

Copyright © 1989, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**THEORETICAL ASPECT OF RELAXATION-BASED
AND NONLINEAR FREQUENCY DOMAIN CIRCUIT
SIMULATION**

by

Tammy Tzu-Chen Huang

Memorandum No. UCB/ERL M89/51

28 April 1989

**THEORETICAL ASPECT OF RELAXATION-BASED
AND NONLINEAR FREQUENCY DOMAIN CIRCUIT
SIMULATION**

by

Tammy Tzu-Chen Huang

Memorandum No. UCB/ERL M89/51

28 April 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

**THEORETICAL ASPECT OF RELAXATION-BASED
AND NONLINEAR FREQUENCY DOMAIN CIRCUIT
SIMULATION**

by

Tammy Tzu-Chen Huang

Memorandum No. UCB/ERL M89/51

28 April 1989

ELECTRONICS RESEARCH LABORATORY

**College of Engineering
University of California, Berkeley
94720**

Theoretical Aspect of Relaxation-Based and Nonlinear Frequency Domain Circuit Simulation

Tammy Tzu-Chen Huang

Ph. D.

Department of Electrical Engineering
and Computer Science

Abstract

Circuit simulation is the key to shorten the design and development time of complex integrated circuits. Several methods have been recently proposed to speed up this task.


For digital circuits, relaxation techniques have been able to achieve up to two orders of magnitude speed-up. For analog circuits where distributed components and steady-state behavior are of importance, harmonic balance methods have been successful. This thesis addresses fundamental theoretical questions on these algorithms: convergence, accuracy, and stability.

In the first part, the numerical properties of Implicit-Implicit-Explicit (IIE) method, a relaxation method, are investigated. The IIE method has been proven to be consistent, stable and convergent. Experimental results showed that IIE gives fairly accurate solutions; however, the time-step required for the iteration is smaller than desired.

In the second part of the thesis, numerical properties of the harmonic-Newton method, a harmonic balance method for microwave circuits, are analyzed. The harmonic-Newton method is shown to converge to the solution of the circuit equations when applied to periodic nonautonomous systems, almost-periodic nonautonomous systems, and periodic autonomous systems.

A complete error analysis on the harmonic-Newton method is also presented. The theoretical results are used to develop an algorithm to estimate the number of harmonics needed for given

error objectives prior to the application of the harmonic-Newton iteration. Simulation results show that this method can effectively improve the overall efficiency of the harmonic-Newton method.



James G. Bell

Chairman of Committee

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Professor Sangiovanni-Vincentelli for his thoughtful guidance and continuous support during the course of my Ph. D. studies. It has been my privilege to be able to participate in this project under his guidance. I am also very grateful to several professors who have contributed their time on my behalf, especially Prof. Brayton, Prof. Pederson, and Prof. Kobayashi who served on my qualifying and thesis committee.

The help of members and ex-members of the CAD group was very important in making my research effort successful. Special thanks to Ken Kundert, Tom Quarles, Resve Saleh, Carl Sechen, Jyuo-Min Shyu, Ricks Spickelmier, Don Webber, Nicholas Weiner, Jacob White, and Hormoz Yaghutiel for their help and useful discussion.

I would like to express my hearty gratitude to my parents, brothers, and sisters for their endless encouragement. My most sincere thanks goes to my husband, Nan-Sheng, who has endured many of my frustrations. His patience and sacrifice I will not soon forget.

This research was supported by grants from JSEP, DARPA, MICRO, Harris, Hewlett-Packard, Hughes, Rockwell, and Xerox.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION	1
PART I RELAXATION-BASED CIRCUIT SIMULATION	
CHAPTER 2. INTRODUCTION	3
CHAPTER 3. OVERVIEW OF CIRCUIT SIMULATION TECHNIQUES	5
3.1. The Circuit Simulation Problem	5
3.2. Numerical Integration in Standard Simulators	6
3.3. Relaxation Technique	9
3.3.1. Linear Relaxation Methods	11
3.3.2. Nonlinear Relaxation Methods	13
3.3.3. Waveform Relaxation Methods	15
3.4. Timing Simulation	17
3.4.1. Timing Analysis Algorithms	17
3.4.2. Numerical Properties	19
3.4.3. Problems with Timing Analysis Algorithms	24
CHAPTER 4. THE IMPLICIT-IMPLICIT-EXPLICIT METHOD	26
4.1. Mathematical Formulation	26
4.2. Numerical Properties of IIE Method	30
4.3. Implementation of IIE as Initial Condition Generator	40
4.3.1. Implementation	40
4.3.2. Implementational Results	43
4.4. Conclusions	54
PART II CIRCUIT SIMULATION IN THE FREQUENCY-DOMAIN	
CHAPTER 5. INTRODUCTION	56
CHAPTER 6. OVERVIEW	59
6.1. Background	59
6.2. Problem Formulation	62
6.3. The Newton Method in the Time Domain	64
6.3.1. Periodic Nonautonomous Systems	64
6.3.2. Periodic Autonomous Systems	66
6.4. The Harmonic Balance Method	68
CHAPTER 7. THE HARMONIC-NEWTON METHOD	72
7.1. Periodic Nonautonomous Systems	72
7.2. Almost-Periodic Nonautonomous Systems	76
7.3. Periodic Autonomous Systems	77
CHAPTER 8. CONVERGENCE OF HARMONIC-NEWTON METHOD	81
8.1. Background	81
8.2. Periodic Nonautonomous Systems	84
8.3. Almost-Periodic Nonautonomous Systems	95
8.4. Periodic Autonomous Systems	99
CHAPTER 9. ERROR ESTIMATION AND CHOOSING NUMBER OF HARMONIC	105

9.1. Theoretical Bounds	105
9.2. Practical Estimates	113
9.3. Simulation Results	116
9.4. Conclusions	140

CHAPTER 10. CONCLUSIONS	142
-------------------------------	-----

REFERENCES

CHAPTER 1

Introduction

Circuit simulators are one of the most important computer-aided design tools for the design and analysis of large scale integrated circuits. A circuit simulator is normally designed or optimized for performing a certain class of circuit analysis such as dc analysis, ac analysis, transient analysis, steady-state analysis, noise, and distortion analysis. In this thesis, we devote ourselves to the time-domain transient analysis and the steady-state analysis, with special emphasis on the theoretical aspects of the algorithms used in these analysis.

In PART I of this thesis, we will concentrate on the time-domain transient analysis, one of the most complicated and expensive type of analysis.

Despite of their accuracy, conventional circuit simulators such as SPICE [1] and ASTAP [2] can only be used on relatively small circuits because the run time goes up rapidly with the circuit size. Because of this limitation, Relaxation-based circuit simulators were proposed. They provide accurate solutions as in conventional simulators with up to two order of magnitude speed improvement for larger digital MOS circuits. In this thesis, various relaxation-based techniques will be discussed. Among them, one special relaxation methods called timing analysis algorithm has a characteristic of performing only one relaxation iteration per time-step while one or more Newton-Raphson iterations may be used to solve each nodal equation. One particular timing analysis algorithm called Implicit-Implicit-Explicit method [3] has nice numerical properties for two-node circuits even when tightly-coupled feedback loops are present. In PART I of this dissertation, we formalize the IIE method and present its numerical properties. Both the theoretical and computational aspects of the IIE method will be discussed in detail. Implementation of the method and its experimental results are also discussed in depth.

In the analysis of many communication circuits and oscillators, one is frequently interested only in the steady-state periodic responses. If conventional numerical techniques are employed, one has no alternative but to integrate the differential equations over a sufficiently long interval of time until the transient waveform dies out. This procedure is acceptable only if the transient response decays rapidly. For many lightly damped circuits, the transient response will last for a long time, resulting in expensive and costly simulations. Frequency-domain simulation techniques on the other hand, assume the absence of transient response and search for steady-state response directly. This significantly reduces the overall simulation CPU time. In PART II of this thesis, we will examine one particular method called the harmonic-Newton method[4] which finds the periodic steady-state solutions in the frequency domain.

When we transform a nonlinear system from time-domain to frequency domain, it becomes an infinite-dimensional problem. In order to make the frequency-domain processing practical, we have to truncate the number of frequency elements, or harmonics, considered. Since some kind of truncation is inevitable, it is very important to study the convergence properties of the numerical method used in order to ensure that the solution obtained is meaningful. The convergence properties and the computation of the error generated by the harmonic-Newton method will be the focus of PART II of this thesis. In addition, we will describe how this error bound can be used to predict the number of harmonics needed for a given error objective before applying the harmonic-Newton method.

Organization of PART I and PART II are similar. First, we formulate the problems, review the basic principle or algorithms used in existing simulation programs, and state the numerical properties and simulation problems associated with these algorithms. Then, we will present the IIE method and the harmonic-Newton method in PART I and PART II respectively, with their numerical properties examined in detail. Finally, some experiment results for both methods will be presented.

PART I : RELAXATION-BASED SIMULATION

CHAPTER 2

Introduction

When performing transient analysis, traditional simulators such as SPICE[1] and ASTAP[2] use a direct method which consists of an implicit integration method, the Newton-Raphson method, and a sparse matrix technique to solve the circuit equations in the time domain. These simulators give accurate time-domain current and voltage waveforms based on device level descriptions of integrated circuits. However, they can only be used on relatively small circuits because the computational run time increase rapidly with the circuit size.

In the past few years, several techniques have been used to extend the capability of circuit simulators to larger circuits. One of the techniques is the timing analysis which was first introduced in MOTIS[5] for simulations of MOS digital circuits. It applies a particular relaxation method on the nonlinear algebraic equations derived from the nonlinear differential equations after the implicit integration method is used. The particular characteristic of these relaxation methods is that they do not carry the iteration to convergence. In fact, each node equation is solved only once at each time point. For circuits with no floating capacitors, i.e. capacitors connecting two nodes, none of which is either ground or a voltage rail, these methods have been shown to be consistent, stable and as accurate as the backward Euler method. Although these methods offer a substantial saving in CPU-time and memory usage, when floating capacitors are present in the circuit, all these methods have serious accuracy problems.

In light of the problem, the Implicit-Implicit-Explicit (IIE) method [3] has nice numerical properties for two-node circuits containing a floating capacitor as shown by experimental results. However, it was not clear how this method behaves when it is applied to a general circuit in which more than two nodes are connected by floating capacitors. The purpose of PART I of this disserta-

tion is to formalize the IIE method and present its numerical properties.

The organization of PART I is as follows. In Chapter 3, we start with an introduction to circuit simulation problems, showing how the differential equations are formulated from the circuit topology. Then, a brief review of some standard simulation approaches and a comprehensive discussion and classification of various relaxation techniques will be given. Finally, several commonly used timing analysis algorithms and their numerical properties such as consistency, stability, and convergence are described.

A detailed discussion of the IIE method is given in Chapter 4. We investigate rigorously its numerical properties on a test problem to provide some insight on how it behaves in general. The experimental results using the IIE as the initial waveform generator in RELAX[6] will also be presented.

In PART I of this thesis, the following conventions are used; Bold face letters represent vectors of variables or functions, and italics are used for scalar variables or functions. For variables or functions, superscript with parentheses is used to denote Newton iteration count while superscript without parentheses is used for node numbers. Subscript with parentheses represents relaxation iteration count, and subscript without parentheses is used as time index. Matrices are written in boldface capital letters, and elements of matrices are given by lower-case letters with subscripts denoting the matrix indices.

CHAPTER 3

Overview of Circuit Simulation Techniques

In this chapter, we will give an overview of various circuit simulation techniques and discuss how circuit equations for transient analysis are formulated. Special attention will be given to timing analysis algorithms, their numerical properties, and inherent problems associated with various techniques.

3.1. The Circuit Simulation Problem

The most general form of equations describing the circuit behavior is

$$F(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{u}(t)) = 0 \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.1.1)$$

where $\mathbf{x}(t)$ is the vector of the circuit variables, $\dot{\mathbf{x}}(t)$ is the time derivative of $\mathbf{x}(t)$, $\mathbf{u}(t)$ is the vector of the independent source, t is time, and F is a function which maps $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$ and t into a vector of real numbers.

The first question to ask is whether a solution of Eqn.(3.1.1) exists and is unique. It turns out that, under rather mild conditions on the continuity and differentiability of F , it can be proven that there exists a unique solution. However, these conditions can be difficult to verify in practice. If the circuit is well designed, it is obvious that the circuit should have unique solution for a given initial condition. In this thesis, we will assume that Eqn.(3.1.1) has a unique solution.

Since it is impossible to find a closed form solution to Eqn.(3.1.1) in general, one must resort to some numerical methods to obtain the solution. The numerical methods available are all iterative. They produce a sequence of approximate solutions which hopefully converge to exact the solution $\mathbf{x}^*(t)$.

3.2. Numerical Integration in Standard Simulators

When doing transient analysis, the values of some branch voltages and branch currents for some time interval, say for $0 \leq t \leq T$, are computed given the initial conditions of the capacitor voltages and inductor currents. This is done by solving systems of nonlinear first-order ordinary differential equations of the form given in Eqn. (3.1.1). In standard simulators, the numerical process of finding the solution is broken into three steps and the following three conventional numerical methods are used at each step of simulation:

- (1) Given a differential equation describing the circuit, an implicit integration method is used to approximate the time derivative operator to yield a discrete-time system of nonlinear algebraic equations.
- (2) The Newton-Raphson (NR) method is applied on the system of nonlinear algebraic equations generated by step 1. This results in a system of linear algebraic equations.
- (3) Apply direct sparse-matrix techniques to obtain the solution to the system of linear algebraic equations generated by step 2.

The implicit integration method used in Step 1 subdivides the interval $[0, T]$ into a finite set of distinct points:

$$t_0 = 0, \quad t_K = T, \quad t_{k+1} = t_k + h_{k+1}, \quad k = 0, 1, 2, \dots, K - 1.$$

The quantities h_{k+1} are called *time steps* and their values are called *step size*. By applying an implicit integration method, Eqn. (3.1.1) is transformed into a discrete time sequence of algebraic equations with $\dot{\mathbf{x}}(t_{k+1})$ being replaced by a combination of \mathbf{x} at t_{k+1} , t_k , and possibly preceding points. In doing so, we form a set of algebraic difference equations which approximate Eqn. (3.1.1). Then, we can solve for $\mathbf{x}(t_{k+1})$ for $k = 0, 1, 2, \dots, K-1$. The fact that we can choose an implicit integration method in a variety of different ways gives rise to a number of integration methods with different numerical properties. Among all the implicit integration methods available, we will consider the backward Euler method with constant step size as an example since it has been widely used and has nice numerical properties.

In the backward Euler method, $\dot{\mathbf{x}}_{k+1}$ is expressed as a function of \mathbf{x}_{k+1} and \mathbf{x}_k :

$$\text{Backward Euler: } \mathbf{x}_{k+1} = \mathbf{x}_k + h \dot{\mathbf{x}}_{k+1} \quad (3.2.1)$$

Since the value of \mathbf{x}_k is known at time t_{k+1} , $\dot{\mathbf{x}}_{k+1}$ can be written as a function of \mathbf{x}_{k+1} , or:

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_{k+1}(\mathbf{x}_{k+1}) \quad \text{for } k = 0, 1, 2, \dots, K-1. \quad (3.2.2)$$

Substituting Eqn. (3.2.2) into Eqn.(3.1.1), we have:

$$\mathbf{F}(\dot{\mathbf{x}}_{k+1}(\mathbf{x}_{k+1}), \mathbf{x}_{k+1}, \mathbf{u}(t_{k+1})) = 0 \quad \text{for } k = 0, 1, 2, \dots, K-1. \quad (3.2.3)$$

Now, Eqn.(3.2.3) is a system of discretized nonlinear equations with \mathbf{x}_{k+1} being the unknown variable.

If the differential equations are nonlinear, the discretized algebraic equations are also nonlinear. These nonlinear equations can be converted into linear equations using the NR method (step 2 of the conventional simulators). Let j be the iteration index of the NR method, then Eqn. (3.2.3) can have the following iteration form:

$$\mathbf{x}_{k+1}^{(j+1)} - \mathbf{x}_{k+1}^{(j)} = -\mathbf{J}^{-1}(\dot{\mathbf{x}}_{k+1}(\mathbf{x}_{k+1}^{(j)}), \mathbf{x}_{k+1}^{(j)}, \mathbf{u}(t_{k+1}))\mathbf{F}(\dot{\mathbf{x}}_{k+1}(\mathbf{x}_{k+1}^{(j)}), \mathbf{x}_{k+1}^{(j)}, \mathbf{u}(t_{k+1})) \quad (3.2.4)$$

$$\text{for } j = 0, 1, 2, \dots \text{ and } k = 0, 1, 2, \dots, K-1.$$

where $\mathbf{J}(\dot{\mathbf{x}}_{k+1}(\mathbf{x}_{k+1}), \mathbf{x}_{k+1}, \mathbf{u}(t_{k+1})) = \frac{\partial \mathbf{F}}{\partial \mathbf{x}}(\dot{\mathbf{x}}_{k+1}(\mathbf{x}_{k+1}), \mathbf{x}_{k+1}, \mathbf{u}(t_{k+1}))$ is the Jacobian matrix. Notice that the computation of each Newton iteration involves the evaluation of the function and its Jacobian matrix.

The linear equations Eqn.(3.2.4) generated by the NR method can be solved using direct sparse matrix techniques such as LU decomposition or Gaussian Elimination (step 3 of conventional simulators). The hierarchical organization of standard transient analysis simulators is shown in Fig. 3.1.

Although these conventional methods have been proven to be extremely reliable, with the increase in circuit size, these standard simulators demand more computation time and have higher storage requirements than desired for the analysis of VLSI circuits. For circuits with less than a thousand devices, because of the efficiency of sparse matrix techniques in solving linear equations, the computation of the Jacobian matrix and the evaluation of the function dominate the complexity

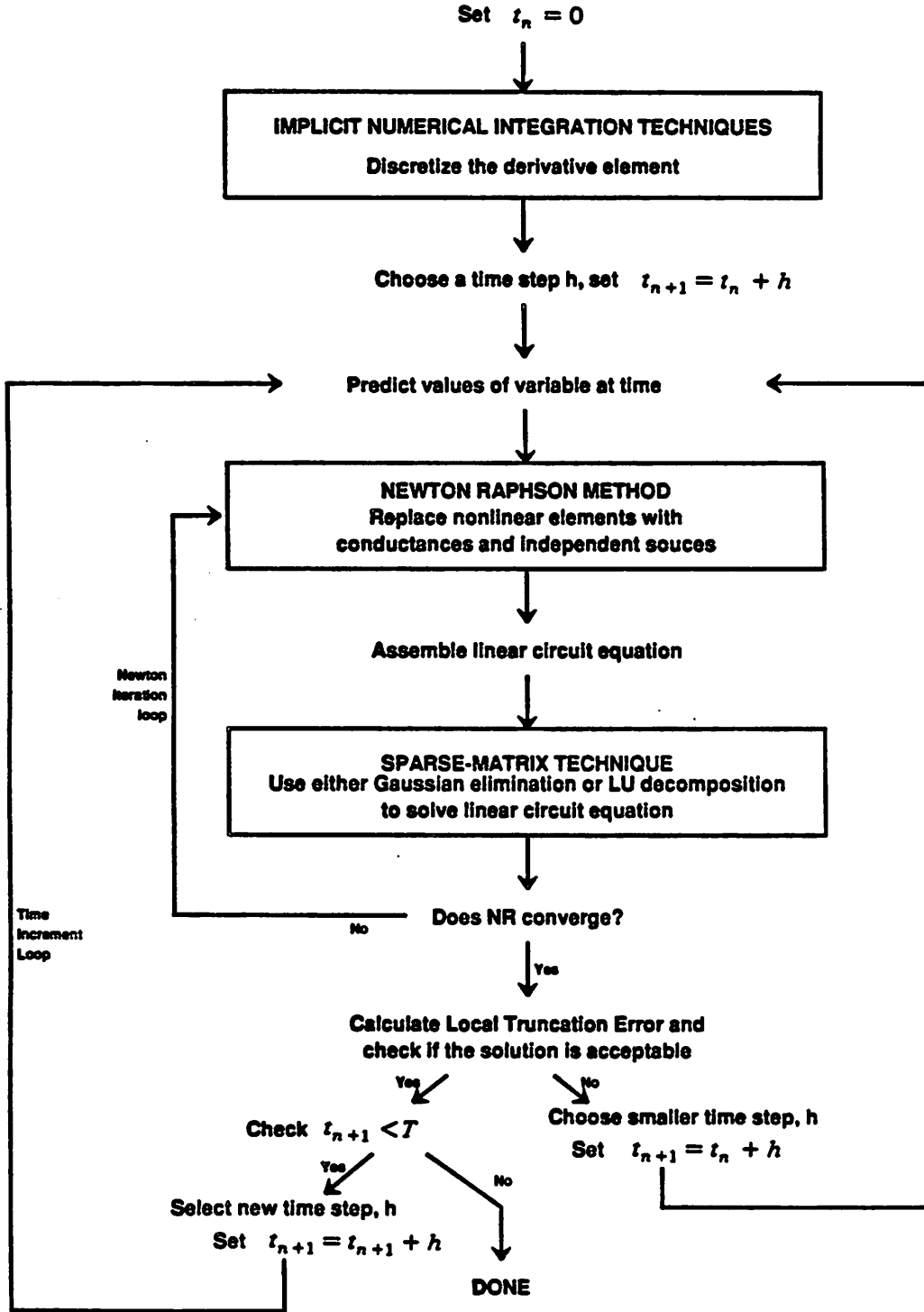


Figure 3.1 Structure of a Transient Analysis Simulator

of the simulation. The computational cost of performing the matrix solution for circuits with more than 1000 nodes grows rapidly with the circuit size despite of the use of sparse matrix techniques[7].

3.3. Relaxation Techniques

Recently, a new class of algorithms have been applied to the electrical integrated circuit simulation problems. These algorithms use relaxation methods at one of the steps in the numerical process to solve the circuit equations. Relaxation-based simulators such as RELAX[6] and SPLICE [8] provide the same level of accuracy as the standard simulators and may significantly reduce the overall simulation run time. Both RELAX and SPLICE have been proven to be very effective for the analysis of digital MOS integrated circuits.

Relaxation methods can be applied at any stage in the solution of Eqn. (3.1.1), as illustrated in Fig. 3.2 [9, 10]. The advantages of relaxation methods are that they avoid solving a large system of equations directly and they permit the simulator to exploit latency efficiently.

Several assumptions are required by the relaxation-based simulators:

- (1) Each MOS device and its interconnections can be modeled by lumped (linear or nonlinear) voltage-controlled capacitors, conductors, and current sources, (i.e. branch equations are such that nodal analysis can be used.)
- (2) Every (internal or external) node in the circuit has a (linear or nonlinear) capacitor connected either to ground or dc supply voltage rails.

For LSI MOS circuits, these assumptions are usually satisfied. Under these assumptions, the nodal equation can be written in the following form:

$$C(\mathbf{v}(t)) \dot{\mathbf{v}}(t) + \mathbf{f}(\mathbf{v}(t), \mathbf{u}(t)) = 0 \quad \mathbf{v}(0) = \mathbf{v}_0 \quad (3.3.1)$$

where \mathbf{v} is the vector of all unknown node voltages, \mathbf{v}_0 is the vector of the initial values of \mathbf{v} , \mathbf{u} is the vector of the independent source waveforms, \mathbf{f} is a continuous function whose components represent the net sum of currents flowing out of the capacitors connected to the node, and C

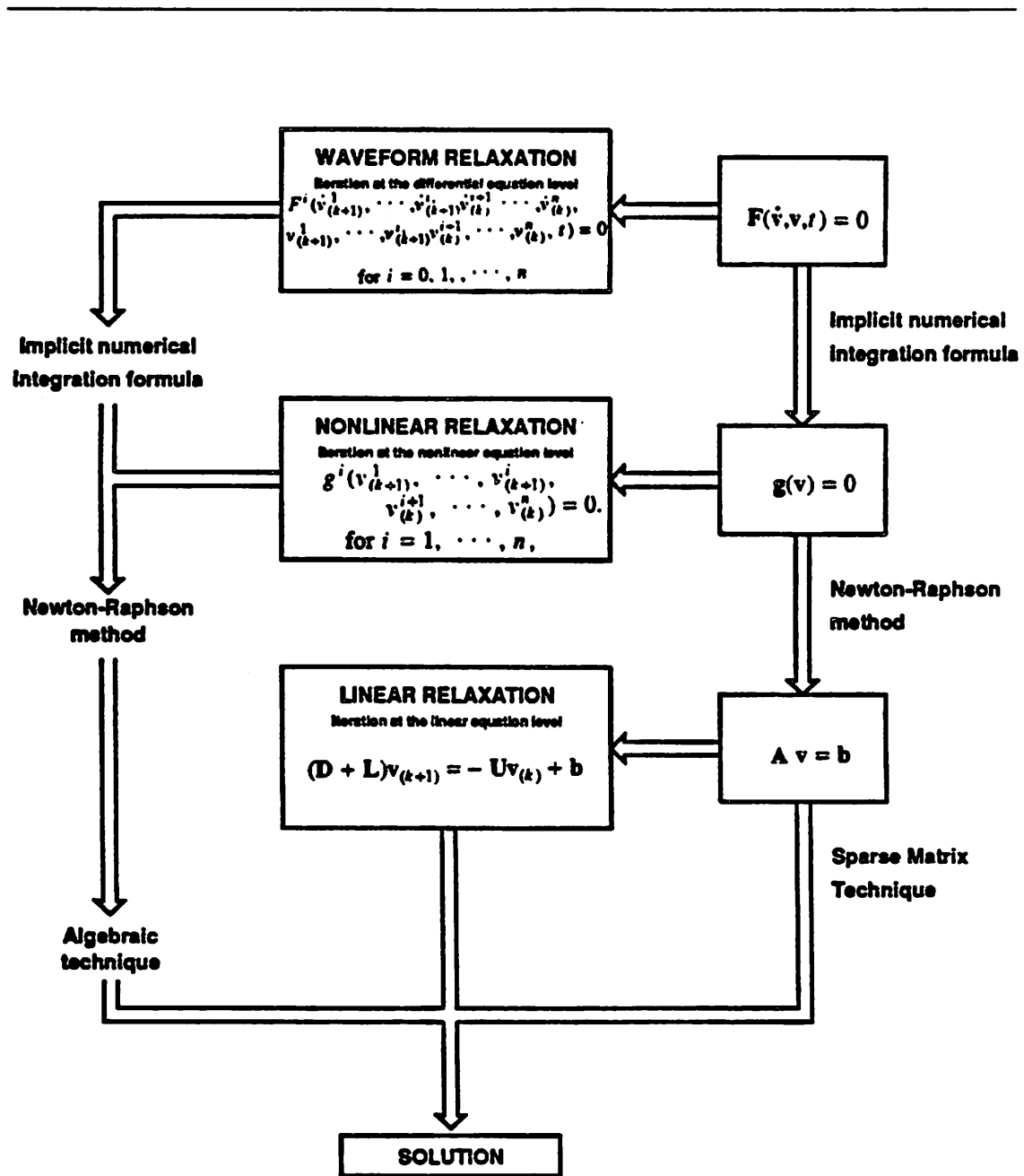


Figure 3.2 The use of relaxation technique at various levels of system of equations using Gauss-Seidel iteration as an example.

represents the nodal capacitance matrix. Under these assumptions, the capacitor matrix has nonzero diagonal entries and is often diagonally dominant.

If we assume that (1) the inverse of $C(v)$ exists and is uniformly bounded with respect to v , u ; (2) $u(t)$ is piecewise continuous; and (3) f is globally Lipschitz continuous with respect to v for all u , then Eqn. (3.3.1) has a unique solution on any finite interval $[0, T]$ [11]. Again, these conditions are difficult to verify in practice; we will also assume that Eqn. (3.3.1) has a unique solution in this thesis.

The two most commonly used relaxation methods are the Gauss-Jacobi (GJ) method and the Gauss-Seidel (GS) Method[12, 13]. We will use these two methods to illustrate how relaxation techniques can be applied at different stages of the numerical process.

3.3.1. Linear Relaxation Methods

Consider a system of n linear equations of the form:

$$\mathbf{A} \mathbf{v} = \mathbf{b} \quad (3.3.2)$$

where $\mathbf{v} = [v^1, \dots, v^n]^T$, $\mathbf{b} = [b^1, \dots, b^n]^T$, $v^i, b^i \in R$, and $\mathbf{A} = (a_{ij})$, $\mathbf{A} \in R^{n \times n}$. Note that the system of equations generated by Step II of standard simulators are written in this form. The solution vector \mathbf{v} exists and is unique if and only if \mathbf{A} is nonsingular and this solution vector is given explicitly by

$$\mathbf{v} = \mathbf{A}^{-1}\mathbf{b}. \quad (3.3.3)$$

We assume throughout this section that the matrix \mathbf{A} is nonsingular. Instead of the direct sparse-matrix method, one can use a linear relaxation method such as the GJ or GS method to solve Eqn. (3.3.2). To be able to use the GJ or GS method, we also need to also assume that the diagonal entries of \mathbf{A} are all nonzero numbers.

Let \mathbf{A} be split into $\mathbf{L} + \mathbf{D} + \mathbf{U}$, where \mathbf{L} is strictly lower triangular, \mathbf{D} is diagonal, and \mathbf{U} is strictly upper triangular, then Eqn. (3.3.2) will have the following form when the Gauss-Jacobi method is applied:

$$D\mathbf{v}_{(k+1)} = -(\mathbf{L} + \mathbf{U})\mathbf{v}_{(k)} + \mathbf{b}, \quad k \geq 0. \quad (3.3.4)$$

where k is the iteration index. The advantage of this method is that all n equations can be solved simultaneously using parallel processing since the order in which the equations are solved is irrelevant.

If the Gauss-Siedel iteration is used, then Eqn. (3.3.2) becomes:

$$(\mathbf{D} + \mathbf{L})\mathbf{v}_{(k+1)} = -\mathbf{U}\mathbf{v}_{(k)} + \mathbf{b} \quad k \geq 0. \quad (3.3.5)$$

It uses the latest estimate $v_{(k+1)}^i$ of the component v^i of the solution vector \mathbf{v} in all subsequent computations. When the linear equation (3.3.2) is obtained by the NR method, this combined method is a special case of the Newton successive overrelaxation (Newton-SOR) method. The order in which these n equations are placed in the matrix form is critical to the speed of convergence. It is obvious that if \mathbf{A} is a lower triangular matrix, it will only take one GS iteration to reach the solution. Hence, it is desirable to arrange the matrix \mathbf{A} as close to lower triangular as possible or to have the matrix \mathbf{U} as sparse as possible.

Since relaxation methods are iterative, it is important to ask under what conditions they are guaranteed to converge to the solution. In the case where \mathbf{A} is diagonally dominant, it has been proven that both the GJ and GS methods converge to the solution. This property is described in the following theorem [12].

Theorem 3.1. Let \mathbf{A} be a strictly or irreducibly diagonally dominant $n \times n$ complex matrix, then for the problem $\mathbf{A}\mathbf{v} = \mathbf{b}$, both the Gauss-Jacobi and the Gauss-Seidel iterations converge to the solution for any given initial guess vector.

The rate of convergence for both methods is linear. That is, after a sufficiently large number of iterations, the error at each iteration decreases according to

$$\|\mathbf{v}_{(k+1)} - \mathbf{v}_{(k)}\| = \epsilon_{(k+1)} \quad (3.3.6)$$

$$\epsilon_{(k+1)} \leq C \epsilon_{(k)}$$

where C is a positive constant smaller than one. Since it is guaranteed that the exact solution can be obtained in one iteration when the sparse-matrix techniques such as Gaussian elimination or LU

decomposition are used, relaxation methods described here are advantageous only when the number of iterations needed to reach the convergence is small. Moreover, the convergence property for the GJ and GS methods is guaranteed only under the condition that A is strictly diagonally dominant. For this reason, sparse-matrix techniques have been used more widely than relaxation techniques at linear system level in conventional circuit simulators.

3.3.2. Nonlinear Relaxation Methods

Relaxation methods can also be used at the nonlinear equation level. Examples of simulators using nonlinear relaxation methods are MOTIS[5] and SPLICE. Consider the following system of n nonlinear equations:

$$g(\mathbf{v}) = 0 \quad (3.3.7)$$

where $g:R^n \rightarrow R^n$ has components $g^1, g^2, \dots, g^i, \dots, g^n$, and $\mathbf{v}:R^n$ has components $v^1, v^2, \dots, v^i, \dots, v^n$, with i being the node index. Recall that the Gauss-Jacobi iteration in the linear algebraic equation level is simply to obtain the solution of the i^{th} equation $v_{(k+1)}^i$ with the other $n-1$ variables held at iteration k , with k being the iteration index. We can extend the same prescription to the nonlinear algebraic equation level. Then, the basic step of the *nonlinear Gauss-Jacobi* iteration is to solve the following equation:

$$g^i(v_{(k)}^1, \dots, v_{(k)}^{i-1}, v_{(k+1)}^i, v_{(k)}^{i+1}, \dots, v_{(k)}^n) = 0. \quad (3.3.8)$$

$$\text{for } i = 1, \dots, n, \quad k \geq 0.$$

Thus, we can solve all $v_{(k+1)}^i$ for $i = 1, \dots, n$ from $\mathbf{v}_{(k)}$ simultaneously.

In a completely analogous fashion, the Gauss-Seidel iteration can also be extended to the *nonlinear Gauss-Seidel* iteration. Written in a component form, the i^{th} equation becomes

$$g^i(v_{(k+1)}^1, \dots, v_{(k+1)}^{i-1}, v_{(k+1)}^i, v_{(k)}^{i+1}, \dots, v_{(k)}^n) = 0. \quad (3.3.9)$$

$$\text{for } i = 1, \dots, n, \quad k \geq 0.$$

If the one-dimensional Newton method is used to solve Eqn. (3.3.8) and Eqn. (3.3.9), then we will have a double loop iteration where the inner loop is the Newton iteration and the outer loop is

the GJ or GS nonlinear iteration. In the case when the Gauss-Jacobi-Newton method is used, the iteration becomes

$$v_{(k+1)}^{i(j+1)} - v_{(k+1)}^{i(j)} = - \left[\frac{\partial g^i}{\partial v^i}(v_{(k)}^1, \dots, v_{(k)}^{i-1}, v_{(k+1)}^{i(j)}, v_{(k)}^{i+1}, \dots, v_{(k)}^n) \right]^{-1} g^i(v_{(k)}^1, \dots, v_{(k)}^{i-1}, v_{(k+1)}^{i(j)}, v_{(k)}^{i+1}, \dots, v_{(k)}^n),$$

for $j = 0, 1, \dots, k = 0, 1, \dots, i = 1, \dots, n.$ (3.3.10)

Similarly, when the Gauss-Seidel-Newton method is used, we have:

$$v_{(k+1)}^{i(j+1)} - v_{(k+1)}^{i(j)} = - \left[\frac{\partial g^i}{\partial v^i}(v_{(k+1)}^1, \dots, v_{(k+1)}^{i(j)}, v_{(k)}^{i+1}, \dots, v_{(k)}^n) \right]^{-1} g^i(v_{(k+1)}^1, \dots, v_{(k+1)}^{i(j)}, v_{(k)}^{i+1}, \dots, v_{(k)}^n),$$

for $j = 0, 1, \dots, k = 0, 1, \dots, i = 1, \dots, n.$ (3.3.11)

We may, of course, replace the NR method with other one-dimensional iteration method as the inner loop to obtain the solutions of Eqn.(3.3.8) and Eqn.(3.3.9).

It is desirable to find the required conditions on g to ensure the convergence of the Gauss-Jacobi-Newton and Gauss-Seidel-Newton method. This is described in the following theorem.

Theorem 3.2. [13]

Let $g(v)$ be continuously differentiable in an open neighborhood S^0 of a point v^* for which $g(v^*) = 0$. Assume that $\frac{\partial g}{\partial v}(v^*)$ is symmetric and positive definite. Consider again the decomposition of $\frac{\partial g}{\partial v}(v)$ into diagonal, strictly lower-triangular, and strictly upper-triangular parts and suppose that the diagonal matrix is nonsingular. Under the above assumptions, the Gauss-Jacobi-Newton and Gauss-Seidel-Newton iterations described in Eqn. (3.3.7) and Eqn. (3.3.8) are well defined on an open ball $B(v^*, \delta)$ belonging to S^0 , and the iterations converge to v^* .

It is clear that the GJ or the GS nonlinear iteration is the primary iteration and the NR iteration is the secondary iteration. We can argue intuitively that it is not necessary to be too critical of the accuracy of the NR method at least for the first few GJ or GS iterations. In fact, it has been

shown in [13], that the asymptotic rate of convergence for the case where one NR iteration is taken for each GJ or GS loop is the same as the case where an infinite number of NR iterations are taken for each GJ or GS loop. Therefore it does not improve the speed of convergence to take more than one NR iterations.

3.3.3. Waveform Relaxation Methods

When relaxation techniques are applied at the differential equation level, we call this class of algorithms *Waveform Relaxation (WR)* method [10]. While relaxation techniques use real vectors (belonging to R^n) as the variables in the linear and nonlinear algebraic equation level, the unknown variables are waveforms (elements of a function space) in the case of WR method.

As in the case of algebraic equation level, both the GJ and the GS relaxations can be used at the differential equation level. As mentioned before, the speed of convergence of the GS method is heavily sensitive to the order in which the equations are arranged. For most MOS circuits, transistors are almost unidirectional from gate to source and from gate to drain. Therefore, the circuit equations can be properly ordered to give the GS iteration the speed advantage over the GJ iteration. If the circuit contains no MOS transmission gate nor any feedback connection, the circuit equation (3.3.1) can be reordered such that the system of equations are in the form:

$$C^i(v^1(t), \dots, v^i(t)) \dot{v}^i(t) + f^i(v^1(t), \dots, v^i(t), u(t)) = 0 \quad \text{for } i = 1, 2, \dots, n \quad (3.3.12)$$

In this special case, only one GS iteration is needed. This is also the main reason why relaxation methods have the speed advantage over conventional circuit simulators for the simulation of MOS circuits. Here we will only discuss the *Waveform-Relaxation Gauss-Seidel Algorithm* used in RELAX.

Consider the first-order differential equations as shown in Eqn. (3.3.1):

$$\begin{aligned}
 C^1(\mathbf{v}(t)) \dot{\mathbf{v}}(t) + f^1(\mathbf{v}(t), \mathbf{u}(t)) &= 0, & v^1(0) &= v^{1_0} \\
 &\vdots & & \\
 &\vdots & &
 \end{aligned} \tag{3.3.13}$$

$$C^n(\mathbf{v}(t)) \dot{\mathbf{v}}(t) + f^n(\mathbf{v}(t), \mathbf{u}(t)) = 0, \quad v^n(0) = v^{n_0}$$

The basic idea of this method is to fix the waveforms v^2, \dots, v^n at some initial guess waveforms and solve the first equation as a one-dimensional differential equation in v^1 . The solution obtained for v^1 is substituted into the second equation which will then be reduced to another first-order differential equation with one variable, v^2 . The new v^2 is again used in the third equation. This process continues until all the equations in Eqn. (3.3.13) are solved. Then the procedure is repeated until the waveforms converge to the solution waveforms. In mathematical term:

$$\begin{aligned}
 C^i(v_{(k+1)}^1(t), \dots, v_{(k+1)}^i(t), v_{(k)}^{i+1}(t), \dots, v_{(k)}^n(t)) &\begin{bmatrix} v_{(k+1)}^1(t) \\ \vdots \\ v_{(k+1)}^i(t) \\ v_{(k)}^{i+1}(t) \\ \vdots \\ v_{(k)}^n(t) \end{bmatrix} \\
 + f^i(v_{(k+1)}^1(t), \dots, v_{(k+1)}^i(t), v_{(k)}^{i+1}(t), \dots, v_{(k)}^n(t)) &= 0, \\
 \text{for } i = 0, 1, \dots, n &\quad \text{and } k = 0, 1, 2, \dots
 \end{aligned} \tag{3.3.14}$$

The outer iteration loop of WR method is the Gauss-Seidel iteration and the inner iteration loop can be any numerical integration method which can be used to solve the resulting one-dimensional differential equation. The convergence of WR method is guaranteed under some conditions similar to the ones needed in the linear or nonlinear algebraic equation cases. The convergence property is stated in the following theorem [14]:

Theorem 3.3.

Assume that $C(\mathbf{v}) \in R^{n \times n}$ is strictly diagonally dominant uniformly over all $\mathbf{v}(t) \in R^n$ and Lipschitz continuous with respect to $\mathbf{v}(t)$ for all $\mathbf{u}(t)$, and the initial guess waveforms are differentiable, then the sequences of waveforms $\{ v_{(k)}^i(t) \}$ generated by the Gauss-Seidel or the Gauss-Jacobi WR algorithm will converge uniformly to the solutions of the above equations for all bounded intervals $[0, T]$.

Waveform relaxation method has been implemented in RELAX. This simulator has been proven to be effective for the simulation of integrated circuits. More detailed description of the program RELAX will be given in a later section.

3.4. Timing Simulation

Timing simulation is a time domain circuit simulation technique which uses a particular non-linear relaxation approach to solve the nonlinear algebraic equations derived from discretizing the circuit nonlinear differential equations. The basic characteristic of timing simulation is that the iteration of the relaxation methods is not carried to convergence: only one sweep is taken. Because of this, the numerical properties such as consistence, stability, and convergence of the integration methods used to discretize the derivative operator no longer hold. In Section 3.4.2, a complete analysis of their numerical properties will be discussed.

Since only one sweep of relaxation is taken, the time steps must be kept sufficiently small to ensure the accuracy of the solutions of the nonlinear equations. However the computational expense of taking one iteration is very small. In addition, the total run time can be reduced even further due to the fact that timing analysis algorithms allow circuit latency to be exploited by using bypass or selective-trace techniques. Thus, the total computer time used in timing simulation is usually much less than that of a standard simulator. However, there are cases where these methods can not obtain an accurate solution. Examples are circuits containing tight feedback loops, pass transistors, or floating elements. We will discuss these problems in more detail in Section 3.4.3.

3.4.1. Timing Analysis Algorithm

To illustrate the basic steps of timing simulation, we again consider the circuits whose node equations can be written as in Eqn. (3.3.1):

$$C(\mathbf{v}(t)) \dot{\mathbf{v}}(t) + \mathbf{f}(\mathbf{v}(t), \mathbf{u}(t)) = 0 \quad \mathbf{v}(0) = \mathbf{v}_0 \quad (3.4.1)$$

To discuss timing analysis methods in this section, we need to assume that no floating capacitor (i.e. capacitors connected between two non-ground nodes) is present in the circuit. Therefore C is a diagonal matrix. We also assume that $C^{-1}(\mathbf{v})$ exists for all \mathbf{v} such that Eqn. (3.4.1) can be written as follows:

$$\dot{\mathbf{v}}(t) + \mathbf{F}(\mathbf{v}(t), \mathbf{u}(t)) = 0 \quad \mathbf{v}(0) = \mathbf{v}_0, \quad (3.4.2)$$

where

$$\mathbf{F}(\mathbf{v}(t), \mathbf{u}(t)) = C(\mathbf{v}(t))^{-1} \mathbf{f}(\mathbf{v}(t), \mathbf{u}(t)) \quad (3.4.3)$$

In timing simulation, the time derivative is replaced by any implicit integration formula such as the Backward Euler or the Trapezoidal formula. In this section, we only consider those methods of which the time derivative is discretized by the Backward Euler formula with constant time step:

$$\dot{\mathbf{v}}_{k+1} = \frac{1}{h} (\mathbf{v}_{k+1} - \mathbf{v}_k) \quad (3.4.4)$$

where \mathbf{v}_k and \mathbf{v}_{k+1} are the computed values of node voltages at time t_{k+1} and t_k , and $h = t_{k+1} - t_k$ is the time step. Thus, the iterative process of the nonlinear equation becomes:

$$\mathbf{v}_{k+1} - \mathbf{v}_k + h \mathbf{F}(\mathbf{v}_{k+1}, \mathbf{u}(t_{k+1})) = 0 \quad (3.4.5)$$

One sweep of nonlinear relaxation methods described in Section 3.3.2 can now be used on Eqn. (3.4.5). If we use one sweep of the Gauss-Jacobi relaxation on Eqn.(3.4.5), we get the following iterative process:

$$v_{k+1}^i = v_k^i - h F^i(v_k^1, \dots, v_{k+1}^i, v_k^{i+1}, \dots, v_k^n, \mathbf{u}(t_{k+1})) \quad (3.4.6)$$

$$i = 1, \dots, n$$

where i is the node index. If one sweep of the NR method is used to obtain the solution of Eqn. (3.4.6), this combined scheme is the one step Gauss-Jacobi-Newton method discussed in Section 3.3.2. with the exception that the outer loop Gauss-Jacobi iteration is carried out only once. More specifically, at each time point t_{k+1} , the Gauss-Jacobi-Newton method computes new values of all node voltages using only one iteration and accept the results as the correct solution at t_{k+1} and goes on to t_{k+2} . With some modification, this method has been used in MOTIS.

We can also use the Gauss-Seidel relaxation method to obtain the solution of Eqn. (3.4.5) as in SPLICE1.3[7]. The iteration process of the Gauss-Seidel method is:

$$v_{k+1}^i = v_k^i - h F^i(v_{k+1}^1, \dots, v_{k+1}^{i-1}, v_{k+1}^i, v_k^{i+1}, \dots, v_k^n, u(t_{k+1})) \quad (3.4.7)$$

$$i = 1, \dots, n$$

where i is the node index. The solution of Eqn. (3.4.7) is also approximated by using one step of the Newton-Raphson algorithm. This resulting timing analysis algorithm is also a combination of the implicit integration method with one sweep of one step the Gauss-Seidel-Newton method (both the inner and the outer loop iterations are carried out only once).

Note that neither method carries the iteration to convergence; In fact, only one sweep of relaxation iteration is taken. Hence the original numerical properties of the backward Euler integration method no longer hold. In the following section the numerical properties of these timing analysis methods which combine the discretization formula, relaxation steps and the Newton-Raphson method will be investigated.

3.4.2. Numerical Properties

The numerical properties of an integration method, such as stability, are studied on test problems[15,16] which are simple enough to allow a theoretical analysis but still general enough that one can gain insight into how the integration method behaves in general. For the analysis of the conventional multistep methods, the test problem consists of a linear, time-invariant, zero-input, asymptotically stable, differential equation.

Unfortunately, this simple test problem cannot be used to evaluate the timing analysis techniques described in the previous section. The test problem chosen should be a circuit that satisfies the following conditions:

- (1) It consists of positive linear time-invariant resistors and capacitors, and linear time-invariant voltage-controlled current sources.

- (2) Each node has a capacitor to ground or to dc supply voltage rails.
- (3) It is an asymptotically stable system.
- (4) There is no floating capacitor in the circuit.

