AN ABSTRACT OF THE DISSERTATION OF

Yutao Hu for the degree of Doctor of Philosophy in

Electrical and Computer Engineering presented on May 28, 2004

Title: Steady-State Analysis Techniques for Coupled Device and Circuit Simulation

Abstract approved: ______Redacted for Privacy______

Kartikeya Mayaram

The focus of this work is on the steady-state analysis of RF circuits using a coupled device and circuit simulator. Efficient coupling algorithms for both the time-domain shooting method and the frequency-domain harmonic balance method have been developed. A modified Newton shooting method considerably improves the efficiency and reliability of the time-domain analysis. Three different implementation approaches of the harmonic balance method for coupled device and circuit simulation are investigated and implemented. These include the quasi-static, non-quasi-static, and modified-Volterra-series approaches. Comparisons of simulation and performance results identify the strengths and weakness of these approaches in terms of accuracy and efficiency.

Steady-State Analysis Techniques for Coupled Device and Circuit Simulation

by

Yutao Hu

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented May 28, 2004
Commencement June 2005

Doctor of Philosophy dissertation of Yutao Hu presented on May 28, 2004.

APPROVED:

Redacted for Privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for Privacy

Dire[illegible]ngineering and Computer Science

Redacted for Privacy

Dean of Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Redacted for Privacy

Yutao Hu, Author

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dr. Kartikeya Mayaram for his continuous support and guidance throughout my graduate studies. I am very grateful for his patience and encouragement during the ups and downs of my Ph.D. research. His constant believing in me is one of the major motivations for me to complete this dissertation and finish my Ph.D. program. He always takes time to give me his technical and professional advice to any problem I have. He reviewed my drafts of papers and dissertation with great patience. More importantly, he is a real model of a professional researcher for me, which will benefit my career in the future.

I would like to thank Dr. Solomon C. S. Yim, Dr. Terri Fiez, Dr. Un-Ku Moon, and Dr. Raghu Settaluri for being my committee members, and for their patience, time, and help for various meetings of my Ph.D. program. I would also like to thank Dr. Chin S. Hsu for being my committee member when I was at Washington State University.

I would like to thank NSF, SRC, and DARPA for their financial support of this research.

My thanks to all the members in my lab, as well as all the former and current members of the AMS group for their help and friendship. Special thanks to Chengang Xu and Xiaochun Duan for being tolerant to my disturbances, for their knowledge, and for always being available as resources for questions and discussions.

Thanks also to the friends of the local Chinese community, volleyball club, and soccer club for their help and all the happy time we shared during the years of my Ph.D. study at Corvallis.

Finally, I would like to thank my family and my loved one, Jie, for all the patience, support, encouragement and unconditional love throughout these years.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

STEADY-STATE ANALYSIS TECHNIQUES FOR COUPLED DEVICE AND CIRCUIT SIMULATION

Chapter 1

INTRODUCTION

1.1 Motivation and Contributions

Simulation is very important in the design of modern integrated circuits, devices, and processes. It provides an efficient way to explore the design space without time-consuming and costly fabrication cycles. With the continued increasing demand for radio frequency (RF) ICs there is a critical need for accurate and efficient simulation of circuits in the steady state. Certain aspects of system performance are easier to characterize and verify in the steady state. Examples of these are large-signal distortion, power, frequency, noise, and transfer characteristics such as gain and impedance.

Steady-state analyses are available in several commercial circuit simulators. However, not much attention has been given to the model accuracy. For RF applications, distributed device effects are important and must be included in simulations via accurate device models. Otherwise the simulated circuit performance will deviate significantly from the actual circuit. In the absence of accurate compact models, coupled or mixed-level circuit and device simulators [1-4] can be used. These simulators provide a framework in which physical (numerical) models for critical devices can be simulated in conjunction with standard circuit-level compact models. To enable accurate simulation of RF circuits, steady-state analyses are required in coupled device and circuit simulators.

The steady-state responses of RF circuits can be obtained by using a conventional transient simulation from some arbitrary initial condition until the transient behavior dies out. However, this straightforward approach can take a long time to be practical for RF ICs. There are a variety of methods that directly compute the steady-state solution more efficiently than the conventional transient simulation. These methods can be classified as frequency-domain and time-domain methods [5]. Harmonic balance is a frequency-domain method in which the coefficients for a truncated Fourier series expansion of the steady-state solution are determined. In the time domain, the shooting method is a popular method to find an initial condition which leads to the steady-state solution.

These two methods have been successfully implemented in circuit-level simulators with only compact models. However, prior to this work they were not available in general coupled device and circuit simulators. The harmonic balance method has previously been applied for the simulation of semiconductor devices in [6, 7]. The emphasis there was on device internal behavior and the analysis was not extended to general coupled device and circuit simulation. This work presents the first implementation of time-domain shooting method and frequency-domain harmonic balance method in the context of coupled device and circuit simulation [8-12].

The focus has been on a general coupled device and circuit simulator – CODECS [2]. In the implementation, the original architecture of CODECS is exploited and the existing coupling simulation framework is used. With critical devices modeled by physical (numerical) models, this simulator can obtain the steady-state solution accurately and efficiently for RF ICs.

For the time-domain method, several modifications and heuristics have been applied to improve the reliability and convergence of the Newton shooting algorithm. A new scheme to bias the numerical devices is implemented to improve the efficiency. A circuit unknown implementation is developed to eliminate the inconsistency and discontinuity in the standard state-based implementation.

Three different implementation approaches of the harmonic balance method for coupled device and circuit simulation are investigated and implemented. These include the quasi-static (QS), non-quasi-static (NQS), and modified-Volterra-series (MVS) approaches. Comparisons of simulation and performance results identify that there is a tradeoff between simulation accuracy and complexity among these approaches. The complex computation of the linearized terminal parameter in the MVS approach is eliminated by using a reasonable approximation. In the QS approach, the computation of the static charge (not available from the device simulator) is avoided by an equation reformulation. A new biasing scheme for numerical devices is proposed that significantly improves the performance of the QS and MVS approaches.

1.2 Dissertation Outline

This dissertation is organized as follows. The current chapter presents the motivation and contributions of this work and outlines the organization of the dissertation. In Chapter 2 the basic RF IC building blocks are reviewed and several different types of steady-state solutions are defined. Three methods to compute the steady state for RF circuits are introduced, along with their advantages and drawbacks. Mixed-level simulation is reviewed in Chapter 3. The circuit levels and device levels of simulation are introduced. Then the coupling algorithms used in CODECS are described and the architecture is illustrated.

The time-domain shooting method and its implementation are presented in Chapter 4. Modifications and heuristics are introduced to improve the reliability and convergence of the Newton shooting method. For coupled device and circuit simulation, a new biasing scheme for numerical devices to improve the efficiency and reliability is described. A circuit unknown implementation of the shooting method is proposed to eliminate the inconsistency and discontinuity presented in state-based implementation.

Chapter 5 first presents an overview of the harmonic balance algorithm. Various harmonic truncation schemes are introduced. The basic theory on frequency remapping is discussed and the remapping scheme is illustrated by an example. The formulation of the harmonic balance equations and the resulting system matrices are shown in detail. To solve the autonomous systems, the standard harmonic balance method is modified and the concept of voltage probe is used to improve convergence.

In Chapter 6, the harmonic balance method for coupled device and circuit simulation is described in the context of CODECS. Three different implementation approaches are introduced. These include quasi-static (QS), non-quasi-static (NQS), and modified Volterra series (MVS) approaches. Qualitative comparison between these three approaches is made.

In Chapter 7, first example circuits with 1D and 2D numerical models have been chosen to demonstrate accurate and efficient steady-state simulation in CODECS. Then, the qualitative comparison between the three implementation approaches for the harmonic balance method identified in Chapter 6 is verified by examples. Finally, a performance comparison of the time-domain shooting method and the QS harmonic balance method for coupled simulation is presented. Conclusions and future work are summarized in Chapter 8.

Chapter 2

STEADY-STATE ANALYSIS FOR RF ICS

2.1 Basic Radio Frequency (RF) IC Building Blocks

Radio frequency (RF) integrated circuits are used in high-frequency communication applications. Figure 2.1 shows the configuration of a simple heterodyne receiver [13]. In this receiver, the signal received from the antenna is first filtered and amplified to get the desired RF signal. Then the first mixer down-converts the RF signal to the intermediate frequency (IF) by mixing with the frequency of the first local oscillator (LO). After filtering and amplification, the signal is down-converted again by the second mixer to the base band. From this example, we can see that RF front ends are constructed primarily of four basic building blocks. They are filters, amplifiers, mixers, and oscillators.

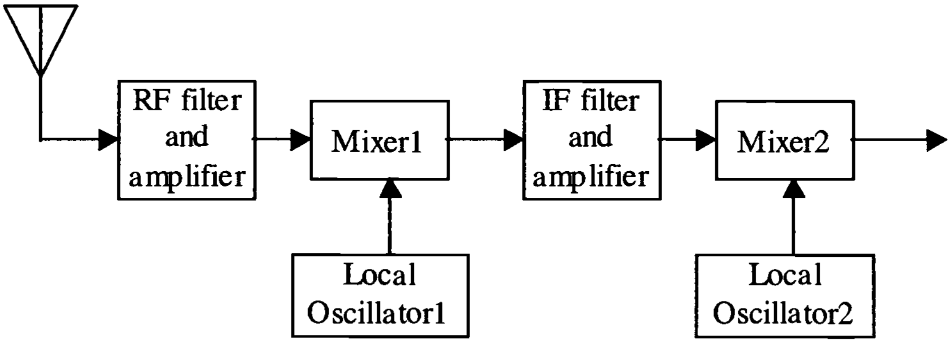


Figure 2.1: The simple heterodyne receiver configuration.

2.1.1 Filters

In a RF application, a high-frequency carrier is modulated by the original information signal. To obtain this signal at the specified bandwidth around the high carrier frequency, band-pass filters are used to filter out the undesired signal such as interference and noise. In this section a parallel RLC band-pass filter is used as an example to illustrate the filter design.

A simple parallel RLC circuit is shown Figure 2.2. The admittance of the RLC tank in s-domain is given by:

$$Y(s) = G + sC + \frac{1}{sL} \tag{2.1}$$

where $G = \frac{1}{R}$. For a sinusoidal input current $i_{in}(t)$ of frequency ω, when the output voltage $v_{out}(t)$ reaches the steady state, the frequency-domain admittance of the RLC tank is given by substituting $s = j\omega$ into Equation (2.1):

$$Y(\omega) = G + j\omega C + \frac{1}{j\omega L} = G + j(\omega C - \frac{1}{\omega L}) \tag{2.2}$$

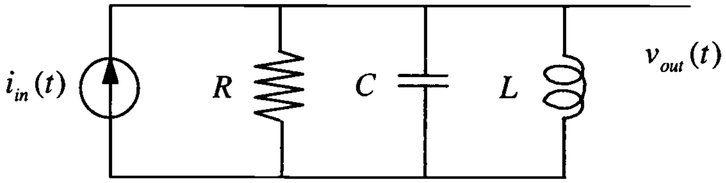


Figure 2.2: The parallel RLC band-pass filter.

From Equation (2.2) we can see that at a certain frequency the inductive and capacitive admittances cancel each other. This frequency is called the resonance frequency ω_0 and is given by:

$$\omega_0 C - \frac{1}{\omega_0 L} = 0 \quad \Rightarrow \quad \omega_0 = \frac{1}{\sqrt{LC}} \tag{2.3}$$

At ω_0, Y in Equation (2.2) is real and reaches its minimum magnitude given by G. Another parameter describing the tuned circuit is the quality factor Q defined as:

$$Q = \omega \cdot \frac{energy\ stored}{average\ power\ dissipated} \tag{2.4}$$

For the parallel RLC circuit, the Q factor at resonance is given by:

$$Q = \omega_0 RC = \frac{R}{\omega_0 L} = \frac{R}{\sqrt{L/C}} \tag{2.5}$$

The frequency response of the parallel RLC circuit is given by $Z(\omega) = \frac{1}{Y(\omega)}$ where $Y(\omega)$ is from Equation (2.2). The magnitude of $Z(\omega)$ is plotted in Figure 2.3, where the bandwidth of this band-pass filter is ω_b [14]. From a simple derivation, the bandwidth for the parallel RLC circuit is given as:

$$\omega_b = \frac{1}{RC} \tag{2.6}$$

Considering Equations (2.5) and (2.6) we have a relation between the bandwidth and the Q factor:

$$\frac{\omega_b}{\omega_0} = \frac{1}{Q} \qquad (2.7)$$

A higher Q factor means a narrower bandwidth and the filter has better frequency selectivity.

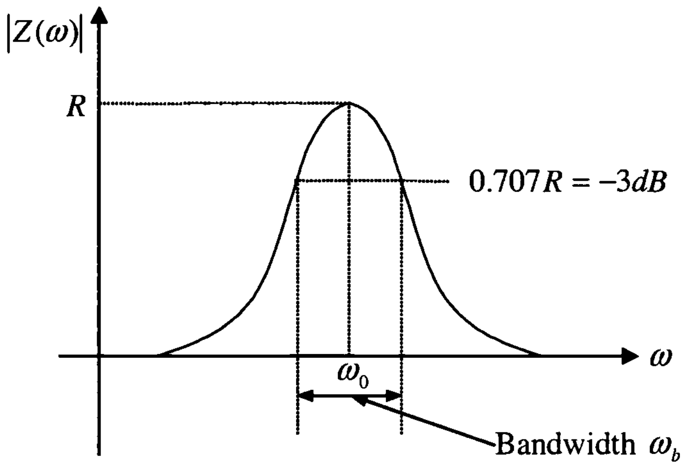


Figure 2.3: The magnitude of $Z(\omega)$ for the parallel RLC band-pass filter.

2.1.2 Amplifiers

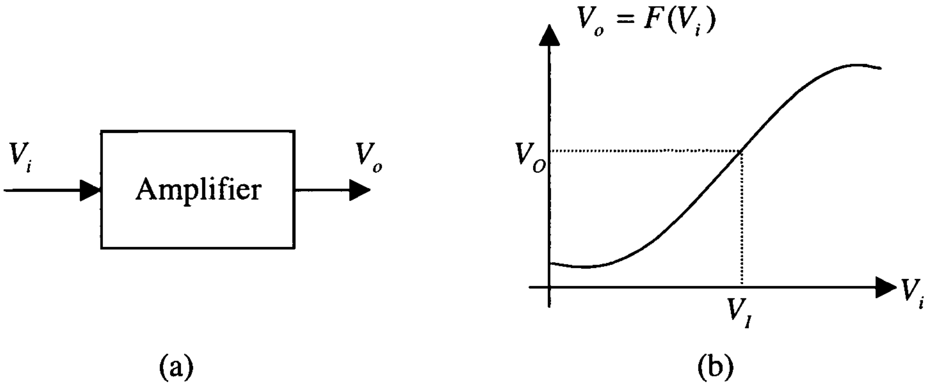
In analog and RF applications, amplifiers are used to increase the amplitude of the signal without introducing distortion. Therefore, a good amplifier should provide gain with high linearity. In reality, all amplifiers are nonlinear. Nonlinearities generate harmonic distortion for a sinusoidal input signal.

To characterize the nonlinearity and harmonic distortion, let us consider a general amplifier configuration in Figure 2.4(a) with the transfer characteristic shown in Figure 2.4(b). For low frequency analysis, all capacitors in the amplifier are ignored and there are no energy-storage elements. The output can be written as a function of the input and expressed as a power series [14]:

$$V_O + v_o = F(V_I + v_i) = a_0 + a_1 v_i + a_2 v_i^{\,2} + a_3 v_i^{\,3} + \cdots \tag{2.8}$$

where $a_0 = F(V_I) = V_O$ is the quiescent output voltage when $v_i = 0$. Equation (2.8) can be rewritten as:

$$v_o = a_1 v_i + a_2 v_i^{\,2} + a_3 v_i^{\,3} + \cdots \tag{2.9}$$



(a)

(b)

Figure 2.4: (a) A general amplifier configuration and (b) its transfer characteristic.

We now assume the input voltage v_i is a pure sinusoidal signal $V_{iA}\cos\omega_1 t$. Substituting $v_i = V_{iA}\cos(\omega_1 t)$ into Equation (2.9), we obtain:

$$\begin{aligned} v_o &= a_1(V_{iA}\cos\omega_1 t) + a_2(V_{iA}\cos\omega_1 t)^2 + a_3(V_{iA}\cos\omega_1 t)^3 + \cdots \\ &= \frac{a_2 V_{iA}}{2} + (a_1 V_{iA} + \frac{3}{4}a_3 V_{iA}^{\,3})\cos\omega_1 t + \frac{a_2 V_{iA}}{2}\cos 2\omega_1 t + \frac{a_3 V_{iA}}{4}\cos 3\omega_1 t + \cdots \end{aligned} \tag{2.10}$$

Equation (2.10) can be rewritten as:

$$v_o = b_0 + b_1\cos\omega_1 t + b_2\cos 2\omega_1 t + b_3\cos 3\omega_1 t + \cdots \tag{2.11}$$

where:

$$b_0 = \frac{a_2 V_{iA}}{2} \cdots$$

$$b_1 = a_1 V_{iA} + \frac{3}{4} a_3 V_{iA}{}^3 + \cdots \approx a_1 V_{iA}$$

$$b_2 = \frac{a_2 V_{iA}}{2} + \cdots$$

$$b_3 = \frac{a_3 V_{iA}}{4} + \cdots \tag{2.12}$$

Form Equations (2.10) and (2.11) we can see that due to the nonlinearity of the amplifier, harmonics of the input frequency are generated at the output. To reduce the harmonics at the output, usually a band-pass filter tuned at the desired frequency is connected at the output of the amplifier. The overall configuration is a band-pass amplifier.

To quantify the levels of the distortion present in the output signal, the nth order harmonic distortion factors HD_n are widely used. HD_n is defined to be the magnitude of the ratio of the nth harmonic to the fundamental. For v_o in Equation (2.11) the 2nd and 3rd harmonic distortion factors are given as:

$$HD_2 = \frac{|b_2|}{|b_1|} \approx \frac{a_2 V_{iA}{}^2}{2} \Big/ a_1 V_{iA} = \frac{1}{2} \frac{a_2}{a_1} V_{iA}$$

$$HD_3 = \frac{|b_3|}{|b_1|} \approx \frac{1}{4} \frac{a_3}{a_1} V_{iA}{}^2 \tag{2.13}$$

To characterize the level of distortion in the entire waveform, the total harmonic distortion (THD) is defined to be:

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty}\left|b_n\right|^2}}{\left|b_1\right|} = \sqrt{\sum_{n=2}^{\infty}\left|HD_n\right|^2} \qquad (2.14)$$

2.1.3 Mixers

As shown in Figure 2.1, the mixer is the heart of the simple heterodyne receiver and has two inputs and one output. One input is for the information signal and the other is for the local oscillator, LO. The mixer translates the information signal from one frequency range to another. It combines signals at two frequencies and produces an output at a third frequency. Ideally, the signal at the output is the input information signal shifted in frequency by an amount equal to the frequency of the LO. Since linear time-invariant networks can not produce spectral components not present in the input, we have to rely on nonlinear or time varying networks to provide the mixing function.

Usually the information input signal is small and the LO is a large periodic signal. For best performance, mixers are designed to respond in a strongly nonlinear fashion to the LO. Thus, mixers behave both near linearly (to the information input) and strongly nonlinearly (to the LO). Since the LO signal is independent of the information signal, it can be considered to be part of the circuit rather than an input to the circuit. As shown in Figure 2.5, the nonlinear mixer circuit driven by the LO can be treated as a linear periodically time-varying circuit [15]. Then the mixer has a single input and a linear periodically time-varying transfer function.

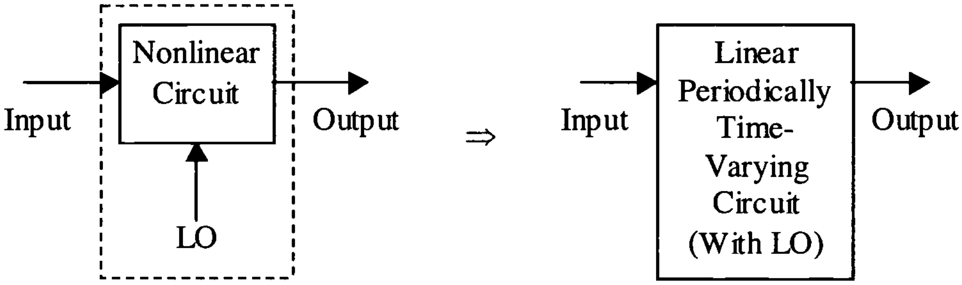


Figure 2.5: The nonlinear mixer circuit driven by the LO can be treated as a linear periodically time-varying circuit.

Using the linear time-varying concept for mixer analysis, the mixer can be represented as:

$$i_{out} = g(t)v_{in} \tag{2.15}$$

where i_{out} is the output current, v_{in} is the input voltage, and $g(t)$ is the periodically time-varying transconductance of the mixer. Since $g(t)$ is due to the LO at a frequency ω_{LO}, we can represent it as:

$$g(t) = g_0 + g_1 \cos \omega_{LO} t + g_2 \cos 2\omega_{LO} t + \cdots \tag{2.16}$$

For $v_{in}(t) = V_1 \cos \omega_{RF} t$ we have from Equations (2.15) and (2.16):

$$i_{out}(t) = (g_0 + g_1 \cos \omega_{LO} t + g_2 \cos 2\omega_{LO} t + \cdots) \cdot V_1 \cos \omega_{RF} t$$

$$= g_0 V_1 \cos \omega_{RF} t + \frac{g_1 V_1}{2} \cos(\omega_{LO} \pm \omega_{RF})t + \frac{g_2 V_1}{2} \cos(2\omega_{LO} \pm \omega_{RF})t + \cdots \tag{2.17}$$

If the component of $i_{out}(t)$ at frequency $\omega_{IF} = \omega_{LO} - \omega_{RF}$ is the desired output, the conversion transconductance is given as:

$$G_{conv} = \frac{|IF\ current\ output|}{|RF\ voltage\ input|} = \frac{g_1 V_1/2}{V_1} = \frac{g_1}{2} \tag{2.18}$$

The frequency translation behavior of the mixer described by Equations (2.15)-(2.17) can be illustrated in the frequency domain by Figure 2.6 [15]. The input is a modulation signal at low frequency. It is up-converted to sidebands above and below each harmonic of the LO. At the output of the mixer, only one sideband is of interest. Therefore, a high Q bandpass filter is used after the mixer to remove all but the desired sideband.

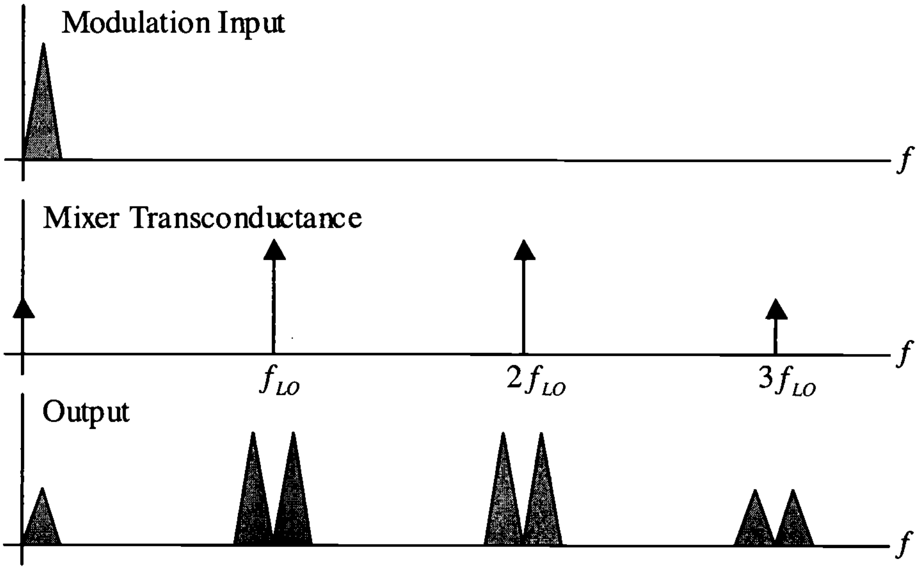


Figure 2.6: The frequency translation behavior of an up-conversion mixer. The input modulation signal is up-converted to the sidebands above and below each harmonic of the LO.

2.1.4 Oscillators

In Figure 2.1, one input of the mixer is driven by a local oscillator. Oscillators generate a periodic reference signal at a particular frequency. This signal is used in the mixer to realize the frequency conversion. In some oscillators, referred to as voltage-controlled oscillators (VCO), the frequency of the output varies proportionally to some input signal. To generate a periodic output, the oscillator circuit must have a self-sustaining mechanism that allows its own noise to grow and eventually become a periodic signal. There are two analysis points of view for oscillators, the negative resistance approach and the feedback approach [14].

An ideal oscillator is a simple LC tank without losses as shown in Figure 2.7(a). Its output voltage is given by:

$$v(t) = V\sin(\omega_o t + \varphi) \tag{2.19}$$

where $\omega_o = \frac{1}{\sqrt{LC}}$, V and φ are determined by the initial conditions. However, a real tank circuit has losses represented by the resistance R in Figure 2.7(b). The tank by itself does not oscillate indefinitely because some of the stored energy is dissipated in R for every cycle. To sustain oscillations this loss of energy must be compensated. In the negative resistance approach an active network is used to generate a resistance equal to $-R$. Then the equivalent parallel resistance seen by the pure LC tank is infinite and the whole circuit can be viewed as a lossless LC tank. In essence, the energy lost in R is replenished by the active circuit in every cycle.

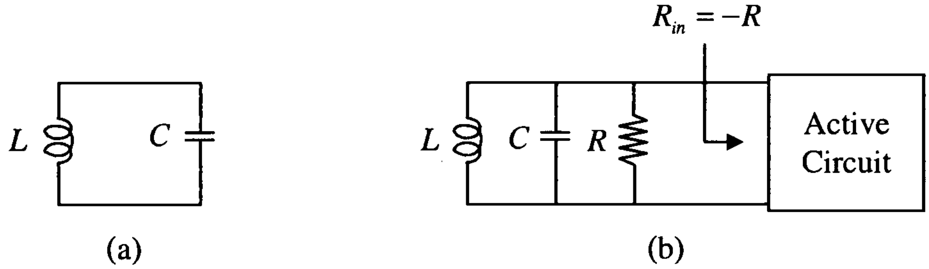


Figure 2.7: (a) An ideal lossless LC tank oscillator, (b) Negative resistance from an active circuit compensates the loss resistance R so that oscillations are sustained.

In the feedback approach, the oscillators are viewed as circuits with feedback. For the simple linear feedback system shown in Figure 2.8(a) [16], its overall transfer function is given as:

$$\frac{Y(s)}{X(s)} = \frac{H(s)}{1-H(s)} \tag{2.20}$$

For some $s = s_o$, if $H(s_o) = +1$, then the oscillation builds up. The oscillation reaches its steady state if s_o is purely imaginary such that $H(s_o = j\omega_o) = +1$. To stabilize the frequency and eliminate harmonic components, a frequency selective network such as a LC tank can be included in the system. Figure 2.8(b) shows the case when the frequency selective network is included in the feedback path.

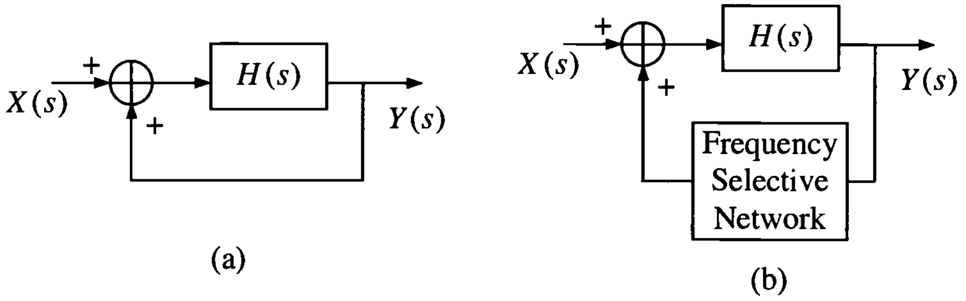


Figure 2.8: (a) Oscillator with simple linear feedback. (b) The frequency selective network is included in the feedback path.

2.2 Steady State

With the continued increasing demand for RF ICs there is a critical need for accurate and efficient simulation of circuits in the steady state. Certain aspects of system performance are easier to characterize and verify in steady state. Examples of these are distortion, power, frequency, noise, and transfer characteristics such as gain and impedance. In the following, we define the meaning of the steady state solutions and introduce several different types of steady state.

A steady state solution of a system of differential equations is a solution that is asymptotically approached as the effect of the initial conditions dies out [5]. In general, a given system of differential equations may not have a steady-state solution, a single steady-state solution, or many such solutions. If there are multiple steady-state solutions, the steady state that is asymptotically approached will depend on the initial conditions. Most practical RF circuits will have at least one steady-state solution [6].

In general, the steady state can be classified in several different types. The DC steady state is an equilibrium point of the circuit and can be obtained by DC analysis. The AC steady state represents the sinusoidal waveforms of the linear circuit driven by sinusoidal inputs. For nonlinear circuits, the system should be first linearized around a DC solution. Obviously this corresponds to the regular small-signal AC analysis.

If the steady-state solution of a circuit is periodic, it is called a periodic steady state. The spectrum of the periodic steady state consists of the components at DC, fundamental frequency, and corresponding harmonics. In the time domain, the periodic steady state can be represented by the solution state vector $x(t)$ which satisfies the periodicity constraint:

$$x(t+T) = x(t) \tag{2.21}$$

where $-\infty < t < \infty$ and T is the period. In the frequency domain, due to the harmonically related components, the periodic steady state can be represented by a Fourier series. For a specified accuracy, the Fourier series with a finite number of terms can be used.

If a nonlinear circuit is driven by several periodic sources at unrelated frequencies, the circuit will have a quasiperiodic steady-state response. The spectrum of the quasiperiodic steady state consists of the components at the frequencies which are the linear combination of the fundamental frequencies of the inputs. It is natural to represent the quasiperiodic signal in the frequency domain. Theoretically if time-domain waveforms are used to represent the quasiperiodic signal, the time interval should be infinite.

The quasiperiodic steady state reduces to a periodic one if the nonlinear circuit is driven by only one periodic input or the multiple periodic inputs are harmonically related. If the circuit is linear or linearized for an infinitesimally small input, the periodic steady state becomes an AC steady state. Further, if the periodic input is zero, then it becomes a DC steady-state problem. Since the steady state with large-signal inputs is of the interest in this work, by default “steady state” will refer to quasiperiodic and/or periodic steady state.

2.3 Computing the Steady State

In RF circuits, large dynamic elements such as large inductors, large capacitors, and high-Q filters are typically used to realize the desired functions. These elements introduce very long time constants in the circuits which make the circuits approach the steady state very slowly. Several methods to compute the steady state for RF circuits are introduced in the following subsections.

2.3.1 Conventional Transient Simulation

Conventional transient simulation numerically integrates the circuit differential equations from an initial condition. The simulation can be run long enough such that the effect of the initial condition dies out. For RF circuits, due to high-Q filters and wide frequency separations, this simulation will be too long to be practical. In these cases, the time interval over which the simulation should be carried out is determined by the large time constants or the lowest frequencies, while the time-steps used by the numerical integration algorithm are limited by the highest frequency. Therefore, a large number of time points have to be simulated. To overcome the difficulty with the conventional transient simulation, a variety of methods have been derived that compute the steady-state solution more efficiently. For RF circuits, the most attractive ones are the time-domain shooting method and the frequency-domain harmonic balance method.

2.3.2 Time-Domain Shooting Method

In the time domain, finding the periodic steady-state solution of a circuit can be recast to finding the initial condition for the circuit's differential equations such that the solution at the end of one period matches the initial condition [17-20, 5]. This can be formulated as a two-point boundary-value problem where the solution over the interval $[0, T]$ is required to satisfy the periodic boundary constraint:

$$x(0) = x(T) \tag{2.22}$$

where $x(0)$ is the initial value and $x(T)$ is the final value. In shooting methods, first the circuit is simulated for one period using some guess for the initial condition. Then the final value $x(T)$ is checked with the guess for the initial value $x(0)$. If they don't match each other, the initial value is adjusted. The circuit is simulated again for one period with the adjusted initial value. This process is repeated until the initial value and the final value are in close agreement. The time-domain shooting method is explained further in Chapter 4.

Usually the shooting method uses the transient analysis over one period to obtain the final value at the end of the period. Compared with the conventional transient simulation method, the shooting method can be viewed as a transient simulation that is accelerated for the circuit to approach the periodic steady state. The acceleration is obtained by adjusting the initial condition at the beginning of each one-period transient simulation. Although the shooting method overcomes the problem due to the large time constants, it still has difficulty in handling circuits with wide frequency separation as does the conventional transient simulation. The time-domain shooting method can only solve periodic steady-state problems and can not, in general, handle the quasiperiodic ones. The frequency-domain harmonic balance method introduced in the following subsection tackles the above problems naturally.

