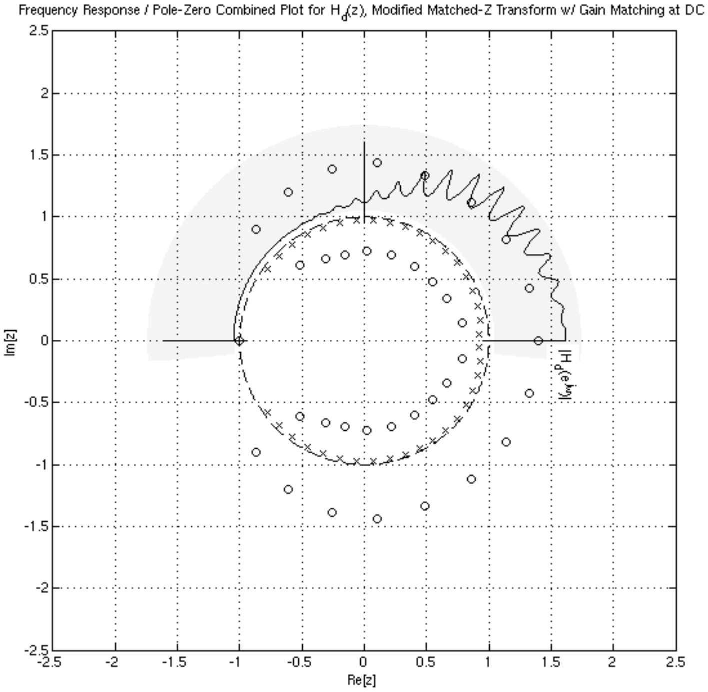


Figure 8-4: Combined Frequency Response and Pole-Zero Plot for Hd(z) for the MMZT with Gain Matching at DC

---

[24] For information regarding the final disposition of the graphical method's publication status, please contact the author of this dissertation after 31 July 2014. Contact information is listed in Appendix 2.

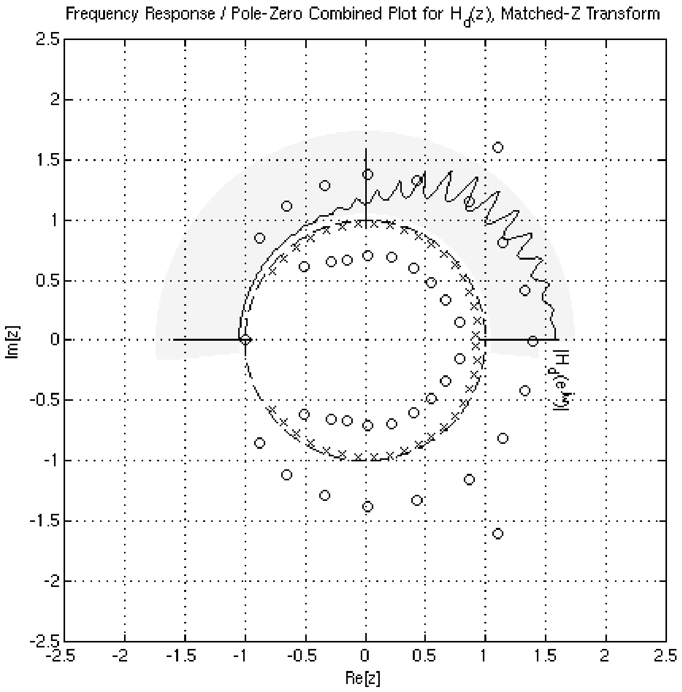


Figure 8-5: Combined Frequency Response and Pole-Zero Plot for ***$H_d(z)$*** for the MZT

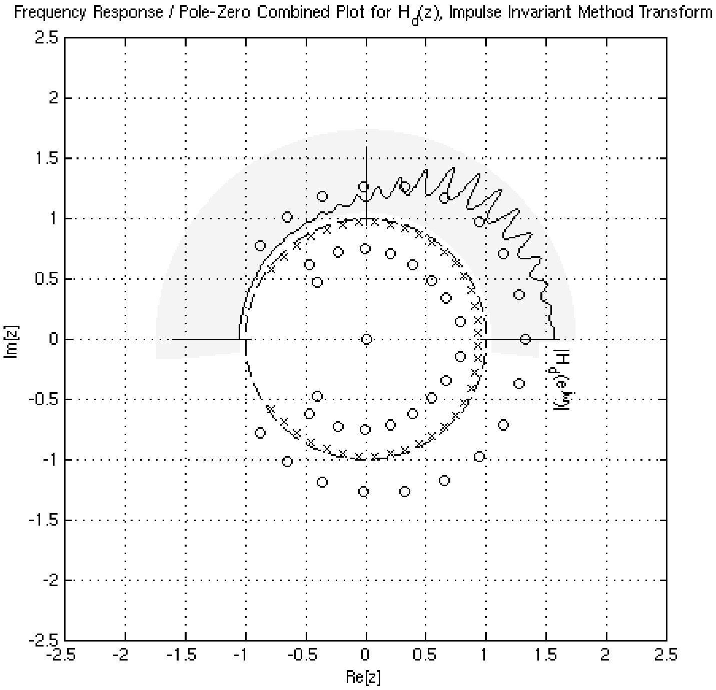


Figure 8-6: Combined Frequency Response and Pole-Zero Plot for ***$H_d(z)$*** for the IIT

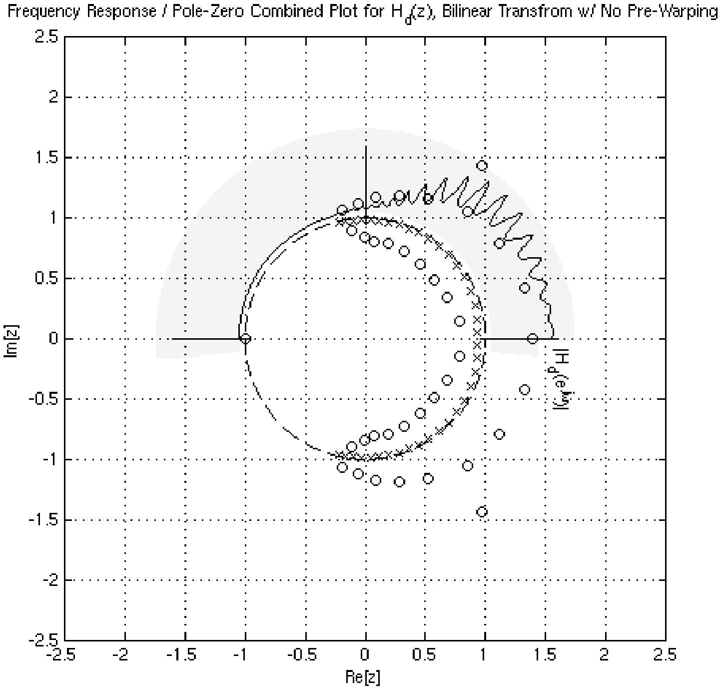


Figure 8-7: Combined Frequency Response and Pole-Zero Plot for $H_d(z)$ for the BLT with no Pre-Warping

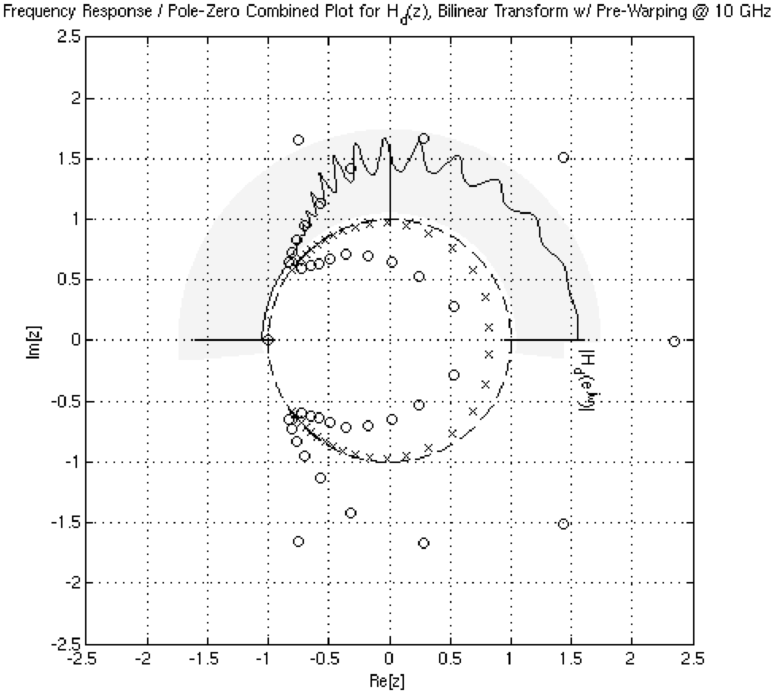


Figure 8-8: Combined Frequency Response and Pole-Zero Plot for $H_d(z)$ for the BLT with Pre-Warping @ 10 GHz

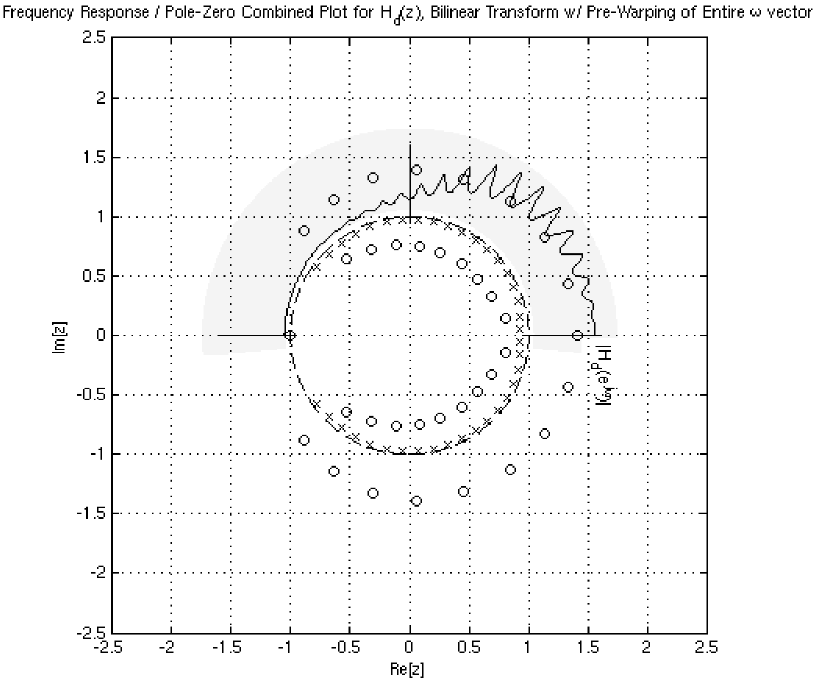


Figure 8-9: Combined Frequency Response and Pole-Zero Plot for *$H_d(z)$* for the BLT with Pre-Warping of Entire ω Vector (Ikai Method)

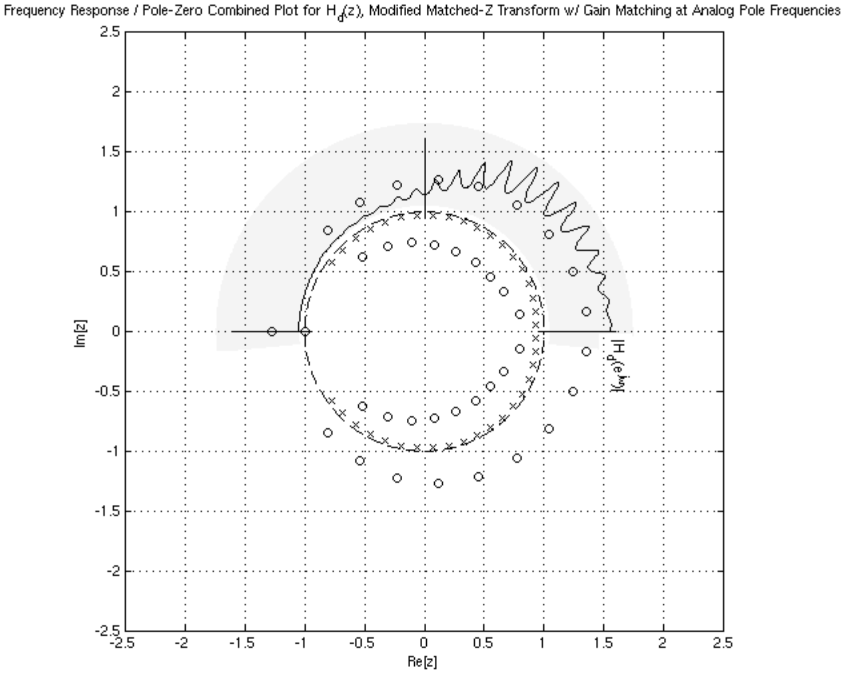


Figure 8-10: Combined Frequency Response and Pole-Zero Plot for *$H_d(z)$* for the MBLT with Pre-warping to Analog Pole Frequencies

8.4 Effect of Individual Pole and Zero Locations on Frequency Response Characteristics

Figure 8-11 shows an enhanced version of the combined frequency response and pole-zero graphical plot, for the case of the MMZT with gain matching of the 1st-order sections at the analog pole frequencies. The enhancements include the addition of dashed lines, for select pole locations, which show very clearly the correlation of the pole locations with the corresponding peaks of the magnitude response curve. Recall from the discussion in the previous section that the discrete frequency variable ω increases counter-clockwise around the unit circle.

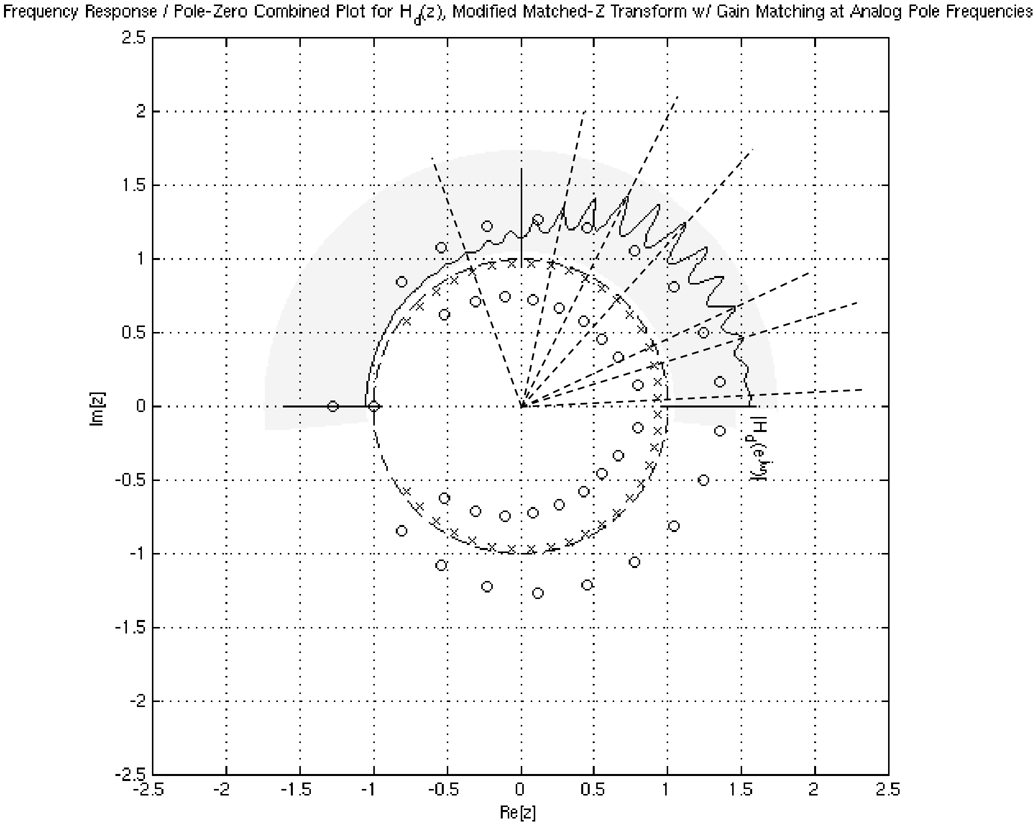


Figure 8-11: Enhanced Version of the Combined Frequency Response and Pole-Zero Plot for *$H_d(z)$* Highlighting Correlation of Pole Positions to Frequency Response Peaks

In contrast to the correlation of the pole locations to the relative maxima (peaks) of the magnitude response curve, it is clear looking at Figure 8-11 that the zero locations do not correlate as strongly to the relative minima (valleys) of the magnitude response. The primary reason for this discrepancy is the more-distant position of the zeros relative to the unit circle, as opposed to the poles, which are very close.

8.4.1 Major Effects of the Poles and Zero Locations Relative to the Unit Circle

One significant factor influencing the relative effect of various poles and zeros on the system response, as known from both control systems theory and digital signal processing theory, is their proximity to the stability boundary in either the s-plane or the z-plane. Specifically, it is known from linear time-invariant (LTI) system theory that poles and zeros which are closer to this boundary have a more direct effect on the system response characteristics than those which are farther away [40], [42], [48].

Recognition of this fact leads immediately to an increased understanding of the nature of the 4th-order test circuit’s characteristics, as observed throughout this project. Specifically, a detailed examination of the z-plane pole-zero plots of Figure 8-3 through Figure 8-10 reveals that the system poles are very close to (but within) the unit circle $|z| = 1$, while the zeros are farther away, both within and outside the unit circle. Furthermore, it has been clear throughout Chapters 6 and 7 that it is the zero positions that tend to vary the most when using the various s- to z-transformations, and that the effect of this variation on the time-domain and frequency-domain responses has been fairly minimal. In contrast, variations in the pole positions, such as those encountered with certain formulations of the bilinear transform, have had a dramatic effect on the overall responses in both the time and frequency domains. Clearly, the pole positions obtained during the initial vector fitting step are of great importance in determining the subsequent performance of the model and its overall characteristics, due to their proximity to the imaginary frequency axis in the s-plane, and thus to the unit circle in the z-plane after their transformation there.