Mathematically, this particular test problem can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0. \quad (3.4.8)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and the eigenvalues of \mathbf{A} are in the open left half complex plane. Let \mathbf{I} be the identity matrix and $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, where \mathbf{L} , \mathbf{D} and \mathbf{U} are all $n \times n$ matrices with \mathbf{L} being strictly lower triangular matrix, \mathbf{D} being diagonal, and \mathbf{U} being strictly upper triangular matrix. Now apply those algorithms discussed in the previous section to this test problem:

- (a) The Gauss-Jacobi integration algorithm:

$$[\mathbf{I} - h\mathbf{D}]\mathbf{x}_{t+1} = [\mathbf{I} + h(\mathbf{L} + \mathbf{U})]\mathbf{x}_t \quad (3.4.9)$$

$$\text{or} \quad \mathbf{x}_{t+1} = \mathbf{M}_{GJ}(h)\mathbf{x}_t \quad (3.4.10)$$

$$\text{where} \quad \mathbf{M}_{GJ} = [\mathbf{I} - h\mathbf{D}]^{-1}[\mathbf{I} + h(\mathbf{L} + \mathbf{U})]$$

- (b) The Gauss-Seidel integration algorithm:

$$[\mathbf{I} - h(\mathbf{D} + \mathbf{L})]\mathbf{x}_{t+1} = [\mathbf{I} + h\mathbf{U}]\mathbf{x}_t \quad (3.4.11)$$

$$\text{or} \quad \mathbf{x}_{t+1} = \mathbf{M}_{GS}(h)\mathbf{x}_t \quad (3.4.12)$$

$$\text{where} \quad \mathbf{M}_{GS} = [\mathbf{I} - h(\mathbf{D} + \mathbf{L})]^{-1}[\mathbf{I} + h\mathbf{U}]$$

Note that the test circuit is linear, hence the inner loop of the NR method is eliminated. The matrices $\mathbf{M}_{GJ}(h)$ and $\mathbf{M}_{GS}(h)$ are called the companion matrices of the methods. If we denote $\mathbf{M}(h)$ as the generic companion matrix of a method, then the method to be investigated using a fixed step size h will give the following solution:

$$\mathbf{x}_t = [\mathbf{M}(h)]^t \mathbf{x}_0 \quad (3.4.13)$$

We now define the numerical properties of the integration algorithm described by Eqn. (3.4.13).

Definition 3.1 (convergence)

Let $\mathbf{x}(t)$ be the exact solution of the test problem. An integration algorithm is convergent if the sequence of the computed solution, \mathbf{x}_k , converges uniformly to $\mathbf{x}(t)$ as the step-size h goes to zero.

One useful measure of the error introduced by the integration technique is the local truncation error whose definition is given as follows:

Definition 3.2 (local truncation error)

Let $\mathbf{x}(t_k)$ be the exact solution of the test problem at time t_k , \mathbf{x}_k be the computed solution at time t_k assuming $\mathbf{x}_{k-1} = \mathbf{x}(t_{k-1})$ (i.e. no error has been made in computing \mathbf{x} at previous time point), and $h = t_k - t_{k-1}$, then the local truncation error ε is defined as:

$$\varepsilon = \|\mathbf{x}(t_k) - \mathbf{x}_k\| \quad (3.4.14)$$

If $\varepsilon = O(h^{r+1})$, then r is said to be the order of the integration method.

Using this definition, we have the following definition for consistency:

Definition 3.3 (consistency)

The numerical method is said to be consistent if

$$\mathbf{x}(t) - \mathbf{x}_k = O(h^p)$$

where $p \geq 2$ for all k .

Based on the above definition, we know that a timing algorithm is consistent if its companion matrix can be expanded as a power series of the step-size h of the following form:

$$\mathbf{M}(h) = \mathbf{I} + h\mathbf{A} + O(h^2)$$

Convergence implies that if the step size is chosen sufficiently small, then the numerical solution can be made arbitrarily close to the exact solution. Consistency only ensures that the local errors are small, but does not tell us anything about how the errors propagate from one time point to the next. To insure convergence of a numerical integration method, we need to verify that it is also stable.

Definition 3.4 (stability)

An integration algorithm is stable if for any $\delta > 0$, $M > 0$ and for all $x_0 \in R^n$, there exists a $\bar{k} > 0$ such that the difference between two numerical solutions generated from two different initial values is bounded. i.e.

$$\|x_k - x'_k\| < M \|x_0 - x'_0\| \quad \text{for all } k \geq \bar{k}, \text{ and for all } h \text{ between } [0, \delta)$$

where x_k and x'_k are the computed sequences given the initial values x_0 and x'_0 respectively.

Stability implies that a small perturbation in the initial value can only cause a bounded change in the answer as h approaches zero. Since each step in the timing analysis algorithms is effectively a new initial value problem, stability guarantees that the change due to a small perturbation at any step in the computation is bounded. The following classical theorem relates consistency, stability, and convergence together.

Theorem 3.4:[15]

If a numerical integration method is consistent and stable, then it is convergent.

The numerical properties of the Gauss-Jacobi and the Gauss-Seidel integration algorithms have been discussed in [17]. It states that the Gauss-Jacobi and the Gauss-Seidel integration algorithms are consistent, stable, hence convergent. Moreover, the Gauss-Jacobi and the Gauss-Seidel integration methods are first-order integration algorithms.

To extend the analysis of stability, let us define A-Stability.

Definition 3.5. (A-Stability)

An integration method is A-stable if for any $N > 0$ and for all $x_0 \in R^n$, there exists a \bar{k} such that

$$\|x_k\| < N \quad \text{for all } k \geq \bar{k} \quad \text{and all } h \text{ between } [0, \infty)$$

where $\{x_k\}$ is the sequence generated by the method.

Stiff problems arise when the circuits under simulation contain bypass or coupling capacitors(inductors) which are several orders of magnitude larger than the parasitic elements of the circuits. This often implies that the solutions contain both the fast components due to the parasitic elements and the slow components due to the bypass and coupling elements.

To analyze stiff circuits effectively, one must use variable step sizes, so that, after the fast transient response has died off, the step size can be increased to quickly observe the slow transient response. To do so, the region of stability of the integration method should be large enough to allow a large step size for large time constants, without being constrained by the small time constants. Clearly, if the method is A-stable, it can be effectively used to handle stiff circuits. It is reasonable to ask under what conditions the GJ and the GS timing analysis methods are suitable for solving stiff circuits. To do so, we need to use the following proposition:

Proposition 3.1 [18]

The sequence of vectors $\mathbf{x}_k = [M(h)]^k \mathbf{x}_0$ is bounded for a given h if and only if the spectrum of $M(h)$ is contained in the unit ball $B(0, 1)$, i.e. all eigenvalues of $M(h)$, $\sigma(M(h))$, are contained in $B(0, 1)$, and no multiple zeros of the minimal polynomial of $M(h)$ has modulus equal to 1.

Theorem 3.5

The Gauss-Jacobi method and the Gauss-Seidel method are A-stable when applied to the test problem described above if A is negative definite and strictly diagonal dominant.

To be effective in circuit analysis, an integration method has to use variable time-step. Standard integration methods control the time step by monitoring the local truncation error. Since these timing algorithms do not carry the nonlinear iteration to convergence, computing the local truncation error of the timing analysis algorithms is very complex. Fortunately, from our analysis, we found that any error contributed by taking only one NR iteration is negligible. This property certainly simplifies the computation of local truncation error.

Theorem 3.6:

The Gauss-Jacobi and Gauss-Seidel integration methods are first order algorithms, and taking only one sweep of the Newton-Raphson method does not produce any first order error.

3.4.3. Problems with Timing Analysis Algorithms

A major problem with timing analysis algorithms is its inaccuracy and instability when they are used to analyze circuits with tightly-coupled feedback loops or bidirectional circuit elements. One such element is the floating capacitor which is often an important element in the design of integrated circuits.

To see the effect of floating capacitors on the timing analysis algorithm, let us consider a linear time-invariant circuit described by

$$C\dot{v} = -Gv \quad v(0) = v_0 \quad (3.4.15)$$

where C is the node capacitance matrix and G is the node conductance matrix. If there is no floating capacitor, then C is diagonal and can easily be inverted. In this case, the analysis described previously applies. When floating capacitors exist, C is no longer diagonal, inverting C is expensive and most of the advantages of the timing simulation algorithms would be lost. If we do not invert C and just apply the timing analysis algorithms directly on Eqn. (3.4.28), then all the algorithms described previously are not even consistent.

Another major drawback of the timing analysis algorithms is that these algorithms are A-stable only under some strict conditions. This forces us to use small step sizes in the numerical computation even though large step sizes may be acceptable. For a technique to be robust, one will prefer any A-stable numerical integration method over these timing analysis algorithms because the time step of an A-stable method can be safely chosen as long as it satisfies the local truncation error criteria. Since these timing analysis methods is not A-stable in all cases, the time step must also be bounded to ensure stability. This requires some knowledge of the time constant for the circuit under analysis. However, it is difficult to estimate the time constant of a circuit, thus these algorithms have to rely on the user to intelligently select an appropriate time step.

Although timing analysis algorithms improve the computational speed by taking one sweep of displacement method, all these problems described above are also due to the fact that the relaxation iterations are not carried to convergence. This severely limits the application of these techniques to

some restricted class of circuit topologies.

CHAPTER 4

The Implicit-Implicit-Explicit Method

As mentioned in Chapter 3, one major drawback of the timing analysis algorithms is the accuracy and stability problems when the circuit under analysis contains elements that form tightly-coupled feedback loops. One such element is the floating capacitor. For this reason, special techniques must be used. The Implicit-Implicit-Explicit (IIE) method has been proposed to handle floating capacitors. In this chapter, we formalize the IIE method and present its numerical properties using the frame work set up in the previous chapter. In particular, we show that the method is consistent, stable and as accurate as the backward Euler method even when floating capacitors are present in the circuit. The details of this algorithm are described in Section 4.1. In Section 4.2, we will investigate its numerical properties, such consistency, stability, and convergence on a test problem. In Section 4.3, some implementation results are presented. Then, the concluding remarks are given in Section 4.4.

4.1. Mathematical Formulation

In this section, we will study the IIE algorithm by applying it to the following system:

$$C(\mathbf{v})\dot{\mathbf{v}} + \mathbf{f}(\mathbf{v}, \mathbf{u}(t)) = 0 \quad \mathbf{v}(0) = \mathbf{v}_0 \quad (4.1.1)$$

Note that we allow floating capacitors to be present in the system.

Before we study this general system, let us first examine a two-node circuit. Applying the backward Euler integration method and nodal analysis to the circuit of Fig. 4.1, we have:

$$\begin{bmatrix} G_1 + G_3 + \frac{C_1 + C_3}{h} & -G_3 - \frac{C_3}{h} \\ -G_3 - \frac{C_3}{h} & G_2 + G_3 + \frac{(C_2 + C_3)}{h} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}_{k+1}$$

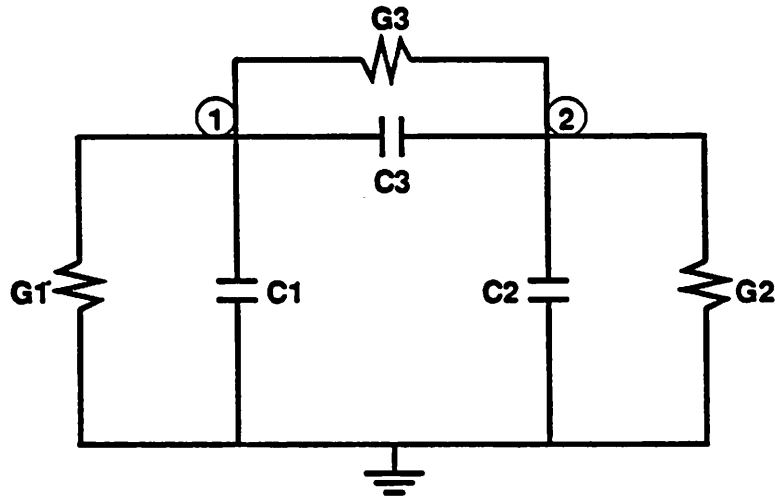


Figure 4.1 Circuit Example

$$= \begin{bmatrix} \frac{(C_1 + C_3)}{h} & -\frac{C_3}{h} \\ -\frac{C_3}{h} & \frac{(C_2 + C_3)}{h} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}_k \quad (4.1.2)$$

where $h = t_{k+1} - t_k$

The key to the IIE method is to decouple Eqn. (4.1.2) by taking the voltage at node (2) one step back in time when solving the first equation. Thus, $G_3 v_{k+1}^2$ is replaced by $G_3 v_k^2$ and the current flowing through the floating capacitor is given by $\frac{C_3}{h} [(v_{k+1}^1 - v_k^1) - (v_k^2 - v_{k-1}^2)]$. Therefore, Eqn. (4.1.2) becomes:

$$\begin{bmatrix} G_1 + G_3 + \frac{C_1 + C_3}{h} & 0 \\ -G_3 - \frac{C_3}{h} & G_2 + G_3 + \frac{(C_2 + C_3)}{h} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}_{k+1}$$

$$= \begin{bmatrix} \frac{(C_1 + C_3)}{h} & -G_3 - \frac{C_3}{h} \\ -\frac{C_3}{h} & \frac{(C_2 + C_3)}{h} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}_k + \begin{bmatrix} 0 & -\frac{C_3}{h} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}_{k-1} \quad (4.1.3)$$

The IIE method has been shown[3] to be unconditionally stable for this particular two node circuit (i.e. all eigenvalues of the companion matrix are less than 1 for all values of the circuit parameters and for all h). In addition, this method is free of oscillatory error for this two node circuit if we keep the time step sufficiently small.

In order to generalize this method, consider the following splitting:

$$C(v) \equiv C_l(v) + C_u(v)$$

where $C_u(v)$ is strictly upper triangular and $C_l(v)$ is lower triangular with nonzero diagonal entries.

Eqn. (4.1.1) can be rewritten as follows:

$$C_l(v)\dot{v} + C_u(v)\dot{v} + f(v, u(t)) = 0 \quad (4.1.4)$$

To solve Eqn. (4.1.4), the time derivative operator is discretized to convert the nonlinear differential equation into a nonlinear algebraic equation. In contrast to other timing algorithms, this algorithm discretizes the derivative operator in Eqn. (4.1.4) at two different time points, t_{k+1} and t_k , i.e.:

$$C_l(v_{k+1})\dot{v}_{k+1} + C_u(v_{k+1})\dot{v}_k + f(v_{k+1}, u(t_{k+1})) = 0 \quad (4.1.5)$$

where \dot{v}_{k+1} and \dot{v}_k are the time derivatives of the vector v evaluated at t_{k+1} and t_k , respectively. If

the backward Euler integration formula is used, then we have:

$$\dot{v}_{k+1} = \frac{v_{k+1} - v_k}{h} \quad \dot{v}_k = \frac{v_k - v_{k-1}}{h} \quad (4.1.6)$$

where $h \equiv t_{k+1} - t_k = t_k - t_{k-1}$, then Eqn. (4.1.5) becomes:

$$\begin{aligned} C_l(v_{k+1})v_{k+1} - C_l(v_{k+1})v_k + C_u(v_{k+1})v_k - C_u(v_{k+1})v_{k-1} \\ = -hf(v_{k+1}, u(t_{k+1})) \end{aligned} \quad (4.1.7)$$

Using one sweep of Gauss-Seidel, we have:

$$\begin{aligned}
 & \mathbf{C}_I^*(v_k, v_{k+1})v_{k+1} + h \begin{bmatrix} f_1(v_{k+1}^1, v_k^2, \dots, v_k^n, \mathbf{u}(t_{k+1})) \\ f_2(v_{k+1}^1, v_{k+1}^2, v_k^3, \dots, v_k^n, \mathbf{u}(t_{k+1})) \\ \dots \\ f_n(v_{k+1}^1, \dots, v_{k+1}^{n-1}, v_{k+1}^n, \mathbf{u}(t_{k+1})) \end{bmatrix} \\
 & = [\mathbf{C}_I^*(v_k, v_{k+1}) - \mathbf{C}_u^*(v_k, v_{k+1})]v_k + \mathbf{C}_u^*(v_k, v_{k+1})v_{k-1} \quad (4.1.8)
 \end{aligned}$$

where $\mathbf{C}_I^*(v_k, v_{k+1})$ and $\mathbf{C}_u^*(v_k, v_{k+1})$ are defined as follows:

$$\mathbf{C}_I^*(v_k, v_{k+1}) \equiv \begin{bmatrix} C_I^{11}(v_{k+1}^1, v_k^2, \dots, v_k^n) & 0 & 0 & 0 \\ C_I^{21}(v_{k+1}^1, v_{k+1}^2, v_k^3, \dots, v_k^n) & C_I^{22}(v_{k+1}^1, v_{k+1}^2, v_k^3, \dots, v_k^n) & 0 & 0 \\ \dots & \dots & \dots & \dots \\ C_I^{n1}(v_{k+1}^1, \dots, v_{k+1}^n) & C_I^{n2}(v_{k+1}^1, \dots, v_{k+1}^n) & \dots & C_I^{nn}(v_{k+1}^1, \dots, v_{k+1}^n) \end{bmatrix}$$

and

$$\mathbf{C}_u^*(v_k, v_{k+1}) \equiv \begin{bmatrix} 0 & C_u^{12}(v_{k+1}^1, v_k^2, \dots, v_k^n) & C_u^{13}(v_{k+1}^1, v_k^2, \dots, v_k^n) & \dots & C_u^{n1}(v_{k+1}^1, v_k^2, \dots, v_k^n) \\ 0 & 0 & C_u^{21}(v_{k+1}^1, v_{k+1}^2, v_k^3, \dots, v_k^n) & \dots & \dots \\ \vdots & \vdots & \vdots & \dots & \dots \\ 0 & \vdots & \vdots & \dots & C_u^{n-1,n}(v_{k+1}^1, \dots, v_{k+1}^{n-1}, v_k^n) \\ & & & & 0 \end{bmatrix}$$

Eqn. (4.1.8) is the formalized version of the IIE method when applied to general circuits. Since there is only one unknown in each row of Eqn. 4.1.8, the solution can be readily obtained.

In general, when a circuit description is read by a timing simulator and translated into data structures in memory, the elements may be read in any order. Unless some form of connection graph is used to establish a precedence order for signal flow, the new node voltage will be computed in an arbitrary order. Since the Gauss-Seidel relaxation is used in the IIE method, the order of processing elements can substantially affect the simulator performance. To improve the performance of the IIE method, one must include a scheduling routine to properly order the system of equations.

In addition, large digital circuits are often relatively inactive. There have been a number of schemes used to avoid the unnecessary computation involving the reevaluation of voltages at nodes which are inactive or *latent*. The selective trace technique used in SPLICE saves a significant amount of computer time. For the IIE method, bypass schemes such as selective trace technique

may be used to exploit latency.

Since only one sweep of the Gauss-Seidel iteration is taken, the stability and accuracy properties of the backward Euler integration method used to discretize the derivative operator no longer hold. In the following section, the numerical properties of this formalized IIE method will be investigated.

Before leaving this section, we would like to mention that in order to compute \mathbf{v}_{k+1} in Eqn. (4.1.6), one needs to know \mathbf{v}_k and \mathbf{v}_{k-1} . To compute \mathbf{v}_1 given \mathbf{v}_0 , one can use any timing analysis method that computes \mathbf{v}_{k+1} based only on \mathbf{v}_k . Once \mathbf{v}_0 and \mathbf{v}_1 are obtained, the algorithm described in this section can be used.

4.2. Numerical Properties of the IIE Method

The test problem chosen for the IIE method is a circuit similar to the test problem described in Section 3.3.2 with the exception that floating capacitors are present. The test circuit must satisfy the following conditions:

- (1) It consists of positive, linear, time-invariant resistors and capacitors, and linear, time-invariant voltage-controlled current sources.
- (2) Each node has a capacitor connected to ground or to dc supply voltage rails.
- (3) It is asymptotically stable.

For this particular class of test problems, Eqn. (4.1.1) can be written as:

$$\mathbf{C}\dot{\mathbf{v}} = -\mathbf{G}\mathbf{v} \quad \mathbf{v}(0) = \mathbf{v}_0 \quad (4.2.1)$$

By conditions (1) and (2), \mathbf{C} is strictly diagonally dominant and \mathbf{G} is the nodal conductance matrix of the circuit with controlled sources. Now let us apply the IIE algorithm described in Section 4.1 to Eqn. (4.2.1) to get the following equation:

$$\mathbf{C}_l \left[\frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{h} \right] + \mathbf{C}_u \left[\frac{\mathbf{v}_k - \mathbf{v}_{k-1}}{h} \right] = -\mathbf{G}_l \mathbf{v}_{k+1} - \mathbf{G}_u \mathbf{v}_k \quad (4.2.2)$$

where \mathbf{C}_l , \mathbf{G}_l are lower triangular matrices (including diagonal elements), and \mathbf{C}_u and \mathbf{G}_u are

strictly upper triangular matrices.

Theorem 4.1 [19]

The IIE algorithm is consistent.

Proof:

Rewrite Eqn. (4.2.2) in a different form:

$$C_I [\mathbf{v}_{k+1} - \mathbf{v}_k] + C_u [\mathbf{v}_k - \mathbf{v}_{k-1}] = -h \mathbf{G}_I \mathbf{v}_{k+1} - h \mathbf{G}_u \mathbf{v}_k \quad (4.2.3)$$

Let $\mathbf{v}(t_k)$ be the exact solution of $C \dot{\mathbf{v}} = -\mathbf{G}\mathbf{v}$, and \mathbf{v}_k be the computed solution using the IIE method. For consistency analysis, we wish to show that the local truncation error is $O(h^2)$, i.e.

$$\mathbf{v}_{k+1} = \mathbf{v}(t_{k+1}) + O(h^2) \quad (4.2.4)$$

The key tool for proving consistency is to use the Taylor series expansion of the exact solution at t_k while assuming that the previously computed values are equal to the exact solution:

$$\mathbf{v}(t_{k+1}) = \mathbf{v}(t_k) + h \dot{\mathbf{v}}(t_k) + O(h^2) \quad (4.2.5)$$

$$\mathbf{v}(t_{k-1}) = \mathbf{v}(t_k) - h \dot{\mathbf{v}}(t_k) + O(h^2) \quad (4.2.6)$$

$$\mathbf{v}_{k-1} = \mathbf{v}(t_{k-1}) \quad (4.2.7)$$

$$\mathbf{v}_k = \mathbf{v}(t_k) \quad (4.2.8)$$

From Eqn. (4.2.5) and Eqn. (4.2.6), we have:

$$\mathbf{v}(t_k) - \mathbf{v}(t_{k-1}) = \mathbf{v}(t_{k+1}) - \mathbf{v}(t_k) + O(h^2) \quad (4.2.9)$$

Substituting Eqn. (4.2.7) and (4.2.8) into Eqn. (4.2.3), we get the following equation:

$$C_I [\mathbf{v}_{k+1} - \mathbf{v}(t_k)] + C_u [\mathbf{v}(t_k) - \mathbf{v}(t_{k-1})] = -h \mathbf{G}_I \mathbf{v}_{k+1} - h \mathbf{G}_u \mathbf{v}(t_k) \quad (4.2.10)$$

Using Eqn. (4.2.9), we have:

$$C_I [\mathbf{v}_{k+1} - \mathbf{v}(t_k)] + C_u [\mathbf{v}(t_{k+1}) - \mathbf{v}(t_k)] = -h \mathbf{G}_I \mathbf{v}_{k+1} - h \mathbf{G}_u \mathbf{v}(t_k) + O(h^2) \quad (4.2.11)$$

Consider the original equation:

$$C \dot{\mathbf{v}} = -\mathbf{G}\mathbf{v} \quad \mathbf{v}(0) = \mathbf{v}_0$$

If we evaluate this equation at time t_k , we have:

$$C[\dot{\mathbf{v}}(t_k)] = -\mathbf{G}\mathbf{v}(t_k) \quad (4.2.12)$$

Since

$$\mathbf{v}(t_{k+1}) = \mathbf{v}(t_k) + h \dot{\mathbf{v}}(t_k) + O(h^2) \quad (4.2.13)$$

Eqn. (4.2.12) becomes:

$$\mathbf{C}[\mathbf{v}(t_{k+1}) - \mathbf{v}(t_k) + O(h^2)] = -h \mathbf{G} \mathbf{v}(t_k) \quad (4.2.14)$$

or

$$[\mathbf{C}_I + \mathbf{C}_u][\mathbf{v}(t_{k+1}) - \mathbf{v}(t_k) + O(h^2)] = -h [\mathbf{G}_I + \mathbf{G}_u] \mathbf{v}(t_k) \quad (4.2.15)$$

Rewrite Eqn. (4.2.15):

$$\begin{aligned} & \mathbf{C}_I [\mathbf{v}(t_{k+1}) - \mathbf{v}(t_k)] + \mathbf{C}_u [\mathbf{v}(t_{k+1}) - \mathbf{v}(t_k)] \\ & = -h \mathbf{G}_I \mathbf{v}(t_k) - h \mathbf{G}_u \mathbf{v}(t_k) + O(h^2) \end{aligned} \quad (4.2.16)$$

Subtract Eqn. (4.2.16) from Eqn. (4.2.11), we have:

$$\mathbf{C}_I [\mathbf{v}_{k+1} - \mathbf{v}(t_{k+1})] = -h \mathbf{G}_I [\mathbf{v}_{k+1} - \mathbf{v}(t_k)] + O(h^2) \quad (4.2.17)$$

Rearranging Eqn. (4.2.17), we can get:

$$[\mathbf{v}_{k+1} - \mathbf{v}(t_{k+1})] = -h \mathbf{C}_I^{-1} \mathbf{G}_I [\mathbf{v}_{k+1} - \mathbf{v}(t_k)] + O(h^2) \quad (4.2.18)$$

Since $\mathbf{v}(t_k) = \mathbf{v}(t_{k+1}) + O(h)$, Eqn. (4.2.18) becomes:

$$[\mathbf{I} + h \mathbf{C}_I^{-1} \mathbf{G}_I][\mathbf{v}_{k+1} - \mathbf{v}(t_{k+1})] = O(h^2) \quad (4.2.19)$$

Now, using the series expansion $[\mathbf{I} + h \mathbf{C}_I^{-1} \mathbf{G}_I]^{-1} = \mathbf{I} + O(h)$, we have:

$$[\mathbf{v}_{k+1} - \mathbf{v}(t_{k+1})] = [\mathbf{I} + O(h)] O(h^2) = O(h^2) \quad (4.2.20)$$

Hence, we conclude that the IIE method is consistent.

Q.E.D.

Corollary 4.1

The IIE algorithm is a first order integration algorithm.

Proof:

The proof follows directly from Theorem 4.1.

Q.E.D.

It is difficult to study the stability of this method by analyzing Eqn. (4.2.2) since it involves the evaluation of \mathbf{v} at three different time points. Let us make the following definition:

$$\mathbf{x}_k \equiv \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{v}_{k-1} \\ \mathbf{v}_k \end{bmatrix}$$

Then, Eqn. (4.2.2) can be rewritten as :

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0 & \mathbf{I} \\ (\mathbf{C}_l + h\mathbf{G}_l)^{-1}\mathbf{C}_u & (\mathbf{C}_l + h\mathbf{G}_l)^{-1}(\mathbf{C}_l - \mathbf{C}_u - h\mathbf{G}_u) \end{bmatrix} \mathbf{x}_k \quad (4.2.21)$$

or

$$\mathbf{x}_{k+1} = \mathbf{M}(h)\mathbf{x}_k \quad (4.2.22)$$

where

$$\mathbf{M}(h) \equiv \begin{bmatrix} 0 & \mathbf{I} \\ (\mathbf{C}_l + h\mathbf{G}_l)^{-1}\mathbf{C}_u & (\mathbf{C}_l + h\mathbf{G}_l)^{-1}(\mathbf{C}_l - \mathbf{C}_u - h\mathbf{G}_u) \end{bmatrix}$$

$\mathbf{M}(h)$ is called the companion matrix of the method. If we assume that a fixed step size h is used, then we have:

$$\mathbf{x}_{k+1} = [\mathbf{M}(h)]^{k+1} \mathbf{x}_0 \quad (4.2.23)$$

Note that the exact solution of Eqn. (4.2.1) at t_{k+1} assuming \mathbf{v}_k being computed exactly is given by:

$$\mathbf{v}_{k+1} = e^{-\mathbf{C}^{-1}\mathbf{G}} \mathbf{v}_k \quad (4.2.24)$$

Expanding the right hand side of Eqn. (4.2.24) as a power series a of the step size h , we have:

$$\mathbf{v}_{k+1} = [\mathbf{I} - h\mathbf{C}^{-1}\mathbf{G} + O(h^2)] \mathbf{v}_k \quad (4.2.25)$$

or

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & \mathbf{I} - h\mathbf{C}^{-1}\mathbf{G} + O(h^2) \end{bmatrix} \mathbf{x}_k \quad (4.2.26)$$

Since we have proven that the IIE method is consistent, the companion matrix of the IIE method should have the same form as the matrix given in the right hand side of Eqn. (4.2.26) with this new variable definition. This property is restated in the following lemma:

Lemma 4.1

The IIE method is consistent and its companion matrix can be expanded as a power series of the step-size h as:

$$M(h) = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & \mathbf{I} \end{bmatrix} + h \begin{bmatrix} 0 & 0 \\ 0 & -\mathbf{C}^{-1}\mathbf{G} \end{bmatrix} + O(h^2) \quad (4.2.27)$$

Proof:

We first rewrite Eqn. (4.2.2) in a different form:

$$(\mathbf{C}_l + h\mathbf{G}_l)\mathbf{v}_{k+1} = (\mathbf{C}_l - \mathbf{C}_u - h\mathbf{G}_u)\mathbf{v}_k + \mathbf{C}_u\mathbf{v}_{k-1} \quad (4.2.28)$$

If we subtract $(\mathbf{C}_l + h\mathbf{G}_l)\mathbf{v}_k$ from both side of Eqn. (4.2.28), we will get:

$$(\mathbf{C}_l + h\mathbf{G}_l)(\mathbf{v}_{k+1} - \mathbf{v}_k) = (-\mathbf{C}_u - h\mathbf{G}_u - h\mathbf{G}_l)\mathbf{v}_k + (\mathbf{C}_u)\mathbf{v}_{k-1} \quad (4.2.29)$$

Since the IIE method is consistent, we have:

$$\mathbf{v}_{k+1} = \mathbf{v}_k + h\dot{\mathbf{v}}_k + O(h^2)$$

$$\text{or } \mathbf{v}_{k+1} - \mathbf{v}_k = h\dot{\mathbf{v}}_k + O(h^2) \quad (4.2.30)$$

and

$$\mathbf{v}_k = \mathbf{v}_{k-1} + h\dot{\mathbf{v}}_k + O(h^2)$$

$$\text{or } \mathbf{v}_k - \mathbf{v}_{k-1} = h\dot{\mathbf{v}}_k + O(h^2) \quad (4.2.31)$$

With a first-order approximation, we can ignore $O(h^2)$. Thus,

$$\mathbf{v}_{k+1} - \mathbf{v}_k = \mathbf{v}_k - \mathbf{v}_{k-1} \quad (4.2.32)$$

Substituting Eqn. (4.2.32) into Eqn. (4.2.29), we have:

$$(\mathbf{C}_l + h\mathbf{G}_l)(\mathbf{v}_k - \mathbf{v}_{k-1}) = (-\mathbf{C}_u - h\mathbf{G}_u - h\mathbf{G}_l)\mathbf{v}_k + (\mathbf{C}_u)\mathbf{v}_{k-1} \quad (4.2.33)$$

Thus, we can solve \mathbf{v}_{k-1} as a function of \mathbf{v}_k :

$$(\mathbf{C} + h\mathbf{G} + h\mathbf{G}_l)\mathbf{v}_k = (\mathbf{C} + h\mathbf{G}_l)\mathbf{v}_{k-1} \quad (4.2.34)$$

$$\text{or } \mathbf{v}_{k-1} = (\mathbf{C} + h\mathbf{G}_l)^{-1}(\mathbf{C} + h\mathbf{G} + h\mathbf{G}_l)\mathbf{v}_k \quad (4.2.35)$$

Substituting Eqn. (4.2.35) into Eqn. (4.2.28), we have the following equation:

$$\begin{aligned} (\mathbf{C}_l + h\mathbf{G}_l)\mathbf{v}_{k+1} &= (\mathbf{C}_l - \mathbf{C}_u - h\mathbf{G}_u)\mathbf{v}_k \\ &+ (\mathbf{C}_u)(\mathbf{C} + h\mathbf{G}_l)^{-1}(\mathbf{C} + h\mathbf{G} + h\mathbf{G}_l)\mathbf{v}_k \end{aligned} \quad (4.2.36)$$

Eqn. (4.2.36) can be simplified into the equation given below:

$$\mathbf{v}_{k+1} = (\mathbf{C}_l + h\mathbf{G}_l)^{-1}[\mathbf{C}_l - h\mathbf{G}_u + \mathbf{C}_u(\mathbf{C} + h\mathbf{G}_l)^{-1}h\mathbf{G}]\mathbf{v}_k \quad (4.2.37)$$

Define

$$\mathbf{M}^*(h) = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & (\mathbf{C}_l + h\mathbf{G}_l)^{-1}[\mathbf{C}_l - h\mathbf{G}_u + \mathbf{C}_u(\mathbf{C} + h\mathbf{G}_l)^{-1}h\mathbf{G}] \end{bmatrix} \quad (4.2.38)$$

Since $\mathbf{M}^*(h)$ was derived based on the expansions of \mathbf{v}_{k+1} and \mathbf{v}_k as given in Eqn. (4.2.30) and Eqn. (4.2.31), $\mathbf{M}^*(h)$ is equal to $\mathbf{M}(h)$ up to the first order of h .

Therefore, $\mathbf{M}(h) = \mathbf{M}^*(0) + h\frac{d}{dh}\mathbf{M}^*(0) + O(h^2)$.

We need to show:

$$i). \mathbf{M}^*(0) = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & \mathbf{I} \end{bmatrix}$$

$$ii). \frac{d}{dh}\mathbf{M}^*(0) = \begin{bmatrix} 0 & 0 \\ 0 & -\mathbf{C}^{-1}\mathbf{G} \end{bmatrix}$$

i. Clearly, by setting $h = 0$, we have:

$$\mathbf{M}^*(0) = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & \mathbf{I} \end{bmatrix} \quad (4.2.39)$$

ii.

$$\frac{d}{dh}\mathbf{M}^*(h) = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{B} \end{bmatrix} \quad (4.2.40)$$

$$\begin{aligned} \text{where } \mathbf{B} = & -(\mathbf{C}_l + h\mathbf{G}_l)^{-1}\mathbf{G}_l(\mathbf{C}_l + h\mathbf{G}_l)^{-1}[\mathbf{C}_l - h\mathbf{G}_u + \mathbf{C}_u(\mathbf{C} + h\mathbf{G}_l)^{-1}h\mathbf{G} \\ & + (\mathbf{C}_l + h\mathbf{G}_l)^{-1}[-\mathbf{G}_u + \mathbf{C}_u(-(\mathbf{C} + h\mathbf{G}_l)^{-1}\mathbf{G}_l(\mathbf{C} + h\mathbf{G}_l)^{-1}h\mathbf{G} + (\mathbf{C} + h\mathbf{G}_l)^{-1}\mathbf{G})] \end{aligned}$$

Setting $h = 0$, we have:

$$\frac{d}{dh}\mathbf{M}^*(0) = \begin{bmatrix} 0 & 0 \\ 0 & -\mathbf{C}_l^{-1}\mathbf{G}_l\mathbf{C}_l^{-1}\mathbf{C}_l + \mathbf{C}_l^{-1}(-\mathbf{G}_u + \mathbf{C}_u\mathbf{C}^{-1}\mathbf{G}) \end{bmatrix} \quad (4.2.41)$$

or

$$\frac{d}{dh}\mathbf{M}^*(0) = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_l^{-1}(-\mathbf{I} + \mathbf{C}_u\mathbf{C}^{-1})\mathbf{G} \end{bmatrix} \quad (4.2.42)$$

Substitute \mathbf{I} by $\mathbf{C}\mathbf{C}^{-1}$, Eqn. (4.2.42) becomes:

$$\begin{aligned} \frac{d}{dh}\mathbf{M}^*(0) &= \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_l^{-1}(-\mathbf{C}\mathbf{C}^{-1} + \mathbf{C}_u\mathbf{C}^{-1})\mathbf{G} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & -\mathbf{C}_l^{-1}(\mathbf{C} - \mathbf{C}_u)\mathbf{C}^{-1}\mathbf{G} \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & -C^{-1}C_1C^{-1}G \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -C^{-1}G \end{bmatrix}$$

Hence

$$\begin{aligned} M(h) &= M^*(0) + h \frac{d}{dh} M^*(0) + O(h^2) \\ &= \begin{bmatrix} 0 & I \\ 0 & I \end{bmatrix} + h \begin{bmatrix} 0 & 0 \\ 0 & -C^{-1}G \end{bmatrix} + O(h^2) \end{aligned} \quad (4.2.43)$$

Q.E.D.

We will now discuss the stability characteristic of the IIE method. Note that stability means that the sequence $\{v_k\}$ be bounded for small step size. In this chapter, we restrict the analysis to constant step-size. Using proposition 3.1, we can prove the following theorem:

Theorem 4.2.

The IIE algorithm is stable.

Proof:

It follows from Lemma 4.1 that:

$$M(h) = \begin{bmatrix} 0 & I \\ 0 & I - hC^{-1}G \end{bmatrix} + O(h^2) \quad (4.2.44)$$

$$\text{or } \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & I \\ 0 & I - hC^{-1}G \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + O(h^2) \quad (4.2.45)$$

Since $x_{1,k+1} = x_{2,k}$

$$x_{2,k+1} = (I - hC^{-1}G)x_{2,k}$$

$$x_{1,k+1} = (I - hC^{-1}G)x_{1,k}$$

we have:

$$\begin{aligned} M(h) &= \begin{bmatrix} I - hC^{-1}G & 0 \\ 0 & I - hC^{-1}G \end{bmatrix} + O(h^2) \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + h \begin{bmatrix} -C^{-1}G & 0 \\ 0 & -C^{-1}G \end{bmatrix} + O(h^2) \end{aligned} \quad (4.2.46)$$

Clearly,

$$\sigma \left[\begin{bmatrix} -C^{-1}G & 0 \\ 0 & -C^{-1}G \end{bmatrix} \right] = \sigma \left[\left[-C^{-1}G \right] \right]$$

By the spectral mapping theorem [18] :

$$\sigma(M(h)) = \left\{ \xi_i \mid \xi_i = 1 + h\lambda_i + O(h^2) \right\},$$

$$\text{where } \lambda \text{ is an element of } \sigma(-C^{-1}G); i = 1, 2, \dots, \sigma \quad (4.2.47)$$

From Eqn. (4.2.47), we have:

$$|\xi_i| = |1 + h\lambda_i + O(h^2)|, \quad i = 1, 2, \dots, \sigma \quad (4.2.48)$$

$$|\xi_i|^2 = [1 + h \operatorname{Re}(\lambda_i)]^2 + [h \operatorname{Im}(\lambda_i)]^2 + O(h^2) \quad (4.2.49)$$

Since $M^*(0) = I$, its eigenvalues are all 1, and 1 is a simple zero of the minimal polynomial of the identity matrix.

$$|\xi_i|^2 = 1 + 2h \operatorname{Re}(\lambda_i) + h^2 (\operatorname{Re}^2(\lambda_i) + \operatorname{Im}^2(\lambda_i)) + O(h^2) \quad (4.2.50)$$

Since we assume that $\operatorname{Re}(\lambda_i) < 0$, there exists a positive number δ such that for all h in $(0, \delta)$

$$|\xi_i|^2 \leq 1 \text{ for all } h \text{ in } [0, \delta) \quad (4.2.51)$$

or

$$\sigma(M(h)) \text{ in } B(0, 1) \quad (4.2.52)$$

Therefore, this algorithm is stable.

Corollary 4.2

The IIE algorithm is convergent.

Proof: The proof follows directly from Theorem 4.1, 4.2, and 3.4.

As mentioned in Section 3.4, none of the one-sweep relaxation timing analysis algorithms described previously are consistent when applied to circuits containing floating capacitors. This is the major drawback for these one-sweep relaxation timing algorithms. From the theorems presented in this section, we can conclude that the IIE method has nicer numerical properties. It eliminates the major obstacle for using one-sweep relaxation timing analysis algorithm.

The importance of stability for a one-sweep relaxation method has been discussed in Section 3.4. Here, we are not only interested in whether the IIE method is stable or not, but also its region

of absolute stability. Let us examine the companion matrix given in Eqn. (4.2.22):

$$M(h) \equiv \begin{bmatrix} 0 & \mathbf{I} \\ (\mathbf{C}_I + h \mathbf{G}_I)^{-1} \mathbf{C}_u & (\mathbf{C}_I + h \mathbf{G}_I)^{-1} (\mathbf{C}_I - \mathbf{C}_u - h \mathbf{G}_u) \end{bmatrix}$$

Consider the new splitting:

- (1) $\mathbf{G} = \mathbf{G}_L + \mathbf{G}_D + \mathbf{G}_U$, where \mathbf{G}_L is strictly lower triangular, \mathbf{G}_D is a diagonal matrix and \mathbf{G}_U is strictly upper triangular.
- (2) $\mathbf{C} = \mathbf{C}_L + \mathbf{C}_D + \mathbf{C}_U$, where \mathbf{C}_L is strictly lower triangular, \mathbf{C}_D is a diagonal matrix and \mathbf{C}_U is strictly upper triangular.

We can rewrite the companion matrix as:

$$M(h) = \tag{4.2.53}$$

$$\begin{bmatrix} 0 & \mathbf{I} \\ [\mathbf{C}_D + \mathbf{C}_L + h \mathbf{G}_L + h \mathbf{G}_D]^{-1} \mathbf{C}_U & [\mathbf{C}_D + \mathbf{C}_L + h \mathbf{G}_L + h \mathbf{G}_D]^{-1} [\mathbf{C}_D + \mathbf{C}_L - \mathbf{C}_U - h \mathbf{G}_U] \end{bmatrix}$$

Define:

$$\mathbf{y} = M(h) \mathbf{x} \tag{4.2.54}$$

$$\|M(h)\| = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{y}\|_\infty}{\|\mathbf{x}\|_\infty} \tag{4.2.55}$$

or

$$\begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = M(h) \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} \tag{4.2.56}$$

or

$$y^1 = x^2 \tag{4.2.57}$$

$$[\mathbf{C}_L + h \mathbf{G}_L] y^2 = -[\mathbf{C}_L + h \mathbf{G}_L] y^2 + \mathbf{C}_U x^1 + [\mathbf{C}_L + \mathbf{C}_D - \mathbf{C}_U - h \mathbf{G}_U] x^2 \tag{4.2.58}$$

Here, we would like to assume that $\|\mathbf{y}\|_\infty = \max_i |y^i| \in y^2$. This is because that the test circuit is asymptotically stable, and as long as we stay in the region of stability, $\|y^2\| \geq \|y^1\|$ since y^1 is further ahead in time. Therefore, we can concentrate on Eqn. (5.2.58) and rewrite it as:

$$y^2 = -[\mathbf{C}_L + h \mathbf{G}_L]^{-1} [\mathbf{C}_L + h \mathbf{G}_L] y^2$$

$$+ [C_L + hG_L]^{-1} C_U x^1 + [C_L + hG_L]^{-1} [C_L + C_D - C_U - hG_U] x^2 \quad (4.2.59)$$

Let $2, k$ be the index such that $\|y\|_\infty = |y^{2,k}|$. Then from the k^{th} equation of Eqn. (4.2.59), we have:

$$\|y\|_\infty \leq s_k \|y\|_\infty + r_k \|x\|_\infty + t_k \|x\|_\infty \quad (4.2.60)$$

where

$$s_i = \sum_{j=1}^{i-1} \left| \frac{C_{ij} + hG_{ij}}{C_{ii} + hG_{ii}} \right| \quad i = 1, \dots, n$$

$$r_i = \sum_{j=i+1}^n \left| \frac{C_{ij}}{C_{ii} + hG_{ii}} \right| \quad i = 1, \dots, n$$

$$t_i = \sum_{j=1}^i \left| \frac{C_{ij}}{C_{ii} + hG_{ii}} \right| + \sum_{j=i+1}^n \left| \frac{C_{ij} + hG_{ij}}{C_{ii} + hG_{ii}} \right| \quad i = 1, \dots, n$$

Therefore,

$$\|M(h)\|_\infty = \max_{x \neq 0} \frac{\|y\|_\infty}{\|x\|_\infty} \quad (4.2.61)$$

$$\leq \frac{r_k + t_k}{1 - s_k}$$

$$\leq \max_{1 \leq i \leq n} \frac{r_i + t_i}{1 - s_i}$$

$$= \max_{1 \leq i \leq n} \frac{\sum_{j=i+1}^n \left| \frac{C_{ij}}{C_{ii} + hG_{ii}} \right| + \sum_{j=1}^i \left| \frac{C_{ij}}{C_{ii} + hG_{ii}} \right| + \sum_{j=i+1}^n \left| \frac{C_{ij} + hG_{ij}}{C_{ii} + hG_{ii}} \right|}{1 - \sum_{j=1}^{i-1} \left| \frac{C_{ij} + hG_{ij}}{C_{ii} + hG_{ii}} \right|} \quad (4.2.62)$$

The region of stability for the IIE method is the set of h that satisfies $\|M(h)\|_\infty < 1$. Thus, a sufficient condition is to have h such that the right hand side of Eqn. (4.2.62) be less than one. It may be hard to evaluate this equation in practice; however, looking at Eqn. (4.2.62), we can see that we need a stronger condition than having both C and G being strictly diagonally dominant to ensure that the IIE method be A-stable. In fact, the off-diagonal terms have to be small enough for the IIE method to stay in the region of stability.

4.3. Implementation of IIE as an Initial Condition Generator

The consistency and stability properties of the formalized IIE algorithm have been established for a class of test circuits. To show the usefulness of this method in the analysis of VLSI MOS circuits, it is important to test these properties on actual MOS circuits used in digital design. Since RELAX uses constant valued waveforms as its initial guess waveforms, we decided to implement an initial guess voltage generator for RELAX using the IIE method to see if any speed improvement could be obtained. In this section, we will describe the implementation of the IIE method in RELAX. The implementation results will also be presented. We start with a brief review of the structure of the RELAX program. Then, a detailed description of how the IIE method is implemented as an initial waveform generator is given. Finally, some implementation results are presented.

4.3.1. Implementation

The first step in RELAX is to read in a circuit description file containing circuit topology, device model parameters, analysis specification, and plotting requests. Before applying the waveform relaxation algorithm described in the previous chapter, RELAX decomposes the circuit into a collection of subcircuits. This is done by partitioning the circuit into clusters of tightly coupled nodes. Since RELAX uses the Gauss-Seidel relaxation iteration, there is a scheduling routine in RELAX which arranges the subcircuits in the natural directionality of the circuit as much as possible.

After the large circuit has been divided into ordered subcircuits, RELAX begins its waveform relaxation process. Instead of performing the relaxation iteration by computing the transient behavior of each subcircuit for the entire user defined simulation interval, RELAX divides the whole simulation interval to many smaller simulation time-slots referred to as a *windows*. RELAX performs waveform relaxation on one window until convergence, then move on to the next window. This method has been proven to increase the efficiency of the waveform relaxation method

especially when applied to digital circuits with logical feedback. In fact, this windowing scheme can improve most of the integration method because it limits how far the local truncation error can propagate in time. An initial guess waveform is chosen for each of the node voltage waveforms (in this case a constant waveform for each node). Within each window, the numerical solution for each of the subcircuit is computed in the order determined by the scheduling routine. The computation is performed using trapezoidal integration algorithm, with the time step controlled by monitoring local truncation error. In the subcircuit, those nodes that are connected to the nodes in other subcircuits are treated as external time-varying voltage sources. In each G-S iteration, the newly computed waveform replaces the waveform generated from the previous iteration. This iteration process is carried all the way to convergence. Then, we move on to the next scheduled subcircuit. This continues until all subcircuits are analyzed for this window.