2.3.3 Frequency-Domain Harmonic Balance Method

The basic idea behind the frequency-domain harmonic balance method [5] is to represent the circuit waveforms by the Fourier sine and cosine series. As mentioned in Section 2.2, for a specified accuracy, practically the Fourier series with a finite number terms is sufficient to represent the periodic and quasiperiodic steady states. The advantage of the frequency-domain representation of signals is that for signals with wide frequency separation or unrelated frequencies only the frequency bins with significant energy need to be chosen to approximate the signals. Thus the actual values of frequencies are not important and the difficulty in simulating and representing quasiperiodic signals using the time-domain methods as described in Section 2.3.1 is avoided.

Linear dynamic operations such as differentiation and integration are transformed to simple algebraic operations in the frequency domain — multiplication and division by the frequency, respectively. Therefore, nonlinear integro-differential equations that describe a circuit in the time domain are replaced by a system of nonlinear algebraic equations in terms of the Fourier coefficients. For this reason, the coefficients of the steady-state solution are an algebraic function of the coefficients of the stimulus and the dynamic aspects of the time-domain steady-state problem are eliminated. The natural approximation of signals as Fourier series guarantees that the solution of the nonlinear algebraic equations is indeed the periodic or quasiperiodic steady state of the system. The frequency-domain harmonic balance method is described in detail in Chapter 5.

A weakness of the harmonic balance method relative to the time-domain methods is its potentially poor handling of strongly nonlinear circuits because their responses have a rich frequency spectrum. These responses are difficult to represent efficiently with sinusoidal basis functions in the frequency domain.

2.4 Summary

In this chapter the basic RF IC building blocks are reviewed. These include filters, amplifiers, mixers, and oscillators. Their basic functions and methods of analysis were described.

Several different types of steady-state solutions are described. Among them, the periodic and quasiperiodic steady states are of interest in this work. Three methods to compute the steady state for RF circuits were introduced. The difficulty with the conventional transient simulation method is discussed. The time-domain shooting method and the frequency-domain harmonic balance method which are implemented in this work are briefly described, along with their advantages and drawbacks.

Chapter 3

MIXED-LEVEL SIMULATION AND CODECS

3.1 Introduction

Simulation is a very important step in the design of modern integrated circuits, devices, and processes. It provides an efficient way to explore the design space without time-consuming and costly fabrication cycles. There are various levels of simulations currently available. From the detailed to the high level, they are IC process simulation, device simulation, and circuit simulation. A simulator that combines two or more levels of simulations is called a mixed-level simulator [1-4]. The key idea behind a mixed-level simulator is the use of detailed forms of simulation on the critical parts of a circuit, to get precise waveform information, and less accurate but faster forms of simulations for the rest of the circuit. Clearly, there is a tradeoff between the accuracy of the simulations and the runtime. The mixed-level circuit and device simulator CODECS is introduced in this chapter.

In the coupled device and circuit simulator (CODECS) [2], devices can be simulated under realistic dc and time-dependent boundary conditions imposed by the surrounding circuit in which they are embedded. Conventional device-level simulation typically allows only voltage or current boundary conditions for a device; and, hence, cannot account for circuit embedding. Since the doping profile and the geometry of the device is supplied to CODECS, the simulator provides a direct link between technology and circuit performance and a predictive capability at the circuit level.

The simulation environment of CODECS enables one to model critical devices within a circuit by physical (numerical) models based upon the solution of Poisson's equation and the current-continuity equations. Analytical models can be used for the noncritical devices. The numerical models in CODECS include physical effects such as bandgap narrowing, Shockley-Hall-Read and Auger recombinations, concentration- and field-dependent mobilities, concentration-dependent lifetimes and avalanche generation. CODECS supports dc, transient, small-signal ac, and pole/zero analyses of circuits containing one- and two-dimensional numerical models for diodes and bipolar transistors, and two-dimensional numerical models for MOSFETs. In addition, dc and transient sensitivities to doping profiles can be computed at the device level. However, steady-state analyses are not available in CODECS which are essential for the accurate simulation of RF circuits. In this work, we extend the capabilities of CODECS using the time-domain shooting method and the frequency-domain harmonic balance method for accurate RF IC steady-state simulation.

This chapter is organized as follows. An overview of the circuit simulation problem and semiconductor device modeling for circuit-level simulation is provided in Section 3.2. The semiconductor device-level simulation and its solution methods are described in Section 3.3. In Section 3.4, the algorithms used to couple the device-

and circuit-level simulators in CODECS are presented and the architecture of CODECS is illustrated.

3.2 Circuit Simulation and Modeling

Circuit-level simulation is one important component of a mixed-level circuit and device simulation. This section provides an overview of the circuit-simulation problem. The nonlinear dc, transient, and small-signal ac analyses are introduced. These analyses are the building blocks for the steady-state analyses described later. Modeling of semiconductor devices plays an important role in circuit simulation. The simulation results are reliable only if accurate models are used for the devices. Commonly used approaches to modeling of semiconductor devices can be classified as: analytical models, empirical/table models, and numerical models. The modeling task involves use of a particular technique to model accurately the static (dc) and dynamic (transient) operation of the device. These approaches result in models with differing accuracy and speed of evaluation.

3.2.1 Circuit Simulation Problem

Circuit simulation involves assembling of the circuit equations and then solving these equations to obtain the solution of the output variables. The assembly of circuit-level equations is based on Kirchoff's current and voltage laws along with the branch-constitutive relations for each element in the circuit [21]. Modified Nodal

Analysis (MNA) [22, 23] is the most popular circuit-level equation assembly approach. MNA accommodates non-nodal elements such as independent voltage sources and yields a smaller system of equations. The unknowns from the MNA approach are the nodal voltages and currents through inductors and voltage sources.

A. *DC and Transient Analyses*

DC operating point analysis can be seen as a special case of the general transient problem. The dc operating point solution provides the initial condition for transient analysis and is computed before starting the transient simulation.

The dynamic response of a circuit can be described by a system of nonlinear differential algebraic equations obtained from the equation-assembly phase. These can be represented as:

$$f(\dot{z}(t), x(t), u(t)) = 0$$

$$z(t) = g(x(t)) \tag{3.1}$$

where x is the vector of unknowns, u is the excitation vector, z is the vector of capacitor charges and inductor fluxes, f is a nonlinear function obtained from the MNA formulation of the circuit equations, and g is a nonlinear function that relates the capacitor charges and inductor fluxes to the capacitor voltages and inductor currents, respectively. To obtain the dc operating point, we set $\dot{z} = 0$ and obtain the following nonlinear algebraic equations:

$$f(0, x^*, u) = 0 \tag{3.2}$$

The solution vector x^* is called the dc operating point solution of the circuit. The capacitor charges and inductor fluxes at the dc operating point, z^*, are given by:

$$z^* = g(x^*) \tag{3.3}$$

Use of Newton's method to solve the nonlinear algebraic Equation (3.2) results in the solution of a linear system of equations for each iteration and is given by:

$$J(x^k)\Delta x^{k+1} = -f(x^k) \tag{3.4}$$

where $J(x^k) = \left[\frac{\partial f}{\partial x}\right]^k$ is the Jacobian matrix, $\Delta x^{k+1} = x^{k+1} - x^k$, and k is the iteration number. Equation (3.4) is solved for Δx^{k+1} at each iteration and the new solution is given by $x^{k+1} = x^k + \Delta x^{k+1}$. When $\left|\Delta x^{k+1}\right|$ and $\left|f(x^{k+1})\right|$ are less than a user-specified error tolerance, the Newton iteration converges and the dc solution is obtained. Alternatively, Equation (3.4) can be written as:

$$x^{k+1} = x^k - \left[J(x^k)\right]^{-1} f(x^k) \tag{3.5}$$

The linear equations (3.4) or (3.5) can be solved by Gaussian elimination or LU decomposition. In general, the system of circuit equations is very sparse and sparse-matrix techniques are used [24].

For transient analysis, the simulation interval is divided into time points and Equation (3.1) is solved at each of these time points. At time point t_{n+1} Equation (3.1) can be expressed as:

$$f(\dot{z}_{n+1}, x_{n+1}, u_{n+1}) = 0$$

$$z_{n+1} = g(x_{n+1}) \tag{3.6}$$

Using integration formulas where time derivatives are replaced by discretized approximations [22, 24], $\dot{z}_{n+1}$ can be expressed in terms of the values of the unknown x at the present and previous time points. This step is called time discretization. For the backward-differentiation formula (BDF) of order k, $\dot{z}_{n+1}$ is given by:

$$\dot{z}_{n+1} = \frac{1}{h_n}\sum_{i=0}^{i=k} z_{n+1-i} = \frac{1}{h_n}\sum_{i=0}^{i=k} g(x_{n+1-i}) \tag{3.7}$$

Substituting $\dot{z}_{n+1}$ from Equation (3.7) into Equation (3.6), we obtain a system of nonlinear algebraic equations. These can be expressed in a general form as:

$$\tilde{f}(x_{n+1}) = 0 \tag{3.8}$$

The nonlinear equations can also be solved using Newton's method as in Equation (3.2) for the dc operating point analysis. Once the Newton iteration converges, the solution at the present time point t_{n+1} is obtained. Then a new time step is selected and the equations are solved at the new time point in a similar manner. The time step should be selected small enough such that the error in time discretization due to the integration formula is less than a user-specified error tolerance.

B. Small-Signal AC Analysis

Small-signal ac analysis is used to determine the response of a circuit to a small frequency-domain input at a dc operating point. The magnitude of the input is so small that the operation of the nonlinear circuit is approximately linear around the dc operating point and no significant harmonics are generated.

Assume the dc operating point is given by x_0 and u_0 for the circuit equation (3.1). The small input sinusoidal perturbation is given by $u = u_0 + \tilde{u}e^{j\omega t}$ and the corresponding solution is $x = x_0 + \tilde{x}e^{j\omega t}$, where ω is the frequency of the source. Equation (3.1) can be expressed by its Taylor series expansion around the dc operating point. As mentioned earlier the magnitude of the input is small enough such that the operation of the circuit is approximately linear. So we can omit the higher order terms and only keep the linear terms in the Taylor series:

$$f(\dot{z}, x, u) = f(0, x_0, u_0) + \frac{\partial f}{\partial \dot{z}}\dot{z} + \frac{\partial f}{\partial x}(x - x_0) + \frac{\partial f}{\partial u}(u - u_0) = 0 \qquad (3.9)$$

$f(0, x_0, u_0)$ is zero since x_0 is the dc operating point solution and the linearized equations are given by:

$$\frac{\partial f}{\partial \dot{z}} g'(x_0) j\omega\tilde{x} + \frac{\partial f}{\partial x}\tilde{x} + \frac{\partial f}{\partial u}\tilde{u} = 0 \qquad (3.10)$$

The small signal ac solution $\tilde{x}$ is given by:

$$\left[j\omega \frac{\partial f}{\partial \dot{z}} g'(x_0) + \frac{\partial f}{\partial x} \right]\tilde{x} = -\frac{\partial f}{\partial u}\tilde{u} \qquad (3.11)$$

Notice that $g'(x_0) = \left.\frac{\partial g}{\partial x}\right|_{x_0}$ is the linearized capacitance and inductance and $\left.\frac{\partial f}{\partial x}\right|_{x_0}$ is the linearized conductance at the dc operating point.

3.2.2 Semiconductor Device Modeling for Circuit Simulation

In the equation-assembly phase of circuit simulation, branch-constitutive relations are required that model the semiconductor devices. Device models describe the physical operation of a device and a relationship between the terminal currents and the terminal voltages is provided. There is always a tradeoff between the accuracy of a model and the computation time required for model evaluation, which includes the calculation of the equivalent conductances, capacitances, charges, and terminal currents. A simulation is accurate only if the models used for the semiconductor devices are accurate. An inaccurate model results in incorrect simulation results. Therefore, modeling is very critical for circuit simulation and this section addresses the different approaches used in modeling.

A. Analytical Models

Analytical models are based on a set of analytical equations from which closed-form expressions can be obtained for the terminal characteristics. These analytical equations are derived from an understanding of the physics of device operation under some restricted conditions. The drift-diffusion equations are used to obtain the current-voltage characteristics under dc conditions. Empirical parameters are introduced to model higher-order physical effects. A model that is used for a particular device is characterized by a set of parameters called the model parameters. These parameters are the various constants that appear in the closed-form expressions relating the terminal currents to the terminal voltages. Their values are determined form the measured characteristics of a device using curve fitting. Although the model is originally based on the physics of device operation, the parameters loose their physical significance due to curve fitting. Therefore, any correlation that existed between the process parameters, such as the doping levels, may disappear, and the model cannot be used to predict the effect of process variations on circuit performance.

Once the dc model has been obtained for the device it is extended to account for the dynamic or transient operation. This requires identifying the depletion regions and the charge-storage regions within the device and modeling them as capacitors attached between the various internal and external terminals of the device. The capacitance models are based on the concept of incremental-charge partitioning. Physical insight into device operation is necessary to determine the charge partitioning. For MOSFET-capacitance models, the application of incremental-charge partitioning provides correct results only for capacitances connected between the gate contact and the other contacts [25].

Even if the analytical model works well under dc conditions, the transient response may be questionable for some regions of device operation. For example, the standard SPICE analytical models for the diode and BJT are inaccurate for operation under high-level-injection conditions. In spite of some of these shortcomings analytical models have been and are extremely useful for circuit design.

B. Table Methods

Unlike analytical models, table models are not based on the physical operation of the device. The basic idea in table models is to store a set of discrete data, for the current-voltage and charge-voltage characteristics, in multi-dimensional arrays or tables. The model-evaluation subroutine of the circuit simulator interpolates between the discrete values that are stored to obtain the current and conductance values for a given set of terminal voltages. Table models ensure accuracy in the current-voltage characteristics, but in general do not address the accuracy in conductances which is of concern for analog circuits. The models are very efficient since no complex function evaluations are required and only simple arithmetic operations have to be performed to compute the conductance and current values. Since table models rely on measured data after devices have been fabricated with a process, they do not posses any predictive capability. The use of table models for modeling the intrinsic capacitance of a MOSFET has been very limited. A charge-partitioning scheme has to be incorporated in the table model similar to that for analytical models.

C. Numerical Models

Numerical models use the solution of the basic physical laws governing device operation for determining the characteristics of a device. The basic physical laws are described by Poisson's equation and the current-continuity equations. Since these equations are partial differential equations (PDEs), numerical methods have to be used to obtain the solution, hence the name numerical device models. These models are accurate and useful for detailed simulations. Since these models are specified by their geometry and doping profile, they can predict the impact of process variations on circuit performance. However, since the evaluation of numerical models requires a long simulation time, it is impractical to simulate a large circuit with a large number of numerical devices. To reduce the simulation time for the numerical models, some simplifications can be introduced such that the equations can be solved efficiently by numerical techniques. These models are referred to as quasi-numerical models since complete numerical solutions are not used. Obviously here the accuracy is sacrificed for efficiency. Numerical models of semiconductor devices will be introduced in detail in the following section.

3.3 Device-Level Simulation

The semiconductor device simulation problem can be formulated as a set of three coupled nonlinear partial differential equations (PDEs) in space and time, which are obtained from the underlying physics. Based on applied terminal voltages the boundary conditions for the PDEs are established and these equations can be solved. The set of continuous PDEs is discretized in both space and time. Space discretization plays an important role in the overall accuracy of a simulation. By space discretization the set of PDEs is transformed into a system of nonlinear differential algebraic equations. For time-domain transient analysis a suitable integration scheme can be applied for time discretization and a system of nonlinear algebraic equations is obtained. These equations are solved using the Newton's method. Small-signal ac analysis is carried out by linearizing the space-discretized device equations at a dc operating point.

3.3.1 Device Simulation Problem

The fundamental equations from the Boltzmann transport equation [26] can be used to describe the operation of a semiconductor device. These are Poisson's equation and the electron- and hole-current continuity equations.

$$\nabla \cdot \varepsilon E = q(N_D - N_A + p - n) \tag{3.12a}$$

$$\frac{1}{q}\nabla \cdot J_n = \frac{\partial n}{\partial t} - (G - R) \tag{3.12b}$$

$$\frac{1}{q}\nabla \cdot J_p = -\frac{\partial p}{\partial t} + (G - R) \tag{3.12c}$$

where

$$E = -\nabla \psi \tag{3.13a}$$

$$J_n = -q\mu_n n \nabla \psi + qD_n \nabla n \tag{3.13b}$$

$$J_p = -q\mu_p p \nabla \psi - qD_p \nabla p \tag{3.13c}$$

and ε is the dielectric constant of the material, q is the electron charge, ψ the electrostatic potential, n (p) electron (hole) concentration, E electric field, J_n (J_p) electron (hole) current density, μ_n (μ_p) electron (hole) mobility, D_n (D_p) electron (hole) diffusivity, N_D (N_A) donor (acceptor) concentration, and G and R

are net generation and recombination rates, respectively. The variables ψ, n, and p are of different orders of magnitudes, hence the device equations have to be scaled in an appropriate manner. The semiconductor device equations after scaling can be expressed as:

$$\nabla \cdot E = (N_D - N_A + p - n) \tag{3.14a}$$

$$\nabla \cdot J_n = \frac{\partial n}{\partial t} - (G - R) \tag{3.14b}$$

$$\nabla \cdot J_p = -\frac{\partial p}{\partial t} + (G - R) \tag{3.14c}$$

and the electric field and current densities are given by:

$$E = -\nabla \psi \tag{3.15a}$$

$$J_n = -\mu_n [n \nabla \psi - \nabla n] \tag{3.15b}$$

$$J_p = -\mu_p [p \nabla \psi + \nabla p] \tag{3.15c}$$

The solution of the above system of equations provides the internal distributions of the electrostatic potential and the carrier densities, and the external terminal currents. These equations cannot be solved analytically and numerical methods have to be used.

3.3.2 Space Discretization

To solve the above basic semiconductor equations numerically these continuous equations are discretized in the space over a simulation domain. The domain is divided into smaller regions, and the discrete problem is solved for each of these regions. For the regions which are part of the boundary, the applied boundary conditions should be considered. After space discretization, a system of nonlinear differential-algebraic equations is obtained.

In CODECS the finite-difference scheme [27] is used for space discretization. The rectangular simulation domain is divided into small nonoverlapping rectangular regions by grid lines parallel to the x and y axes. The discretized equations are assembled at each grid node by approximating the spatial derivatives with difference expressions. For a rectangular mesh with grid spacing $h_i = x_{i+1} - x_i$ and $k_j = y_{j+1} - y_j$, the spatial derivatives of a function $f(x, y)$ at grid node $(i, j)$are approximated by:

$$\left[\frac{\partial f}{\partial x}\right]_{i,j} = \frac{f(x_{i+1/2}, y_j) - f(x_{i-1/2}, y_j)}{\frac{h_i + h_{i-1}}{2}} \tag{3.16a}$$

$$\left[\frac{\partial f}{\partial y}\right]_{i,j} = \frac{f(x_i, y_{j+1/2}) - f(x_i, y_{j-1/2})}{\frac{k_j + k_{j-1}}{2}} \tag{3.16b}$$

where $x_{i-1/2} = x_i - \frac{h_{i-1}}{2}$, $x_{i+1/2} = x_i + \frac{h_i}{2}$, $y_{j-1/2} = y_j - \frac{k_{j-1}}{2}$, and $y_{ij+1/2} = y_j + \frac{k_j}{2}$ as shown in Figure 3.1.

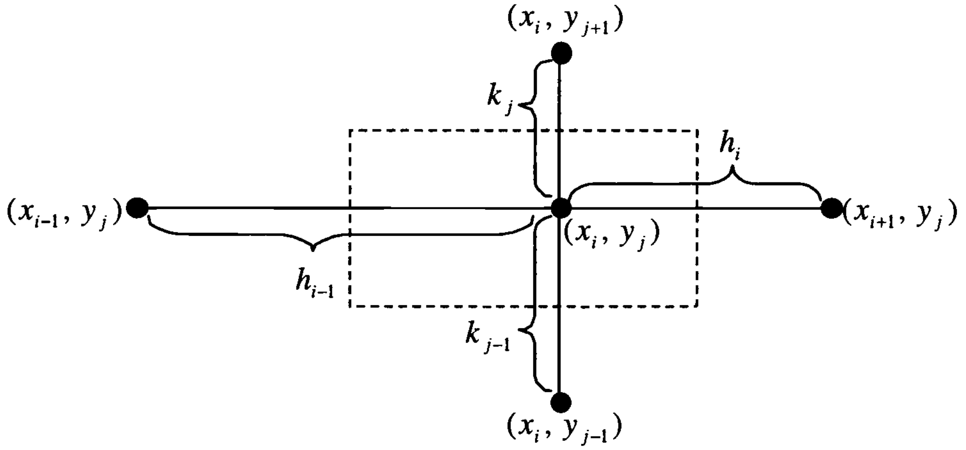


Figure 3.1: Schematic of the grid used for finite-difference space discretization.

Use of this approximation results in the following discretizations for the Possion's and the current-continuity equations:

$$\frac{E_x\big|_{i+1/2,j} - E_x\big|_{i-1/2,j}}{\frac{h_i + h_{i-1}}{2}} + \frac{E_y\big|_{i,j+1/2} - E_y\big|_{i,j-1/2}}{\frac{k_i + k_{i-1}}{2}} = [N + p - n]_{i,j} \tag{3.17a}$$

$$\frac{J_{nx}\big|_{i+1/2,j} - J_{nx}\big|_{i-1/2,j}}{\frac{h_i + h_{i-1}}{2}} + \frac{J_{ny}\big|_{i,j+1/2} - J_{ny}\big|_{i,j-1/2}}{\frac{k_i + k_{i-1}}{2}} = \left[\frac{\partial n}{\partial t}\right]_{i,j} - [G - R]_{i,j} \tag{3.17b}$$

$$\frac{J_{px}\big|_{i+1/2,j} - J_{px}\big|_{i-1/2,j}}{\frac{h_i + h_{i-1}}{2}} + \frac{J_{py}\big|_{i,j+1/2} - J_{py}\big|_{i,j-1/2}}{\frac{k_i + k_{i-1}}{2}} = -\left[\frac{\partial p}{\partial t}\right]_{i,j} + [G - R]_{i,j} \tag{3.17c}$$

In the above equations the values of $E_{x,j}$, $J_{nx,y}$ and $J_{px,y}$ are required at the midpoints of each interval $[x_i, x_{i+1}]$ or $\lfloor y_j, y_{j+1} \rfloor$. These values can be approximated from the nodal values of the electrostatic potential and the carrier concentrations and are given as:

$$E_x\Big|_{i+1/2,j} = -\frac{\psi_{i+1,j} - \psi_{i,j}}{h_i} \tag{3.18a}$$

$$J_{nx}\Big|_{i+1/2,j} = \frac{\mu_n\big|_{i+1/2,j}}{h_i}\left[n_{i+1,j}B(\psi_{i+1,j} - \psi_{i,j}) - n_{i,j}B(-(\psi_{i+1,j} - \psi_{i,j}))\right] \tag{3.18b}$$

$$J_{px}\Big|_{i+1/2,j} = \frac{\mu_p\big|_{i+1/2,j}}{h_i}\left[p_{i,j}B(\psi_{i+1,j} - \psi_{i,j}) - p_{i+1,j}B(-(\psi_{i+1,j} - \psi_{i,j}))\right] \tag{3.18c}$$

where $B(x) = \frac{x}{e^x - 1}$ is the Bernoulli's function. So far the device equations have been discretized in space using a suitable grid and the unknowns are the electrostatic potential and the electron and hole concentrations at each grid node.

3.3.3 Solution Methods for Device Equations

After space discretization, the equations at all grid nodes can be expressed in a symbolic form as:

Poisson's equation: $$F_\psi(\psi, n, p) = 0 \tag{3.19a}$$

n continuity equation: $$F_n(\psi, n, p) - \frac{\partial n}{\partial t} = 0 \tag{3.19b}$$

p continuity equation: $$F_p(\psi, n, p) + \frac{\partial p}{\partial t} = 0 \tag{3.19c}$$

The complete system of equations can be represented in a general form as:

$$F(\dot{w}(t), w(t)) = 0 \tag{3.20}$$

where w is the vector of electrostatic potential, ψ, electron concentration, n, and hole concentration, p, at each grid node.

A. DC and Transient Analysis

For DC analysis, the system of nonlinear differential-algebraic equations (3.20) reduces to a system of nonlinear algebraic equations by setting $\dot{w} = 0$ and is given by:

$$F(0, w^{*}) = 0 \tag{3.21}$$

These equations can be solved by a direct method using a Newton scheme. A large amount of storage and computational effort is required, since a large system of equations has to be solved. Some relaxation-based approaches can be used to solve the problem efficiently. However, under high-level injection conditions the relaxation methods converge extremely slowly and may require a very large number of iterations to reach convergence [2].

For transient analysis, the time-derivative terms $\frac{\partial n}{\partial t}$ and $\frac{\partial p}{\partial t}$ in Equation (3.20) have to be discretized in time. As in the circuit-simulation problem described in Section 3.2.1.A this can be done by use of an integration formula. The time discretization results in a system of nonlinear algebraic equations and again this can be solved by Newton's method or relaxation-based approaches.

B. Small-Signal AC Analysis

Small-signal ac analysis involves finding the ac response at an established dc operating point (ψ_0, n_0, p_0, V_0) where V_0 is the applied bias. Following the same derivation as in the circuit simulation problem described in Section 3.2.1.B, the small-signal ac solution is given by:

$$\begin{bmatrix} \frac{\partial F_\psi}{\partial \psi} & \frac{\partial F_\psi}{\partial n} & \frac{\partial F_\psi}{\partial p} \\ \frac{\partial F_n}{\partial \psi} & \frac{\partial F_n}{\partial n} - j\omega I & \frac{\partial F_n}{\partial p} \\ \frac{\partial F_p}{\partial \psi} & \frac{\partial F_p}{\partial n} & \frac{\partial F_p}{\partial p} + j\omega I \end{bmatrix} \begin{bmatrix} \tilde{\psi} \\ \tilde{n} \\ \tilde{p} \end{bmatrix} = - \begin{bmatrix} \frac{\partial F_\psi}{\partial V} \\ \frac{\partial F_n}{\partial V} \\ \frac{\partial F_p}{\partial V} \end{bmatrix} \tilde{V} \tag{3.22}$$

where all the derivatives are evaluated at the dc operating point.

3.4 Coupled Device and Circuit Simulation (CODECS)

In this section, first the circuit-level dc problem is used to illustrate the relationship between the analytical model evaluation and circuit simulation. If the same relationship is used between the numerical device models and circuit simulation, then the result is a two-level Newton algorithm. The algorithm is used to describe the architecture for coupling the device simulator to the circuit simulator. This architecture allows complete decoupling between the circuit and device simulators. Since the algorithm is motivated by analytical models, the numerical models are viewed as another model type from the circuit simulation point of view. The interface between the two simulators is well defined and the interface routines necessary for this coupling are described. The full-Newton algorithm, the general formulation for coupling the device and circuit simulators, is then described. The full-Newton algorithm can also be implemented in this architecture in a decoupled manner.

3.4.1 Two-Level Newton Algorithm

Consider a dc problem for a circuit with semiconductor devices modeled by nonlinear analytical models. For these analytical models, the terminal characteristics are described by a closed-form expression, $i = I(V)$. Then the nonlinear algebraic equations describing the dc problem are given by:

$$f(I(V), V, u) = 0 \tag{3.23}$$

where V is the unknown vector and u is the dc excitation. When Newton's method is used to solve the nonlinear circuit equations, a linear system of equations is solved at each iteration until convergence is achieved. As in Section 3.2.1.A, the linear iteration equation is:

$$\left[\frac{\partial f}{\partial I} \frac{\partial I}{\partial V}\bigg|_{V^k} + \frac{\partial f}{\partial V}\bigg|_{V^k} \right] \Delta V^{k+1} = -f(I(V^k), V^k, u) \tag{3.24}$$

where k is the iteration number. The equivalent conductance, G_{eq}, is defined as:

$$G_{eq}^k = \frac{\partial I}{\partial V}\bigg|_{V^k} \tag{3.25}$$

The task of analytical model evaluation is to calculate G_{eq}^k and $I(V^k)$ for the terminal voltage V^k. These are then used in Equation (3.24) to carry out the circuit simulation.

To couple the numerical devices with a circuit simulator in a similar manner, the equivalent conductances and terminal currents have to be calculated for numerical devices. But unlike analytical models, the current-voltage characteristics are not known as closed-form expressions for the numerical devices. So G_{eq} and $I(V)$ cannot be calculated by function evaluations and numerical techniques as described in Section 3.3 must be used. The partial-differential equations describing a numerical device have to be solved for each operating point specified by the terminal voltages. As shown in Section 3.3, these device-level nonlinear equations are also solved by a Newton's method. Once the equations have been solved for an applied terminal voltage V, the equivalent conductances and terminal currents can be calculated as described in Section 3.4.2 and the circuit-level iteration from Newton's method as in Equation (3.24) can be performed. The overall solution technique is a two-level Newton scheme. The inner Newton's method is used for the device-level simulation and the outer one is for the circuit-level simulation.

3.4.2 Calculation of Conductances

After space and time discretization the device-level equations can be represented as a set of nonlinear algebraic equations:

$$F(w, V) = 0 \qquad (3.26)$$

where w is the vector of internal variables. The dependence of the boundary condition V is explicitly written in the above equation. Let $i = I(w, V)$ represent the terminal currents as a function of w and V. i is calculated by summing the current density components around a contact. It should be noted that w can be expressed as a function of V from Equation (3.26). The system of Equation (3.26) is solved for an applied voltage V_0 by Newton's method where

$$\Delta w = -J_w^{-1} F(w, V_0) \tag{3.27}$$

is solved at each iteration. $J_w = \frac{\partial F}{\partial w}$ is the Jacobian matrix of the device-level equations. When Newton's method converges, w is the solution of $F(w, V_0) = 0$ and $I(w, V_0)$ can be calculated.

To calculate the linearized conductance $G_{eq} = \frac{\partial i}{\partial V}$, the chain rule has to be used:

$$G_{eq} = \frac{\partial i}{\partial V} = \frac{\partial I}{\partial w}\frac{\partial w}{\partial V} + \frac{\partial I}{\partial V} \tag{3.28}$$

where $\frac{\partial I}{\partial w}$ and $\frac{\partial I}{\partial V}$ are obtained by symbolic differentiation of the function $I(w, V)$. $\frac{\partial w}{\partial V}$ is calculated in the following manner. The partial derivative of Equation (3.25) with respect to V is:

$$J_w \frac{\partial w}{\partial V} + J_V = 0 \tag{3.29}$$

with $J_V = \frac{\partial F}{\partial V}$. From Equation (3.29) we can solve for $\frac{\partial w}{\partial V}$ as:

$$\frac{\partial w}{\partial V} = -J_w^{-1} J_V \tag{3.30}$$

Since J_w is available in its LU factors that were calculated during the solution of Equation (3.26) by use of Equation (3.27), only forward and backward substitutions are required in calculating $\frac{\partial w}{\partial V}$ which is computationally inexpensive. Then G_{eq} can be calculated from Equation (3.28) and is given by:

$$G_{eq} = -\frac{\partial I}{\partial w} J_w^{-1} J_V + \frac{\partial I}{\partial V} \tag{3.31}$$

3.4.3 Architecture of CODECS

It is clear from the previous sections that a numerical device model is similar to an analytical device model in several respects for circuit simulation. Given the terminal voltage, the equivalent conductances and terminal currents have to be calculated and used in the circuit-level equations. For an analytical model this task involves function evaluations, whereas for a numerical device the three PDEs have to be solved. The interface to a circuit simulator can be identical for the two types of models as shown in Figure 3.2, where the task of model evaluation is illustrated. The interface to the circuit simulator is through routines for model evaluation, and for loading the equivalent currents and conductances in the circuit-level Jacobian and right-hand side vector.

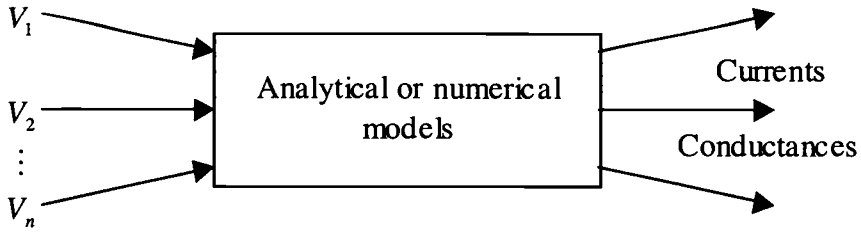


Figure 3.2: The task of model evaluation.

The overall framework of CODECS is shown in Figure 3.3. The circuit simulator is the controlling program. It supports analytical models for the circuit elements and also stores the vector of node voltages. These voltages are available to the model-evaluation subroutines that calculate the equivalent conductances and terminal currents for a device. The numerical devices are simulated by the device simulator of CODECS, and the interface to the circuit simulator is identical to that for analytical models. Device-level simulation is used to solve the PDEs for a numerical device for given terminal voltages. Then the terminal conductances and currents are calculated at the operating point, and assembled in the circuit-level Jacobian matrix and right-hand side vector.