8.4.2 Major Effects of the Poles and Zero Locations Relative to the Nature of the Phase Response

It is also known from (LTI) system theory that finite impulse response (FIR) filters with linear phase have zero positions which are reciprocal pairs and complex conjugate pairs [42]. Careful examination of the z-plane pole-zero plots of Figure 8-3 through Figure 8-10 reveals that the zero positions of $H_d(z)$ for the 4th-order test circuit, even though the transfer function is IIR in nature, share nearly this same characteristic. In this case, the zero locations alternate between the inside and outside of the unit circle with increasing ω, and they clearly occur in complex conjugate pairs, unless they are on the real axis.

It can also be easily demonstrated that the zeros basically mimic the reciprocal behavior expected for linear-phase FIR systems. Figure 8-12 shows the positions of hypothetical reciprocal zeros calculated from the known zero locations of $H_d(z)$. It is seen that the reciprocal locations are close to the actual zero locations, and thus it is reasonable to conclude the system may exhibit similar phase response characteristics.

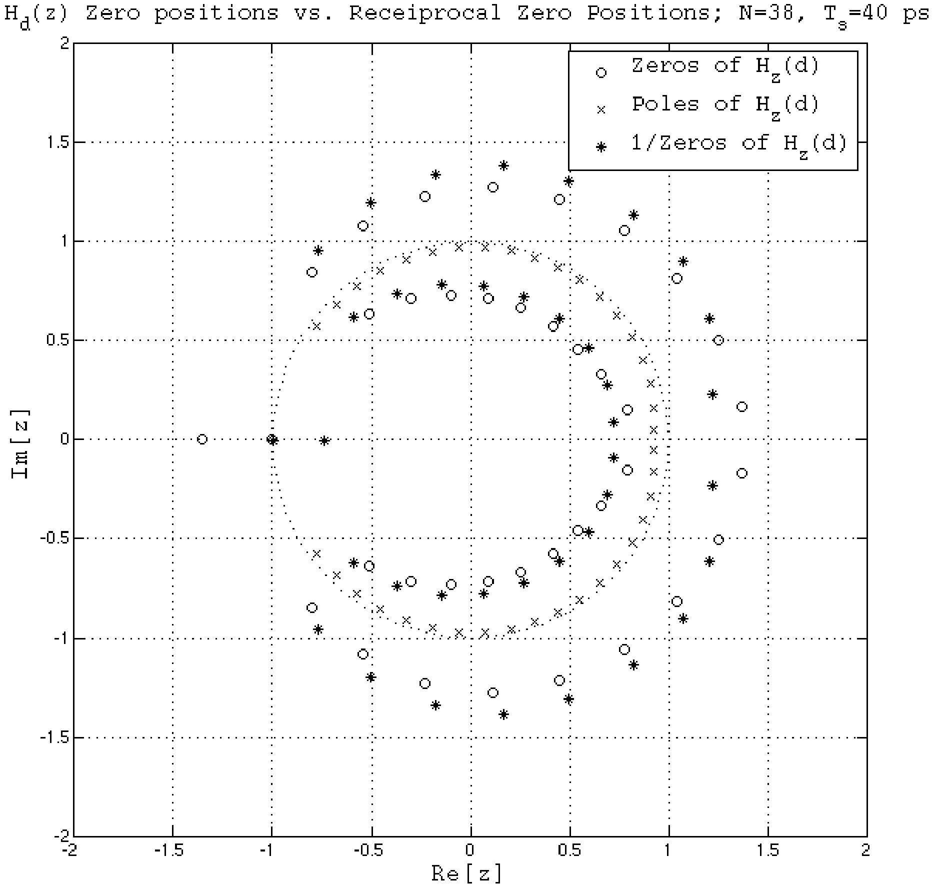


Figure 8-12: Comparison of the Zero Positions of *$H_d(z)$* with their Hypothetical Reciprocal Zero Locations

Finally, it should be noted that the characteristic of alternating zeros which somewhat approximate reciprocal, conjugate pairs is present regardless of which of the s- to z-domain transformations was initially utilized to form the transfer function $H_d(z)$, with the exception of the BLT variations of Figure 8-7 and Figure 8-8. Indeed, it should be recalled from Chapters 6 and 7 that these two BLT transforms did not exhibit linear phase characteristics. As such, it is clearly apparent that the phase characteristics prevalent for $H_d(z)$ transforms should be expected, and that their linear nature is somewhat predictable from the zero locations in the z-plane.

8.4.3 Potential for Model Simplification Utilizing Pole-Zero Cancellation

Another obvious application of the pole-zero mapping techniques available in both the s- and z-domains would be assessment of whether the transfer function models $H_a(s)$ and $H_d(z)$ could be simplified to a lower-order approximation. One method for doing this might be to look for poles and zeros that are near to one another, and thus partially or completely cancel each other. Pole-zero cancellation techniques are commonly used in the fields of control systems theory, for example, as a compensation technique [40], [48].

To demonstrate these techniques, the vector fitting algorithm employed previously to find $H_a(s)$ was run again on the frequency response $H_a(j\Omega)$ of the 4th-order test circuit, but this time the order was forced to be higher than that of $N = 38$ found previously. In this case, $N = 42$ was chosen as the order, resulting in a transfer function $H_a(s)$ with four extra poles and zeros than for the case where $N = 38$. The results are shown in the s-plane in Figure 8-13, where the extra four poles and extra four zeros are shown near the origin, and they are shown to cancel completely. The pole-zero map for the original case of $N = 38$ is included in Figure 8-14 for easy comparison.

From this diagram, it is seen that the vector fitting method chooses the extra poles and zeros so that they exactly cancel. What is also clear is that, in general, the poles and zeros for the $N = 38$ model for the 4th-order test circuit are quite far apart, and that there is little opportunity for pole-zero cancelation. Figure 8-15 and Figure 8-16 show the corresponding diagrams in the z-plane for $H_d(z)$ found using the MMZT, and they force a similar conclusion, i.e., there is little opportunity for pole-zero cancellation.

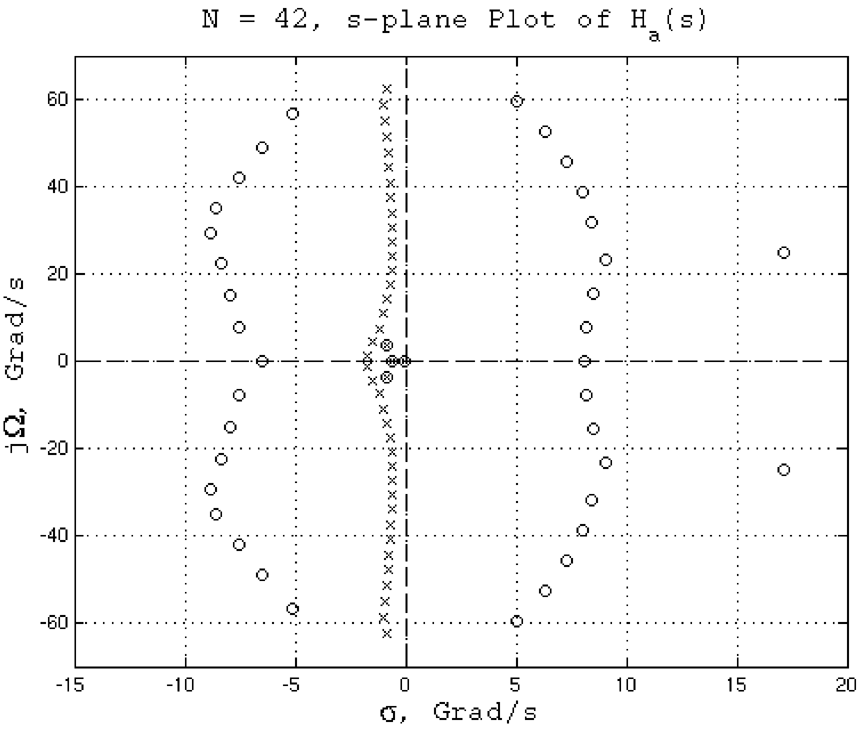


Figure 8-13: Pole-Zero Plot in s-Plane Showing the Effect of Pole-Zero Cancellation on $H_a(s)$, for 4th-order Test Circuit with $N=42$

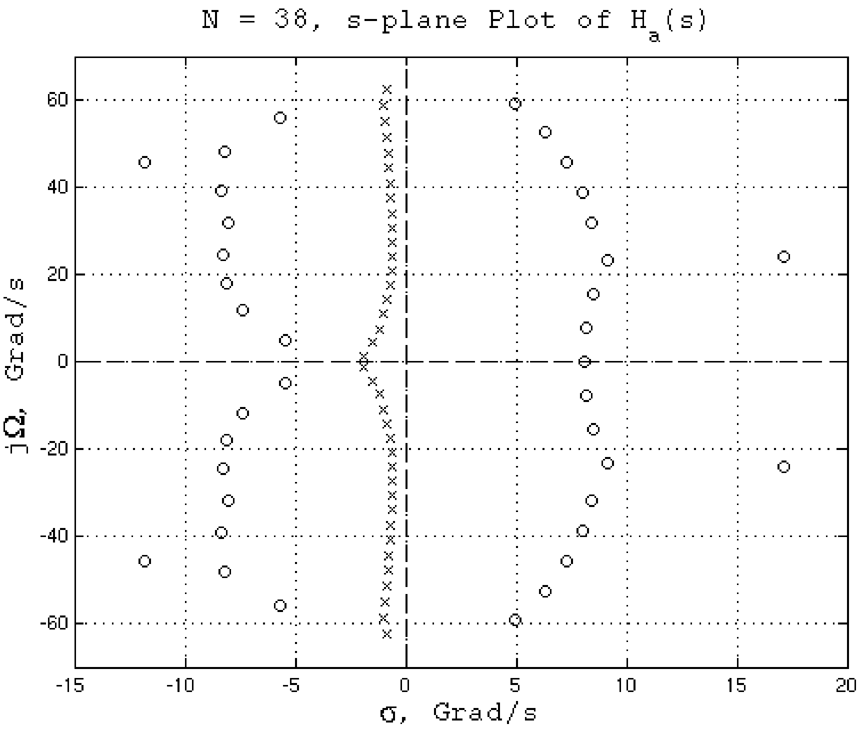


Figure 8-14: Original Pole-Zero Plot in s-Plane of $H_a(s)$, for 4th-order Test Circuit with $N=38$

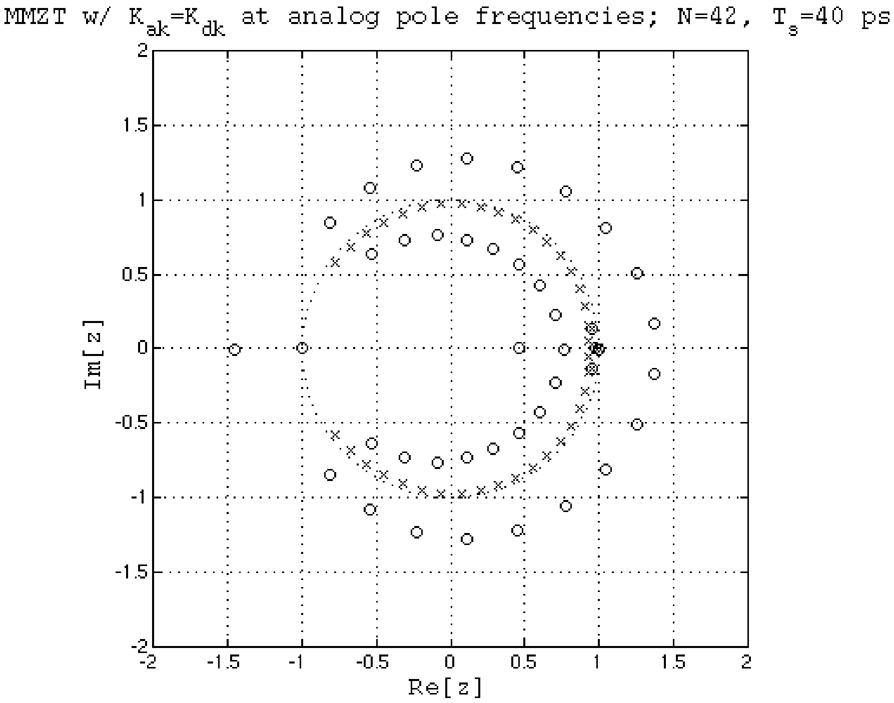


Figure 8-15: Pole-Zero Plot in z-Plane Showing the Effect of Pole-Zero Cancellation on $H_d(z)$, for 4th-order Test Circuit with N=42, $T_s=40$ ps

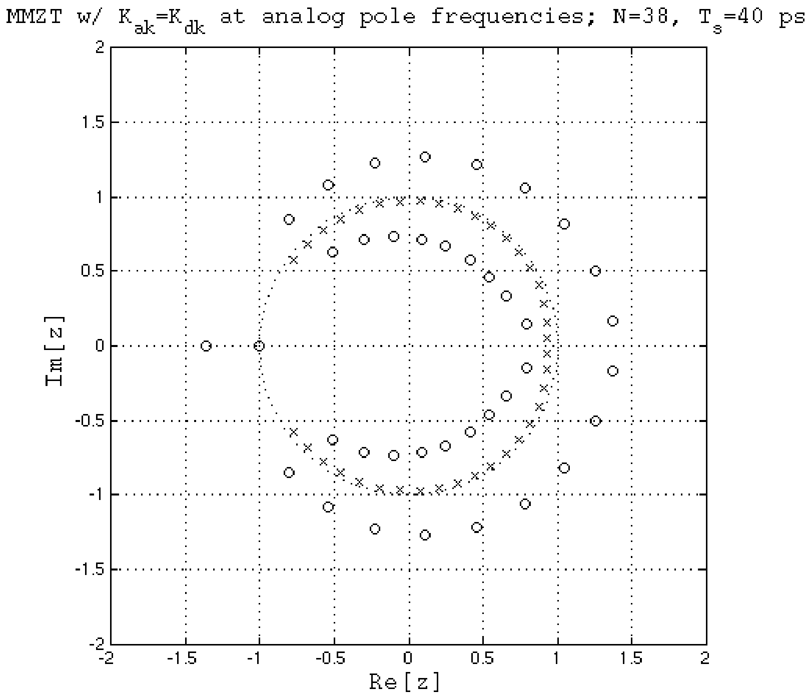


Figure 8-16: : Pole-Zero Plot in z-Plane of $H_d(z)$, for 4th-order Test Circuit with $N=38$, $T_s=40$ ps

8.5 Model Frequency Bandwidth and Sampling Rate Considerations

It is clear that the process for generating $H_d(z)$ from $H_a(s)$ throughout this project involved a two part process: 1) a system identification step, performed using scattering parameter characterization in the analog frequency domain, followed by a vector fitting step to form $H_a(s)$; and 2) transforming $H_a(s)$ to $H_d(z)$ using an s- to z-domain transformation. For both of these steps, there are decision points which have great implications for model accuracy, model complexity, and model usability for subsequent analysis.

8.5.1 Frequency Bandwidth Determination During the System Identification Process

A major consideration when performing the system identification step in the analog frequency domain is the analog frequency range over which the identification should occur. Generally the range is selected based on the assumed or known frequency spectra of the input and output signals present in the system, such that the resulting model has sufficient bandwidth to not overly distort the signals, either by attenuation or dispersion effects. In signal integrity engineering, there are also commonly-employed metrics based on signal risetime of the input signal [1], [2], while in transmission line theory the relationship of the wavelength to the physical dimensions of the circuit is often a primary consideration [6].

There are pragmatic limitations regarding bandwidth, however, as extending the model over a wider frequency range tends to make the rational function models derived using the vector fitting step more complex and of higher order. It was seen in Chapter 5, for example, that the frequency range over which the identification is performed directly influenced the resulting order of the model, along with other factors such as how closely the frequency response of the fitted model $H_a(s)$ needed to match the original frequency response $H_a(j\Omega)$.

8.5.2 Sampling Rate Determination During the s- to z-Domain Transformation

Similarly, when working with the transfer function $H_d(z)$ in the discrete frequency domain, it is known there are considerations related to the sampling rate f_s (or, equivalently, the sampling interval T_s) that are driven by the Shannon sampling theorem [41], which states the analog signal must be sampled at a rate that is greater than twice the largest frequency present in the signal. These considerations are primarily oriented toward avoiding frequency aliasing, or the introduction of phantom frequencies, into the discrete representation of the signal. It was found in this project, however, that there are additional considerations for selecting T_s which are also very pragmatic in nature. These considerations are related to the readability of the resulting pole-zero maps and frequency response plots that form the basis for many of the analysis methods utilized extensively in this project. Interestingly, these considerations tend to favor selecting a sampling frequency f_s that is no higher than necessary (or, equivalently, selecting a sampling interval T_s which is no smaller than necessary). Figure 8-17 shows the z-plane map for the transfer function $H_d(z)$ of the 4th-order test circuit, recalculated with $T_s = 30$ ps. Comparing directly with Figure 8-16, for which $T_s = 40$ ps, note that the poles and zeros are now concentrated around a smaller arc of the unit circle, and take up less of the overall plot, making it harder to read the pole and zero locations.