Let us focus on the circuit equation used in RELAX. To ensure charge conservation, the decomposed differential equations generated by the waveform relaxation algorithm use charge as the state variable. That is, the multistep integration algorithm is applied to

$$\dot{\mathbf{q}}(\mathbf{v}(t)) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) \quad (4.3.1)$$

RELAX uses trapezoidal method and only need to save one copy of the variable vector. However, the IIE method uses voltage as variable, and needs to save two set of vectors at two different time points. In order to minimize the additional storage and implementation effort needed to implement the IIE method in RELAX, we define a new set of charge definition to overcome this problem. Since the IIE method uses backward Euler method to discretize the time derivative, Eqn. (4.3.1) becomes:

$$\frac{\mathbf{q}_{k+1}}{h} - \frac{\mathbf{q}_k}{h} = \mathbf{f}(\mathbf{q}_{k+1}, \mathbf{u}_{k+1}) \quad (4.3.2)$$

$$\text{Define:} \quad \hat{\mathbf{q}}_{k+1} \equiv \mathbf{C}_l(\mathbf{v})\mathbf{v}_{k+1} + \mathbf{C}_u(\mathbf{v})\mathbf{v}_k \quad (4.3.3)$$

$$\hat{\mathbf{q}}_k \equiv \mathbf{C}_l(\mathbf{v})\mathbf{v}_k + \mathbf{C}_u(\mathbf{v})\mathbf{v}_{k-1} \quad (4.3.4)$$

Substituting this definition of charge into Eqn. (4.3.2), we have:

$$\frac{C_l(v)v_{k+1} + C_u(v)v_k}{h} - \frac{C_l(v)v_k + C_u(v)v_{k-1}}{h} = f(\hat{q}_{k+1}(v_{k+1}, v_k), u_{k+1}) \quad (4.3.5)$$

Then applying one sweep of the Gauss-Seidel method to this equation, we can obtain the same equation as Eqn. (4.1.8) in the IIE method. For example, consider the circuit given in Fig. 4.2.

The first step is to write the KCL equation using charge as the variable:

$$\dot{q}_1 + \dot{q}_3 = -G_1 v_1$$

$$\dot{q}_2 + \dot{q}_3 = -G_2 v_2$$

Then, we apply backward Euler by using the new charge variables defined in this section and apply one sweep of Gauss-Seidel:

$$\hat{q}_{1_{k+1}} - \hat{q}_{1_k} + \hat{q}_{3_{k+1}} - \hat{q}_{3_k} = -hG_1 v_{1_{k+1}}$$

$$\hat{q}_{2_{k+1}} - \hat{q}_{2_k} + \hat{q}_{3_{k+1}} - \hat{q}_{3_k} = -hG_2 v_{2_{k+1}}$$

Substituting Eqn. (4.3.3) and Eqn. (4.3.4), we have:

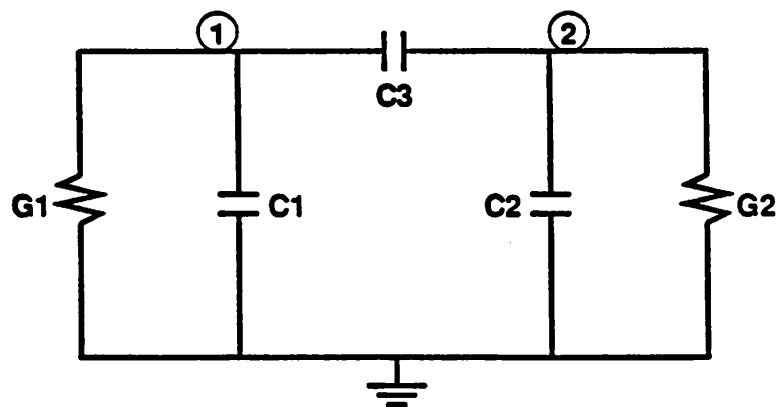


Figure 4.2 Circuit Example

$$C_1 v_{1_{k+1}} - C_1 v_{1_k} + C_3 v_{1_{k+1}} - C_3 v_{2_k} + C_3 v_{1_k} - C_3 v_{2_{k-1}} = -hG_1 v_{1_{k+1}}$$

$$C_2 v_{2_{k+1}} - C_2 v_{2_k} + C_3 v_{2_{k+1}} - C_3 v_{2_{k+1}} + C_3 v_{2_k} - C_3 v_{1_k} = -hG_2 v_{2_{k+1}}$$

or

$$\begin{bmatrix} C_1 + C_3 + hG_1 & 0 \\ -C_3 & C_2 + C_3 + hG_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{k+1} = \begin{bmatrix} C_1 + C_3 & C_3 \\ -C_3 & C_2 + C_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_k + \begin{bmatrix} 0 & -C_3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{k-1} \quad (4.3.6)$$

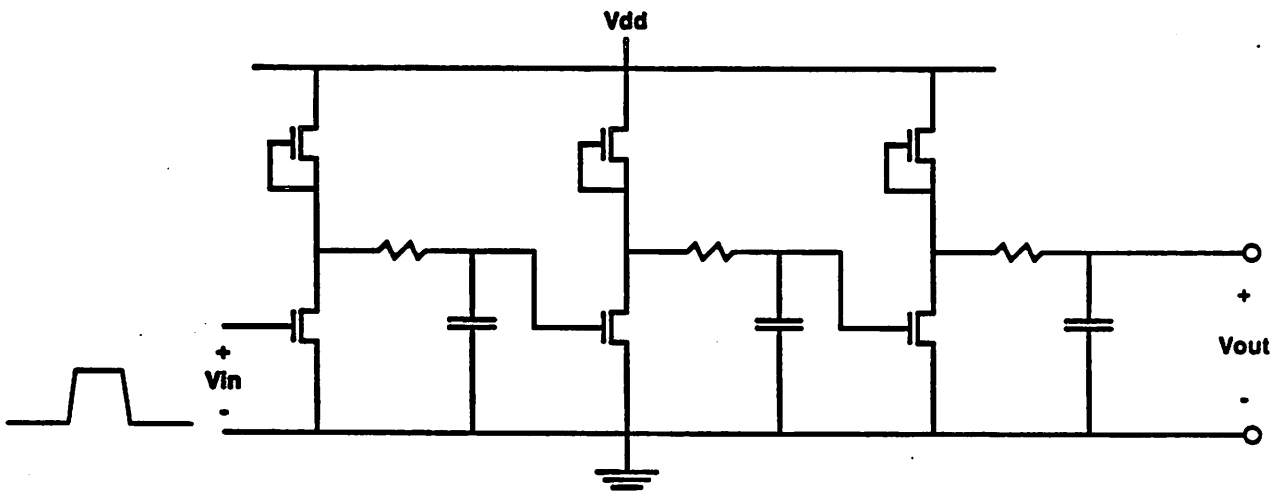
If we apply the IIE method directly to the original problem using node voltages as the variables, we will get the same equation as Eqn. (4.3.6).

Other than the modification stated above, a time-step control scheme similar to the one used in RELAX is adopted for the IIE method. In addition, the technique that exploits the latency in RELAX is also used.

4.3.2. Implementation Results

The circuit in Fig. 4.3(a) is a three stage inverter chain. Fig. 4.3(b) shows the difference between the generated initial waveform and the final solution. Note that the error between the two waveforms is very small. The circuit in Fig. 4.4(a) is an enhancement-load NMOS bootstrapped inverter. A floating capacitor, *bootstrap*, forces the load transistor to turn hard on when the input voltage is rising; thus, improving the speed of the circuit. Again, Fig. 4.4(b) shows that the difference between the initial guess waveform and the exact solution is almost negligible. Similar results are obtained for the circuits given in Fig. 4.5(a), a shift register, and Fig. 4.6(a), an inverter with delay, with their simulated waveforms given in Fig. 4.5(b) and Fig. 4.6(b) respectively.

In the above test circuits, we have focused on tightly coupled circuits, especially those circuits with floating capacitors. The results confirm that the IIE method itself is fairly accurate as predicted in the previous section. In fact, except for the inverter with delay, it is quite difficult to distinguish between the initial guess waveform and the final solution. The shift register is considered to be the most tightly coupled circuits among these test circuits. It forces the IIE method to



Inverter Chain

Figure 4.3a Three Stage Inverter Chain.

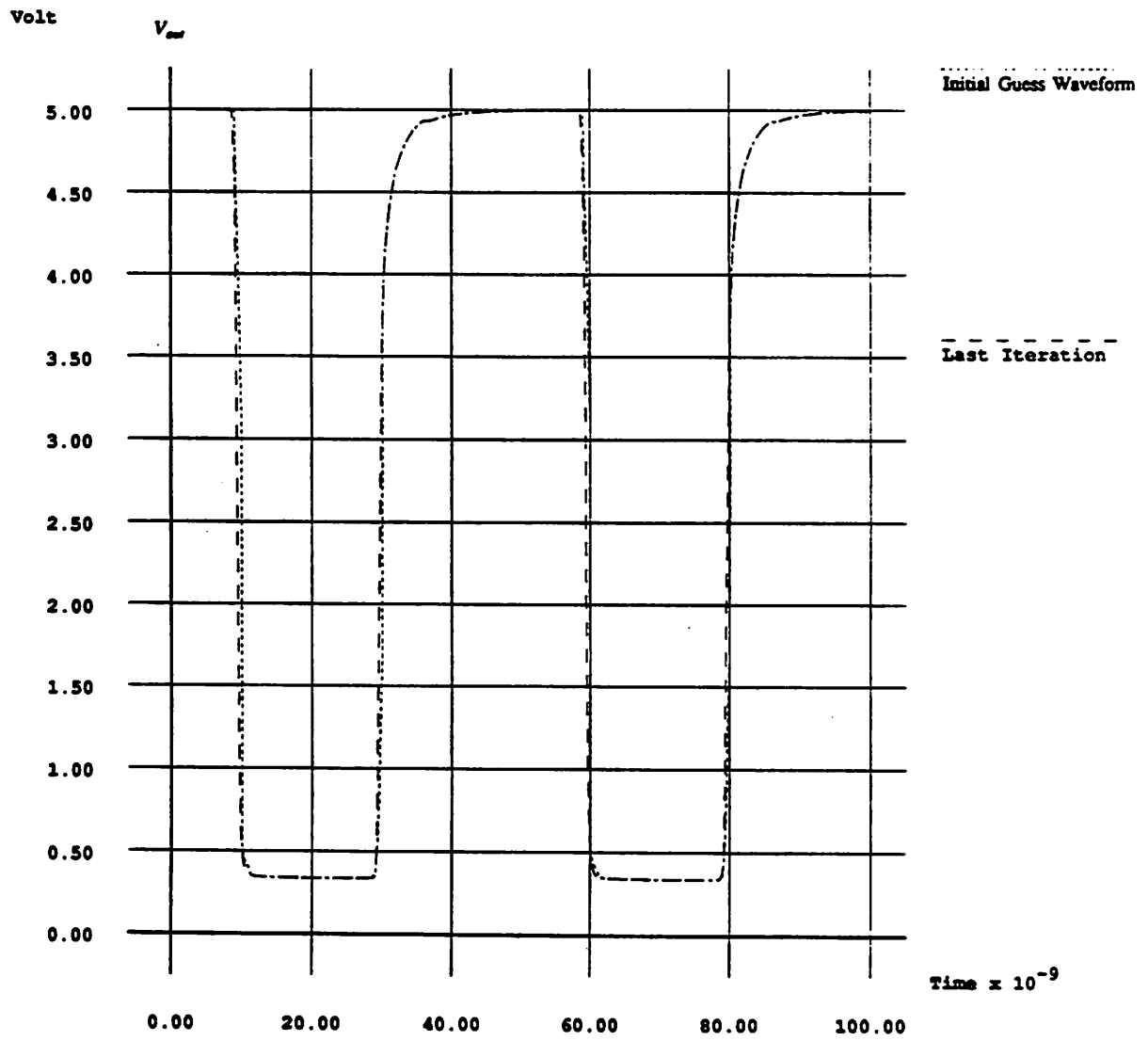


Figure 4.3b Initial Guess Waveform and the Final Solution for the Three Stage Inverter Chain.

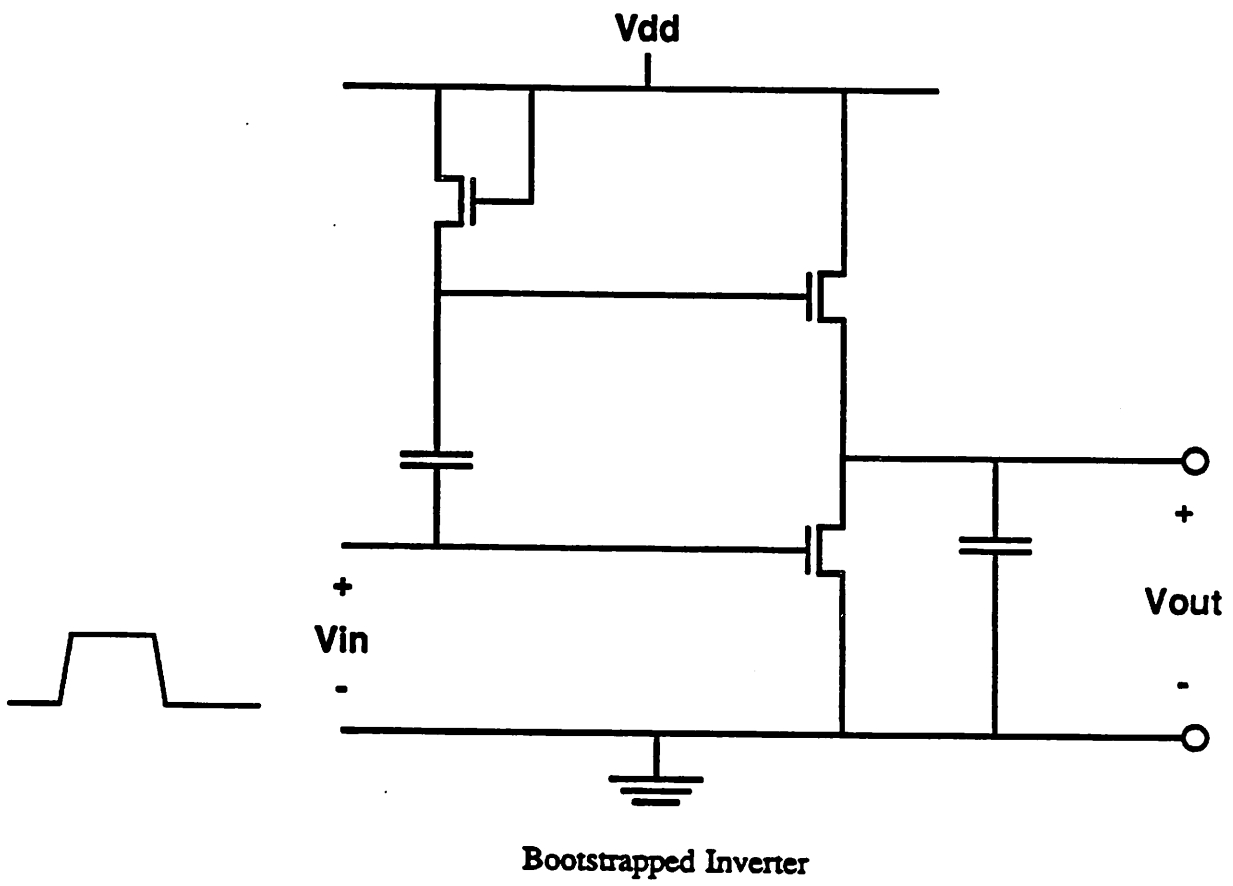


Figure 4.4a Enhancement-Load NMOS Bootstrapped Inverter.

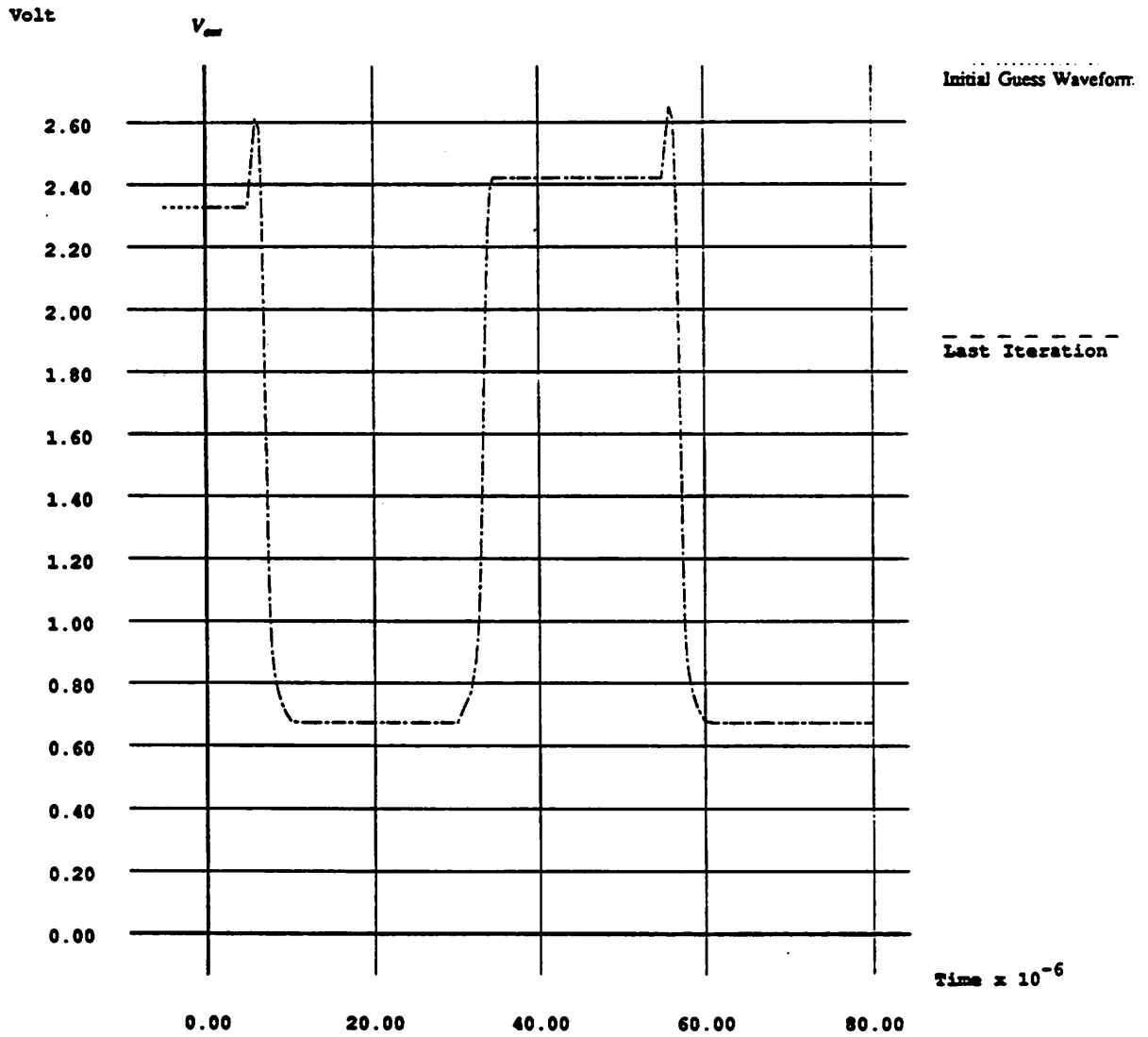


Figure 4.4b Initial Guess Waveform and the Final Solution for the Bootstrapped Inverter.

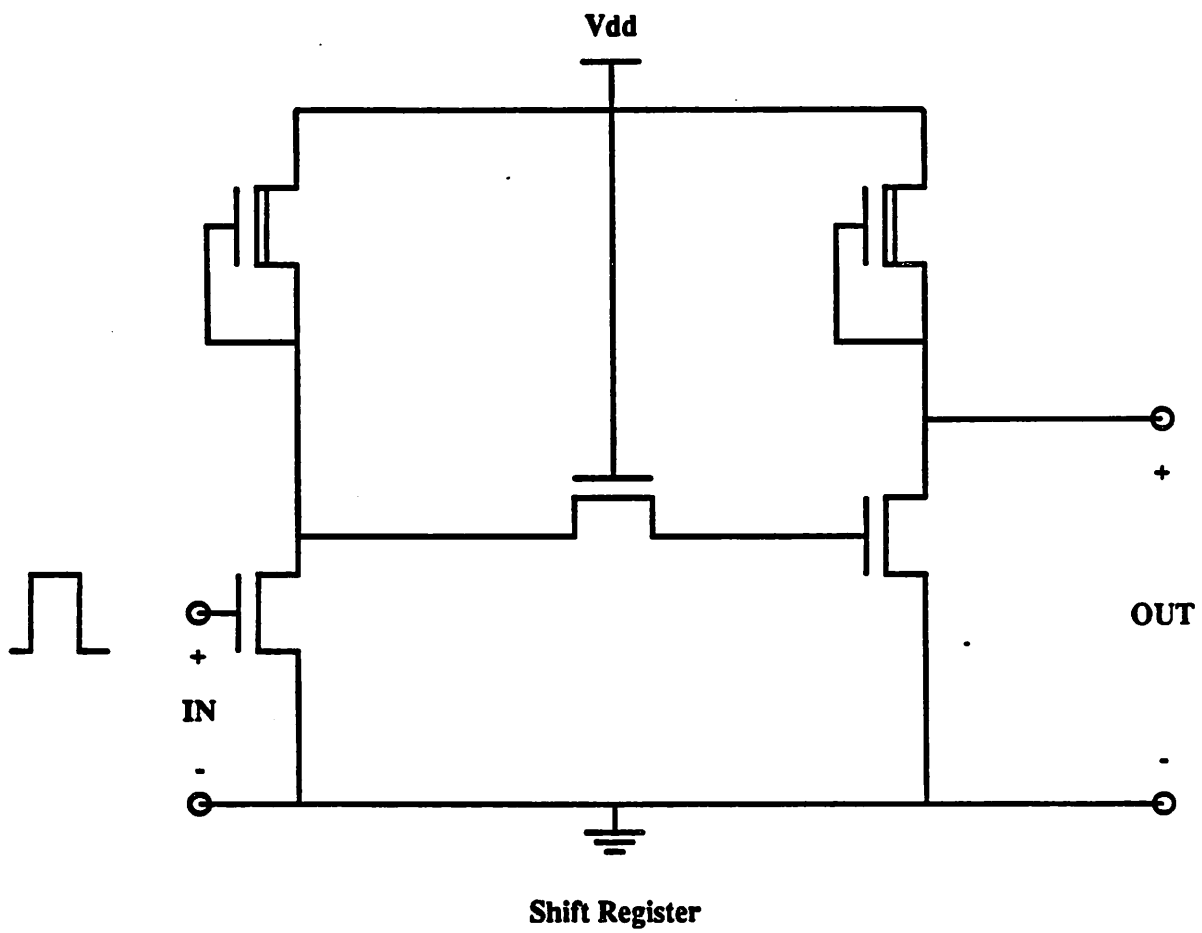


Figure 4.5a Shift Register.

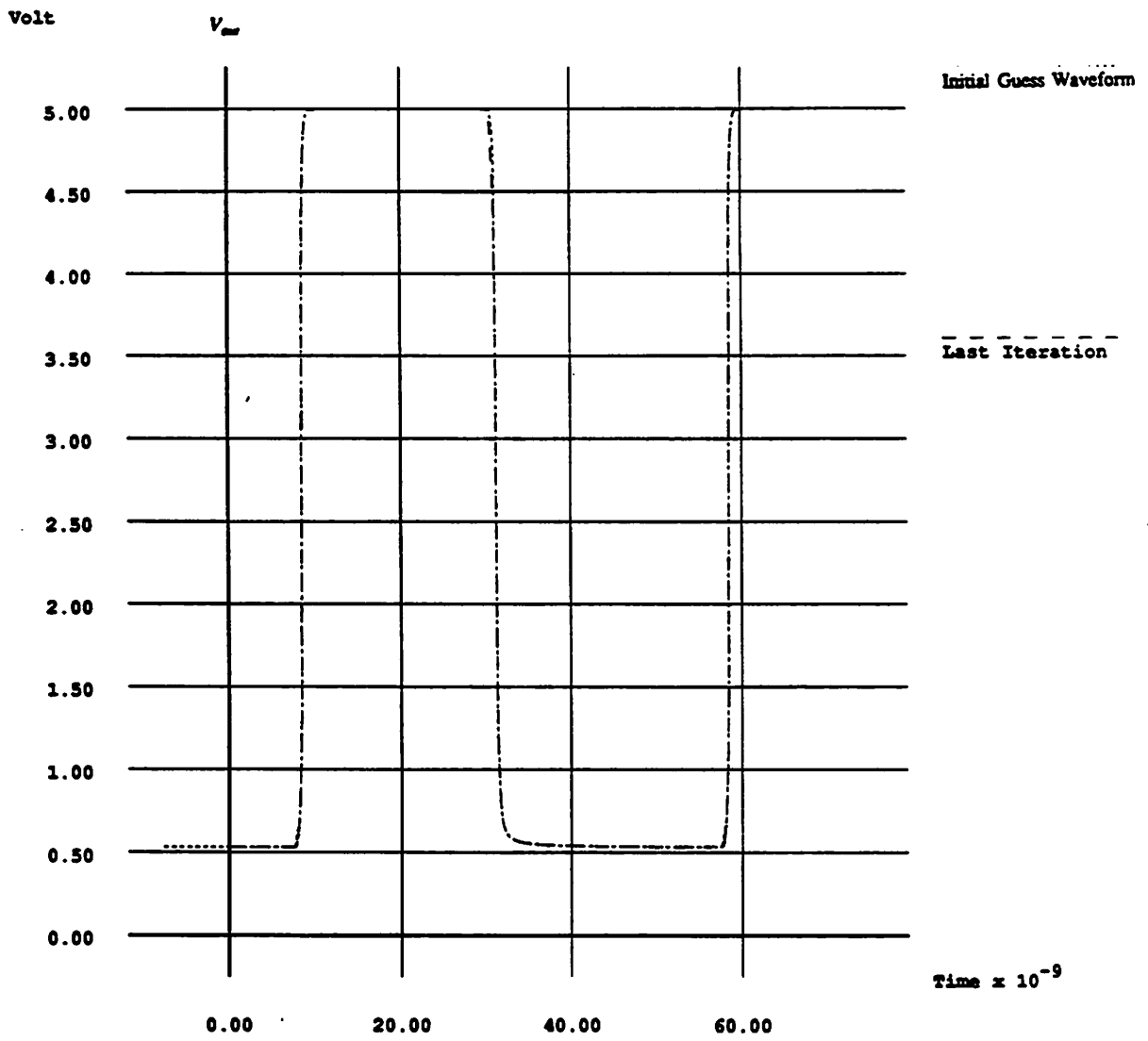
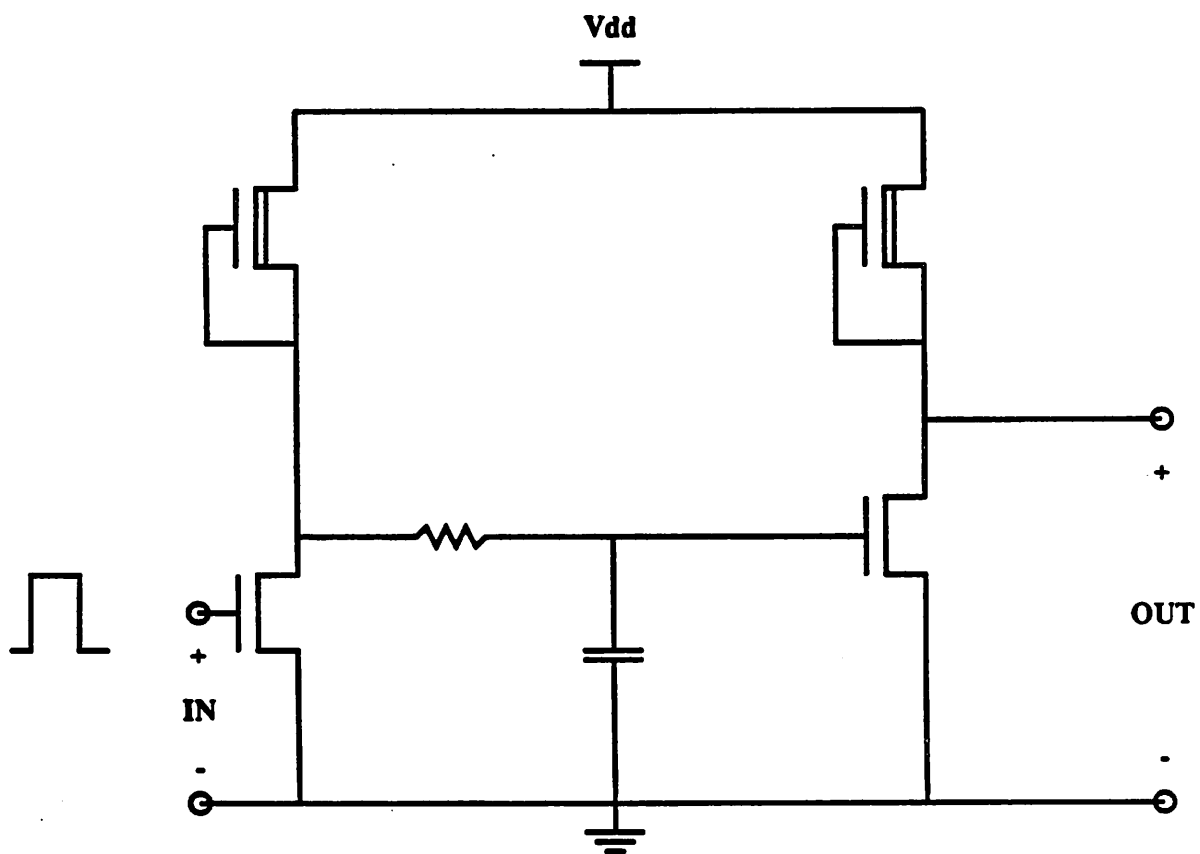


Figure 4.5b Initial Guess Waveform and the Final Solution for the Shift Register.



Inverter with Delay

Figure 4.6a Inverter with Delay.

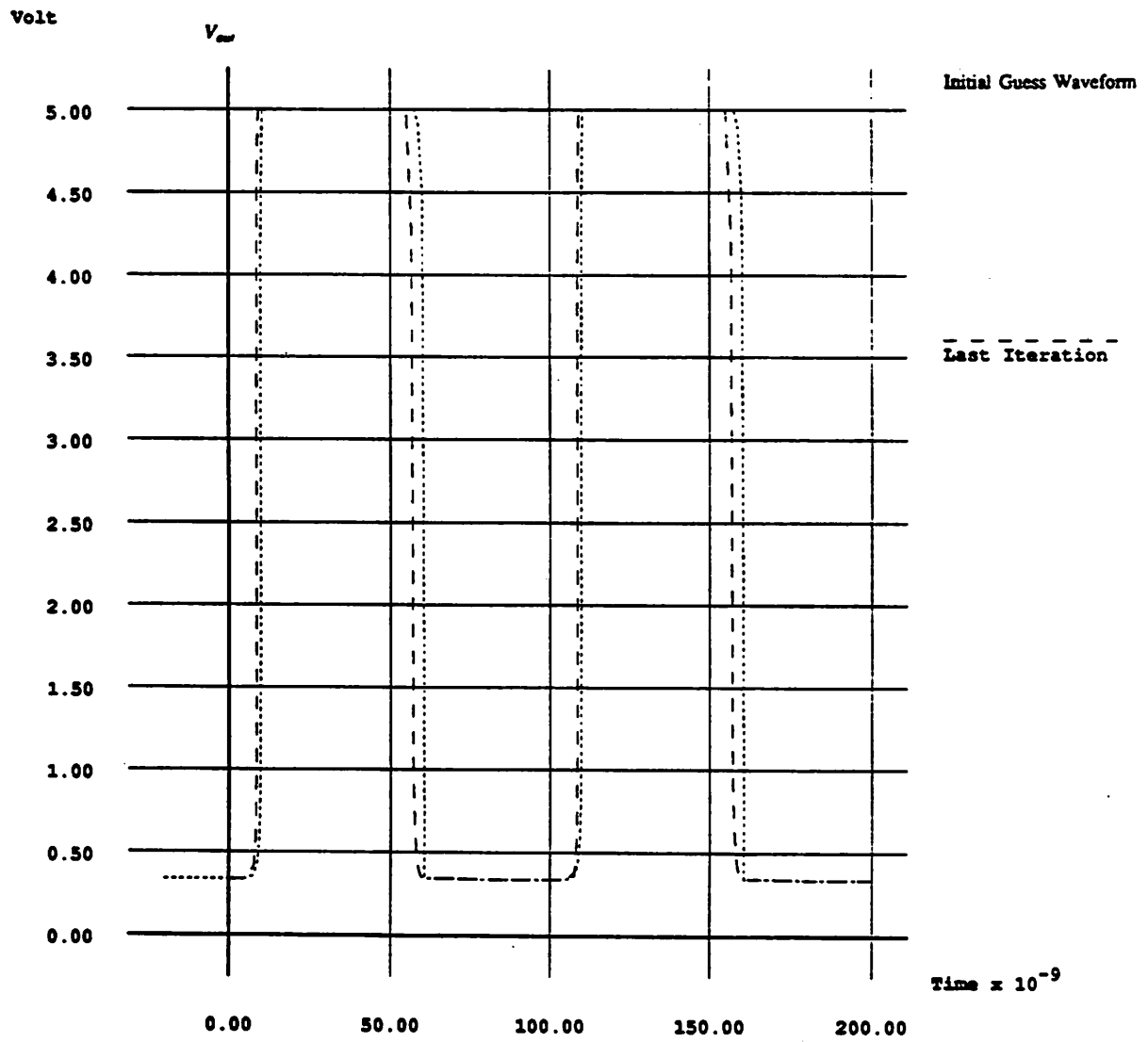


Figure 4.6b Initial Guess Waveform and the Final Solution for the Inverter with Delay.

use a very small time step hence increases the total CPU time. Although floating capacitors no longer pose a problem to the simulation if the time step is chosen to be small enough, it tends to decrease the overall efficiency of the program. The simulation run-time requirements of these circuits are shown in Table 4.1.

Circuit	with IIE method	without IIE method
3 stages inverter chain	2.94s	2.84s
bootstrapped inverter	0.35s	0.33s
shift register	3.76s	0.77s
inverter with delay	1.25s	1.27s

Table 4.1 Comparison of RELAX with and without IIE initial guess waveform generator

The run-time comparisons of some bigger digital circuits are given in Table 4.2. It indicates that RELAX with the initial guess waveform generator usually requires about the same or more CPU time for most digital circuits compared with the original RELAX program. One obvious reason is that there is more overhead cost in generating the initial guess waveform. However, the major reason is that the time step required for the IIE method is smaller than one used by the trapezoidal method used in RELAX.

Circuit	Devices	Nodes	with IIE method	without IIE method
cmos inverter chain	767	259	18.32s	18.07s
cmos inverter chain	3071	1027	74.51s	72.81s
input decoder	2607	282	81.52s	75.09s
input decoder	253	49	7.24s	7.32s
domino cmos circuits	8	12	1.62s	1.63s
nmos depletion-load integrator	64	13	1.05s	1.17s

Table 4.2 Comparison of RELAX with and without IIE initial guess waveform generator

As mentioned before, we are approximating the derivatives of the voltage at current time point by the voltage values of previous and current time points. Since RELAX computes all node voltages of a subcircuit at the same time and carries the iteration to convergence at each time point, it approximates the derivatives using the voltage value at previous time point t_x , and the newly cal-

culated voltage value at current time point t_{k+1} . The IIE method on the other hand, uses one Gauss Seidel iteration, approximates the derivatives using the voltage values at either t_{k-1} and t_k , or t_k and the newly calculated voltage value at current time point t_{k+1} depending on whether the voltage at t_{k+1} has been computed or not. Intuitively, in order for the approximations of the derivatives computed using voltage values at t_{k-1} and t_k to have the same accuracy as the ones computed using voltage values at t_k and t_{k+1} , the time step of the former (IIE) method will most likely need to be twice smaller than that of the latter method (RELAX). Fig. 4.7 shows a clearer picture why the IIE method probably needs to use twice as small a time step to achieve the same accuracy.

We can also look at this problem from a different view point. Consider the following linear time-invariant circuit:

$$C\dot{v} = -Gv$$

At any time t_{k+1} , this circuit satisfies the KCL equation:

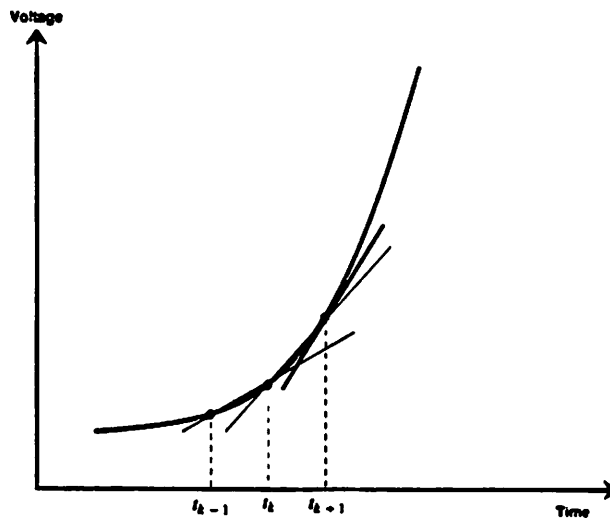


Figure 4.7

$$C\dot{\mathbf{v}}(t_{k+1}) = -G\mathbf{v}(t_{k+1}) \quad (4.3.8)$$

Let $\dot{\mathbf{v}}$ be the computed solution of trapezoidal method used in RELAX. At time t_{k+1} this method satisfies following equation:

$$C\dot{\mathbf{v}}_{k+1} = -G\mathbf{v}_{k+1} \quad (4.3.9)$$

For the IIE method however, the capacitor matrix is split into two triangular matrices and the derivative operator is discretized at two different time points to yield the following equation:

$$C_l(\mathbf{v}_{k+1})\dot{\mathbf{v}}_{k+1} + C_u(\mathbf{v}_{k+1})\dot{\mathbf{v}}_k = -G_l\mathbf{v}_{k+1} - G_u\mathbf{v}_k \quad (4.3.10)$$

where \mathbf{v}_k is the computed solution for the IIE method. Clearly, if the time step is too large, the solution of Eqn. (4.3.10) can be very far away from satisfying the KCL equations without considering the error produced by the implicit integration method. Thus, we can conclude that comparing with the trapezoidal rule, a smaller time step is needed for the IIE method to achieve the same satisfaction of KCL equations.

Another consideration is that we are unable to proof that the IIE method is A-stable. In fact, from our experience, the local truncation error may propagate rapidly in some cases and cause instability if the time-step is chosen to be too big.

This seems to be an inherent problem associated with the IIE method and other timing analysis algorithms. If we choose a stricter time step control scheme, we will spend a lot of time just to generate the initial guess voltage waveform, and hence increase the total CPU time. If we do not, it will take RELAX additional iterations just to correct the error caused by the inaccuracy of the initial guess waveform. Therefore, adding the IIE method as the initial guess waveform generator has failed to improve the overall performance of RELAX.

4.4. Conclusions

In Part I of this thesis, we have discussed various different numerical integration methods for circuit simulation with special attention given to timing analysis algorithms. As pointed out earlier, the major problem with timing analysis algorithms is that they may fail when floating capacitors

are present in the circuits.

We have also formalized the IIE algorithm and investigated its numerical properties. Based on the theoretical finding, the IIE method was proven to be consistent, stable, and convergent for circuits containing floating capacitors. However, to be effective in circuit simulation, it is desirable for an integration method to be A-stable and able to control the time step simply by monitoring the local truncation error. The timing analysis algorithms we introduced in this thesis are shown to be A-stable only on small classes of test circuits. This is one of the major drawbacks of timing analysis algorithms.

Finally, we have described the implementation of the IIE method as an initial guess waveform generator for RELAX. Although it does not improve the overall performance of RELAX, it confirms that the IIE method itself can give stable and accurate solutions for circuits with tightly coupled feedback.

PART II : CIRCUIT SIMULATION IN THE FREQUENCY-DOMAIN

CHAPTER 5

Introduction

The second part of this thesis focus on the simulation techniques which can be used to find periodic steady-state responses for various circuits of interest. Periodic waveforms play an important role in the analysis of many classes of circuits. Examples include the analysis of microwave circuits and circuits in power transmission systems. Periodic steady-state analysis of oscillators or periodically forced nonlinear systems is also an important example.

Many computational methods related to this class of problems have been explored [1,20,21,22] . One classical approach to finding the periodic steady-state response is to integrate the system of equations from some initial conditions until the transient response becomes negligible. Another time domain analysis technique uses the Newton method [20,21]. This technique is very popular because it can be applied to both autonomous (unforced) and nonautonomous (forced) circuits. Let us assume that, for the system under study, there exists a periodic solution $\mathbf{x}(t)$ of period T . The Newton algorithm searches for the steady-state solution by formulating the system as a two-point boundary value problem, using the fact that $\mathbf{x}(0) = \mathbf{x}(T)$ in the steady state. The basic approach is to find the right initial (or boundary) periodic state $\mathbf{x}(0)$ such that when we integrate this system from 0 to T , $\mathbf{x}(T) = \mathbf{x}(0)$.

Time-domain simulation techniques may be very inefficient for a certain class of circuits when the periodic steady-state solutions are of the primary interest. Two examples are high-Q circuits and circuits containing many distributed elements. The quality factor, Q , measures the relative damping in a damped oscillation. The less damping the oscillation has, the larger the factor Q is. For example, a lossless resonant circuit has zero damping or infinite Q . For lightly damped (or high Q) systems, it takes a long time before the systems reach the steady state. Since traditional

time-domain simulators integrate the system equations until the transient response dies off, the integration could extend over many periods, making the computation very costly and inefficient. Circuits containing distributed elements are difficult and often impractical to simulate in the time domain because the distributed elements are described by partial differential equations. Although distributed components can also be approximated by collections of lumped components, it usually requires a large number of these components to achieve sufficient accuracy.

Simulation in the frequency domain can avoid those problems mentioned above. First of all, it finds periodic steady-state solutions directly without having to wait until the transient responses die off. This greatly improves simulation efficiency for high-Q circuits. As for distributed elements, systems containing those elements can be modeled as algebraic equations in the frequency domain, making it much easier to obtain the solutions.

In Chapter 6, we will start with problem formulation and discuss the Newton method used in time-domain simulators. Then, we will review the method of harmonic-balance as a general approach to converting a set of differential equations into a system of nonlinear algebraic equations. These nonlinear algebraic equations can then be solved to obtain the periodic steady-state solutions. In Chapter 7, we will show how we can combine the harmonic-balance method with the Newton-Raphson method to form the *harmonic-Newton* method which has been implemented in *Spectre* [4,23]. So far, *Spectre* is limited to the simulation of nonautonomous systems. In Section 7.3, we will discuss how we can extend the harmonic-Newton method to handle autonomous systems.

The convergence properties of the harmonic-Newton method will be discussed rigorously in Chapter 8. In Chapter 9, we will focus on the computation of the error generated by the harmonic-Newton method and show how this error bound can be used to predict the number of harmonics needed for a given error objective before applying the harmonic-Newton method, thus improving the efficiency of *Spectre* significantly.

In PART II of this thesis, we adopt the following notation; lower-case letters are used to denote time-domain variables and functions, and upper-case letters are used for frequency-domain

variables and functions. Subscripts H represents the number of harmonics contained in the variables or function, superscripts are used for node number, and superscripts in parentheses represent the Newton iteration count. Vectors of variables or functions are in boldface. The superscripts C on upper-case letter variables or functions means the Fourier coefficient of these variables and functions are in complex form. The bar over variables or functions implies that discrete Fourier transform is used.

CHAPTER 6

Overview

In this chapter, we discuss how the circuit equations for steady-state analysis are formulated. The Newton method which has been widely used in the time domain to obtain solutions of periodic systems is then introduced. Then we focus on the harmonic-balance method and discuss how we can use the method to convert a system of differential equations into a system of algebraic equations. In this chapter, we will delay the discussion of the application of those simulation algorithms on some special classes of circuits such as mixers, that have a steady-state response containing almost-periodic signals until later sections.

6.1. Background

In order to understand the problem better, we first start with some background on periodic and almost-periodic functions. To distinguish between periodic and almost-periodic functions, we first give the definition of a translation number. The real number $\tau(\epsilon)$ is called a translation number for $f(x)$ corresponding to ϵ if

$$|f(x + \tau(\epsilon)) - f(x)| \leq \epsilon \quad \text{for } -\infty < x < \infty. \quad (6.1.1)$$

Note that a translation number $\tau(\epsilon)$ of $f(x)$ is also a translation number corresponding to any $\epsilon' > \epsilon$.

To understand this, let us consider the function $s(t) = e^{j\omega_1 t} + e^{j\omega_2 t}$ where j is the imaginary number $\sqrt{-1}$. If there exist two integers n_1 and n_2 distinct from zero such that $n_1 \frac{2\pi}{\omega_1} = n_2 \frac{2\pi}{\omega_2}$, then $s(t)$ is a periodic function with period $n_1 \frac{2\pi}{\omega_1}$. If there is no integer multiple of $\frac{2\pi}{\omega_1}$ which is equal to a integer multiple of $\frac{2\pi}{\omega_2}$, then, for any arbitrarily small number $\delta > 0$, there exists a pair of arbitrarily large integers n_1 and n_2 such that $\left| n_1 \frac{2\pi}{\omega_1} - n_2 \frac{2\pi}{\omega_2} \right| < \delta$. Let τ be any number

between $n_1 \frac{2\pi}{\omega_1}$ and $n_2 \frac{2\pi}{\omega_2}$, and ε be some real number such that $|s(t-\tau) - s(t)| \leq \varepsilon$, then $\tau(\varepsilon)$ is a translation number of $s(t)$ and τ is almost a period for $s(t)$ since the difference between $s(t + \tau)$ and $s(t)$ can be arbitrary small. We will now give the formal definition of almost-periodic functions.

Definition:

A continuous function $f(x)$, $-\infty < x < \infty$ is called almost-periodic if for any $\varepsilon > 0$, there exists a positive real number $L(\varepsilon)$ such that within each interval of length $L(\varepsilon)$ on x there is at least one translation number $\tau(\varepsilon)$.

Obviously, a continuous periodic function of period T_0 is a special case of an almost-periodic function. The period kT_0 for $k = \pm 1, \pm 2, \dots$ are translation numbers $\tau(\varepsilon)$ for all $\varepsilon \geq 0$. The Fourier series of a given periodic function f is defined as an infinite series

$$f(t) = \sum_{k=-\infty}^{\infty} F^C[k\omega_0] e^{j\omega_0 k t} \quad (6.1.2)$$

with coefficients satisfying

$$F^C[k\omega_0] = \frac{1}{T_0} \int_0^{T_0} f(t) e^{-j\omega_0 k t} dt \quad (6.1.3)$$

where $T_0 > 0$ is the period of the function f and $\omega_0 = 2\pi/T_0$ is the fundamental frequency of the function.