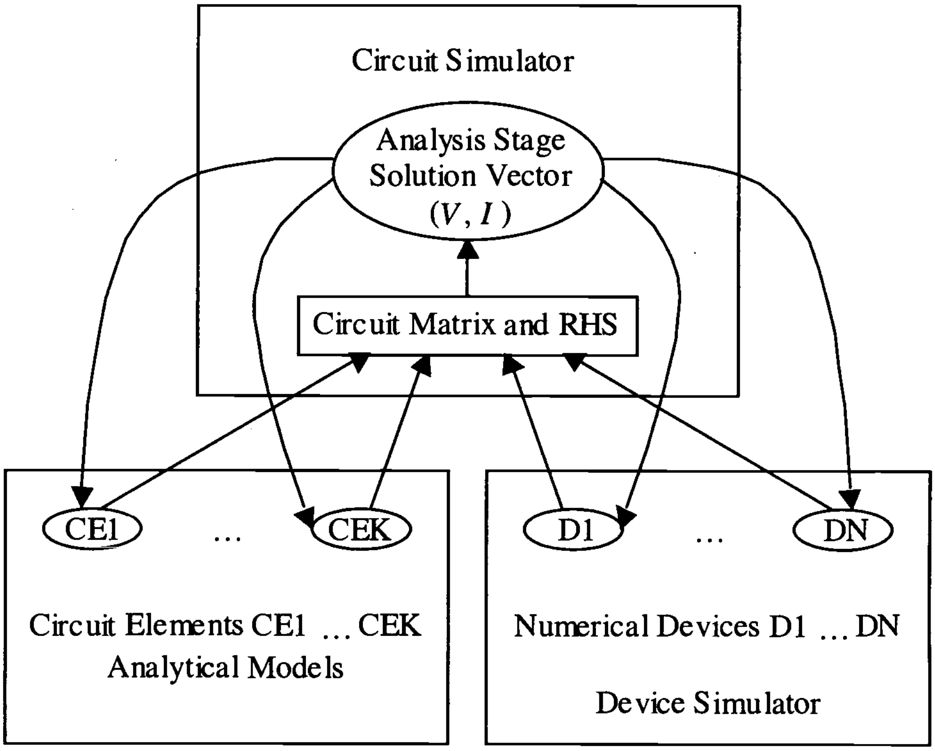


Figure 3.3: Architecture of CODECS. Numerical devices are interfaced with the circuit simulator in a manner similar to that for analytical devices. The circuit node voltages establish the boundary conditions for the numerical devices. The PDEs are solved by the device-level simulator of CODECS.

3.4.4 Full-Newton Algorithm

An alternate way to view coupled device and circuit simulation is to treat the numerical devices and circuit-level elements as one problem. Then the device-level and circuit-level equations are combined and expressed as one system of equations. The complete system of nonlinear equations can be solved by Newton's method. Unlike the two-level Newton algorithm where the device and circuit-level unknowns are solved separately in a decoupled manner, the complete set of unknowns is solved simultaneously. Combining the device-level Equation (3.26) with the circuit-level Equation (3.23), the complete system of equations is given by:

$$F(w,V)=0 \tag{3.32}$$

$$f(I(w,V),V,u)=0 \tag{3.33}$$

Using Newton's method Equations (3.32) and (3.33) are solved and the iteration equations are:

$$J_w\Delta w+J_V\Delta V=-F(w,V) \tag{3.34}$$

$$\frac{\partial f}{\partial I}\left(\frac{\partial I}{\partial w}\Delta w+\frac{\partial I}{\partial V}\Delta V\right)+\frac{\partial f}{\partial V}\Delta V=-f(I(w,V),V,u) \tag{3.35}$$

From Equation (3.34), Δw can be solved as:

$$\Delta w=J_w^{-1}(-F(w,V)-J_V\Delta V) \tag{3.36}$$

Substituting Δw from Equation (3.36) into Equation (3.35), we obtain:

$$\left[\frac{\partial f}{\partial I}\left(-\frac{\partial I}{\partial w}J_w^{-1}J_V+\frac{\partial I}{\partial V}\right)+\frac{\partial f}{\partial V}\right]\Delta V=-f(I(w,V),V,u)+\frac{\partial f}{\partial I}\left[\frac{\partial I}{\partial w}J_w^{-1}F(w,V)\right]$$

(3.37)

This equation can be rewritten as:

$$\left[\frac{\partial f}{\partial I}G_{eq}+\frac{\partial f}{\partial V}\right]\Delta V=-f(I(w,V),V,u)+\frac{\partial f}{\partial I}\left[\frac{\partial I}{\partial w}J_w^{-1}F(w,V)\right] \tag{3.38}$$

with $G_{eq}=-\frac{\partial I}{\partial w}J_w^{-1}J_V+\frac{\partial I}{\partial V}$. Although this G_{eq} has the same form as Equation (3.31) for the two-level Newton algorithm, they are different. This G_{eq} is evaluated based on the device internal state at the current iteration, while G_{eq} in Equation

(3.31) is evaluated based on the converged device internal state for given terminal voltages. The same situation applies to $I(w,V)$ in these two cases. Equation (3.38) is similar in form to that obtained with the analytical models or by use of the two-level Newton algorithm. Now the numerical model evaluation consists of calculation of G_{eq}, $I(w,V)$, and $\frac{\partial I}{\partial w} J_w^{-1} F(w,V)$ at each iteration. These are then used in the circuit-level Equation (3.38). Thus the full Newton algorithm can also be used to embed numerical models within a circuit-simulation program and implemented in the framework of CODECS shown in Figure 3.3.

Chapter 4

TIME-DOMAIN SHOOTING METHOD

4.1 Introduction

The time-domain shooting method [5, 17-20] as introduced in Section 2.3 is an efficient approach to compute the steady-state solution of RF circuits. Due to extremely large time constants introduced by large inductors, large capacitors, and narrowband high-Q filters, RF circuits approach the steady state very slowly with the DC solution as an initial condition. For this reason, conventional transient simulation can become impractical especially when computationally expensive models (numerical models) are included as in CODECS. In this chapter we describe in detail the shooting algorithm and its practical implementation.

This chapter is organized as follows. The formulation of the periodic steady-state problem with the shooting method is presented in Section 4.2. An efficient approach to obtain the sensitivity matrix required by the shooting method is provided in Section 4.3. In Section 4.4, the formulation of the shooting method is modified to obtain the steady state of autonomous systems where the period is also an unknown. In Section 4.5, the implementation details and convergence heuristics are described. Some implementation issues specific to CODECS are discussed in Section 4.6 and this chapter is summarized in Section 4.7.

4.2 Solution of the Periodic Steady-State Problem with the Shooting Method

4.2.1 The Shooting Method

Consider the system of equations

$$f(\dot{x}, x, t) = 0 \tag{4.1}$$

where $x(t)$ is a vector of state variables and f is a nonlinear algebraic function. Define ϕ to be the function that maps the state x_0 at t_0 to the solution of Equation (4.1). That is:

$$x(t) = \phi(x_0, t_0, t) \tag{4.2}$$

This function is referred to as the state-transition function. In circuit-level simulation, the state vector, x, includes the voltages across the capacitors and currents through the inductors. The state-transition function ϕ can be obtained by a regular transient analysis starting from the initial state x_0 at t_0.

From the definition of a periodic function, the state variable $x(t)$ is periodic with T if $x(t) = x(t+T)$ for all t. This is a difficult condition to apply in practice because the condition must be verified over all t. However, if $x(t)$ is the solution of a differential equation that is smooth, then by uniqueness,

$$x(t) = x(t+T) \tag{4.3}$$

for some t implies that the condition is true for all t [5]. Fortunately, the system of differential equations used to describe practical circuits with analytical and/or numerical devices has a unique transient solution starting from an arbitrary initial condition. With t in Equation (4.3) set to zero and using Equation (4.2), the periodic boundary constraint (4.3) becomes:

$$x(0) - x(T) = x(0) - \phi(x(0), 0, T) = 0 \qquad (4.4)$$

The condition of Equation (4.4) is illustrated in Figure 4.1 where $x(T) = \phi(x(0), 0, T)$ is the final state corresponding to the initial state $x(0)$.

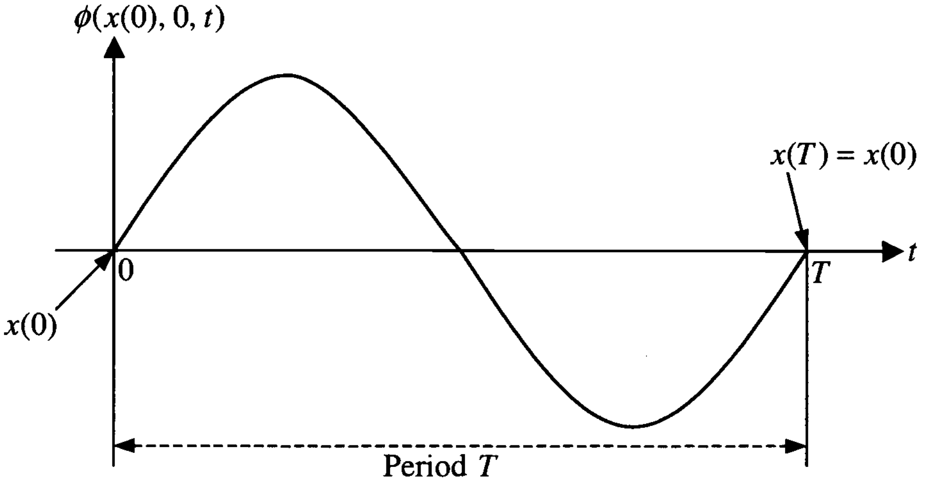


Figure 4.1: Periodic boundary constraint.

Therefore, finding the periodic solution of Equation (4.1) is equivalent to solving for the root of the implicit nonlinear Equation (4.4). This is the starting point for the shooting method. Once $x(0)$ in Equation (4.4) is found, a numerical solution of Equation (4.1) with $x(0)$ as the initial condition leads directly into the periodic steady-state solution.

4.2.2 Shooting with Newton's Method

If Newton's method is used to solve the nonlinear Equation (4.4), the following iterative equation is obtained:

$$x^{(j+1)}(0) = x^{(j)}(0) - \left[I - J^{(j)}(T)\right]^{-1} \left[x^{(j)}(0) - x^{(j)}(T)\right] \qquad (4.5)$$

where I is the identity matrix, $x^{(j)}(T) = \phi(x^{(j)}(0), 0, T)$ is computed by a conventional transient analysis starting from the initial state $x^{(j)}(0)$ over the time interval T, and

$$J^{(j)}(T) = \left.\frac{\partial x(T)}{\partial x(0)}\right|_{x(0)=x^{(j)}(0)} \qquad (4.6)$$

is the sensitivity matrix of the final state $x^{(j)}(T)$ with respect to changes in the initial state $x^{(j)}(0)$. This sensitivity matrix is used to determine the correction in the initial state once the difference between the initial state and the final state is found. In the next section, an efficient computation of the sensitivity matrix by means of sensitivity circuits is described.

From the point of view of a Newton iteration, the shooting method solves the periodic boundary constraint (4.4) by computing the solution for a succession of transient analyses over the period T. Each of these transient analyses starts out with an improved guess for the initial state eventually resulting in the steady-state solution. First the circuit is simulated for one period using some guess for the initial state. Then the final state $x^i(T)$ is checked with the guess for the initial state $x^i(0)$. If they don't match each other, the initial state is adjusted according to Equation

(4.5). The circuit is simulated again for one period with the adjusted initial state. This process is repeated until the initial state and the final state match within some specified tolerances. The solution procedure for the shooting method is illustrated in Figure 4.2 where $x^0(0)$ is the initial guess for the initial condition and $x^*(0)$ is the initial condition leading to the steady-state solution.

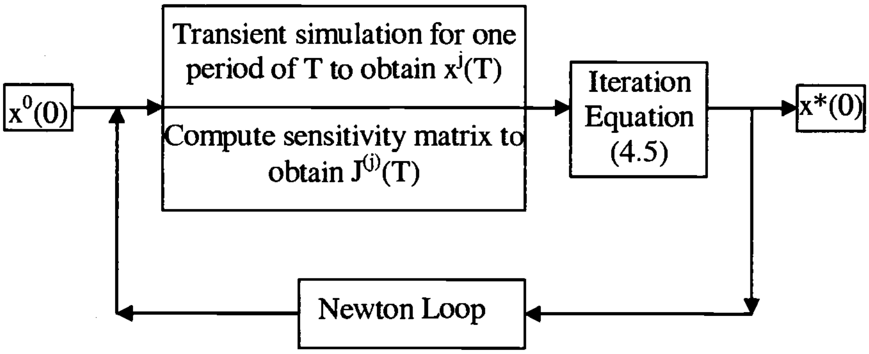

$x^0(0)$
Transient simulation for one period of T to obtain $x^j(T)$
Compute sensitivity matrix to obtain $J^{(j)}(T)$
Iteration Equation (4.5)
$x^*(0)$
Newton Loop

Figure 4.2: Solution procedure for the shooting method.

An alternative method to solve the nonlinear Equation (4.4) is the extrapolation method [28]. However, the extrapolation method has been shown to be not as robust as the Newton's method in [29]. Recently, the homotopy method [30, 31] was successfully implemented to solve Equation (4.4) [32-35]. The homotopy method is a robust and accurate numerical technique for solving nonlinear algebraic equations and can be used as a backup when Newton's method fails. In this work Newton's method is used to solve the nonlinear equation in the shooting method formulation. Some modifications and heuristics will be described in Section 4.5 which considerably improve the reliability and convergence of Newton's method.

4.3 Sensitivity Matrix Computation by Sensitivity Circuits

In this section, an efficient computation of the sensitivity matrix in Equation (4.6) by means of sensitivity circuits is described.

4.3.1 Sensitivity Circuits

Sensitivity circuits [36] are an efficient approach to compute the sensitivity matrix $J^{(j)}(T)$ in Equation (4.6). To develop the sensitivity circuits, a general sparse-tableau formation of the circuit equations is used such that the result will be valid for any other specific formulation. The sparse-tableau formulation [37] is:

$$Ai = 0 \quad \text{(KCL)} \tag{4.7}$$

$$A^T e = v \quad \text{(KVL)} \tag{4.8}$$

$$i_r = f_r(v_r) \quad \text{(resistive branch)} \tag{4.9}$$

$$q_c = f_c(v_c) \quad \text{(capacitive branch)} \tag{4.10}$$

$$\lambda_l = f_l(i_l) \quad \text{(inductive branch)} \tag{4.11}$$

where

$$\frac{dq_c}{dt} = i_c; \quad v_c(0) = v_{c0}$$

$$\frac{d\lambda_l}{dt} = v_l; \quad i_l(0) = i_{l0}$$

and i and v are the vectors of branch currents and voltages respectively, and e is the vector of node voltages. i_r and v_r are the vectors of currents and voltages of resistive branches, i_c and v_c are the vectors of currents and voltages of capacitive branches, and i_l and v_l are the of vectors of currents and voltages of inductive branches. q_c is the vector of charges for capacitive branches and λ_c is the vector of fluxes for inductive branches. The branch constraints due to the independent sources i_s and v_s are:

$$Ci_s + Dv_s = F(t) \tag{4.12}$$

The branch current and voltage vectors have been partitioned so that:

$$i = \begin{bmatrix} i_r \\ i_c \\ i_l \\ i_s \end{bmatrix} \text{ and } v = \begin{bmatrix} v_r \\ v_c \\ v_l \\ v_s \end{bmatrix}$$

and the initial state vector is partitioned as:

$$x(0) = \begin{bmatrix} v_{c0} \\ i_{l0} \end{bmatrix} \tag{4.13}$$

The sensitivity of the circuit with respect to the initial state of one of the state variables $x_n(0)$ is given by the first-order partial derivatives of Equations (4.7)-(4.13) to the initial state and described by the following set of equations:

$$A\frac{\partial i}{\partial x_n(0)} = 0 \qquad \text{(KCL)} \tag{4.14}$$

$$A^T \frac{\partial e}{\partial x_n(0)} = \frac{\partial v}{\partial x_n(0)} \qquad \text{(KVL)} \tag{4.15}$$

$$\frac{\partial i_r}{\partial x_n(0)} = \left.\frac{\partial f_r}{\partial v_r}\right|_{v_r(t)} \frac{\partial v_r}{\partial x_n(0)} \qquad \text{(resistive branch)} \qquad (4.16)$$

$$\frac{\partial q_c}{\partial x_n(0)} = \left.\frac{\partial f_c}{\partial v_c}\right|_{v_c(t)} \frac{\partial v_c}{\partial x_n(0)} \qquad \text{(capacitive branch)} \qquad (4.17)$$

$$\frac{\partial \lambda_l}{\partial x_n(0)} = \left.\frac{\partial f_l}{\partial i_l}\right|_{i_l(t)} \frac{\partial i_l}{\partial x_n(0)} \qquad \text{(inductive branch)} \qquad (4.18)$$

where

$$\frac{d}{dt}\left[\frac{\partial q_c}{\partial x_n(0)}\right] = \frac{\partial i_c}{\partial x_n(0)}$$

and

$$\frac{\partial v_c(0)}{\partial x_n(0)} = \begin{cases} e_j, & \text{if } x_n \text{ is the branch voltage of the } j\text{th capacitor} \\ 0, & \text{if } x_n \text{ is an inductor current} \end{cases}$$

where e_j is the unit vector.

$$\frac{d}{dt}\left[\frac{\partial \lambda_l}{\partial x_n(0)}\right] = \frac{\partial v_l}{\partial x_n(0)}$$

and

$$\frac{\partial i_l(0)}{\partial x_n(0)} = \begin{cases} e_j, & \text{if } x_n \text{ is the current in the } j\text{th inductor} \\ 0, & \text{if } x_n \text{ is a capacitor branch voltage} \end{cases}$$

Finally,

$$C\frac{\partial i_s}{\partial x_n(0)} + D\frac{\partial v_s}{\partial x_n(0)} = 0 \qquad (4.19)$$

Equation (4.19) means that the independent current and voltage sources become open and short circuits, respectively, in the sensitivity circuits.

Equations (4.14)-(4.19) describe a linear version of the original circuit with the same topology. This linear circuit is called the sensitivity circuit. The major difference between the two circuits is in the initial conditions for the differential equations and the independent sources. In the sensitivity circuit, all the independent sources are zero as shown in Equation (4.19), the initial condition of the capacitor (inductor) with respect to which the sensitivity is calculated is 1, and all other initial conditions are 0. There is a sensitivity circuit for each capacitor and each inductor in the circuit. The solutions of each of these sensitivity circuits, with the initial condition mentioned above, at $t = T$ is the set of partial derivatives required to form the sensitivity matrix, $J(T)$, used in the Newton iterative Equation (4.5).

4.3.2 Sensitivity Computation along with Transient Simulation

In the numerical calculation of the transient response employing implicit integration methods the nonlinear elements are replaced by linear approximations. In calculating the circuit response at time t, when the Newton iteration converges, the linear circuit matrix at time t is available in a LU-factored form. Since the sensitivity circuit is a linearized version of the original circuit at the current operating point, the sensitivity circuit will have the same linear circuit matrix at each time point during the transient simulation. If the sensitivity circuit is solved together with the original circuit at each time point, the already LU-factored linear circuit matrix can be used and only inexpensive forward and backward substitutions have to be carried out to solve the sensitivity circuit.

To illustrate this further, let us consider the system of nonlinear differential algebraic equations describing a circuit as:

$$f(\dot{x}(t), x(t), u(t)) = 0 \tag{4.20}$$

The linear circuit matrix with respect to $x(t)$ at time t is obtained by:

$$\frac{\partial f(\dot{x}, x, u)}{\partial x} = \frac{\partial f}{\partial \dot{x}} \cdot \frac{\partial \dot{x}}{\partial x} + \frac{\partial f}{\partial x} \tag{4.21}$$

For a linear multistep integration formula used during a transient simulation:

$$\dot{x} = \alpha x + \beta_x \tag{4.22}$$

where α is a constant and β_x is a function of x at previous timepoints. The linear circuit matrix in Equation (4.21) is:

$$\frac{\partial f}{\partial \dot{x}} \cdot \frac{\partial(\alpha x + \beta_x)}{\partial x} + \frac{\partial f}{\partial x} = \frac{\partial f}{\partial \dot{x}} \cdot \alpha + \frac{\partial f}{\partial x} \tag{4.23}$$

The sensitivity of Equation (4.20) with respect to the initial condition $x(0)$ is given as:

$$\frac{\partial f(\dot{x}, x, u)}{\partial x(0)} = \frac{\partial f}{\partial \dot{x}} \cdot \frac{\partial \dot{x}}{\partial x(0)} + \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial x(0)} = \frac{\partial f}{\partial \dot{x}} \cdot \dot{z} + \frac{\partial f}{\partial x} \cdot z = 0 \tag{4.24}$$

where $z = \frac{\partial x}{\partial x(0)}$ is the sensitivity matrix that has to be computed. If the same integration formula as Equation (4.22) is used here, Equation (4.24) can be transformed as:

$$\frac{\partial f}{\partial \dot{x}} \cdot (\alpha z + \beta_z) + \frac{\partial f}{\partial x} \cdot z = 0 \quad \Rightarrow \quad (\frac{\partial f}{\partial \dot{x}} \cdot \alpha + \frac{\partial f}{\partial x}) z = -\frac{\partial f}{\partial \dot{x}} \beta_z \tag{4.25}$$

where β_z is a function of z at previous time points. It can be seen that the system matrix for the sensitivity Equation (4.25) is the same as Equation (4.23) obtained from the transient simulation.

4.4 Shooting Method for Autonomous Systems

A system is autonomous if both the system and its inputs do not vary with time. For a circuit, it means all the branch relationships are time invariant and the independent sources are constant valued. Oscillators are autonomous circuits that have non-constant periodic solutions [5]. Compared with forced circuits, oscillators have two problems associated with the steady-state analysis. The first is that the period of the steady state of an oscillator is not prefixed and should be solved in the steady-state analysis. The second is that since there is no input to fix the phase, if one steady state exists, any time shifted version is also a steady-state solution.

To handle autonomous systems, shooting methods for the forced system should be modified to handle the above two problems. Since the phase of the steady state solution is not of importance, some value can be assigned to one of the state variables so that the phase is fixed and the shooting method targets one solution. At the same time, the period of oscillation T is added to the list of unknowns. Thus, a solvable system with N unknowns and the same number of equations is obtained.

With these modifications, the modified Newton equation for the shooting method is given by [18]:

$$y^{(j+1)} = y^{(j)} - \left[I' - J_T^{(j)}(T^{(j)})\right]^{-1} \left[x^{(j)}(0) - x^{(j)}(T^{(j)})\right] \tag{4.26}$$

where y is given by $\left[x_1(0) \quad x_2(0) \quad \cdots \quad x_{k-1}(0) \quad T \quad x_{k+1}(0) \quad \cdots \quad x_N(0)\right]^T$, i.e., the initial value of the state variable x_k is replaced by the unknown period T. I' is the identity matrix with the kth diagonal element set to zero instead of one. The kth column of I' is zero because the initial state chosen obviously does not depend on the period of the current iteration. The sensitivity matrix $J_T^{(j)}(T^{(j)})$ is given by:

$$J_T^{(j)}(T^{(j)}) = \left.\frac{\partial x(T)}{\partial y}\right|_{x(0)=x^{(j)}(0),\, T=T^{(j)}} \tag{4.27}$$

Compared with the sensitivity matrix in Equation (4.6), in $J_T^{(j)}(T^{(j)})$ the kth column is replaced by $\left.\frac{\partial x(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\, T=T^{(j)}}$ which is the sensitivity vector of the final state $x^{(j)}(T^{(j)})$ with respect to changes in the period $T^{(j)}$. This sensitivity vector can not be obtained from the sensitivity circuits. The following shows an easy way to calculate this sensitivity vector. For a capacitor and inductor:

$$v_c(T) - v_c(0) = \frac{1}{C}\int_0^T i_c(t)dt \quad \Rightarrow \quad \frac{\partial v_c(T)}{\partial T} = \frac{1}{C} i_c(T) \tag{4.28}$$

$$i_l(T) - i_l(0) = \frac{1}{L}\int_0^T v_l(t)dt \quad \Rightarrow \quad \frac{\partial i_l(T)}{\partial T} = \frac{1}{L} v_l(T) \tag{4.29}$$

Where v_c and v_l are the voltages of the capacitor and inductor, and i_c and i_l are the currents of the capacitor and inductor, respectively. The capacitor currents i_c and inductor voltages v_l are calculated during transient simulation and the sensitivity of the final state to the period can be obtained readily by Equations (4.28) and (4.29).

The remaining question that needs to be answered is which state variable in $x(0)$ should be chosen and replaced by the unknown period T. In other words, how should k be selected? First, k should be selected such that $x_k^{(j+1)}(0)$ lies in the range of the orbit of the oscillator. Otherwise, the algorithm can not converge to that orbit and the steady-state oscillatory response cannot be determined. Secondly, $\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}}$ should not be zero. Because this element of the vector $\left.\frac{\partial x(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}}$ will become a diagonal element in the new sensitivity matrix $J_T^{(j)}(T^{(j)})$, and a zero value will cause the matrix to be nonsingular. We choose k by selecting the maximum element of $\left.\frac{\partial x(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}}$ [29], i.e.,

$$\left|\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}}\right| \geq \left|\left.\frac{\partial x_n(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}}\right|, \quad n=1,\cdots,N \qquad (4.30)$$

such that $\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}} \neq 0$. This selection rule also ensures that $x_k^{(j+1)}(0)$ is in the range of the oscillation orbit. $\left.\frac{\partial x_k(t)}{\partial t}\right|_{x(0)=x^{(j)}(0)} = 0$ means that $x_k^{(j)}(t)$ is at its minimum or maximum values or local extreme points. Therefore, if $\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\,T=T^{(j)}} \neq 0$, $x_k^{(j)}(T^{(j)})$ is between those extreme values and thus in the range of the oscillation orbit. This is illustrated in Figure 4.3 where $x_k(T)$ at point C lies between the extreme values at points A and B. The new initial value of the kth state variable, $x_k^{(j+1)}(0)$, is not given by Equation (4.26) since it has been replaced by the unknown period T. Usually $x_k^{(j+1)}(0)$ is set to the final value of the current iteration, $x_k^{(j)}(T^{(j)})$. Since $x_k^{(j)}(T^{(j)})$ is in the range of the oscillation orbit, so is $x_k^{(j+1)}(0)$.

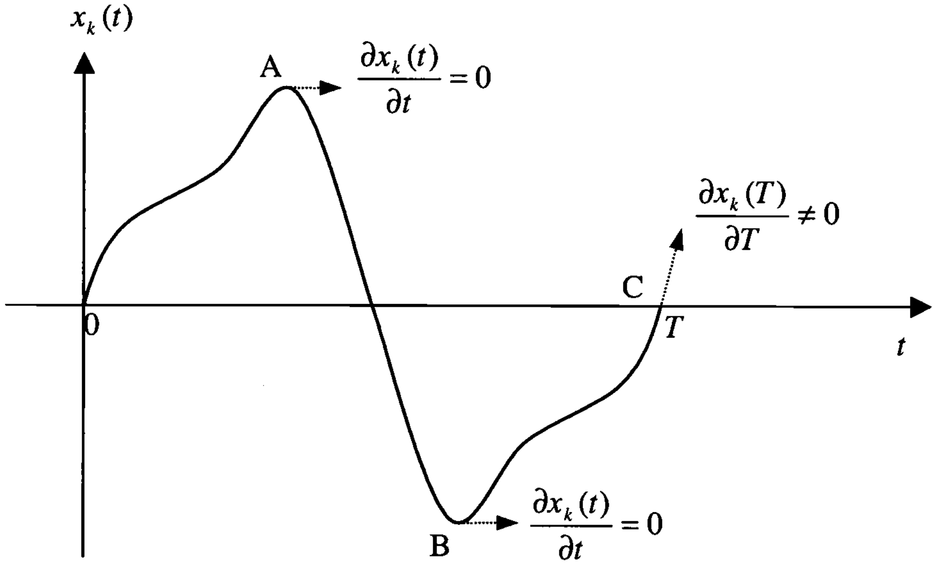

$x_k(t)$
A
$\frac{\partial x_k(t)}{\partial t} = 0$
$\frac{\partial x_k(T)}{\partial T} \neq 0$
C
0
T
t
B
$\frac{\partial x_k(t)}{\partial t} = 0$

Figure 4.3: Waveform of a state variable $x_k(t)$ in one period. If $\frac{\partial x_k(T)}{\partial T} \neq 0$, then $x_k(T)$ lies between the extremes and thus is in the range of its orbit.

4.5 Implementation Consideration and Heuristics

In this chapter, the time-domain shooting method has been presented for both nonautonomous and autonomous systems. The basic idea is the use of Newton's method to solve the periodic boundary constraint of Equation (4.4) to obtain an initial state from which the transient analysis leads directly to the periodic steady-state solution. As with all Newton-based algorithms, good convergence is not achieved for a wide variety of practical nonlinear problems unless some modifications and heuristics are used. In this section, the implementation details and convergence heuristics to the Newton algorithm are described which considerably improve the reliability and convergence of the algorithm.

4.5.1 State Elimination

As discussed in Section 2.3, the long time constants in RF circuits are usually introduced by the large linear capacitors and inductors that make the circuits approach the steady state very slowly. The states of these linear capacitors and inductors are the more dominant ones in determining the steady state. Compared with these dominant states, the states associated with the parasitic capacitances of semiconductor devices have less of an effect on the steady state. Even though their effect is small, it is found that they are very sensitive to the more dominant states and hinder convergence. One way to solve this problem is to not update these states based on the Newton shooting method. Thus, $x_n^{(j+1)}(0) = x_n^{(j)}(T)$ for these states and they are eliminated from the sensitivity matrix. Also the size of the sensitivity matrix is reduced in this way and the simulation is more efficient. Therefore, only the states of linear capacitors and inductors in the circuit are considered in this implementation of the shooting method.

4.5.2 Damped Newton Algorithm

The Newton iteration may overshoot if the current guess is far from the solution. To prevent the overshoot, a damped Newton algorithm should be used since damping reduces the effect of the sensitivity matrix on the iteration. The Newton iteration Equation (4.5) with a damping parameter α is [19, 20]:

$$x^{(j+1)}(0) = x^{(j)}(0) - \left[I - \alpha J^{(j)}(T)\right]^{-1} \left[x^{(j)}(0) - x^{(j)}(T)\right] \tag{4.31}$$

where $0 \le \alpha \le 1$. When an iteration result is close to the final solution, α should be close to one and Equation (4.31) reduces to the undamped Newton method. When the iteration result is far away from the final solution, α should be close to zero and then $x^{(j+1)}(0) \cong x^{(j)}(T)$, that is, the method degenerates to a regular transient analysis. Therefore, the damping parameter should be a proper function of the distance between the iteration result and the final solution. In [29], the function used is:

$$\alpha = 1 - d^m \tag{4.32}$$

where m is a positive integer and chosen experimentally to be 10. d is the distance and is given by:

$$d = \max_{n \in \{1,\, 2,\, \cdots,\, N\}} \left| x_n^{(j)}(T) - x_n^{(j)}(0) \right| \tag{4.33}$$

The above method to determine the damping parameter is implemented in the shooting method of CODECS.

4.5.3 More Heuristics for Autonomous Systems

The oscillator is an example of an autonomous system for which the period of oscillation is an unknown. Heuristics are necessary to ensure reliable convergence of Newton's algorithm for such a system. The heuristics that have been used in this implementation are described here.

A transient analysis is performed in the beginning for several periods without any sensitivity computation and Newton iteration. This ensures that the extremely fast transients in the start-up phase have died out. In this way the initial guess is closer to the solution and Newton's method should be more reliable and efficient. Also in this interval a pulse is applied to a voltage source to build up the oscillations.

The sensitivity computation is carried out for the current period to calculate the new initial state only when the error of the last period is less than an acceptable threshold. Otherwise, the transient analysis continues to the next period. d in Equation (4.33) is used as the error which indicates the distance of the latest value of the state vector from the solution. This heuristic prevents the iterations from going to a wrong solution.

As for nonautonomous systems, a damped Newton iteration is used to prevent overshoot with the regular Newton method. The damped Newton iteration equation for an autonomous system is:

$$y^{(j+1)} = y^{(j)} - \left[I' - \alpha J_T^{(j)}(T^{(j)})\right]^{-1} \left[x^{(j)}(0) - x^{(j)}(T^{(j)})\right] \qquad (4.34)$$

It should be emphasized that the kth diagonal element of $J_T^{(j)}(T^{(j)})$, $\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\, T=T^{(j)}}$, is replaced by $\frac{1}{\alpha^2}\cdot\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\, T=T^{(j)}}$. When damping is strong and α is close to zero, the kth diagonal element of $I' - \alpha J_T^{(j)}(T^{(j)})$, $\frac{1}{\alpha}\cdot\left.\frac{\partial x_k(T)}{\partial T}\right|_{x(0)=x^{(j)}(0),\, T=T^{(j)}}$, is going to be very large and the desired result $T^{(j+1)} \approx T^{(j)}$ is obtained.