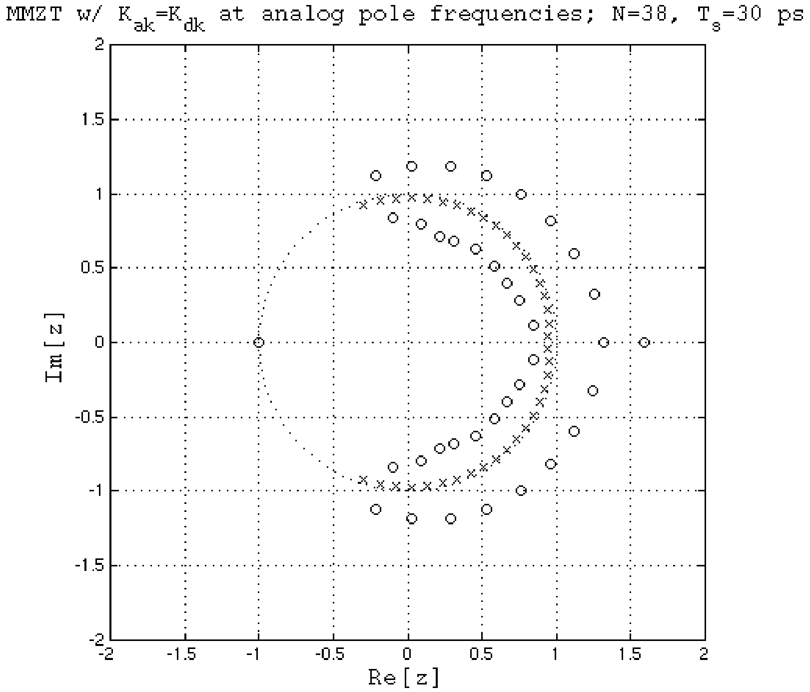


Figure 8-17: Pole-Zero Plot in z-Plane of $H_d(z)$, for 4th-order Test Circuit with $N=38$, $T_s=30$ ps

Thus there is a tension and potential tradeoff, in both the analog domains and in the discrete domains, between mathematical and physical theoretical accuracy, and pragmatic limitations which, when imposed, make application of the methods more effective and easier. In the analog frequency domain, during the system identification step, the tradeoff is between increased model bandwidth (and thus less potential for signal distortion) and higher order of the resulting transfer function $H_a(s)$. In the discrete frequency domain, during the s- to z-domain transformation step, the tradeoff is between higher sampling rate (and thus less opportunity for phantom frequencies due to aliasing) and readability of the resulting graphical diagrams used to assess the system characteristics.

8.6 Conclusions Regarding z-Plane Analysis of Signal Integrity Interconnect Models

This chapter addressed three aspects of z-domain analysis. The first consisted of the introduction of an enhanced graphical plotting technique, which combines the pole-zero map of the discrete transfer function $H_d(z)$ in the z-plane with a plot the frequency response $H_d(e^{j\omega})$ on the same figure. This technique provides for the simultaneous evaluation of both the pole-zero map the frequency magnitude response, and, optionally, the frequency phase response. **At the time just prior to the publication of this dissertation, this new graphical plotting technique had been under consideration for patent protection, and will now be submitted for IEEE publication** [94], [95]**.**

The second discussion focused on the evaluation of the effect of the pole-zero positions in the z-plane, with several implications. It was highlighted that the dominant poles and zeros can be identified simply by noting their proximity to the unit circle and that, as such, the methods of this project naturally lead to the poles dominating the response. Next, an example was shown for an artificially forced high-order model that lead to excess poles and zeros that completely cancel. Examination of the cancellation effect, however, led to the observation that there is likely to be little opportunity for further pole-zero cancellation, due to the fact that the zeros and poles are, generally, relatively far apart. Finally, it was noted that the nearly linear phase characteristics of these systems is generally explainable by the pattern of zero positions that occur in complex-conjugate, and nearly reciprocal pairs, very similar to a known characteristic of linear phase FIR filters.

The third discussion centered on pragmatic considerations related to careful selection of the frequency bandwidth of the model when performing the system identification step in the analog frequency domain, and careful selection of the sampling interval when performing the s- to z- domain transformation of the transfer function. It was shown that increased frequency bandwidth leads to increased model complexity, which is undesirable, while increased sampling rates lead to a compressed spacing of the poles and zeros on the z-plane pole-zero map, which make the plots less readable and the graphical assessment techniques less useful.

The next chapter represents the closing topic of this research project, namely, proposed methodology enhancements which will improve the effectiveness of signal integrity analysis and design processes of the type utilized in industrial design team environments. These enhancements will be comprised mainly of the complex discrete frequency domain techniques introduced and heavily utilized in Chapters 6, 7, and 8, and their associated characteristics and implications.

Addendum 8A: MATLAB Computer Program Listing for Combined Frequency Response and Pole-Zero Map

```
function freqpzplot_bwaxes(B,A)

%% Function Description
% 08/02/2013: PED - Created function to implement combine frequency
%             response and pole-zero plot on same graphical view.
%             B is an input: The NUMERATOR of the z-domain transfer
%             function.
%             A is an input: The DENOMINATOR of the z-domain %
%             transfer function.
%             omega is an input: The digital frequency vector w.
% 09/06/2013  PED - Modified function for black & white coloring for
%             final thesis formatting.

format compact;  % tightens loose format, if present
format long e ;  % sets output variables to double precision
%
N=1024;
omega = linspace(0,2*pi,N);
% create vector omega for 0 to 2*pi.
x = cos(omega);
% generate x-coordinate for unit circle
y = sin(omega);
% generate y-coordinate for unit circle
xl=linspace(-1.6, -0.95, 100);
xr=linspace(0.95, 1.6, 100);
yc=linspace(0.95, 1.6, 100);
%
rp = 1.05; % augment radius to serve as basis for w-plot
xp = rp.*cos([omega(63*N/64:N) omega(1:33*N/64)]);
% x-coordinate for w-plot augmented semi-circle axis
yp = rp.*sin([omega(63*N/64:N) omega(1:33*N/64)]);
% y-coordinate for w-plot augmented semi-circle axis
%%
% Compute DTFT using freqz
Homega=freqz(B,A,omega(1:N/2));
Hmag=abs(Homega);
%
% Compute poles and zeros.
zeros_z=roots(B); poles_z=roots(A);

% Compute "wrap-around" re-mapped Hmag
%
Y=1.2.*max(Hmag); Yprime=0.75;
% Yprime compresses the Hmag amplitude.
rplus=Yprime.*(Hmag(1:N/2))./Y;
xprime=(rp+rplus).*cos(omega(1:N/2));
yprime=(rp+rplus).*sin(omega(1:N/2));
```

```
% Generate doughnut for Hmag response background.
xp1 = rp.*cos([omega(63*N/64:N) omega(1:33*N/64)]); % x-coordinate
% for w-plot max-range semi-circle
yp1 = rp.*sin([omega(63*N/64:N) omega(1:33*N/64)]);

xp2 = (rp+1.1.*max(rplus)).*cos([omega(63*N/64:N)
omega(1:33*N/64)]);
% x-coordinate for w-plot max-range semi-circle
yp2 = (rp+1.1.*max(rplus)).*sin([omega(63*N/64:N)
omega(1:33*N/64)]);
% y-coordinate for w-plot max-range semi-circle

C=0.96*[1 1 1];
% Shading hue for the semi-circular Hmag area of the pole-zero plot.

% Plotting routines

plot(x,y,'k--'); % plot unit circle
hold on;
fill([xp1,fliplr(xp2)],[yp1,fliplr(yp2)],C,'EdgeColor','None');
% Define and fill Hmag shading area.
hold on; plot(xl,zeros(100,1),'k');
% plot "remapped y-axis line" at omega=pi
hold on; plot(xr,zeros(100,1),'k');
% plot "remapped y-axis line" at omega=0
hold on; plot(zeros(100,1),yc,'k');
% plot "remapped y-axis line" at omega=pi/2
% hold on; plot(xp,yp,'k');
% plot w-plot "remapped x-axis" circle
hold on; plot(xprime,yprime,'k');
% plot frequency response "wrapped-around" the pole-zero plot.
plot(real(zeros_z),imag(zeros_z),'ok',real(poles_z),imag(poles_z)
'xk'); % plot pole-zero map
axis([-2.5,2.5,-2.5,2.5]); axis square; hold off; xlabel('Re[z]');
ylabel('Im[z]');
grid on;
set(gca,'Layer','top');
% Bring grid to top layer of plot (i.e., forward)
% grid off;
```

Chapter 9: Signal Integrity Methodology Enhancements Resulting from the Utilization of the Discrete Transfer Function Model $H_d(z)$

9.1 Introduction

In Chapters 2 and 3, it was established that the current state of signal integrity engineering analysis methods are primarily based in the continuous-time domain and analog-frequency domain. The typical output data which are generated, usually by simulation, are continuous time-domain waveforms representing the signals traveling in the system and frequency-domain scattering parameter representations of the channel characteristics [1], [4], [72]. The discussion continued in Chapter 4 with the use of system identification methods to generate frequency transfer functions $H_a(j\Omega)$ and $H_d(e^{j\omega})$ in the analog and digital frequency domains. From these functions, the transfer functions $H_a(s)$ and $H_d(z)$ were ultimately generated in the complex frequency domains, as described in Chapters 5 through 7. This was done using a least-squares vector fitting process to generate $H_a(s)$, followed by application of various s- to z- transformations to generate $H_d(z)$.

A primary advantage of working in both the analog and digital complex frequency domains was shown, in Chapters 5 through 7, to be increased insight into the nature of the system, largely due to the use of graphical pole-zero analysis of the transfer functions. Further, the advantages of visualization in the z-plane over the s-plane were highlighted, due to the finite nature and better scaling of the zeros and poles relative to the origin and the real and imaginary axes. Finally, it was also demonstrated in Chapter 8 that it is possible to gain considerable insight into the nature of the system by examining the model extensively in the z-plane, and to leverage this additional insight into efforts to improve the quality of the analysis, or at least to gain confidence that an appropriate level of complexity has been obtained for the transfer function model [6].

In this chapter, these methods, which are new in their systematic application to signal integrity applications, will be discussed in the context of an enhanced signal integrity engineering methodology. Such methodologies are generally used in an industrial setting for the design of complex electronics systems, such as computers [4]. They can often involve teams of multiple signal integrity engineers, collectively running hundreds or thousands of time-domain simulations over the course of many months. Any improvements to the efficiency and effectiveness of the design methodologies thus could have great implications for improving the quality of the design, minimizing the expended resources, and limiting the time duration required to complete the design cycle.

The discussion will begin with a description of signal integrity engineering design methodologies typically used today, both for basic and sophisticated users. Next, a summary of the various s- to z-transformations available will be presented in a form that allows for their evaluation for use in such a methodology. Finally, a proposal for an enhanced comprehensive signal integrity design methodology which incorporates these new methods from the field of digital signal processing will be introduced.

9.2 Summary of a Typical Legacy Signal Integrity Engineering Process Flow

Figure 9-1 shows a block diagram which simplistically represents the core activities in signal integrity engineering, which are the generation a SPICE netlist to represent the system model, the execution of the model in the SPICE simulator, and the review and interpretation of the resulting continuous-time waveforms.

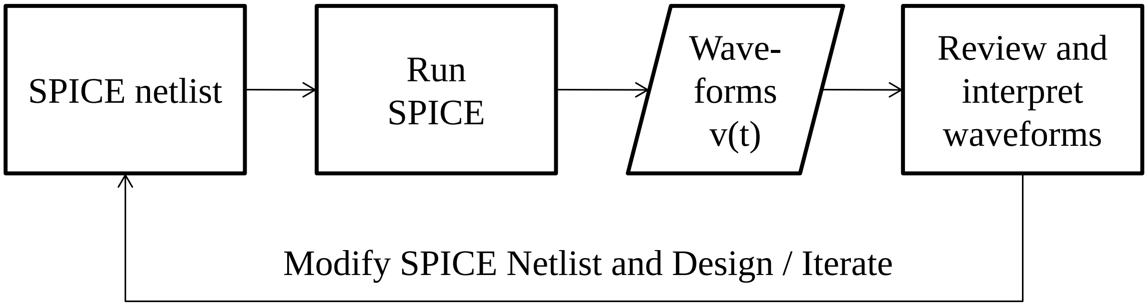


Figure 9-1: Core Activities Related to SPICE Simulation in Signal Integrity Engineering

After the waveforms have been reviewed, the SPICE netlist is usually modified so that the simulation can be re-run, by changing one or more particular parameters and evaluating the effect on the waveform quality. This is done iteratively, with the resulting modifications ultimately resulting in an improved waveform shape, and thus better switching performance at the receiver and overall system timing performance [2].

The block diagram of Figure 9-1 is, of course, overly simplistic in that it does not address the linkage that must exist between the SPICE netlist model and the physical system which it represents. The SPICE netlist must account for the system interconnect's electrical characteristics, and accurately represent such things as the printed wiring board (PWB) traces' propagation delay, as well as other electrical aspects of the interconnect. These include the interconnect's characteristic impedance, pad and via parasitics, and conductor and dielectric losses. In addition to the interconnect model, the SPICE netlist would also include models for the transmitter (Tx) and receiver (Rx) circuits, and any termination intentionally applied to the network to improve signal integrity characteristics [1], [2].

The signal integrity analysis must account for all of these factors when the initial SPICE netlist model is generated. After the initial waveforms have been evaluated, an assessment must be performed of how any proposed changes to the SPICE netlist electrical model will manifest themselves as physical changes in the system. Those physical changes must be validated relative to other system design constraints, such as physical size and other mechanical features, thermal performance, power consumption, system cost, design complexity, and field serviceability [7]. Figure 9-2 shows an enhanced version of Figure 9-1 which accounts for this interaction with the overall system design process.

Other details added to Figure 9-2 show the system identification step used to generate the distributed models for the system interconnects. It was shown in previous chapters that scattering parameter characterization is a useful method for identifying data from which the complex frequency domain models $H_a(s)$ and $H_d(z)$ can be subsequently derived. Similarly, it is common in the formation of SPICE netlist models for the SPICE simulation engine to call scattering parameters directly, as most modern industrial SPICE simulators can handle scattering parameter data directly as input. While this can be a useful feature, one problem with this approach is that it can lead to transient simulation results that are inaccurate due to poor passivity or causality attributes, or both [63]. Furthermore, it is possible this inaccuracy will be significant enough to affect the design attributes, yet subtle enough that the effect may be undetectable without careful examination [63].

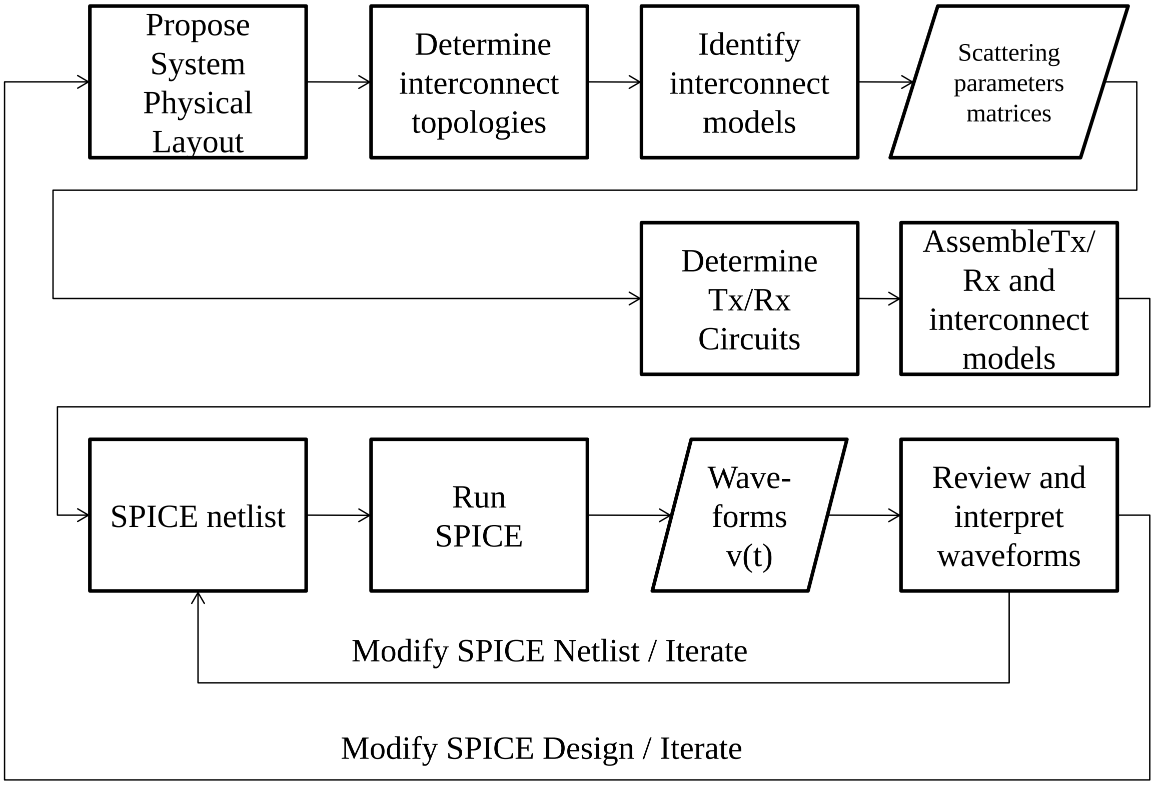


Figure 9-2: Typical Signal Integrity Engineering Design Process Flow for Basic Users, Including Interaction with System Design Process

To address these potential modeling problems, many sophisticated signal integrity methodologies today incorporate additional model assessment steps, shown in Figure 9-3, to check the scattering parameter matrices for causal and passive behavior prior to invoking the time-domain SPICE simulator. These assessment steps are typically performed in the analog frequency Ω-domain, and thus the z-domain methods utilized in the project may offer an additional enhancement to the flow of Figure 9-3.

Finally, note that these assessment methods, as depicted in Figure 9-3 and as related specifically to the field of signal integrity engineering, are directed at the scattering parameter matrices themselves. They are not, as such, directly applied to the system voltage transfer functions $H_a(s)$ and $H_d(z)$ generally considered in previous chapters, and used so prevalently in digital signal processing and control systems engineering. Note also that the methodologies described by the flow diagrams of Figure

9-1 through Figure 9-3 are used widely in industrial signal integrity design methodologies today. The next sections will discuss how these methodologies might be further enhanced using some of the digital signal processing methods applied in this project, specifically in the s-plane and z-plane, as described in previous chapters.