If f is not periodic, it has been shown in [24] that the class of almost-periodic functions is identical to the closure of the class of all finite sums of trigonometric functions. This means that every almost-periodic function can be approximated uniformly for $-\infty < t < \infty$ by a sum of a countable number of trigonometric functions:

$$f(t) = \sum_{k=-\infty}^{\infty} F^C[\omega_k] e^{j\omega_k t} \quad (6.1.4)$$

where the coefficients $F^C[\omega_k]$ is any complex number and ω_k can be any real quantities. Here, we limit ourselves to the discussion of series having at most denumerably many elements, i.e. functions whose spectrum is not continuous but consists of discrete lines. Functions having continuous

spectrum are no longer periodic in the time domain and therefore are not in the scope of the thesis.

Moreover, since we are only concerned with real functions, the following conditions are satisfied:

(1) The imaginary part of $F^C[0]$ is equal to zero, and (2) $F^C[-k\omega] = (F^C[k\omega])^*$.

We use $AP(\Lambda)$ to denote the set of all almost-periodic waveforms over the set of frequencies

$\Lambda = \{0, \omega_1, \omega_2, \dots\}$. Then, all almost-periodic waveforms can be written as

$$f(t) = \sum_{\omega_k \in \Lambda} F^C[\omega_k] e^{j\omega_k t} \quad (6.1.5)$$

If there is a set of d independent frequencies $\{\lambda_1, \lambda_2, \dots, \lambda_d\}$ such that

$$\Lambda = \left\{ \omega \mid \omega = k_1\lambda_1 + k_2\lambda_2 + \dots + k_d\lambda_d : k_1, k_2, \dots, k_d \in \mathbb{Z} \right\} \quad (6.1.6)$$

then Λ is said to be a module of dimension d and the frequencies $\{\lambda_1, \lambda_2, \dots, \lambda_d\}$ are referred to as the fundamental frequencies and they form a basis for Λ .

To solve an equation in the frequency domain, it is necessary to truncate the number of harmonics, H , considered. Let $f(t)$ be a continuous periodic vector function of period $2\pi/\omega_o$, then the trigonometric polynomial

$$f_H(t) = \sum_{k=-H}^{H-1} F^C[k\omega_o] e^{jk\omega_o t} \quad (6.1.7)$$

is a truncated trigonometric polynomial of the given periodic function $f(t)$. Let us denote such a truncation operator as P_H . Then, the truncated polynomial $f_H(t)$ of a periodic function $f(t)$ is given as follows:

$$f_H(t) = P_H f(t). \quad (6.1.8)$$

Some further description is needed in order to extend the definition of truncation to almost-periodic functions. Let $f(t)$ be a continuous almost-periodic vector function with fundamental frequencies $\{\lambda_1, \dots, \lambda_d\}$. Following are two different ways of truncating the set of frequencies in almost-periodic circuits:

(1) Those frequencies with indices k_j whose absolute values are greater than H are truncated.

Thus, the truncated frequency space Λ_H is:

$$\Lambda_H(\lambda_1, \lambda_2, \dots, \lambda_d) = \left\{ \omega \mid \omega = k_1\lambda_1 + k_2\lambda_2 + \dots + k_d\lambda_d: \right. \\ \left. k_j \in \mathbf{Z}; |k_j| \leq H \text{ for } 1 \leq j \leq d; k_1 \geq 0 \right\}$$

(2) The absolute sum of all the indices k_j is confined to be less than or equal to H . The resulting truncated frequency space Λ_H is:

$$\Lambda_H(\lambda_1, \lambda_2, \dots, \lambda_d) = \left\{ \omega \mid \omega = k_1\lambda_1 + k_2\lambda_2 + \dots + k_d\lambda_d: \right. \\ \left. k_j \in \mathbf{Z}; \sum_{j=1}^d |k_j| \leq H \text{ for } 1 \leq j \leq d; k_1 \geq 0 \right\}$$

Then the truncated trigonometric polynomial is in the form:

$$f_H(t) = \sum_{\omega_k \in \Lambda_H} F^C[\omega_k] e^{j\omega_k t} \quad (6.1.9)$$

where $F^C[\omega_k] \in C^n$ is a vector of Fourier coefficient corresponding to each $\omega_k \in \Lambda_H = \{0, \omega_1, \dots, \omega_K\}$, and $K = \frac{1}{2}((2H+1)^d + 1)$ for the first definition and $K = \frac{2^{d-1} H^d}{d!}$

if the second definition is used. Let us denote the operator which truncates an almost-period function $f(t)$ to order H as P'_H . Thus, the truncated polynomial $f_H(t)$ of an almost-period function $f(t)$ is given by:

$$f_H(t) = P'_H f(t) \quad (6.1.10)$$

6.2. Problem Formulation

For the analysis of the second half of the thesis, we assume that the circuit under study satisfies the following two conditions:

- (1) The circuit is asymptotically stable and, if the system is nonautonomous, it must have a steady-state solution for the given excitation.
- (2) All nonlinear devices must be lumped and their constitutive relationship must be algebraic, and continuously differentiable with respect to the voltage.

The condition that the constitutive relationship of the nonlinear devices must be described by algebraic equations assures us that the response waveform is periodic when a periodic input is placed on the circuit. When applying the harmonic-balance method, we need to transform the stimulus of each nonlinear device into a time-domain waveform, calculate the resulting response waveform, and then transform the response back into the frequency-domain. If the nonlinear devices can not be described by algebraic equations (i.e. the devices have memory), then the response waveform will not be periodic and can not be accurately transformed back and forth between the time domain and the frequency domain.

In the time domain, a nonautonomous circuit can be modeled as a system of n nonlinear differential equations, here written in a compact form as:

$$g(\mathbf{v}, t) = -\mathbf{u}_s(t)$$

$$\text{or } \mathbf{f}(\mathbf{v}, t) = g(\mathbf{v}, t) + \mathbf{u}_s(t) = \mathbf{i}(\mathbf{v}(t)) + \dot{\mathbf{q}}(\mathbf{v}(t)) + \int_0^t \mathbf{y}(t-\tau)\mathbf{v}(\tau)d\tau + \mathbf{u}_s(t) = 0 \quad (6.2.1)$$

where $\mathbf{i}, \mathbf{q} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are differentiable functions representing respectively the sum of the currents exiting the nodes due to the nonlinear conductors and the sum of the charge exiting the nodes due to the capacitors, \mathbf{y} is the matrix-valued impulse response of the circuit with capacitors and nonlinear devices removed, $\mathbf{v} : \mathbb{R} \rightarrow \mathbb{R}^n$ is the vector of unknown node voltage waveforms, $\mathbf{u}_s : \mathbb{R} \rightarrow \mathbb{R}^n$ is the vector of source current waveforms, and \mathbf{f} is the function that maps the node voltage waveforms into the sum of the currents exiting each node. Using the impulse response to represent the linear portion of the circuit, we can include linear distributed devices in the circuit.

A system is said to be nonautonomous if the time variable t is present explicitly in the function $\mathbf{f}(\mathbf{v}, t)$. For periodic nonautonomous systems, we assume throughout this thesis that all inputs are periodic with period T_o . Since nonlinear devices are assumed to be algebraic, the circuit equations are also periodic with period T_o :

$$\mathbf{f}(\mathbf{v}, t) = \mathbf{f}(\mathbf{v}, t+T_o) \quad (6.2.2)$$

For almost periodic nonautonomous systems such as mixers, which have two or more inputs with arbitrary frequencies and amplitudes, the signal waveforms are made up of several periodic

waveforms which are not harmonically related.

A circuit is said to be autonomous if the function f is independent of t . This implies that all elements have to be time invariant and all independent sources are constant valued. An oscillator is an autonomous system because it can oscillate by itself without any external input. The system of equations of a nonlinear oscillator assume the form:

$$f(\mathbf{v}, t) = g(\mathbf{v}, t) = 0 \quad (6.2.3)$$

However, for the methods to be discussed in this chapter, different forms are used to describe autonomous and nonautonomous systems. Specifically, an n -node periodic nonautonomous system is described the following state equation:

$$\dot{\mathbf{x}} = \mathbf{s}(\mathbf{x}, t) \quad (6.2.4)$$

where \mathbf{x} is the vector of the state variable, both \mathbf{x} and \mathbf{s} are periodic n -tuple vectors with period T_o , and \mathbf{s} has a continuous first partial derivative with respect to \mathbf{x} for all \mathbf{x} and all t . Since the system (including all inputs) are periodic with period T_o , we have:

$$\mathbf{s}(\mathbf{x}, t) = \mathbf{s}(\mathbf{x}, t+T_o) \quad (6.2.5)$$

Similarly, for autonomous systems, the state equation of a nonlinear oscillator assumes the form:

$$\dot{\mathbf{x}} = \mathbf{s}(\mathbf{x}) \quad (6.2.6)$$

6.3. The Newton Method in the Time Domain

In this section, we first discuss how the Newton method can be applied to periodic nonautonomous systems in the time domain. Then we show how this method can be extended to periodic autonomous systems.

6.3.1. Periodic Nonautonomous Systems

Consider the system of equations described in Eqn. (6.2.4)

$$\dot{\mathbf{x}} = \mathbf{s}(\mathbf{x}, t) \quad (6.3.1)$$

We will assume that this equation has a steady-state periodic solution $\mathbf{x}(t)$ of period T_o with initial

state $\mathbf{x}(0)$. Integrating both side of Eqn.(6.3.1) from time 0 to t , we obtain

$$\mathbf{x}(t) = \int_0^t \mathbf{s}(\mathbf{x}(\tau), \tau) d\tau + \mathbf{x}(0) \equiv \mathbf{x}(t, \mathbf{x}(0)) \quad (6.3.2)$$

Since the system has a steady-state solution with period T_o , in steady state, we have

$$\mathbf{x}(0) = \mathbf{x}(T_o) \quad (6.3.3)$$

Therefore, we can treat the problem described by Eqn.(6.3.1) as a two-point boundary value problem. From Eqn. (6.3.2) and (6.3.3), we have

$$\mathbf{x}(T_o) = \int_0^{T_o} \mathbf{s}(\mathbf{x}(\tau), \tau) d\tau + \mathbf{x}(0) \quad (6.3.4)$$

Our goal is to find $\mathbf{x}(0)$ such that at $t = T_o$, $\mathbf{x}(T_o, \mathbf{x}(0)) = \mathbf{x}(0)$. This relationship is seldom satisfied for an arbitrary choice of $\mathbf{x}(0)$. Let us define a new function

$$\mathbf{E}(\mathbf{x}(0)) \equiv \int_0^{T_o} \mathbf{s}(\mathbf{x}(\tau), \tau) d\tau + \mathbf{x}(0) \quad (6.3.5)$$

then, what we want to have is:

$$\mathbf{x}(0) = \mathbf{E}(\mathbf{x}(0)) \quad (6.3.6)$$

Observe that Eqn.(6.3.6) is identical to the standard form for a fixed-point iteration. Hence, if the function $\mathbf{E}(\mathbf{x}(0))$ is a contraction mapping, the solution $\mathbf{x}(0)$ can be found by applying the fixed-point iteration algorithm or the more efficient Newton-Raphson algorithm described in the following:

$$\mathbf{x}^{(j+1)}(0) = \mathbf{x}^{(j)}(0) - [\mathbf{I} - \mathbf{E}'(\mathbf{x}^{(j)}(0))]^{-1} \mathbf{E}(\mathbf{x}^{(j)}(0)) \quad (6.3.7)$$

where

$$\mathbf{E}'(\mathbf{x}^{(j)}(0)) \equiv \left. \frac{\partial \mathbf{E}(\mathbf{x}(0))}{\partial \mathbf{x}(0)} \right|_{\mathbf{x}(0)=\mathbf{x}^{(j)}(0)} \quad (6.3.8)$$

Note that it is necessary to determine the Jacobian matrix $\mathbf{E}'(\mathbf{x}^{(j)}(0))$ before we can evaluate Eqn. (6.3.7), and there exist some efficient numerical techniques for computing the Jacobian matrix. One method is the numerical differentiation technique. Another more efficient method is to evaluate the Jacobian matrix by transient analysis of sensitivity networks [21].

The Newton method converges quadratically to the desired initial state $\mathbf{x}^*(0)$, if the initial guess $\mathbf{x}^{(0)}(0)$ is close to $\mathbf{x}^*(0)$ and $[\mathbf{I} - \mathbf{E}'(\mathbf{x}(0))]$ is nonsingular in the neighborhood of $\mathbf{x}^*(0)$. If the initial guess of the initial state is too far away from $\mathbf{x}^*(0)$, this method may diverge and another initial guess has to be chosen to start the iteration again. If the network has more than one periodic solutions of the same period T_o , it may converge to any one of these multiple solutions, depending on the choice of the initial guess.

6.3.2. Periodic Autonomous Systems

As discussed in the previous section, periodic autonomous systems can be described by the following state equation:

$$\dot{\mathbf{x}} = \mathbf{s}(\mathbf{x}), \quad (6.3.9)$$

Let us assume that Eqn. (6.3.9) has a periodic solution of period T which is generally unknown. This additional unknown T causes the algorithm presented in the previous section to fail since the number of unknowns $n+1$ is greater than the number of constraints n . Under this condition, the system does not have an isolated solution. This is expected since autonomous systems have arbitrary time origins. For example, if some solution $\mathbf{x} = a \cos \omega t$ satisfies the state equations, then a $\cos(\omega t + \theta)$ also satisfies the equations. In fact, even if the oscillation period T is given, the algorithm described in the previous section is still not applicable because the matrix $[\mathbf{I} - \mathbf{E}']$ may become singular at the oscillating frequency.

To salvage this algorithm, let us redefine the two-point boundary value problem as:

$$\mathbf{x}(0) = \mathbf{x}(T) = \int_0^T \mathbf{s}(\mathbf{x}(t)) dt + \mathbf{x}(0). \quad (6.3.10)$$

Again, our goal is to find $\mathbf{x}(0)$ such that at $t = T$, $\mathbf{x}(T, \mathbf{x}(0)) = \mathbf{x}(0)$. Similar to the nonautonomous case, we can rewrite Eqn. (6.3.10) into:

$$\mathbf{x}(0) = \mathbf{E}(T, \mathbf{x}(0)) \equiv \int_0^T \mathbf{s}(\mathbf{x}(t)) dt + \mathbf{x}(0), \quad (6.3.11)$$

Observe that the period T is now added to the argument of the function $\mathbf{E}(T, \mathbf{x}(0))$ so that it can be

determined along with the unknown initial states $\mathbf{x}(t)$. However, the number of unknowns is still greater than the number of equations. In this case, one can either assume an appropriate value for one of these $n+1$ unknowns, or find another independent equation relating these variables such that this system has a unique solution. Recall that autonomous systems have arbitrary time origins, any arbitrary time shift of one periodic solution can be used as the initial state. Hence, we are free to set a value for one of the n initial state variables, $\mathbf{x}^i(0)$, and remove it from the unknown initial state vector and add the period T as a new variable to the vector. Define the new initial state variable vector as:

$$\mathbf{y}(0) = [x^1(0), \dots, x^{i-1}(0), T, x^{i+1}(0), \dots, x^n(0)]^T \quad (6.3.12)$$

Rewrite Eqn.(6.3.11) using this newly defined vector:

$$\mathbf{y}(0) = \bar{\mathbf{E}}(\mathbf{y}(0)) \quad (6.3.13)$$

where $\bar{\mathbf{E}}(\mathbf{y}(0))$ is equal to $\mathbf{E}(T, \mathbf{x}(0))$ with $x^i(0)$ being fixed. Then the unknown $\mathbf{y}(0)$ can be calculated by the Newton iteration technique.

$$\mathbf{y}^{(j+1)}(0) = \mathbf{y}^{(j)}(0) - \left[\frac{\partial \mathbf{H}}{\partial \mathbf{y}}(\mathbf{y}^{(j)}(0)) \right]^{-1} \mathbf{H}(\mathbf{y}^{(j)}) \quad (6.3.14)$$

where

$$\mathbf{H} = \mathbf{y}(0) - \bar{\mathbf{E}}(\mathbf{y}(0)) \quad (6.3.15)$$

and the Jacobian matrix $\frac{\partial \mathbf{H}}{\partial \mathbf{y}}(\mathbf{y}^{(j)}(0))$ is given by

$$\frac{\partial \mathbf{H}}{\partial \mathbf{y}}(\mathbf{y}^{(j)}(0)) = \left[\Phi^{1(j)}, \dots, \Phi^{i-1(j)}, -s(\mathbf{x}(T^{(j)})), \Phi^{i+1(j)}, \dots, \Phi^{n(j)} \right] \quad (6.3.16)$$

with Φ^l being the l^{th} column of $\left[\mathbf{I} - \frac{\partial \bar{\mathbf{E}}(\mathbf{y}(0))}{\partial \mathbf{y}(0)} \right]$, or the l^{th} column of $\left[\mathbf{I} - \frac{\partial \mathbf{E}(\mathbf{x}(0))}{\partial \mathbf{x}(0)} \right]$ of the

periodic nonautonomous systems for $l = 1, \dots, i-1, i+1, \dots, n$. Note that

$$i^{\text{th}} \text{ column of } \frac{\partial \mathbf{H}}{\partial \mathbf{y}}(\mathbf{y}^{(j)}(0)) = \frac{\partial \mathbf{H}(\mathbf{y}^{(j)}(0))}{\partial T} \Big|_{T^{(j)}} = \frac{\partial \bar{\mathbf{E}}(\mathbf{y}^{(j)}(0))}{\partial T} \Big|_{T^{(j)}} = -s(\mathbf{x}(T^{(j)})) \quad (6.3.17)$$

Under the assumption that there exists a unique periodic solution for the periodic autonomous system, the Newton method in the time domain will converge as long as the initial guess is sufficiently close to the correct solution. The final solution consists of not only a desired initial

state $\mathbf{x}(0)$, but also the period T of oscillation.

6.4. The Harmonic-Balance Method

The harmonic-balance method is a general method for finding solutions of periodic systems. This method transforms the difficult problem of finding periodic solutions of nonlinear differential equations to an easier problem of solving systems of nonlinear algebraic equations. The idea is to represent the system variables by Fourier series and try to find a set of Fourier coefficients which satisfy the system of equations in the frequency domain. For simplicity, we will only focus on periodic nonautonomous systems in this section and briefly discuss its application to the autonomous case.

Consider the same periodic nonautonomous system of equation, as given in Eqn. (6.2.4):

$$\dot{\mathbf{x}} = \mathbf{s}(\mathbf{x}, t) \quad (6.4.1)$$

where \mathbf{x} and \mathbf{s} are n -tuple vectors which are periodic in t with period $T_o = \frac{2\pi}{\omega_o}$, and \mathbf{s} has a continuous first partial derivative with respect to \mathbf{x} for all \mathbf{x} and all t .

Under these mild conditions, the solution $\mathbf{x}(t)$ can be represented using the Fourier series expansion:

$$\mathbf{x}(t) = \sum_{k=-\infty}^{\infty} \mathbf{X}^C[k\omega_o] e^{jk\omega_o t} \quad (6.4.2)$$

Let $\mathbf{X} = [\cdots, \mathbf{X}^C[-k\omega_o], \cdots, \mathbf{X}^C[0], \cdots, \mathbf{X}^C[k\omega_o], \cdots]^T$ be the vector of Fourier coefficients. Since $\mathbf{s}(\mathbf{x}, t)$ is also a periodic function with the same period, it can be written as

$$\mathbf{s}(\mathbf{x}, t) = \sum_{k=-\infty}^{\infty} \mathbf{S}^C[k\omega_o] e^{jk\omega_o t} \quad (6.4.3)$$

where $\mathbf{S} = [\cdots, \mathbf{S}^C[-k\omega_o], \cdots, \mathbf{S}^C[0], \cdots, \mathbf{S}^C[k\omega_o], \cdots]^T$ is the vector of Fourier coefficients of $\mathbf{s}(t)$.

Using the orthogonal property of the function $e^{jk\omega_o t}$, we have the following equation for the Fourier coefficients:

$$j k \omega_o X^C [k \omega_o] = S^C [k \omega_o](X) \quad (6.4.4)$$

for all integers k .

Now we have replaced Eqn. (6.4.1) by infinitely many nonlinear algebraic equations as stated in Eqn.(6.4.4). It is seldom possible to solve infinite dimensional problems such as the one described by Eqn.(6.4.4). Hence we need to find a finite dimensional system to approximate it.

Define

$$P_H \mathbf{x}(t) \equiv \mathbf{x}_H(t) = \sum_{k=1-H}^{H-1} X^C [k \omega_o] e^{jk \omega_o t} \quad (6.4.5)$$

as the truncated Fourier expansion of $\mathbf{x}(t)$. Applying the same truncation to Eqn. (6.4.1), we get a reduced system:

$$\frac{d \mathbf{x}_H}{dt} = \mathbf{s}_H(\mathbf{x}_H, t)$$

or

$$j k \omega_o \sum_{k=1-H}^{H-1} X^C [k \omega_o] e^{jk \omega_o t} = \sum_{k=1-H}^{H-1} S^C [k \omega_o] e^{jk \omega_o t} \quad (6.4.6)$$

This is equivalent to solving Eqn. (6.4.4) for $k = 1-H, \dots, H-1$. It is to be expected that, for H sufficiently large, the solution of Eqn. (6.4.6), $\mathbf{x}_H(t)$ is a reasonable approximation to the exact solution $\hat{\mathbf{x}}(t)$ of Eqn. (6.4.1). If a periodic nonautonomous system can be described by the state equation of the form: $\dot{\mathbf{x}}(t) = \mathbf{s}(\mathbf{x}, t)$, and if an isolated solution exists, then, the harmonic-balance method produces an arbitrarily accurate solution for H sufficiently large[25].

For periodic autonomous systems, it is easier to use the feedback system representation as shown in Fig. 6.1. Assume that the feedback system can be described by the following equation:

$$\mathbf{g} \cdot \mathbf{h}(\mathbf{e}(t)) + \mathbf{e}(t) = 0 \quad (6.4.7)$$

where $\mathbf{h}:R^n \rightarrow R^m$ is a nonlinear function with $\mathbf{h}(\mathbf{e}(t))$ having the same period as \mathbf{e} , and $\mathbf{g}:R^m \rightarrow R^n$ is a linear function. We assume that the solution can be written in the form:

$$\mathbf{e}(t) = \sum_{k=-\infty}^{\infty} \mathbf{E}^C [k \omega] e^{jk \omega t} \quad (6.4.8)$$

where ω is the unknown angular frequency. Since $\mathbf{h}(\mathbf{e}(t))$ is periodic with the same period as $\mathbf{e}(t)$,

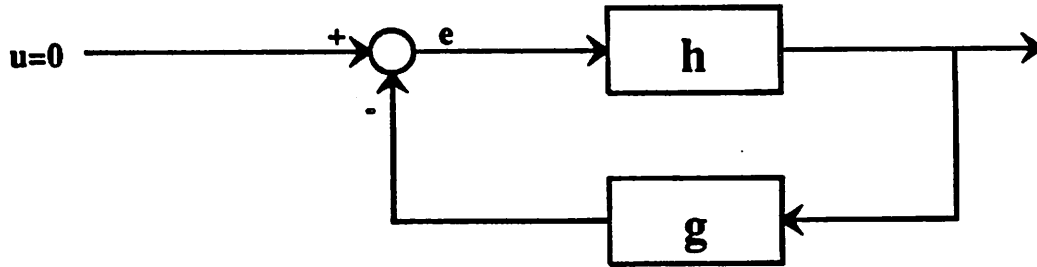


Figure 6.1 Feedback System Representation

it can be written as:

$$\mathbf{h}(\mathbf{e}(t)) = \sum_{k=-\infty}^{\infty} \mathbf{H}^C[k\omega] e^{jk\omega t} \quad (6.4.9)$$

Let $\mathbf{G}(jk\omega)$ be the Fourier transform of \mathbf{g} . Using the harmonic-balance method, we can transform Eqn. (6.4.7), the time-domain feedback system description, into a frequency domain description:

$$\mathbf{G}(jk\omega) \mathbf{H}[k\omega] + \mathbf{E}[k\omega] = 0 \quad \text{for all } k. \quad (6.4.10)$$

Because autonomous systems do not have isolated solutions, we need to either assume an appropriate value for one of the node voltage or find another independent equation relating these unknown variables, as was done in the time-domain Newton method described in Section 6.3.2.

Again, a sensible approximation to the infinite-dimensional problem is to take some sufficiently large integer H , and set $\mathbf{E}^C(k\omega) = 0$ for all $|k| \geq H$. Then, we are left with a finite set of equations with a finite number of unknowns.

Under some mild condition, it has been shown that the solution obtained by the harmonic-balance method is the approximate periodic solution of the Eqn (6.4.7)[26]. Under the same condition, the existence of a harmonic balance solution implies that the autonomous system has a solution.

CHAPTER 7

The Harmonic-Newton Method

In the time domain, a circuit can be modeled by a system of n nonlinear differential equations as described by Eqn. (6.2.1) which is again given below:

$$\mathbf{f}(\mathbf{v}, t) = \mathbf{i}(\mathbf{v}(t)) + \dot{\mathbf{q}}(\mathbf{v}(t)) + \int_0^t \mathbf{y}(t-\tau)\mathbf{v}(\tau)d\tau + \mathbf{u}_s(t) = 0 \quad (7.1)$$

In this chapter, we will discuss how we can solve the equation above in the frequency domain using the harmonic-Newton method. This method had been introduced and implemented in Spectre[4], a simulation package designed to analyze quickly large nonautonomous circuits with nonlinear devices. The harmonic-Newton method first converts the system of differential equations into a system of nonlinear algebraic equations using the technique of harmonic balance described in the previous chapter and then solves the system of equations using the Newton-Raphson method. In Section 7.1, we will focus on the periodic nonautonomous systems. Extension of this method to almost-periodic circuits, a special case of nonautonomous system is exploited in Section 7.2.

So far, the frequency-domain simulator, Spectre is limited to the simulation of nonautonomous systems. In Section 7.3, we will describe how the harmonic-Newton method can be modified to extend its application to periodic autonomous (oscillator) systems in which the period of oscillation is, in general, unknown.

7.1. Periodic Nonautonomous Systems

Under the conditions stated in Chapter 6, the solution $\mathbf{v}(t)$ can be expressed in Fourier series as:

$$\mathbf{v}(t) = \sum_{k=-\infty}^{\infty} \mathbf{V}^C[k\omega_o] e^{jk\omega_o t} \quad (7.1.1)$$

where ω_o is the angular frequency of the circuit. Let

$$\mathbf{V} = [\cdots, \mathbf{V}^C[-k\omega_o], \cdots, \mathbf{V}^C[0], \cdots, \mathbf{V}^C[k\omega_o], \cdots]^T$$

be the vector of Fourier coefficients with $\mathbf{V}^C[k\omega_o] \in C^n$ for all k . Since the nonlinear function $\mathbf{f}(\mathbf{v}, t)$ is periodic with the same period $2\pi/\omega_o$, it can be written as

$$\mathbf{f}(\mathbf{v}, t) = \sum_{k=-\infty}^{\infty} \mathbf{F}^C[k\omega_o] e^{jk\omega_o t} \quad (7.1.2)$$

where $\mathbf{F} = [\cdots, \mathbf{F}^C[-k\omega_o], \cdots, \mathbf{F}^C[0], \cdots, \mathbf{F}^C[k\omega_o], \cdots]^T$ is the vector of Fourier coefficients of $\mathbf{f}(t)$, and $\mathbf{F}^C[k\omega_o] \in C^n$ for all integer k .

Applying the harmonic balance method described in the previous chapter to this system, we have:

$$\mathbf{F}^C[k\omega_o](\mathbf{V}) = 0 \quad \text{for all } k. \quad (7.1.3)$$

To solve the system in the frequency domain, it is necessary to truncate the number of harmonics, H , considered. Define

$$\begin{aligned} \mathbf{v}_H(t) &\equiv P_H \mathbf{v}(t) \equiv \sum_{k=1-H}^{H-1} \mathbf{V}^C[k\omega_o] e^{jk\omega_o t} \\ &= \mathbf{V}^R[0] + \sum_{k=1}^{H-1} \left\{ \mathbf{V}^R[k\omega_o] \cos(k\omega_o t) - \mathbf{V}^I[k\omega_o] \sin(k\omega_o t) \right\} \end{aligned} \quad (7.1.4)$$

and

$$\begin{aligned} \mathbf{f}_H(\mathbf{v}, t) &\equiv P_H \mathbf{f}(\mathbf{v}, t) \equiv \sum_{k=1-H}^{H-1} \mathbf{F}^C[k\omega_o] e^{jk\omega_o t} \\ &= \mathbf{F}^R[0] + \sum_{k=1}^{H-1} \left\{ \mathbf{F}^R[k\omega_o] \cos(k\omega_o t) - \mathbf{F}^I[k\omega_o] \sin(k\omega_o t) \right\} \end{aligned} \quad (7.1.5)$$

as Fourier expansions of $\mathbf{v}(t)$ and $\mathbf{f}(\mathbf{v}, t)$ of order H . $\mathbf{V}^R[0]$ and $\mathbf{F}^R[0]$ are the real part of $\mathbf{V}^C[0]$ and $\mathbf{F}^C[0]$ respectively. However, for the purpose of convenience, we define $\mathbf{V}^R[k\omega_o]$ and $\mathbf{F}^R[k\omega_o]$ as twice of the real part of $\mathbf{V}^C[k\omega_o]$ and $\mathbf{F}^C[k\omega_o]$ respectively. Similarly, $\mathbf{V}^I[k\omega_o]$ and $\mathbf{F}^I[k\omega_o]$ are twice of the imaginary part of $\mathbf{V}^C[k\omega_o]$ and $\mathbf{F}^C[k\omega_o]$ respectively. Since $\mathbf{f}(\mathbf{v}, t)$ and $\mathbf{v}(t)$ are real vectors of the same dimension n , $\mathbf{V}^R[0]$, $\mathbf{V}^R[\omega_o]$, $\mathbf{V}^I[\omega_o]$ are also n -tuple vectors.

We further define:

$$\mathbf{V}_H \equiv \begin{bmatrix} \mathbf{V}^R[0] \\ \mathbf{V}^R[\omega_o] \\ \mathbf{V}^I[\omega_o] \\ \vdots \\ \mathbf{V}^R[(H-1)\omega_o] \\ \mathbf{V}^I[(H-1)\omega_o] \end{bmatrix} \quad \mathbf{F}_H \equiv \begin{bmatrix} \mathbf{F}^R[0] \\ \mathbf{F}^R[\omega_o] \\ \mathbf{F}^I[\omega_o] \\ \vdots \\ \mathbf{F}^R[(H-1)\omega_o] \\ \mathbf{F}^I[(H-1)\omega_o] \end{bmatrix} \quad (7.1.6)$$

where $\mathbf{V}_H \in R^{(2H-1)n}$ and $\mathbf{F}_H : R^{(2H-1)n} \rightarrow R^{(2H-1)n}$. Similar definitions are used for $\mathbf{i}_H(t)$, $\mathbf{y}_H(t)$, and $\mathbf{q}_H(t)$.

Since \mathbf{y} is the linear matrix-valued impulse response of the circuit with voltage and current defined as the input and output variables respectively, the zero-state current waveform can be obtained by convoluting the impulse response with the node voltage. The zero-state current waveform in the frequency domain is then the product of the Fourier transform of the impulse response and the Fourier transform of the voltage, i.e.:

$$\int_{-\infty}^t \mathbf{y}(t-\tau)\mathbf{v}(\tau) d\tau \longleftrightarrow \mathbf{Y}\mathbf{V} \quad (7.1.7)$$

Note that \mathbf{Y} is the node admittance matrix and is block diagonal

With these new definitions, we can describe the reduced system of order H as:

$$\mathbf{F}_H(\mathbf{V}_H) = \mathbf{I}_H(\mathbf{V}_H) + \Omega \mathbf{Q}_H(\mathbf{V}_H) + \mathbf{Y}_H \mathbf{V}_H + \mathbf{U}_H = 0 \quad (7.1.8)$$

where Ω is a matrix containing block elements $\Omega[k, l]$ given below:

$$\Omega[k, l] = \begin{cases} \begin{bmatrix} 0 & \Omega^{mn}[k, k] \\ -\Omega^{mn}[k, k] & 0 \end{bmatrix} & k = l \\ 0 & k \neq l \end{cases} \quad (7.1.9)$$

where k, l is the frequency index, m, n is the node index of the circuit, and the matrix $\Omega^{mn}[k, k]$ is a diagonal matrix with diagonal element equal to $k\omega_o$ for all k .

It is to be expected that, if H is chosen sufficiently large, the voltage $\mathbf{v}_H(t)$ determined by Eqn. (7.1.8) may be a reasonable approximation to the exact solution $\hat{\mathbf{v}}(t)$ of Eqn. (7.1). To solve the reduced system, we apply the Newton-Raphson method to Eqn. (7.1.8) to get the following sequence of iterations:

$$\mathbf{V}_H^{(j+1)} = \mathbf{V}_H^{(j)} - \mathbf{J}_H^{-1}(\mathbf{V}_H^{(j)}) \mathbf{F}_H(\mathbf{V}_H^{(j)}) \quad (7.1.10)$$

where $\mathbf{J}_H(\mathbf{V}_H) \in R^{(2H-1)n \times (2H-1)n}$ is referred as the *Spectral Jacobian*[4] and is given by:

$$\mathbf{J}_H(\mathbf{V}_H) = \frac{\partial \mathbf{F}_H(\mathbf{V}_H)}{\partial \mathbf{V}_H} = \frac{\partial \mathbf{I}_H(\mathbf{V}_H)}{\partial \mathbf{V}_H} + \Omega \frac{\partial \mathbf{Q}_H(\mathbf{V}_H)}{\partial \mathbf{V}_H} + \mathbf{Y}_H \quad (7.1.11)$$

The spectral Jacobian is a block matrix whose $[k, l]$'s block $\mathbf{J}_H[k, l]$ is given by:

$$\mathbf{J}_H[k, l] = \begin{bmatrix} \frac{\partial \mathbf{F}[k \omega_o]}{\partial \mathbf{V}[l \omega_o]} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{F}^R[k \omega_o]}{\partial \mathbf{V}^R[l \omega_o]} & \frac{\partial \mathbf{F}^R[k \omega_o]}{\partial \mathbf{V}^I[l \omega_o]} \\ \frac{\partial \mathbf{F}^I[k \omega_o]}{\partial \mathbf{V}^R[l \omega_o]} & \frac{\partial \mathbf{F}^I[k \omega_o]}{\partial \mathbf{V}^I[l \omega_o]} \end{bmatrix} \quad (7.1.14)$$

$$= \begin{bmatrix} \frac{\partial \mathbf{I}^R[k \omega_o]}{\partial \mathbf{V}^R[l \omega_o]} & \frac{\partial \mathbf{I}^R[k \omega_o]}{\partial \mathbf{V}^I[l \omega_o]} \\ \frac{\partial \mathbf{I}^I[k \omega_o]}{\partial \mathbf{V}^R[l \omega_o]} & \frac{\partial \mathbf{I}^I[k \omega_o]}{\partial \mathbf{V}^I[l \omega_o]} \end{bmatrix} + \Omega \begin{bmatrix} \frac{\partial \mathbf{Q}^R[k \omega_o]}{\partial \mathbf{V}^R[l \omega_o]} & \frac{\partial \mathbf{Q}^R[k \omega_o]}{\partial \mathbf{V}^I[l \omega_o]} \\ \frac{\partial \mathbf{Q}^I[k \omega_o]}{\partial \mathbf{V}^R[l \omega_o]} & \frac{\partial \mathbf{Q}^I[k \omega_o]}{\partial \mathbf{V}^I[l \omega_o]} \end{bmatrix}$$

$$+ \begin{bmatrix} \mathbf{Y}^R[k, l] & -\mathbf{Y}^I[k, l] \\ \mathbf{Y}^I[k, l] & \mathbf{Y}^R[k, l] \end{bmatrix} \quad (7.1.15)$$

Note that each component of the matrices in Eqn. (7.1.15) is also a block matrix of dimension $n \times n$. Since the function is nonlinear, the derivatives in Eqn. (7.1.15) should be calculated in the time domain and then converted back to the frequency domain.

Let us now summarize the harmonic-Newton method as follows:

STEP-I:

Choose the number of harmonics H and truncate all higher harmonic terms. Set $j = 0$.

Choose an initial guess voltage vector $\mathbf{V}_H^{(j)}$.

STEP-II:

Evaluate $\mathbf{F}_H(\mathbf{V}_H^{(j)})$ and the spectral Jacobian $\mathbf{J}_H(\mathbf{V}_H^{(j)})$ of the network.

STEP-III:

Generate new node voltage spectra using update Eqn. (7.1.10)

$$\mathbf{V}_H^{(j+1)} = \mathbf{V}_H^{(j)} - \mathbf{J}_H^{-1}(\mathbf{V}_H^{(j)}) \mathbf{F}_H(\mathbf{V}_H^{(j)})$$

STEP-IV:

Check if the circuit is harmonic balanced by checking if $\|F_H\| \leq \epsilon$, where ϵ is the threshold for the circuit to be considered harmonic balanced. If no, set $j = j + 1$ and go to STEP-II.

STEP-V:

Done.

7.2. Almost-Periodic Nonautonomous Systems

The harmonic-Newton method described above can also be applied to almost-periodic circuits. The only difference is in the way that the sets of frequencies are truncated.

Let us refer back to Eqn. (7.1). Since $v(t)$ is almost periodic, both $i(t)$, and $q(t)$ are almost periodic. When applying the harmonic-Newton method described above to almost periodic systems, all terms of Eqn. (7.1) can be transformed into the frequency domain using the almost periodic Fourier transform (APFT)[23]. While $v(t)$, $i(t)$, $q(t)$ are vectors of waveforms, V , I , Q , the APFT of $v(t)$, $i(t)$, $q(t)$, are vectors of spectra. Define the truncated vectors V_H , I_H , Q_H as follows:

$$V_H = [V^R [0], V^R [\omega_1], V^I [\omega_1], \dots , V^R [\omega_K], V^I [\omega_K]]$$

or

$$v_H(t) \equiv P'_H v(t) \equiv \sum_{\omega_k \in \Lambda_H} \left\{ V^R \cos(\omega_k t) + V^I \sin(\omega_k t) \right\}$$

Note that $V^R [0]$ is the real part of $V^C [0]$; but $V^R [\omega_k]$ and $V^I [\omega_k]$ are defined as twice of the real part and imaginary part of $V^C [\omega_k]$ respectively. Similarly, the vectors I_H and Q_H are:

$$I_H = [I^R [0], I^R [\omega_1], I^I [\omega_1], \dots , I^R [\omega_K], I^I [\omega_K]]^T$$

$$Q_H = [Q^R [0], Q^R [\omega_1], Q^I [\omega_1], \dots , Q^R [\omega_K], Q^I [\omega_K]]^T$$

All components in the vectors V_H , I_H , Q_H are themselves vectors of R^n .

As described in the previous section, the Fourier transform of the convolution integral

$\int_{-\infty}^t y(t-\tau)v(\tau)d\tau$ is equal to the product of Y and V . With these new definitions, we can describe

the reduced system of order H as:

$$\mathbf{F}_H(\mathbf{V}_H) = \mathbf{I}_H(\mathbf{V}_H) + \Omega \mathbf{Q}_H(\mathbf{V}_H) + \mathbf{Y}_H \mathbf{V}_H + \mathbf{U}_H = 0$$

Thus, the sequence of the harmonic-Newton iteration for almost periodic circuits is the same as that of periodic circuits:

$$\mathbf{V}_H^{(j+1)} = \mathbf{V}_H^{(j)} - \mathbf{J}_H^{-1}(\mathbf{V}_H^{(j)}) \mathbf{F}_H(\mathbf{V}_H^{(j)})$$

The spectral Jacobian matrix is given as follows:

$$\begin{aligned} \mathbf{J}_H[k, l] = & \begin{bmatrix} \frac{\partial \mathbf{I}^R[\omega_k]}{\partial \mathbf{V}^R[\omega_l]} & \frac{\partial \mathbf{I}^R[\omega_k]}{\partial \mathbf{V}^I[\omega_l]} \\ \frac{\partial \mathbf{I}^I[\omega_k]}{\partial \mathbf{V}^R[\omega_l]} & \frac{\partial \mathbf{I}^I[\omega_k]}{\partial \mathbf{V}^I[\omega_l]} \end{bmatrix} + \Omega \begin{bmatrix} \frac{\partial \mathbf{Q}^R[\omega_k]}{\partial \mathbf{V}^R[\omega_l]} & \frac{\partial \mathbf{Q}^R[\omega_k]}{\partial \mathbf{V}^I[\omega_l]} \\ \frac{\partial \mathbf{Q}^I[\omega_k]}{\partial \mathbf{V}^R[\omega_l]} & \frac{\partial \mathbf{Q}^I[\omega_k]}{\partial \mathbf{V}^I[\omega_l]} \end{bmatrix} \\ & + \begin{bmatrix} \mathbf{Y}^R[k, l] & -\mathbf{Y}^I[k, l] \\ \mathbf{Y}^I[k, l] & \mathbf{Y}^R[k, l] \end{bmatrix} \end{aligned} \quad (7.1.16)$$

where Ω is a block matrix consisting of $\{ \Omega[k, l] \}$:

$$\Omega[k, l] = \begin{cases} \begin{bmatrix} 0 & \Omega^{mn}[k, k] \\ -\Omega^{mn}[k, k] & 0 \end{bmatrix} & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

$\Omega^{mn}[k, k]$ in the above equation is a diagonal matrix with diagonal element equal to ω_k for all k .

7.3. Periodic Autonomous Systems

A periodic autonomous system can be described in the time domain by the following differential equation:

$$\mathbf{i}(\mathbf{v}(t)) + \dot{\mathbf{q}}(\mathbf{v}(t)) + \int_0^t \mathbf{y}(t-\tau)\mathbf{v}(\tau)d\tau = 0 \quad (7.3.1)$$

Note that no external source is present and the frequency of the circuit ω is generally unknown.

Applying the harmonic-Newton method with H harmonics, we can transform Eqn. (7.3.1) into a frequency domain description:

$$\mathbf{I}_H(\mathbf{V}_H) + \Omega(\omega) \mathbf{Q}_H(\mathbf{V}_H) + \mathbf{Y}_H \mathbf{V}_H = 0 \quad (7.3.2)$$

This equation is identical to Eqn. (7.1.8) of the periodic nonautonomous systems except that no external source \mathbf{U}_H is present in this case.

Unfortunately, Eqn. (7.3.2) doesn't have isolated solutions because there are $(2H-1)n + 1$ real unknowns ($V^R[0], V^R[\omega], V^I[\omega], \dots, V^R[(H-1)\omega], V^I[(H-1)\omega]$ and ω), but only $(2H-1)n$ real equations. Moreover, the harmonic-Newton method will fail easily in this case because the Jacobian matrix may become singular at oscillating frequency which results in nonconvergence of the Newton iteration. This is in fact as it should be because the time origin is arbitrary for an autonomous system. More specifically, if some vector $[\bar{V}^C[(1-H)\omega], \dots, \bar{V}^C[(H-1)\omega]]^T$ satisfies the equations, then the vector $[\bar{V}^C[(1-H)\omega] \exp i\theta, \dots, \bar{V}^C[(H-1)\omega] \exp i\theta]^T$ will also satisfy the same set of equations for any arbitrary real θ .

One way to reduce the number of solutions to one is to add artificially a condition that a nonzero element $V^{C^{i_0}}[k_0 \omega]$ for some $i_0, k_0 \neq 0$ be real and positive, i.e. $\arg V^{C^{i_0}}[k_0 \omega] = 0$. However, \arg is a continuous function only in subsets of the complex plane that do not contain or encircle the origin. If one adds the condition stated above, it is necessary to impose another condition later to prevent the function from being discontinuous on certain sets.

An easier way to solve this problem and at the same time avoid discontinuity is to add a condition that a nonzero element $V^{C^{i_0}}[k_0 \omega]$ for some $i_0, k_0 \neq 0$ be real, i.e. $V^{I^{i_0}}[k_0 \omega] = 0$. This will reduce the number of solutions satisfying Eqn.(7.3.2) to two isolated ones. Note that it is important to choose i_0 and a nonzero k_0 such that the real part of $V^{C^{i_0}}[k_0 \omega]$ is not equal to zero. If the real part of $V^{C^{i_0}}[k_0 \omega]$ turns out to be zero, we have not successfully fixed the time origin. Therefore, another i_0 and k_0 have to be chosen. We can also verify this condition throughout the Newton iteration to make sure that the real part of $V^{C^{i_0}}[k_0 \omega]$ remains nonzero. If iteration result shows that the real part of $V^{C^{i_0}}[k_0 \omega]$ is too small, we can then choose another i_0' and k_0' . We would like to point out that changing i_0 and k_0 affects the Newton iteration process only mildly. Two actions have to be taken: (1) the location of "1" in the last row of the Jacobian matrix should be changed

accordingly, and (2) the updated solution vector $V^{C(j)}$ needs to be multiplied by $\exp(-i\theta)$ where

$$\theta = \arctan \left[\frac{V^{i'o}[k_o\omega]}{V^{R'i}[k_o\omega]} \right]$$

The Newton iteration can still continue and if it converges, it will converge to the correct solution.

Incorporating this condition into Eqn. (7.3.1) and keeping only the first H harmonics, we have:

$$\tilde{F}_H(\mathbf{x}_H, t) \equiv \left\{ \begin{array}{l} \tilde{g}_H(\mathbf{x}_H, t) \equiv \dot{\mathbf{i}}_H(\mathbf{v}_H(t)) + \dot{\mathbf{q}}_H(\mathbf{v}_H(t)) + \int_0^t \mathbf{y}_H(t-\tau)\mathbf{v}_H(\tau)d\tau \\ \frac{1}{T} \int_0^T v^{i'o}(t) \sin(k_o \omega t) \end{array} \right\} = 0 \quad (7.3.3)$$

where the new variable vector \mathbf{x}_H is given by $[\mathbf{v}_H, \omega]^T$. A frequency domain equivalent of Eqn. (7.3.3) is

$$\tilde{F}_H(\mathbf{X}_H) \equiv \left\{ \begin{array}{l} \tilde{G}_H(\mathbf{V}_H, \omega) \equiv \mathbf{I}_H(\mathbf{V}_H) + \Omega(\omega) \mathbf{Q}_H(\mathbf{V}_H) + \mathbf{Y}_H \mathbf{V}_H \\ V^{i'o}[k_o\omega] \end{array} \right\} = 0 \quad (7.3.4)$$

where the vector \mathbf{X}_H is given by $[\mathbf{V}_H, \omega]^T$.

The harmonic-Newton iteration of Eqn. (7.3.4) is given by:

$$\left[\begin{array}{l} \mathbf{V}_H^{(j+1)} \\ \omega^{(j+1)} \end{array} \right] = \left[\begin{array}{l} \mathbf{V}_H^{(j)} \\ \omega^{(j)} \end{array} \right] - \bar{\mathbf{J}}_H(\mathbf{V}_H^{(j)}, \omega^{(j)})^{-1} \tilde{F}_H(\mathbf{V}_H^{(j)}, \omega^{(j)}) \quad (7.3.5)$$

where $\bar{\mathbf{J}}_H(\mathbf{V}_H, \omega)$ has the following form:

$$\bar{\mathbf{J}}_H(\mathbf{V}_H, \omega) = \left[\begin{array}{c|c} \mathbf{J}_H(\mathbf{V}_H) & \partial \tilde{F}_H / \partial \omega \\ \hline 0 \cdots 0 \ 1 \ 0 \cdots 0 & 0 \end{array} \right] \quad (7.3.6)$$

with $\mathbf{J}_H(\mathbf{V}_H)$ being the same Jacobian matrix as that of the periodic nonautonomous system, and $\partial \tilde{F}_H / \partial \omega$ being the derivative of the original function with respect to ω . The "1" in the last row of the Jacobian matrix corresponds to the location of this particular $V^{i'o}[k_o\omega]$ in the vector \mathbf{X}_H .