Finally, the change in the period T is not allowed to exceed ten percent of the current period to prevent overshoot of the Newton iteration.

4.6 Shooting Method for Coupled Device and Circuit Simulation

In this work, the time-domain shooting method has been implemented in the coupled device and circuit simulator CODECS. As described in Chapter 3, in CODECS the circuit-level simulation engine controls the analytical device evaluator and the numerical device simulator. Accurate terminal conductances and capacitances for the numerical devices are supplied by the device-level simulator and included in the circuit-level system of equations. Following this structure, the shooting method is implemented at the circuit level in this work. Also CODECS has the standard transient analysis that is required for the shooting method. During the implementation some problems which are specific to numerical devices were found and solved.

4.6.1 Numerical Device Biasing

In CODECS, the numerical devices are biased up to an initial state starting from the equilibrium point. The equilibrium point is the device state when all the terminal voltages are zero. Usually the terminal voltages from the initial state are so large that many small voltage steps are needed for convergence of the numerical devices. At each bias increment several iterations are required for convergence. Thus, the present biasing scheme is inefficient and time consuming. Notice that at the end of jth period the final state $x^{(j)}(T)$ and the new initial state $x^{(j+1)}(0)$ are obtained and at the same time the numerical devices have been biased to $x^{(j)}(T)$ by the transient simulation. To bias the numerical devices to $x^{(j+1)}(0)$, we only need to bias them up a step $\Delta x^{(j)} = x^{(j+1)}(0) - x^{(j)}(T)$ from $x^{(j)}(T)$. Since the damping scheme in Section 4.4.2 is used in the Newton iteration, $\Delta x^{(j)}$ is always a relatively small step even when the iteration is far away from convergence. It can even be zero when the new initial state is given the value of the final state of the last period, i.e., $x^{(j+1)}(0) = x^{(j)}(T)$. For small-step bias increments, the numerical device simulator converges readily. The new biasing scheme is illustrated in Figure 4.4 where $\Delta x_{eq}^{(j)} = x^{(j+1)}(0)$ is the large voltage step if the device is biased from the equilibrium point. A significant improvement in the efficiency of the shooting method for coupled device and circuit simulation is achieved.

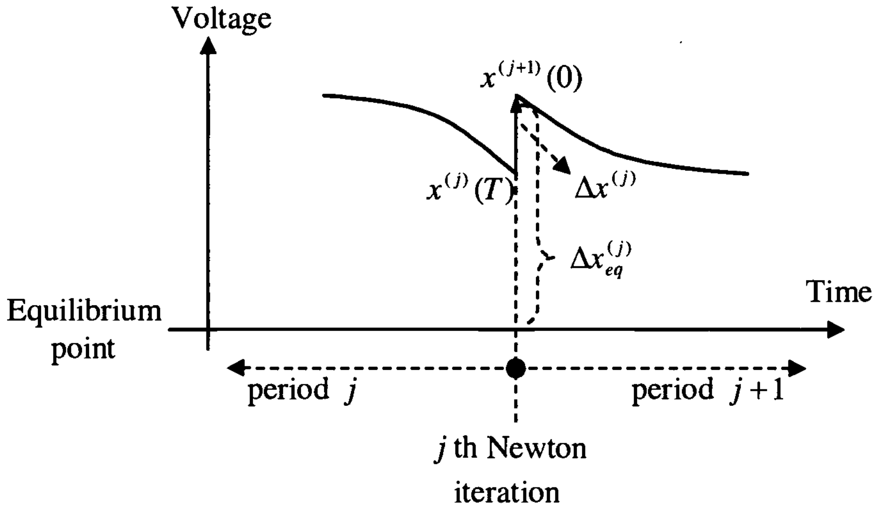


Figure 4.4: At the end of jth period the numerical devices have been biased to $x^{(j)}(T)$ by the transient simulation. To bias the numerical devices to $x^{(j+1)}(0)$, one only needs to bias them by a small step $\Delta x^{(j)} = x^{(j+1)}(0) - x^{(j)}(T)$ from $x^{(j)}(T)$ instead of the large voltage step $\Delta x_{eq}^{(j+1)} = x^{(j+1)}(0)$ from the equilibrium point.

4.6.2 Circuit Unknown Implementation vs. State Implementation

It should be realized that so far the unknowns x are the circuit states which only include the voltages across capacitors and the currents through inductors. We call this the state implementation of the shooting method and most of the previous work has used this approach. From this implementation approach, the Newton iteration at the end of a one-period transient can only provide the new initial values for capacitor voltages and inductor currents. However, to start a new transient simulation other devices have to be evaluated based on the final values of the last transient analysis. Obviously there is an inconsistency between these new initial conditions which represent the circuit solution at the $t = 0$ time point. This inconsistency causes a discontinuity between the $t = 0$ time point and the first time point after that. The symptom is that the local error can not be reduced using smaller time steps. Finally the simulator issues a "Time step too small" error and the simulation is aborted. A circuit description that includes only analytical devices can handle this discontinuity better and the simulation is seldom aborted. However, for numerical devices, this problem occurs frequently.

A solution to this problem is motivated from the ordinary transient simulation. An ordinary transient simulation starts with a DC analysis. The DC solution vector of circuit unknowns is used as the initial condition for the transient analysis. These circuit unknowns include the nodal voltages and currents through inductors and voltage sources. Since all the initial conditions of devices are derived from the same vector of circuit unknowns, the initial values for capacitor voltages and inductor currents are consistent with the ones for other devices. If the new transient analysis in the shooting method also started from a solution vector of circuit unknowns, the problem due to the inconsistent initial conditions can be overcome. This requires that the circuit unknowns are used as the unknowns in the shooting method instead of the circuit states. Then the Newton iteration of the shooting method can provide a solution vector of the circuit unknowns from which the initial conditions of all devices can be derived. This approach is referred to as the circuit unknown implementation. In this approach, the sensitivity matrix to the circuit unknowns are required and can also be computed efficiently by the sensitivity circuits described in Section 4.3.

Since the unknowns for the shooting method now are the circuit unknowns instead of the circuit states, the implementation of the state elimination discussed in Section 4.5.1 needs further explanation for the circuit unknown approach. Let us recall the computation of the sensitivity matrix along with the transient analysis in Section 4.3 and rewrite Equation (4.25) here:

$$(\frac{\partial f}{\partial \dot{x}} \cdot \alpha + \frac{\partial f}{\partial x})z = -\frac{\partial f}{\partial \dot{x}} \beta_z \tag{4.35}$$

where z is the sensitivity matrix at the current time point and β_z is a function of z at previous time points. When forming the right-hand side of Equation (4.35), only the sensitivities of the components corresponding to the considered states are loaded. Then the effect of other states is eliminated from the computation of the sensitivity matrix.

Because the unknowns have changed, Equations (4.28) and (4.29) that are used to calculate the sensitivity of the final values to the unknown period for autonomous systems are invalid for the circuit unknown implementation. However, again the sensitivities can be obtained by simple calculations. Note that the sensitivity of the final values to the unknown period is just the time derivative evaluated at the end of the period, i.e.,

$$\frac{\partial x(T)}{\partial T} = \left. \frac{\partial x(t)}{\partial t} \right|_{t=T} \tag{4.36}$$

This time derivative can be numerically approximated by an integration formula. The natural choice is the use of the same formula as the transient simulation since the coefficients of the integration formula are already available. When using this method, to ensure the accuracy of the approximation it is necessary to use a small enough time step during the transient simulation. In fact, Equations (4.28) and (4.29) are using the same method because the capacitor currents i_c and the inductor voltages v_l in those equations are calculated in device evaluation routines by applying the same integration formula on the capacitor voltages v_c and the inductor currents i_l.

4.7 Summary

In this chapter, the periodic steady-state problem was formulated as a periodic boundary constraint equation. The root of this implicit nonlinear equation provides the solution and the method is referred to as a shooting method. When Newton's method is applied to the solution of these equations a sensitivity matrix computation is needed. The sensitivity matrix can be obtained simultaneously with the transient analysis. By fixing the value of one state variable and adding the period as an unknown, the shooting method can be modified to handle autonomous systems such as oscillators. Modifications and heuristics that have been applied to improve the reliability and convergence of the Newton shooting algorithm have been described.

When the shooting method is implemented in a coupled device and circuit simulator, two problems associated with numerical devices need to be solved. These include a scheme to bias the numerical devices to the new initial condition and a circuit unknown implementation to eliminate the inconsistency and discontinuity in the standard state implementation. These modifications are essential for efficient and reliable time-domain periodic steady-state analysis for coupled device and circuit simulation.

Chapter 5

FREQUENCY-DOMAIN HARMONIC BALANCE METHOD

5.1 Introduction

Harmonic balance (HB) is a nonlinear frequency-domain method [5]. This name is due to the original view that this method balanced the currents between the linear and nonlinear subcircuits in the frequency domain. An application of the HB method yields a system of equations for a circuit in the frequency domain. This formulation provides significant advantages in terms of accuracy and efficiency compared with the time-domain method.

First consider the problems with the time-domain method for high frequency RF applications. From Section 2.1, we can see that the mixer converts the RF signal to the IF frequency band by mixing with the LO signal. Usually the frequency separation between these signals is very wide and can be of several orders of magnitude. In other words, the period of the lowest frequency is several orders larger than the period of the highest frequency. To capture the steady state of this circuit, even with the time-domain shooting method that is used to speed up the periodic steady state simulation, tens of periods of the lowest frequency have to be simulated. To resolve the highest frequency signal, adequate time points should be sampled in one period of this signal. As a consequence, a large number of time points have to be simulated. Furthermore, if the frequencies are incommensurate (two frequencies are said to be incommensurate if their ratio is not a rational number), strictly speaking it is impossible to represent the steady-state waveforms in the time domain. To obtain accurate results in the time domain, the time step should be very small and simulations should be carried out with tight tolerances. These require a long simulation time. Also the time-domain method generally has difficulty in handling frequency domain device models used in RF applications.

The harmonic balance method overcomes these difficulties by solving the system of equations in the frequency domain. The periodic and quasiperiodic steady-state solutions are directly computed. Since the system is formulated in the frequency domain, the HB method can handle multi-tone widely separated frequencies and incommensurate frequencies naturally. It is ideal for handling models represented in the frequency domain and provides accurate frequency-domain solutions.

In the harmonic balance method, the circuit waveforms are represented by the Fourier sine and cosine series. Then the unknowns are the frequency-domain Fourier coefficients instead of the time-domain waveforms. This approximation of a time-domain waveform as a Fourier series naturally and efficiently guarantees that the solution obtained is indeed the periodic or quasiperiodic steady state of the system. Since the coefficients of the steady-state response are an algebraic function of the coefficients of the stimulus, the dynamic aspect of the problem is eliminated. Therefore, the nonlinear integro-differential equations that describe a circuit are converted by the Fourier transform into a system of nonlinear algebraic equations whose solution is the steady-state response of the circuit. Usually these equations are solved iteratively by a Newton method.

This chapter is organized as follows. The discrete Fourier transform for a periodic signal is presented in Section 5.2. In Section 5.3, the quasiperiodic steady-state problem is described. Harmonic truncation methods that choose the significant frequency components for both periodic and quasiperiodic signals are described in Section 5.4. Frequency remapping transforms a quasiperiodic problem to a simpler periodic one which is easier to solve by the harmonic balance method. In Section 5.5, the basic theory on remapping is discussed and the remapping scheme is illustrated by an example. The general form of the harmonic balance equations is derived in Section 5.6 and the solution of these equations with Newton's method is detailed in Section 5.7. The harmonic balance techniques for autonomous systems are reviewed in Section 5.8. This chapter is summarized in Section 5.9.

5.2 Discrete Fourier Transform

Since the harmonic balance method requires the system of equations in the frequency domain, the Fourier transform has to be used to transform the circuit unknowns between the time and frequency domains. Because the circuit unknowns in the time domain are real-valued waveforms, a simplified version of the standard Discrete Fourier Transform is introduced here.

Consider a periodic waveform $x(t)$ with period T. This waveform can be represented by a Fourier series with the fundamental frequency $\omega_1 = 2\pi / T$:

$$x(t) = X_0 + \sum_{m=1}^{\infty} (X_m^C \cos \omega_m t + X_m^S \sin \omega_m t) \tag{5.1}$$

where $\omega_m = m\omega_1$ is the mth harmonic, X_0 is the DC value, and X_m^C and X_m^S are the Fourier coefficients of the cosine and sine terms, respectively. To represent the problem on a computer, we have to truncate the harmonics to a finite set. If the energy in the harmonics higher than M is negligible, the waveform $x(t)$ can be approximated as:

$$x(t) \approx X_0 + \sum_{m=1}^{M} (X_m^C \cos \omega_m t + X_m^S \sin \omega_m t) \tag{5.2}$$

As the spectrum of $x(t)$ is finite, it is possible to sample the waveform at a finite number of time points and calculate its Fourier coefficients. Since the number of unknown Fourier coefficients is $2M+1$, at least the same number of time points have to be sampled on the waveform. This results in a set of $2M+1$ equations which in the matrix form are given as:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{2M-1} \\ x_{2M} \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} X_0 \\ X_1^C \\ X_1^S \\ \vdots \\ X_M^C \\ X_M^S \end{bmatrix} \tag{5.3}$$

where

$$\Gamma^{-1} = \begin{bmatrix} 1 & \cos\omega_1 t_0 & \sin\omega_1 t_0 & \cdots & \cos\omega_M t_0 & \sin\omega_M t_0 \\ 1 & \cos\omega_1 t_1 & \sin\omega_1 t_1 & \cdots & \cos\omega_M t_1 & \sin\omega_M t_1 \\ 1 & \cos\omega_1 t_2 & \sin\omega_1 t_2 & \cdots & \cos\omega_M t_2 & \sin\omega_M t_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\omega_1 t_{2M-1} & \sin\omega_1 t_{2M-1} & \cdots & \cos\omega_M t_{2M-1} & \sin\omega_M t_{2M-1} \\ 1 & \cos\omega_1 t_{2m} & \sin\omega_1 t_{2M} & \cdots & \cos\omega_M t_{2M} & \sin\omega_M t_{2M} \end{bmatrix} \tag{5.4}$$

and x_s is the value of $x(t)$ at the time sample point t_s ($s = 0, 1, 2, \ldots 2M$). In our implementation, we simply choose time points such that they are equally spaced within the period, i.e. $t_s = sT/(2M+1)$ ($s = 0, 1, 2, \ldots 2M$). Equation (5.3) can be compactly written as $x = \Gamma^{-1}X$. Γ is the inverse of Γ^{-1}. Γ and Γ^{-1} are a discrete Fourier transform pair which transform a signal between the time-domain samples and the frequency-domain Fourier coefficients.

5.3 Quasiperiodic Steady-State Problem

If the frequencies of the input periodic sources for a nonlinear circuit are unrelated, the steady-state response of the circuit is quasiperioidic. The spectrum of a quasiperiodic response has significant energy at the frequencies of a linear combination of the fundamental frequencies. Consider H linearly independent fundamental frequencies Ω_1, Ω_2, ..., Ω_H, the quasiperiodic waveform $x(t)$ with these H fundamentals can be expressed as:

$$x(t) = X_0 + \sum_{m=1}^{M} (X_m^C \cos\omega_m t + X_m^S \sin\omega_m t) \tag{5.5}$$

where X_0 is the DC component and

$$\omega_m \in \{\omega \mid \omega = k_1\Omega_1 + k_2\Omega_2 + \cdots + k_H\Omega_H\,;\, k_1, k_2, \ldots, k_H \in Z\} \qquad (5.6)$$

which is an integer combination of the fundamentals Ω_1, Ω_2, ..., Ω_H. An order number equal to the absolute sum of the fundamental indices k_h can be assigned to each frequency component in Equation (5.6). For example, the frequency component $\omega_m = k_1\Omega_1 + k_2\Omega_2 + \cdots + k_H\Omega_H$ has an order number of $|k_1| + |k_2| + \cdots + |k_H|$. For a physical system with a finite energy, similar to the periodic case where the amplitudes at the high-order harmonics decrease rapidly, the frequency components in Equation (5.6) with large enough order numbers should be negligible.

5.4 Harmonic Truncation

To represent the harmonic balance problem on a computer, a finite number of frequency components have to be chosen as in Equations (5.2) and (5.5). This is done by keeping the frequency components with significant energy and truncating the ones with negligible energy. Therefore, the error introduced by the harmonic truncation is negligible

For a periodic signal there is only one fundamental frequency, hence, the truncation is straightforward. Practically the high order harmonics have negligible energy, so only the DC and low order harmonics are kept as in Section 5.2. The frequency set with the fundamental frequency Ω and a truncation of order P is given by:

$$T_P = \{\omega \mid \omega = k\Omega, 0 \le k \le P\} \qquad (5.7)$$

Obviously the total number of frequencies generated by this scheme is $P+1$. Figure 5.1 illustrates the truncation of order $P=9$ for a periodic signal.

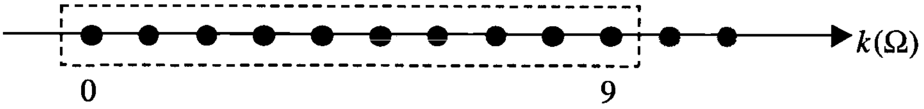


Figure 5.1: The truncation scheme of order $P=9$ for a periodic signal.

For a quasiperiodic signal with multiple fundamental frequencies, the truncation should be chosen carefully such that all frequencies containing significant energy are kept in the finite set and the same frequency is not chosen twice. Two popular methods for harmonic truncation are the box and diamond truncation methods [5]. In the box truncation, an order P truncation is given by:

$$B_P = \{\omega \mid \omega = k_1\Omega_1 + k_2\Omega_2 + \cdots + k_H\Omega_H, |k_1| \le P, |k_2| \le P, \cdots, |k_H| \le P\} \qquad (5.8)$$

$$k_1 \ge 0, k_h \ge 0 \text{ if } k_1 = k_2 = \cdots = k_{h-1} = 0 \text{ for } 1 \le h \le H \qquad (5.9)$$

Equation (5.9) ensures that the image frequencies which are negative to each other are not both included for the real-value waveforms. The box truncation scheme generates a total of $\frac{1}{2}\left((2P+1)^H + 1\right)$ frequencies. Figure 5.2 illustrates the box truncation of order $P=3$ for the $H=2$ case. In this two-fundamental case, the truncation results in a rectangular grid on frequency indices as shown in Figure 5.2.

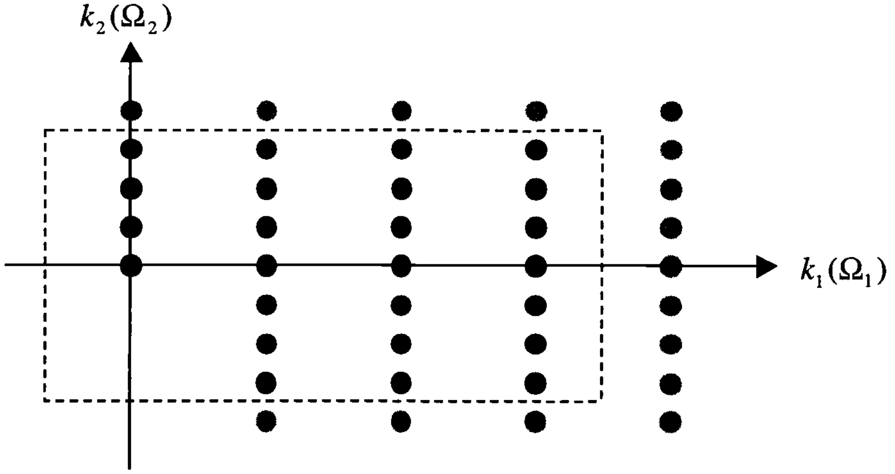


Figure 5.2: The two-tone ($H = 2$) box truncation scheme of order $P = 3$.

In a P order diamond truncation, any frequency component of the chosen frequency set has an order number less than or equal to P:

$$D_P = \{\omega \,|\, \omega = k_1\Omega_1 + k_2\Omega_2 + \cdots + k_H\Omega_H, |k_1| + |k_2| + \cdots + |k_H| \le P\} \qquad (5.10)$$

The same constraint Equation (5.9) should also be applied here to exclude one of the image frequencies. The diamond truncation scheme generates approximately a total of $\dfrac{2^{H-1}P^H}{H!}$ frequencies. Figure 5.3 illustrates the diamond truncation of order $P = 3$ for the $H = 2$ case. In this two-fundamental case, the truncation results in a "diamond" grid as shown in Figure 5.3. The diamond truncation is more efficient than the box truncation since more higher order frequency components containing negligible energy are excluded.

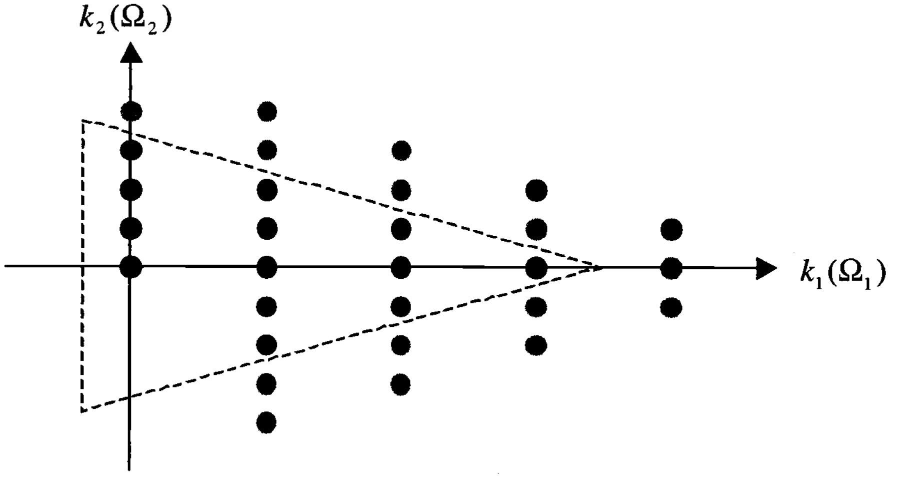


Figure 5.3: The two-tone $(H = 2)$ diamond truncation scheme of order $P = 3$.

5.5 Frequency Remapping

Once the spectrum of a signal has been truncated to a finite-frequency set using one of the truncation schemes in above section, the waveform should be sampled at a finite number of time points and its Fourier coefficients can be calculated as in Equations (5.3) and (5.4). For quasiperiodic signals, if the time-points are not chosen carefully, for example time points are simply equally spaced, the transform matrix Γ^{-1} in Equations (5.3) and (5.4) can be ill-conditioned [5]. This ill-conditioning can make inverting Γ^{-1} difficult and magnify numerical and aliasing errors. Several techniques have been developed to tackle this problem. The most efficient and attractive approach is the frequency remapping method [6]. As we can see in the following explanation, after frequency remapping the quasiperiodic signals can be treated essentially identical to periodic signals. The same discrete Fourier transform as in Section 5.2 can be used for quasiperiodic signals without the ill-conditioning problem.

Another approach to solve the quasiperiodic problem is the use of a complicated multi-dimensional discrete Fourier transform [38] to transform the quasiperiodic signals between the time and frequency domains. In this work the simple frequency remapping method is used.

5.5.1 Basic Theory on Frequency Remapping

Let us examine a quasiperiodic signal $x(t)$ applied to a algebraic nonlinearity $g(x)$. The complex form of Equation (5.5) is:

$$x(t) = \sum_{m=-M}^{M} X_m \exp(j\omega_m t) \tag{5.11}$$

where $\omega_{-m} = -\omega_m$, $X_{-m} = X_m{}^*$, and X_0 is real such that $x(t)$ is a real-value signal. Using the Taylor series to expand the nonlinearity around X_0 and applying the multinomial theorem, $g(x)$ is given by:

$$g(x(t)) = \sum_{k=0}^{\infty} \frac{g^{(k)}(X_0)}{k!} \left(\sum_{m=-M}^{M} X_m \exp(j\omega_m t) - X_0 \right)^k \tag{5.12}$$

$$= \sum_{k=0}^{\infty} \frac{g^{(k)}(X_0)}{k!} \sum_{n_1+\cdots+n_{M+1}=k} \frac{k!}{n_1!\cdots n_{M+1}!} X_{-M}^{n_1} \cdots X_M^{n_{M+1}} \exp(j(n_1\omega_{-M} + \cdots n_{M+1}\omega_M))$$

where $n_i = m_i + M + 1$. From Equation (5.12), one can see that the amplitude of the frequency component $\exp(j(n_1\omega_{-M} + \cdots + n_M\omega_M))$ is not a function of the frequencies $\omega_1, \cdots, \omega_M$. From this one should not conclude that the Fourier coefficient at $\omega = n_1\omega_{-M} + \cdots + n_M\omega_M$ is independent of the frequencies $\omega_1, \cdots, \omega_M$. This is because different integer combinations of $\omega_1, \cdots, \omega_M$ may result in the same frequency value. Recall from Equation (5.6) that ω_m is an integer combination of the fundamentals Ω_1, Ω_2, ..., Ω_H. Since these fundamentals are linearly independent, the possibility of resulting in the same frequency value from different integer combinations of $\omega_1, \cdots, \omega_M$ only depends on indices k_1, k_2, ..., k_H, i.e., the harmonic and mixing relation among $\omega_1, \cdots, \omega_M$. Therefore, the Fourier coefficients of the output of the nonlinearity are only independent of the fundamental frequencies Ω_1, Ω_2, ..., Ω_H.

For the purposes of evaluating the nonlinear devices, the actual fundamental frequencies are of no importance and can be chosen freely. In particular, the fundamentals can be chosen to be multiples of some arbitrary frequency so that the resulting signals will be periodic. We can see the ill-conditioning problem with the quasiperiodic Fourier transform is avoided because the original frequency set is mapped to a new one such that it becomes a periodic case. The Fourier coefficients of the nonlinear device can be evaluated using the new frequency set. The actual time-domain waveform is obtained by evaluating Equation (5.5) with the resulting Fourier coefficients and the actual frequency set. This is the basic theory of frequency remapping.

The discussion above can be summarized in matrix form:

$$\hat{\Gamma} g(\hat{\Gamma}^{-1} X) = \Gamma g(\Gamma^{-1} X) \tag{5.13}$$

where Γ and $\hat{\Gamma}$ are the Fourier transform matrix using the original frequency set and the remapped frequency set, respectively. If the Fourier coefficients of the input signal are known, then the Fourier coefficients of the output of a nonlinearity can be obtained using the Fourier transform based on another frequency set. Notice that in order to use the same frequency set for Γ and Γ^{-1}, the components of $g(x)$ in Equation (5.12) at frequencies other than ω_m are ignored. This is a good approximation because in the harmonic balance analysis the frequency set of ω_m is large enough such that the magnitudes of the components that are ignored are very small.

5.5.2 Remapping Scheme and Example

A remapping scheme maps the original frequency set to a new one. The requirements for the scheme are that the resulting signal in the new frequency set is periodic and no two original frequencies can be mapped to the same new frequency. The optimal scheme will generate a new frequency set which is densely packed, where each remapped frequency corresponds to an original frequency such that there is a one-on-one relationship between the original and remapped frequencies.

In [5, 6], the remapping principle was discussed and various remapping schemes for different frequency truncations were introduced. Here, an example is used to illustrate how the frequencies are remapped. Consider a box truncation of order $P = 2$ shown in Figure 5.4. The fundamental frequencies are f_1=1GHz and f_2=1MHz. Based on the box truncation remapping scheme in [6], the new fundamental frequencies are $\hat{f}_1 = 2P + 1 = 5$ and $\hat{f}_2 = 1$, respectively. In Table 5.1 the actual frequencies and remapped integer frequencies are listed. The remapped integer frequencies are also labeled on the frequency grids in Figure 5.4. The actual and remapped spectral representations of the truncated quasiperiodic waveform are illustrated in Figure 5.5. Obviously the originally sparse spectrum is densely packed by this frequency remapping scheme.

Remapped f	k_1	k_2	Actual f (GHz)
0	0	0	0
1	0	1	0.001
2	0	2	0.002
3	1	-2	0.998
4	1	-1	0.999
5	1	0	1
6	1	1	1.001
7	1	2	1.002
8	2	-2	1.998
9	2	-1	1.999
10	2	0	2
11	2	1	2.001
12	2	2	2.002

Table 5.1: Correspondence between the actual and remapped integer frequencies for the box truncation remapping scheme.

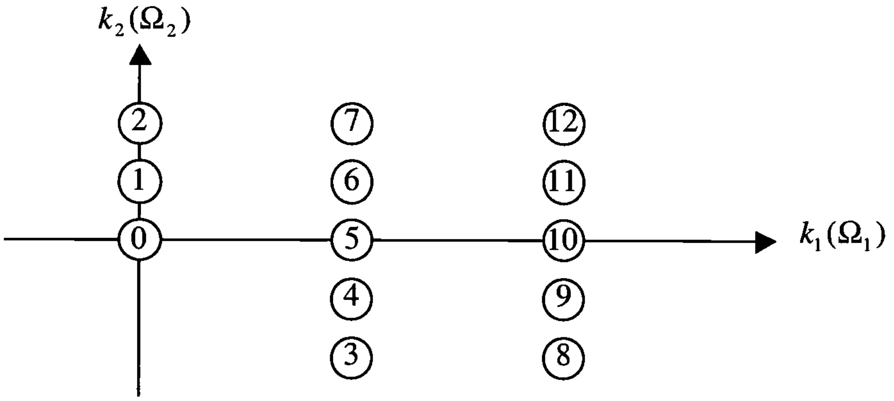


Figure 5.4: Remapping for the box truncation of order $P=2$. The numbers in the circles are the remapped integer frequencies corresponding to each (k_1,k_2) mixing term.

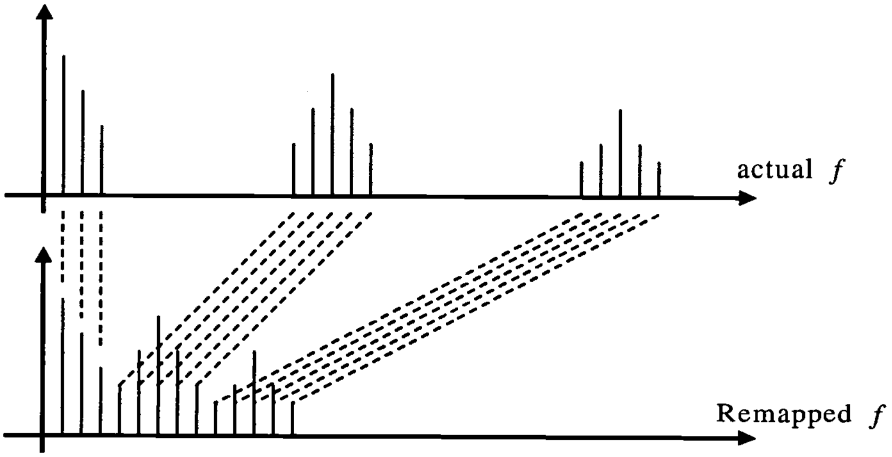


Figure 5.5: The actual and remapped spectral representations of a quasiperiodic signal (not to scale) with an order-2 box truncation.

5.6 Formulating Harmonic Balance Equations

The circuit behavior can be described in the time domain by a system of equations:

$$f(x(t),t) = i(x(t)) + \frac{d}{dt}q(x(t)) + s(t) = 0 \qquad (5.14)$$

where x is the vector of circuit waveforms, i is the vector of contributions from nonreactive elements to the circuit equations such as linear and nonlinear resistors, q is the vector of contributions from reactive elements such as capacitors and inductors, and s is a quasiperiodic stimulus vector.

When applying harmonic balance to Equation (5.14), x, s, and $f(x)$ are transformed into the frequency domain using the Fourier transform in Equation (5.4). The frequency domain representation of Equation (5.14) is:

$$F(X) = \Gamma i(\Gamma^{-1}X) + \Omega\Gamma q(\Gamma^{-1}X) + S = 0 \qquad (5.15)$$

where X is the vector of circuit waveforms in the frequency domain represented by their Fourier coefficients, Γ and Γ^{-1} are the Fourier transform matrix pair, S is the frequency-domain representation of the stimulus vector s, Ω is a block-diagonal matrix representing the time derivative operation:

$$\Omega = \begin{bmatrix} 0 & & & \\ & \varpi_1 & & \\ & & \ddots & \\ & & & \varpi_M \end{bmatrix}, \quad \varpi_m = \begin{bmatrix} 0 & \omega_m \\ -\omega_m & 0 \end{bmatrix} \qquad (5.16)$$

Note that $\Gamma\left(\frac{d}{dt}q(x)\right) = \Omega\Gamma q(\Gamma X)$ follows the differentiation rule of the Fourier series. Equation (5.15) is a nonlinear algebraic equation that is solved to obtain the frequency-domain solution X. If frequency remapping is used for the quasiperiodic case, Γ and Γ^{-1} are the Fourier transform matrix pair using the remapped frequency set. It is important to realize that frequency remapping is only valid for evaluating algebraic nonlinearities. So when assembling Ω and S, the original frequency set must be used. Equation (5.15) shows that in the harmonic balance method the nonlinear components (i and q) are evaluated by transforming the spectrum of circuit unknowns X into time-domain waveforms x, calculating the response waveforms $i(x)$ and $q(x)$, and then transforming these waveforms back into the frequency domain.