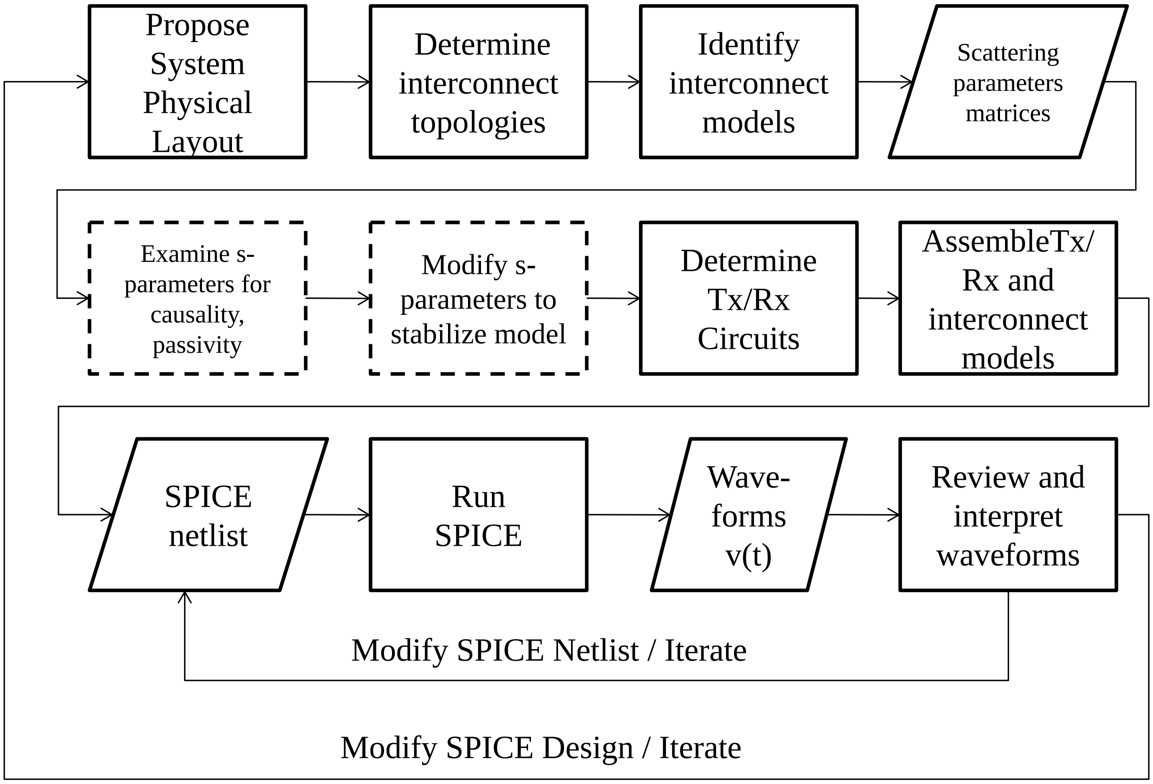


Figure 9-3: Typical Signal Integrity Engineering Design Process Flow for Sophisticated Users, Including Interaction with System Design Process

9.3 Comprehensive Summary and Comparison of the s-Domain to z-Domain Transformations Utilized in this Research Project

As described in Chapters 6 through 8, the various s- to z-domain transformations used for this project have various advantages and disadvantages. As such, it would be useful to summarize these characteristics such that they can be systematically assessed for their application to the signal integrity engineering flow. The following pages summarize, for each transformation, the following six attributes:

1. Pole mapping algorithm and effectiveness, where preference is given to algorithms which map the s-plane poles to the z-plane using the straightforward $z = e^{sT_s}$ alogrithm;
2. Zero mapping algorithm and effectiveness, where preference is given to algorithms that map the s-plane zeros at $s \to \infty$ to the Nyquist frequency in the z-plane;
3. Starting form of the expression for $H_a(s)$, where preference is given to those transformations which work directly with the pole-residue form;
4. Matching obtained with the magnitude response curve of $H_a(j\Omega)$;
5. Matching obtained with the phase response curve of $H_a(j\Omega)$;
6. Matching obtained with the time-domain response calculated using $H_a(s)$.

These attributes are summarized for the individual transforms in Table 9-1 through Table 9-8, with some short explanations and supporting detail for each case. Finally, Table 9-9 summarizes and compares the attributes of all eight transformations, with an overall score calculated for each transform, as a measure of its effectiveness for signal integrity engineering applications.

When utilizing these transforms for the creation of the discrete transfer function $H_d(z)$, a methodical consideration of the above attributes would be a useful step. This will be added as a formal process step to the final, enhanced signal integrity methodology presented at the end of this chapter. Note that it is not assumed that only a single s- to z-transformation should be utilized in all signal integrity engineering applications; there are too many variations in the performance of the transformations, and thus some tradeoffs arise. In addition to the six attributes listed above, there could be additional attributes that may be important when selecting a particular method. Some examples of additional attributes that may be important in a particular application would be:

1. the presence of outlying zeros far from the z-plane origin;
2. significant changes to the zero positions which are calculated by the transformation as T_s is varied;
3. Numerical stability of the transform as it is implemented to $H_a(s)$.

There are obviously numerous other attributes that could be applied.

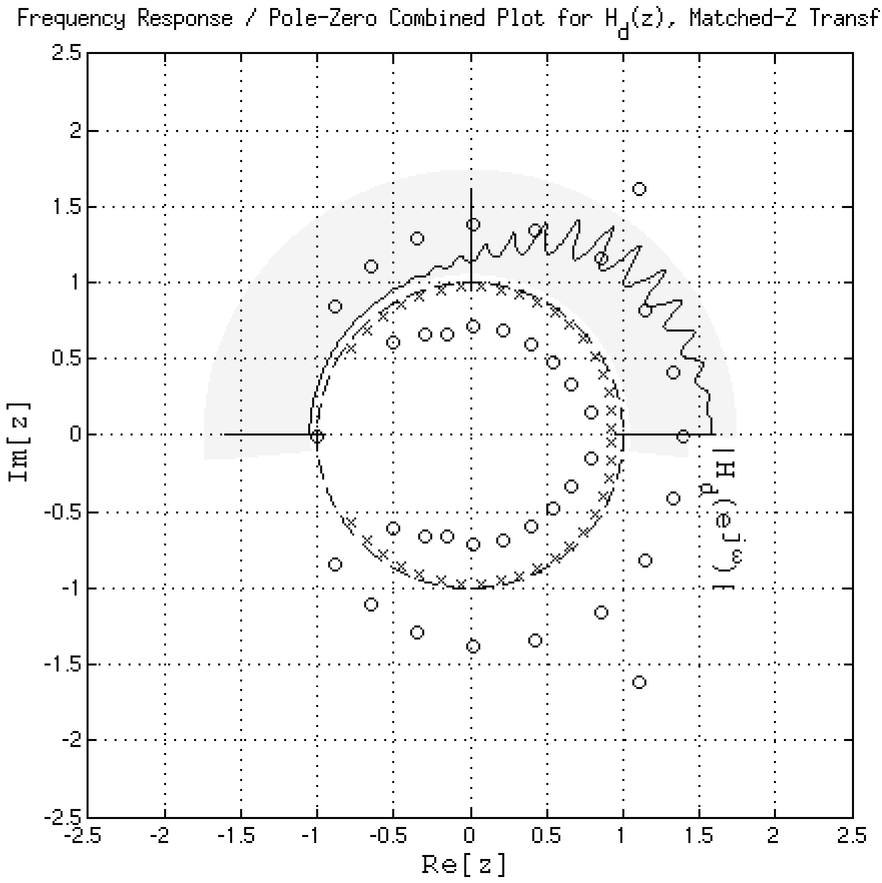


Figure 9-4: Combined Pole-Zero / Frequency Response Plot for the Matched-Z Transformation (MZT)

Matched-Z Transform (MZT)		**±**
Criterion	**Results**	
Pole Mapping Method	Map via $z = e^{sT_s}$	++
Zero Mapping Method	Map via $z = e^{sT_s}$ Zeros at ∞ map to $z = -1$	++
Starting Form for $H_a(s)$	$H_a(s)$ must be in pole-zero factored form	-
Magnitude Resp. Matching to $H_a(s)$	Fair; gain only matched at one frequency point	-
Phase Response Matching to $H_a(s)$	Excellent	++
Time Response Matching to $H_a(s)$	Very good	+

Table 9-1: MZT Attributes

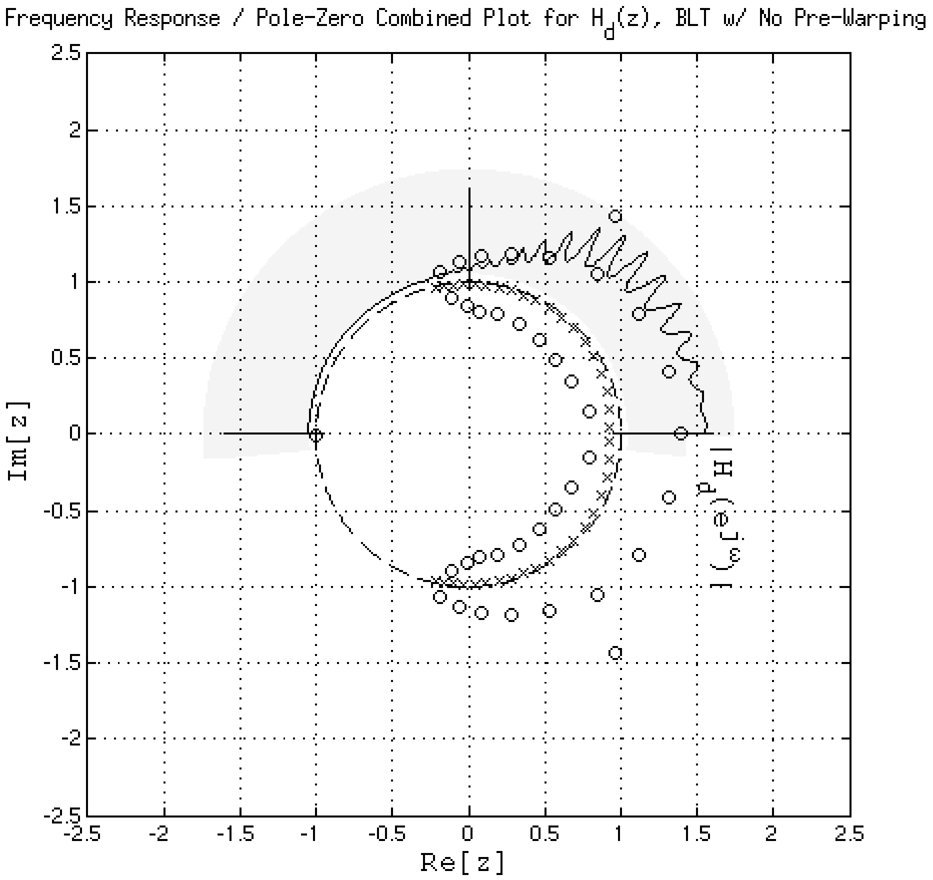


Figure 9-5: Combined Pole-Zero / Frequency Response Plot for the Bilinear Transformation with no Pre-Warping (BLT0)

BLT w/ no Pre-Warping (BLT0)		±
Criterion	**Results**	
Pole Mapping Method	Poor mapping via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to warping	--
Zero Mapping Method	Poor mapping via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to warping	--
Starting Form for $H_a(s)$	$H_a(s)$ may be in pole-residue form	++
Magnitude Resp. Matching to $H_a(s)$	Poor; gain only matched at DC	--
Phase Response Matching to $H_a(s)$	Fair	-
Time Response Matching to $H_a(s)$	Fair	-

Table 9-2: BLT w/ no Pre-Warping Attributes

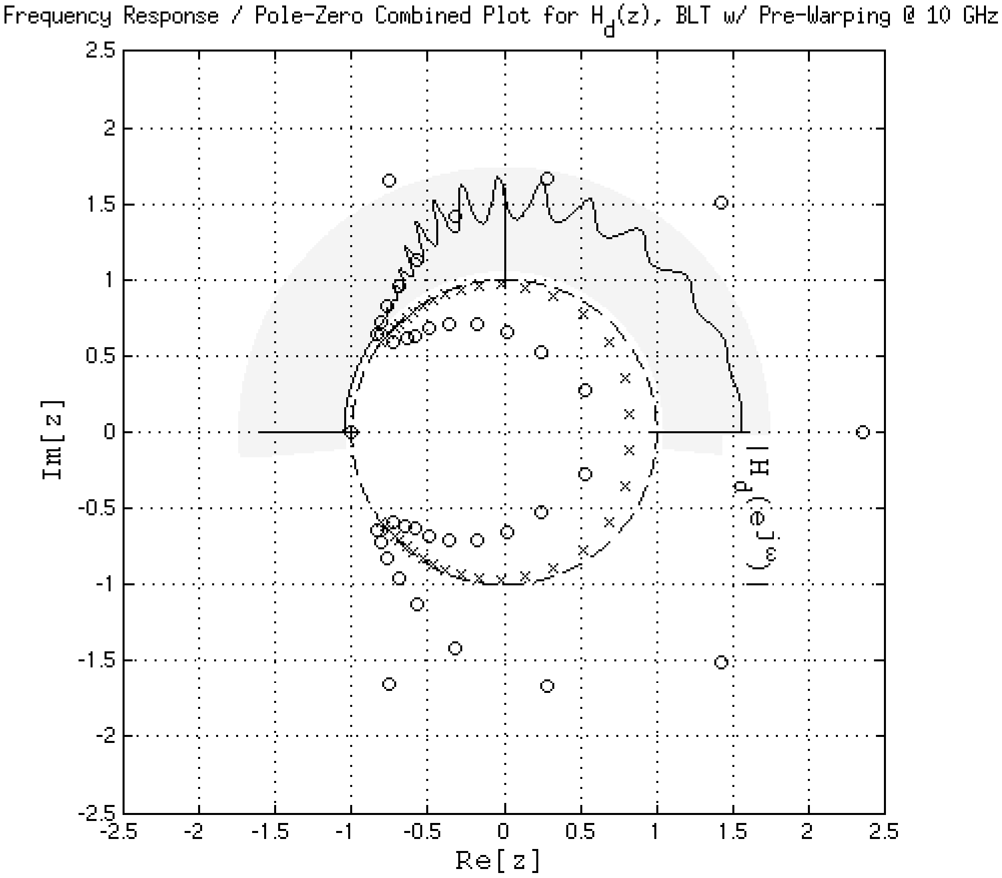


Figure 9-6: Combined Pole-Zero / Frequency Response Plot for the Bilinear Transformation with Pre-Warping Only at 10GHz (BLT1)

BLT w/ Pre-Warping @ 10 GHz (BLT1)		±
Criterion	**Results**	
Pole Mapping Method	Poor mapping via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to warping	--
Zero Mapping Method	Poor mapping via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to warping	--
Starting Form for $H_a(s)$	$H_a(s)$ may be in pole-residue form	++
Magnitude Resp. Matching to $H_a(s)$	Poor; gain only matched at single frequency	--
Phase Response Matching to $H_a(s)$	Poor	--
Time Response Matching to $H_a(s)$	Poor	--

Table 9-3: BLT w/ Pre-Warping @ 10 GHz Attributes

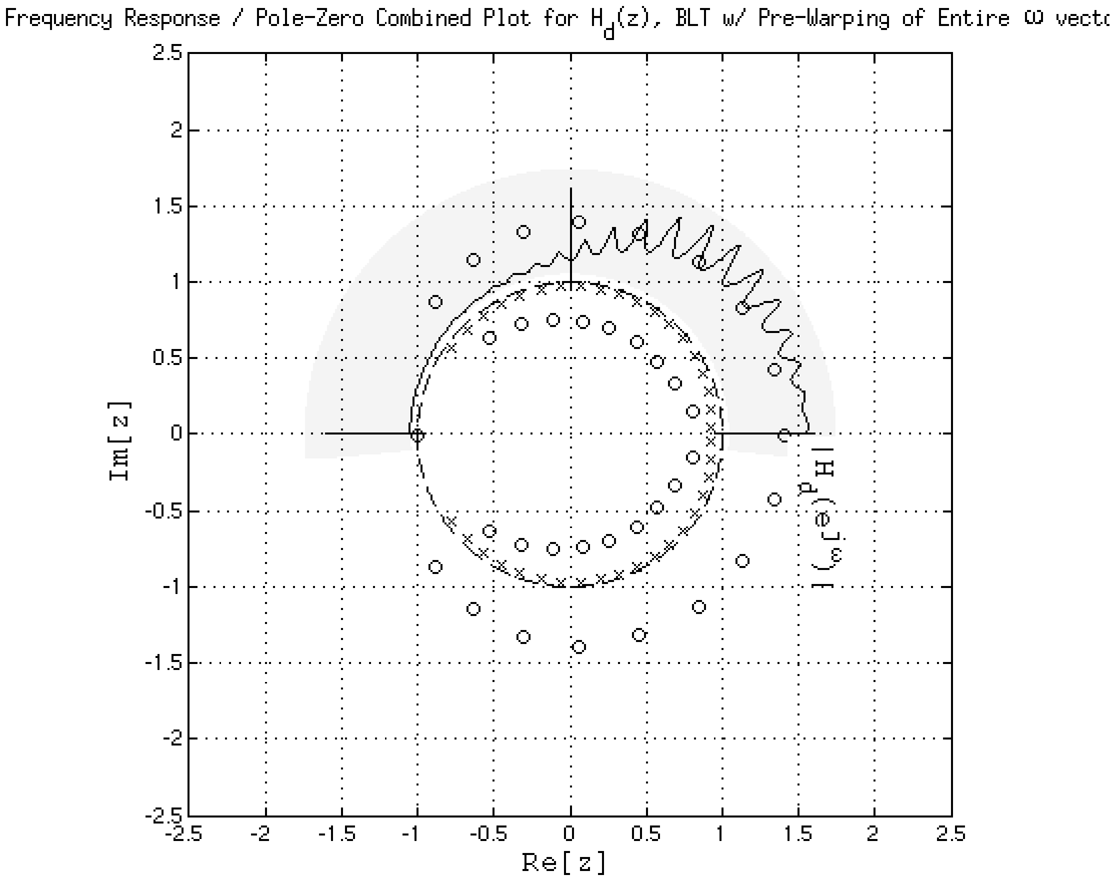


Figure 9-7: Combined Pole-Zero / Frequency Response Plot for Bilinear Transformation with Pre-Warping of Entire ω Vector (Ikai Method) (BLT2)