Thus, the characteristics and the structure of the Jacobian matrix is preserved.

As mentioned before, the extra constraint given in Eqn. (7.3.4) reduces the number of solutions to two isolated ones, thus there is still no unique solution for Eqn. (7.3.4). For nonlinear systems, the fact that solutions are isolated but not unique does not result in Jacobian matrix being singular at oscillating frequency. If the system is linear, $\tilde{\mathbf{J}}_H(\mathbf{V}_H, \omega)$ is still singular at oscillating frequency. This makes sense intuitively since for a linear oscillator not only the phase of the solution is arbitrary, but also the amplitude \mathbf{V}_H can be arbitrary. For this reason, we not only have to fix the phase of the voltage vector but also the amplitude of the oscillation to prevent the Jacobian matrix from becoming singular at oscillating frequency. Fortunately, most of the autonomous circuits are oscillators which require nonlinear elements to provide sufficient negative resistance to start the oscillation while the negative feedback decreases as the amplitude grows, causing the oscillation to be sustained only at some fixed amplitude. Thus, the amplitude of oscillation for an autonomous system is usually not arbitrary. Since the Jacobian matrix of a nonlinear system is in general nonsingular at oscillating frequency as long as the solutions are isolated, the modified method described in this section can be applied successfully to most autonomous systems. Depending on the choice of the initial condition, the harmonic-Newton method will converge to one of the isolated solutions.

CHAPTER 8

Convergence of the Harmonic-Newton Method

In Chapter 7, we discussed how we can use the harmonic-Newton method to find periodic or almost-periodic steady-state solutions of autonomous or nonautonomous systems in the frequency domain. We also pointed out that in order to use the harmonic-Newton method, it is necessary to truncate the number of harmonics, H , to some finite number so that the infinite-dimensional system can be approximated by a finite-dimensional one. Since truncation is necessary, we need to study the convergence properties of the harmonic-Newton method in order to ensure that the solutions obtained are meaningful after truncation has taken place. The harmonic-Newton method is said to be convergent if for any H considered, the sequence of iterations defined by this algorithm converges to a fixed point, such that these fixed points themselves form a sequence of solutions whose limit is the exact solution of the given circuit system as $H \rightarrow \infty$. In this chapter, we will study in detail the convergence properties of the harmonic-Newton method and show that under some conditions, the harmonic-Newton method is a convergent method for periodic autonomous and nonautonomous systems.

8.1. Background

In this section, we introduce some classic theorems which will be used later in the convergence proofs.

Theorem A: [13]

Assume that the function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is Gateaux-differentiable on a convex set D . Then for any $x, y \in D$, the following inequality holds:

$$\|f(y) - f(x)\| \leq \max_{0 \leq t \leq 1} \|f'(x + t(y-x))\| \cdot \|x - y\|$$

where f' is the first Gateaux-derivative of f .

Theorem B: [13]

Assume that the function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is twice Gateaux-differentiable on a convex set D .

Then for any $x, y \in D$, the following inequality holds:

$$\|f'(y) - f'(x)\| \leq \max_{0 \leq t \leq 1} \|f''(x + t(y-x))\| \cdot \|x - y\|$$

where f' and f'' are the first and second Gateaux-derivatives of f respectively.

Theorem C: [27]

Let x be a periodic function with an interval of periodicity $[0, T_0]$. If the function itself together with its derivatives up to order $\kappa-1$, $\kappa \geq 1$ are continuous, and its κ -th derivative is piecewise continuous in the interval, then there exists a constant $A > 0$ such that the Fourier coefficients of the function satisfy the inequality $|X[k\omega_0]| \leq \frac{A}{|k|^{\kappa+1}}$.

Parseval's Equation: [24]

Periodic Functions

Let $f(x)$ be an arbitrary continuous function of period $T_0 = \frac{2\pi}{\omega_0}$. Then, the Parseval's

equality $\sum_{k=-\infty}^{\infty} |F[k\omega_0]|^2 = \frac{1}{T_0} \int_0^{T_0} |f(x)|^2 dx$ holds for all $f(x)$, where the Fourier coefficients

$F[k\omega_0]$ of $f(x)$ is given by

$$F[k\omega_0] = \frac{1}{T_0} \int_0^{T_0} f(x) e^{-jk\omega_0 x} dx$$

Almost-Periodic Functions

For every almost-periodic function, $f(x) = \sum_{\omega_k \in \Lambda} F[\omega_k] e^{j\omega_k x}$, there exists a mean value

$M(f(x))$ defined as:

$$M(f(x)) \equiv \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T f(x) dx$$

i.e., the expression $\frac{1}{T} \int_0^T f(x) dx$ approaches a definite finite limit as $T \rightarrow \infty$. The Parseval's equation holds for almost-periodic function and can be written in exactly the same form as in the case of periodic systems:

$$\sum_{\omega_k \in \Lambda} |F[\omega_k]|^2 = M(|f(x)|^2).$$

Newton-Kantorovich Theorem: [13]

Assume that $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is differentiable on a convex set D_0 which is inside of D , and that for any $x, y \in D_0$, the following inequality holds

$$\|f'(y) - f'(x)\| \leq \gamma \|x - y\|$$

where f' is the first derivatives of f .

Also assume that there exists an $x^{(0)} \in D_0$ such that $\|f'(x^{(0)})^{-1}\| \leq \beta$ and $\alpha \equiv \beta\gamma\eta \leq \frac{1}{2}$, where $\eta \geq \|f'(x^{(0)})^{-1}f(x^{(0)})\|$.

Set

$$t^* = (\beta\gamma)^{-1}[1 - (1 - 2\alpha)^{\frac{1}{2}}], \quad t^{**} = (\beta\gamma)^{-1}[1 + (1 - 2\alpha)^{\frac{1}{2}}]$$

and assume that $\bar{B}(x^{(0)}, t^*)$ lies inside of D_0 . Then the Newton iteration

$$x^{(j+1)} = x^{(j)} - f'(x^{(j)})^{-1}f(x^{(j)}), \quad j = 0, 1, \dots$$

is well-defined, with $x^{(j+1)}$ remaining in $\bar{B}(x^{(0)}, t^*)$ and converging to the solution x^* of $F(x) = 0$ which is unique in $\bar{B}(x^{(0)}, t^{**}) \cap D_0$.

Moreover, the following inequality which bounds the difference between the j -th iteration and the exact solution holds;

$$\|x^* - x^{(j)}\| \leq (\beta\gamma 2^j)^{-1} (2\alpha)^{2^j}, \quad j = 0, 1, \dots$$

In the following three sections, we prove the convergence of the harmonic-Newton method for periodic nonautonomous, periodic autonomous, and almost-periodic circuits, as the number of

harmonics considered goes to infinity. These proofs appear to be complicated and lengthy; however, they are all based on the same concept.

To begin, we must first assume that there exists an isolated solution, $\hat{v}(t)$, to the original differential equation. Let $v_H^*(t)$ be the computed solution obtained by the harmonic-Newton method with H harmonics. To show that the harmonic-Newton method does converge to a solution $v_H^*(t)$ and this solution can be arbitrarily close to the exact solution $\hat{v}(t)$ as $H \rightarrow \infty$, let us examine the norm of the difference between $v_H^*(t)$ and $\hat{v}(t)$:

$$\|v_H^*(t) - \hat{v}(t)\| \leq \|v_H^*(t) - \hat{v}_H(t)\| + \|\hat{v}_H(t) - \hat{v}(t)\| \quad (8.1)$$

The strategy in all the proofs is to handle those two terms in the right hand side of the inequality separately. The second term $\|\hat{v}_H(t) - \hat{v}(t)\|$, represents the error due to the truncation of higher order terms; and this term can be easily shown to approach zero as the number of harmonics, H , approaches infinity. The detail of these proofs for periodic nonautonomous systems, almost-periodic circuits, and periodic autonomous systems can be found in Lemma 8.1, 8.4, and 8.7.

To prove that the first term in the right hand side of Eqn.(8.1) vanishes as $H \rightarrow \infty$, we will use the truncated exact solution, $\hat{v}_H(t)$, as the initial guess voltage for the harmonic-Newton method. We can show that this particular choice of initial guess voltage satisfies all the assumptions stated in the Newton-Kantorovich theorem. Therefore we can use the Newton-Kantorovich theorem to show that the solution of the harmonic-Newton method converge to a solution, $v_H^*(t)$, and the error between the computed solution and the initial guess is bounded. Once the error bound is found, we can prove that this error, $\|\hat{v}_H(t) - v_H^*(t)\|$, approaches zero as $H \rightarrow \infty$. Thus, we can prove that the solution obtained by the harmonic-Newton method will converge to the exact solution as H goes to infinity.

8.2. Periodic Nonautonomous Systems

In this section, we show the convergence of the harmonic-Newton method for periodic nonautonomous systems. We will assume throughout this section that the following assumptions are satisfied:

- (1) The system $f(v, t)$ and its partial derivatives with respect to v are continuously differentiable with respect to v and t in the region $D \times L$, where D is a convex bounded region in the v space and L is the real line.
- (2) $f(v, t) = 0$ is a periodic nonautonomous system.
- (3) If $\varphi(t)$ is the exact solution of the equation $f(v, t) = 0$ and it is in the interior of D , then $\varphi(t)$ is an isolated periodic solution of the equation (i.e. there exists a small number δ such that for all $v \in B = \{v \mid \|v - \varphi\| \leq \delta\}$, $\frac{\partial f}{\partial v}(v, t)$ is nonsingular.)

These assumptions will be referred as Assumption 1, 2, and 3.

Lemma 8.1:

Assume that Assumptions 1, 2, and 3 are satisfied, then the truncated exact solution

$$\varphi_H(t) = P_H \varphi(t) = \hat{V}^R[0] + \sum_{k=1}^{H-1} \left[\hat{V}^R[k \omega_0] \cos(k \omega_0 t) - \hat{V}^I[k \omega_0] \sin(k \omega_0 t) \right]$$

is an approximate solution of the truncated system $f_H(v_H, t) = P_H f(v_H, t) = 0$ (i.e. $f_H(\varphi_H, t) \rightarrow 0$ as $H \rightarrow \infty$).

Proof:

Since $f(\varphi, t) = 0$,

$$f(\varphi_H, t) = f(\varphi_H, t) - f(\varphi, t) \tag{Lem8.1.1}$$

$$P_H f(\varphi_H, t) = P_H [f(\varphi_H, t) - f(\varphi, t)] \tag{Lem8.1.2}$$

or

$$f_H(\varphi_H, t) = P_H [f(\varphi_H, t) - f(\varphi, t)]$$

Let $C_1 \equiv \max_{D \times L} \left\| \frac{\partial f}{\partial v}(v, t) \right\|$

By Theorem A:

$$\begin{aligned} \|f(\varphi_H, t) - f(\varphi, t)\| &\leq \max_{D \times \mathcal{L}} \left\| \frac{\partial f}{\partial v}(v, t) \right\| \cdot \|\varphi_H - \varphi\| \\ &\leq C_1 \|\varphi_H - \varphi\| \end{aligned} \quad (\text{Lem8.1.3})$$

Define

$$\vartheta(t) - \varphi_H(t) = \sum_{k=H}^{\infty} \left[\hat{V}^R[k\omega_o] \cos(k\omega_o t) - \hat{V}^I[k\omega_o] \sin(k\omega_o t) \right] \quad (\text{Lem8.1.4})$$

$$\equiv \sum_{k=H}^{\infty} \left[\underline{a}_k \cos(k\omega_o t) - \underline{b}_k \sin(k\omega_o t) \right] \quad (\text{Lem8.1.5})$$

and

$$\dot{\vartheta}(t) = \sum_{k=1}^{\infty} \left[-k\omega_o \hat{V}^I[k\omega_o] \cos(k\omega_o t) - k\omega_o \hat{V}^R[k\omega_o] \sin(k\omega_o t) \right] \quad (\text{Lem8.1.6})$$

$$\equiv \sum_{k=1}^{\infty} \left[\underline{a}_k \cos(k\omega_o t) - \underline{b}_k \sin(k\omega_o t) \right] \quad (\text{Lem8.1.7})$$

Note that $\underline{a}_k = \frac{\underline{b}_k}{k\omega_o}$, $\underline{b}_k = -\frac{\underline{a}_k}{k\omega_o}$.

$$\vartheta(t) - \varphi_H(t) = \sum_{k=H}^{\infty} \left[\frac{\underline{b}_k}{k\omega_o} \cos(k\omega_o t) - \frac{-\underline{a}_k}{k\omega_o} \sin(k\omega_o t) \right] \quad (\text{Lem8.1.8})$$

$$\|\vartheta(t) - \varphi_H(t)\|^2 \leq \frac{1}{\omega_o^2} \frac{1}{H^2} \sum_{k=H}^{\infty} \left[\|\underline{a}_k\|^2 + \|\underline{b}_k\|^2 \right] \quad (\text{Lem8.1.9})$$

By Bessel's inequality:

$$\|\vartheta(t) - \varphi_H(t)\|^2 \leq \frac{1}{\omega_o^2} \frac{1}{H^2} \|\dot{\vartheta}\|^2 \quad (\text{Lem8.1.10})$$

Using Theorem C, we can show that $\|\dot{\vartheta}\| < \infty$. Define $C \equiv \|\dot{\vartheta}\|$, we have:

$$\|f(\varphi_H, t) - f(\varphi, t)\| \leq \frac{1}{\omega_o} \frac{1}{H} C_1 C \quad (\text{Lem8.1.11})$$

for all $H \geq H_{crit}$ such that $\varphi_H(t) \in D$. Since the basis is orthogonal,

$$\begin{aligned} \|f_H(\varphi_H, t)\| &= \|P_H[f(\varphi_H, t) - f(\varphi, t)]\| \\ &\leq \|f(\varphi_H, t) - f(\varphi, t)\| \leq \frac{1}{\omega_o} \frac{1}{H} C_1 C \end{aligned} \quad (\text{Lem8.1.12})$$

Therefore,

$$f_H(\varphi_H, t) \rightarrow 0 \text{ as } H \rightarrow \infty. \quad (\text{Lem8.1.13})$$

By Parseval's equation:

$$\|f_H(t)\|^2 \equiv \frac{1}{T_o} \int_0^{T_o} \|f_H(t)\|^2 dt = \|F_H\|^2 \quad (\text{Lem8.1.14})$$

Thus,

$$\|F_H(\hat{V}_H)\| = \|f_H(\hat{v}_H, t)\| \leq \frac{1}{\omega_o} \frac{1}{H} C_1 C \quad (\text{Lem8.1.15})$$

Therefore

$$f_H(\hat{v}_H, t) \rightarrow 0 \text{ as } H \rightarrow \infty$$

Q.E.D.

Lemma 8.2:

Assume that Assumptions 1, 2, and 3 are satisfied, then there is a positive integer H_o such that for any $H \geq H_o$, $\det J_H(\hat{V}_H) \neq 0$ provided that H_o is sufficiently large.

Proof:

$J_H(\hat{V}_H)$ is the Jacobian matrix of $F_H(\hat{V}_H)$. To find the basic property of this Jacobian matrix, let us consider the following linear system:

$$J_H(\hat{V}_H)Y_H = Z_H \quad (\text{Lem8.2.1})$$

where

$$Y_H = [Y^R [0], Y^R [\omega_o], Y^I [\omega_o], \dots, Y^R [(H-1)\omega_o], Y^I [(H-1)\omega_o]]^T$$

$$\text{and } Z_H = [Z^R [0], Z^R [\omega_o], Z^I [\omega_o], \dots, Z^R [(H-1)\omega_o], Z^I [(H-1)\omega_o]]^T$$

are any vectors with the same dimension as V_H and $Y^R [0], Y^R [k\omega_o], Y^I [k\omega_o], Z^R [0], Z^R [k\omega_o], Z^I [k\omega_o] \in R^n$ for $k = 1, \dots, H-1$. Define

$$y_H(t) \equiv Y^R [0] + \sum_{k=1}^{H-1} \left[Y^R [k\omega_o] \cos(k\omega_o t) - Y^I [k\omega_o] \sin(k\omega_o t) \right]$$

$$z_H(t) \equiv Z^R [0] + \sum_{k=1}^{H-1} \left[Z^R [k\omega_o] \cos(k\omega_o t) - Z^I [k\omega_o] \sin(k\omega_o t) \right]$$

To show that $\det J_H(\hat{V}_H) \neq 0$ is equivalent to showing that the mapping $J_H(\hat{V}_H)$ is one-to-one or the null space of $J_H(\hat{V}_H)$ is $\{ 0 \}$.

First, we need to prove that Eqn. (Lem8.2.1) is equivalent to the following equation:

$$P_H \left[\frac{\partial f}{\partial v}(\varphi_H, t) y_H(t) \right] = z_H(t) \quad (\text{Lem8.2.2})$$

The DC term of Eqn. (Lem8.2.2) is

$$\begin{aligned} & \frac{1}{T_o} \int_0^{T_o} \frac{\partial f}{\partial v}(\varphi_H, t) \left[Y^R[0] + \sum_{k=1}^{H-1} \left[Y^R[k\omega_o] \cos(k\omega_o t) - Y^I[k\omega_o] \sin(k\omega_o t) \right] \right] dt \\ & = Z^R[0] \end{aligned} \quad (\text{Lem8.2.3})$$

This can be written as

$$\begin{aligned} & \left[\frac{1}{T_o} \int_0^{T_o} \frac{\partial f}{\partial v}(\varphi_H, t) dt \quad \frac{1}{T_o} \int_0^{T_o} \frac{\partial f}{\partial v}(\varphi_H, t) \cos(\omega_o t) dt \quad \cdots \quad - \frac{1}{T_o} \int_0^{T_o} \frac{\partial f}{\partial v}(\varphi_H, t) \sin((H-1)\omega_o t) dt \right] \\ & \begin{bmatrix} Y^R[0] \\ Y^R[\omega_o] \\ Y^I[\omega_o] \\ \vdots \\ Y^R[(H-1)\omega_o] \\ Y^I[(H-1)\omega_o] \end{bmatrix} = Z^R[0] \end{aligned} \quad (\text{Lem8.2.4})$$

This is equivalent to the following:

$$\begin{aligned} & \left[\frac{\partial F^R[0]}{\partial V^R[0]} \quad \frac{\partial F^R[0]}{\partial V^R[\omega_o]} \quad \cdots \quad \frac{\partial F^R[0]}{\partial V^I[(H-1)\omega_o]} \right] \begin{bmatrix} Y^R[0] \\ Y^R[\omega_o] \\ Y^I[\omega_o] \\ \vdots \\ Y^R[(H-1)\omega_o] \\ Y^I[(H-1)\omega_o] \end{bmatrix} = Z^R[0] \\ & \text{or } [\text{the first row of } J_H(\hat{V}_H)] \begin{bmatrix} Y^R[0] \\ Y^R[\omega_o] \\ Y^I[\omega_o] \\ \vdots \\ Y^R[(H-1)\omega_o] \\ Y^I[(H-1)\omega_o] \end{bmatrix} = Z^R[0] \end{aligned}$$

Let's define $f' = \frac{\partial f}{\partial v}(\varphi, t)$. The $\cos(k\omega_o t)$ term of Eqn. (Lem8.2.2) for $k = 1, \dots, H-1$ is:

$$\begin{aligned} & \frac{1}{T_o} \int_0^{T_o} \left[Y^R [0] + \sum_{k=1}^{H-1} Y^R [k \omega_o] \cos(k \omega_o t) - Y^I [k \omega_o] \sin(k \omega_o t) \right] \cos(k \omega_o t) dt \\ & = Z^R [k \omega_o] \end{aligned} \quad (\text{Lem8.2.5})$$

Again, we can write the left hand side as:

$$\begin{aligned} & \left[\frac{1}{T_o} \int_0^{T_o} f \cos(k \omega_o t) dt \quad \frac{1}{T_o} \int_0^{T_o} f \cos(\omega_o t) \cos(k \omega_o t) dt \quad \dots \quad - \frac{1}{T_o} \int_0^{T_o} f \sin((H-1)\omega_o t) \cos(k \omega_o t) dt \right] \\ & \begin{bmatrix} Y^R [0] \\ Y^R [\omega_o] \\ Y^I [\omega_o] \\ \vdots \\ Y^R [(H-1)\omega_o] \\ Y^I [(H-1)\omega_o] \end{bmatrix} = Z^R [k \omega_o] \end{aligned} \quad (\text{Lem8.2.6})$$

This is equivalent to:

$$\begin{aligned} & \left[\frac{\partial F^R [k \omega_o]}{\partial V^R [0]} \quad \frac{\partial F^R [k \omega_o]}{\partial V^R [\omega_o]} \quad \dots \quad \frac{\partial F^R [k \omega_o]}{\partial V^I [(H-1)\omega_o]} \right] \begin{bmatrix} Y^R [0] \\ Y^R [\omega_o] \\ Y^I [\omega_o] \\ \vdots \\ Y^R [(H-1)\omega_o] \\ Y^I [(H-1)\omega_o] \end{bmatrix} = Z^R [k \omega_o] \\ & \text{or } \left[\text{the } (2k-1)\text{th row of } J_H(\hat{V}_H) \right] \begin{bmatrix} Y^R [0] \\ Y^R [\omega_o] \\ Y^I [\omega_o] \\ \vdots \\ Y^R [(H-1)\omega_o] \\ Y^I [(H-1)\omega_o] \end{bmatrix} = Z^R [k \omega_o] \end{aligned}$$

Similar results can be obtained for any $\sin(k \omega_o t)$ term for $k = 1, \dots, H-1$. Hence, Eqn. (Lem8.2.1) is equivalent to Eqn.(Lem8.2.2).

Now, we define the following operators:

$$W^1 y_H(t) \equiv \frac{\partial f}{\partial v}(\hat{v}_H, t) y_H(t) \quad (\text{Lem8.2.7})$$

$$W^2 y_H(t) \equiv P_H \left[\frac{\partial f}{\partial v}(\hat{v}_H, t) y_H(t) \right] \quad (\text{Lem8.2.8})$$

The norm of the operator $\|W^1 - W^2\|$ is given by:

$$\|W^1 - W^2\| = \max_{\|y_H\|} \frac{\|W^1 y_H - W^2 y_H\|}{\|y_H\|} \quad (\text{Lem8.2.9})$$

$$\begin{aligned} \|W^1 y_H - W^2 y_H\| &= \left\| \frac{\partial f}{\partial v}(\vartheta_H, t) y_H(t) - P_H \left[\frac{\partial f}{\partial v}(\vartheta_H, t) y_H(t) \right] \right\| \\ &\leq \left\| \frac{\partial f}{\partial v}(\vartheta_H, t) - P_H \frac{\partial f}{\partial v}(\vartheta_H, t) \right\| \|y_H(t)\| \end{aligned} \quad (\text{Lem8.2.10})$$

Theorem C and Assumption 1 imply $\left\| \frac{\partial f}{\partial v}(\vartheta_H, t) - P_H \frac{\partial f}{\partial v}(\vartheta_H, t) \right\|$ decreases faster than $\frac{1}{H^2}$ as $H \rightarrow \infty$. Therefore, $\|W^1 - W^2\|$ and $|\det W^1 - \det W^2|$ can be arbitrarily small for sufficient large H .

By Assumption 3, there exists a small number δ such that for all $v \in B = \{v \mid \|v - \vartheta\| \leq \delta\}$, $\frac{\partial f}{\partial v}(v)$ is nonsingular. Also from Lemma 8.1, we know that for any δ , there exists an H' such that $\|\vartheta(t) - \vartheta_{H'}(t)\| < \delta$. Hence $W^1 = \frac{\partial f}{\partial v}(\vartheta_H, t)$ is nonsingular for $H \geq H'$. There exists an H'' such that $|\det W^1 - \det W^2| < |\det W^1|$ for $H \geq H''$. Let $H_o = \max(H', H'', H_{crit})$, then W^2 is nonsingular for all $H \geq H_o$. This means that $P_H \left[\frac{\partial f}{\partial v}(\vartheta_H, t) y_H(t) \right] = z_H(t)$ is a one-to-one function for $H \geq H_o$. (i.e. $z_H(t) = 0$ implies $y_H(t) = 0$)

Since Eqn. (Lem8.2.1) is equivalent to Eqn. (Lem8.2.2), the following statements hold:

$$z_H(t) = 0 \text{ is equivalent to } Z_H = 0$$

$$y_H(t) = 0 \text{ is equivalent to } Y_H = 0$$

Hence, $Z_H = 0$ implies $Y_H = 0$. Therefore $J_H(\hat{V}_H)$ has a null space of $\{0\}$, or $J_H(\hat{V}_H)$ is nonsingular for $H \geq H_o$. Thus, there exists an $\beta > 0$ such that $\|J_H(\hat{V}_H)^{-1}\| \leq \beta$ for $H \geq H_o$.

Q.E.D.

Lemma 8.3:

Let Assumptions 1, 2, and 3 be satisfied. Let

$$v_H^a(t) = V^{aR} [0] + \sum_{k=1}^{H-1} \left[V^{aR} [k \omega_o] \cos(k \omega_o t) - V^{aI} [k \omega_o] \sin(k \omega_o t) \right]$$

and

$$\mathbf{v}_H^b(t) = \mathbf{V}^{b^R}[0] + \sum_{k=1}^{H-1} \left[\mathbf{V}^{b^R}[k\omega_o] \cos(k\omega_o t) - \mathbf{V}^{b^I}[k\omega_o] \sin(k\omega_o t) \right]$$

be any two voltage vectors in \mathbf{D} , then

$$\|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \leq C_2 \|\mathbf{V}_H^a - \mathbf{V}_H^b\|$$

where C_2 is a positive constant such that $C_2 \geq \max_{\mathbf{D} \times \mathcal{L}} \left\| \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{v}, t) \right\|$.

Proof:

By Theorem B:

$$\begin{aligned} & \|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \\ & \leq \max_{0 \leq \tau \leq 1} \left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{V}_H^b + \tau(\mathbf{V}_H^a - \mathbf{V}_H^b)) \right\| \|\mathbf{V}_H^a - \mathbf{V}_H^b\| \end{aligned} \quad (\text{Lem8.3.1})$$

Choose $\mathbf{E}_H \equiv (\mathbf{V}_H^b + \tau(\mathbf{V}_H^a - \mathbf{V}_H^b))$ such that

$$\left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \right\| = \max_{0 \leq \tau \leq 1} \left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{V}_H^b + \tau(\mathbf{V}_H^a - \mathbf{V}_H^b)) \right\| \quad (\text{Lem8.3.2})$$

Hence

$$\|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \leq \left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \right\| \|\mathbf{V}_H^a - \mathbf{V}_H^b\| \quad (\text{Lem8.3.3})$$

We define the following functions $\mathbf{a}_H(t)$, $\mathbf{b}_H(t)$, and $\mathbf{c}_H(t)$ as:

$$\begin{aligned} \mathbf{a}_H(t) & \equiv \mathbf{A}^R[0] + \sum_{k=1}^{H-1} \left[\mathbf{A}^R[k\omega_o] \cos(k\omega_o t) - \mathbf{A}^I[k\omega_o] \sin(k\omega_o t) \right] \\ \mathbf{b}_H(t) & \equiv \mathbf{B}^R[0] + \sum_{k=1}^{H-1} \left[\mathbf{B}^R[k\omega_o] \cos(k\omega_o t) - \mathbf{B}^I[k\omega_o] \sin(k\omega_o t) \right] \\ \mathbf{c}_H(t) & \equiv \mathbf{C}^R[0] + \sum_{k=1}^{H-1} \left[\mathbf{C}^R[k\omega_o] \cos(k\omega_o t) - \mathbf{C}^I[k\omega_o] \sin(k\omega_o t) \right] \end{aligned}$$

where $\mathbf{A}^R[0]$, $\mathbf{A}^R[k\omega_o]$, $\mathbf{A}^I[k\omega_o]$, $\mathbf{B}^R[0]$, $\mathbf{B}^R[k\omega_o]$, $\mathbf{B}^I[k\omega_o]$, $\mathbf{C}^R[0]$, $\mathbf{C}^R[k\omega_o]$, $\mathbf{C}^I[k\omega_o] \in R^n$ for $k = 1, \dots, H-1$.

Consider the following system:

$$\left\langle \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \mathbf{A}_H, \mathbf{B}_H \right\rangle = \mathbf{C}_H \quad (\text{Lem8.3.4})$$

We want to prove that the system described by the equation above is equivalent to the following system:

$$P_H \left[\left\langle \frac{\partial^2 f}{\partial v^2}(e_H) a_H(t), b_H(t) \right\rangle \right] = c_H(t) \tag{Lem8.3.5}$$

where $e_H(t) \equiv E^R[0] + \sum_{k=1}^{H-1} \left[E^R[k\omega_o] \cos(k\omega_o t) - E^I[k\omega_o] \sin(k\omega_o t) \right]$.

Define $f'' \equiv \frac{\partial^2 f}{\partial v^2}(e_H, t)$, then the DC term of Eqn. (Lem8.3.5) can be written as:

$$\frac{1}{T_o} \int_0^{T_o} \left\langle f'' \left[A^R[0] + \sum_{k=1}^{H-1} \left[A^R[k\omega_o] \cos(k\omega_o t) - A^I[k\omega_o] \sin(k\omega_o t) \right] \right], \right. \\ \left. \left[B^R[0] + \sum_{k=1}^{H-1} \left[B^R[k\omega_o] \cos(k\omega_o t) - B^I[k\omega_o] \sin(k\omega_o t) \right] \right] \right\rangle dt = C^R[0] \tag{Lem8.3.6}$$

or

$$\left[\begin{array}{c} \frac{1}{T_o} \int_0^{T_o} f'' dt \\ \frac{1}{T_o} \int_0^{T_o} f'' \cos(\omega_o t) dt \\ - \frac{1}{T_o} \int_0^{T_o} f'' \sin(\omega_o t) dt \\ \dots \\ \frac{1}{T_o} \int_0^{T_o} f'' \cos((H-1)\omega_o t) dt \\ - \frac{1}{T_o} \int_0^{T_o} f'' \sin((H-1)\omega_o t) dt \end{array} \right] \begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \left[\begin{array}{c} - \frac{1}{T_o} \int_0^{T_o} f'' \sin((H-1)\omega_o t) dt \\ - \frac{1}{T_o} \int_0^{T_o} f'' \sin((H-1)\omega_o t) \cos(\omega_o t) dt \\ \frac{1}{T_o} \int_0^{T_o} f'' \sin((H-1)\omega_o t) \sin(\omega_o t) dt \\ \dots \\ - \frac{1}{T_o} \int_0^{T_o} f'' \sin((H-1)\omega_o t) \cos((H-1)\omega_o t) dt \\ \frac{1}{T_o} \int_0^{T_o} f'' \sin((H-1)\omega_o t) \sin((H-1)\omega_o t) dt \end{array} \right] \\ \left[\begin{array}{c} A^R[0] \\ A^R[\omega_o] \\ A^I[\omega_o] \\ \vdots \\ A^R[(H-1)\omega_o] \\ A^I[(H-1)\omega_o] \end{array} \right]^T \left[\begin{array}{c} B^R[0] \\ B^R[\omega_o] \\ B^I[\omega_o] \\ \vdots \\ B^R[(H-1)\omega_o] \\ B^I[(H-1)\omega_o] \end{array} \right] = C^R[0]$$

This is equivalent to:

$$\begin{array}{c}
 \left[\begin{array}{ccc}
 \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [0] \partial \mathbf{V}^R [0]} & & \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [(H-1)\omega_o] \partial \mathbf{V}^R [0]} \\
 \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [0] \partial \mathbf{V}^R [\omega_o]} & \dots\dots & \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [(H-1)\omega_o] \partial \mathbf{V}^R [\omega_o]} \\
 \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [0] \partial \mathbf{V}^I [\omega_o]} & \dots\dots & \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [(H-1)\omega_o] \partial \mathbf{V}^I [\omega_o]} \\
 \vdots & \dots\dots & \vdots \\
 \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [0] \partial \mathbf{V}^I [(H-1)\omega_o]} & \dots\dots & \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [(H-1)\omega_o] \partial \mathbf{V}^I [(H-1)\omega_o]} \\
 \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [0] \partial \mathbf{V}^I [(H-1)\omega_o]} & & \frac{\partial^2 \mathbf{F}^R [0]}{\partial \mathbf{V}^R [(H-1)\omega_o] \partial \mathbf{V}^I [(H-1)\omega_o]}
 \end{array} \right] \\
 \\
 \left[\begin{array}{c}
 \mathbf{A}^R [0] \\
 \mathbf{A}^R [\omega_o] \\
 \mathbf{A}^I [\omega_o] \\
 \vdots \\
 \mathbf{A}^R [(H-1)\omega_o] \\
 \mathbf{A}^I [(H-1)\omega_o]
 \end{array} \right]^T \cdot \left[\begin{array}{c}
 \mathbf{B}^R [0] \\
 \mathbf{B}^R [\omega_o] \\
 \mathbf{B}^I [\omega_o] \\
 \vdots \\
 \mathbf{B}^R [(H-1)\omega_o] \\
 \mathbf{B}^I [(H-1)\omega_o]
 \end{array} \right] = \mathbf{C}^R [0]
 \end{array}$$

or $\langle [\text{the first row block of } \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}] \mathbf{A}_H, \mathbf{B}_H \rangle = \mathbf{C}^R [0]$ (Lem8.3.7)

Similar results can be obtained for all other terms associated with Eqn.(Lem8.3.5). Hence, we can conclude that Eqn. (Lem8.3.4) is equivalent to Eqn. (Lem8.3.5).

Using Parseval's equation as in Lemma 8.1, we can claim that $\|\mathbf{c}_H(t)\| = \|\mathbf{C}_H\|$. Therefore,

$$\begin{aligned}
 \left\| \left\langle \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \mathbf{A}_H, \mathbf{B}_H \right\rangle \right\| &= \left\| \left\| P_H \left[\left\langle \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{e}_H, t) \mathbf{a}_H(t), \mathbf{b}_H(t) \right\rangle \right] \right\| \right\| \\
 &\leq \left\| \left\langle \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{e}_H, t) \mathbf{a}_H(t), \mathbf{b}_H(t) \right\rangle \right\|
 \end{aligned}$$

This implies that

$$\left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \right\| \leq \left\| \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{e}_H, t) \right\| \leq C_2$$

Applying this result to Eqn. (Lem8.3.3), we have the following inequality:

$$\|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \leq C_2 \|\mathbf{V}_H^a - \mathbf{V}_H^b\|$$

Q.E.D.

Theorem 8.1:

Assume that Assumptions 1, 2, and 3 are satisfied. Let \mathbf{V}_H^* be the solution of $\mathbf{F}_H(\mathbf{V}_H) = 0$ obtained by the harmonic-Newton method, then $\mathbf{v}_H^*(t)$ converges to $\hat{\mathbf{v}}(t)$ as $H \rightarrow \infty$.

Proof:

Let $\hat{\mathbf{v}}_H(t)$, the truncated exact solution, be the initial guess vector for the harmonic-Newton method. From Lemma 8.1, we know that the following inequality holds for all $H \geq H_{crit}$ such that $\hat{\mathbf{v}}_H(t) \in \mathbf{D}$,

$$\|\mathbf{F}_H(\hat{\mathbf{V}}_H)\| \leq \frac{1}{\omega_o} \frac{1}{H} C_1 C \quad (8.2.1)$$

where $C_1 \equiv \max_{\mathbf{D} \times \mathcal{L}} \|\frac{\partial \mathbf{f}}{\partial \mathbf{v}}(\mathbf{v}, t)\|$ and $C \equiv \|\hat{\mathbf{v}}\|$.

From Lemma 8.2,

$$\|\mathbf{J}_H(\hat{\mathbf{V}}_H)^{-1}\| \leq \beta \quad (8.2.2)$$

for $H \geq H_o$. Eqn. (8.2.1) and Eqn. (8.2.2) imply that

$$\|\mathbf{J}_H(\hat{\mathbf{V}}_H)^{-1} \mathbf{F}_H(\hat{\mathbf{V}}_H)\| \leq \frac{1}{\omega_o} \frac{1}{H} C_1 C \beta \equiv \eta \quad (8.2.3)$$

Also from Lemma 8.3, we know that for any $\mathbf{v}_H^a(t)$ and $\mathbf{v}_H^b(t) \in \mathbf{D}$, we have

$$\|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \leq \gamma \|\mathbf{V}_H^a - \mathbf{V}_H^b\|, \quad (8.2.4)$$

where $\gamma \equiv C_2 \geq \max_{\mathbf{D} \times \mathcal{L}} \|\frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{v}, t)\|$.

Let $\alpha \equiv \beta \gamma \eta$, then there exists an $\hat{H} \geq H_o$ such that $\alpha \leq \frac{1}{2}$ for all $H \geq \hat{H}$. Set

$t^* = (\beta \gamma)^{-1} \left[1 - (1 - 2\alpha)^{\frac{1}{2}} \right]$. There exists an $H^* \geq \hat{H}$ such that $\bar{\mathbf{B}}(\hat{\mathbf{v}}_H, t^*)$ lies inside of \mathbf{D} . By the

Newton-Kantorovich Theorem, we know that the Newton iteration is well-defined, and the sequence of solutions remains in $\bar{\mathbf{B}}(\hat{\mathbf{v}}_H, t^*)$ and converges to the solution of $\mathbf{F}_H(\mathbf{V}_H^*) = 0$ which is unique in

$\bar{\mathbf{B}}(\hat{\mathbf{v}}_H, t^{**})$ for all $H \geq H^*$, where $t^{**} \equiv (\beta \gamma)^{-1} \left[1 + (1 - 2\alpha)^{\frac{1}{2}} \right]$. Moreover, the error bound $\|\mathbf{V}_H^* - \hat{\mathbf{V}}_H\|$ satisfies the following inequality:

$$\|\mathbf{V}_H^* - \hat{\mathbf{V}}_H\| = \|\mathbf{v}_H^*(t) - \hat{\mathbf{v}}_H(t)\| \leq (\beta \gamma)^{-1} (2\alpha)$$

Thus, we have

$$\begin{aligned}
 \|\mathbf{v}_H^*(t) - \hat{\mathbf{v}}(t)\| &\leq \|\mathbf{v}_H^*(t) - \hat{\mathbf{v}}_H(t)\| + \|\hat{\mathbf{v}}_H(t) - \hat{\mathbf{v}}(t)\| \\
 &\leq (\beta\gamma)^{-1}(2\alpha) + \frac{1}{\omega_o} \frac{1}{H} C \\
 &= 2\eta + \frac{1}{\omega_o} \frac{1}{H} C \\
 &= \frac{2}{\omega_o} \frac{1}{H} C C_1 \beta + \frac{1}{\omega_o} \frac{1}{H} C
 \end{aligned} \tag{8.2.5}$$

Therefore, $\mathbf{v}_H^*(t) \rightarrow \hat{\mathbf{v}}(t)$ as $H \rightarrow \infty$.

Q.E.D.

8.3. Almost-Periodic Nonautonomous Systems

Let $\mathbf{f}(\mathbf{v}, t) = 0$ be a given real nonlinear system where $\mathbf{v}(t)$ and $\mathbf{f}(\mathbf{v}, t)$ are real vectors of the same dimension n . Let $\hat{\mathbf{v}}(t) = \sum_{\omega_k \in \Lambda} \hat{\mathbf{V}}^R[\omega_k] \cos(\omega_k t) - \hat{\mathbf{V}}^I[\omega_k] \sin(\omega_k t)$ be the isolated periodic solution of $\mathbf{f}(\mathbf{v}, t) = 0$, where $\hat{\mathbf{V}}^R[\omega_k]$, $\hat{\mathbf{V}}^I[\omega_k]$ is a n -tuple vector for all $\omega_k \in \Lambda$.

We assume:

- (1) $\mathbf{f}(\mathbf{v}, t)$ and its partial derivatives with respect to \mathbf{v} are continuously differentiable with respect to \mathbf{v} and t in the region $D \times L$, where D is a convex bounded region of \mathbf{v} space and L is the real line.
- (2) $\mathbf{f}(\mathbf{v}, t) = 0$ is an almost-periodic nonautonomous system.
- (3) If $\hat{\mathbf{v}}(t)$ is an exact solution of $\mathbf{f}(\mathbf{v}, t) = 0$ and lies in the interior of D , then $\hat{\mathbf{v}}(t)$ is an isolated periodic solution. (i.e. there exists a small number δ such that for all $\mathbf{v} \in B = \{\mathbf{v} \mid \|\mathbf{v} - \hat{\mathbf{v}}\| \leq \delta\}$, $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}(\mathbf{v}, t)$ is nonsingular.)
- (4) Define $s_H \equiv \sum_{\omega_k \in \Lambda'_H} \hat{\mathbf{V}}^R[\omega_k] \cos(\omega_k t) - \hat{\mathbf{V}}^I[\omega_k] \sin(\omega_k t)$, where

$$\Lambda'_H \equiv \left\{ \omega \mid \omega = k_1 \lambda_1 + k_2 \lambda_2 + \dots + k_d \lambda_d; k_j \in \mathbf{Z}; |k_j| = H \text{ for } 1 \leq j \leq d; k_1 \geq 0 \right\}$$

or

$$\Lambda'_H \equiv \left\{ \omega \mid \omega = k_1 \lambda_1 + k_2 \lambda_2 + \dots + k_d \lambda_d; k_j \in \mathbf{Z} \text{ for } 1 \leq j \leq d; \sum_{j=1}^d |k_j| = H; k_1 \geq 0 \right\}$$

depending on which truncation method is used, then the series $\{ |s_H| \}$ must decrease faster than the series $\{ \frac{1}{H^2} \}$ as $H \rightarrow \infty$.

These assumptions will be referred as Assumption 1 - 4 in this section.

Lemma 8.4:

Let Assumptions 1, 2, 3, and 4 be satisfied, then

$$\hat{v}_H(t) \equiv \sum_{\omega_k \in \Lambda_H} \hat{V}^R[\omega_k] \cos(\omega_k t) - \hat{V}^I[\omega_k] \sin(\omega_k t)$$

is an approximate solution of $f_H(v_H, t) = P'_H f(v_H, t) = 0$ (i.e., $f_H(\hat{v}_H, t) \rightarrow 0$ as $H \rightarrow \infty$).

Proof:

Let $C_1 \equiv \max_{D \times \mathcal{L}} \left\| \frac{\partial f}{\partial v}(v, t) \right\|$, then by Theorem A:

$$\begin{aligned} \|f(\hat{v}_H, t) - f(\hat{v}, t)\| &\leq \max_{D \times \mathcal{L}} \left\| \frac{\partial f}{\partial v}(v, t) \right\| \|\hat{v}_H - \hat{v}\| \\ &\leq C_1 \|\hat{v}_H - \hat{v}\| \end{aligned} \tag{Lem8.4.1}$$

For any given $\varepsilon > 0$, there exists a number H_{crit} large enough such that v_H lies inside of D and

$$\|\hat{v}(t) - \hat{v}_H(t)\| = \left\| \sum_{\omega_k \in \Lambda - \Lambda_H} \hat{V}^R[\omega_k] \cos(\omega_k t) - \hat{V}^I[\omega_k] \sin(\omega_k t) \right\| < \varepsilon \tag{Lem8.4.2}$$

for all $H > H_{crit}$. Since the basis is orthogonal,

$$\|f_H(\hat{v}_H, t)\| \leq \|f(\hat{v}_H, t) - f(\hat{v}, t)\| < C_1 \varepsilon \quad \text{for all } H > H_{crit} \tag{Lem8.4.3}$$

Therefore,

$$f_H(\hat{v}_H, t) \rightarrow 0 \text{ as } H \rightarrow \infty. \tag{Lem8.4.4}$$

By Parseval's equation:

$$\|F_H(\hat{V}_H)\| = \|f_H(\hat{v}_H, t)\| \rightarrow 0 \text{ as } H \rightarrow \infty. \tag{Lem8.4.5}$$

Q.E.D.

Lemma 8.5 :

Let the conditions stated in Assumptions 1, 2, 3, and 4 be satisfied, then there exists a positive integer H_o such that $\det J_H(\hat{V}_H) \neq 0$ (or there exist some $\beta > 0$ such that $\|J_H^{-1}(\hat{V}_H)\| \leq \beta$) for any $H \geq H_o$ provided that H_o is sufficiently large.

Proof:

Let us consider the following linear system:

$$J_H(\hat{V}_H)Y_H = Z_H \quad (\text{Lem8.5.1})$$

$$\text{where } y_H(t) \equiv \sum_{\omega_k \in \Lambda_H} Y^R[\omega_k] \cos(\omega_k t) - Y^I[\omega_k] \sin(\omega_k t)$$

$$z_H(t) \equiv \sum_{\omega_k \in \Lambda_H} Z^R[\omega_k] \cos(\omega_k t) - Z^I[\omega_k] \sin(\omega_k t)$$

and $Y^R[\omega_k]$, $Y^I[\omega_k]$, $Z^R[\omega_k]$, and $Z^I[\omega_k] \in R^n$.

Using similar method as in Lemma 8.2, we can conclude that the system described by Eqn. (Lem8.5.1) is equivalent to the following system:

$$P'_H \left[\frac{\partial f}{\partial v}(\hat{v}_H, t) y_H(t) \right] = z_H(t) \quad (\text{Lem8.5.2})$$

In addition, $P'_H \left[\frac{\partial f}{\partial v}(\hat{v}_H, t) y_H(t) \right] = z_H(t)$ is a one-to-one function for $H \geq H_o$. (i.e. $z_H(t) = 0$ implies $y_H(t) = 0$)

Since Eqn. (Lem8.5.1) is equivalent to Eqn. (Lem8.5.2), $J_H(\hat{V}_H)$ has a null space of $\{ 0 \}$ or $J_H(\hat{V}_H)$ is nonsingular for $H \geq H_o$. Therefore, there exists a number $\beta > 0$ such that $\|J_H(\hat{V}_H)^{-1}\| \leq \beta$ for $H \geq H_o$.

Q.E.D.

Lemma 8.6:

Let the conditions given in Assumptions 1, 2, 3, and 4 be satisfied. Let

$$v_H^a(t) \equiv \sum_{\omega_k \in \Lambda_H} V^{a^R}[\omega_k] \cos(\omega_k t) - V^{a^I}[\omega_k] \sin(\omega_k t)$$

and

$$\mathbf{v}_H^b(t) \equiv \sum_{\omega_k \in \Lambda_H} \mathbf{V}^{bR}[\omega_k] \cos(\omega_k t) - \mathbf{V}^{bI}[\omega_k] \sin(\omega_k t)$$

be any two voltage vectors lying inside of \mathbf{D} , then

$$\|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \leq C_2 \|\mathbf{V}_H^a - \mathbf{V}_H^b\|$$

where C_2 is a positive constant such that $C_2 \geq \max_{\mathbf{D} \times \mathcal{L}} \left\| \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{v}, t) \right\|$.