Another form of the harmonic balance equation is obtained by transforming Equation (5.15) into the time domain by multiplication with Γ^{-1}

$$i(x) + \Gamma^{-1}\Omega\Gamma q(x) + s = 0 \tag{5.16}$$

Here x and s are the vectors of $2M+1$ time-domain samples of unknown waveforms and stimuli, respectively, if M frequencies are chosen by harmonic truncation. This form clearly shows the relation between the harmonic balance equation and its time-domain counterpart. We can see that the time derivative in Equation (5.14) is approximated by $\Gamma^{-1}\Omega\Gamma$ in the harmonic balance method.

5.7 Solving Harmonic balance Equations with Newton's Method

In this section, the harmonic balance equation is solved by Newton's method. The assembly of system matrices used in the Newton iteration equation is discussed in detail.

5.7.1 Newton Iteration Equation for the Harmonic Balance Problem

Since the harmonic balance method results in a system of nonlinear algebraic equations, Newton's method can be applied to solve these equations iteratively. The $(j+1)$th iteration using Newton's method for Equation (5.15) is:

$$X^{(j+1)} = X^{(j)} - \left(\left. \frac{\partial F}{\partial X} \right|_{X^{(j)}} \right)^{-1} F(X^{(j)}) \tag{5.17}$$

where $X^{(j+1)}$ is the vector of unknowns at iteration $j+1$. The initial guess $X^{(0)}$ takes the DC solution of the system. The iterative process given by Equation (5.17) will converge quadratically when the iterates are close to the final solution.

The Jacobian matrix $\frac{\partial F}{\partial X}$ of the harmonic balance problem in Equation (5.17) can be derived by applying the chain rule to differentiate Equation (5.15):

$$\frac{\partial F}{\partial X} = \Gamma \frac{\partial i(x)}{\partial x} \frac{\partial x}{\partial X} + \Omega\Gamma \frac{\partial q(x)}{\partial x} \frac{\partial x}{\partial X} = \Gamma G \Gamma^{-1} + \Omega \Gamma C \Gamma^{-1} \tag{5.18}$$

where $G = \frac{\partial i(x)}{\partial x}$ and $C = \frac{\partial q(x)}{\partial x}$ are the linearized conductance and capacitance matrices, respectively. These linearized matrices can be obtained by multiple small-signal analyses around the operating points specified at each time sample point.

To illustrate the solution procedure, consider a nonlinear capacitor example. Let its current and voltage in the frequency domain be represented by I and V and in the time domain by i and v, respectively. The frequency domain equation for the nonlinear capacitor is given by the second term in Equation (5.15):

$$I = \Omega\Gamma q(\Gamma^{-1}V) \tag{5.19}$$

The ($j+1$)th iteration equation using Newton's method is:

$$I^{(j+1)} = \Omega\Gamma C(v^{(j)})\Gamma^{-1}V^{(j+1)} + \Omega\Gamma(q(v^{(j)}) - C(v^{(j)})v^{(j)}) \tag{5.20}$$

where $v^{(j)} = \Gamma V^{(j)}$ and $C(v^{(j)}) = \left.\frac{\partial q}{\partial v}\right|_{v^{(j)}}$ is the linearized capacitance in the time domain. To form the linear equations for the ($j+1$)th iteration, $\Omega\Gamma C(v^{(j)})\Gamma^{-1}$ is stamped into the harmonic balance Jacobian and the second term $\Omega\Gamma(q(v^{(j)}) - C(v^{(j)})v^{(j)})$ is loaded into the right-hand side. The derivation for a nonlinear resistor is similar.

5.7.2 System Matrices in Our implementation

The actual structure of matrices Γ, G, Ω, and C depends on how the time samples and Fourier coefficients of all circuit waveforms are arranged in their vector forms. Theoretically there is no advantage of one arrangement over another except that some of them make these matrices easier to assemble and manipulate. As shown in Section 5.2, if M frequencies (excluding DC) are chosen to represent the circuit waveforms in the frequency domain, for each circuit waveform the number of unknown Fourier coefficients is $2M+1$ and the same number of time points have to be sampled on the waveform. If there are N waveforms in the circuit, the total number of unknown Fourier coefficients is $N(2M+1)$. Then the number of equations from the harmonic balance analysis is also $N(2M+1)$ and the size of matrixes Γ, G, Ω, and C is $N(2M+1)\times N(2M+1)$. The following Fourier transform shows the arrangement of these $N(2M+1)$ Fourier coefficient unknowns and time sample points in our implementation:

$$x=\begin{bmatrix}\bar{x}_0\\ \bar{x}_1\\ \bar{x}_2\\ \vdots\\ \bar{x}_{2M-1}\\ \bar{x}_{2M}\end{bmatrix}=\Gamma^{-1}\begin{bmatrix}\bar{X}_0\\ \bar{X}_1^C\\ \bar{X}_1^S\\ \vdots\\ \bar{X}_M^C\\ \bar{X}_M^S\end{bmatrix}=X \tag{5.21}$$

where $\bar{x}_s=\begin{bmatrix}x_{1,s}\\ x_{2,s}\\ \vdots\\ x_{N,s}\end{bmatrix}$, $\bar{X}_0=\begin{bmatrix}X_{1,0}\\ X_{2,0}\\ \vdots\\ X_{N,0}\end{bmatrix}$, $\bar{X}_m^C=\begin{bmatrix}X_{1,m}^C\\ X_{2,m}^C\\ \vdots\\ X_{N,m}^C\end{bmatrix}$, and $\bar{X}_m^S=\begin{bmatrix}X_{1,m}^S\\ X_{2,m}^S\\ \vdots\\ X_{N,m}^S\end{bmatrix}$. $x_{n,s}$ is the value of $x_n(t)$ at the time point t_s (s = 0, 1, 2, ...$2M$), $X_{n,0}$ is the DC value of $x_n(t)$, and $X_{n,m}^C$ and $X_{n,m}^S$ are the Fourier coefficients of $x_n(t)$ at the mth frequency ω_m for cosine and sine terms, respectively. The Fourier transform matrix Γ^{-1} is:

$$\Gamma^{-1}=\begin{bmatrix}I & C_{1,0} & S_{1,0} & \cdots & C_{M,0} & S_{M,0}\\ I & C_{1,1} & S_{1,1} & \cdots & C_{M,1} & S_{M,1}\\ I & C_{1,2} & S_{1,2} & \cdots & C_{M,2} & S_{M,2}\\ \vdots & \vdots & \vdots & & \vdots & \vdots\\ I & C_{1,2M-1} & S_{1,2M-1} & \cdots & C_{M,2M-1} & S_{M,2M-1}\\ I & C_{1,2M} & S_{1,2M} & \cdots & C_{M,2M} & S_{M,2M}\end{bmatrix} \tag{5.22}$$

where I is the $N \times N$ identity matrix, $C_{m,s} = \begin{bmatrix} \cos\omega_m t_s & & \\ & \ddots & \\ & & \cos\omega_m t_s \end{bmatrix} = \cos\omega_m t_s \cdot I$, and $S_{m,s} = \sin\omega_m t_s \cdot I$. Γ is obtained by inverting Γ^{-1}. According to the unknown arrangement in Equation (5.21), the structure of Ω is:

$$\Omega = \begin{bmatrix} 0 & & & \\ & \varpi_1 & & \\ & & \ddots & \\ & & & \varpi_M \end{bmatrix}, \quad \varpi_m = \begin{bmatrix} 0 & \overline{\omega}_m \\ -\overline{\omega}_m & 0 \end{bmatrix} \tag{5.23}$$

where 0 is an $N \times N$ zero matrix, $\overline{\omega}_m = \begin{bmatrix} \omega_m & & \\ & \ddots & \\ & & \omega_m \end{bmatrix} = \omega_m \cdot I$, and I is the $N \times N$ identity matrix. As mentioned in Section 5.6, it is important to realize that here ω_m is the frequency from the original frequency set rather than the remapped frequency set. G and C in Equation (5.18) are block diagonal matrices, with the diagonal elements representing the circuit linearized at the sampled time points:

$$G = \begin{bmatrix} \overline{G}_0 & & & \\ & \overline{G}_1 & & \\ & & \ddots & \\ & & & \overline{G}_{2M} \end{bmatrix}, \quad C = \begin{bmatrix} \overline{C}_0 & & & \\ & \overline{C}_1 & & \\ & & \ddots & \\ & & & \overline{C}_{2M} \end{bmatrix} \tag{5.24}$$

where $\overline{G}_s = \left.\frac{\partial i(x)}{\partial x}\right|_{x_s}$ and $\overline{C}_s = \left.\frac{\partial q(x)}{\partial x}\right|_{x_s}$ are the $N \times N$ linearized conductance and capacitance matrices at time point t_s, respectively. The correspondence between the block matrices on the diagonal of G and the sampled points on circuit waveforms is illustrated in Figure 5.6. $\overline{G}_s$ and $\overline{C}_s$ are similar to the Jacobian matrixes used in time-domain circuit analysis, hence, they are very sparse.

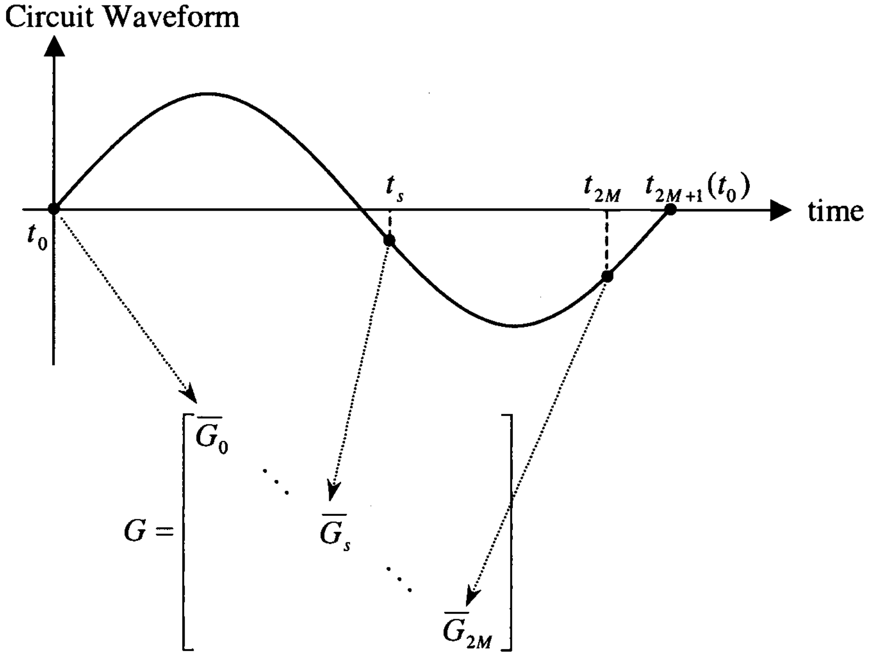


Figure 5.6: The correspondence between the block matrices on the diagonal of G and the sampled points on circuit waveforms.

It is interesting to inspect the structure of the harmonic balance Jacobian matrix in Equation (5.18) and compare it with the time-domain Jacobian matrix. Due to the Fourier transform matrices Γ and Γ^{-1}, each structural non-zero entry in the time-domain Jacobian matrix inflates into a dense $(2M+1)\times(2M+1)$ block in harmonic balance Jacobian matrix. The inflation and structure of the harmonic balance Jacobian matrix is illustrated in Figure 5.7. For a large number of unknown waveforms due to a large circuit size or device-level simulation, factorization of such a large dense harmonic balance Jacobian matrix by a direct method is prohibitively expensive. Krylov subspace methods [39-41] have been shown to be effective when applied to the solution of the large-scale system of harmonic balance equations.

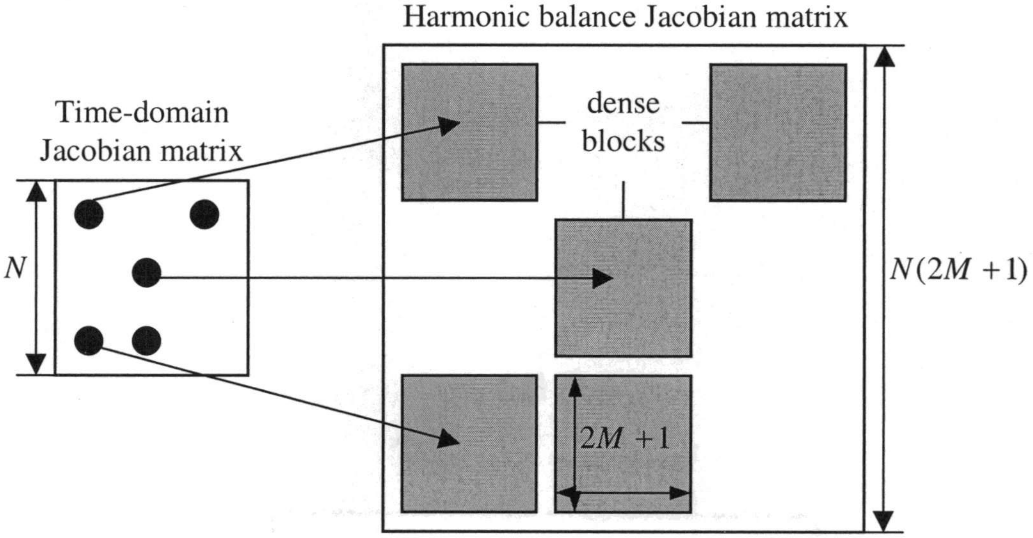


Figure 5.7: The structure of the harmonic balance Jacobian matrix. Each structural non-zero entry in the time-domain Jacobian matrix inflates into a dense $(2M+1)\times(2M+1)$ block in the harmonic balance Jacobian matrix.

5.8 Harmonic Balance Techniques for Autonomous Systems

In this section, the harmonic balance method is modified to handle autonomous systems. To improve the convergence, the concept of a probe [42] is introduced. The circuit-probe combined system and its solution methods from [42, 43] are reviewed.

5.8.1 Problem with Autonomous Systems

As discussed in Section 4.4, compared with forced circuits, autonomous systems such as oscillators have two problems associated with the steady-state analysis. The first is that the period of the steady state of an oscillator is not prefixed and should be solved in the steady-state analysis. The second is that since there is no input to fix the phase, if one steady state exists, any time shifted version is also a steady-state solution. As with the time-domain shooting method, the harmonic balance method can be modified to handle oscillators by adding the fundamental frequency ω to the list of unknowns and an equation to enforce the constraint that isolates equivalent solutions from one another.

One easy way to isolate solutions is to choose some signal in the circuit and force the sine or cosine part of its fundamental to be zero [5]. For this approach to work it is necessary that the magnitude of the chosen signal at the fundamental frequency is not zero originally. With this strategy, the harmonic balance equations for the oscillator problem are:

$$F(X,\omega) = \Gamma i(\Gamma^{-1}X) + \Omega(\omega)\Gamma q(\Gamma^{-1}X) + S = 0 \qquad (5.25)$$

$$X_{n,1}^{S} = 0$$

Where $X_{n,1}^{S}$ is the sine part of the fundamental of the chosen signal. Using Newton's method to solve the two equations, the $(j+1)$th iteration is obtained:

$$\begin{bmatrix} \dfrac{\partial F(X^{(j)},\omega^{(j)})}{\partial X} & \dfrac{\partial F(X^{(j)},\omega^{(j)})}{\partial \omega} \\ (e_{n,1}^{S})^{T} & 0 \end{bmatrix} \begin{bmatrix} \Delta X^{(j+1)} \\ \Delta \omega^{(j+1)} \end{bmatrix} = - \begin{bmatrix} F(X^{(j)},\omega^{(j)}) \\ X_{n,1}^{S\ (j)} \end{bmatrix} \tag{5.26}$$

where
$$\frac{\partial F(X,\omega)}{\partial \omega} = \frac{\partial \Omega(\omega)}{\partial \omega}\Gamma q(\Gamma^{-1}X) = \frac{\Omega(\omega)}{\omega}\Gamma q(\Gamma^{-1}X) \tag{5.27}$$

$e_{n,1}^{S}$ is the unit vector that selects the $X_{n,1}^{S}$ from the unknown vector X and $\dfrac{\partial F(X,\omega)}{\partial X}$ is given in Equation (5.18).

The above method is very restrictive on the initial guess for the oscillator frequency and circuit unknowns. Without a good initial guess it will converge to the trivial DC solution or diverge. The reason of the convergence difficulty is that the oscillator frequency and circuit unknowns are included in the same unknown vector and updated at the same time for each iteration. Before the final solution is reached, the circuit unknowns don't satisfy the KCL and may be unphysical. Updating the oscillator frequency along with these solutions causes the Newton iteration convergence difficulties.

5.8.2 Voltage Probe

To overcome the above convergence difficulty, the concept of probe is introduced in [42]. The analysis of an autonomous circuit is converted into the analysis of a series of closely related non-autonomous circuit-probe combined systems. Then the numerical efficiency of the harmonic balance method for non-autonomous circuits can be made use of.

The voltage probe is a voltage source at a specified frequency and open circuit at all other frequencies. As shown in Figure 5.8(a), a probe is composed of an ideal independent voltage source and a filter. The filter disconnects the probe from its terminal for frequencies different from that of the probe source. For the circuit-probe combination shown in Figure 5.8(b), the condition for oscillation is satisfied when the probe current is zero for a finite probe voltage. In this case, all the circuit equations are satisfied with the probe not being a part of the circuit. The frequency of the probe is taken as the oscillation frequency which is added as an unknown.

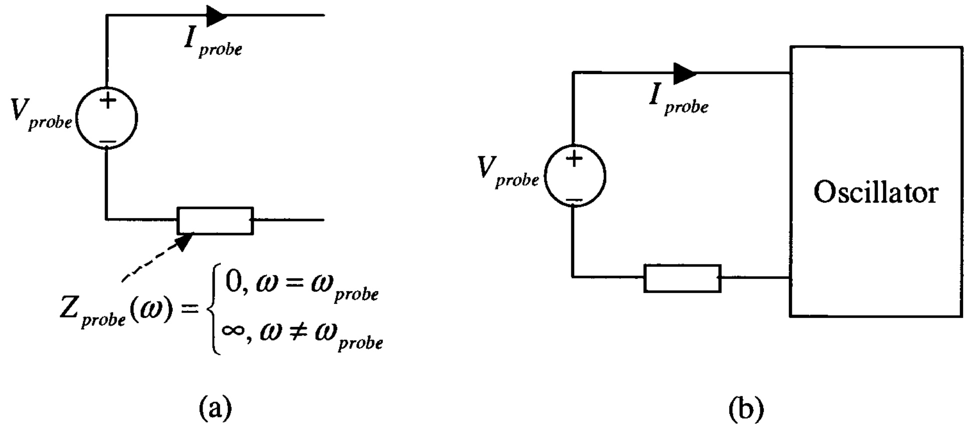

I_{probe}
V_{probe}
+
$Z_{probe}(\omega) = \begin{cases} 0, \omega = \omega_{probe} \\ \infty, \omega \neq \omega_{probe} \end{cases}$
(a)
I_{probe}
V_{probe}
+
Oscillator
(b)

Figure 5.8: (a) The voltage probe, and (b) the circuit-probe combination.

5.8.3 Two-Level Newton Method for the Circuit-Probe System

In [42] the harmonic balance analysis of an oscillator is formulated as a two-level Newton problem. The upper level solves the probe current to be zero and is given by:

$$I^C_{probe}(V^C_{probe}, \omega) = 0$$

$$I^S_{probe}(V^C_{probe}, \omega) = 0 \qquad (5.28)$$

where I^C_{probe} and I^S_{probe} are the cosine and sine components of the probe current, respectively. V^C_{probe} is the cosine component of the probe voltage and ω is the oscillation frequency. The sine component of the probe voltage, V^S_{probe}, is set to zero to fix the phase of V_{probe} and select one of the many equivalent solutions. The probe current is obtained at the lower level using a standard harmonic balance analysis on the forced system for the circuit-probe combination. The probe magnitude and frequency is solved by the upper level iteration and applied to the lower level. It should be noted that the frequency is updated only at the upper level and based on the lower level solution which satisfies KCL for the circuit-probe combination. This results in an improved convergence.

The $(j+1)$th iteration using Newton's method for Equation (5.28) is:

$$\begin{bmatrix} \dfrac{\partial I^C_{probe}}{\partial V^C_{probe}} & \dfrac{\partial I^C_{probe}}{\partial \omega} \\ \dfrac{\partial I^S_{probe}}{\partial V^C_{probe}} & \dfrac{\partial I^S_{probe}}{\partial \omega} \end{bmatrix}_i \begin{bmatrix} \Delta V^C_{probe}{}^{(j+1)} \\ \Delta \omega^{(j+1)} \end{bmatrix} = -\begin{bmatrix} I^C_{probe}(V^C_{probe}{}^{(j)}, \omega^{(j)}) \\ I^S_{probe}(V^C_{probe}{}^{(j)}, \omega^{(j)}) \end{bmatrix} \qquad (5.29)$$

Since the probe current is one of the unknowns at the lower level, the right-hand side of the above equation is available when the lower-level Newton iteration converges. The 2×2 Jacobian matrix in Equation (5.29) can also be obtained by a simple computation using the lower level solution. From Equation (5.25), the frequency domain representation of the circuit-probe combination is given as:

$$F(X,\omega)=0 \tag{5.30}$$

where X is the vector of the circuit unknowns plus the current through the voltage probe. The sensitivity with respect to V_{probe}^{C} and ω can be computed as:

$$\frac{\partial F}{\partial X}\cdot\frac{\partial X}{\partial V_{probe}^{C}}=-\frac{\partial F}{\partial V_{probe}^{C}} \tag{5.31}$$

$$\frac{\partial F}{\partial X}\cdot\frac{\partial X}{\partial \omega}=-\frac{\partial F}{\partial \omega} \tag{5.32}$$

The lower-level Jacobian matrix, $\frac{\partial F}{\partial X}$, is available in LU-factored form when the lower-level Newton iteration converges and $\frac{\partial F}{\partial \omega}$ is calculated using Equation (5.27). Since the probe voltage V_{probe}^{C} is a component of the unknown vector X, $\frac{\partial F}{\partial V_{probe}^{C}}$ is the corresponding column in $\frac{\partial F}{\partial X}$. Therefore, $\frac{\partial X}{\partial V_{probe}^{C}}$ and $\frac{\partial X}{\partial \omega}$ can be obtained from Equations (5.31) and (5.32), respectively. Since the probe current is also a component of the unknown vector X, all entries of the 2x2 Jacobian matrix in Equation (5.29) can be extracted from the sensitivity solution of Equations (5.31) and (5.32).

For the unknowns in the upper level, the probe voltage and oscillation frequency, the two-level Newton method requires an initial guess. In [42], with the initial guess for the frequency, the initial guess for the probe voltage is selected to minimize the magnitude of the probe current. The curve (a) in Figure 5.9 shows the relation between the magnitude of the probe current and the initial guess for the probe voltage. The minimum is used as the initial guess for the probe voltage. Since the goal is to solve the probe current to zero, this minimum can serve as a good starting point for the Newton iterations. However, if the initial guess for the frequency is far from the oscillation frequency, a minimum for the probe current may not exist as shown by curve (b) in Figure 5.9 and the method fails. Even when there is a minimum, the starting point selected using this scheme may not lead to convergence. As a natural extension of the two-level Newton method, a continuation method as presented in [43] can be used.

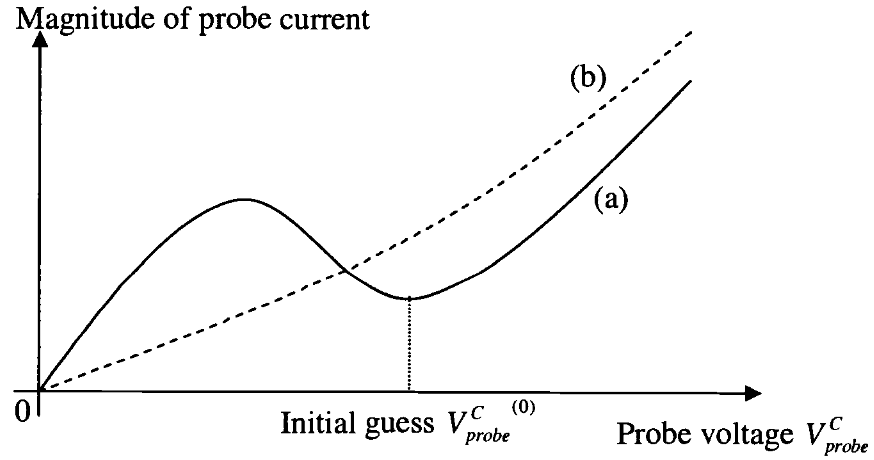


Figure 5.9: Selection of the initial guess for the probe voltage by using the minimum of the magnitude of the probe current at the initial estimate for the oscillation frequency. Curve (a) depicts the condition when there is a minimum, whereas for curve (b) no minimum exists and the scheme fails.

5.8.4 Continuation Method for the Circuit-Probe System

The continuation strategy in [43] is similar to the selection of the initial guess for the probe voltage in [42]. Instead of keeping the frequency fixed while searching for a good initial guess for the probe voltage, I_{probe}^{S} is fixed to zero and the frequency is varied. That is, a curve is traced for:

$$I_{probe}^{S}(V_{probe}^{C}, \omega) = 0 \qquad (5.33)$$

with V_{probe}^{C} as a continuation parameter. The tracing starts with a small signal for the probe voltage and the oscillation frequency estimated from a linear analysis. This method works because the actual process of oscillation build-up starts from a small amplitude. As the probe voltage increases, we let the frequency adjust to satisfy Equation (5.33). As the curve is traced, a change in sign of I_{probe}^{C} indicates that the probe current (both sine and cosine components) crosses zero and the solution at the point where the sign changes is the steady-state solution of the oscillator. Figure 5.10 illustrates typical curves traced and the corresponding solution point. Because the curve can be very complex with many turning points, the probe voltage increments should be small enough to ensure convergence. A curve tracing algorithm in [43] can be used to control the step-length automatically. Since the frequency is an unknown, a two-level Newton's formulation should be used to solve Equation (5.33) to ensure the convergence. Instead of Equation (5.33) the curve of $I_{probe}^{C}(V_{probe}^{C}, \omega) = 0$ can also be traced in this continuation method.

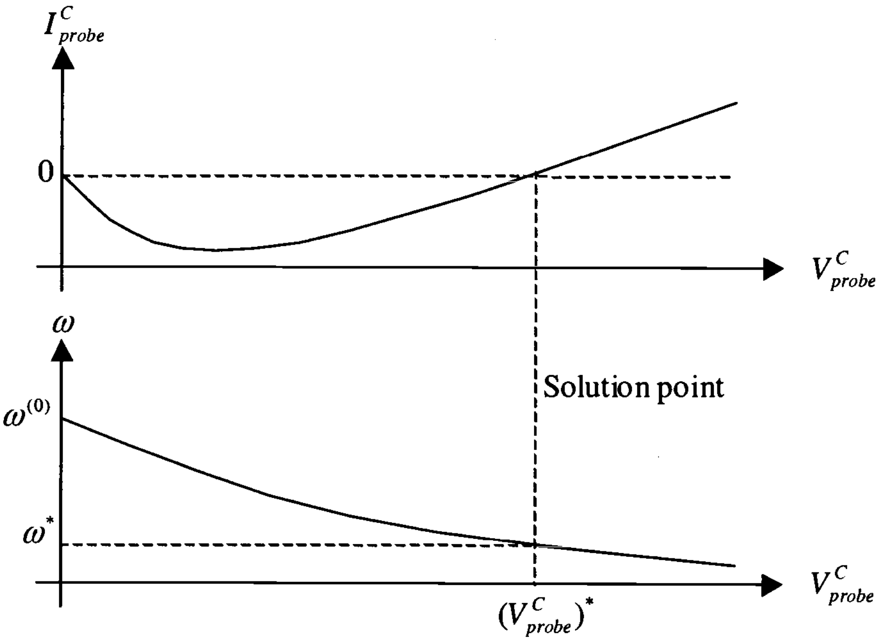


Figure 5.10: The curves traced in the continuation method. The solution is obtained when I_{probe}^{C} changes sign.

5.9 Summary

This chapter first presents an overview of the harmonic balance algorithm. The harmonic balance method overcomes several difficulties faced by time-domain methods by solving the system of equations in the frequency domain. The periodic and quasiperiodic steady-state solutions are computed directly. Multi-tone problems with widely separated frequencies and incommensurate frequencies can be easily handled. Furthermore, accurate frequency-domain solutions are obtained. Various harmonic truncation schemes are introduced. The basic theory on frequency remapping is discussed and a remapping scheme is illustrated by an example. The formulation of the harmonic balance equations and solution with Newton's method is also described. The assembly of system matrices used in the Newton iteration equation is shown in detail. To solve autonomous systems, the standard harmonic balance method has to be modified and the voltage probe method improves convergence.

Chapter 6

HARMONIC BALANCE METHOD FOR COUPLED DEVICE AND CIRCUIT SIMULATION

6.1 Introduction

The harmonic balance method has previously been applied for the simulation of semiconductor devices in [6, 7]. Although numerical devices can be connected with linear circuit elements, the emphasis there is on device internal behavior and the harmonic balance method is not extended to general coupled device and circuit simulation.

In this chapter, the harmonic balance method discussed in Chapter 5 is implemented in the coupled device and circuit simulator CODECS [2]. Three different implementation approaches are introduced. These include quasi-static (QS), non-quasi-static (NQS), and modified Volterra series (MVS) approaches. All the implementations follow the coupling simulation framework illustrated in Figure 3.3. From the circuit-level point of view, the coupled simulation can handle and interpret device-level models in different ways. As discussed later in this section, the NQS approach makes use of all the information provided by the numerical device models, the QS approach simplifies the models based on the quasi-static assumption, and the MVS approach approximates models by the modified Volterra series [44, 45]. These approaches are described and compared qualitatively in this chapter. Then in Chapter

7 simulation examples are used for verification and quantitative comparisons. It will be shown that the accuracy of the QS and MVS methods depends on the operating condition of numerical devices and that there is a tradeoff between simulation accuracy and complexity among these methods.

This chapter is organized as follows. The NQS approach, which applies the harmonic balance method on the complete coupled system, is derived in Section 6.2. The QS approach motivated from the circuit-level point of view is introduced in Section 6.3. The method is also derived mathematically and its quasi-static property is identified. In Section 6.4, the derivation of the modified Volterra series model [44] is reviewed and its implementation details in conjunction with the harmonic balance analysis of CODECS are described. The qualitative comparison between these three approaches is summarized in Section 6.5. The implementation details of QS and MVS approaches are presented in Section 6.6. Finally this chapter is summarized in Section 6.7.

6.2 Non-Quasi-Static (NQS) Implementation

The complete system of equations with both circuit-level and device-level unknowns is

$$f_d = d(w,v) + \frac{d}{dt}(aw) = 0 \tag{6.1}$$

$$f_c = I(w,v) + \frac{d}{dt}D(w,v) + i(v) + \frac{d}{dt}q(v) + s = 0 \tag{6.2}$$

where Equation (6.1) represents the device-level equations after space discretization. w and v are the device-level and circuit-level unknowns, respectively. $d(w,v)$ is the static term of the device-level equations. Comparing with Equations (3.19a-c), $w=[\psi\ n\ p]'$, $d=[F_\psi\ F_n\ F_p]'$, and a is a constant vector whose elements take values +1, -1, or 0 depending on the hole continuity equation, electron continuity equation, or Poission's equation, respectively. In Equation (6.1) the dependence on the terminal voltages v is explicitly written. Equation (6.2) represents the circuit-level equations. I and dD/dt are the conduction and displacement terms of the terminal currents from numerical devices described by Equation (6.1). i is the current through nonreactive elements described by compact models and q is the contribution from compact models for the reactive elements.

Using the harmonic balance method, d/dt can be represented by $T=\Gamma^{-1}\Omega\Gamma$ as in Equation (5.32). Solving the complete set of nonlinear equations using Newton's method results in the following iteration equation

$$\left(\begin{bmatrix} J_w & J_v \\ G_w^I & G_v^I+G \end{bmatrix}+T\begin{bmatrix} a & 0 \\ C_w^D & C_v^D+C \end{bmatrix}\right)\begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix}=-\begin{bmatrix} f_d \\ f_c \end{bmatrix} \tag{6.3}$$

where: $J_w=\frac{\partial d}{\partial w}$, $J_v=\frac{\partial d}{\partial v}$, $G_w^I=\frac{\partial I}{\partial w}$, $G_v^I=\frac{\partial I}{\partial v}$, $C_w^D=\frac{\partial D}{\partial w}$, $C_v^D=\frac{\partial D}{\partial v}$, $G=\frac{\partial i}{\partial v}$, $C=\frac{\partial q}{\partial v}$.