BLT w/ Pre-Warping of Entire ω Vector (Ikai Method) (BLT2)		±
Criterion	**Results**	
Pole Mapping Method	Excellent map via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to pre-warping	++
Zero Mapping Method	Good map via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to pre-warping; outlier zeros	+
Starting Form for $H_a(s)$	Required fitting to a prewarped version of $H'_a(s')$	--
Magnitude Resp. Matching to $H_a(s)$	Excellent	++
Phase Response Matching to $H_a(s)$	Excellent	++
Time Response Matching to $H_a(s)$	Very Good	+

Table 9-4: BLT w/ Pre-Warping of Entire ω Vector (Ikai Method) Attributes

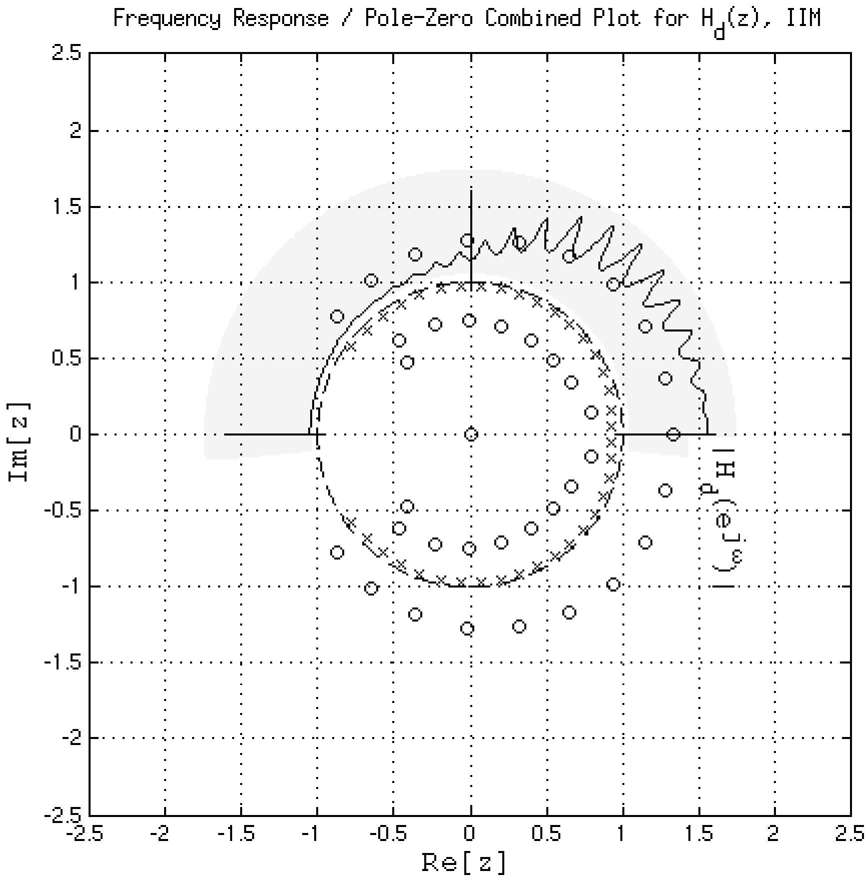


Figure 9-8: Combined Pole-Zero / Frequency Response Plot for Impulse Invariant Transformation (IIT)

Impulse Invariant Transform (IIT)		±
Criterion	**Results**	
Pole Mapping Method	Map via $z = e^{sT_s}$	++
Zero Mapping Method	Poor mapping of zeros to z-plane	--
Starting Form for $H_a(s)$	$H_a(s)$ may be in pole-residue form	++
Magnitude Resp. Matching to $H_a(s)$	Excellent	++
Phase Response Matching to $H_a(s)$	Excellent	++
Time Response Matching to $H_a(s)$	Very Good	+

Table 9-5: IIT Attributes

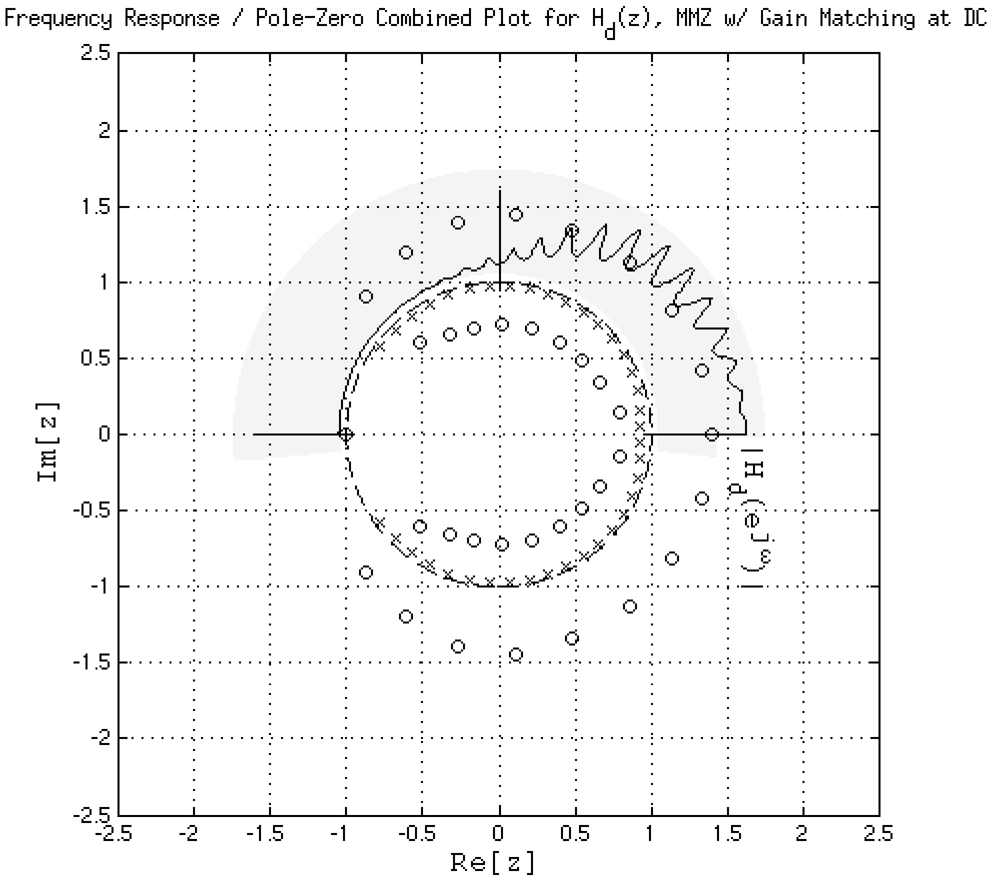


Figure 9-9: Combined Pole-Zero / Frequency Response Plot for Modified Matched-Z Transformation w/ Gain Match @ DC (MMZT1)

Modified Matched-Z Transform with Gain Matching @ DC (MMZT1)		±
Criterion	**Results**	
Pole Mapping Method	Map via $z = e^{sT_s}$	++
Zero Mapping Method	Map applied to 1^{st}-order Sections Zero of $H_{ak}(s)$ at ∞ maps to $z = -1$; Gain constant K_{dk} matched to K_{ak} at DC	+
Starting Form for $H_a(s)$	$H_a(s)$ may be in pole-residue form	++
Magnitude Resp. Matching to $H_a(s)$	Fair; gain droops when nearing f_N	-
Phase Response Matching to $H_a(s)$	Very good	+
Time Response Matching to $H_a(s)$	Very good	+

Table 9-6: MMZT with Gain Matching @ DC Attributes

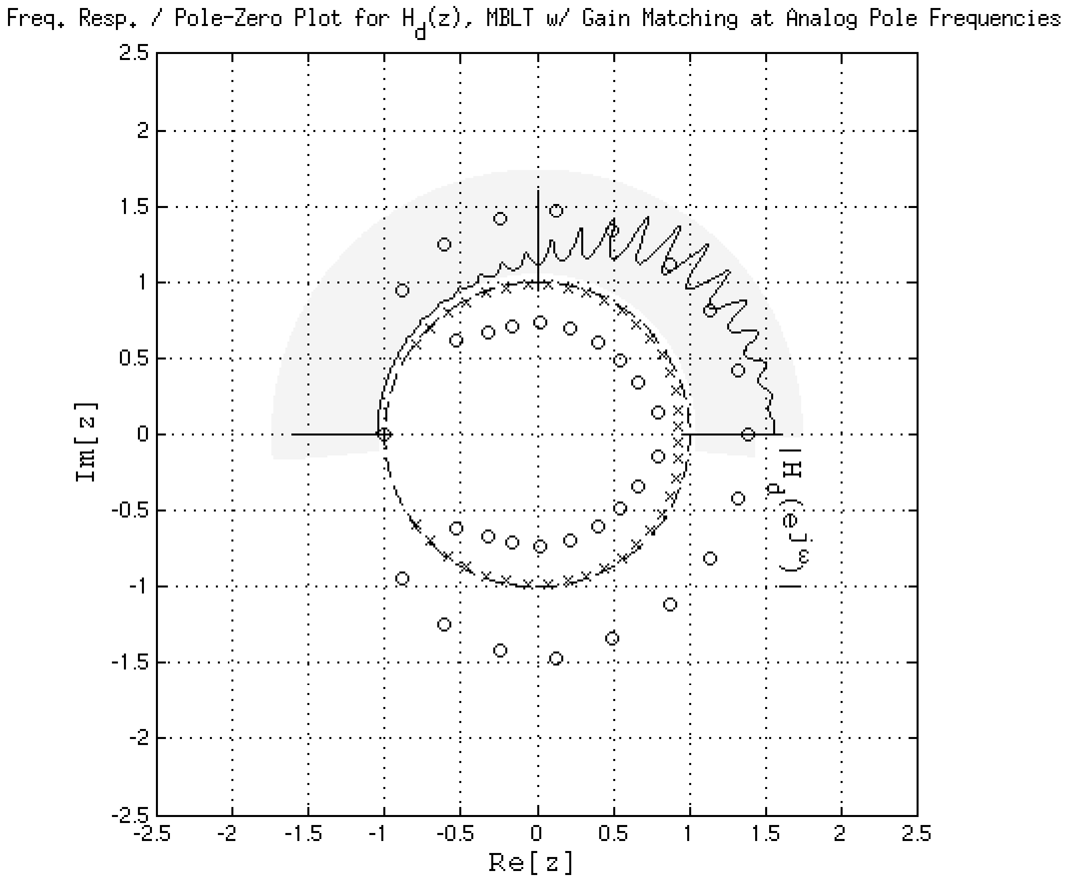


Figure 9-10: Combined Pole-Zero / Frequency Response Plot for Modified Bilinear Transformation (MBLT)

Modified Bilinear Transform (MBLT)		±
Criterion	**Results**	
Pole Mapping Method	Excellent map via $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to matching response at analog pole frequencies	++
Zero Mapping Method	Good mapping using $s = \frac{2}{T_s}\frac{z-1}{z+1}$ due to matching resp. at analog pole frequencies; outlier zeros	+
Starting Form for $H_a(s)$	$H_a(s)$ may be in pole-residue form	++
Magnitude Resp. Matching to $H_a(s)$	Fair/Good; the response matches at analog pole frequencies, but not between those frequencies	
Phase Response Matching to $H_a(s)$	Excellent	++
Time Response Matching to $H_a(s)$	Very Good	+

Table 9-7: MBLT Attributes

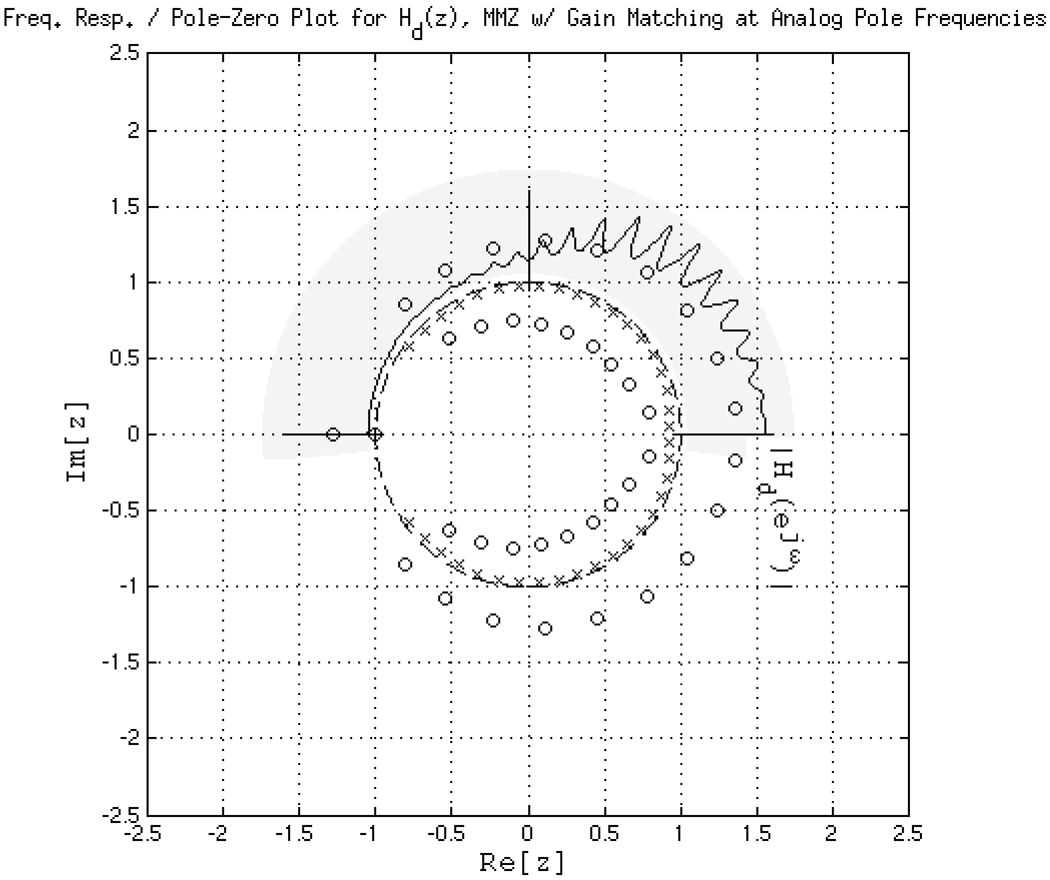


Figure 9-11: Combined Pole-Zero / Frequency Response Plot for Modified Matched-Z Transform with Gain Match at Analog Pole Frequencies (MMZT2)

Modified Matched-Z Transform with Gain Matching at Analog Pole Frequencies (MMZT2)		±
Criterion	**Results**	
Pole Mapping Method	Map via $z = e^{sT_s}$	++
Zero Mapping Method	Map applied to 1st-order Sections Zero of $H_{ak}(s)$ at ∞ maps to $z = -1$; Gain constant K_{dk} matched to K_{ak} at the analog pole frequency	++
Starting Form for $H_a(s)$	$H_a(s)$ may be in pole-residue form	++
Magnitude Resp. Matching to $H_a(s)$	Excellent	++
Phase Response Matching to $H_a(s)$	Excellent	++
Time Response Matching to $H_a(s)$	Very good	+

Table 9-8: MMZT with Gain Matching at Analog Pole Frequencies Attributes

Summary of Key Attributes for all s- to z-Domain Transform Utilized in this Project

Criterion	**MZT Matched-Z Transform**	**BLT0 BLT w/ No Pre-Warping**	**BLT1 BLT w/ Pre-Warping at 10 GHz**	**BLT2 Pre-Warping of Entire ω- Vector**	**IIT Impulse Invariant Transform**	**MMZT1 $K_{ak} = K_{dk}$ at $s = 0$ (DC)**	**MBLT**	**MMZT2 $K_{ak} = K_{dk}$ at $s = a_k$ frequencies**
Pole Mapping Method	++	--	--	++	++	++	++	++
Zero Mapping Method	++	--	--	+	--	+	+	++
Starting Form for $H_a(s)$	-	++	++	--	++	++	++	++
Magnitude Resp. Matching to $H_a(s)$	-	--	--	++	++	-	neutral	++
Phase Response Matching to $H_a(s)$	++	-	--	++	++	+	++	++
Time Response Matching to $H_a(s)$	+	-	--	+	+	+	+	+
Overall Transform Score[25]	**5**	**-6**	**-8**	**6**	**7**	**6**	**8**	**11**

Table 9-9: Summary of Key Attributes for All s- to z-Domain Transforms Utilized in This Research Project

---

[25] Scoring is determined as follows:
A rating of ‘++’ = 2 points.
A rating of ‘+’ = 1 point.
A rating of ‘neutral’ = 0 points.
A rating of ‘-‘ = -1 point.
A rating of ‘- -‘ = -2 points.