Proof:

Choose $\mathbf{E}_H \equiv (\mathbf{V}_H^b + \tau(\mathbf{V}_H^a - \mathbf{V}_H^b))$ such that

$$\left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \right\| = \max_{0 \leq \tau \leq 1} \left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{V}_H^b + \tau(\mathbf{V}_H^a - \mathbf{V}_H^b)) \right\| \quad (\text{Lem8.6.1})$$

Hence,

$$\|\mathbf{J}_H(\mathbf{V}_H^a) - \mathbf{J}_H(\mathbf{V}_H^b)\| \leq \left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \right\| \cdot \|\mathbf{V}_H^a - \mathbf{V}_H^b\| \quad (\text{Lem8.6.2})$$

$$\text{Let } \mathbf{a}_H(t) \equiv \sum_{\omega_k \in \Lambda_H} \mathbf{A}^R[\omega_k] \cos(\omega_k t) - \mathbf{A}^I[\omega_k] \sin(\omega_k t) \quad \mathbf{A}^R[\omega_k], \mathbf{A}^I[\omega_k] \in R^n$$

$$\mathbf{b}_H(t) \equiv \sum_{\omega_k \in \Lambda_H} \mathbf{B}^R[\omega_k] \cos(\omega_k t) - \mathbf{B}^I[\omega_k] \sin(\omega_k t) \quad \mathbf{B}^R[\omega_k], \mathbf{B}^I[\omega_k] \in R^n$$

$$\mathbf{c}_H(t) \equiv \sum_{\omega_k \in \Lambda_H} \mathbf{C}^R[\omega_k] \cos(\omega_k t) - \mathbf{C}^I[\omega_k] \sin(\omega_k t) \quad \mathbf{C}^R[\omega_k], \mathbf{C}^I[\omega_k] \in R^n$$

Using the same argument as in Lemma 8.3, we can conclude that the following two systems are equivalent.

$$\left\langle \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \mathbf{A}_H, \mathbf{B}_H \right\rangle = \mathbf{C}_H \quad (\text{Lem8.6.3})$$

$$P'_H \left[\left\langle \frac{\partial^2 \mathbf{f}_H}{\partial \mathbf{v}^2}(\mathbf{e}_H, t) \mathbf{a}_H(t), \mathbf{b}_H(t) \right\rangle \right] = \mathbf{c}_H(t) \quad (\text{Lem8.6.4})$$

Using the Parseval's equation for almost-periodic function, we can claim that $\|\mathbf{c}_H(t)\| = \|\mathbf{C}_H\|$,

Therefore,

$$\left\| \left\langle \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \mathbf{A}_H, \mathbf{B}_H \right\rangle \right\| \leq \left\| \left\langle \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{e}_H, t) \mathbf{a}_H(t), \mathbf{b}_H(t) \right\rangle \right\|$$

This implies that $\left\| \frac{\partial \mathbf{J}_H}{\partial \mathbf{V}_H}(\mathbf{E}_H) \right\| \leq \left\| \frac{\partial^2 \mathbf{f}}{\partial \mathbf{v}^2}(\mathbf{e}_H, t) \right\| \leq C_2$

Applying this result into Eqn. (Lem8.6.3), we can conclude:

$$\|J_H(\mathbf{V}_H^a) - J_H(\mathbf{V}_H^b)\| \leq C_2 \|\mathbf{V}_H^a - \mathbf{V}_H^b\|$$

Q.E.D.

Theorem 8.2:

Let the conditions stated in Assumptions 1, 2, 3, and 4 be satisfied. Let \mathbf{v}_H^* be the solution of $\mathbf{f}_H(\mathbf{v}_H^*, t) = 0$ obtained by the harmonic-Newton method, then $\mathbf{v}_H^*(t)$ converges to $\hat{\mathbf{v}}(t)$ as $H \rightarrow \infty$.

Proof:

Following the same argument as used in Theorem 8.1 and by Lemma 8.4, 8.5, and 8.6, we can show that there exist a H^* such that the Newton iteration is well-defined and converges to a solution of $\mathbf{F}_H(\mathbf{V}_H^*) = 0$ for all $H \geq H^*$.

Moreover, for any ε , there is an H such that

$$\|\mathbf{v}_H^*(t) - \hat{\mathbf{v}}(t)\| \leq 2\beta C_1 \varepsilon + \varepsilon$$

Therefore, $\mathbf{v}_H^*(t) \rightarrow \hat{\mathbf{v}}(t)$, as $H \rightarrow \infty$.

Q.E.D.

8.4. Periodic Autonomous System

As described in Chapter 7, in order to solve the autonomous system, we need to add an additional constraint. Thus, the system of equations become:

$$\bar{\mathbf{f}}(\mathbf{x}, t) \equiv \left\{ \begin{array}{l} \bar{\mathbf{g}}(\mathbf{x}, t) \equiv \mathbf{i}(\mathbf{v}(t)) + \dot{\mathbf{q}}(\mathbf{v}(t)) + \int_0^t \mathbf{y}(t-\tau) \mathbf{v}(\tau) d\tau \\ \frac{1}{T} \int_0^T \mathbf{v}^i(t) \sin(k_o \omega t) dt \end{array} \right\} = 0 \quad (8.4.1)$$

where $\mathbf{x} \equiv [\mathbf{v}, \omega]^T$.

The frequency domain equivalent system is:

$$\bar{\mathbf{F}}(\mathbf{X}) \equiv \left\{ \begin{array}{c} \bar{\mathbf{G}}(\mathbf{V}, \omega) \equiv \mathbf{I}(\mathbf{V}) + j\Omega(\omega)\mathbf{Q}(\mathbf{V}) + \mathbf{YV} \\ V^{i_0} [k_0 \omega] \end{array} \right\} = 0 \quad (8.4.2)$$

where $\mathbf{X} \equiv [\mathbf{V}, \omega]^T$.

We make the following assumptions before continuing the discussion: We assume:

- (1) $\bar{\mathbf{f}}(\mathbf{x}, t)$ and its partial derivatives with respect to \mathbf{x} are continuously differentiable with respect to \mathbf{x} and t in the region $\bar{\mathbf{D}} \times \mathbf{L}$, where $\bar{\mathbf{D}}$ is a convex bounded region of \mathbf{x} -space and \mathbf{L} is the real line.
- (2) $\bar{\mathbf{f}}(\mathbf{x}, t) = 0$ is a periodic autonomous system.
- (3) There exist an isolated periodic solution, $\hat{\mathbf{x}}(t)$. (i.e. there exists a small number δ such that for all $\mathbf{x} \in \mathbf{B} = \{\mathbf{x} \mid \|\mathbf{x} - \hat{\mathbf{x}}\| \leq \delta\}$, $\frac{\partial \bar{\mathbf{f}}}{\partial \mathbf{x}}(\mathbf{x}, t)$ is nonsingular)

These assumptions will be referred as Assumption 1, 2, and 3, in this section.

Lemma 8.7:

Let the conditions stated in Assumptions 1, 2, and 3 be satisfied, then

$$\hat{\mathbf{x}}_H(t) \equiv \left[\begin{array}{c} \hat{\mathbf{V}}^R [0] + \sum_{k=1}^{H-1} \left[\hat{\mathbf{V}}^R [k \hat{\omega}] \cos(k \hat{\omega} t) - \hat{\mathbf{V}}^I [k \hat{\omega}] \sin(k \hat{\omega} t) \right] \\ \hat{\omega} \end{array} \right]$$

is an approximate solution of $\bar{\mathbf{f}}_H(\mathbf{x}_H, t) = P_H [\bar{\mathbf{f}}(\mathbf{x}_H, t)] = 0$. (i.e. $\bar{\mathbf{f}}_H(\hat{\mathbf{x}}_H, t) \rightarrow 0$ as $H \rightarrow \infty$).

Proof:

$$\bar{\mathbf{f}}_H(\hat{\mathbf{v}}_H, \hat{\omega}, t) = P_H [\bar{\mathbf{f}}(\hat{\mathbf{v}}_H, \hat{\omega}, t) - \bar{\mathbf{f}}(\hat{\mathbf{v}}, \hat{\omega}, t)] \quad (\text{Lem8.7.1})$$

Note that

$$\| \bar{\mathbf{f}}(\hat{\mathbf{v}}_H, \hat{\omega}, t) - \bar{\mathbf{f}}(\hat{\mathbf{v}}, \hat{\omega}, t) \| = \| \bar{\mathbf{g}}(\hat{\mathbf{v}}_H, \hat{\omega}, t) - \bar{\mathbf{g}}(\hat{\mathbf{v}}, \hat{\omega}, t) \|$$

Let $C_1 \equiv \max_{\mathbf{D} \times \mathbf{L}} \left\| \frac{\partial \bar{\mathbf{g}}}{\partial \mathbf{x}}(\mathbf{x}, t) \right\|$, then by Theorem A:

$$\| \bar{\mathbf{g}}(\hat{\mathbf{x}}_H, t) - \bar{\mathbf{g}}(\hat{\mathbf{x}}, t) \| \leq C_1 \| \hat{\mathbf{v}}_H - \hat{\mathbf{v}} \|$$

Using the arguments similar to those used in Lemma 8.1, we can conclude that:

$$\|\hat{v}(t) - \hat{v}_H(t)\|^2 \leq \frac{1}{\hat{\omega}^2} \frac{1}{H^2} \|\dot{\hat{v}}\|^2 \quad (\text{Lem8.7.2})$$

From Theorem C, $C \equiv \|\dot{\hat{v}}\| < \infty$. Since the basis are orthogonal,

$$\|\tilde{F}_H(\hat{x}_H, t)\| \leq \|\tilde{f}(\hat{x}_H, t) - \tilde{f}(\hat{x}, t)\| \leq \frac{1}{\hat{\omega}} \frac{1}{H} C_1 C$$

Therefore,

$$\tilde{F}_H(\hat{x}_H, t) \rightarrow 0 \text{ as } H \rightarrow \infty. \quad (\text{Lem8.7.3})$$

Let

$$\tilde{f}_H(t)(\hat{x}_H, t) \equiv \left[\begin{array}{c} \tilde{g}_H(\hat{x}_H, t) = G^R[0] + \sum_{k=1}^{H-1} \left[G^R[k\hat{\omega}] \cos(k\hat{\omega}t) - G^I[k\hat{\omega}] \sin(k\hat{\omega}t) \right] \\ 0 \end{array} \right]$$

By Parseval's equation:

$$\|\tilde{g}_H(\hat{x}_H, t)\|^2 \equiv \|\tilde{G}_H(X_H)\|^2 \quad (\text{Lem8.7.4})$$

Therefore,

$$\|\tilde{F}_H(\hat{V}_H)\| = \|\tilde{G}_H(X_H)\| = \|\tilde{g}_H(\hat{x}_H, t)\| = \|\tilde{f}_H(\hat{x}_H, t)\| \leq \frac{1}{\hat{\omega}} \frac{1}{H} C_1 C \quad (\text{Lem8.7.5})$$

Therefore, we conclude that $\tilde{F}_H(\hat{v}_H, t) \rightarrow 0$ as $H \rightarrow \infty$.

Q.E.D.

Lemma 8.8 :

Let the conditions stated in Assumptions 1, 2, and 3 be satisfied, then there exists a positive integer H_o such that $\det \tilde{J}_H(\hat{X}_H) = \det \frac{\partial \tilde{F}_H}{\partial X_H}(\hat{X}_H) \neq 0$ (i.e. $\|\tilde{J}_H^{-1}(\hat{X}_H)\| \leq \bar{\beta}$ where $\bar{\beta} > 0$) for any $H \geq H_o$ provided that H_o is sufficiently large.

Proof:

Let us consider this linear system:

$$\tilde{J}_H(X_H) \tilde{Y}_H = \tilde{Z}_H \quad (\text{Lem8.8.1})$$

$$\text{where } \tilde{Y}_H = [Y^R[0], Y^R[\hat{\omega}], Y^I[\hat{\omega}], \dots, Y^R[(H-1)\hat{\omega}], Y^I[(H-1)\hat{\omega}], \hat{\omega}]^T$$

$$\tilde{Z}_H = [Z^R[0], Z^R[\hat{\omega}], Z^I[\hat{\omega}], \dots, Z^R[(H-1)\hat{\omega}], Z^I[(H-1)\hat{\omega}], 0]^T$$

$$\text{Let } \bar{y}_H(t) \equiv \begin{bmatrix} \mathbf{Y}^R [0] + \sum_{k=1}^{H-1} \left[\mathbf{Y}^R [k \hat{\omega}] \cos(k \hat{\omega} t) - \mathbf{Y}^I [k \hat{\omega}] \sin(k \hat{\omega} t) \right] \\ \hat{\omega} \end{bmatrix}$$

$$\bar{z}_H(t) \equiv \begin{bmatrix} \mathbf{Z}^R [0] + \sum_{k=1}^{H-1} \left[\mathbf{Z}^R [k \hat{\omega}] \cos(k \hat{\omega} t) - \mathbf{Z}^I [k \hat{\omega}] \sin(k \hat{\omega} t) \right] \\ 0 \end{bmatrix}$$

where $\mathbf{Y}^R [0], \mathbf{Y}^R [k \hat{\omega}], \mathbf{Y}^I [k \hat{\omega}], \mathbf{Z}^R [0], \mathbf{Z}^R [k \hat{\omega}], \mathbf{Z}^I [k \hat{\omega}] \in R^n$.

Using similar argument as in Lemma 8.2, we can conclude that the system described by Eqn. (Lem8.8.1) is equivalent to the following system:

$$P_H \begin{bmatrix} \frac{\partial \bar{g}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_H, t) \bar{y}_H(t) \\ \frac{1}{\hat{T}} \int_0^T \frac{\partial v_{i_0}}{\partial \mathbf{x}} \sin(k_0 \hat{\omega} t) dt \cdot \bar{y}_H(t) \end{bmatrix} = \bar{z}_H(t) \quad (\text{Lem8.8.2})$$

In addition,

$$P_H \left[\frac{\partial \bar{F}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_H, t) \bar{y}_H(t) \right] = \bar{z}_H(t)$$

is a one-to-one function for $H \geq H_0$. Since Eqn. (Lem8.8.1) is equivalent to Eqn. (Lem8.8.2), $\bar{J}_H(\hat{\mathbf{x}}_H)$ has a null space of $\{0\}$, or $\bar{J}_H(\hat{\mathbf{x}}_H)$ is nonsingular for $H \geq H_0$.

Therefore, there exists an $\bar{\beta} > 0$ such that $\|\bar{J}_H(\hat{\mathbf{x}}_H)^{-1}\| \leq \bar{\beta}$ for $H \geq H_0$.

Q.E.D.

Lemma 8.9:

Let the conditions given in Assumptions 1, 2, and 3 be satisfied. Let

$$\mathbf{x}_H^a(t) = \begin{bmatrix} \mathbf{X}^{aR} [0] + \sum_{k=1}^{H-1} \left[\mathbf{X}^{aR} [k \omega^a] \cos(k \omega^a t) - \mathbf{X}^{aI} [k \omega^a] \sin(k \omega^a t) \right] \\ \omega^a \end{bmatrix} \in \bar{D}.$$

and

$$\mathbf{x}_H^b(t) = \begin{bmatrix} \mathbf{X}^{bR} [0] + \sum_{k=1}^{H-1} \left[\mathbf{X}^{bR} [k \omega^b] \cos(k \omega^b t) - \mathbf{X}^{bI} [k \omega^b] \sin(k \omega^b t) \right] \\ \omega^b \end{bmatrix} \in \bar{D}.$$

then $\|\bar{J}_H(\mathbf{X}_H^a) - \bar{J}_H(\mathbf{X}_H^b)\| \leq \bar{C}_2 \|\mathbf{X}_H^a - \mathbf{X}_H^b\|$ where \bar{C}_2 be a positive constant such that

$$\bar{C}_2 \geq \max_{\mathbf{D} \times \mathcal{L}} \left\| \frac{\partial^2 \bar{F}}{\partial \mathbf{x}^2}(\mathbf{x}, t) \right\|.$$

Proof: Choose $\tilde{\mathbf{E}}_H \equiv (\mathbf{X}_H^b + \tau(\mathbf{X}_H^a - \mathbf{X}_H^b))$ such that

$$\left\| \frac{\partial \tilde{\mathbf{J}}_H}{\partial \mathbf{X}_H}(\tilde{\mathbf{E}}_H) \right\| = \max_{0 \leq \tau \leq 1} \left\| \frac{\partial \tilde{\mathbf{J}}_H}{\partial \mathbf{X}_H}(\mathbf{X}_H^b + \tau(\mathbf{X}_H^a - \mathbf{X}_H^b)) \right\| \quad (\text{Lem8.9.1})$$

$$\text{Hence, } \left\| \tilde{\mathbf{J}}_H(\mathbf{X}_H^a) - \tilde{\mathbf{J}}_H(\mathbf{X}_H^b) \right\| \leq \left\| \frac{\partial \tilde{\mathbf{J}}_H}{\partial \mathbf{X}_H}(\tilde{\mathbf{E}}_H) \right\| \|\mathbf{X}_H^a - \mathbf{X}_H^b\| \quad (\text{Lem8.9.2})$$

$$\text{Define } \tilde{\mathbf{e}}_H(t) \equiv \begin{bmatrix} \mathbf{E}^R[0] + \sum_{k=1}^{H-1} \left[\mathbf{E}^R[k\omega^\epsilon] \cos(k\omega^\epsilon t) - \mathbf{E}^I[k\omega^\epsilon] \sin(k\omega^\epsilon t) \right] \\ \omega^\epsilon \end{bmatrix}.$$

Let

$$\tilde{\mathbf{a}}_H(t) \equiv \begin{bmatrix} \mathbf{A}^R[0] + \sum_{k=1}^{H-1} \left[\mathbf{A}^R[k\omega^\epsilon] \cos(k\omega^\epsilon t) - \mathbf{A}^I[k\omega^\epsilon] \sin(k\omega^\epsilon t) \right] \\ \omega^\epsilon \end{bmatrix}$$

$$\tilde{\mathbf{b}}_H(t) \equiv \begin{bmatrix} \mathbf{B}^R[0] + \sum_{k=1}^H \left[\mathbf{B}^R[k\omega^\epsilon] \cos(k\omega^\epsilon t) - \mathbf{B}^I[k\omega^\epsilon] \sin(k\omega^\epsilon t) \right] \\ \omega^\epsilon \end{bmatrix}$$

$$\tilde{\mathbf{c}}_H(t) \equiv \begin{bmatrix} \mathbf{C}^R[0] + \sum_{k=1}^{H-1} \left[\mathbf{C}^R[k\omega^\epsilon] \cos(k\omega^\epsilon t) - \mathbf{C}^I[k\omega^\epsilon] \sin(k\omega^\epsilon t) \right] \\ 0 \end{bmatrix}$$

with

$$\tilde{\mathbf{A}}_H \equiv [\mathbf{A}^R[0], \mathbf{A}^R[\omega^\epsilon], \dots, \mathbf{A}^I[(H-1)\omega^\epsilon], \omega^\epsilon]^T$$

$$\tilde{\mathbf{B}}_H \equiv [\mathbf{B}^R[0], \mathbf{B}^R[\omega^\epsilon], \dots, \mathbf{B}^I[(H-1)\omega^\epsilon], \omega^\epsilon]^T$$

$$\tilde{\mathbf{C}}_H \equiv [\mathbf{C}^R[0], \mathbf{C}^R[\omega^\epsilon], \dots, \mathbf{C}^I[(H-1)\omega^\epsilon], 0]^T$$

Using similar arguments as in Lemma 8.3 and Lemma 8.8, we can conclude that the following two systems are equivalent:

$$\left\langle \frac{\partial \tilde{\mathbf{J}}_H}{\partial \mathbf{X}_H}(\tilde{\mathbf{E}}_H) \tilde{\mathbf{A}}_H, \tilde{\mathbf{B}}_H \right\rangle = \tilde{\mathbf{C}}_H \quad (\text{Lem8.9.3})$$

$$P_H \left[\left\langle \frac{\partial^2 \tilde{\mathbf{f}}_H}{\partial \mathbf{x}^2}(\tilde{\mathbf{e}}_H, t) \tilde{\mathbf{a}}_H(t), \tilde{\mathbf{b}}_H(t) \right\rangle \right] = \tilde{\mathbf{c}}_H(t) \quad (\text{Lem8.9.4})$$

Using the technique similar to the one used in Lemma 8.1, we can claim:

$$\left\| \left\langle \frac{\partial \tilde{\mathbf{J}}_H}{\partial \mathbf{X}_H}(\tilde{\mathbf{E}}_H) \tilde{\mathbf{A}}_H, \tilde{\mathbf{B}}_H \right\rangle \right\| \leq \left\| \left\langle \frac{\partial^2 \tilde{\mathbf{f}}_H}{\partial \mathbf{x}^2}(\tilde{\mathbf{e}}_H, t) \tilde{\mathbf{a}}_H(t), \tilde{\mathbf{b}}_H(t) \right\rangle \right\|$$

$$\text{This implies } \left\| \frac{\partial \tilde{\mathbf{J}}_H}{\partial \mathbf{X}_H}(\tilde{\mathbf{E}}_H) \right\| \leq \left\| \frac{\partial^2 \tilde{\mathbf{f}}_H}{\partial \mathbf{x}^2}(\tilde{\mathbf{e}}_H, t) \right\| \leq \tilde{C}_2$$

Therefore, we conclude that:

$$\|\tilde{J}_H(\mathbf{X}_H^a) - \tilde{J}_H(\mathbf{X}_H^b)\| \leq \tilde{C}_2 \|\mathbf{X}_H^a - \mathbf{X}_H^b\|$$

Q.E.D.

Theorem 8.3:

Let the conditions stated in Assumptions 1, 2, and 3 be satisfied. Let \mathbf{V}_H^* and ω^* be the solution of $\tilde{F}_H(\mathbf{V}_H^*, \omega^*) = 0$ obtained by the harmonic-Newton method, then $\mathbf{v}_H^*(t)$ and ω^* converge to $\hat{\mathbf{v}}(t)$ and $\hat{\omega}$ respectively as $H \rightarrow \infty$.

Proof:

Following the same argument as in Theorem 8.1 and by Lemma 8.7, 8.8, and 8.9, we can show that for some H^* , the Newton iteration is well-defined and converge to a solution of $\tilde{F}_H(\mathbf{X}_H^*) = 0$ for $H \geq H^*$.

Moreover,

$$\|\mathbf{X}_H^* - \hat{\mathbf{X}}_H\| \leq \frac{2}{\hat{\omega}} \frac{1}{H} C C_1 \beta + \frac{1}{\hat{\omega}} \frac{1}{H} C$$

Therefore, $\mathbf{X}_H^* \rightarrow \hat{\mathbf{X}}$, or $\mathbf{v}_H^*(t) \rightarrow \hat{\mathbf{v}}(t)$ and $\omega^* \rightarrow \hat{\omega}$ as $H \rightarrow \infty$.

Q.E.D.

CHAPTER 9

Error Estimation and Choosing Number of Harmonics

In the previous chapter, we discussed the convergence of the harmonic-Newton method when higher order harmonics are truncated. In order to implement the harmonic-Newton method effectively in *Spectre*, the discrete Fourier transform (DFT) is used for circuits that have a periodic response. For almost-periodic signals, *Spectre* uses the almost-periodic Fourier transform (APFT)[23]. In Section 9.1, we will discuss how the error bound changes when DFT and APFT are used. In Section 9.2, we will discuss a practical procedure to estimate this error bound and show how we can use this estimation to predict the number of harmonics needed prior to the application of the harmonic-Newton method. Finally, the experience results on error prediction will be presented.

9.1. Theoretical Bound

When transforming a function using DFT, the number of harmonics generated is determined by the number of time-domain samples taken within a given time interval. If the number of samples is not large enough, the set of frequencies generated by DFT will not be able to cover all high order harmonics. This error, although similar to the truncation error, has some different impact. Not only those high harmonic terms are lost, but also the lower order harmonics are affected. In fact, all those high order harmonics which do not show up in the DFT outputs are "aliased" back to affect the low order harmonics. We now briefly discuss the aliasing distortion of a function.

Let F^C and \bar{F}^C be the Fourier series coefficient and discrete Fourier transform coefficient of the periodic function $f(t)$. These coefficients are given by the following expressions:

$$\mathbf{F}^C [k \omega_o] = \frac{1}{T_o} \int_0^{T_o} f(t) e^{-jk \omega_o t} dt \quad \text{for } k = -\infty, \dots, \infty,$$

$$\bar{\mathbf{F}}^C [k \omega_o] = \frac{1}{M} \sum_{m=0}^{M-1} f(mh) e^{-j2\pi km/M} \quad \text{for } k = 1-H, \dots, H-1,$$

where H is the total number of harmonics generated, $M = 2H - 1$ is the number of time-domain samples taken within time interval T_o , and $h = \frac{T_o}{M}$ is the sample interval. The inverse discrete

Fourier transform (IDFT) of $\bar{\mathbf{F}}_H^C$ is given by

$$f(mh) = \sum_{k=1-H}^{H-1} \bar{\mathbf{F}}^C [k \omega_o] e^{j2\pi km/M} \quad \text{for } m = 0, \dots, 2H-1.$$

We define the DFT coefficient vector of H harmonics as

$$\bar{\mathbf{F}}_H^C = [\bar{\mathbf{F}}^C [(1-H)\omega_o], \dots, \mathbf{F}^C [0], \dots, \bar{\mathbf{F}}^C [(H-1)\omega_o]]^T$$

It can be shown that the relationship between the Fourier series and DFT when H is the number of harmonics generated is expressed by the following equation:

$$\bar{\mathbf{F}}^C [k \omega_o] = \sum_{v=-\infty}^{\infty} \mathbf{F}^C [(k + vM)\omega_o] \quad \text{for } 1-H \leq k \leq H-1$$

where v is an integer indicating the repetition of the signal spectrum at integer multiples of the sampling frequency $M \omega_o$. From the equation above, we can see that all harmonics in \mathbf{F}^C which are higher than or equal to H are being added back to low harmonic terms of $\bar{\mathbf{F}}^C$, affecting the accuracy of the lower harmonics.

In this section, we assume that all the computations are made with infinite precision, and the error contributed by computing the DFT and its inverse is negligible. Then the error between the computed solution obtained by the harmonic-Newton method and the exact solution is only due to the accumulated aliasing distortion. To study the error produced by the harmonic-Newton method, we first look at the periodic nonautonomous nonlinear system. The results will later be extended to autonomous and almost-periodic systems.

Consider the same nonlinear differential equations for periodic nonautonomous systems as in Chapter 7:

$$\mathbf{f}(\mathbf{v}, t) = \mathbf{i}(\mathbf{v}(t)) + \dot{\mathbf{q}}(\mathbf{v}(t)) + \int_{-\infty}^{\infty} \mathbf{y}(t - \tau) \mathbf{v}(\tau) d\tau + \mathbf{u}_s(t) \quad (9.1.1)$$

We denote $\bar{\mathbf{F}}_H^C$, $\bar{\mathbf{V}}_H^C$, and $\bar{\mathbf{I}}_H^C$, and $\bar{\mathbf{Q}}_H^C$ as the discrete Fourier transform of $\mathbf{f}(\mathbf{v}, t)$, $\mathbf{v}(t)$, $\mathbf{i}(t)$, and $\mathbf{q}(t)$ respectively, and $\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)$ as the Jacobian matrix of $\bar{\mathbf{F}}_H^C(\bar{\mathbf{V}}_H^C)$, then the harmonic-Newton iteration using DFT is given as:

$$\mathbf{V}_H^{C(u+1)} = \mathbf{V}_H^{C(u)} - (\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^{C(u)}))^{-1} \bar{\mathbf{F}}_H^C(\bar{\mathbf{V}}_H^{C(u)}) \quad (9.1.2)$$

where

$$\bar{\mathbf{F}}_H^C(\bar{\mathbf{V}}_H^C) = \bar{\mathbf{I}}_H^C(\bar{\mathbf{V}}_H^C) + j\Omega \bar{\mathbf{Q}}_H^C(\bar{\mathbf{V}}_H^C) + \bar{\mathbf{Y}}_H^C \bar{\mathbf{V}}_H^C + \bar{\mathbf{U}}_H^C = 0 \quad (9.1.3)$$

In order to prove the convergence of the harmonic-Newton method when DFT is used, we need to set some conditions on the circuits under analysis.

Assumption 9.1:

In addition to the same assumptions stated in Theorem 8.1, we also assume that for any $\bar{\mathbf{V}}_H^{aC}$, $\bar{\mathbf{V}}_H^{bC}$ with $\bar{\mathbf{v}}_H^a(t)$ and $\bar{\mathbf{v}}_H^b(t) \in \mathbf{D}$, there exists a positive constant γ such that:

$$\|\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^{aC}) - \bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^{bC})\| \leq \gamma \|\bar{\mathbf{V}}_H^{aC} - \bar{\mathbf{V}}_H^{bC}\|$$

Lemma 9.1:

Assume that the conditions stated in Assumption 9.1 are satisfied. Let $\hat{\mathbf{v}}(t)$ be the isolated periodic solution of $\mathbf{f}(\mathbf{v}, t) = 0$, then the vector $\bar{\mathbf{V}}_H^C$ is an approximate solution of $\bar{\mathbf{F}}_H^C(\bar{\mathbf{V}}_H^C) = 0$. i.e.

$$\bar{\mathbf{F}}_H^C(\bar{\mathbf{V}}_H^C) \rightarrow 0 \text{ as } H \rightarrow \infty$$

Proof:

The evaluation of all nonlinear devices of the system is done in the time domain then converted into the frequency domain. This implies that

$$\bar{\mathbf{I}}^C[k\omega_o](\bar{\mathbf{V}}^C) = \bar{\mathbf{I}}^C[k\omega_o](\hat{\mathbf{V}}^C)$$

and

$$\bar{\mathbf{Q}}^C[k\omega_o](\bar{\mathbf{V}}^C) = \bar{\mathbf{Q}}^C[k\omega_o](\hat{\mathbf{V}}^C)$$

Hence, each DFT component can be expressed as:

$$\begin{aligned}
 \bar{F}^C[k\omega_o](\bar{V}^C) &= \bar{I}^C[k\omega_o](\bar{V}^C) + j k\omega_o \bar{Q}^C[k\omega_o](\bar{V}^C) + Y^C \bar{V}^C[k\omega_o] + \bar{U}[k\omega_o]^C \\
 &= \sum_{v=-\infty}^{\infty} \bar{I}^C[(k+vM)\omega_o](\hat{V}^C) + j \sum_{v=-\infty}^{\infty} k\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C) \\
 &\quad + \sum_{v=-\infty}^{\infty} Y^C \hat{V}^C[(k+vM)\omega_o] + \sum_{v=-\infty}^{\infty} U^C[(k+vM)\omega_o] \\
 &= \sum_{v=-\infty}^{\infty} \bar{I}^C[(k+vM)\omega_o](\hat{V}^C) + j \sum_{v=-\infty}^{\infty} (k+vM)\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C) \\
 &\quad - j \sum_{v=-\infty}^{\infty} (k+vM)\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C) + j \sum_{v=-\infty}^{\infty} k\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C) \\
 &\quad + \sum_{v=-\infty}^{\infty} Y^C \hat{V}^C[(k+vM)\omega_o] + \sum_{v=-\infty}^{\infty} U^C[(k+vM)\omega_o] \\
 &= \sum_{v=-\infty}^{\infty} F^C[(k+vM)\omega_o](\hat{V}^C) + j \sum_{v=-\infty}^{\infty} (k - (k+vM))\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C)
 \end{aligned}$$

Since $F^C(\hat{V}^C) = I^C(\hat{V}^C) + j\Omega Q^C(\hat{V}^C) + Y^C \hat{V}^C + U^C = 0$, or

$$F^C[k\omega_o](\hat{V}^C) = 0 \quad \text{for } k = 0, 1, 2, \dots \quad (\text{Lem9.1.1})$$

Hence,

$$\bar{F}^C[k\omega_o](\bar{V}^C) = j \sum_{v=-\infty}^{\infty} (vM)\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C) \quad (\text{Lem9.1.2})$$

$$\begin{aligned}
 \|\bar{F}_H^C(\bar{V}_H^C)\| &\leq \sum_{k=1-H}^{H-1} |\bar{F}^C[k\omega_o](\bar{V}_H^C)| \\
 &\leq \sum_{k=1-H}^{H-1} \sum_{v=-\infty}^{\infty} |j(vM)\omega_o Q^C[(k+vM)\omega_o](\hat{V}^C)| \\
 &\leq \sum_{k=H}^{\infty} |f_{\text{loor}}(\frac{k+H-1}{M})\omega_o| (|Q^C[k\omega_o](\hat{V}^C)| + |Q^C[-k\omega_o](\hat{V}^C)|) \\
 &\leq \sum_{k=H}^{\infty} 2 |f_{\text{loor}}(\frac{k+H-1}{M})\omega_o| \cdot |Q^C[k\omega_o](\hat{V}^C)| \equiv \sum_{k=H}^{\infty} S[k\omega_o] \quad (\text{Lem9.1.3})
 \end{aligned}$$

By Theorem C of Chapter 8, we know that $\sum_{k=0}^{\infty} S[k\omega_o]$ is bounded.

$$\sum_{k=H}^{\infty} S[k\omega_o] = \sum_{k=0}^{\infty} S[k\omega_o] - \sum_{k=0}^{H-1} S[k\omega_o]$$

Since $\sum_{k=0}^{H-1} S[k\omega_o] \rightarrow \sum_{k=0}^{\infty} S[k\omega_o]$ as $H \rightarrow \infty$, therefore, $\sum_{k=H}^{\infty} S[k\omega_o] \rightarrow 0$ as $H \rightarrow \infty$.

Thus we have, $\|\bar{F}_H^C(\bar{V}_H^C)\| \rightarrow 0$ as $H \rightarrow \infty$.

Q.E.D.

Lemma 9.2 :

Assume that the conditions stated in Assumption 9.1 are satisfied, then $\det \bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C) \neq 0$, or $\| \bar{\mathbf{J}}_H^C^{-1}(\bar{\mathbf{V}}_H^C) \| \leq \bar{\beta}$.

Proof:

$\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)$ is the Jacobian matrix of $\bar{\mathbf{F}}_H^C(\bar{\mathbf{V}}_H^C)$. To find the basic property of this Jacobian matrix, let us consider this linear system:

$$\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C) \bar{\mathbf{Y}}_H^C = \bar{\mathbf{Z}}_H^C \quad (\text{Lem9.2.1})$$

$$\text{where } \bar{\mathbf{Y}}_H^C = [\bar{\mathbf{Y}}^C[(1-H)\omega_o], \dots, \bar{\mathbf{Y}}^C[0], \dots, \bar{\mathbf{Y}}^C[(H-1)\omega_o]]^T$$

$$\bar{\mathbf{Z}}_H^C = [\bar{\mathbf{Z}}^C[(1-H)\omega_o], \dots, \bar{\mathbf{Z}}^C[0], \dots, \bar{\mathbf{Z}}^C[(H-1)\omega_o]]^T$$

$$\text{and } \mathbf{y}(mh) \equiv \sum_{k=1-H}^{H-1} \bar{\mathbf{Y}}^C[k\omega_o] e^{j2\pi km/M} \quad \text{for } m = 0, \dots, 2H-1.$$

$$\mathbf{z}(mh) \equiv \sum_{k=1-H}^{H-1} \bar{\mathbf{Z}}^C[k\omega_o] e^{j2\pi km/M} \quad \text{for } m = 0, \dots, 2H-1.$$

To show that $\det \bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C) \neq 0$ is equivalent to showing that the mapping $\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)$ is one-to-one, or the null space of $\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)$ is $\{0\}$. First, we need to show that the system described by Eqn. (Lem9.2.1) is equivalent to the following system when evaluated at $t = mh$ for $m = 0, \dots, 2H-1$:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{v}}(\hat{\mathbf{v}}, t) \mathbf{y}(t) = \mathbf{z}(t) \quad (\text{Lem9.2.2})$$

If we take DFT on both sides of Eqn. (Lem 9.2.2) and expand $\mathbf{y}(t)$ using its IDFT coefficients, then the $k\omega_o$ term in Eqn. (Lem9.2.2) is given by:

$$\begin{aligned} & \frac{1}{M} \sum_{m=0}^{M-1} \frac{\partial \mathbf{f}}{\partial \mathbf{v}}(\hat{\mathbf{v}}, mh) \left[\sum_{\bar{k}=1-H}^{H-1} \bar{\mathbf{Y}}^C[\bar{k}\omega_o] e^{j2\pi \bar{k}m/M} \right] e^{-j2\pi km/M} \\ &= \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{z}(mh) e^{-j2\pi km/M} \end{aligned} \quad (\text{Lem9.2.3})$$

or

$$\frac{1}{M} \left[\sum_{m=0}^{M-1} \frac{\partial f}{\partial \mathbf{v}}(\hat{\mathbf{v}}, mh) e^{\frac{j2\pi(1-H-k)m}{M}} \cdots \sum_{m=0}^{M-1} \frac{\partial f}{\partial \mathbf{v}}(\hat{\mathbf{v}}, mh) \cdots \sum_{m=0}^{M-1} \frac{\partial f}{\partial \mathbf{v}}(\hat{\mathbf{v}}, mh) e^{\frac{j2\pi(H-1-k)m}{M}} \right] \begin{bmatrix} \bar{\mathbf{Y}}^C [(1-H) \omega_o] \\ \vdots \\ \bar{\mathbf{Y}}^C [0] \\ \vdots \\ \bar{\mathbf{Y}}^C [(H-1) \omega_o] \end{bmatrix} = \bar{\mathbf{Z}}^C [k \omega_o] \quad (\text{Lem9.2.4})$$

This is equivalent to:

$$\left[\frac{\partial \bar{\mathbf{F}}_H^C[k \omega_o]}{\partial \bar{\mathbf{V}}_H^C[(1-H) \omega_o]} \cdots \frac{\partial \bar{\mathbf{F}}_H^C[k \omega_o]}{\partial \bar{\mathbf{V}}_H^C[0]} \cdots \frac{\partial \bar{\mathbf{F}}_H^C[k \omega_o]}{\partial \bar{\mathbf{V}}_H^C[(H-1) \omega_o]} \right] \begin{bmatrix} \bar{\mathbf{Y}}^C [(1-H) \omega_o] \\ \vdots \\ \bar{\mathbf{Y}}^C [0] \\ \vdots \\ \bar{\mathbf{Y}}^C [(H-1) \omega_o] \end{bmatrix} = \bar{\mathbf{Z}}^C [k \omega_o]$$

or

$$\left[\text{the row of } \bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C) \text{ corresponding } k \omega_o \right] \begin{bmatrix} \bar{\mathbf{Y}}^C [(1-H) \omega_o] \\ \vdots \\ \bar{\mathbf{Y}}^C [0] \\ \vdots \\ \bar{\mathbf{Y}}^C [(H-1) \omega_o] \end{bmatrix} = \bar{\mathbf{Z}}^C [k \omega_o]$$

Hence, Eqn. (Lem9.2.1) is equivalent to Eqn.(Lem9.2.2) for all $t = mh$ where $m = 0, \dots, 2H-1$.

Since $\frac{\partial f}{\partial \mathbf{v}}$ is nonsingular for all t , therefore $\mathbf{z}(t) = 0$ implies $\mathbf{y}(t) = 0$. Moreover,

$$\mathbf{z}(t) = 0 \text{ for } t = mh \text{ where } m = 0, \dots, 2H-1 \iff \bar{\mathbf{Z}}_H^C = 0$$

$$\mathbf{y}(t) = 0 \text{ for } t = mh \text{ where } m = 0, \dots, 2H-1 \iff \bar{\mathbf{Y}}_H^C = 0$$

Therefore, $\bar{\mathbf{Z}}_H^C = 0$ implies $\bar{\mathbf{Y}}_H^C = 0$. Hence, $\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)$ has a null space of $\{0\}$, or $\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)$ is non-

singular. Therefore, $\|\bar{\mathbf{J}}_H^C(\bar{\mathbf{V}}_H^C)^{-1}\| \leq \bar{\beta}$.

Q.E.D.

Theorem 9.1:

Assume that the conditions stated in Assumption 9.1 are satisfied. Let $\bar{V}_H^{C^*}$ satisfies $\bar{F}_H^C(\bar{V}_H^{C^*}) = 0$, then $\bar{V}_H^{C^*}$ converges to \hat{V}^C as $H \rightarrow \infty$.

Proof:

Following the same argument as in Theorem 8.1 and by Lamma 9.1, 9.2 and Assumption 9.1, we can show that for some H^* , the Newton iteration is well-defined and converge to a solution of $\bar{F}_H(\bar{V}_H^{*C}) = 0$ for all $H \geq H^*$,

Moreover,

$$\begin{aligned} \|\bar{V}_H^* - \hat{V}\| &\leq \|\bar{V}_H^* - \bar{V}_H\| + \|\bar{V}_H - \hat{V}\| \\ &\leq \sum_{k=H}^{\infty} | \text{floor}(\frac{k+H-1}{M}) \omega_o | | 2Q^C[k\omega_o](\hat{V}^C) | \bar{\beta} + \sum_{k=H}^{\infty} 2 | \hat{V}^C[k\omega_o] | \end{aligned} \quad (9.1.8)$$

Therefore, $\bar{V}_H^* \rightarrow \hat{V}$ as $H \rightarrow \infty$.

Q.E.D.

Recall that the bound of the difference between the exact solution and the truncated solution (presented in Section 8.2) is given by:

$$\|\bar{V}_H^* - \hat{V}\| \leq \frac{2}{\omega_o} \frac{1}{H} C C_1 \beta + \frac{1}{\omega_o} \frac{1}{H} C \quad (9.1.9)$$

where $C \equiv \|\hat{V}\|$, $C_1 \equiv \max_{D \times \mathcal{L}} \|\frac{\partial f}{\partial v}(v, t)\|$, and $\beta \geq \|\mathbf{J}_H(\hat{V}_H)^{-1}\|$.

Notice that there are two terms in the right hand side of both Eqn. (9.1.8) and (9.1.9). The term $\sum_{k=H}^{\infty} 2 | \hat{V}[k\omega_o] |$ in Eqn. (9.1.8) can be related to $\frac{1}{\omega_o} \frac{1}{H} C$ in Eqn. (9.1.9) since they all represent the error between the initial guess solution and the exact solution. Since $|\hat{V}[k\omega_o]| = \frac{1}{k\omega_o} |\dot{\hat{V}}[k\omega_o]|$, the term $\sum_{k=H}^{\infty} 2 | \hat{V}[k\omega_o] |$ is approximately twice the quantity $\frac{1}{\omega_o} \frac{1}{H} C$. The quantity $\frac{1}{\omega_o} \frac{1}{H} C$ is the error due to the truncation of higher order terms alone while the term $\sum_{k=H}^{\infty} 2 | \hat{V}[k\omega_o] |$ is the error due to the truncation of higher order terms plus the error contributed by the aliasing of higher order terms to the lower harmonics. As we expected, for

periodic circuits, the error due to truncating higher order terms is equal to the error contributed by the aliasing.

The term $\sum_{k=H}^{\infty} | \text{floor}(\frac{k+H-1}{M}) \omega_o | | Q^C[k \omega_o] | \bar{\beta}$ can also be related to $\frac{1}{\omega_o} \frac{1}{H} C C_1 \beta$

since both terms stem from the quantity $\|J_H^{-1}(V_H^0) F_H(V_H^0)\|$, where V_H^0 is the initial guess voltage vector for the harmonic-Newton method. Unlike the truncation of the Fourier series coefficients where the error of $F_H(V_H^0)$ is contributed by all terms in the equation, we found that for DFT, the error of $\bar{F}_H(\bar{V}_H)$ is contributed by the terms with derivative operator only (i.e. the term associated with charge). It is difficult to formulate explicitly the response of a nonlinear device from a stimulus represented in the frequency domain. Therefore, during each evaluation, we have to transform the stimulus of each nonlinear device into a time-domain waveform, calculate the resulting response waveform, and transform the response back into the frequency domain. When Fourier series expansion is used and truncation of higher order harmonics takes place, signals are distorted. Moreover, these error is accumulated each time we transform the signal back and forth between the time-domain and the frequency-domain. Hence, the error in evaluating F_H will be contributed by all terms in the system equation. When DFT and IDFT are used, there is no error produced at the sampled time point when transforming the signal back and forth between the time-domain and frequency-domain, except for the terms associated with the derivative operator. This is due to the $jk \omega_o$ factor produced by the derivative operator.

Similar results can be derived for autonomous systems. The difference between the computed solution, $\bar{X}_H^{C^*}$, and the exact solution, \hat{X}^C is also bounded. This is given by the following inequality:

$$\| \bar{X}_H^{C^*} - \hat{X}^C \| \leq \sum_{k=H}^{\infty} \left[| \text{floor}(\frac{k+H-1}{M}) \omega | | 2Q^C[k \omega_o](\hat{V}^C) | \bar{\beta} + 2 | \hat{V}^C[k \omega_o] | \right] \quad (9.1.11)$$

where $\bar{\beta} \geq \| \bar{J}_H^C(\bar{X}_H^C)^{-1} \|$, with (\bar{X}_H^C) being the initial guess of the solution, and \bar{J}_H^C being the Jacobian matrix of \bar{F}_H^C .

For almost-periodic circuits, *Spectre* uses a special Fourier transform algorithm called *Almost-Periodic Fourier Transform* (APFT), a variation of the Gram-Schmidt orthogonalization procedure[23, 16]. In APFT, the sampling time points are randomly selected. This makes the exact computation of the aliasing error impossible. However, we expect the aliasing error to be at most as large as the error contributed by truncation of higher harmonics; hence, the total error is estimated to be at most twice the error contributed solely by truncation. Recall that for almost periodic circuits, the difference between the computed solution, $\mathbf{v}_H^*(t)$, and the exact solution, $\hat{\mathbf{v}}(t)$ when truncation error is the sole contributor of error is bounded and is given by:

$$\|\mathbf{v}_H^*(t) - \hat{\mathbf{v}}(t)\| \leq 2\beta C_1 \|\hat{\mathbf{v}}_H(t) - \hat{\mathbf{v}}(t)\| + \|\hat{\mathbf{v}}_H(t) - \hat{\mathbf{v}}(t)\| \quad (9.1.12)$$

where β is the norm of the Jacobian matrix, and $C_1 \equiv \max_{\mathbf{D} \times \mathcal{L}} \left\| \frac{\partial \mathbf{f}}{\partial \mathbf{v}}(\mathbf{v}, t) \right\|$. It is still difficult to estimate this error bound or express it in a closed form. However, we can control the total error by monitoring the additional error contributed by reducing the set of frequency from Λ_H to Λ_{H-1} . If we define $error_H \equiv \|\bar{\mathbf{V}}_H^* - \hat{\mathbf{V}}\|$ to be the error between the computed solution using APFT and the exact solution, then

$$error_{H-1} - error_H = 2*(2\beta C_1 + 1) \sum_{\omega_k \in \Lambda'_H} |\hat{\mathbf{V}}[\omega_k]| \quad (9.1.13)$$

where Λ'_H can be $\left\{ \omega \mid \omega = k_1\lambda_1 + k_2\lambda_2 + \dots + k_d\lambda_d : k_j \in \mathbf{Z}; |k_j| = H \text{ for } 1 \leq j \leq d; k_1 \geq 0 \right\}$

or $\left\{ \omega \mid \omega = k_1\lambda_1 + k_2\lambda_2 + \dots + k_d\lambda_d : k_j \in \mathbf{Z}; \sum_{j=1}^d |k_j| = H \text{ for } 1 \leq j \leq d; k_1 \geq 0 \right\}$ depending

on which truncation is used.

9.2. Practical Estimate

Currently, *Spectre* relies on the user to choose intelligently the number of harmonics H and is unable to give any feedback to the user regarding the accuracy of this particular choice of harmonics. We would like to make use of the error bounds derived in the previous section to predict the number of harmonics required for a given error objective before we start the harmonic-Newton

iteration. In this section, we will discuss periodic circuits first. Then the results will be extended into almost-periodic circuits.