Let: $A_{dd}=J_w+Ta$

$A_{dc}=J_v$

$$A_{cd} = G_w^I + TC_w^D$$

$$A_{cc} = G_v^I + TC_v^D$$

Equation (6.3) becomes

$$\begin{bmatrix} A_{dd} & A_{dc} \\ A_{cd} & A_{cc} + (G + TC) \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix} = -\begin{bmatrix} f_d \\ f_c \end{bmatrix} \tag{6.4}$$

From the first equation, Δw can be solved as

$$\Delta w = A_{dd}^{\ -1}(-f_d - A_{dc}\Delta v) \tag{6.5}$$

Substituting into the second equation, we have

$$\left(G_{eq} + (G + TC)\right)\Delta v = -f_c - I_{eq} \tag{6.6}$$

where

$$G_{eq} = A_{cd} A_{dd}^{\ -1}(-A_{dc}) + A_{cc} \tag{6.7}$$

$$I_{eq} = A_{cd} A_{dd}^{\ -1}(-f_d) \tag{6.8}$$

G_{eq} and $I_{eq} + I(w, v) + TD(w, v)$ are the contributions of the numerical device to the circuit-level Jacobian matrix and RHS. The solution procedure for the non-quasi-static implementation approach in CODECS is shown in Figure 6.1, where G_{eq} and I_{eq} are computed from A_{dd}, A_{dc}, A_{cd}, and A_{cc} as in Equations (6.7) and (6.8). We can see that the harmonic balance method is implemented at the device level and interfaces with the circuit-level harmonic balance equations by G_{eq} and $I_{eq} + I(w, v) + TD(w, v)$. In this way, the numerical device model can be treated as a conventional model and the non-quasi-static approach can also be implemented within the existing framework of CODECS. It should be noted that, G_{eq}, the device contribution to the circuit-level Jacobian matrix, is a full matrix. This is because not only the displacement current $TD(w, v)$ but also the conduction current $I(w,v)$ are functions of the terminal voltages at all sampled time points due to the internal dynamics of the device.

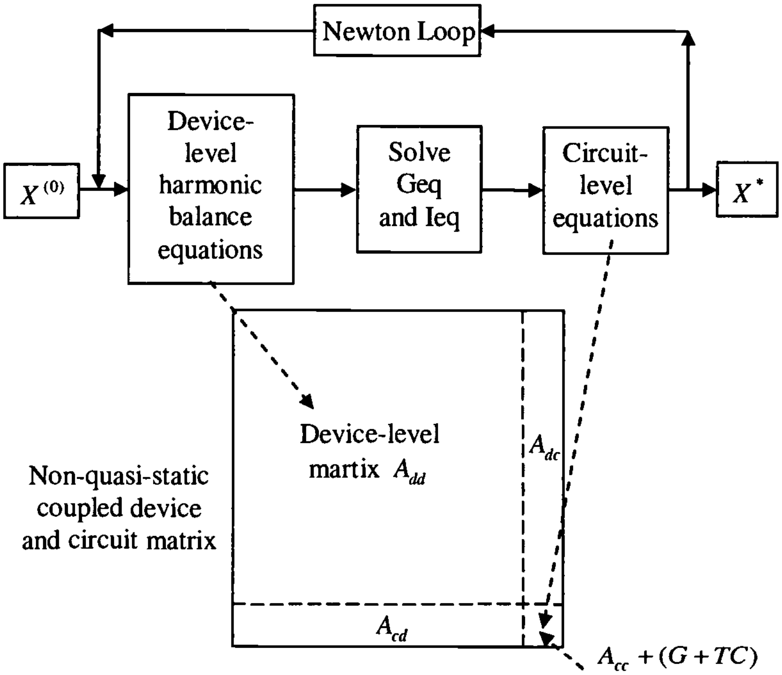

Newton Loop
$X^{(0)}$
Device-level harmonic balance equations
Solve Geq and Ieq
Circuit-level equations
X^*
Device-level martix A_{dd}
A_{dc}
A_{cd}
$A_{cc}+(G+TC)$
Non-quasi-static coupled device and circuit matrix

Figure 6.1: Solution procedure for the non-quasi-static approach in CODECS.

For the non-quasi-static approach, the internal dynamics of the semiconductor devices are taken into account and the results are as accurate as the numerical models themselves. Since the harmonic balance method is implemented at the device level, the Fourier coefficients of a large number of device-level unknowns have to be solved. For a numerical device with L mesh points, the number of device-level unknowns is approximately $3L$ because each mesh points has three variables, the electrostatic potential and the electron and hole concentrations. If M frequencies are chosen, the total number of Fourier coefficients to be solved by the harmonic balance method is $3L\times(2M+1)$. With fine meshed numerical devices L can be very large. This results in a large device-level system matrix $A_{dd}=J_w+Ta$. As mentioned at the end of Section 5.7, factorization of such a large dense harmonic balance Jacobian matrix by direct methods is prohibitively expensive and the simulation is extremely time and memory consuming. Krylov subspace methods have to be used to make the simulation feasible. Furthermore, since the harmonic balance method is implemented at the device level, this approach requires extensive modifications to the internals of the device simulator.

6.3 Quasi-Static (QS) Implementation

From the circuit-level point of view, the numerical device can be treated as a combination of nonlinear conductances and capacitances as discussed in Section 3.4. For the quasi-static implementation, the harmonic balance method is implemented only at the circuit level. For each sampled time point, using the device-level simulator, the numerical device is biased to the corresponding terminal voltages and the low frequency terminal conductances and capacitances are calculated by an AC analysis. These parameters for different time points are loaded into the Jacobian matrix and the right-hand side to assemble the circuit-level harmonic balance equations. Obviously this approach is consistent with the framework of CODECS and can be implemented easily. The solution procedure for the quasi-static implementation in CODECS is illustrated in Figure 6.2. In the figure, $X^{(0)}$ is the frequency-domain initial guess of the circuit-level waveforms and is set to the value at the DC operating point and X^* is the final solution when the convergence criterion is met. With M chosen frequencies, $2M+1$ low-frequency AC analyses for numerical devices at $2M+1$ sampled time points are required to obtain the linearized conductance g, linearized capacitance c, current i, and charge q. g and c are stamped into the circuit-level system matrix G and C to generate the harmonic balance Jacobian. i and q are used to load the right hand side. Here the contribution of numerical device to matrices G and C are block-diagonal with each diagonal block corresponding to each sampled time point. In contrast, for the non-quasi-static approach the device-level contribution to the circuit-level Jacobian matrix, G_{eq}, is a full matrix as discussed in Section 6.2. Here the semiconductor devices are represented as nonlinear terminal conductances and capacitances which are instantaneous functions of the terminal voltages, hence, this is a quasi-static approach.

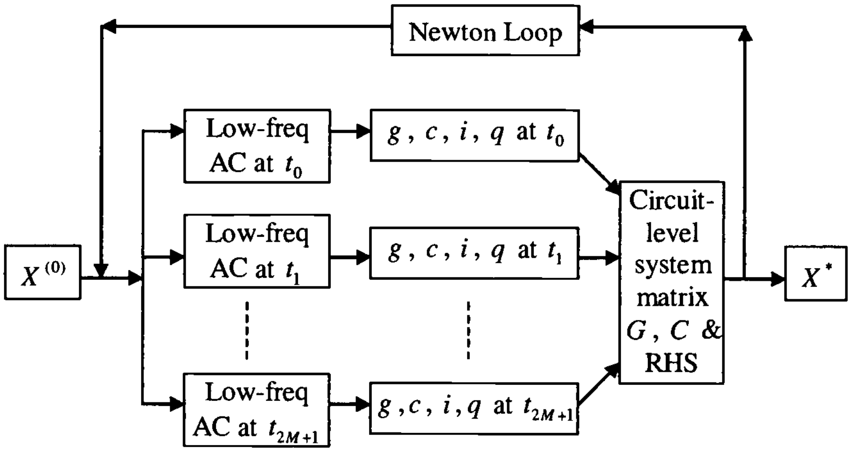

Newton Loop
$X^{(0)}$
Low-freq AC at t_0
Low-freq AC at t_1
Low-freq AC at t_{2M+1}
g, c, i, q at t_0
g, c, i, q at t_1
g,c, i,q at t_{2M+1}
Circuit-level system matrix G, C & RHS
X^*

Figure 6.2: Solution procedure for a quasi-static implementation of the harmonic balance method in CODECS. Low-frequency AC analyses of the device are carried out at different sampled time points to construct the circuit-level matrix and RHS.

The quasi-static approach can also be explained mathematically by using Equations (6.1) and (6.2). Since the device internal dynamics are ignored, Equation (6.1) becomes

$$f_d = d(w, v) = 0 \tag{6.9}$$

The terminal current of the device is given by

$$i_d = I(w,v) + \frac{d}{dt} D(w,v) \tag{6.10}$$

From Equation (6.9), the internal state w can be expressed as a static function of v

$$w = \tilde{d}(v) \tag{6.11}$$

Substituting into Equation (6.10), we have

$$i_d = I(\tilde{d}(v),v) + \frac{d}{dt} D(\tilde{d}(v),v) = I_{qs}(v) + \frac{d}{dt} D_{qs}(v) \tag{6.12}$$

Where $I_{qs}(\cdot) = I(\tilde{d}(\cdot),\cdot)$ is the static current of the device and $D_{qs}(\cdot) = D(\tilde{d}(\cdot),\cdot)$ is the static charge. Obviously, the nonlinear current I_{qs} and nonlinear charge D_{qs} are functions of the instantaneous terminal voltage v and their linearization can be obtained from an AC analysis of the numerical devices at a low frequency.

In the quasi-static approach, the circuit-level harmonic balance method combined with a device simulator uses only circuit-level waveforms as unknowns. The size of the system matrix is very small and the burden of solving a large system matrix from the non-quasi-static method in Section 6.1 is eliminated. Due to the small system matrix at the circuit level, Newton's iteration doesn't take much time even when the direct method is used to factorize the harmonic balance Jacobian. The overall simulation is efficient and usually the $2M+1$ AC analyses that are applied to the numerical devices take a significant portion of the simulation time. Any device simulator that can provide the terminal capacitances and conductances for numerical models can be used in this approach. Thus, application specific device simulators can be included in this harmonic balance analysis framework without extensive modifications as required in the non-quasi-static approach. Since the terminal parameters are evaluated based on a static solution of devices, this is a quasi-static approach. Hence, the results may not be accurate when the input frequency is comparable with the device transition frequency.

The limitations of the quasi-static approach are outlined in the following two observations. First, the behavior of the internal states of the numerical devices, such as the electrostatic potential and the carrier concentrations, is approximated by the static solution at the sampled time points. This static solution is then used as the DC operating point to compute the terminal characteristics by an AC analysis. This approximation is only valid for sufficiently slow variations in the boundary conditions (the terminal voltages). Secondly, the terminal conductances and capacitances of the numerical devices are frequency dependent [25], which requires knowledge of the frequency to calculate these correctly. The quasi-static approach uses the low-frequency terminal conductances and capacitances.

To obtain an accurate steady-state solution for high-frequency applications, the device internal dynamics have to be considered. The non-quasi-static approach in Section 6.1 addresses this problem by solving a large number of device-level dynamic equations using the harmonic balance method. This approach is accurate for any frequency but time and memory consuming. In contrast, the accuracy of the quasi-static approach is limited by the device transition frequency but the simulation is very efficient. An approach that uses the harmonic balance method only at the circuit level (as in the quasi-static approach) and accounts for the device internal dynamics as well is desirable. Such a method will be more efficient than the non-quasi-static approach and more accurate than the quasi-static approach.

Recall that the terminal conductances and capacitances are frequency dependent and the quasi-static approach uses the low frequency values. A modified-Volterra-series approach makes use of the frequency dependence to include the device internal dynamics and is discussed in the next section.

6.4 Modified Volterra Series (MVS) Implementation

In this section, the derivation of the modified Volterra series method [44, 45] is reviewed and its implementation details in conjunction with harmonic balance analysis of CODECS are described.

6.4.1 Modified Volterra Series Method

A nonlinear dynamic system model based on the modified Volterra series is proposed in [44] and used to simulate electronic devices with a $50\,\Omega$ linear load by the harmonic balance method.

The output current $i(t)$ of a nonlinear dynamic device with an input voltage $v(t)$ can be modeled by the well-known Volterra series [46]

$$i(t) = y_0 + \sum_{n=1}^{\infty} y_n(t) \tag{6.13}$$

where y_0 is a constant independent of the input voltage and

$$y_n(t) = \int \cdots \int_{-\infty}^{+\infty} \cdots \int h_n(\tau_1, \tau_2, \cdots, \tau_n) \prod_{i=1}^{n} [v(t-\tau_i) d\tau_i] \tag{6.14}$$

$h_n(\tau_1, \tau_2, \cdots, \tau_n)$ is the nth-order time-domain Volterra kernel of the system. Practically a dynamic device has finite memory time such that $-T_A \le \tau \le +T_B$. Then Equation (6.14) becomes

$$y_n(t) = \int \cdots \int_{-T_A}^{+T_B} \cdots \int h_n(\tau_1, \tau_2, \cdots, \tau_n) \prod_{i=1}^{n} [v(t-\tau_i) d\tau_i] \tag{6.15}$$

By introducing the difference $v(t-\tau)-v(t)$ which represents the difference of the signal $v(t-\tau)$ with respect to $v(t)$, Equation (6.13) is transformed to the modified Volterra series

$$i(t) = z_0(v(t)) + \sum_{n=1}^{\infty} z_n(t) \tag{6.16}$$

where $z_0(v(t))$ is the response without the system dynamics, i.e., the static current of the device, and

$$z_n(t) = \int \cdots {}_{-T_A}^{+T_B} \cdots \int g_n\{v(t), \tau_1, \tau_2, \cdots, \tau_n\} \prod_{i=1}^{n} [(v(t-\tau_i) - v(t))d\tau_i] \tag{6.17}$$

represents the nth-order dynamic current. $g\{\cdot\}$is the nth-order modified Volterra kernel and is a nonlinear function of $v(t)$. When the difference $v(t-\tau)-v(t)$ is small, a truncation to the first order dynamics is a reasonable approximation. So we have

$$i(t) \cong z_0(v(t)) + \int_{-T_A}^{+T_B} \quad g_1\{v(t), \tau_1\}[v(t-\tau_1) - v(t)]d\tau_1 \tag{6.18}$$

In the harmonic balance method, the input voltage $v(t)$ can be expressed as

$$v(t) = \sum_{m=-M}^{+M} V_m e^{j2\pi f_m t} \tag{6.19}$$

where M is the number of frequencies chosen for the harmonic balance analysis and V_m is the Fourier coefficient of the mth frequency. With this input voltage, Equation (6.18) can be expressed in the form

$$i \cong z_0(v) + \sum_{m=-M}^{+M} D\{v, f_m\} V_m e^{j2\pi f_m t} \tag{6.20}$$

where $D\{v, f_m\} = Y(v, f_m) - Y(v,0)$ (6.21)

is the difference between the terminal admittances at frequency f_m and DC when the device is biased at a terminal voltage v [44, 45]. This can be obtained from a small-signal analysis at the device simulator level. Equations (6.20) and (6.21) are the modified Volterra series representation for devices implemented in the harmonic balance analysis of CODECS.

As stated earlier, Equation (6.18) is obtained by truncating Equation (6.16) to the first order of the device dynamics based on the assumption that the difference $v(t-\tau)-v(t)$ is small. The region of validity for this assumption is one of the limitations of the MVS approach. A small difference can be achieved by a relatively slow change of $v(t)$ in the finite memory interval $-T_A \leq \tau \leq +T_B$. For a given system (i.e., given T_A and T_B), a slow change of $v(t)$ means a low frequency, f_v, and/or small amplitude, A_v. For a sinusoidal input voltage, [45] shows that the truncation error of the MVS model is bounded by $k(f_v A_v)^2$ (where k is a constant). This suggests that the constant error loci have an upper limit defined by a hyperbolic function in the A_v/f_v space. The constant error locus of the MVS model is qualitatively illustrated in Figure 6.3. Form the figure we can see that the MVS model can obtain accurate results for high frequency input signals as long as the magnitudes are relatively small.

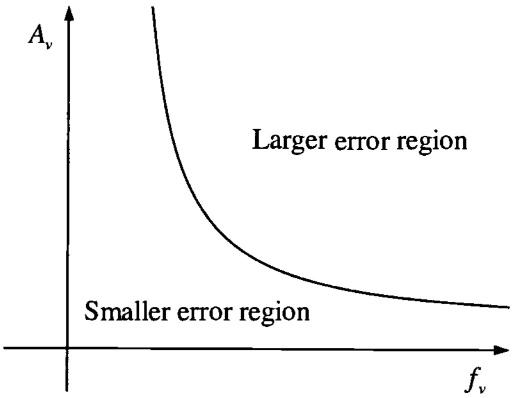


Figure 6.3: Constant truncation error locus of the modified Volterra series model.

Equation (6.20) shows that in the MVS model the device dynamic current is obtained by multiplying the harmonic components of the terminal voltage with the terminal admittance differences $D\{\cdot\}$ at the corresponding frequencies. It can also be shown that if $Y(v,f_m)$ are approximated by using the low-frequency terminal parameter, then the MVS model will degenerate to the quasi-static model described by Equation (6.12). In this case, $D\{\cdot\}$ is given by

$$D\{v,f_m\} = [g_{LF}(v) + j2\pi f_m c_{LF}(v)] - Y(v,0) \tag{6.22}$$

where $g_{LF}(v)$ and $c_{LF}(v)$ are the terminal conductance and capacitance at a low frequency, respectively. Since $g_{LF}(v) = Y(v,0)$, Equation (6.22) becomes

$$D\{v,f_m\} = j2\pi f_m c_{LF}(v) \tag{6.23}$$

Substituting into Equation (6.23) and considering Equation (6.20), we have

$$i = z_0(v) + c_{LF}(v)\sum_{m=-M}^{+M} j2\pi f_m V_m e^{j2\pi f_m t} = z_0(v) + c_{LF}(v)\frac{dv(t)}{dt} \tag{6.24}$$

Compared with Equation (6.12), Equation (6.24) represents the same quasi-static model because $c_{LF}(v) = \frac{d}{dv} D_{qs}(v)$.

6.4.2 Implementation in Harmonic Balance Analysis

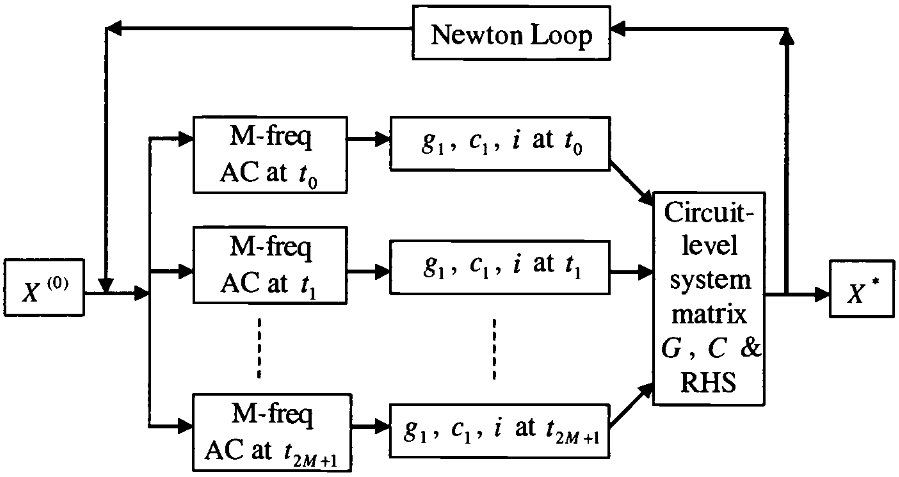
The terminal current of the MVS model can be readily calculated from Equation (6.20), but the linearized terminal parameter for stamping into the circuit-level harmonic balance matrix needs further derivation. Equation (6.20) shows that the terminal current is a function of voltage in both time and frequency domains. $D\{\cdot\}$ is also a function of the terminal voltage. Taking the derivative with the terminal voltage, $\frac{di}{dv}$ can be expressed as

$$\frac{di}{dv} = \frac{dz_0}{dv} + \sum_{m=-M}^{+M} [\frac{dD\{v, f_m\}}{dv} V_m + D\{v, f_m\} \frac{dV_m}{dv}] e^{j2\pi f_m t} \qquad (6.25)$$

$\frac{dz_0}{dv} = Y(0)$ is the DC terminal conductance and is available from an AC analysis. In the harmonic balance analysis, $\frac{dV_m}{dv}$ is a column of the DFT transform matrix Γ. $\frac{dD\{v, f_m\}}{dv}$ involves the second-order derivative of the terminal current. Usually the device simulator only provides the terminal admittances which are the first-order derivatives. The second-order derivatives have to be calculated. This can be done numerically from tables of the first-order derivatives as a function of the terminal voltages. However, the simulation complexity increases and the numerical stability and convergence are degraded. For this reason, a more efficient calculation procedure needs to be developed.

Since the linearized parameters of the nonlinear circuit elements stamped in the Jacobian matrix only affect the search direction of Newton's method, and not the final solution, approximations can be made. Usually the magnitude at the fundamental frequency is dominant in a spectrum with one tone. Thus a reasonable approximation is the use of the linearized terminal admittance at the fundamental frequency. With this approximation, only an AC analysis at the fundamental frequency is required to obtain the admittance. Therefore, the complex calculation from Equation (6.25) is eliminated along with any associated numerical stability and convergence problems. Simulations using this approximation show that the steady-state results are accurate and the simulation is efficient. For our MVS harmonic balance implementation, the terminal admittances at the fundamental frequency are used to stamp the system matrix while the right-hand-side is stamped by the terminal currents from Equation (6.20).

The solution procedure for the MVS implementation in CODECS is illustrated in Figure 6.4. In this figure, g_1 and c_1 are the linearized conductance and capacitance at the fundamental frequency. i is the total terminal current including the dynamic part calculated from Equation (6.20). For the quasi-static approach in Figure 6.2, i is only the dc current and charge q has to be used to account for the dynamic current. From Figure 6.4, M AC analyses at the M frequencies used for the harmonic balance analysis are carried out at each sampled time point. In contrast, only one low frequency AC analysis is required at each sampled time point in the quasi-static implementation. Therefore, the MVS implementation requires more time per Newton iteration compared with the quasi-static approach.



Newton Loop
$X^{(0)}$
M-freq AC at t_0
M-freq AC at t_1
M-freq AC at t_{2M+1}
g_1, c_1, i at t_0
g_1, c_1, i at t_1
g_1, c_1, i at t_{2M+1}
Circuit-level system matrix G, C & RHS
X^*

Figure 6.4: Solution procedure for a modified-Volterra-series approach of the harmonic balance method in CODECS. AC analyses of the numerical device at M chosen frequencies are carried out at different sampled time points to construct the circuit-level matrix and RHS.

6.5 Qualitative Comparison

Equation (6.21) shows that the MVS approach also computes the terminal parameters based on the static solution of the internal states of devices as in the QS approach. For this reason, the large-signal internal dynamics are not considered. However, in the MVS approach the terminal admittance is obtained by AC analyses at each chosen frequency f_m as in Equation (6.21). These AC analyses take into account the device small-signal internal dynamics. Therefore, one can say that the

MVS approach "partly" considers the internal dynamics of numerical devices. In contrast, for the QS approach only the low frequency terminal admittances are used and no small-signal internal dynamics are considered. Therefore, an improvement in accuracy is expected for the MVS approach compared with the QS approach, especially for high frequency and small amplitude input signals. For the NQS approach, since the harmonic balance method is also applied to the device-level equations as in Section 6.2, all the device internal dynamics are considered. From a device modeling point of view, the MVS approach is in between the NQS and QS approaches. The difference between these three implementation approaches is based on the handling of the numerical device dynamics as summarized in Table 6.1. The device terminal dynamics represent the displacement terminal currents from the numerical devices.

	Device terminal dynamics	Large-signal internal dynamics	Small-signal internal dynamics
NQS method	Yes	Yes	Yes
QS method	Yes	No	No
MVS method	Yes	No	Yes

Table 6.1: The differences between three approaches for harmonic balance analysis in a coupled device and circuit simulator.

Based on the discussion of the previous three sections, the performance comparisons of these three implementation approaches for harmonic balance analysis in the coupled device and circuit simulation are summarized in Table 6.2. These conclusions will be verified by examples and results in Section 7.2.

	NQS method	QS method	MVS method
HB level	Circuit & device	Circuit	Circuit
# Unknowns	Large	Small	Small
Memory used	Large	Small	Small
Simulation time	Slow	Fast	Fair
Result accuracy	Excellent	Limited by f_T	Improved for small amplitude signals
Modification to device simulator	Extensive	None	None

Table 6.2: Summary of the performance comparison of NQS, QS, and MVS approaches for harmonic balance method in coupled device and circuit simulation.

6.6 Implementation Details of QS and MVS Approaches

As shown in Equation (5.36), the terminal charge q due to the nonlinear capacitors from the previous iteration is required to load the right hand side for the present iteration. For the QS approach, this is the static charge given by $D_{qs}(v)$ in Equation (6.12). In CODECS, although the terminal capacitance can be obtained by an AC analysis, the corresponding static charge at a DC bias is not provided by the device simulator. A reformulation solves this problem as shown below. From Equation (5.35) we have the current of the nonlinear capacitor in the time domain as:

$$i(v^{(j)}) = \Gamma^{-1}I^{(j)} = \Gamma^{-1}\Omega\Gamma q(v^{(j)}) \qquad (6.26)$$

j indicates the solution at the jth Newton iteration. Since current is the time derivative of charge, we can also express the current of the nonlinear capacitor as:

$$i(v^{(j)}) = \left.\frac{\partial q}{\partial t}\right|_{v^{(j)}} = \left.\frac{\partial q}{\partial v}\right|_{v^{(j)}} \left.\frac{\partial v}{\partial t}\right|_{v^{(j)}} = C(v^{(j)})\left.\frac{\partial v}{\partial t}\right|_{v^{(j)}} \tag{6.27}$$

Similar to Equation (6.26), $\frac{\partial v}{\partial t}$ can be expressed as $\Gamma^{-1}\Omega\Gamma v$ in the harmonic balance method and Equation (6.27) becomes:

$$i(v^{(j)}) = C(v^{(j)})\Gamma^{-1}\Omega\Gamma v^{(j)} \tag{6.28}$$

From Equations (6.26) and (6.28) we can solve for $\Omega\Gamma q(v^{(j)})$:

$$\Omega\Gamma q(v^{(j)}) = \Gamma C(v^{(j)})\Gamma^{-1}\Omega\Gamma v^{(j)} \tag{6.29}$$

Therefore, Equation (5.36) can be rewritten as:

$$I^{(j+1)} = \Omega\Gamma C(v^{(j)})\Gamma^{-1}V^{(j+1)} + \Gamma C(v^{(j)})\Gamma^{-1}\Omega\Gamma v^{(j)} - \Omega\Gamma C(v^{(j)})v^{(j)} \tag{6.30}$$

The terminal charge of the nonlinear capacitors is, therefore, not required to assemble the harmonic balance equations.

During the harmonic balance simulation of circuits with numerical devices using the QS and MVS approaches, it is found that the performance bottleneck is the multiple AC analyses for numerical devices at each Newton iteration. These AC analyses take most of the simulation time. The AC analysis is carried out at a DC operating point. Therefore, before the calculation of the linearized parameters, a DC analysis has to be performed. Originally in CODECS, for each DC analysis of numerical devices, first the equilibrium point where the terminal voltages are zero is established and then devices are biased up to the desired terminal voltages. Usually the terminal voltages are so large that it takes many small voltage steps and many iterations in each voltage step to bias the device voltages from their equilibrium points. This voltage stepping at each sampled time point makes the DC analysis and further AC analyses time-consuming. Before stepping the device bias voltages for the next time point, the numerical devices have been biased to the voltages at the current time point, which can be made use of for the next time point. So an alternative biasing scheme similar to the one in Section 4.6.1 is to bias the numerical devices using the voltage difference between these two time points, instead of biasing devices to the large terminal voltages from the equilibrium point. Usually the voltage difference between two neighboring time points is small and the device simulator converges rapidly. Using this new biasing scheme, each numerical device is biased from the equilibrium point only once at the first sampled time point of the first iteration and then the voltages of the numerical device are stepped by the small voltage differences, thereafter. The new biasing scheme is illustrated in Figure 6.5. A significantly improvement in the performance of the harmonic balance analysis is obtained. Simulation time comparison between the two biasing schemes is provided in Section 7.2.2 for several example circuits. Results verify the efficiency improvement.

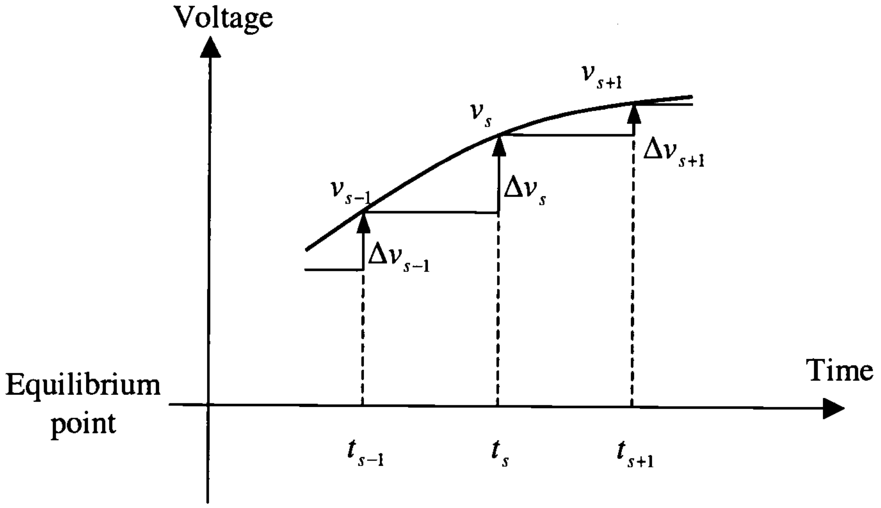


Figure 6.5: Numerical devices need to be stepped by only the small voltage difference between the two consecutive time points Δv instead of the large voltage difference from the equilibrium point v.

6.7 Summary

In this chapter, approaches for implementing the harmonic balance method in the coupled device and circuit simulator CODECS are explored. Three different approaches were introduced. These include the quasi-static (QS), the non-quasi-static (NQS), and the modified Volterra series (MVS) approaches. The architecture of CODECS is exploited and all the implementations follow the existing coupling simulation framework.

In the NQS approach, the harmonic balance method is implemented at both the device and circuit levels. All the internal dynamics of semiconductor devices are considered and the results are as accurate as the numerical models themselves. Also because of the device-level implementation, a large system of equations has to be solved and the simulation is time and memory consuming. Furthermore, this approach requires extensive modification of the device simulator. In the QS approach the harmonic balance method is implemented only at the circuit level. The terminal parameters of the numerical devices obtained by a low frequency AC analysis are loaded into the circuit-level harmonic balance equations. Since the internal dynamics of the semiconductor devices are not considered, this is a quasi-static analysis and the accuracy is limited by the transition frequency of devices. Due to the circuit-level implementation, the system of harmonic balance equations is small and the simulation is very efficient. Furthermore, no modifications are required for the device simulator. The MVS approach improves the accuracy of the QS approach for high frequency and small amplitude signals. The improvement is achieved at the price of a longer simulation time.

The complex computation of the linearized terminal parameter in the MVS approach is eliminated by using a reasonable approximation. In the QS approach, the static charge is not available from the device simulator and its calculation is avoided by an equation reformulation. A new biasing scheme for numerical devices is proposed that significantly improves the performance of the QS and MVS approaches.

Chapter 7

EXAMPLES AND RESULTS

7.1 Time-Domain Shooting Method

The time-domain shooting method implemented in CODECS is used to obtain the periodic steady state solution for several example circuits. The results from our implementation are verified with the results obtained by conventional transient simulations. Also the computational performance of these two methods is compared.

7.1.1 Examples

The first example is a X3 frequency multiplier circuit of Figure B.5 of [29]. A one-dimensional numerical model with 61 mesh points is used for the bipolar transistor in the circuit with the doping profile shown in Figure 7.1. The steady-state solution of this circuit is obtained after 6 periods of transient analysis using the time-domain shooting method. The conventional transient simulation takes 1500 periods to reach the steady state. Newton's algorithm in the shooting method speeds up the convergence of the steady-state simulation significantly. The output voltage waveforms for several periods during the analysis are plotted in Figure 7.2. The Newton iteration is carried out at the end of each period to calculate the initial guesses for the next period. The output waveform starts from its DC value and very quickly approaches the steady-state solution as shown in the second, third, and fourth periods. The magnitude of the waveform in the fourth period is larger than the final solution, and Newton's algorithm corrects this by finding a new initial guess for the fifth period. Finally the analysis converges in the sixth period where the initial value at $t = 0$ is equal to the final value at $t = 0.2\mu s$. Also, in the steady state the output voltage waveform has three cycles in each original period and thus the "X3 frequency multiplier" is realized.