As Table 9-9 above depicts, the most effective transforms, as listed in order of their overall transform score, are as follows:

1. MMZT with 1st-order section gains K_{dk} and K_{ak} matched at the analog pole frequencies (MMZT2);
2. MBLT;
3. IIT;
4. BLT using the Ikai method (BLT2);
5. MMZT with gains matched at DC (MMZT1);
6. Traditional Matched-Z transform (MZT);
7. BLT with no pre-warping applied (BLT0);
8. BLT with pre-warping applied only at 10 GHz (BLT1).

The scores for each transformation as listed in Table 9-9 should be considered to be a general ranking, and a particular application may warrant a unique choice of transform. This new transform assessment procedure can now be incorporated as an additional tool in the overall signal integrity engineering design flow. **The criteria of Table 9-9 has been published internally at IBM Corp., and is under evaluation for use in its industrial signal integrity design processes** [97]**.**

The next section will describe how this procedure could be integrated into the design flows of Figure 9-2 and Figure 9-3, with the end result being an enhanced methodology that offers great potential for improving the quality and efficiency of the signal integrity engineering activities associated with complex system designs.

9.4 Optimized Signal Integrity Engineering Process Flow Incorporating Discrete Frequency Domain Techniques

The signal integrity processes outlined in Figure 9-1 through Figure 9-3 are characterized by a complete emphasis on continuous-time domain and analog-frequency domain techniques for the final evaluation of the system. Further, the frequency-domain analysis tends to be limited to interpretation of the scattering matrices of the system channel, which generally does not account for the system's source and load characteristics. The source and load characteristics can be included as an inherent part of the transfer functions $H_a(s)$ and $H_d(z)$, if desired. It is clear from the discussions of the previous chapters that much insight is gained by the study of the system's poles and zeros in the complex s-plane and z-plane, and the methodologies of Figure 9-1 through Figure 9-3 could clearly be augmented by adding these processes to the overall methodology.

Figure 9-12 shows a proposed, enhanced process for a signal integrity engineering design methodology which is augmented by analysis steps from the discrete frequency domain. Note the similarity of the upper part of the diagram with that of Figure 9-3, but now the lower part of the diagram shows the addition of the s- to z-domain transformation assessment steps and the subsequent pole-zero mapping steps in the complex s-plane and z-plane. For the new process, not only are the time-domain waveforms examined, but also the s-plane and z-plane maps.

Optionally, additional time-domain waveforms can be generated independently using $H_a(s)$, $H_d(z)$, or both. These waveforms can be calculated using the MATLAB function timeresp.m, which works directly with the partial fraction expansion form of $H_a(s)$, using the same process that was implemented in Chapter 5 [61]. Alternatively, the time response can be obtained using the MATLAB function lsim.m, which works with the rational function numerator or denominator polynomials of either $H_d(z)$ or $H_a(s)$ [80]. There are now as many as three sets of time-domain data to work with, consisting of that generated by SPICE, that generated from $H_a(s)$, and that generated from $H_d(z)$. In addition, there are two sets of complex frequency domain data, namely, the analog and digital transfer functions themselves. In the old methodology, only the time-domain data generated by SPICE was available.

It should be noted that the new steps in the process of Figure 9-12 are implemented outside of the SPICE simulation loop. Even though it takes extra initial effort to perform these steps, the simulation steps are relatively fast from a computational standpoint, and the additional insight gained from looking at the pole-zero maps and associated frequency responses should allow for a faster optimization of the final design solution. Indeed, the true advantage of the new process of Figure 9-12 is that it might save iterations on the SPICE simulation process, which is computationally intensive.

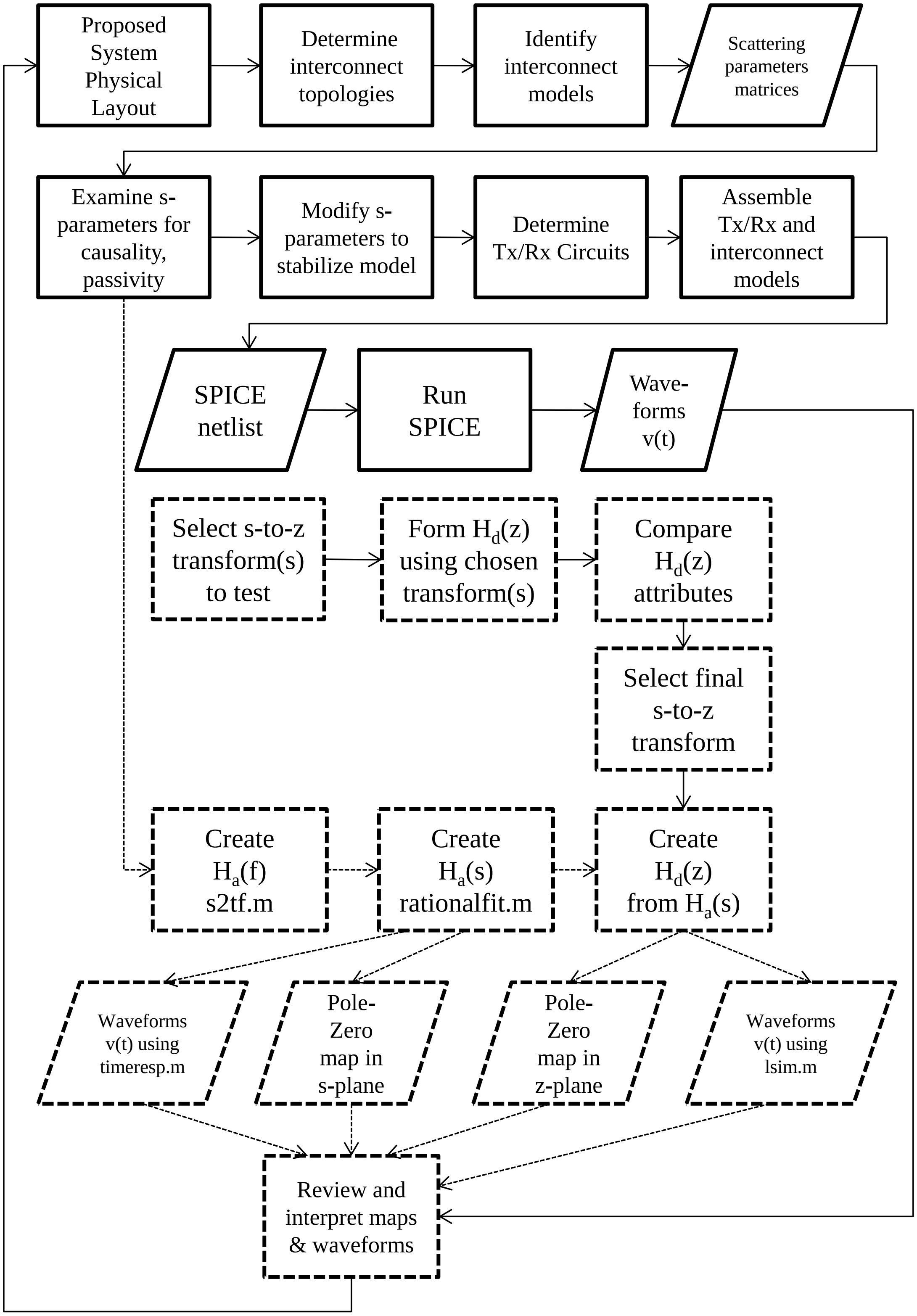


Figure 9-12: Enhanced Comprehensive Signal Integrity Design Methodology Incorporating Discrete Frequency Domain Methods (New Process Steps Shown in Dashed Boxes)

The enhanced methodology of Figure 9-12 has been published internally at IBM Corp., and is under evaluation for its adoption for use in its industrial design processes [98]**.**

9.5 Conclusions Regarding Applicability to an Industrial Signal Integrity Design Process

This chapter has presented two major results which highlight the great potential for improving the state of signal integrity engineering design methodologies. The first was a process for systematic assessment and ranking of the major attributes of the eight s- to z-domain transformation methods used throughout this research project to calculate the discrete transfer function $H_d(z)$ from the analog transfer function $H_a(s)$. These eight methods include three variations of the widely known bilinear transformation (BLT), including the patented method by Ikai [91]. Also included among the eight are the well-known impulse invariant transformation (IIT), which dominates the field of digital control theory, and the matched-Z transformation (MZT), which is widely used in the design of digital filters in the field of digital signal processing. The remaining three transforms, the modified bilinear transform (MBLT), and two variations of the modified matched-Z transform (MMZT), were developed in the course of this project, and were optimized for use in signal integrity engineering applications.

More importantly, however, the summary presented in Table 9-9 provides the basis for an evaluation methodology that allows systematic assessment of the transforms for use in signal integrity engineering applications, as part of the overall signal integrity engineering design flow. This enhanced signal integrity design flow is the second major result. It is clear from the process flow diagram of Figure 9-12 that utilization of discrete complex frequency domain techniques, centered in the z-plane, provide both an additional viewpoint from which to assess the system (the discrete frequency z- and ω-domains) and additional time-domain data with which to validate the original SPICE time-domain solution. Even better, these new process steps can be implemented in parallel with the traditional SPICE process, and thus can be executed independently of the original design flow to whatever degree of effort and redundancy is desired. This is a great advantage because it does not have the potential to disrupt the original flow, but rather purely augments it.

The summary of Table 9-9 and the methodology of Figure 9-12, along with the development of the MBLT and MMZT in Chapter 7, the combined pole-zero graphical analysis technique developed in Chapter 8, and the frequency scaling technique developed in Chapter 5 to enable conversion from the partial fraction expansion form to pole-zero factored form of $H_a(s)$, are the major results of this research project. They represent a successful attempt to transfer and extend discrete frequency-domain methods commonly used in two distinct fields of electrical engineering (digital control systems engineering and digital signal processing engineering) to a third field, signal integrity engineering, which has been historically dominated by continuous time-domain methods and analog frequency-domain methods.

Chapter 10: Conclusions and Future Work

10.1 Summary of Contributions

For this project, several new contributions have been made to two subfields of electrical engineering, signal integrity engineering and digital signal processing engineering. In addition, several of the new methods have potential application within other subfields of electrical engineering.

10.1.1 Methods Applicable to the Field of Signal Integrity Engineering

The stated goal for the project was to transfer and adapt various methods from the subfield of digital signal processing to the subfield of signal integrity engineering. To this end, the new methods emphasized the use of complex frequency domain pole-zero mapping techniques, and highlighted the advantages of the z-plane over the s-plane. From this standpoint, the project has made several significant contributions to the subfield of signal integrity engineering:

1) the comprehensive, self-contained summary of the various theoretical foundations of the subfields of signal integrity engineering and digital signal processing engineering which are particularly well-suited for technology transfer between the two subfields, as described in Chapter 3, and to be used internally by the signal integrity community within IBM Corp.;
2) a comprehensive summary and example application of the process for converting a two-port scattering matrix $\underline{S}$ to an analog frequency transfer function $H_a(j\Omega)$, as described in Chapter 4, and for which the MATLAB code will be used internally by the signal integrity community within IBM Corp.;
3) a comprehensive summary and example application of the process for converting an analog frequency transfer function $H_a(j\Omega)$ to a complex analog frequency transfer function $H_a(s)$ using vector fitting, as described in Chapter 5, and for which the MATLAB code will be used internally by the signal integrity community within IBM Corp.;

4) the development of a new process for converting the transfer function $H_a(s)$ from partial fraction expansion form to pole-zero factored form utilizing low-pass to low-pass filter transformations, as described in Chapter 5, **and published in the public domain by IBM Corp. as a technical disclosure publication** [89];
5) the application of the widely-used bilinear transformation (BLT), matched-Z transformation (MZT), and impulse invariant transformation (IIT) to signal integrity applications, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [94];
6) the development of the new graphical analysis technique by which the z-plane map of $H_d(z)$ and the frequency response $H_d(e^{j\omega})$ are plotted on the same diagram, as described in Chapter 8, **to be submitted for publication in the public domain as IEEE peer-reviewed journal articles** [94], [95];
7) the development of the modified bilinear transform (MBLT) to first map the 1st-order partial fraction terms from the overall partial fraction expansion of $H_a(s)$ to the z-plane, then combine the resulting overall partial fraction expansion of $H_d(z)$ to a factored pole-zero form of the numerator and denominator polynomials, as described in Chapter 7, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [95];
8) the development of the modified matched-Z transform (MMZT) to first map the 1st-order partial fraction terms from the overall partial fraction expansion of $H_a(s)$ to the z-plane, then combine the resulting overall partial fraction expansion of $H_d(z)$ to a factored pole-zero form of the numerator and denominator polynomials, as described in Chapter 7, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [95];
9) the development of the assessment criteria and process for assessment of the various s- to z- transformations when used for signal integrity engineering applications, as described in Chapter 9, **published internally within IBM Corp., and to be used internally by the signal integrity community within IBM Corp.** [97]; and

10) the enhanced signal integrity engineering design methodology by which the system model and transfer functions $H_a(s)$ and $H_d(z)$ are examined in the complex frequency domains, and the new insight is used to more efficiently iterate on the final design solution, as described in Chapter 9, **published internally within IBM Corp., and to be used internally by the signal integrity community within IBM Corp.** [98]**.**

Some of the new contributions described in this subsection have applications to the subfield of digital signal processing engineering; these will be described in the next section.

10.1.2 Methods Applicable to the Field of Digital Signal Processing Engineering

A beneficial result of developing new contributions to the subfield of signal integrity engineering utilizing digital signal processing methods is the enhancement of some of those same methods for the subfield of DSP itself. Those contributions are listed again here:

1) the comprehensive, self-contained summary of the various theoretical foundations of the subfields of signal integrity engineering and digital signal processing engineering which are particularly well-suited for technology transfer between the two subfields, as described in Chapter 3, and to be used internally by the signal integrity community within IBM Corp., applies equally as well to the subfield of digital signal processing engineering;
2) the comprehensive summary and example application of the process for converting an analog frequency transfer function $H_a(j\Omega)$ to a complex analog frequency transfer function $H_a(s)$ using vector fitting, as described in Chapter 5, and for which the MATLAB code will be used internally by the signal integrity community within IBM Corp., is directly applicable in the subfield of digital signal processing engineering, for the case where a frequency response is the only data available;
3) the development of a new process for converting the transfer function $H_a(s)$ from partial fraction expansion form to pole-zero factored form utilizing low-pass to low-pass filter transformations, as described in Chapter 5, **and published in the**

public domain by IBM Corp. as a technical disclosure publication [89], applies equally as well to the subfield of digital signal processing engineering;

4) the application of the widely-used bilinear transformation (BLT), matched-Z transformation (MZT), and impulse invariant transformation (IIT) to signal integrity applications, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [94], applies equally as well to the subfield of digital signal processing engineering;
5) the development of the new graphical analysis technique by which the z-plane map of $H_d(z)$ and the frequency response $H_d(e^{j\omega})$ are plotted on the same diagram, as described in Chapter 8, **to be submitted for publication in the public domain as IEEE peer-reviewed journal articles** [94], [95], applies equally as well to the subfield of digital signal processing engineering;
6) the development of the modified bilinear transform (MBLT) to first map the 1st-order partial fraction terms from the overall partial fraction expansion of $H_a(s)$ to the z-plane, then combine the resulting overall partial fraction expansion of $H_d(z)$ to a factored pole-zero form of the numerator and denominator polynomials, as described in Chapter 7, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [95], applies equally as well to the subfield of digital signal processing engineering;
7) the development of the modified matched-Z (MMZT) transform to first map the 1st-order partial fraction terms from the overall partial fraction expansion of $H_a(s)$ to the z-plane, then combine the resulting overall partial fraction expansion of $H_d(z)$ to a factored pole-zero form of the numerator and denominator polynomials, as described in Chapter 7, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [95], applies equally as well to the subfield of digital signal processing engineering; and
8) the development of the assessment criteria and process for assessment of the various s- to z- transformations when used for signal integrity engineering applications, as described in Chapter 9, **published internally within IBM Corp., and to be used internally by the signal integrity community within IBM Corp.**

[97], applies equally as well to the subfield of digital signal processing engineering.

Some of the new contributions described in this subsection and in the previous subsection have applications to other subfields of electrical engineering; these applications may or not be immediately recognized by practitioners in those subfields. These applications will be described in the next section.

10.1.3 Methods Applicable to Fields Outside of Signal Integrity Engineering

Another beneficial, and perhaps somewhat unexpected, result of developing new contributions to the subfield of signal integrity engineering utilizing digital signal processing methods is the applicability of these methods for other subfields of electrical engineering. Some suggested applications are listed here:

1) the comprehensive summary and example application of the process for converting an analog frequency transfer function $H_a(j\Omega)$ to a complex analog frequency transfer function $H_a(s)$ using vector fitting, as described in Chapter 5, and for which the MATLAB code will be used internally by the signal integrity community within IBM Corp., is directly applicable in the subfield of analog control engineering, for the case where a frequency response is the only data available;
2) the development of a new process for converting the transfer function $H_a(s)$ from partial fraction expansion form to pole-zero factored form utilizing low-pass to low-pass filter transformations, as described in Chapter 5, **and published in the public domain by IBM Corp. as a technical disclosure publication** [89], is directly applicable in the subfields of analog and digital control engineering;
3) the application of the widely-used bilinear transformation (BLT), matched-Z transformation (MZT), and impulse invariant transformation (IIT) to signal integrity applications, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [94], applies equally as well to the subfield of digital control engineering;

4) the development of the new graphical analysis technique by which the z-plane map of $H_d(z)$ and the frequency response $H_d(e^{j\omega})$ are plotted on the same diagram, as described in Chapter 8, **to be submitted for publication in the public domain as IEEE peer-reviewed journal articles** [94], [95], applies equally as well to the subfield of digital control engineering;
5) the development of the modified bilinear transform (MBLT) to first map the 1st-order partial fraction terms from the overall partial fraction expansion of $H_a(s)$ to the z-plane, then combine the resulting overall partial fraction expansion of $H_d(z)$ to a factored pole-zero form of the numerator and denominator polynomials, as described in Chapter 7, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [95], applies equally as well to the subfield of digital control engineering; and
6) the development of the modified matched-Z (MMZT) transform to first map the 1st-order partial fraction terms from the overall partial fraction expansion of $H_a(s)$ to the z-plane, then combine the resulting overall partial fraction expansion of $H_d(z)$ to a factored pole-zero form of the numerator and denominator polynomials, as described in Chapter 7, **to be submitted for publication in the public domain as an IEEE peer-reviewed journal article** [95], applies equally as well to the subfield of digital control engineering; and
7) the development of the assessment criteria and process for assessment of the various s- to z- transformations when used for signal integrity engineering applications, as described in Chapter 9, **published internally within IBM Corp., and to be used internally by the signal integrity community within IBM Corp.** [97], applies equally as well to the subfield of digital control systems engineering.