Eqn. (9.1.8) and (9.1.11) give the upper bound of the difference between the computed solution and the exact solution for periodic nonautonomous systems and periodic autonomous systems. Note that it is difficult to determine the values of β (norm of the Jacobian matrix) and $|V[k\omega_o]|$ (absolute values of the Fourier coefficients of the solution) since these values are unknown prior to applying the harmonic-Newton method. Hence, we propose a algorithm to estimate these values for any given circuit.

In order to estimate the norm of $\hat{V}[k\omega_o]$, we use Theorem C of Chapter 8, which is repeated here:

Let x be a periodic function with an interval of periodicity $[0, T_o]$. If the function itself together with its derivatives up to order $\kappa-1$, $\kappa \geq 1$ are continuous, and its κ -th derivative is piecewise continuous in the interval, then there exists a constant $A > 0$ such that the Fourier coefficients $X^C[k\omega_o]$ satisfy the inequality $|X^C[k\omega_o]| \leq \frac{A}{|k|^{\kappa+1}}$.

Note that this theorem implies that the series $X^R[k\omega_o]$ and $X^I[k\omega_o]$ approach zero no slower than $\frac{1}{k^{\kappa+1}}$.

Since we are only interested in finding the error bound, we can approximate $|\hat{V}[k\omega_o]|$ by its bound $\frac{A}{k^{\kappa+1}}$ as stated in the theorem above. The remaining task is to find a practical and low cost way to estimate κ and A . The quantity κ measures the smoothness of the function or the nonlinearity of the circuit. For linear circuits, κ is infinite; in fact, it only has one harmonic component which is the same as that of the input. For weakly nonlinear circuits, κ should be fairly large since the amplitude of the Fourier coefficients is expected to vanish very rapidly as k gets large. For nonlinear circuits which produce large harmonic terms, a smaller κ is expected. We further assume that the voltage and the current spectra have similar shapes as illustrated in Fig. 9.1. Thus, the κ extracted from the current spectra can be used for the voltage spectra and vice versa.

In order to obtain A and κ , We take advantage of the initial guess voltage waveform generated by *Spectre*. *Spectre's* initial guess, \bar{V}^0 , is obtained by running an AC analysis on each input frequency. This initial guess usually consists of only the first harmonic term. The value A^i can be estimated to be $\max_k |\bar{V}^i[k\omega_o]|$ if more than one harmonic terms are generated. Since $\bar{F}^i(\bar{V}^0)$ represents the current flowing into node i when the initial voltage, \bar{V}^0 , is placed on the circuit, we can approximate κ by calculating $\bar{F}^i(\bar{V}^0)$ up to some maximum number of harmonics, H_{max} , and choose the corresponding κ^i which minimizes the difference between $\frac{A^i}{|k|^{\kappa^i+1}}$ and the magnitude of the corresponding $\bar{F}^i[k\omega]$. One illustration of the magnitude of $\bar{F}^i(\bar{V}^0)$ is shown in Fig. 9.1. Since \bar{V}^0 is just an initial guess, we will be wasting a lot of computational effort to choose a fancy routine to minimize the error between curve $\frac{A^i}{|k|^{\kappa^i+1}}$ and $\bar{F}^i[k\omega_o]$. Therefore, we only choose randomly one coefficient $k_o \omega_o$ of F^i with k_o sufficiently close to H_{max} , and find the corresponding κ using the equation $F^i[k_o \omega_o] = \frac{A^i}{|k_o|^{\kappa^i+1}}$. The global κ of the circuit can be computed by some

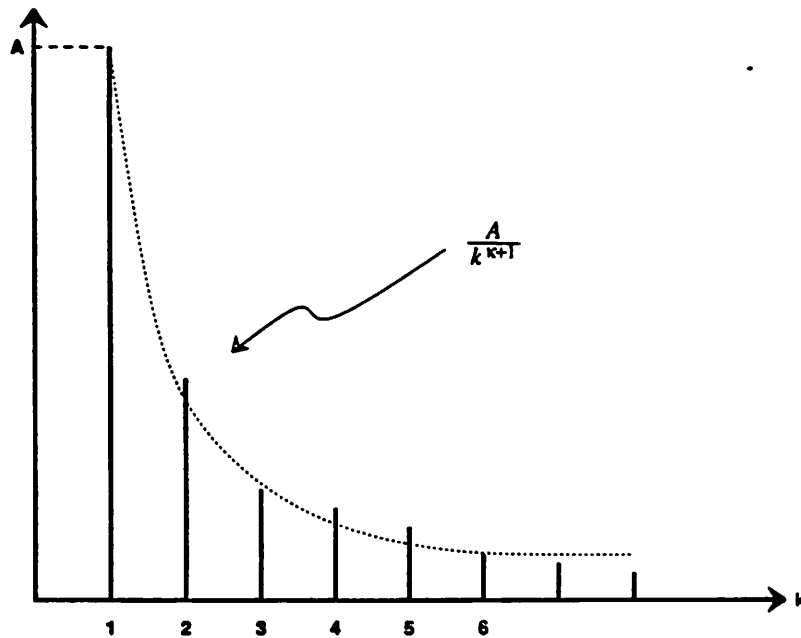


Figure 9.1 Illustration of magnitude of $\bar{F}^i(\bar{V}^0)$.

weighted sum of κ^i . In our case, we choose the average of the two quantities $(\sum_i \kappa^i)/n$ and $\min_i(\kappa^i)$.

The flow chart of the technique described above in predicting the number of harmonics for periodic circuits is shown in Fig. 9.2. Two variables need to be set by the user before applying this algorithm. They are H_{\max} , the maximum number of harmonics desired, and $error_{\max}$, the maximum error allowed.

We also propose a similar technique to predict the number of harmonics for almost-periodic systems. At this point, *Spectre* limits the set of the fundamental frequencies to have two elements $\{\lambda_1, \lambda_2\}$; hence, we will only focus on the case where the circuit has two input signals whose frequencies are not harmonically related. Similar to periodic systems, the major task here is to estimate $|V[\omega_k]|$ for all $\omega_k \in \Lambda_{H_{\max}}$. Since we are unable to find a theorem analogous to Theorem C of Chapter 8 for almost periodic systems to bound the coefficients $|\hat{V}[\omega_k]|$, we need to resort to some heuristic method. Based on our experiments with many almost-periodic systems, we propose the following scheme: First, we need to determine two important parameters κ_1, κ_2 , each measuring the shape of the voltage spectrum when the other input is not presented (i.e. when $\omega_k = k_1\lambda_1$ or $\omega_k = k_2\lambda_2$). The quantities κ_1 and κ_2 can be obtained by the same method as the one used in periodic systems to obtain κ . For the intermodulation terms where $\omega_k = k_1\lambda_1 + k_2\lambda_2$ with k_1 and k_2 being two nonzero integers, we use some weighted function of κ_1 and κ_2 to estimate the voltage magnitude. In fact, the weighted function is

$$\frac{|k_1|}{|k_1| + |k_2|} \kappa_1 + \frac{|k_2|}{|k_1| + |k_2|} \kappa_2.$$

The flow chart is shown in Fig. 9.3.

9.3. Simulation Results

The algorithm described in the previous section has been implemented into *Spectre* to evaluate its accuracy and also to verify the theories derived in Chapter 7 and Chapter 9. Since *Spectre*

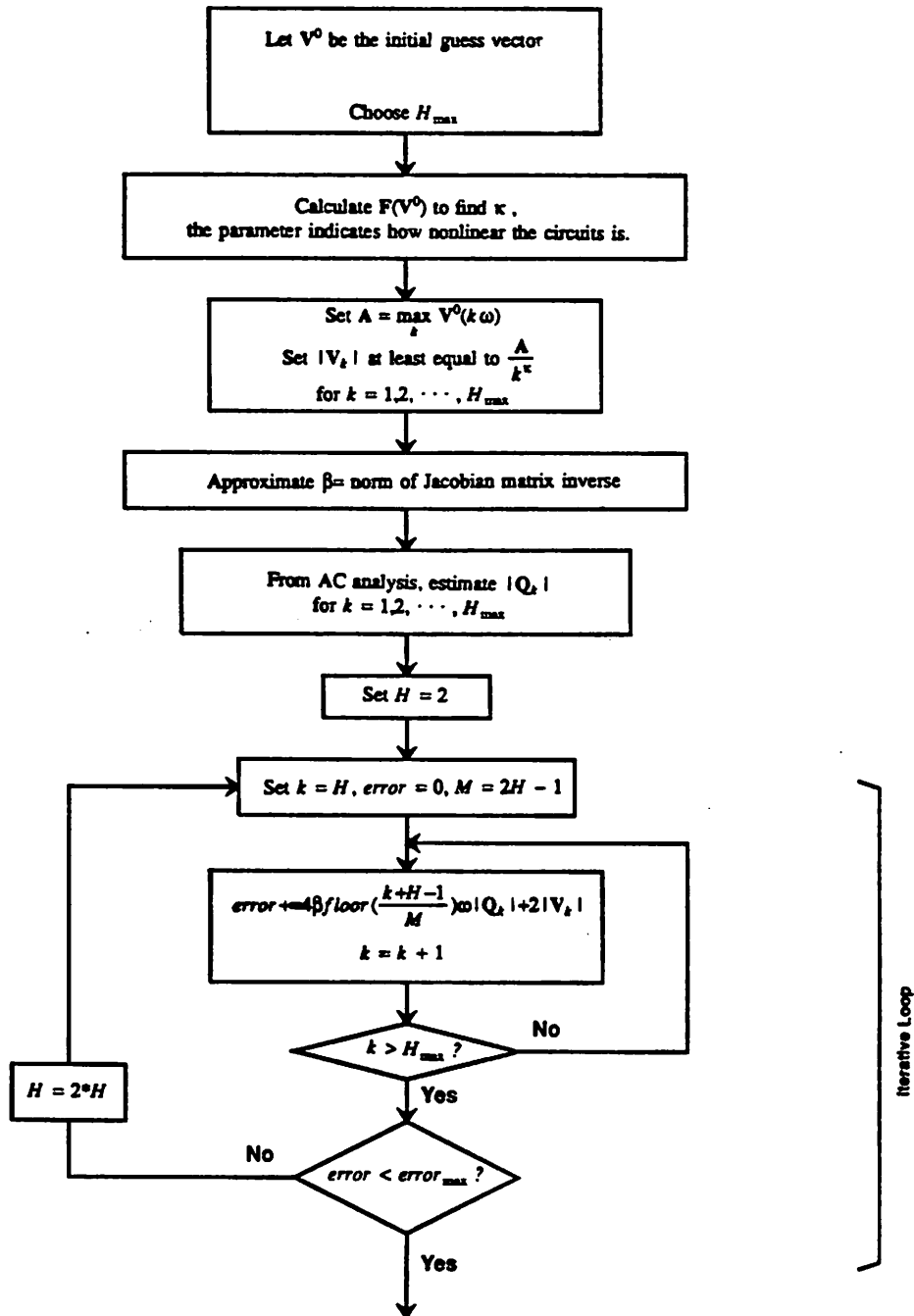


Figure 9.2 Flow Chart to estimate number of harmonic in periodic circuits.

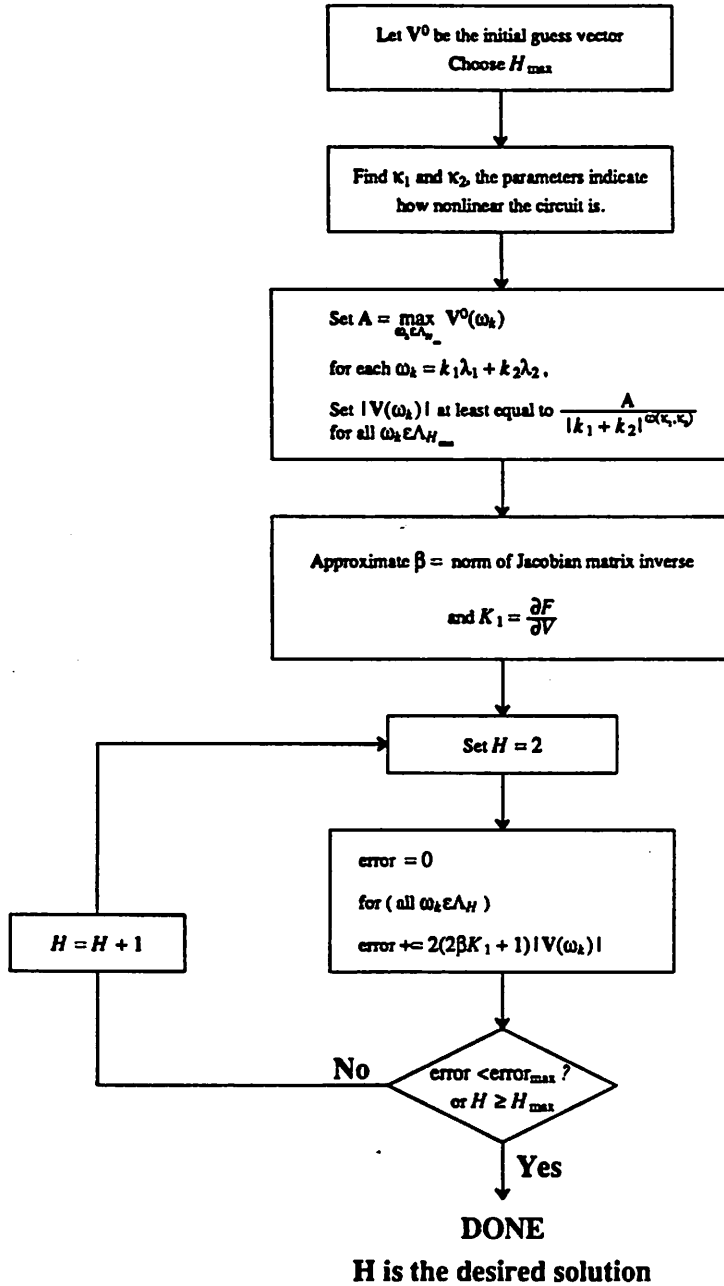


Figure 9.3 Flow Chart to estimate number of harmonic in almost-periodic circuits.

is limited to simulations of nonautonomous systems, we can only test the algorithm for periodic nonautonomous and almost-periodic nonautonomous circuits.

For periodic nonautonomous circuits, we used three different traveling-wave distributed amplifiers as test circuits. The first test circuit is a GaAs traveling-wave distributed amplifier (denoted as GaAs amplifier in Table 9.1). Simulation results of the first test circuit with two different input voltage magnitudes and different maximum allowed errors are given in Fig. 9.4, Fig. 9.5, Fig. 9.6, and Fig. 9.7. GaAs amplifier (I) has input voltage magnitude of -10dB, GaAs amplifier (II) has input voltage magnitude of -0dB, and the input voltage magnitude of GaAs amplifier (III) is -5dB.

The second test circuit is also a GaAs traveling-wave distributed amplifier (denoted as J.Orr amplifier in Table 9.1) which has a different design from the first test circuit. Simulation results of two different input frequencies are given in Fig. 9.8 and Fig. 9.9. J.Orr amplifier (I) has input frequency of 1GHz while J.Orr amplifier (II) has input frequency of 2GHz. The last amplifier is an npn bipolar traveling-wave distributed amplifier. The simulation result is given in Fig.9.10. The results for periodic nonautonomous circuits are summarized in Table 9.1.

test circuit	Err. (V) Allowed	Est. H	Err. (V) Estimated	Err. (V) Observed	% error
GaAs amplifier (I)	8e-3	4	1.2e-5	1.1e-4	0.02
GaAs amplifier (II)	1.5e-2	8	3.4e-2	1.0e-1	5.7
GaAs amplifier (III)	5e-3	16	3.e-3	1.2e-2	0.7
J.Orr amplifier (I)	2e-2	8	1.7e-3	1.3e-4	0.09
J.Orr amplifier (II)	2e-2	8	3.9e-3	1.1e-4	0.22
amplifier	6e-3	4	1.5e-4	5.1e-3	0.8

Table 9.1 Comparison of Error Estimated and Observed for Periodic Nonautonomous Circuits

For almost-periodic circuits, we used a GaAs double balanced mixer as our test circuit. Since the input frequencies and the frequencies of the intermodulation terms differ in several orders of magnitude, it is almost impossible to observe these intermodulation terms and the input frequencies at

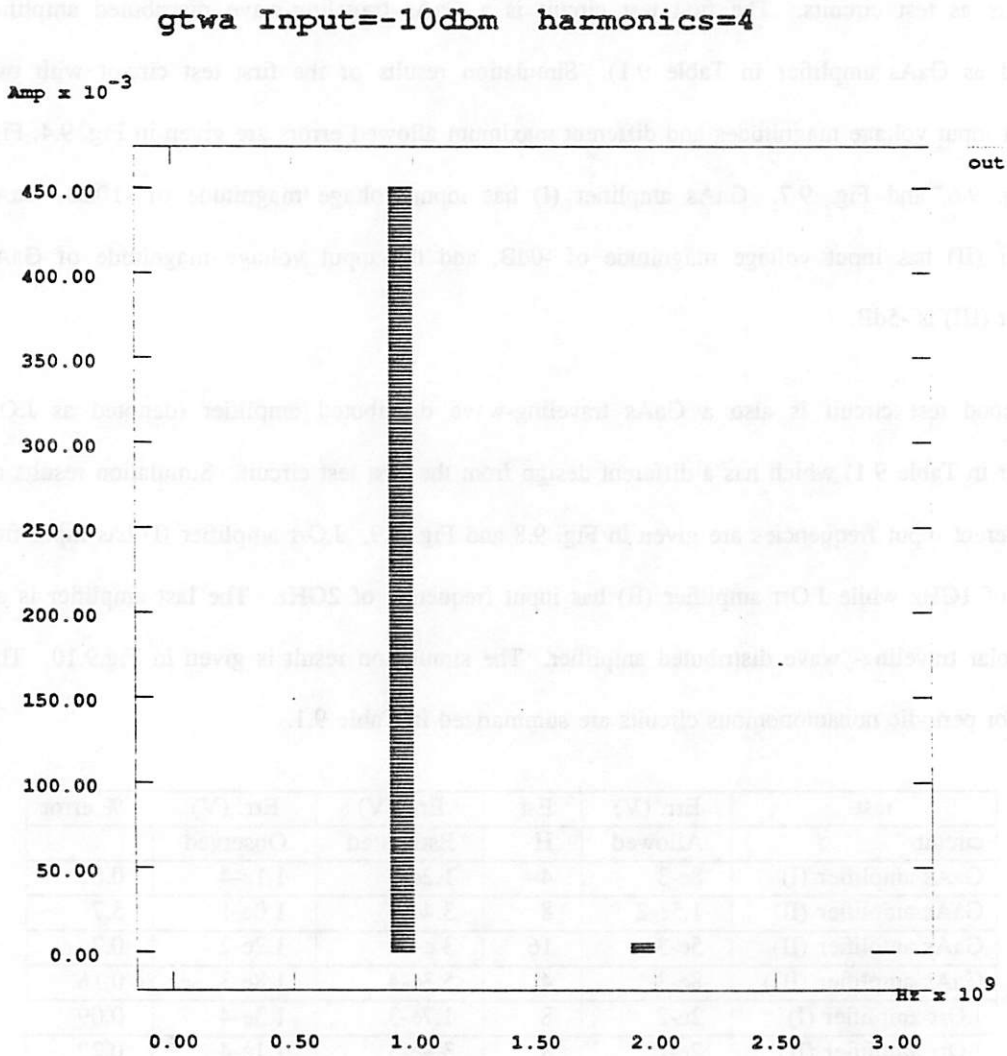


Figure 9.4 (a) Output Spectrum of GaAs amplifier (I) with 4 Harmonics.

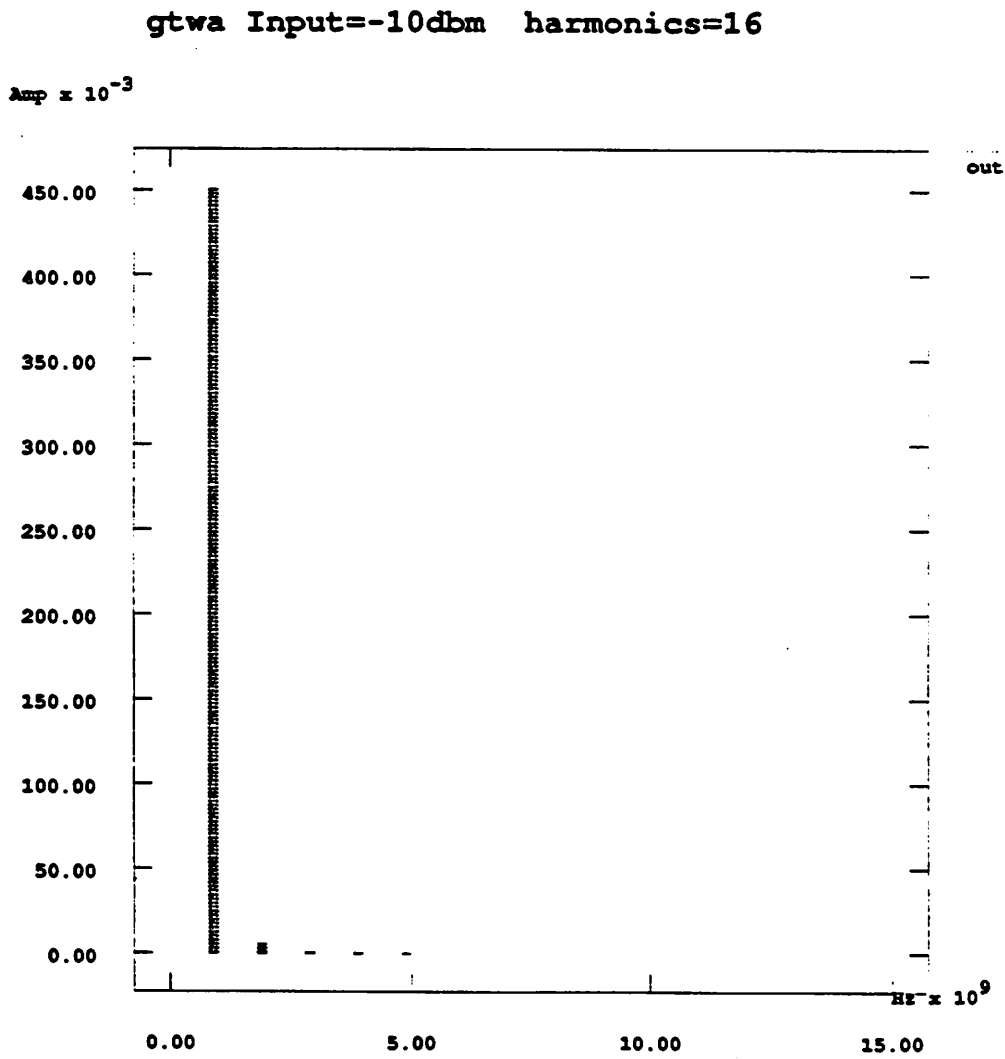


Figure 9.4 (b) Output Spectrum of GaAs amplifier (I) with 16 Harmonics.

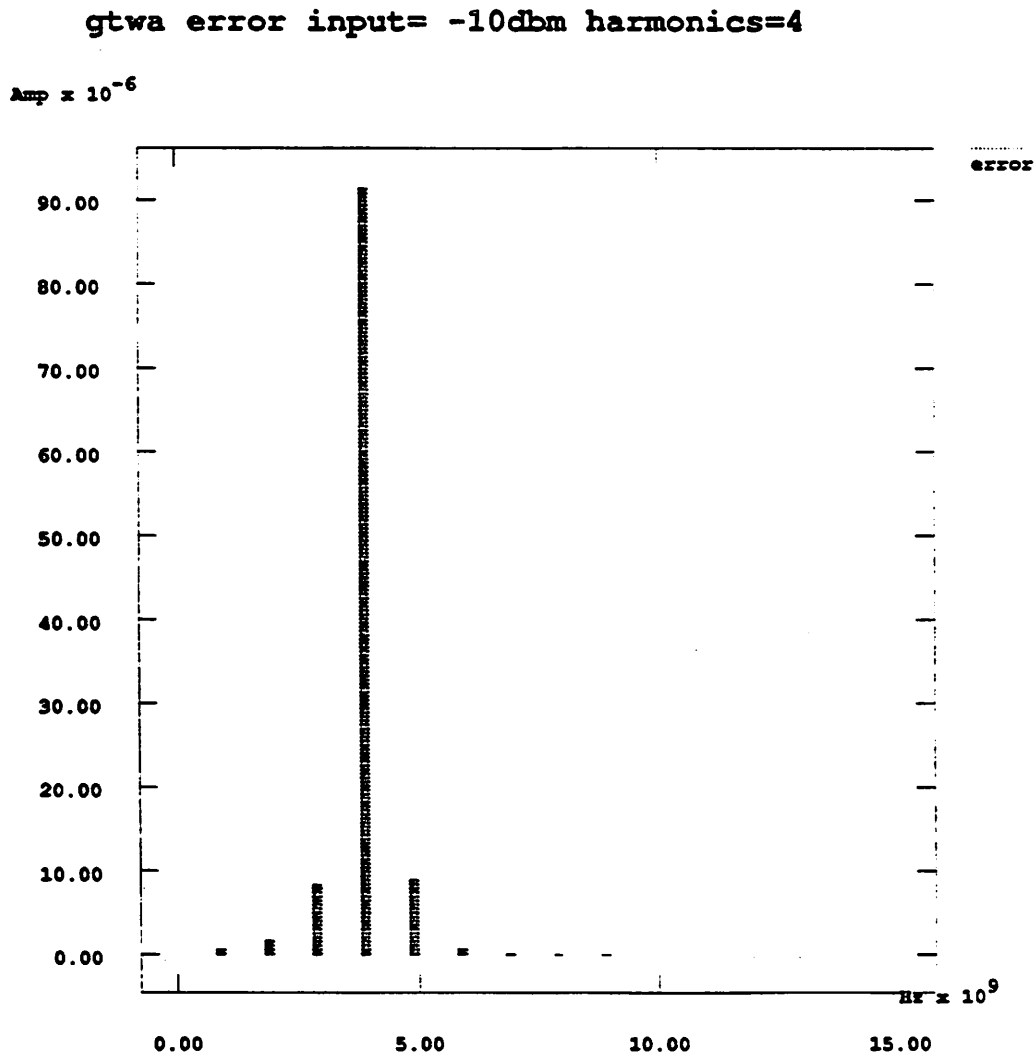


Figure 9.4 (c) The Difference Between Output Spectrum of GaAs amplifier (I) with 4 and 16 Harmonics.

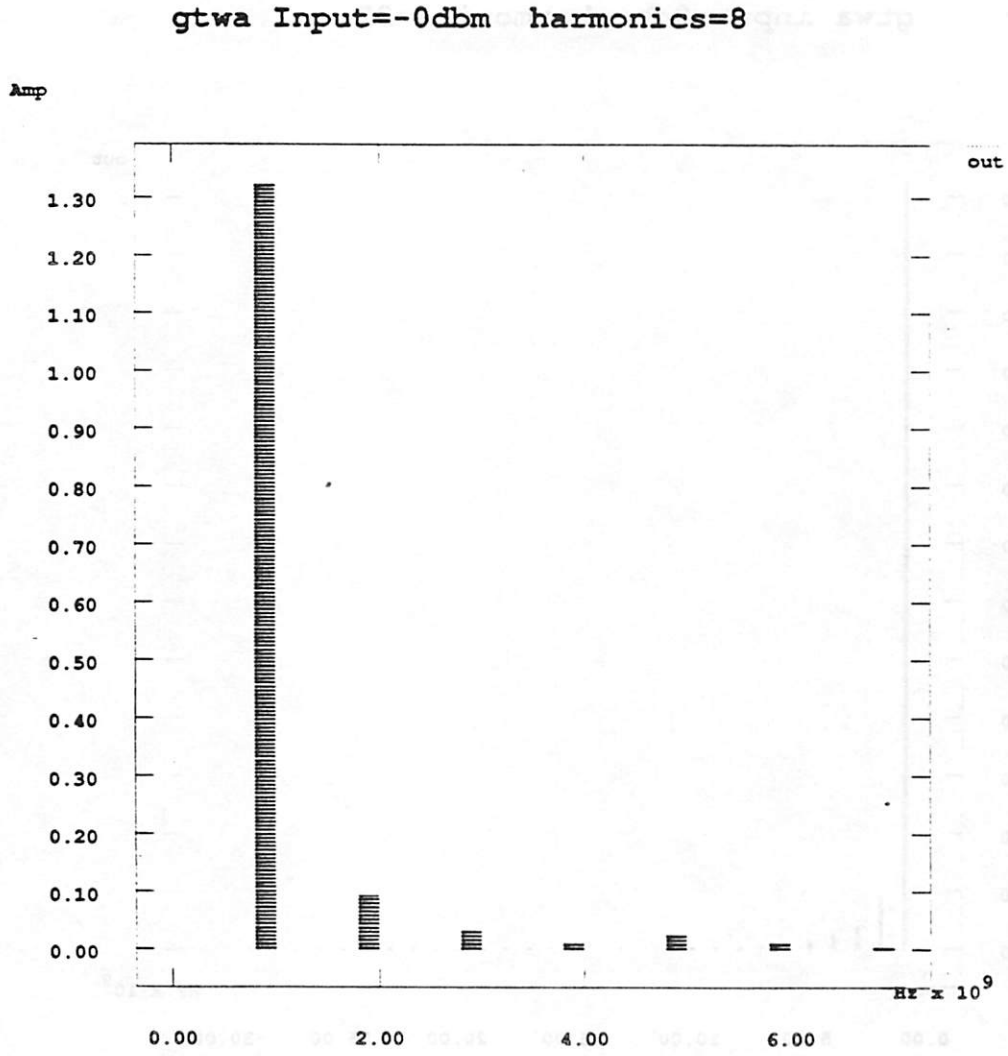


Figure 9.5 (a) Output Spectrum of GaAs amplifier (II) with 8 Harmonics.

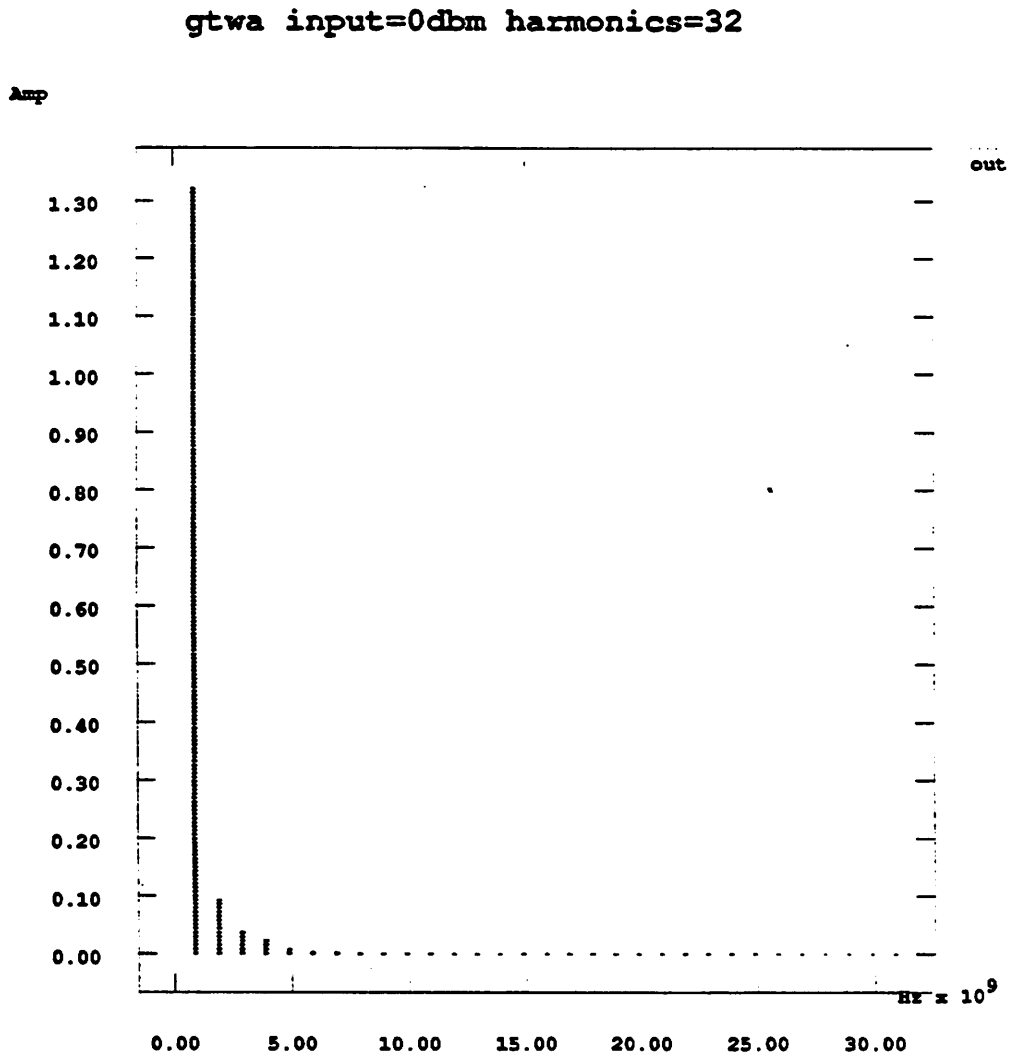


Figure 9.5. (b) Output Spectrum of GaAs amplifier (II) with 32 Harmonics.

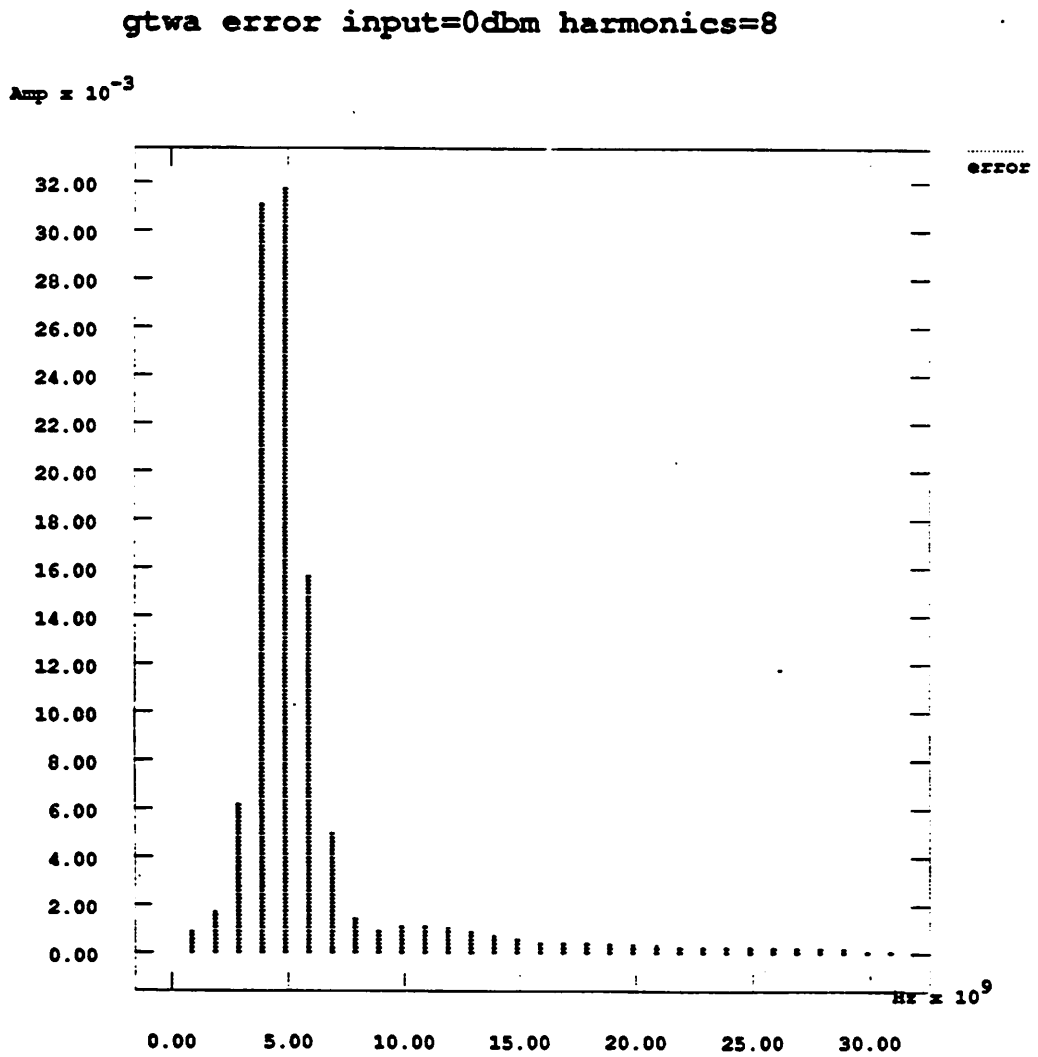


Figure 9.5 (c) Difference Between Output Spectrum of GaAs amplifier (II) with 8 and 32 Harmonics.

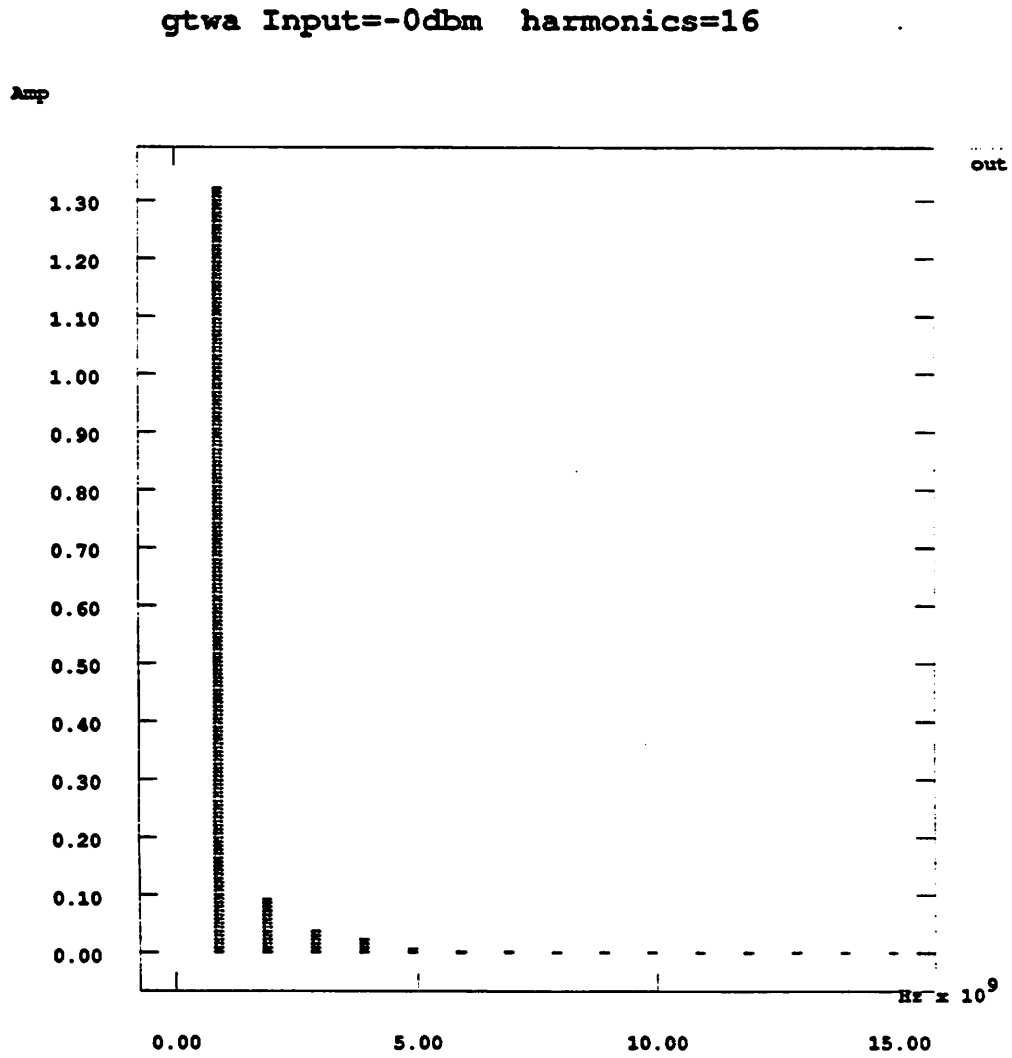


Figure 9.6 (a) Output Spectrum of GaAs amplifier (II) with 16 Harmonics.

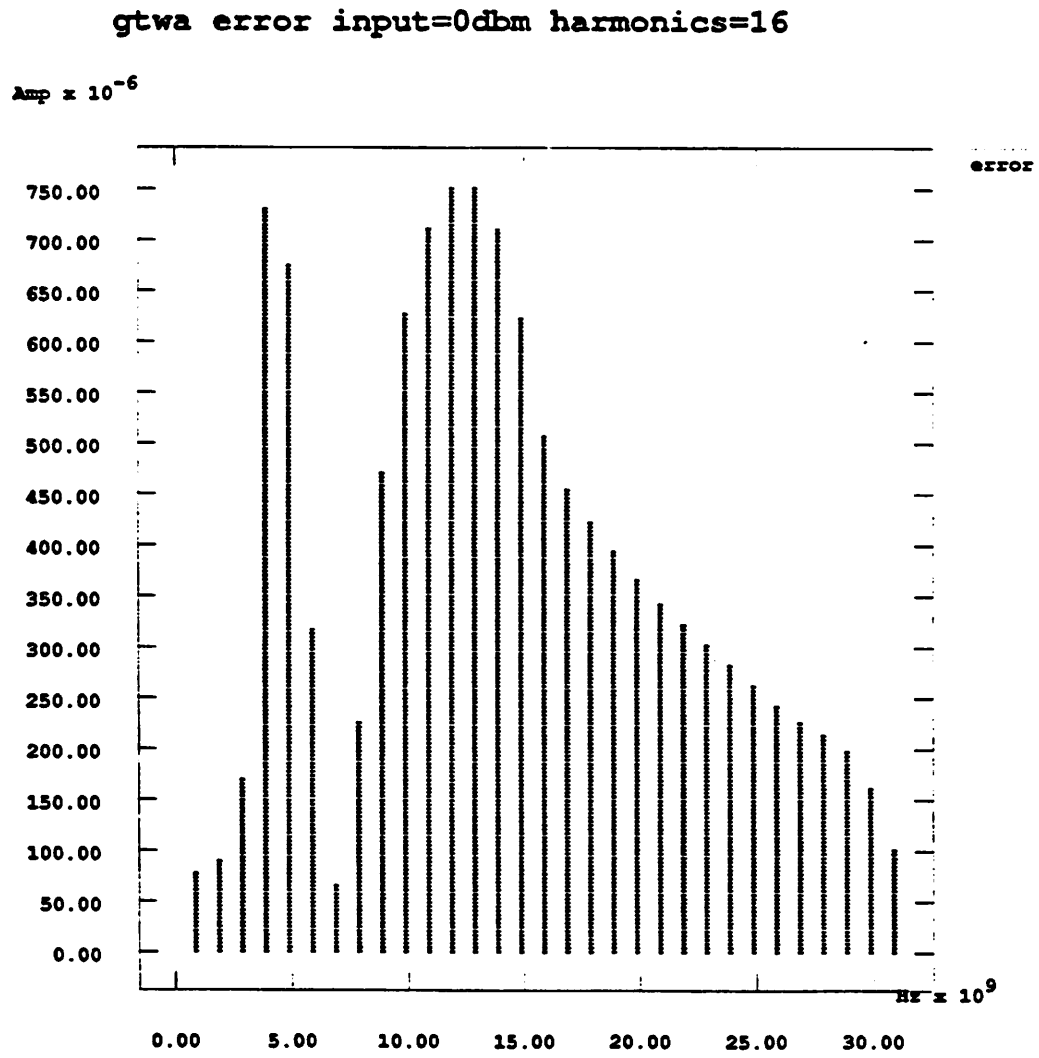


Figure 9.6 (b) Difference Between Output Spectrum of GaAs amplifier (II) with 16 and 32 Harmonics.

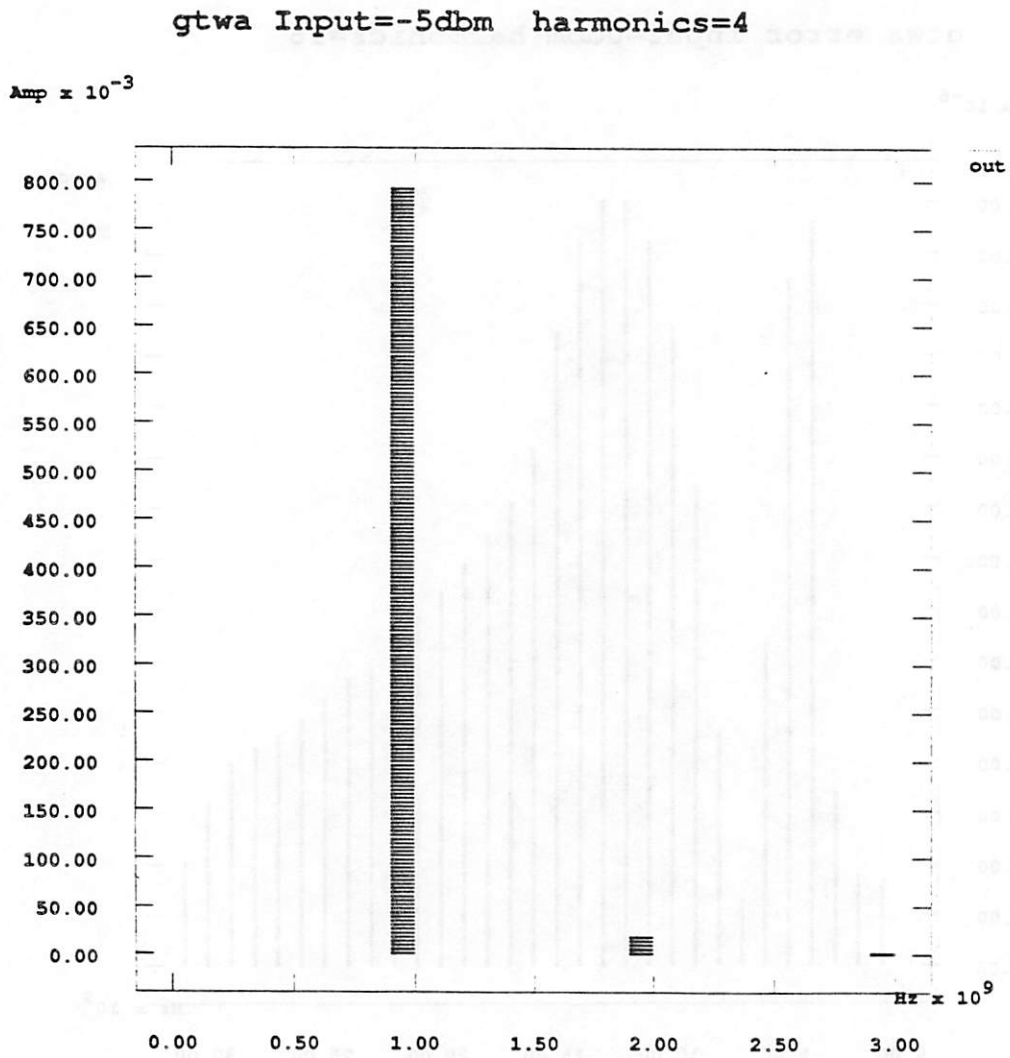


Figure 9.7 (a) Output Spectrum of GaAs amplifier (III) with 4 Harmonics.