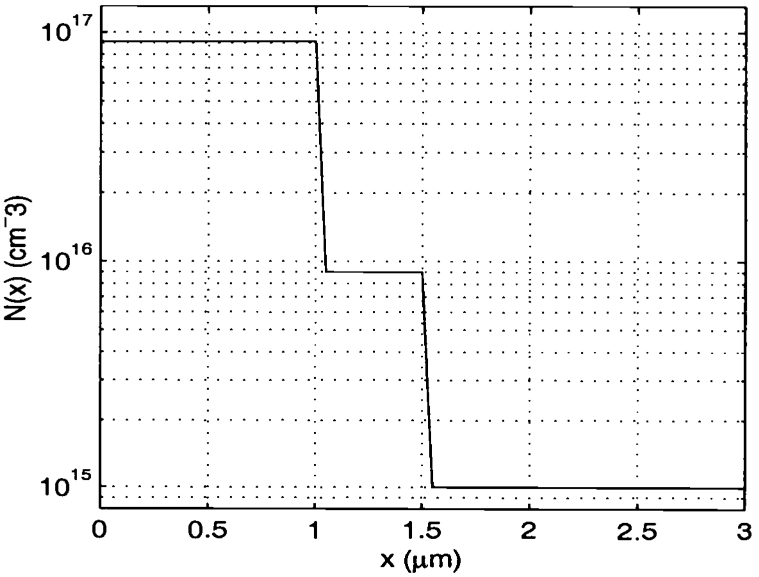


Figure 7.1: Doping profile and dimension of the 1D numerical NPN bipolar transistor used in the X3 frequency multiplier.

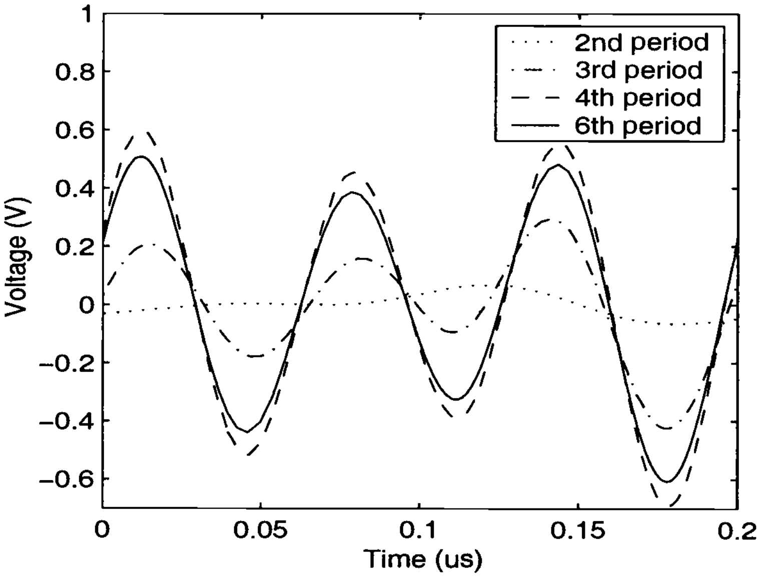


Figure 7.2: Output voltage waveforms for four periods during the steady-state simulation of the X3 frequency multiplier circuit using the time-domain shooting method.

The second example is the DC power supply circuit of Figure B.1 of [29]. The diode in this circuit is modeled by a one-dimensional numerical model with 191 mesh points and the doping profile is plotted in Figure 7.3. The steady state is obtained after 6 periods of the transient analysis using the time-domain shooting method. The normalized harmonics of the steady-state voltage waveform at the cathode of the diode are obtained by applying a FFT on the time-domain waveform and are plotted in Figure 7.4. From this figure, the magnitude of the harmonics drops slowly due to the strong nonlinearity of the diode. If the harmonic balance method is used to determine the steady state of this circuit, many harmonics have to be computed to obtain an accurate result. With the time-domain shooting method this circuit can be readily simulated. In each period of the time-domain shooting method, circuits are analyzed by a transient simulation. Generally, the transient simulation handles strongly nonlinear circuits efficiently. Thus the time-domain shooting method is robust for accurate steady-state simulation of strongly nonlinear circuits [15].

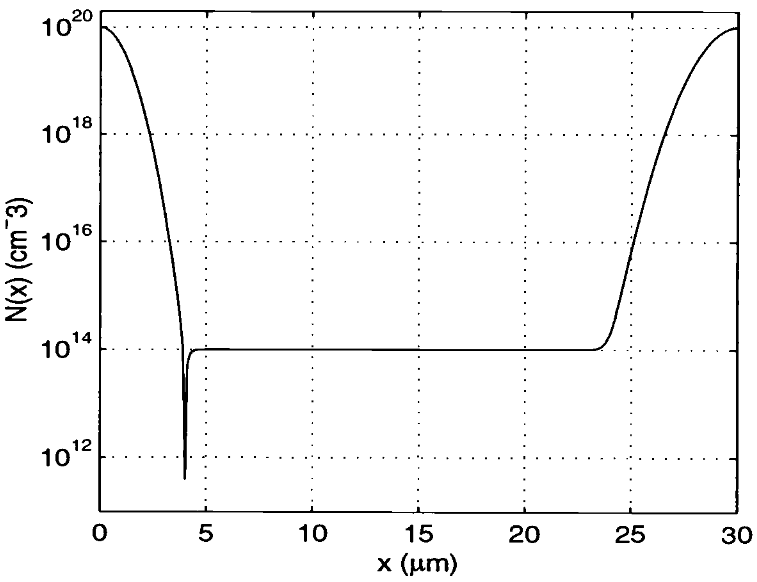


Figure 7.3: Doping profile and dimension of the 1D numerical diode used in the DC power supply circuit.

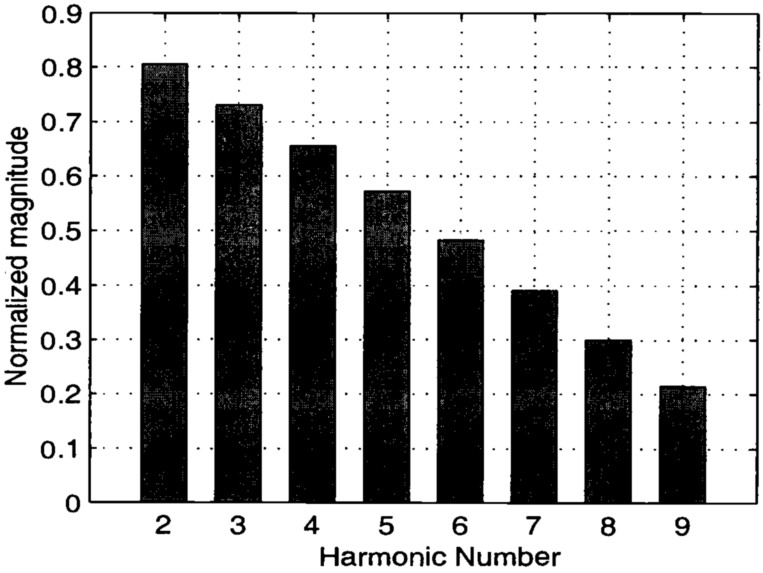


Figure 7.4: Normalized harmonics of the steady-state voltage waveform at the cathode of the diode for the DC power supply circuit using the time-domain shooting method followed by a FFT.

As an autonomous system, a typical BJT Colpitts oscillator is chosen from [47]. The same one-dimensional numerical BJT as in Figure 7.1 is used here. The steady-state solution of this circuit is obtained with 18 periods of the transient simulation using the Newton shooting algorithm. Among them, sensitivity computation is carried out in 13 periods and the Newton iteration as in Equation (4.5) is performed 10 times. At the end of the 6th and 12th periods, the sensitivity matrix has been calculated but the Newton iteration is not performed. This is because the error is larger than the acceptable threshold as discussed in the heuristics of Section 4.4.3. The steady-state solution is verified with a conventional transient simulation.

To demonstrate the effect of the numerical model on the steady-state solution, a high frequency Colpitts oscillator (Figure B.8 of [29]) is used. The one-dimensional numerical BJT model as in Figure 7.1 is used and the corresponding analytical model is extracted from the numerical one. The steady-state solutions of this circuit with the analytical and numerical models are obtained by the time-domain shooting method. The oscillation frequency is 0.8GHz for the analytical model and 0.72GHz for the numerical model. The normalized harmonics of the output waveforms for both models are shown in Figure 7.5. The total harmonic distortions are 7.7% and 13.4% for the analytical and numerical models, respectively. The same simulation has been performed for the Colpitts oscillator from [47]. The oscillation frequency is 60.9 MHz for the analytical model and 61.6MHz for the numerical model. The normalized harmonics of the output waveforms are shown in Figure 7.6. The total harmonic distortions are 1.6% and 0.8% for the analytical and numerical models, respectively. The differences in oscillation frequencies and harmonic distortion are much larger for the high frequency oscillator than the low frequency oscillator.

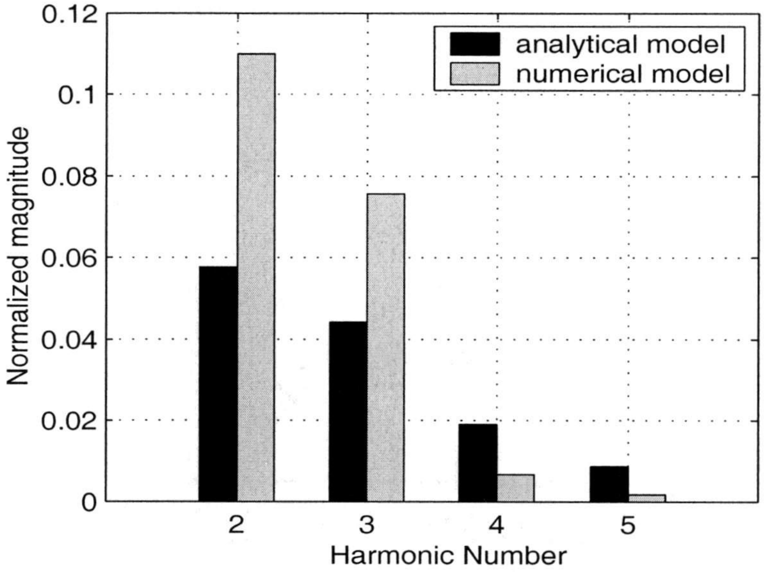


Figure 7.5: Normalized harmonics of the steady-state output waveforms for the high frequency Colpitts oscillator comparing analytical and numerical models.

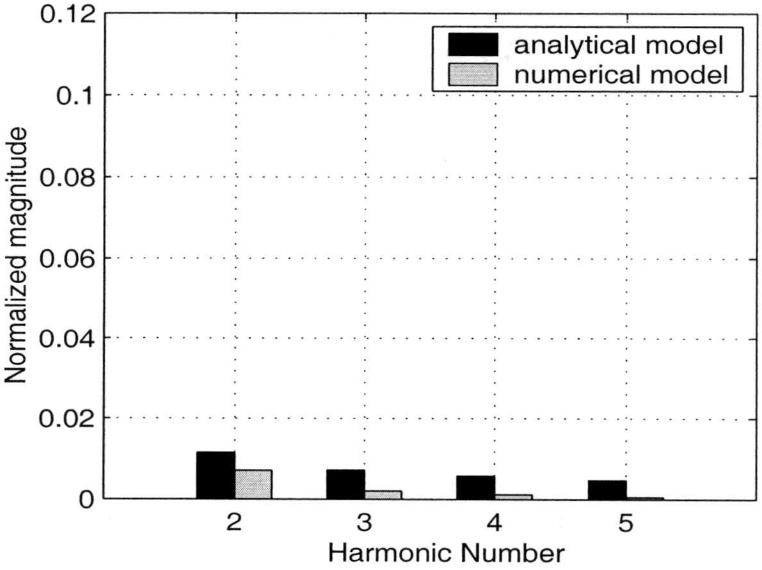


Figure 7.6: Normalized harmonics of the steady-state output waveforms for the BJT Colpitts oscillator from [47] comparing analytical and numerical models.

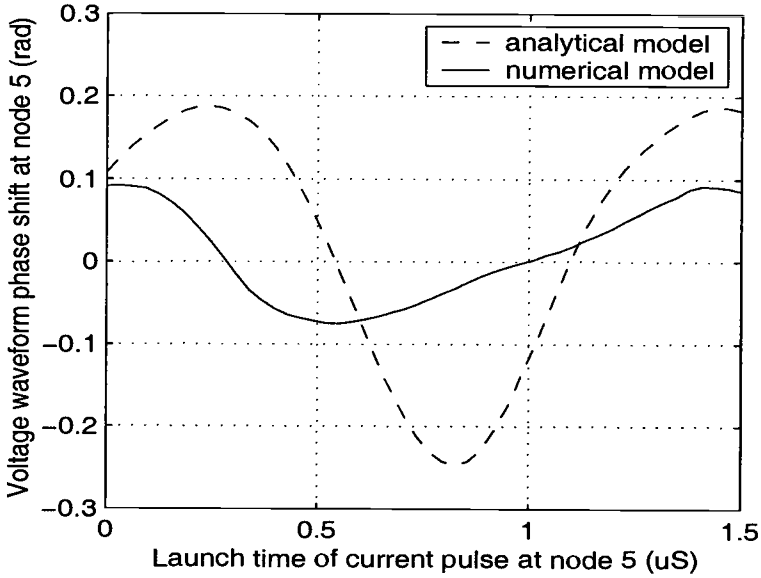


Figure 7.7: Phase shift due to a current impulse at the output node for phase noise analysis of the high frequency oscillator comparing analytical and numerical models.

Finally, for phase noise analysis of the high frequency oscillator [29], the impulse sensitivity functions (ISF) [47] at the output node are simulated with both the analytical and numerical models. The phase shift is plotted in Figure 7.7 and it can be seen that the difference between the two models is significant. These results show that numerical models are essential for accurate simulation of high frequency RF circuits.

7.1.2 Performance Results

To evaluate the performance efficiency of the time-domain shooting method, the conventional transient simulation is used to determine the steady state for several example circuits with one- and two-dimensional numerical models for semiconductor devices. The number of periods required by the conventional transient simulation and the time-domain shooting method are summarized in Table 7.1. It is seen that the time-domain steady-state method is much more efficient than the conventional transient simulation method.

Example Circuits	Conventional transient simulation (# of periods)	Time-domain shooting method (# of periods)
DC supply*	80	6
CB amplifier*	30	4
EC xfrmr osc*	185	25
Freq multiplier*	1500	6
LC EC osc*	22	9
SCP amplifier [14]	182	6
H.F. Colpitts*	20	12
Colpitts [47]	84	18
Demodulator [48]	12000	4

Table 7.1: Performance comparison for conventional transient simulation and the time-domain shooting method. The numbers of periods are directly proportional to the runtime performance. ('*'—circuits from [29]).

In transient simulation of circuits containing numerical devices, the computationally intensive part is the numerical model evaluation. The overhead due to the sensitivity computation required by the time-domain shooting method is negligible, especially when state elimination as described in Section 4.4.1 is implemented. Therefore, the ratio of the numbers of periods required is approximately equal to the ratio of the simulation time required. The time-domain shooting method is much more efficient with high-Q and lightly damped circuits where conventional transient simulation approaches the steady state very slowly. Furthermore, it will result in a significant reduction of simulation time when fine meshed numerical devices are used in circuits.

7.2 Frequency-Domain Harmonic Balance Method

In this section, several example circuits with 1D and 2D numerical models have been chosen to demonstrate accurate and efficient simulation of the quasi-static harmonic balance implementation in CODECS. For these examples the operating frequencies are relatively low such that the device internal dynamics are not significant. The quasi-static characteristics are illustrated by fast turn-on and turn-off transients of a MOS transistor. The quasi-static implementation is compared with the non-quasi-static and MVS implementations in terms of accuracy and simulation performance. Their advantages and disadvantages identified in Chapter 6 are verified. All the following simulations are carried out on SUN Ultra10 workstation.

7.2.1 Results from Quasi-Static Implementation

A single-BJT mixer circuit is taken directly from [14]. The configuration is shown in Figure 7.8. The mixer circuit down-converts a 499.9MHz RF signal to a 100KHz IF, using a 500MHz LO. A resonant RLC circuit having a Q of 100

($R = 15k\Omega$, $C = 66.667nF$, $L = 37.995\mu H$) and a resonant frequency of 100KHz is used. The high-Q filter, filters out the IF-band distortion products and other LO harmonic bands. The voltage source parameters used are $V_{CC} = 10V$, $V_{DC} = 0.7V$, $V_{RF} = 0.05V$ at 499.9MHz, and $V_{LO} = 0.15V$ at 500MHz. A 2D numerical BJT device is used and the doping profile is shown in Figure 7.9. A harmonic balance analysis with 27 frequencies chosen by the box truncation method in Section 5.4 was carried out. The simulation took 11 iterations and the output voltage spectrum is shown in Figure 7.10. A magnified baseband is shown in Figure 7.11. If the time-domain method is used to solve this mixer, at least 50,000 time points need to be simulated in each cycle to obtain the same result. The high Q tank also requires additional cycles to reach the steady state. Therefore, the time-domain method is impractical here while the harmonic balance method can get the result readily.

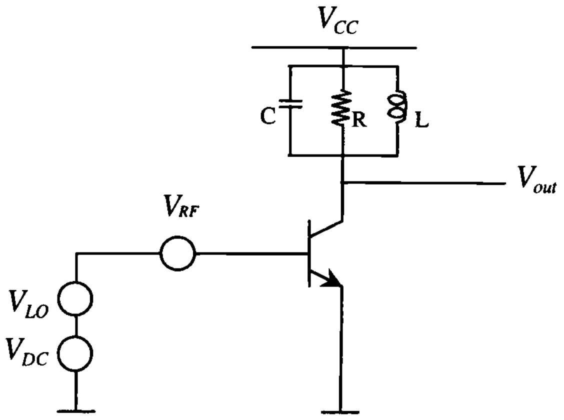


Figure 7.8: A single-BJT mixer circuit.

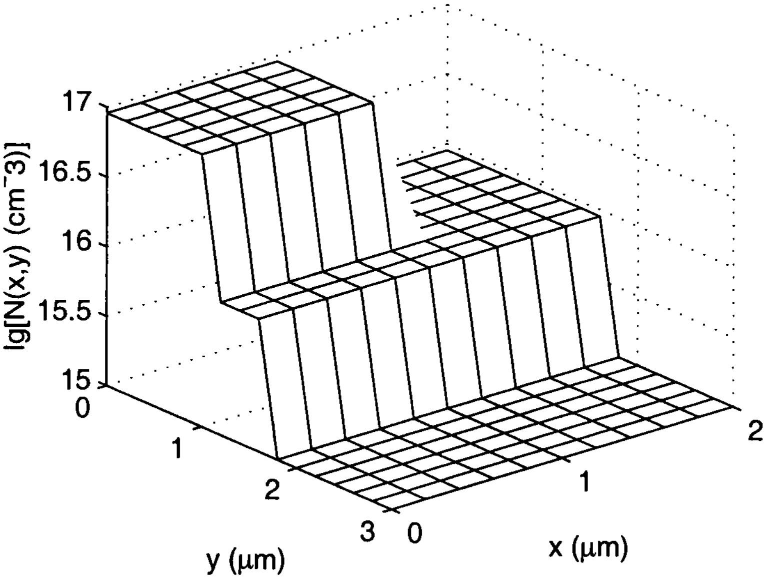


Figure 7.9: Doping profile of a 2D numerical BJT.

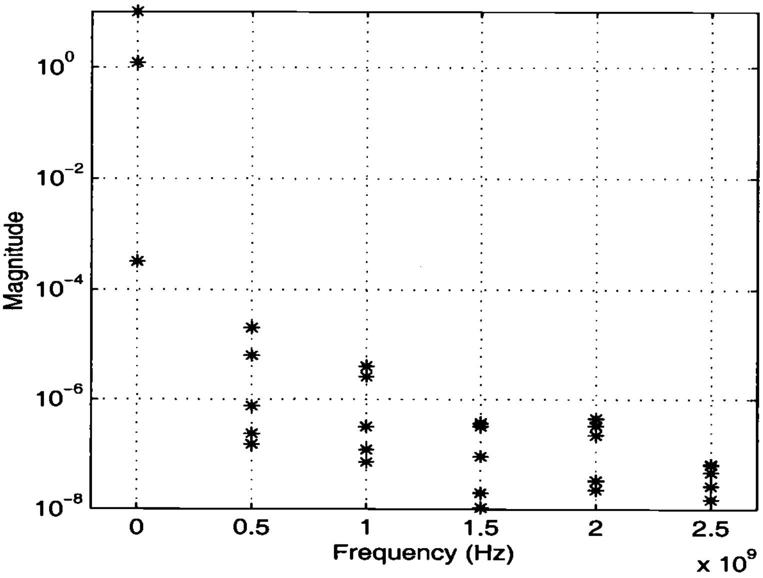


Figure 7.10: Output voltage spectrum for the single-BJT mixer.

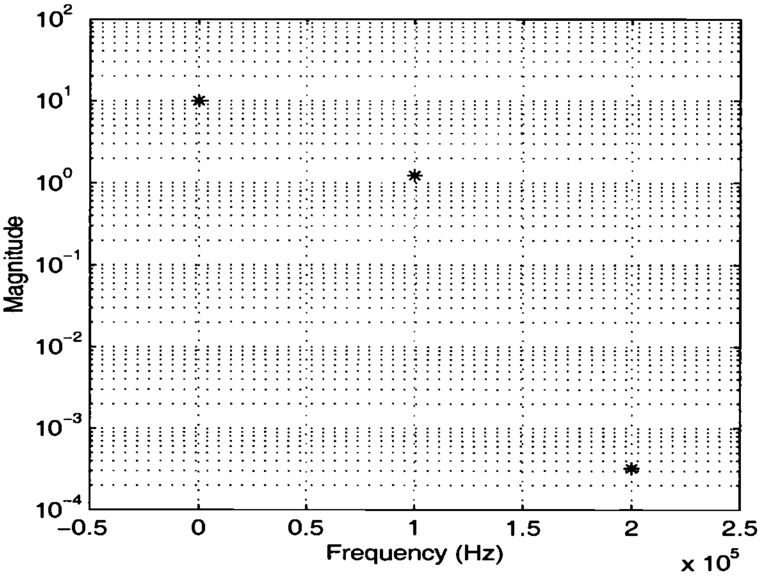


Figure 7.11: Baseband spectrum of the output voltage for the single-BJT mixer.

The second example is a double-balanced mixer circuit from [14] shown in Figure 7.12. The circuit is first simulated using the quasi-static harmonic balance method and then the result is verified with the spectrum obtained from a long transient simulation. The mixer circuit down-converts a 1MHz RF signal to a 0.1MHz IF, using a 1.1MHz LO. A resonant RLC circuit having a Q of 50 (R_C=5KΩ, C=7.958nF, L=318.31μH) and a resonant frequency of 0.1MHz is used. The voltage and current source parameters used are V_{CC}=V_{EE}=10V, I_{EE}=4mA, V_{RF}=0.01V at 1MHz, and V_{LO}=0.1V at 1.1MHz. A 2D numerical BJT model shown in Figure 7.9 was used for transistors Q1 and Q2 while a compact analytical model was used for Q3–Q6. A harmonic balance analysis with 27 frequencies chosen by the box truncation method was carried out. The simulation took 7 iterations and 24 minutes. The output voltage spectrum is shown in Figure 7.13. Also shown in this figure is the spectrum obtained from a long transient simulation which took about 6 hours. In practice, the separation between the IF and LO frequencies is usually much larger than in this example and the time-domain simulation will be impractical. From the result we can see that the harmonic balance method is accurate and efficient.

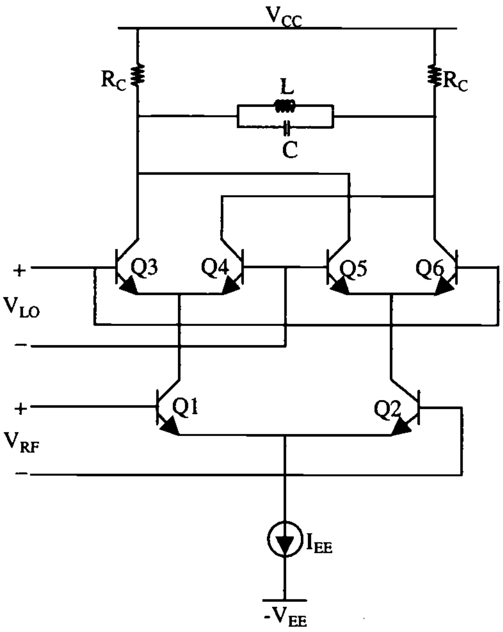


Figure 7.12: The circuit schematic of a double-balanced mixer.

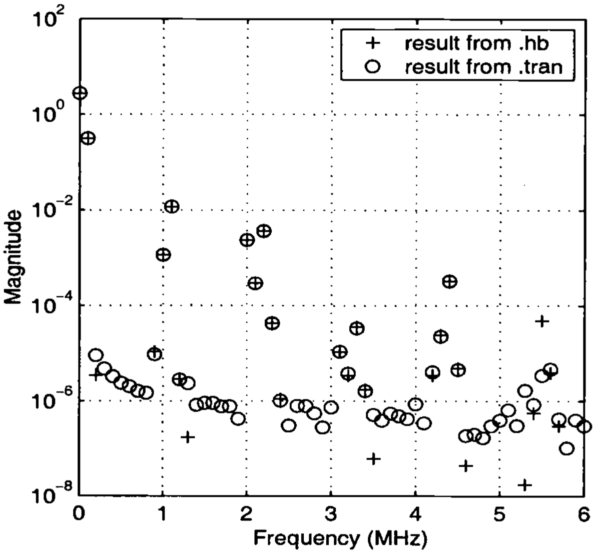


Figure 7.13. Output voltage spectrum for the double-balanced mixer from quasi-static harmonic balance and transient simulations.

To observe the performance of the quasi-static harmonic balance analysis, steady-state solutions of more example circuits from [29] and [14] were obtained. The number of Newton iterations required and the simulation times are summarized in Table 7.2. Given that numerical devices are used in the circuits, the simulations are very efficient as shown in the table.

Example circuits	# of iters	Time (min)
Simple rectifier	16	0.6
Single-BJT mixer	11	2.1
MOSFET CS amplifier	6	0.6
SCP amplifier	3	0.9
DC power supply	39	0.75
CB Class-C amp. (Q=10)	14	1
CB Class-C amp. (Q=50)	13	1
CB ampl. (w/ parasitics)	53	6.4
X3 frequency multiplier	8	0.5
Double-balanced mixer	7	24

Table 7.2: Performance of the quasi-static harmonic balance analysis for coupled device and circuit simulation.

To verify the quasi-static property of the steady-state solution obtain by the QS harmonic balance analysis, we consider the turn-on and turn-off transients of a MOS transistor. Figure 7.14 shows the simple circuit used to perform the simulation [2]. The 2D numerical MOSFET in the circuit has 31x19=589 mesh points and its doping profile is shown in Figure 7.15. The gate voltage is ramped from 0V to 5V in 50psec for turn on and from 5V to 0V for turn off. The voltage is held constant at 5V for 150psec such that all transients die out before the start of another ramp. These are fast switching conditions for the MOS transistor and lead to non-quasi-static behavior.

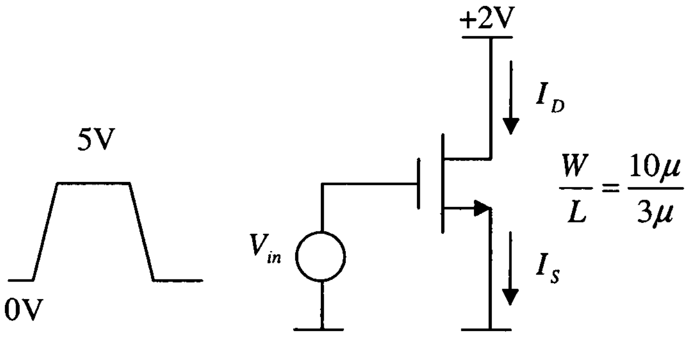


Figure 7.14: The simple circuit to observe the non-quasi-static behavior during the turn-on and turn-off transients of a MOSFET. A 2D numerical model is used for the MOSFET.

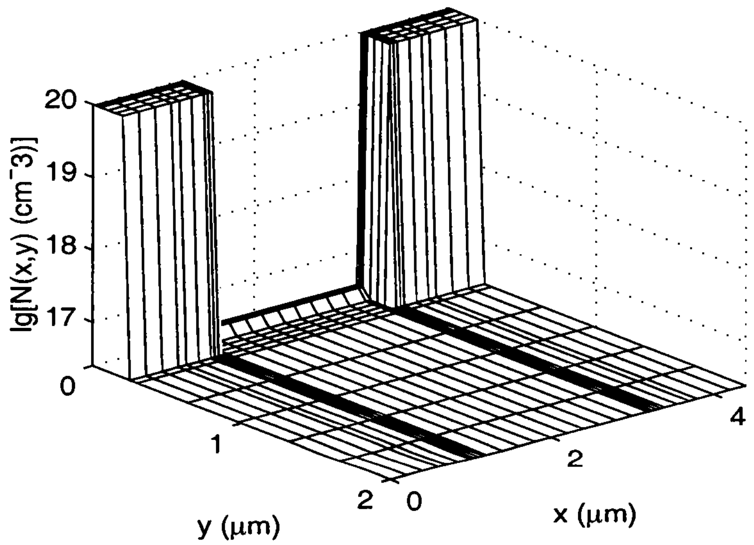


Figure 7.15: Doping profile of a 2D numerical MOSFET.

Due to the extremely fast transition in the currents, 100 harmonics are chosen for the harmonic balance analysis. The steady-state analysis of this circuit is first carried out using quasi-static harmonic balance method. The simulated quasi-static drain and source currents as a function of time, during turn on, are plotted in Figure

7.16. The currents for the turn off are plotted in Figure 7.17. The small ringing in the current waveforms is due to the Gibb's phenomena [49]. Then the non-quasi-static currents are obtained by a transient analysis of CODECS. For comparison, these results are also plotted in Figures 7.16 and 7.17 for turn on and turn off, respectively. Here using transient analysis instead of non-quasi-static harmonic balance is based on two considerations. First if the non-quasi-static harmonic balance is used, the total number of nonlinear equations to be solved is 186,327, even with one-carrier simulation. Obviously it is very expensive. Secondly, for this one-tone simple circuit, startup transient quickly dies out and steady-state solution is readily obtained in only several periods of transient simulation.

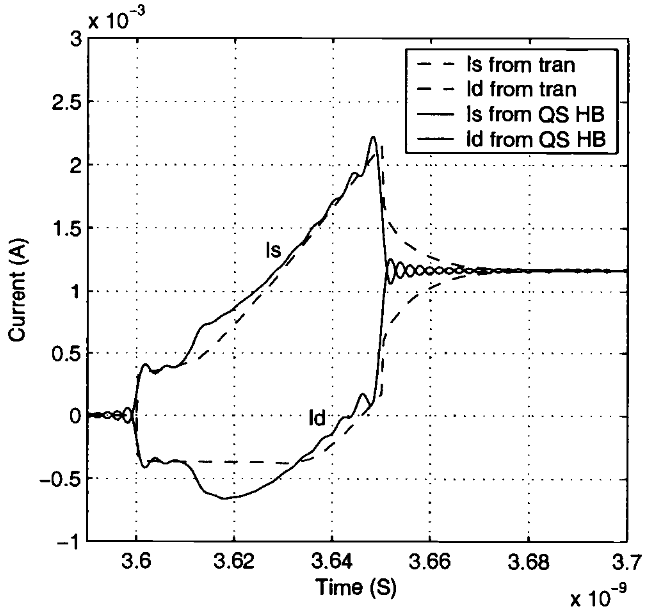


Figure 7.16: Comparison of the quasi-static and non-quasi-static source and drain currents of a MOSFET for the turn-on transient simulated by quasi-static harmonic balance and transient analyses, respectively.

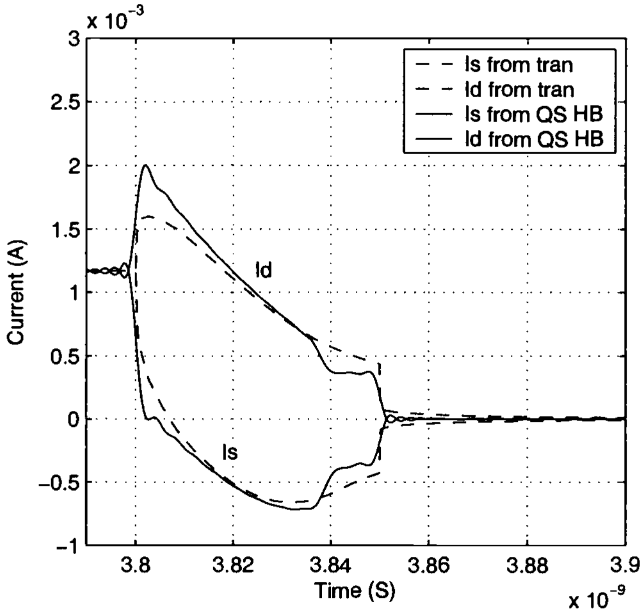


Figure 7.17: Comparison of the quasi-static and non-quasi-static source and drain currents of a MOSFET for the turn-off transient simulated by quasi-static harmonic balance and transient analyses, respectively.