It may also be possible to extend these methods to the subfield of image processing, which can be considered as two-dimensional digital signal processing, with the dependent variables being in spatial-domain dimensions, as opposed to time-domain dimensions [43].

10.2 Known Limitations

The methods developed for this project, and their associated applications, are novel to the signal integrity engineering and digital signal processing subfields. Nevertheless, there are some known limitations of these methods, especially regarding their scope and depth of use. Although some of these limitations are due to fundamental limits imposed by the governing physical or mathematical laws, in most cases the limitations are simply to the scope of the examples selected as representative test cases. Some of these known limitations to the scope of this work are listed in the next section for the purposes of setting the context for potential future work.

10.2.1 Random vs. Deterministic Channels

One fundamental characteristic of the types of channels generally found within computer hardware is that the channels consist of guided wave structures such as transmission lines (widely used in printed circuit boards), or dielectric waveguides and fiber optic cables (used in advanced applications). In virtually every practical case these guided wave channels can be considered as deterministic, as opposed to stochastic. The latter would be found commonly with wireless communications applications, for example. Thus the propagation of signals in these guided wave channels is very predictable, as the channel characteristics do not vary with time due to weather conditions, etc., and the use of stochastic methods is generally not required.

10.2.2 Least Squares Estimation vs. other Estimation Methods

One attribute of the deterministic channel assumption is that estimation methods generally considered for stochastic systems, such as the maximum-likelihood estimator (MLE) [99], [100], are not generally required, and the optimization routines applied to the system identification and vector fitting processes can be least-squares error (LSE) based. LSE estimation methods were used in the ARX system identification methods and vector fitting methods demonstrated in Chapters 4 and 5, for example. The assumption of LSE based methods worked well for this project, but other optimization criteria could be used and may be optimal for other applications.

10.3 Future Directions

As opposed to the non-applicability of some methods, such as the stochastic methods described in the previous section, there are other methods available that would be natural extensions to the work done for this project, due primarily to nature of the systems generally being considered in signal integrity engineering. One such characteristic of a typical system is its potential to be treated as a multi-port system. This can manifest itself in at least two forms, that that being a system utilizing differential signaling and thus differential transmission line structures, and the other being the undesirable effect of coupled noise between transmission line structures. Both of these scenarios result in the system being more appropriately considered as a multi-port system, or even a multiple input, multiple output (MIMO) system.

10.3.1 Differential vs. Single-Ended Signaling, Interconnect Structures, and Scattering Parameters

For the case of differential signaling, the transmission line structures are generally optimized so that both lines of the differential pair structure see the same noise environment. This has the effect of canceling out the noise, as a differential signal is intentionally launched as a voltage waveform between the two signal conductors.

From a practical standpoint, the biggest impact of a differential system is that it takes a four-port scattering matrix to describe the system using conventional scattering parameters. This limits the use of the s2tf.m MATLAB function [58], [76] to convert from scattering parameters to the system's analog transfer function $H_a(s)$. One potential way around this problem would be to convert the conventional scattering parameters to differential scattering parameters. This should enable the use of s2tf.m directly, but this has not been tried for this project, and thus should be considered a future extension.

Note that, even though the differential system requires a four-port matrix of conventional scattering parameters, the system can still be considered to be a single-input, single-output (SISO) system, since there is still a single input signal and a single output signal under consideration. It is only the description of the transmission line structure which has become more complex.

10.3.2 Inclusion of Coupled Noise Attributes and Multiple-Input Multiple-Output Systems

In signal integrity engineering, the inclusion of coupled noise effects is usually addressed by considering a multiple-input, multiple-output (MIMO) system [7], and is a significantly more difficult problem than a simple SISO differential system. Generally speaking, a coupled noise simulation requires both an N-port scattering parameter network, and the consideration of multiple input stimuli and output response signals. Generally, transfer function methods based on functions such as $H_a(s)$ and $H_d(z)$ are limited to SISO systems, and MIMO systems must be handled by state space methods [40]. Accordingly, the problem of coupled noise analysis can only be handled by state space methods, and has not been addressed in this project.

10.3.3 Formulation of Stability, Causality, and Passivity Conditions for System Transfer Functions in the z-Domain

Finally, another obvious extension to this project would be increased usage of the pole-zero maps in the z-plane to assess the system stability, causality, and passivity characteristics directly. The study of these characteristics in the s-plane is an active area of research in signal integrity engineering [63], [101], [102]. It is known, for example, that causality can be determined in the z-plane when the region of convergence is also known [42]. Extension and reformulation of these methods to the z-plane would be an obvious future direction for this research.

Bibliography

[1] H. W. Johnson and M. Graham, *High-speed digital design: a handbook of black magic*. Englewood Cliffs, N.J.: Prentice Hall, 1993.

[2] B. Young, *Digital signal integrity: modeling and simulation with interconnects and packages*. Upper Saddle River, NJ: Prentice Hall, 2001.

[3] G. K. Bartley and P. E. Dahlen, “The electrical challenges of packaging the IBM AS/400,” in *IEEE 6th Topical Meeting on Electrical Performance of Electronic Packaging, 1997*, 1997, pp. 3–4.

[4] H. W. Johnson and M. Graham, *High-speed signal propagation : advanced black magic*. Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[5] R. E. Matick, *Transmission lines for digital and communication networks : an introduction to transmission lines, high-frequency and high-speed pulse characteristics and applications*. New York: IEEE Press, 1995.

[6] R. A. Chipman, *Schaum’s outline of theory and problems of transmission lines*. New York: McGraw-Hill, 1968.

[7] R. R. Tummala and E. J. Rymaszewski, *Microelectronics packaging handbook*. New York: Chapman & Hall, 1997.

[8] A. D. Chen, D. Freeman, and B. H. Leitao, “IBM Redbooks | IBM Power 770 and 780 Technical Overview and Introduction,” 06-Jun-2013. [Online]. Available: http://www.redbooks.ibm.com/abstracts/redp4924.html?Open&cm_sp=MTE27984. [Accessed: 06-Nov-2013].

[9] P. E. Dahlen, T. Timpane, D. J. Becker, T. W. Liang, W. D. Martin, P. Rudrud, and G. K. Bartley, “Maintaining System Signal and Power Integrity Characteristics as Part of a Module Cost-Reduction Exercise,” in *2007 IEEE Electrical Performance of Electronic Packaging*, 2007, pp. 83–86.

[10] “IBM Intellistation POWER 185 Express Workstation, Product Brief IND01659-USEN-01,” *IBM Corp*, p. 5, 2006.

[11] “IBM System p5 185 Express Server, Product Brief PSD01600-USEN-01,” p. 5, 2006.

[12] Q. Dino, B. Kerry, and C. Puneet, “IBM Redbooks | IBM Power Systems 775 for AIX and Linux HPC Solution,” 19-Oct-2012. [Online]. Available: http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248003.html?Open. [Accessed: 06-Nov-2013].

[13] J. Tate, D. Bryant, and C. Burns, “IBM Redbooks | Implementing the IBM SAN Volume Controller and FlashSystem 820,” 06-Sep-2013. [Online]. Available: http://www.redbooks.ibm.com/abstracts/sg248172.html?Open. [Accessed: 06-Nov-2013].

[14] G. Edlund, *Timing Analysis and Simulation for Signal Integrity Engineers (Prentice Hall Modern Semiconductor Design Series)*, 1st edition. Prentice Hall, 2007.

[15] P. E. Dahlen, R. J. Gravrok, D. L. Heckmann, and M. O. Maxson, “Transmission line bounding models,” U.S. Patent US7010768 B2, 07-Mar-2006.

[16] A. Deutsch, G. V. Kopcsay, P. W. Coteus, C. W. Surovic, P. E. Dahlen, D. L. Heckmann, and D.-W. Duan, “Frequency-dependent losses on high-performance interconnections,” *IEEE Trans. Electromagn. Compat.*, vol. 43, no. 4, pp. 446–465, Nov. 2001.

[17] A. Deutsch, G. V. Kopcsay, P. W. Coteus, C. W. Surovic, P. Dahlen, D. L. Heckmann, and D. W. Duan, “Bandwidth prediction for high-performance interconnections,” in *50th Electronic Components and Technology Conference*, Las Vegas, NV, USA, 2000, pp. 256–266.

[18] S. L. Syverson, D. L. Heckmann, T. J. Timpane, and P. E. Dahlen, “Evaluation of frequency-dependent transmission line model extraction methods based on laboratory measurements,” in *Electronic Components and Technology Conference, 2004. Proceedings. 54th*, 2004, vol. 2, pp. 1416–1419 Vol.2.

[19] D. M. Pozar, *Microwave engineering*. Reading, Mass.: Addison-Wesley, 1990.

[20] W. Weeks, A. Jimenez, G. Mahoney, D. Mehta, H. Qassemzadeh, and T. Scott, “Algorithms for ASTAP–A network-analysis program,” *Circuit Theory IEEE Trans. On*, vol. 20, no. 6, pp. 628–634, 1973.

[21] L. W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*. 1975.

[22] *ANSYS HFSS*. ANSYS, Inc. .” [Online]. Available: http://www.ansys.com [Accessed: 17-Feb-2014].

[23] *EMSIM*. IBM Corporation.

[24] *PROPCALC*. IBM Corporation.

[25] W. T. Weeks, “Calculation of Coefficients of Capacitance of Multiconductor Transmission Lines in the Presence of a Dielectric Interface,” *IEEE Trans. Microw. Theory Tech.*, vol. 18, no. 1, pp. 35–43, 1970.

[26] S. Grivet-Talocia, “Package macromodeling via time-domain vector fitting,” *IEEE Microw. Wirel. Compon. Lett.*, vol. 13, no. 11, pp. 472–474, 2003.

[27] R. R. Tummala, I. Microelectronics, P. Society, G. I. of T. P. R. Center, and P. & M. T. S. Components, *Proceedings, 3rd International Symposium on Advanced Packaging Materials : processes, properties and interfaces : Chateau Elan, Braselton, Georgia, March 9-12, 1997*. Reston, VA; Piscataway, NJ: Imaps; IEEE Service Center, 1997.

[28] *PowerSPICE User's Guide*, Version 1.5. Hopewell Junction, NY: IBM Corporation, 2005.

[29] Synopsis, *HSPICE User Guide: Basic Simulation and Analysis*, Version F-2011.09. 2011.

[30] "Cadence PSpice A/D and Advanced Analysis." [Online]. Available: http://www.cadence.com/products/pcb/spice/pages/default.aspx. [Accessed: 17-Feb-2014].

[31] "Cadence Spectre Circuit Simulator." [Online]. Available: http://www.cadence.com/products/cic/spectre_circuit/pages/default.aspx. [Accessed: 17-Feb-2014].

[32] I. B. M. Corporation, *IBM HSS System Level Simulator (HSSCDR)*, vol. 3.0.0 b35. 2012.

[33] "ANSYS DesignerSI." [Online]. Available: http://www.ansys.com/Products/Simulation+Technology/Electronics/Signal+Integrity/ANSYS+DesignerSI. [Accessed: 17-Feb-2014].

[34] "ADS 2013.06 | Agilent." [Online]. Available: http://www.home.agilent.com/en/pd-2289752/ads-201306?nid=-34346.1056823.00&cc=US&lc=eng. [Accessed: 17-Feb-2014].

[35] "Quantum Channel Designer | SiSoft." [Online]. Available: http://www.sisoft.com/products/quantum-channel-designer.html. [Accessed: 17-Feb-2014].

[36] A. Sanders, M. Reso, and J. D. Ambrosia, "Channel Compliance Testing Utilizing Novel Statistical Eye Methodology," in *DesignCon 2004*, 2004.

[37] "Cadence Allegro Sigrity Serial Link Analysis Option." [Online]. Available: http://www.cadence.com/products/sigrity/Allegro_Sigrity_Serial_Link_Analysis_Option/pages/default.aspx. [Accessed: 17-Feb-2014].

[38] J. D. Rockrohr, *High Speed Serdes Devices and Applications*. Boston, MA: Boston, MA: Springer-Verlag US, 2009.

[39] J. W. Nilsson, *Electric circuits*, 2nd ed. Reading, Mass.: Addison-Wesley Pub. Co., 1986.

[40] K. Ogata, *Modern Control Engineering*, 2nd edition. Prentice Hall Asia, 1990.

[41] S. K. Mitra, *Digital signal processing: a computer-based approach*, 2nd ed. Boston: McGraw-Hill/Irwin, 2001.

[42] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.

[43] H. H. Ottesen, *Digital Signal Processing I with MATLAB - Course Lecture Notes*. 2003.

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

[45] D. Deschrijver, B. Haegeman, and T. Dhaene, "Orthonormal Vector Fitting: A Robust Macromodeling Tool for Rational Approximation of Frequency Domain Responses," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 216–225, 2007.

[46] Y. S. Mekonnen and J. E. Schutt-Aine, "Broadband macromodeling of sampled frequency data using z-domain vector-fitting method," in *IEEE Workshop on Signal Propagation on Interconnects, 2007. SPI 2007*, 2007, pp. 45–48.

[47] B. Nouri, R. Achar, and M. S. Nakhla, "z-Domain Orthonormal Basis Functions for Physical System Identifications," *IEEE Trans. Adv. Packag.*, vol. 33, no. 1, pp. 293–307, 2010.

[48] G. F. Franklin and J. D. Powell, *Feedback control of dynamic systems*. Reading, Mass.: Addison-Wesley, 1994.

[49] G. F. Franklin and J. D. Powell, *Digital control of dynamic systems*. Reading, Mass.: Addison-Wesley Pub. Co., 1980.

[50] K. Gröchenig, *Foundations of time-frequency analysis: with 15 figures*. Boston: Birkhäuser, 2001.

[51] M. A. Pinsky, *Introduction to Fourier Analysis and Wavelets*. American Mathematical Soc., 2002.

[52] W. Li and M. R. Tucker, "Method for decomposing signals into efficient time-frequency representations for data compression and recognition," US5453945 A, 26-Sep-1995.

[53] M. A. Donarski, "Heyser spiral low frequency correction of FIR filters," US8306242 B2, 06-Nov-2012.

[54] S. Bachelor, R. Thompson, and J. Seawall, “Systems and methods implementing frequency-steered acoustic arrays for 2d and 3d imaging,” WO2005008272 A2, 27-Jan-2005.

[55] A. Taleb, “Low-Complexity Spectral Analysis/Synthesis Using Selectable Time Resolution,” US20100250265 A1, 30-Sep-2010.

[56] M. A. Meier, “Converted mode seismic survey design,” WO2008024150 A2, 28-Feb-2008.

[57] S. Ramo, T. V. Duzer, and J. R. Whinnery, *Fields and waves in communication electronics*. New York: Wiley, 1994.

[58] *s2tf.m Function Reference - MATLAB RF Toolbox Documentation*. The Mathworks, Inc.

[59] D. E. Johnson, *Introduction to filter theory*. Prentice-Hall, 1976.

[60] A. Ubolli and B. Gustavsen, “Comparison of Methods for Rational Approximation of Simulated Time-Domain Responses: ARMA, ZD-VF, and TD-VF,” *IEEE Trans. Power Deliv.*, vol. 26, no. 1, pp. 279–288, 2011.

[61] *timeresp.m Function Reference - MATLAB RF Toolbox Documentation*. The Mathworks, Inc.

[62] eCircuit Center Web Page, vol. 2010. [Online]. Available:
http://www.ecircuitcenter.com/SPICEtopics.htm .

[63] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, “Stability, Causality, and Passivity in Electrical Interconnect Models,” *IEEE Trans. Adv. Packag.*, vol. 30, no. 4, pp. 795–808, 2007.

[64] P. Optical Internetworking Forum and L. L. W. Group, *Common Electrical I/O (CEI) - Electrical and Jitter Interoperability agreements for 6G+ bps and 11G+ bps IO: IA # OIF-CEI-02.0*. 2005.

[65] L. Ljung, *System identification : theory for the user*. Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[66] S. M. Kay, *Fundamentals of statistical signal processing*. Englewood Cliffs, N.J: Prentice-Hall PTR, 1993.

[67] *arx.m Function Reference - MATLAB System Identification Toolbox Documentation*. The Mathworks, Inc.

[68] A. Semlyen and A. Dabuleanu, “Fast and accurate switching transient calculations on transmission lines with ground return using recursive convolutions,” *IEEE Trans. Power Appar. Syst.*, vol. 94, no. 2, pp. 561–571, 1975.

[69] *IdEM 2.4*. Dept. Electronics, Politecnico di Torino.

[70] M. R. Wohlers, *Lumped and distributed passive networks: a generalized and advanced viewpoint*. Academic Press, 1969.

[71] C. Warwick, “Understanding the Kramers-Kronig Relation Using A Pictorial Proof: White Paper.” Agilent Technologies, Inc., 31-Mar-2010.

[72] R. Achar and M. S. Nakhla, “Simulation of high-speed interconnects,” *Proc. IEEE*, vol. 89, no. 5, pp. 693–728, 2001.