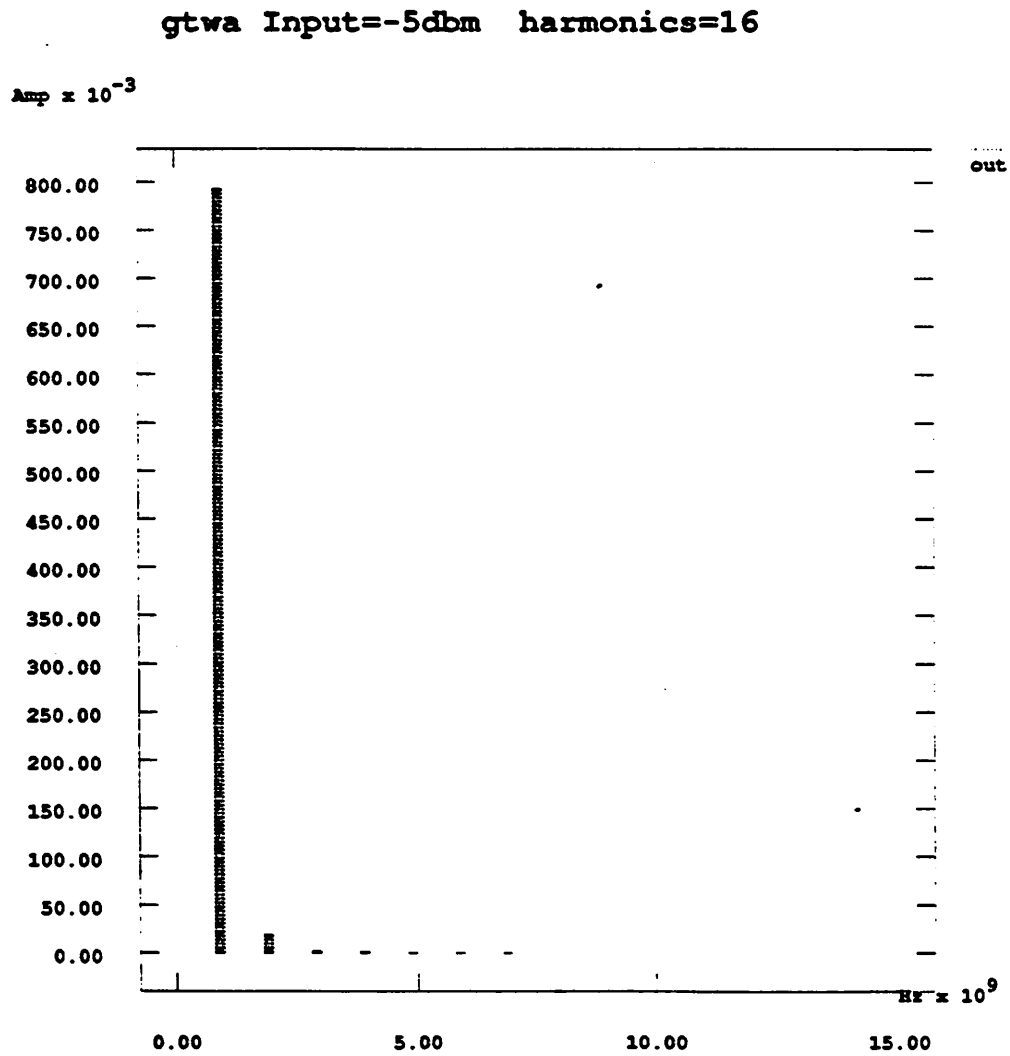


Figure 9.7 (b) Output Spectrum of GaAs amplifier (III) with 16 Harmonics.

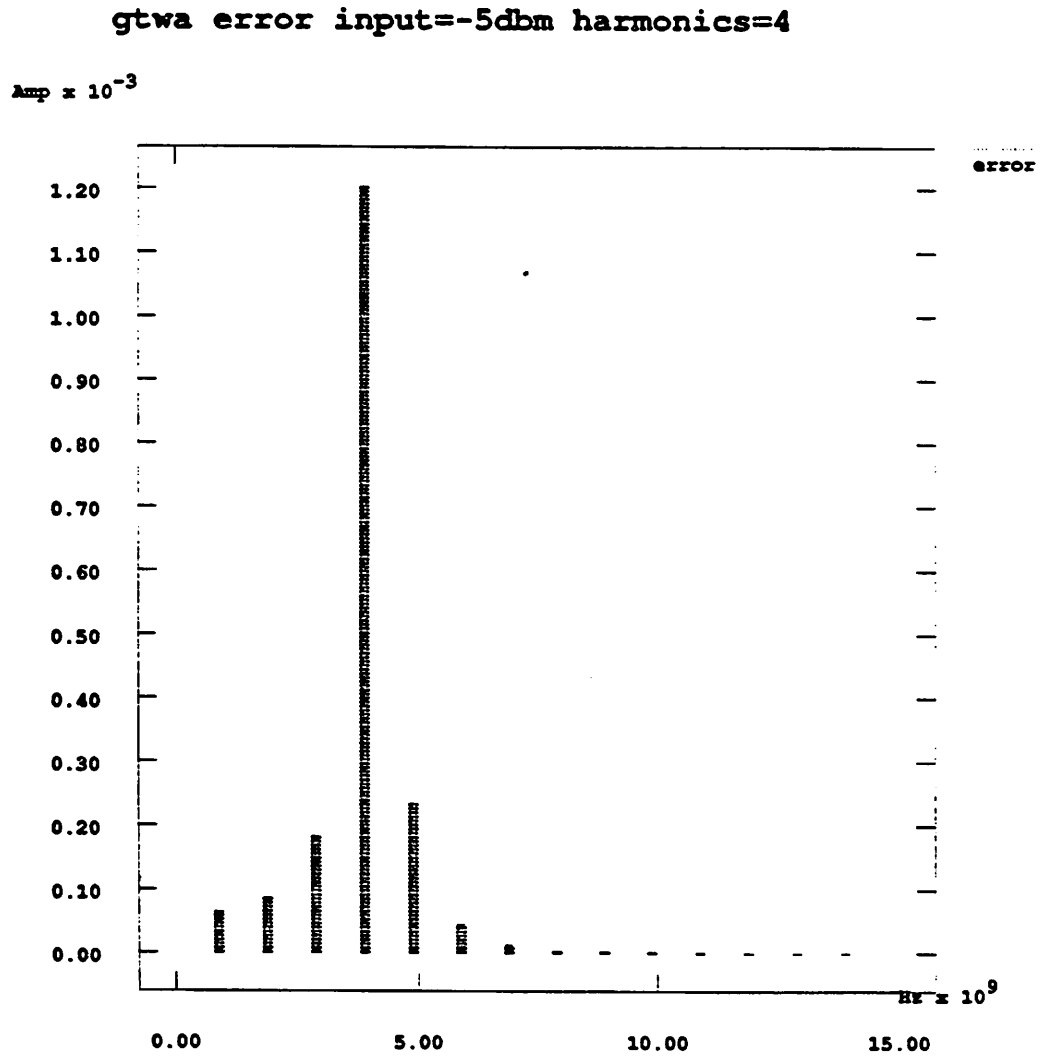


Figure 9.7 (c) Difference Between Output Spectrum of GaAs amplifier (III) with 4 and 16 Harmonics.

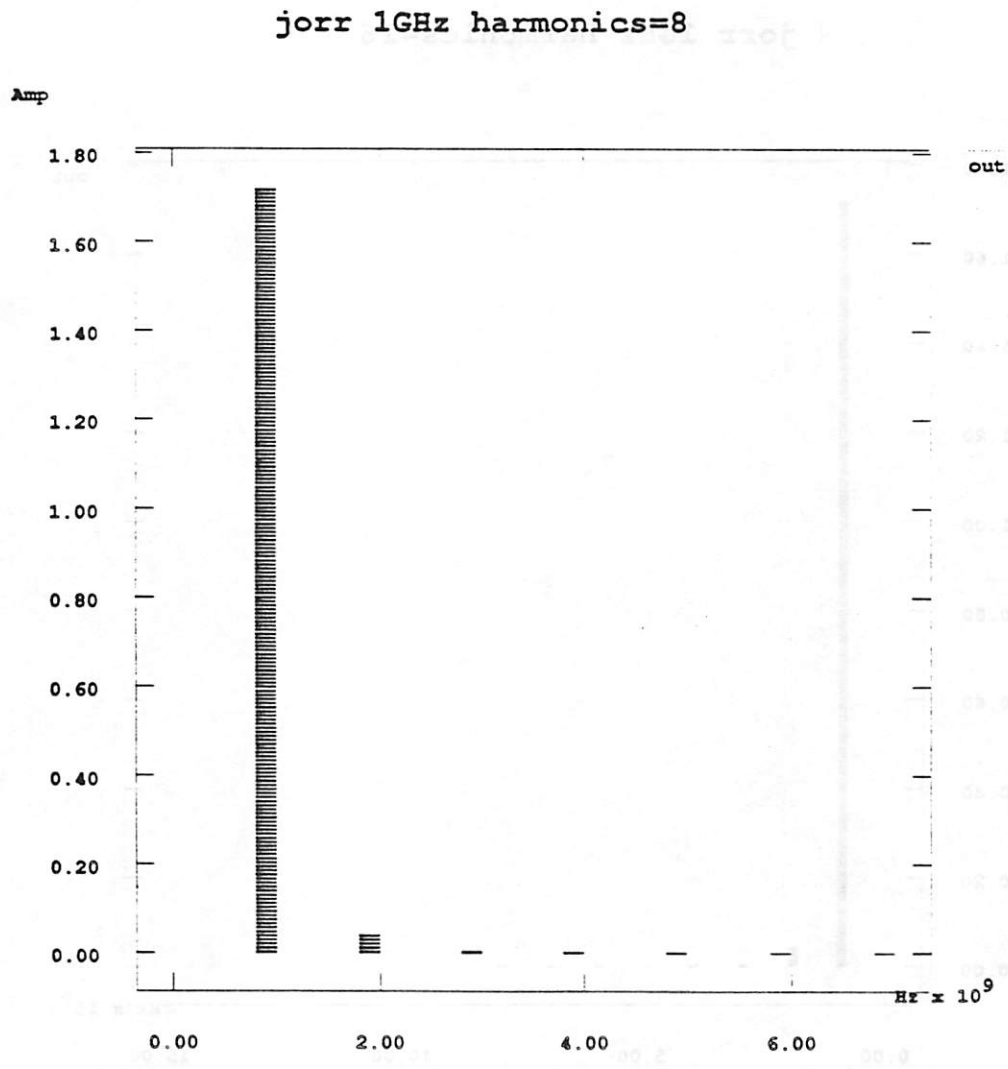


Figure 9.8 (a) Output Spectrum of J.Orr amplifier (I) with 8 Harmonics.

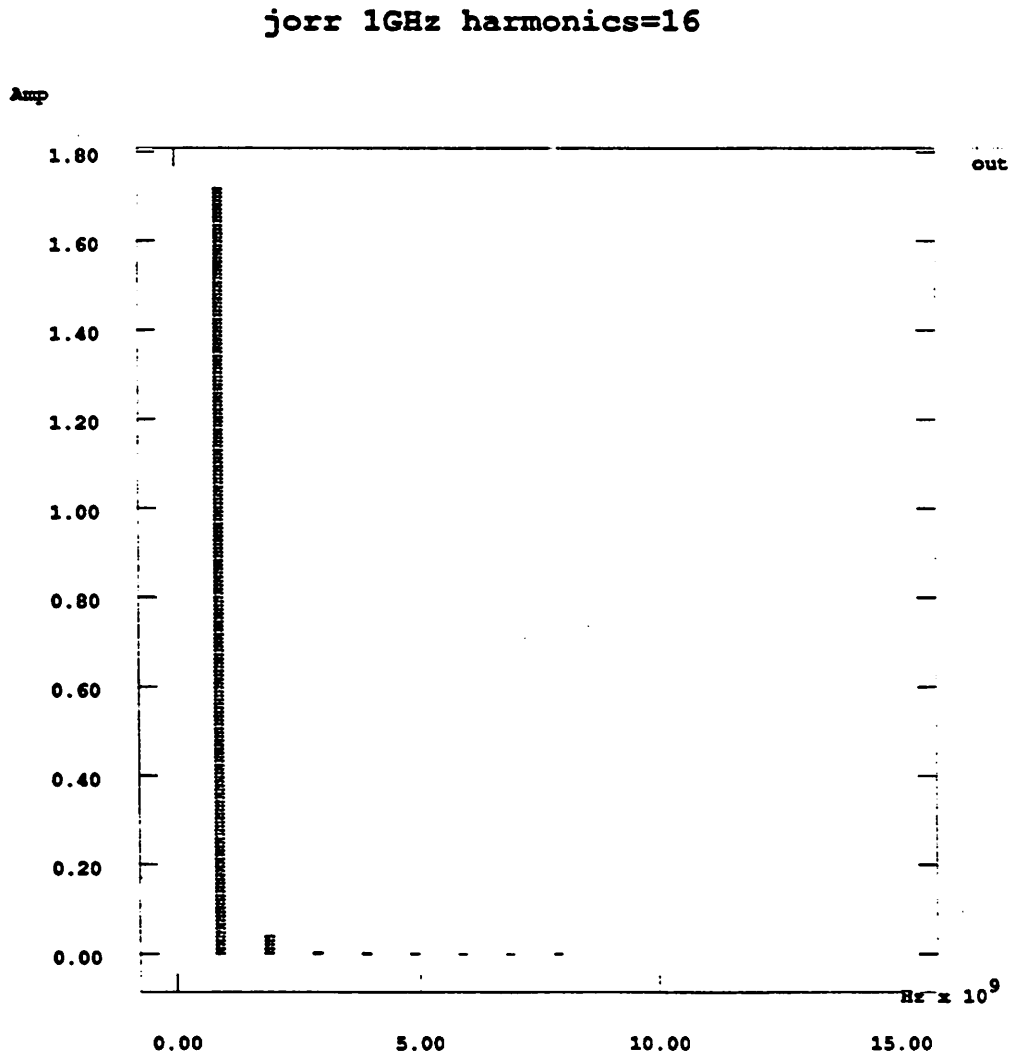


Figure 9.8 (b) Output Spectrum of J.Orr amplifier (I) with 16 Harmonics.

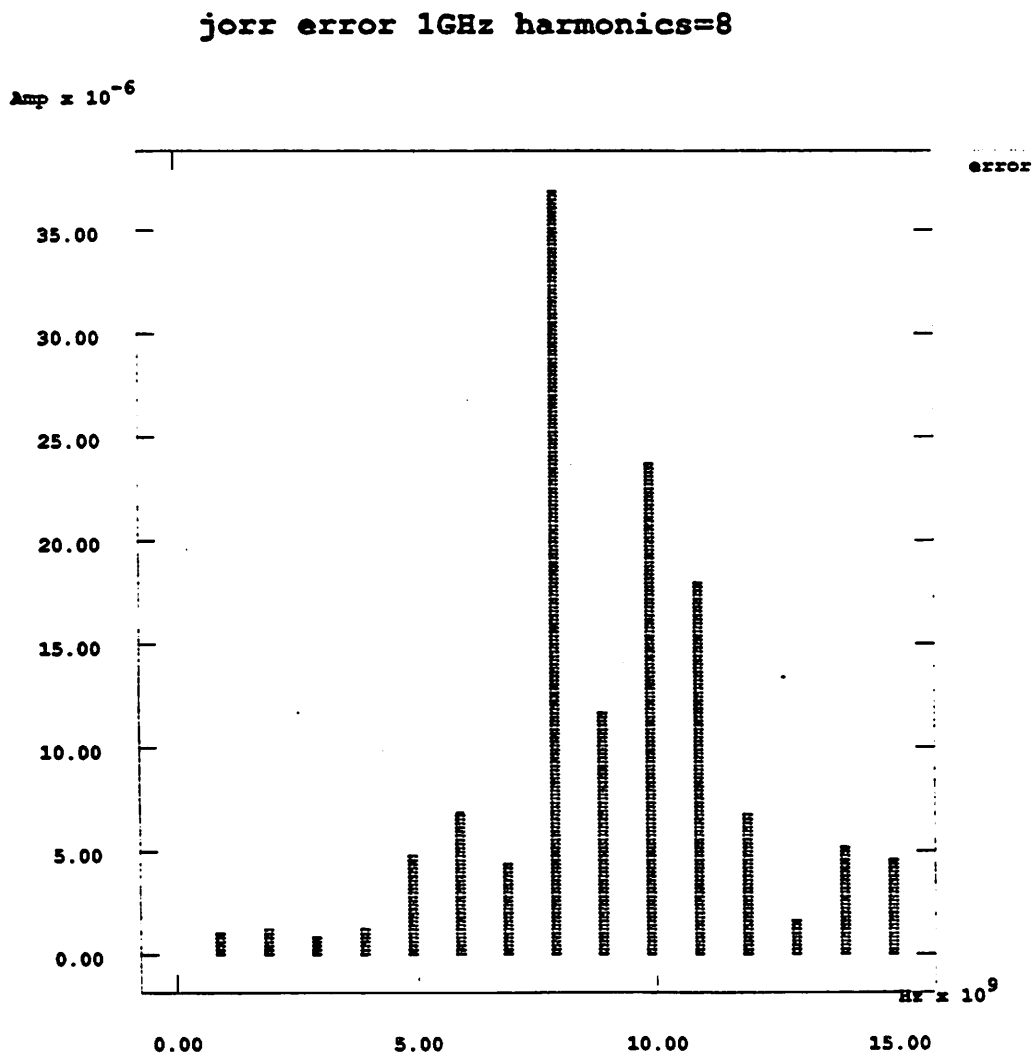


Figure 9.8 (c) Difference Between Output Spectrum of J.Orr amplifier (I) with 8 and 16 Harmonics.

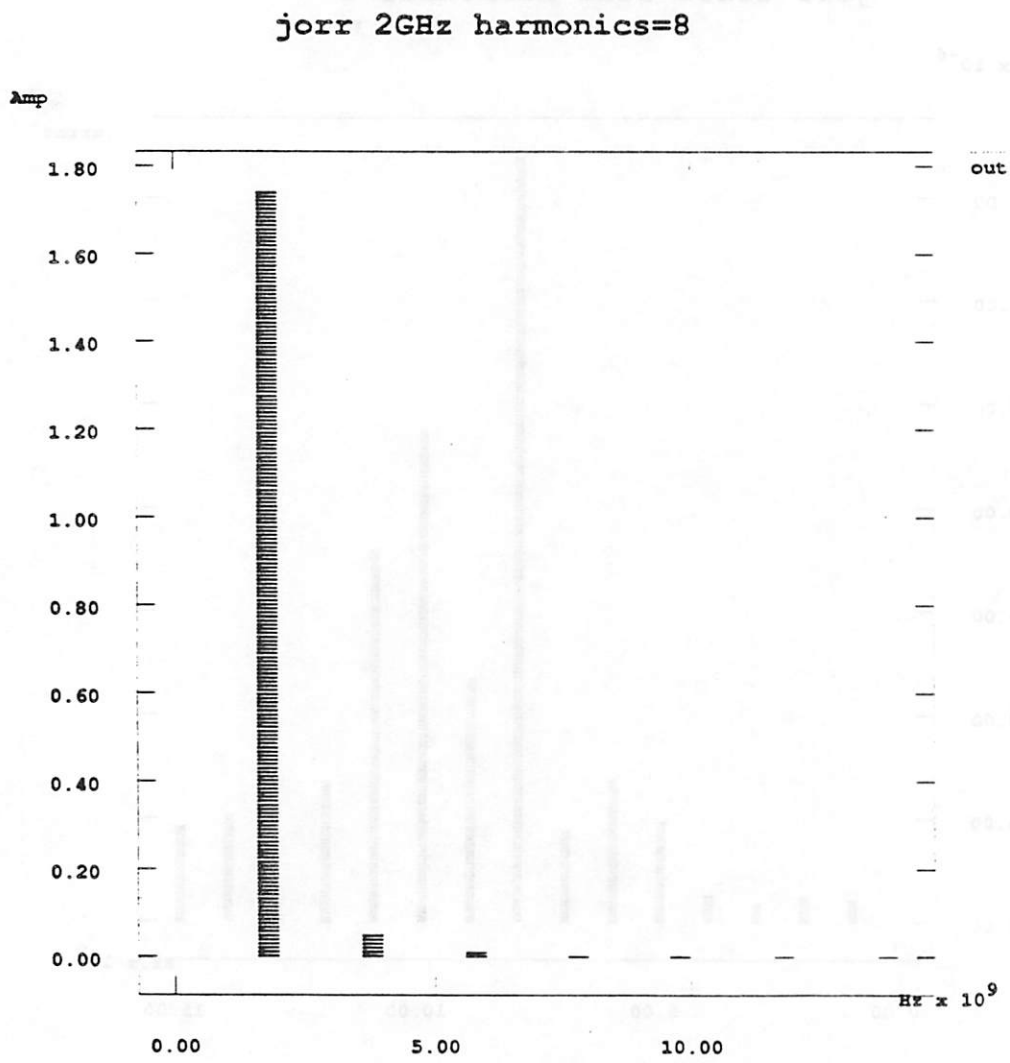


Figure 9.9 (a) Output Spectrum of J.Orr amplifier (II) with 8 Harmonics.

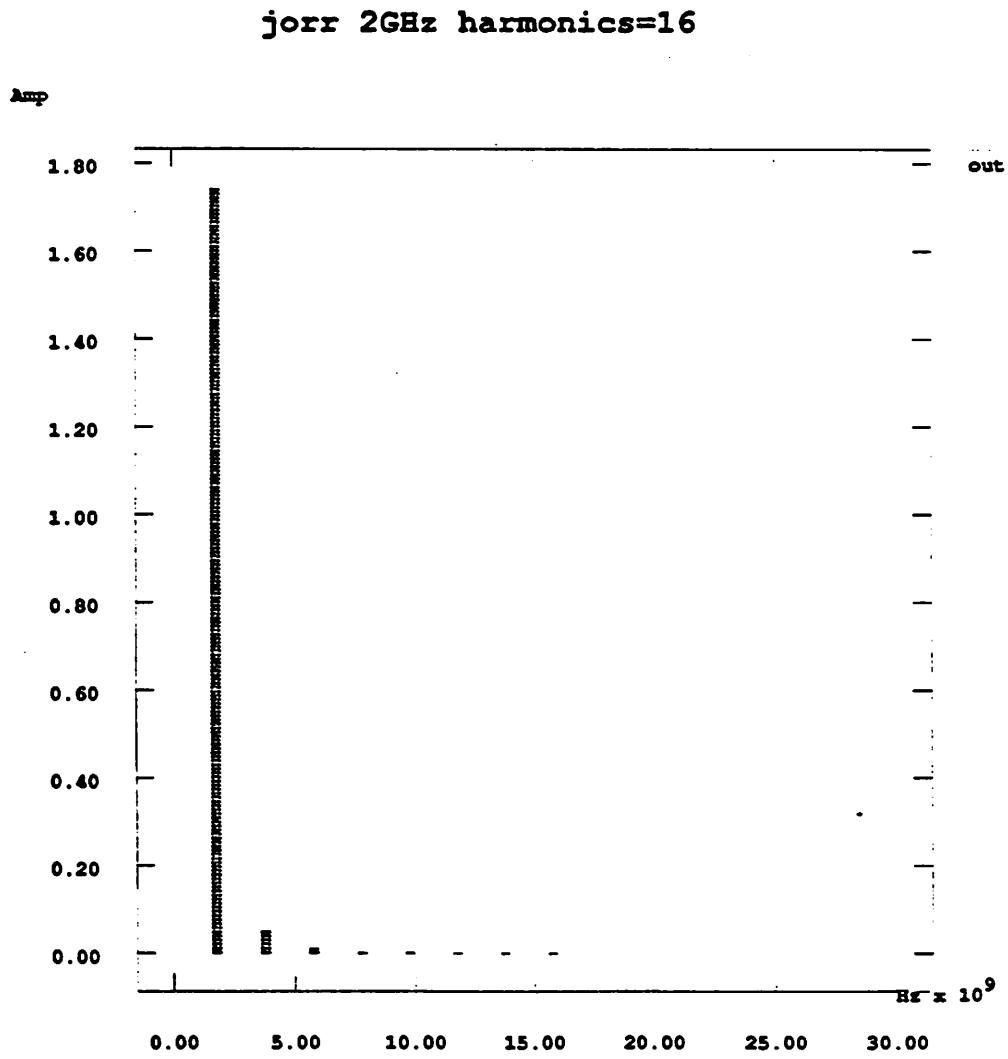


Figure 9.9 (b) Output Spectrum of J.Orr amplifier (II) with 16 Harmonics.

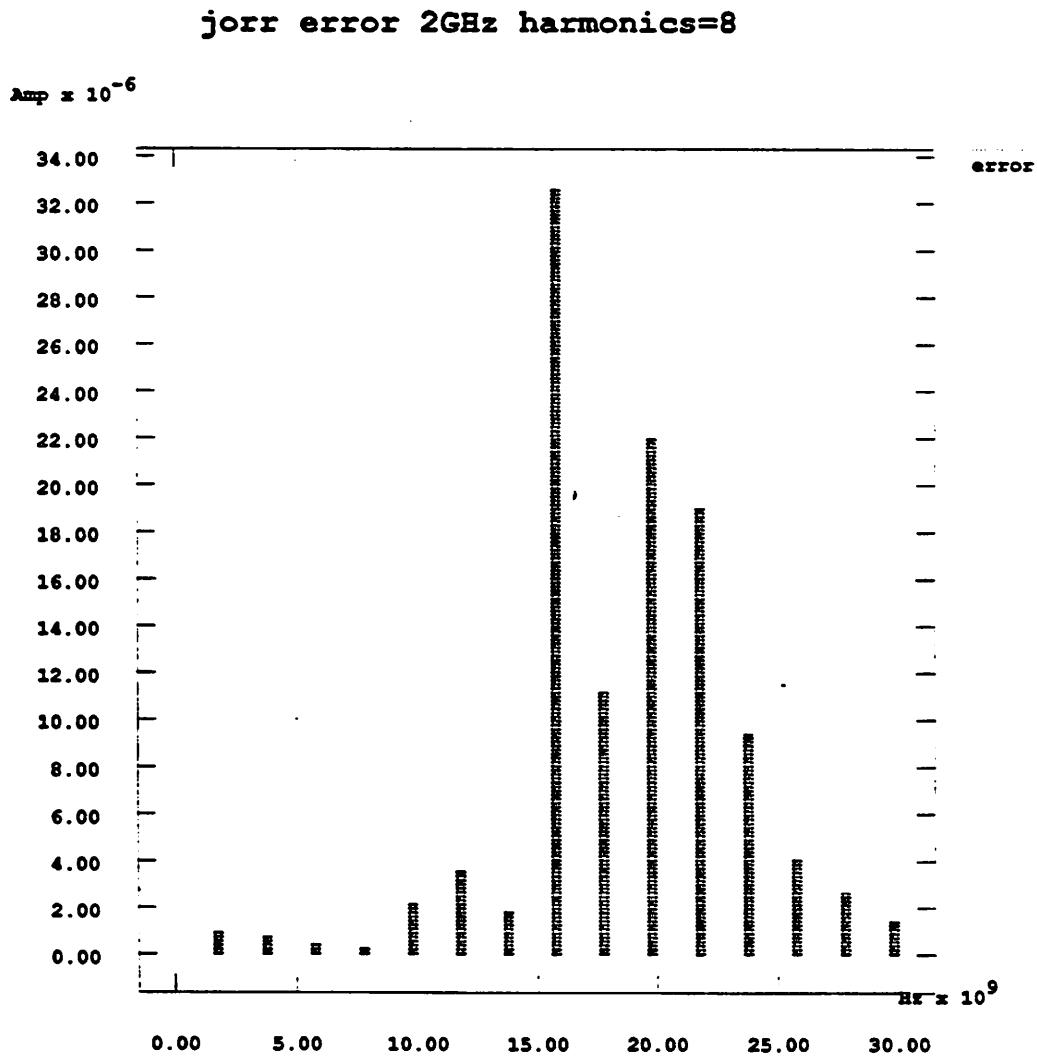


Figure 9.9 (c) Difference Between Output Spectrum of J.Orr amplifier (II) with 8 and 16 Harmonics.

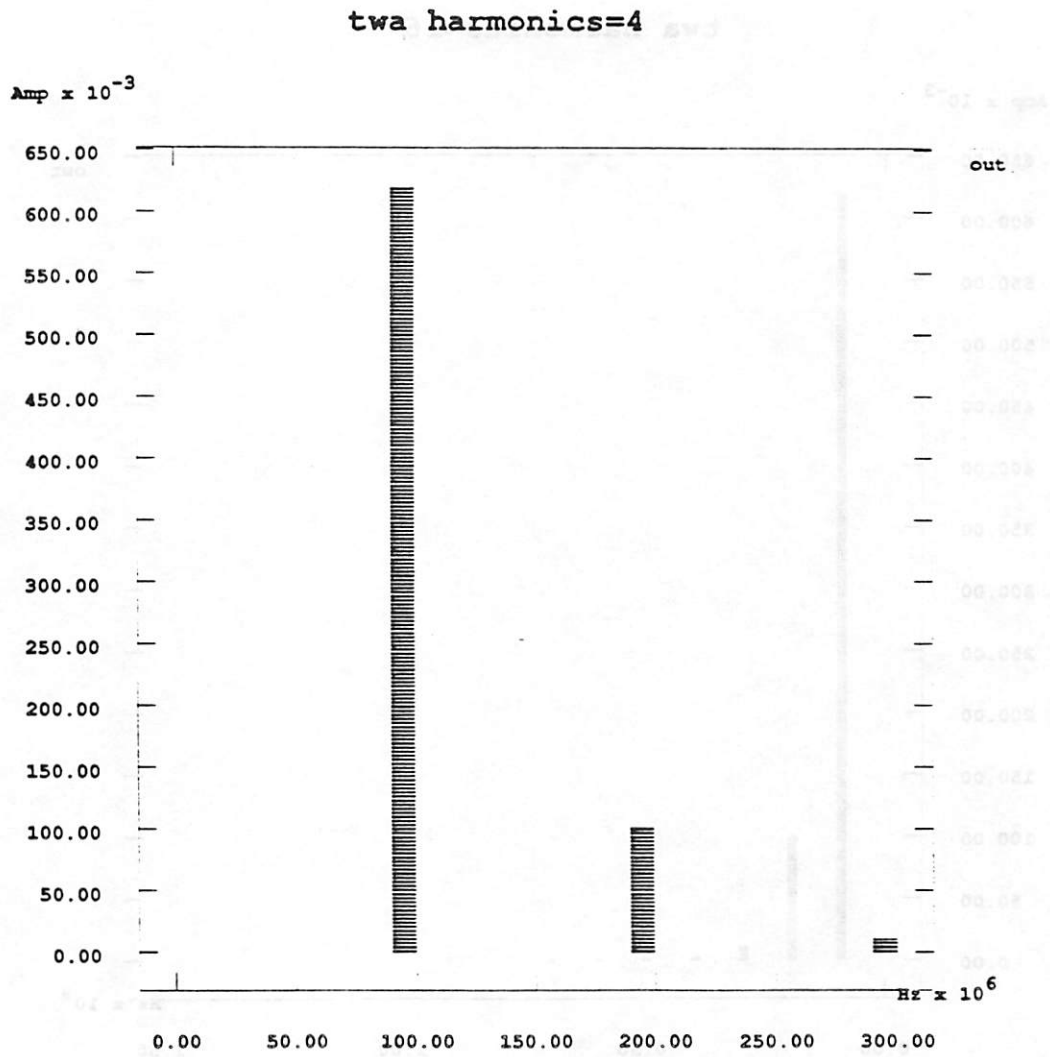


Figure 9.10 (a) Output Spectrum of amplifier with 4 Harmonics.

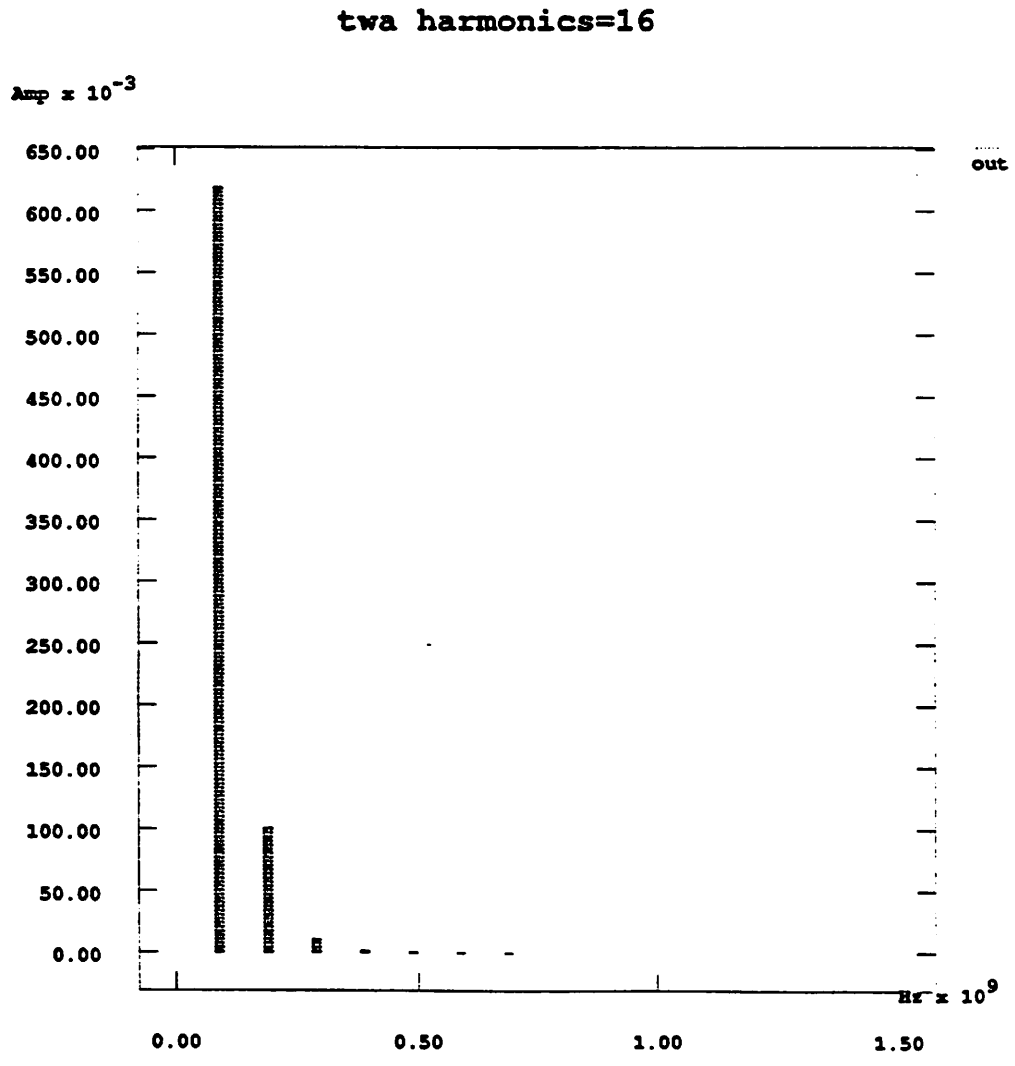


Figure 9.10 (b) Output Spectrum of amplifier with 16 Harmonics.

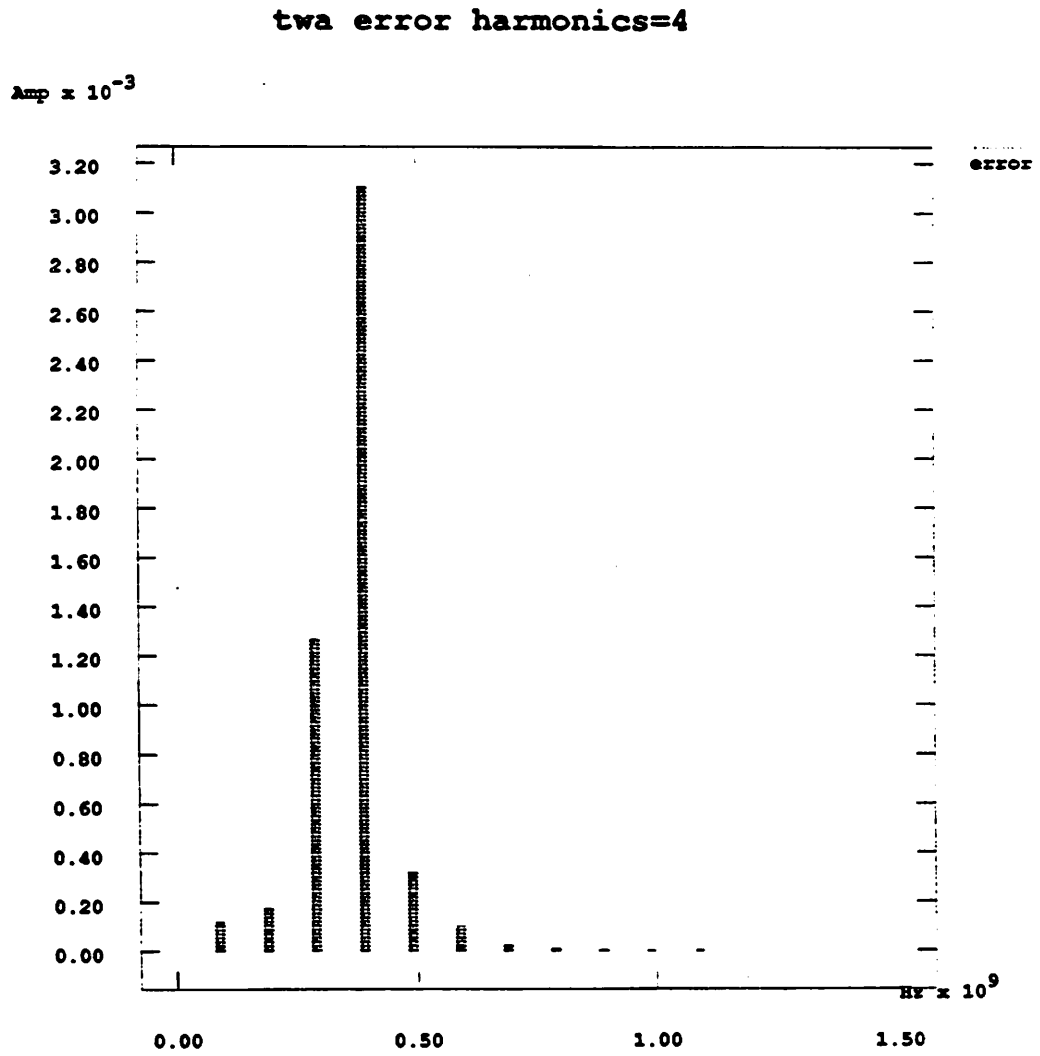


Figure 9.10 (c) Difference Between Output Spectrum of amplifier with 4 and 16 Harmonics.

the same time. Thus, the simulation plots for the almost-periodic case are not included in this thesis. The simulation results for this almost-periodic circuit are summarized in Table 9.2. As we can see from Table 9.1 and 9.2, the prediction errors are all reasonably small for both the periodic and almost periodic cases regardless of the course and inaccurate initial guess voltage used.

test	Err. (V)	Est.	Err. (V)	Err. (V)	% error
circuit	Allowed	H	Estimated	Observed	
mixer	5e-4	2	3.4e-4	1.5e-4	0.86
mixer	5e-6	4	3.5e-6	4.6e-6	0.005

Table 9.2 Comparison of Error Estimated and Observed for Almost-Periodic Nonautonomous Circuits

As mentioned before, the two important elements which greatly impact the accuracy of this algorithm are A and κ . Since both quantities are determined using the initial guess voltage generated by *Spectre*, the accuracy of this algorithm relies heavily on the accuracy of this initial guess voltage. If the initial guess voltage is close to the exact solution, the estimated error will be close to the exact error observed. Since the initial guess voltage usually consists of only the fundamental harmonic, this initial guess voltage can be far from the exact solution. If we can find a way to generate a better initial guess solution, we will not only be able to improve greatly the accuracy of this scheme but also improve the speed of convergence of the harmonic-Newton iteration at the same time.

9.4. Conclusion

In PART II of this thesis, we introduced the harmonic-Newton method for finding the steady-state solutions of nonlinear circuits. In particular, we have discussed the convergence of this method for periodic nonautonomous systems, almost periodic systems, and periodic autonomous systems when Fourier series expansion is used. We also showed that the harmonic-Newton method is convergent when DFT is used.

A complete error analysis has also been presented. We utilized the theoretical analysis by proposing an efficient scheme to estimate the number of harmonics needed prior to applying the harmonic-Newton method for a given error objective. This significantly improves the efficiency of *Spectre*, a program built on the harmonic-Newton method. Simulation results prove the robustness of this method and therefore confirm the usefulness of the theory derived.

CHAPTER 10

Conclusions

In this thesis, we looked at the circuit simulation problem from both the time domain and the frequency domain point of view. We first discussed the limitations of conventional simulation methods when time-domain transient analysis is of primary interest, and presented the motivation for using relaxation-based techniques. We then studied various relaxation-based techniques: linear relaxation, nonlinear relaxation, waveform relaxation, and timing analysis techniques, with their numerical properties summarized and problems presented.

In light of their problems, we described and formalized the Implicit-Implicit Explicit (IIE) method and studied rigorously its numerical properties. We showed that the Implicit-Implicit Explicit method is consistent, stable, and convergent even when floating capacitors are present in the circuits.

Based on the IIE method, we implemented an initial waveform generator for RELAX. The implementation results proved that the IIE method is indeed accurate and convergent. In fact, in most of our test circuits, the waveforms generated by the initial waveform generator are indistinguishable from the final solutions. However, we also found that the time step required for the IIE method is smaller than the one used by the trapezoidal method used in RELAX. This reduces the efficiency of the simulator and results in increase in CPU time.

Since transient analysis is very time consuming, it will be desirable to bypass the analysis of transient responses if steady-state solutions are of the only concern. This is especially true when the circuit under analysis has a low damping factor. This leads naturally to frequency-domain simulation since frequency-domain analysis only evaluates the steady-state waveforms. Frequency domain techniques have an additional advantage that distributed elements such as resistors, inductors, and capacitors in transmission lines are much easier to model. Thus, in the second part of this

thesis, we examined an important frequency-domain simulation technique, the harmonic-Newton method.

We first extended the harmonic-Newton method to autonomous systems and showed how we can form the new Jacobian matrix for these systems. Then, we looked at the implementation aspect of the harmonic-Newton method. In order to perform circuit analysis in the frequency domain, it is necessary to truncate the number of harmonics considered to reduce the infinite-dimensional problem into finite dimension. We proved the convergence of the harmonic-Newton method. This is important since it guarantee that the solutions obtained after truncation takes place are meaningful.

When DFT is used, not only the truncation error, but also the aliasing distortion is introduced. This distortion coupled with the additional error arose when signals are being transformed between the time domain and the frequency domain, makes the total error hard to predict. In the thesis, we also proved the convergence of the harmonic-Newton method when DFT is used and presented the corresponding theoretical bounds. Based on the theoretical findings, we implemented an error estimation scheme which can be used to choose the number of harmonics needed for a given error objective. This scheme is proved to be rebust and useful and can significantly improve the efficiency of Spectre.

REFERENCES

1. L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," *Electronics Research Laboratory Rep. No. ERL-M520*, University of California, Berkeley, (May 1975).
2. W. T. Weeks, A. J. Jimenez, G. W. Mahoney, D. Mehta, H. Qassemzadeh, and T.R. Scott, "Algorithms for ASTAP -- A Network Analysis Program," *IEEE Trans. on Circ. Theory* CT-10 pp. 628-234 (Nov, 1973).
3. A. R. Newton, "The Analysis of Floating Capacitors for Timing Simulation," *Proc. 13th Asilomar Conf. Circuit Syst. Comput.*, (Nov. 1979).
4. K. S. Kundert and A. Sangiovanni-Vincentelli, "Simulation of Nonlinear Circuits in the Frequency Domain," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems* CAD-5 pp. 521-535 (Oct 1986).
5. B. R. Chawla, H. K. Gummel, and P. Kozak, "MOTIS - a MOS timing simulator," *IEEE Trans. Circ. and Sys.* 22 pp. 901-909 (1975).
6. J. White and A. Sangiovanni-Vincentelli, "RELAX2: A New Waveform Relaxation Approach for the Analysis of LSI MOS Circuits," *Proc. Int. Symp. on Circ. and Sys.*, (May 1983).
7. A. R. Newton, "The Simulation of Large Scale Integrated Circuits," *Ph.D. dissertation Electronic Research Laboratory, Memo. no. ERL-M78/52* University of California, Berkeley, (1978).
8. R. A. Saleh, J. E. Kleckner, and A. R. Newton, "Iterated Timing Analysis and SPLICE1," *ICCAD'83 Digest*, (1983).
9. A. Sangiovanni-Vincentelli, *EE221 Lecture notes* 1982.
10. E. Lelarsmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, "The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits," *IEEE Trans. on CAD of Ic and Sys.* 1, no. 3 pp. 131-145 (July 1982).
11. J. K. Hale, *Ordinary Differential Equations*, John Wiley and Sons, Inc. (1969).
12. R. A. Varga, *Matrix Iterative Analysis*, Prentice-Hall (1969).
13. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press (1970).
14. J. White and A. L. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer Academic Publisher (1987).
15. C. W. Gear, *Numerical Initial Value Problems for Ordinary Differential Equations*, Prentice-Hall (1971).
16. G. Dahlquist and A. Bjorck, *Numerical Methods*, Prentice-Hall (1974).
17. G. De Micheli and A. Sangiovanni-Vincentelli, "Characterization of Integration Algorithms for the Timing Analysis of MOS VLSI Circuits," *Int. J. Circuit Theory Appl.*, pp. 299-309 (Oct. 1982).
18. C. A. Desoer, *Notes for a Second Course on Linear Systems*, D. Van Nostrand (1971).

19. V. Visvanathan, *Private Communication*
20. T. J. Aprille and T. N. Trick, "A Computer Algorithm to Determine the Steady-State Response of Nonlinear Oscillators," *IEEE Trans. Circuits Theory* CT-19, no. 4 pp. 354-360 (July 1972).
21. F. R. Colon and T. N. Trick, "Fast Periodic Steady-State Analysis for Large-Signal Electronic Circuits," *IEEE J. Solid-State Circuits* sc-8, no. 4(August 1973).
22. S. Skelboe, "Computation of the Periodic Stead-State Response of Nonlinear Networks by Extrapolation Methods," *IEEE Trans. Circuits and Systems* CAS-27 no. 3 pp. 161-175 (March 1980).
23. K. S. Kundert, G. B. Sorkin, and A. Sangiovanni-Vincentelli, "The Applying of Harmonic Balance to Almost-Periodic Circuits," *IEEE Trans. on Microwave Theory and Techniques* MTT-36, no. 2 pp. 366-378 (Feb. 1988).
24. H. Bohr, *Almost Periodic Functions*, Chelsea Publishing Company (1947).
25. M. Urabe, "Galerkin's Procedure for Nonlinear Periodic Systems," *Arch. Rat. Mech. and Anal.* 20 pp. 120-152 (1965).
26. A. I. Mees, *Dynamics of Feedback Systems*, John Wiley & Sons (1981).
27. Vaclav Cizek, *Discrete Fourier Transforms and Their Applications*, Adam Hilger Ltd (1986).