Figure 7.16 shows that as the gate voltage exceeds the threshold voltage the currents from the QS harmonic balance simulation respond sharply. This is because for the quasi-static analysis channel charge reaches the steady-state value instantaneously. While for the currents from non-quasi-static analysis, due to the finite transit time of the carriers, the drain current increases after a time delay. Both the drain and source currents increase more smoothly as show in Figure 7.16. Based on the same reasoning, when the gate voltage reaches its steady-state value and becomes a constant, the quasi-static currents reach their steady-state values instantaneously whereas the non-quasi-static currents decrease or increase exponentially. Similar arguments can be used to explain the turn-off characteristics of the source and drain currents as shown in Figure 7.17.

7.2.2 Comparison between the QS and NQS implementations

To compare the accuracy of the quasi-static and non-quasi-static implementations, a simple common-source amplifier with a 2D numerical MOSFET model is used and shown in Figure 7.18. The input voltage waveform to the gate is given as $V_{in} = 2 + 0.5\sin(2\pi \cdot f_{in} \cdot t)$. The transition frequency of the MOSFET is 2GHz at $V_{gate} = 2V$ and $V_{drain} = 5V$. To evaluate the accuracy of the results at different frequencies, simulations were carried out at f_{in}=500MHz, 2GHz, and 5GHz, respectively. The steady-state output voltages obtained from the quasi-static and non-quasi-static harmonic balance methods were compared with accurate results from a time-domain analysis as shown in Figure 7.19. From this figure we can see that the results from the non-quasi-static implementation are in perfect agreement with the transient simulation while the quasi-static method is good only for relatively low frequencies. For this particular example, the maximum error is 3.8% for the quasi-static result at a frequency of 2GHz and 16.8% at 5GHz.

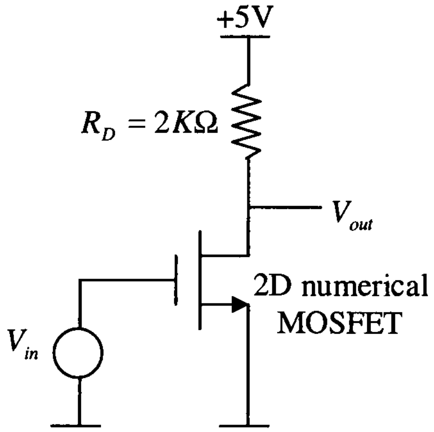
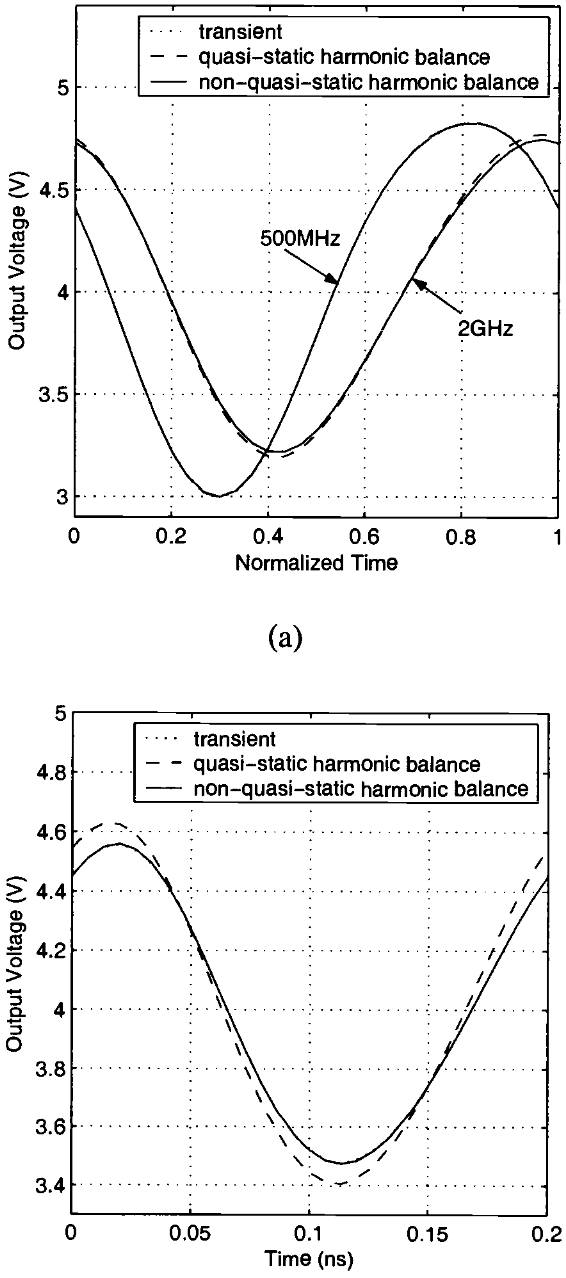


Figure 7.18: A simple common-source amplifier with a 2D numerical MOSFET model.



(a)

(b)

Figure 7.19: The steady-state output voltages of a simple common-source amplifier from transient, quasi-static and non-quasi-static harmonic balance simulations with (a) f_{in} =500MHz and 2GHz, and (b) f_{in} =5GHz.

The simulation complexity of the quasi-static and non-quasi-static implementations is compared for the simple common-source amplifier in Table 7.3 and a source-coupled-pair amplifier in Table 7.4. The numerical MOSFET has 589 mesh points and a two-carrier simulation is carried out. The simulation time for the non-quasi-static method is obtained using both the default solver Sparse in CODECS and a more efficient linear solver UMFPACK [50] to solve the large system of device-level equations. From these tables we can see that, for the same number of chosen frequencies, the quasi-static method is very efficient due to a smaller matrix size. Since a much larger number of harmonic balance equations at the device level have to be solved, the non-quasi-static method requires a significantly larger computational time and memory even when an efficient sparse matrix solver such as UMFPACK is used. The non-quasi-static simulation can be even impractical with fine meshed numerical devices and a large number of simulation frequencies.

	Quasi-static method	Non-quasi-static method	
		Sparse	UMFPACK
#Freqs chosen	4	4	4
Simulation time	83s	10737s	1853s
Memory used	9.9M	215M	141M
#Eqns for HB at circuit level	35	35	35
#Eqns for HB at device level	0	9429	9429
Largest matrix	1347 x 1347	9429x9429	9429x9429

Table 7.3: Comparison of simulation complexity using quasi-static and non-quasi-static approaches to obtain the steady state of a simple common-source amplifier with a numerical MOSFET model. The quasi-static method is orders of magnitude faster than the non-quasi-static implementation even when an efficient sparse matrix solver UMFPACK is used.

	Quasi-static method	Non-quasi-static method	
		Sparse	UMFPACK
#Freqs chosen	4	4	4
Simulation time	121s	19650s	3471s
Memory used	13.4M	409M	165M
#Eqns for HB at circuit level	63	63	63
#Eqns for HB at device level	0	18858	18858
Largest matrix	1347 x 1347	9429x9429	9429x9429

Table 7.4: Comparison of simulation complexity using quasi-static and non-quasi-static approaches to obtain the steady state of a source-coupled-pair amplifier with a numerical MOSFET model. The quasi-static method is orders of magnitude faster than the non-quasi-static implementation even when an efficient sparse matrix solver UMFPACK is used.

7.2.3 Comparison between the QS and MVS Implementations

The MVS harmonic balance analysis is carried out on the same simple common-source amplifier as used in Section 7.2.2 with $V_{in} = 2 + 0.1\sin(2\pi \cdot f_{in} \cdot t)$ and f_{in} =10GHz. The steady-state drain current is compared with the results from the quasi-static harmonic balance and accurate time-domain simulations as shown in Figure 7.20. From this figure we can see that the current from the quasi-static method has significant magnitude and phase errors at high frequencies. These errors are corrected by the MVS method and a more accurate result is obtained.

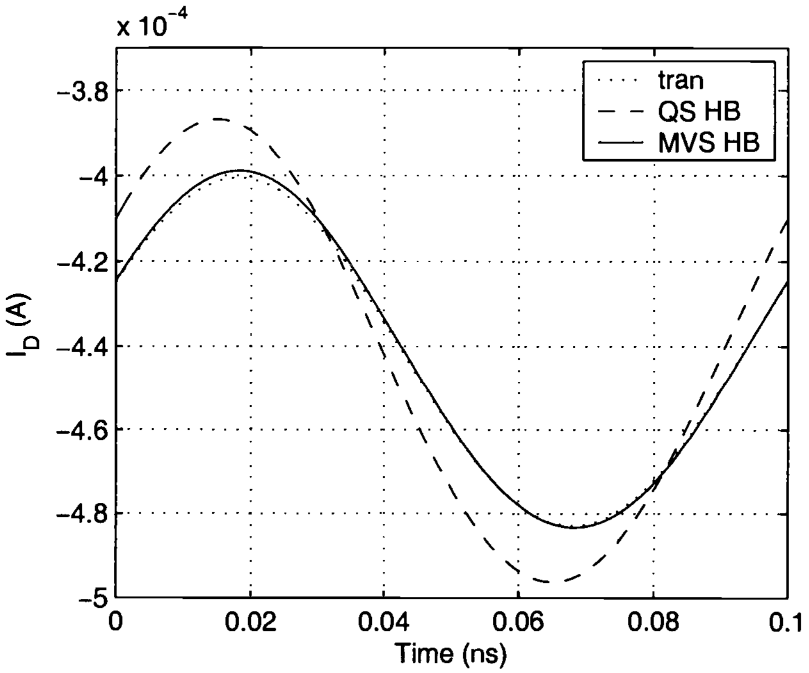


Figure 7.20: The steady-state drain currents of a simple common-source amplifier with f_{in} =10GHz from time-domain, quasi-static and MVS harmonic balance simulations.

To study the accuracy improvement of the MVS method over the QS method and their regions of validity, the 5% constant error loci of the simple CS amplifier are plotted as functions of the amplitude V_A and the frequency f_{in} of the sinusoidal input voltage. The error is given by the ratio of maximum discrepancy of the instantaneous value and the fundamental magnitude of the drain currents. Figure 7.21 is the loci without the output load and Figure 7.22 shows the loci with $2K\Omega$ resistor load. For the NQS method the region of validity is the whole plane whereas for the QS and MVS methods it is the area to the lower left of the corresponding loci. The MVS method has a larger region of validity than the QS method and is more accurate for high frequencies and small amplitudes. The error loci of the MVS method are in agreement with the qualitative plot shown in Figure 6.3. The accuracy of the MVS method depends on a trade off between the amplitude and frequency of the sinusoidal input voltage. For the QS method, the accuracy is mainly dependent on the input frequency. Comparisons between these two figures show that the error loci are configuration dependent. The loci are shifted to the right and regions of validity are enlarged for the configuration with the output load. For this numerical MOSFET, the QS method is accurate enough (error < 5%) up to 1GHz which is the half of the MOSFET transition frequency of 2GHz.

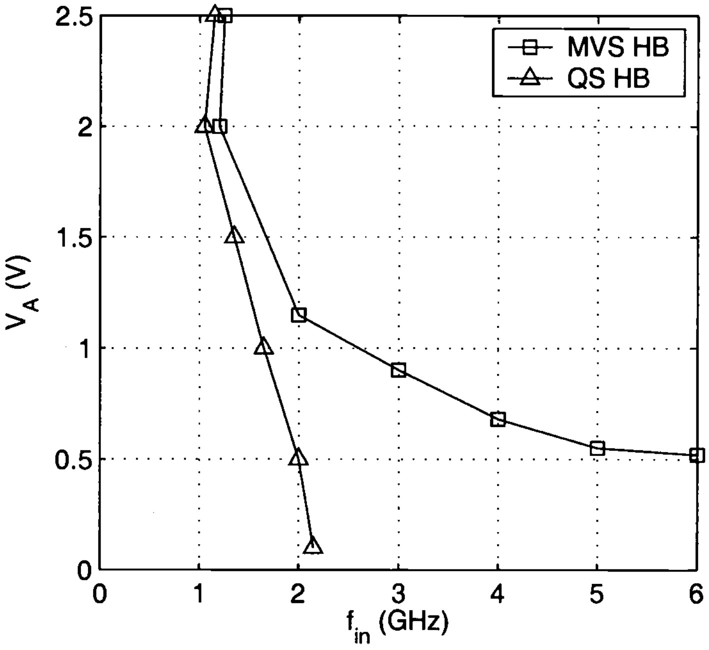


Figure 7.21: 5% constant error loci of MVS and QS methods for the simple common-source amplifier without an output load.

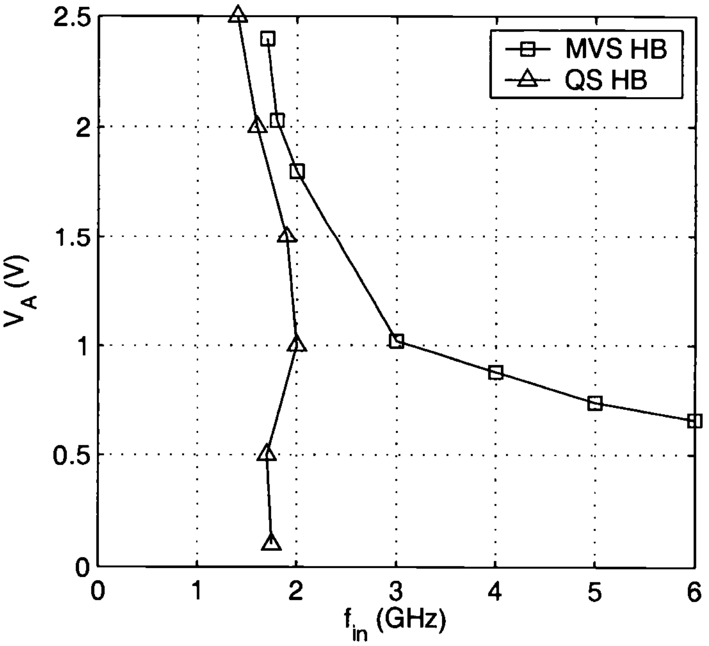


Figure 7.22: 5% constant error loci of MVS and QS methods for the simple common-source amplifier with a $2K\Omega$ resistor load.

To compare the performance of the QS and MVS methods, several example circuits with numerical devices and different number of harmonics are simulated using both methods. The simulation time and the memory used are summarized in Table 7.5. The table shows that the MVS method is slower than the QS method. This is because the MVS method requires multiple AC analyses for all frequencies chosen at each sampled time point while the QS method needs only one low frequency AC analysis. When more frequencies are used, more AC analysis are required in the MVS method at each sampled time point, and the MVS method is slower compared with the QS method. From the table we can also see that memory requirements of the MVS method are the same as the QS method because additional AC analyses in the MVS method do not require more memory.

	# of freqs	QS method		MVS method	
		Time (sec.)	Memory (MB)	Time (sec)	Memory (MB)
Simple rectifier	10	9	6.5	35	6.5
	30	37	7.2	365	7.2
CS amplifier	3	14	7.7	21	7.7
	10	30	7.9	87	7.9
SCP amplifier	3	22	9.3	52	9.3
	10	54	9.7	369	9.7
DC supply	10	84	6.8	100	6.8
	20	100	7.7	390	7.7
CB Class-C amp. (Q=10)	10	73	7.9	152	7.9
	20	259	10.4	579	10.4
CB Class-C amp. (Q=50)	10	78	8.0	155	8.0
	20	232	10.5	624	10.5
X3 frequency multiplier	6	12	7.2	25	7.2
	12	26	7.8	80	7.8

Table 7.5: Performance comparison of the QS and MVS harmonic balance methods for circuits with numerical devices and different number of harmonics.

7.3 Comparison Between Shooting and QS Harmonic Balance Methods

In this section, the comparison of the simulation performance of the time-domain and QS harmonic balance analyses for coupled device and circuit simulation is demonstrated using some typical RF circuits. The steady-state analyses of the circuits [29, 14] with large-signal inputs and numerical models using both methods were performed. The simulation results are the same for the two methods. The performance results and memory usage are summarized in Table 7.6.

Circuit	Shooting method			Harmonic balance		
	#iter	Time (sec)	Memory used (MB)	#iter	Time (sec)	Memory used (MB)
Periodic forced circuits						
Simple rectifier	2	28	6.4	16	37	7.2
DC power supply	6	81	6.6	39	45	6.8
CB amplifier	4	82	7.0	13	58	8.0
CB (w/ parasitics)	4	254	7.1	53	385	7.7
X3 freq. multiplier	6	10	6.2	8	32	6.8
MOS CS amplifier	3	554	9.8	5	188	10
SCP amplifier	6	1170	13	3	225	14
Mixers						
Single-BJT mixer	X	X	X	11	127	9.0
Double-bal. mixer	X	X	X	7	1440	20
Oscillators						
Colp. Oscil. (60MHz)	18	108	6.7	C	267	6.4
LC EC oscillator	10	41	6.4	C	1210	8.4
TNT oscillator	7	42	6.4	*	*	*
Wien oscillator	5	33	6.4	*	*	*

Table 7.6: Performance and memory comparison of time-domain and QS harmonic balance methods for coupled device and circuit simulation. ("X"—time-domain method is impractical, "*"—harmonic balance method fails to converge, "C"—continuation method is used.)

From these tables it is clear that both methods solve periodic forced circuits for coupled device and circuit simulation with good computational performance. The QS harmonic balance method uses slightly more memory than the shooting method. Comparing the number of iterations, for highly nonlinear circuits the QS harmonic balance method required a significantly larger number of iterations for convergence than the time-domain method. However, each iteration consumes less time and the overall performance is usually comparable with the time-domain method. It should be noted that for the MOS CS amplifier and the SCP amplifier with numerical device models, the harmonic balance method takes 3 and 5 times less simulation time than the shooting method, respectively. This is because these two circuits are weakly nonlinear and the harmonic balance method converges in a small number of iterations. The transient simulation is time-consuming in these cases.

In handling multi-tone circuits such as mixers, the harmonic balance method is excellent while the time-domain shooting method is usually impractical. Let us take the single-BJT mixer circuit in the table as an example. The harmonic balance analysis took 11 iterations and 127 seconds. For the time-domain method a simulation of several periods with at least 50,000 time points in each period should be carried out.

For autonomous circuits such as oscillators, the time-domain shooting method is efficient due to several modifications and heuristics in Chapter 4. To improve the convergence property of the harmonic balance method for oscillator circuits, the voltage probe and a technique based on a continuation method were introduced in [42] and [43], respectively. With numerical models in CODECS, the harmonic balance method is not very reliable. Furthermore, the computational costs are higher than that of the time-domain shooting method.

Finally the performance comparison of the two methods for coupled simulation is summarized in Table 7.7.

	Shooting method	QS Harmonic balance
Periodic forced highly nonlinear circuits	excellent	Excellent
Multi-tone cases (Mixers)	poor	Excellent
Oscillators	excellent	Fair
Numerical devices with a large # of mesh points	Good	Excellent

Table 7.7: Summary of performance comparison of the time-domain shooting method and the QS harmonic balance method for coupled device and circuit simulation.

7.4 Summary

In this chapter several examples are used to verify the implementations of the time-domain shooting method and the frequency-domain harmonic balance method in the context of coupled device and circuit simulation. Also the steady state results of RF circuits with analytical and numerical models for semiconductor devices were compared. The differences between these indicate that numerical models are essential for accurate simulation of high frequency circuits.

The advantages and limitations of the QS, NQS and MVS harmonic balance implementations are verified as well. The non-quasi-static approach is accurate but requires significant computing resources. The implementation of the quasi-static approach is straightforward and the simulation is efficient. However, the accuracy is limited by the transition frequency of numerical devices. The modified Volterra series approach has improved accuracy compared with the quasi-static approach for high frequency and small amplitude signals. This improvement is achieved at the expense of longer simulation times.

Finally, a performance comparison of time-domain and QS harmonic balance methods for coupled simulation is presented. For the periodic forced circuits, the two methods are comparable. The harmonic balance method is excellent in handling multi-tone circuits where the time-domain method is usually impractical. For autonomous circuits, the time-domain shooting method is efficient and reliable while the harmonic balance method needs more convergence improvements for circuits with numerical devices.

Chapter 8

CONCLUSIONS

8.1 Summary of Contributions

This work presents the first implementation of steady-state analysis in the context of coupled device and circuit simulation. Both time-domain shooting method and frequency-domain harmonic balance method have been implemented in a general coupled device and circuit simulator – CODECS. In the implementation, the original architecture of CODECS is exploited allowing reuse of the existing coupling simulation framework. With critical devices modeled by physical (numerical) models, this simulator can obtain the steady-state solution accurately and efficiently for RF IC building blocks.

The time-domain shooting method has been implemented with Newton's method. Modifications, such as state elimination and damping, and heuristics have been applied to improve the reliability and convergence of the Newton shooting algorithm. A new scheme to bias the numerical devices is implemented to improve the efficiency. A circuit unknown implementation is developed to eliminate the inconsistency and discontinuity in the standard state-based implementation.

Three different implementation approaches of the harmonic balance method for coupled device and circuit simulation are investigated and implemented. These include the quasi-static (QS), non-quasi-static (NQS), and modified-Volterra-series

(MVS) approaches. Comparisons of simulation and performance results verify that there is a tradeoff between simulation accuracy and complexity among these approaches.

Although the steady-state methods were only implemented into CODECS, the generality of numerical device simulators was kept in mind in the course of implementation. The time-domain shooting method and the QS and MVS approaches of the frequency-domain harmonic balance method were implemented at the circuit level with minor changes to the device simulator. The standard analysis routines available in device simulators such as DC biasing, small-signal analysis, and transient analysis were used to connect the numerical device simulator with the circuit-level steady-state analyses. For the NQS approach, although extensive modifications to the device simulator are required, the interface of the numerical devices with the circuit-level equations was not changed. This decoupled implementation strategy enables incorporation of general purpose device simulators in the coupled device and circuit simulator CODECS. This implies that steady-state simulations of technologies such as SiGe and SOI, and also optical devices can be made available in CODECS. Furthermore, the decoupling between circuit-level computation and device simulators indicates a decoupling between the numerical device evaluations. Therefore, numerical device evaluations can be readily parallelized using a cluster of workstations.

8.2 Future Work

In this work, the non-quasi-static approach of the harmonic balance method for coupled device and circuit simulation has been implemented for one-dimensional diode and two-dimensional MOSFET and is not available for two-dimensional diodes and one- and two-dimensional BJTs. Extensions of this approach to all types of numerical devices in CODECS is necessary to simulate circuits with different numerical models.

The non-quasi-static approach results in a large device-level system matrix. For the quasi-static and the modified-Volterra-series approaches, a large circuit size also generates a large harmonic balance Jacobian matrix. Factorization of such a large matrix is expensive and the simulation is extremely time and memory consuming. To solve the large-scale system of harmonic balance equations, efficient Krylov subspace methods should be considered.

In the modified-Volterra-series approach, the first-order modified Volterra series is applied to periodic steady-state problems. It will be interesting to investigate the second-order or even the third-order modified Volterra series and observe the accuracy improvement and the tradeoff with simulation time. The challenge here is to efficiently compute the second- or third-order modified Volterra kernels for the numerical devices. Also the modified-Volterra-series approach needs further investigation for quasiperiodic steady-state problems.

The parallel implementation of numerical device evaluations for speed improvements is another area that needs further investigation. In addition, the incorporation of general purpose device simulators for simulation of other technologies in CODECS is important. As discussed at the end of Section 8.1, the implementation strategy in this work has paved the road for these two future activities.

Finally, the harmonic balance method with Newton's method is not very reliable for autonomous circuits with numerical models. More robust methods, such as homotopy methods, are recommended to improve the convergence.

BIBLIOGRAPHY

[1] W. L. Engl, R. Laur, and H. K. Dirks, "MEDUSA—A simulator for modular circuits," *IEEE Transactions on Computer-Aided Design*, vol. 1, pp. 85-93, Apr. 1982.

[2] K. Mayaram, "CODECS: A mixed-level circuit and device simulator," *Memo. No. UCB/ERL M88/71*, Nov. 1988.

[3] J. G. Rollins and J. Choma, "Mixed-mode PISCES-SPICE coupled circuit and device solver," *IEEE Transactions on Computer-Aided Design*, vol. 7, pp. 862-867, Aug. 1988.

[4] T. Grasser, "Mixed-mode device simulation," *a dissertation for Ph.D. degree*, Technischen Universität Wien, May 1999. http://www.arcs.ac.at/DissDB/diss/TW/EL/trn100008.

[5] K. S. Kundert, J. K. White, A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*, Kluwer Academic Publishers, 1990.

[6] B. Troyanovsky, "Frequency domain algorithm for simulating large signal distortion in semiconductor devices," *Ph.D. dissertation*, Stanford University, Nov. 1997.

[7] F. M. Rotella, G. Ma, Z. Yu, and R. W. Button, "Modeling, analysis, and design of RF LDMOS devices using harmonic-balance device simulation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, pp. 991-999, Jun. 2000.

[8] Y. Hu and K. Mayaram, "Periodic steady-state analysis for coupled device and circuit simulation," *Proc. SISPAD 2000*, pp. 90-93, Sep. 2000.

[9] Y. Hu and K. Mayaram, "Harmonic balance analysis for coupled device and circuit simulation," *Proc. RAWCON 2001*, pp. 137-140, Aug. 2001.

[10] Y. Hu and K. Mayaram, "An efficient algorithm for large-signal frequency-domain coupled device and circuit simulation," *Proc. ISCAS 2002*, vol. 5, pp. 329-332, May 2002.

[11] Y. Hu and K. Mayaram, "A comparison of non-quasi-static and quasi-static harmonic balance implementations for coupled device and circuit simulation," *Proc. CICC 2003*, pp. 99-102, Sep. 2003.

[12] Y. Hu, X. Duan, and K. Mayaram, "A comparison of time-domain and harmonic balance steady-state analyses for coupled device and circuit simulation," *Proc. NEWCAS 2004*, Jun. 2004.

[13] B. Razavi, "Challenges in portable RF transceiver design," *IEEE Circuits and Devices Magazine*, vol. 12, pp. 12-25, Sep. 1996.

[14] D. O. Pederson and K. Mayaram, *Analog Integrated Circuits for Communication: Principles, Simulation and Design*, Kluwer Academic Publishers, 1991.

[15] K. S. Kundert, "Introduction to RF simulation and its application," *IEEE Journal on Solid-State Circuits,* vol. 34, pp. 1298-1319, Sep. 1999.

[16] B. Razavi, *RF Microelectronics*, Prentice Hall, 1998.

[17] T. J. Aprille and T. N. Trick, "Steady-state analysis of nonlinear circuits with periodic inputs," *Proc. IEEE,* vol. 60, pp. 108-114, Jan. 1972.

[18] T. J. Aprille and T. N. Trick, "A computer algorithm to determine the steady-state response of nonlinear oscillator," *IEEE Transactions on Circuit Theory,* vol. CT-19, pp. 354-360, Jul. 1972.

[19] F. R. Colon and T. N. Trick, "Fast periodic steady-state analysis for large-signal electronic circuits," *IEEE Journal on Solid-State Circuits*, vol. SC-8, pp. 260-268, Aug. 1973.

[20] F. B. Grosz and T. N. Trick, "Some modifications to Newton's method for the determination of the steady-state response of nonlinear oscillatory circuits," *IEEE Transactions on Computer-Aided Design,* vol. 1, pp. 116-119, Jul. 1982.

[21] L. O. Chua, C. A. Desoer, E. S. Kuh, *Linear and Nonlinear Circuits*, McGraw-Hill Book Company, NY 1987.

[22] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," *Memo. No. ERL-M520*, Electronics Research Laboratroy, University of California, Berkeley, May 1975.

[23] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol CAS-22, pp. 504-509, June 1975

[24] W. J. MaCalla, *Computer-Aided Circuit Simulation Techniques*, Kluwer Publishers, 1987.

[25] S. E. Laux, "Techniques for small-signal analysis of semiconductor devices," *IEEE Transactions on Electron Devices*, vol. ED-32, pp. 2028-2037, Oct. 1985.

[26] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer-Verlag, New York, 1984.

[27] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Clarendon Press, Oxford, 1978.

[28] S. Skelboe, "Computation of the periodic steady-state response of nonlinear networks by extrapolation methods," *IEEE Transactions on Circuits and Systems*, vol. CAS-27, pp. 161-175, Mar. 1980.

[29] P. N. Ashar, "Implementation of algorithms for the periodic-steady-state analysis of nonlinear circuts," *Memo. No. UCB/ERL M89/31*, Mar. 1989.

[30] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*, Springer-Verlag, NY, 1990.

[31] L. T. Watson, S. C. Billups, and A. P. Morgan, "Algorithm 652: HOMOPACK: a suite of codes for globally convergent homotopy algorithms," *ACM Transactions on Mathematical Software*, vol. 13, pp. 281-310, Sep. 1987.

[32] E. Fung, "Homotopy methods for dc operating point and periodic steady-state analysis: SPICE3 implementation," *M.S. Thesis*, University of California at Berkeley, 1998.

[33] L. Trajkovic, E. Fung, and S. Sanders, "HomSPICE: simulator with homotopy algorithms for finding dc and steady-state solutions of nonlinear circuits," *Proc. ISCAS*, Jun. 1998

[34] W. Ma, "SspiceHom: A simulator for periodic steady-state analysis of nonlinear circuits using homotopy methods," *M.S. Thesis*, Washington State University, May 2001.

[35] W. Ma, L. Trajkovic, K. Mayaram, "HomSSPICE: A homotopy-based circuit simulator for periodic steady-state analysis of oscillators," *Proc. ISCAS 2002*, vol. 1, pp. 645-648, May 2002

[36] T. N. Trick, F. R. Colon, and S. P. Fan, "Computation of capacitor voltage and inductor current sensitivities with respect to initial conditions for the steady-state analysis of nonlinear periodic circuits," *IEEE Transactions on Circuit and Systems*, vol. CAS-22, pp. 391-396, May 1975.

[37] G. D. Hachtel, R. K. Brayton, and F. G. Gustavson, "The sparse tableau approach to network analysis and design," *IEEE Transactions on Circuit and Theory*, vol. CT-18, pp. 101-113, Jan. 1971.

[38] P. L. Heron and M. B. Steer, "Jacobian calculation using the multidimensional fast Fourier transform in the harmonic balance analysis of nonlinear circuits," *IEEE Transactions on Microwave Theory and Techniques*, vol. 38, pp. 429-431, April 1990.

[39] R. Freund, G. H. Golub, and N. M. Nachtigal, "Iterative solution of linear systems," *Acta Numeric*, pp. 57-100, 1992.

[40] R. C. Melville, P. Feldmann, and J. Roychowdhury, "Efficient multi-tone distortion analysis of analog integrated circuits," *Proc. CICC*, pp. 241-244, May 1995.

[41] R. Telichevesky, K. Kundert, I. Elfadel, and J. White, "Fast simulation algorithms for RF circuits," *Proc. CICC*, pp. 437-444, May 1996.

[42] E. Ngoya, A. Suarez, R. Sommet, and R. Quere, "Steady state analysis of free or forced oscillators by harmonic balance and stability investigation of periodic and quasi-periodic regimes," *International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, vol. 5, pp. 210-233, Mar. 1995.

[43] M. Gourary, S. Ulyanov, M. Zharov, S. Rusakov, K. Gullapalli, and B. Mulvaney, "Simulation of high-Q oscillators," *Proc. ICCAD 98*, pp. 162-169, 1998.

[44] F. Filicori, G. Vannini, V. A. Monaco, "A nonlinear integral model of electron device for HB circuit analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, pp. 1456-1465, Jul. 1992.

[45] D. Mirri, G. Iuculano, F. Filicori, G. Pasini, G. Vannini, and G. P. Gualtieri, "A modified Volterra series approach for nonlinear dynamic systems modeling," *IEEE Transactions on Circuits and Systems I*, vol. 49, pp. 1118-1128, Aug. 2002.

[46] M. Schetzen, *The Volterra and Wiener Theories of Non-Linear Systems*. New York: Wiley-Interscience, 1980.

[47] A. Hajimiri and T. H. Lee, "A general theory of phase noise in electrical oscillators," *IEEE Journal on Solid-State Circuits,* vol. 33, pp. 179-194, Feb. 1998.

[48] M. Kakizaki and T. Sugawara, "A modified Newton method for the steady-state analysis," *IEEE Transactions on Computer-Aided Design*, vol. 4, pp. 662-667, Oct. 1985.

[49] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems*, Prentice Hall, 1997.

[50] T. A. Davis, *UMFPACK Version 4.3 User Guide*, Tech Report TR-03-008, CIS Dept., Univ. of Florida, 2003. http://www.cise.ufl.edu/research/sparse/umfpack/