[73] W. T. Beyene and J. Schutt-Aine, “Accurate frequency-domain modeling and efficient circuit simulation,” *Microw. Theory Tech. IEEE Trans. On*, vol. 45, no. 10, pp. 1941–1947, 1997.

[74] B. Gustavsen, “Time delay identification for transmission line modeling,” in *8th IEEE Workshop on Signal Propagation on Interconnects, 2004. Proceedings*, 2004, pp. 103–106.

[75] I. J. Bahl and P. Bhartia, *Microwave solid state circuit design*. Wiley, 1988.

[76] G. González, *Microwave transistor amplifiers: analysis and design*. Prentice Hall, 1997.

[77] P. J. Pupalaikis, “The Relationship Between Discrete- Frequency S-parameters and Continuous-Frequency Responses,” presented at the DesignCon 2012.

[78] *c2d.m Function Reference - MATLAB System Identification Toolbox Documentation*. The Mathworks, Inc.

[79] *impinvar.m Function Reference - MATLAB Signal Processing Toolbox Documentation*. The Mathworks, Inc.

[80] *lsim.m Function Reference - MATLAB Control Systems Toolbox Documentation*. The Mathworks, Inc.

[81] H. H. Ottesen, *Digital Signal Processing II with MATLAB - Course Lecture Notes*. 2003.

[82] Y. S. Mekonnen and J. E. Schutt-Aine, “Fast Macromodeling Technique of Sampled Time/Frequency Data Using z-domain Vector-Fitting Method,” in *2007 IEEE Electrical Performance of Electronic Packaging*, 2007, pp. 47–50.

[83] D. Deschrijver and T. Dhaene, “Broadband macromodelling of passive components using orthonormal vector fitting,” *Electron. Lett.*, vol. 41, no. 21, pp. 1160–1161, 2005.

[84] C.-U. Lei and N. Wong, "IIR Approximation of FIR Filters Via Discrete-Time Hybrid-Domain Vector Fitting," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 533–537, 2009.

[85] *rationalfit.m Function Reference - MATLAB RF Toolbox Documentation*. The Mathworks, Inc.

[86] F.-Y. Chang, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *Microw. Theory Tech. IEEE Trans. On*, vol. 37, no. 12, pp. 2028–2038, 1989.

[87] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, Jan. 1996.

[88] *residue.m Function Reference - MATLAB Documentation*. The Mathworks, Inc.

[89] P. E. Dahlen, "Utilization of Laplace-Domain Frequency Scaling Circuit Techniques to Enable Numerical Conversion of Transfer Functions from Pole-Residue Form to Pole-Zero Factored Form," *IPcom Prior Art Database*, no. IPCOM000234122D, Jan. 2014.

[90] *etfe.m Function Reference - MATLAB System Identification Toolbox Documentation*. The Mathworks, Inc.

[91] Y. Ikai, "Discretization processing method of transfer function in continuous time systems, system and program therefore, and compensator and feedback control system using the same," US6996592 B2, 07-Feb-2006.

[92] "Bilinear transform," *Wikipedia, the free encyclopedia*. 11-Aug-2013.

[93] A. Ubolli and B. Gustavsen, "Applicability of time domain and z-domain Vector Fitting to rational modeling from time domain responses with consideration to circuit solver integration method," in *2010 IEEE 14th Workshop on Signal Propagation on Interconnects (SPI)*, 2010, pp. 69–72.

[94] P. E. Dahlen and H. H. Ottesen, "Discrete-Time Signal Integrity Models Obtained Using s- to z-Domain Transforms: Part 1 - Well-Known Transforms," to be submitted for publication.

[95] P. E. Dahlen and H. H. Ottesen, "Discrete-Time Signal Integrity Models Obtained Using s- to z-Domain Transforms: Part 2 - Improved Transforms Optimized for Signal Integrity Applications," to be submitted for publication.

[96] "LTI system theory," *Wikipedia, the free encyclopedia*. 15-Oct-2013.

[97] P. E. Dahlen, "Summary of Key Attributes for Various s- to z-Domain Transforms for Signal Integrity Applications," IBM Corp. internal publication.

[98] P. E. Dahlen, “Enhanced Comprehensive Signal Integrity Design Methodology Incorporating Discrete Frequency Domain Methods,” IBM Corp. internal publication.

[99] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*. Prentice-Hall PTR, 1998.

[100] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall PTR, 1998.

[101] A. Chinea and S. Grivet-Talocia, “A Passivity Enforcement Scheme for Delay-Based Transmission Line Macromodels,” *IEEE Microw. Wirel. Compon. Lett.*, vol. 17, no. 8, pp. 562–564, 2007.

[102] R. Mandrekar and M. Swaminathan, “Causality enforcement in transient simulation of passive networks through delay extraction,” in *9th IEEE Workshop on Signal Propagation on Interconnects, 2005. Proceedings*, 2005, pp. 25–28.

Appendix 1: Summary of MATLAB Functions

This appendix contains descriptions of the common MATLAB functions used in this research project. The descriptions are from the MATLAB help documentation, as accessed from the MATLAB command window.

A1.1 MATLAB Function s2tf.m

```
s2tf Calculate transfer function from 2-port S-parameters
    TF = s2tf(S_PARAMS, Z0, ZS, ZL, OPTION) or TF =
    s2tf(S_OBJ, ZS, ZL,OPTION) calculates a voltage or
    power wave transfer function from 2-port scattering
    parameters defined by a 2x2xK complex numeric array
    S_PARAMS or by an 2-port sparameters object S_OBJ. s2tf
    uses one of 3 definitions, determined by the value of
    the OPTION    argument.

      OPTION = 1: the voltage transfer function from
      incident voltage to load voltage
      TF1 = VL/Va = ((ZS+conj(ZS))/conj(ZS)) *
        S21 * (1 + GAMMAL) * (1 - GAMMAS) /
        (2*(1 - S22 * GAMMAL) * (1 - GAMMAIN * GAMMAS))

    Va is the incident voltage which is the output voltage
    of the source when the input port is conjugately
    matched,
      Va = (conj(ZS)/(ZS+conj(ZS))) * VS

      OPTION = 2: the voltage transfer function from source
      voltage to load voltage
      TF2 = VL/VS = S21 * (1 + GAMMAL) * (1 - GAMMAS) /
      (2*(1 - S22 * GAMMAL) * (1 - GAMMAIN * GAMMAS))

    Here, VS is the source voltage, ZS is the source
    impedance, and VL is the output voltage over load
    impedance ZL.
```

```
  OPTION = 3: the power wave transfer function
  TF3 = BP2/AP1 = SQRT(RS*RL) * S21 * (1 + GAMMAL) *
  (1 - GAMMAS) /((1 - S22 * GAMMAL) *
  (1 - GAMMAIN * GAMMAS) * ZL)
```

BP2 is the transmitted power wave at the 2nd port and AP1 is the incident power wave to the 1st port, defined by

```
  BP2 = SQRT(RL)/ZL * VL; AP1 = VS/(2*SQRT(RS))
```

and RL = real(ZL), RS = real(ZS).

The reflection coefficients are defined as:

```
  GAMMAIN = S11 + (S12 * S21 * GAMMAL)/
  (1 - S22 * GAMMAL)
  GAMMAL  = (ZL - Z0)/(ZL + Z0)
  GAMMAS  = (ZS - Z0)/(ZS + Z0)
```

Z0 is the reference impedance of S-parameters. The default is 50 ohms. The default values of ZS and ZL are also 50 ohms. The default value of OPTION is 1.

Reference: Guillermo Gonzalez, Microwave Transistor Amplifiers: Analysis and Design, 2nd edition, Prentice Hall, 1996

See also powergain, rationalfit, snp2smp, gammain, gammaout, sparameters

A1.2 MATLAB Function timeresp.m

```
timeresp Compute the time response of a rational function
    object.
    [Y, T] = timeresp(H, U, TS) computes the output signal,
    Y, that the rational function object, H, produces in
    response to the given input signal U.

      Y(n) = SUM(C.*X(n - DELAY/TS)) + D*U(n - DELAY/TS)
      where  X(n+1) = F*X(n) + G*U(n), X(1) = 0 and
      F = EXP(A*TS),  G = (F-1) ./ A;

    A, C, DELAY and D are the properties of
    RFMODEL.RATIONAL object H:

              A: Complex vector of poles of the rational
              function
              C: Complex vector of residues of the rational
              function
              D: Scalar value specifying direct feedthrough
          DELAY: Delay time (s)

    H is the handle to the RFMODEL.RATIONAL object. U is
    the input signal.    TS is the sample time of U in
    seconds. Y is the output signal at corresponding time,
    T, in seconds.

    See also rfmodel.rational, rfmodel.rational/ispassive,
    rfmodel.rational/stepresp, rfmodel.rational/freqresp,
    rfmodel.rational/writeva, rationalfit
```

A1.3 MATLAB Function arx.m

```
arx  Compute least squares estimate of arx models.

   SYS = arx(DATA, ORDERS)
    estimates an arx model SYS using input-output or time
    series data DATA.
    SYS is an IDPOLY model representing the equation:
      A(q) y(t) = B(q) u(t-nk) + e(t)
    arx estimates the coefficients of A and B polynomials
    along with their standard deviations. The sample time
    of SYS is equal to that of DATA.

    DATA is time- or frequency domain estimation data
    represented by an IDDATA or IDFRD object. See HELP
    IDDATA or HELP IDFRD.

    ORDERS = [na nb nk], the orders of A and B polynomials
    in the arx model.
    For multi-output systems, ORDERS has as many rows as
    there are outputs na is then an ny|ny matrix whose i-j
    entry gives the order of the polynomial (in the delay
    operator) relating the j:th output to thei:th output.
    Similarly nb and nk are ny|nu matrices. (ny:# of
    outputs, nu:# of inputs). For a time series, ORDERS =
    na only.

See also arxOptions, arxstruc, ar, armax, bj, iv4, iv4,
n4sid, oe, nlarx.
```

A1.4 MATLAB Function c2d.m

```
c2d  Converts continuous-time dynamic system to discrete
time.

    SYSD = c2d(SYSC,TS,METHOD) computes a discrete-time
    model SYSD with sampling time TS that approximates the
    continuous-time model SYSC.
    The string METHOD selects the discretization method
    among the following:
       'zoh'       Zero-order hold on the inputs
       'foh'       Linear interpolation of inputs
       'impulse'   Impulse-invariant discretization
       'tustin'    Bilinear (Tustin) approximation.
       'matched'   Matched pole-zero method (for SISO
        systems only).
    The default is 'zoh' when METHOD is omitted. The
    sampling time TS should be specified in the time units
    of SYSC (see "TimeUnit" property).

    c2d(SYSC,TS,OPTIONS) gives access to additional
    discretization options. Use C2DOPTIONS to create and
    configure the option set OPTIONS. For example, you can
    specify a prewarping frequency for the Tustin method
    by:
       opt =
       c2dOptions('Method','tustin','PrewarpFrequency',.5);
       sysd = c2d(sysc,.1,opt);

    For state-space models,
       [SYSD,G] = c2d(SYSC,Ts,METHOD)
    also returns the matrix G mapping the states xc(t) of
    SYSC to the states xd[k] of SYSD:
       xd[k] = G * [xc(k*Ts) ; u[k]]
    Given an initial condition x0 for SYSC and an initial
    input value u0=u(0), the equivalent initial condition
    for SYSD is (assuming u(t)=0 for t<0):
       xd[0] = G * [x0;u0] .


    See also c2dOptions, d2c, d2d, DynamicSystem.
```

A1.5 MATLAB Function impinvar.m

```
impinvar Impulse invariance method for analog to digital
filter conversion.
    [BZ,AZ] = impinvar(B,A,Fs) creates a digital filter
    with numerator and denominator coefficients BZ and AZ
    respectively whose impulse response is equal to the
    impulse response of the analog filter with
    coefficients B and A sampled at a frequency of Fs
    Hertz.  The B and A coefficients will be scaled by
    1/Fs.

    If you don't specify Fs, it defaults to 1 Hz.

    [BZ,AZ] = impinvar(B,A,Fs,TOL) uses the tolerance TOL
    for grouping repeated poles together.  Default value is
    0.001, i.e., 0.1%.

    NOTE: the repeated pole case works, but is limited by
    the ability of the function ROOTS to factor such
    polynomials.

    % EXAMPLE: Illustrate the relationship between digital
    % and analog frequency responses.
    [b,a] = butter(4,0.3,'s');
    [bz,az] = impinvar(b,a,10);
    [Ha,Wa] = freqs(b,a,512);
    [Hz,Wz] = freqz(bz,az,512,10);
    plot(Wa/(2*pi),20*log10(abs(Ha)),'LineWidth',2);
    hold on;
    plot(Wz,20*log10(abs(Hz)),'r--');
    xlabel('Frequency (Hz)'), ylabel('Magnitude (dB)');
    title('Magnitude Response Comparison');
    legend('Analog Filter','Digital Filter');

    See also bilinear.
```

A1.6 MATLAB Function lsim.m

```
lsim  Simulate time response of dynamic systems to
arbitrary inputs.

    lsim(SYS,U,T) plots the time response of the dynamic
    system SYS to the input signal described by U and T.
    The time vector T is expressed in the time units of SYS
    and consists of regularly spaced time samples. The
    matrix U has as many columns as inputs in SYS and its
    i-th row specifies the input value at time T(i). For
    example,
            t = 0:0.01:5;   u = sin(t);   lsim(sys,u,t)
    simulates the response of a single-input model SYS to
    the input u(t)=sin(t) during 5 time units.

    For discrete-time models, U should be sampled at the
    same rate as SYS (T is then redundant and can be
    omitted or set to the empty matrix). For continuous-
    time models, choose the sampling period T(2)-T(1) small
    enough to accurately describe the input U.  lsim issues
    a warning when U is undersampled and hidden
    oscillations may occur.

    Y = lsim(SYS,U,T) returns the output history Y. No plot
    is drawn on the screen. The matrix Y has LENGTH(T) rows
    and as many columns as outputs in SYS. For state-space
    models,
       [Y,T,X] = lsim(SYS,U,T,X0)
    also returns the state trajectory X, a matrix with
    LENGTH(T) rows and as many columns as states.

    See also lsimplot, gensig, step, impulse, initial,
    DynamicSystem.
```

A1.7 MATLAB Function residue.m

```
residue Partial-fraction expansion (residues).
    [R,P,K] = residue(B,A) finds the residues, poles and
    direct term of a partial fraction expansion of the
    ratio of two polynomials B(s)/A(s). If there are no
    multiple roots,
       B(s)       R(1)       R(2)             R(n)
       ----  =  -------- + -------- + ... + -------- + K(s)
       A(s)     s - P(1)   s - P(2)         s - P(n)
    Vectors B and A specify the coefficients of the
    numerator and denominator polynomials in descending
    powers of s.  The residues are returned in the column
    vector R, the pole locations in column vector P, and
    the direct terms in row vector K.  The number of poles
    is n = length(A)-1 = length(R) = length(P). The direct
    term coefficient vector is empty if
    length(B) < length(A), otherwise
    length(K) = length(B)-length(A)+1.

    If P(j) = ... = P(j+m-1) is a pole of multplicity m,
    then the expansion includes terms of the form
            R(j)        R(j+1)                R(j+m-1)
          -------- + ------------   + ... + ------------
          s - P(j)   (s - P(j))^2           (s - P(j))^m

    [B,A] = residue(R,P,K), with 3 input arguments and 2
    output arguments, converts the partial fraction
    expansion back to the polynomials with coefficients in
    B and A.

    Warning: Numerically, the partial fraction expansion of
    a ratio of polynomials represents an ill-posed problem.
    If the denominator polynomial, A(s), is near a
    polynomial with multiple roots, then small changes in
    the data, including roundoff errors, can make
    arbitrarily large changes in the resulting poles and
    residues. Problem formulations making use of state-
    space or zero-pole representations are preferable.

    Class support for inputs B,A,R:
```

```
   float: double, single

See also poly, roots, deconv.
```

A1.8 MATLAB Function etfe.m

```
etfe   Computes the Empirical Transfer Function Estimate
and Periodogram.
    G = etfe(DATA)   or   G = etfe(DATA,M)

    DATA is an IDDATA object and contains the input-output
    data or a time series. See HELP IDDATA. If an input is
    present G is returned as the etfe (the ratio of the
    output Fourier transform to the input Fourier
    transform) for the data. For a time series  G is
    returned as the periodogram (the normed absolute square
    of the Fourier transform) of the data. G is returned as
    an IDFRD object. See HELP IDFRD.

    With M specified, a smoothing operation is performed on
    the raw spectral estimates using a Hamming Window,
    giving a frequency resolution of about pi/M. Default, M
    = [], gives no smoothing.

    For non-periodic data, the transfer function is
    estimated at 128 equally spaced frequencies between 0
    (excluded) and pi. This number can be changed to N  by
    G = etfe(DATA,M,N).

    PERIODIC DATA: If the (input) data is marked as
    periodic (DATA.Period = integer) and contains an
    integer number of periods, the frequency response is
    computed at the frequencies k*2*pi/period for k=0 up to
    the Nyquist frequency. To compute the spectrum of a
    periodic signal S, it must be an input signal:
    DATA = iddata([],S,'Ts',Ts,'Period',per). For periodic
    data, the arguments N and M are ignored.

    FREQUENCY DOMAIN DATA: If the data set is defined in
    the frequency domain, G is returned as an IDFRD object,
    defined from the ratio of output to input at all
    frequencies, where the input is non-zero. If M is
    defined, the corresponding smoothing is applied.
```

Appendix 2: Author Contact Information

For any questions regarding this Ph.D. dissertation, the author may be contacted at the address below.

Name: Paul E. Dahlen
Senior Technical Staff Member

Address: IBM Corporation
3605 Highway 52 N
MS: 3HHA/040-3
Rochester, MN 55901

Office phone: (507) 253-0690

e-mail: pdahlen@us.ibm.com
dahl0089@umn.edu