# Efficient Nonlinear Distortion Analysis of RF Circuits

by

Dani A. Tannir

Department of Electrical & Computer Engineering

McGill University, Montreal

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Engineering.

Copyright ©Dani A. Tannir 2006

October 4, 2006



Library and Archives Canada

Published Heritage Branch

395 Wellington Street Ottawa ON K1A 0N4 Canada Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington Ottawa ON K1A 0N4 Canada

> Your file Votre référence ISBN: 978-0-494-28625-8 Our file Notre référence ISBN: 978-0-494-28625-8

### NOTICE:

The author has granted a nonexclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or noncommercial purposes, in microform, paper, electronic and/or any other formats.

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

#### AVIS:

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

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

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

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



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

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

### Acknowledgments

After two years of hard work and dedication, it is finally here, the moment at which I have completed and am ready to submit my Masters thesis. I thank God for giving me the ability and the opportunity to achieve this goal. Although this thesis marks the pinnacle of my academic achievements thus far, it is important to remember that this work could not have been completed without the assistance of a select group of people.

First and foremost I would like to say a special word of gratitude to my research supervisor, professor Roni Khazaka. I am very grateful for all his assistance, advice and contributions to my research and to the rest of my academic needs throughout my time as a Masters student. His insight and technical feedback have had a significant positive influence on this research and for these reasons I consider Professor Khazaka as a true mentor, colleague and friend. I look forward to working with him in the future.

The two years that I have spent as a Masters student have also been my first in Montreal and in Canada. During this time, I have been fortunate to meet several people that have helped me settle and adjust to my new life. I would specifically like to thank Sadok Aouini, Simon Hong, Tarek Alhajj, Arash Khajooeizadeh, Ahmed Abdelaaty, Yousef Elkurdi, Usman Khalid, Baker Haddadin, Ashraf Haddad, Mohammad Taherzadeh and the rest of the MACS lab gang for their help, support and friendship. I would also like to thank all the people that have had a positive influence throughout my academic life, from friends, relatives and professors, including everyone back home in Lebanon and in Saudi Arabia, including Manar El-Chammas, Tammam Yamout and Professor Mazen Saghir. Finally, it is important to remember that none of this would have been possible without the tremendous guidance and support that I have received from my family throughout my life. To my elder brother, Mustapha, who has been my best friend throughout my life and for all the support, advice and good times we have had together. And also, to my parents, Abdallah and Zouhar, who have provided me with the best possible lifestyle and for all their care, sacrifice and patience through the many years and the difficult times. Without them I would not be the person that I am today and for this I am eternally grateful.

### Abstract

The third order intercept point (IP3) is an important figure of merit which gives a measure of the linearity of communication circuits. There are two classes of methods for calculating the IP3 of a circuit. The first is analytical and is usually based on Volterra series. This approach is cumbersome and is difficult to apply to arbitrary circuits with arbitrary non-linearities. The second class of methods is based on multi-tone simulations and is general and flexible but requires significant CPU cost. In this thesis a new method based on the computation of the circuit moments is proposed. The new approach uses the circuit moments in order to numerically compute the Volterra kernels. This automates the process of numerically obtaining such kernels for any circuits. The proposed approach is simple to apply and presents a significant improvement in CPU cost over existing methods.

### Abrégé

Le point d'interception de troisième ordre (IP3) est un important critère qui procure une mesure de la linéarité des circuits de communication. Il y a deux classes de méthode pour calculer le IP3 d'un circuit. La première est une méthode analytique et est généralement basée sur les séries de Volterra. Cette approche est encombrante et difficilement applicable à des circuits arbitraires non-linéaires. La deuxième classe de méthode est basée sur des simulations à fréquences multiples et est générale et flexible, mais requiert un important nombre de calculs. Dans ce mémoire, une nouvelle méthode basée sur le calcul des moments d'un circuit est proposée. Cette nouvelle approche utilise les moments du circuit afin de calculer de façon numérique les noyaux de Volterra. Celle-ci automatise le processus de calcul numérique pour obtenir ces noyaux pour tous les types de circuits et par conséquent permet de calculer le IP3 efficacement. La méthode proposée est facile d'application et présente une amélioration significative du nombre de calculs requis par rapport aux méthodes existantes.

# **Table of Contents**

List of Tables viii				
List of Figures ix				
1	Intr	oductio	n	1
	1.1	Backg	round and Motivation	1
	1.2	Contri	butions	4
	1.3	Organi	ization of the Thesis	4
2	Sim	ulation	of RF Circuits	6
	2.1	Introd	uction	6
	2.2	Modifi	ied Nodal Analysis Formulation	7
	2.3	Shooti	ng Methods	9
	2.4	Harmo	onic Balance Method	10
		2.4.1	HB Formulation	10
		2.4.2	Solution of HB Equations	13
		2.4.3	Computation of Objective Function	15
		2.4.4	Harmonic Balance Jacobian	18
		2.4.5	Fourier Transform for Almost Periodic Input Signals	19
		2.4.6	Convergence	22
	2.5	Conclu	usion	23
3	Dist	ortion A	Analysis of RF Circuits	25
	3.1	Linear	Versus Nonlinear Distortion	25

	3.2	Import	tance of Nonlinear Distortion	26
	3.3	Termin	nology and Figures of Merit	28
3.4 Distortion Analysis Methods			tion Analysis Methods	31
		3.4.1	Distortion Analysis using Harmonic Balance	31
		3.4.2	Linear Centric Models	32
		3.4.3	Simplified Newton Method	34
	3.5	Volteri	ra Series	35
		3.5.1	Volterra Series Formulation	35
		3.5.2	Limitations of Volterra Series	37
		3.5.3	Time Varying Volterra Series	37
	3.6	Conclu	usion	38
4	Dire	ct Num	erical Computation of Intermodulation Distortion	39
	4.1	Introdu	uction	39
	4.2	Calcul	ation of the Moments	41
		4.2.1	Obtaining the Derivatives	44
	4.3	Numer	rical Computation of Volterra Kernels for	
		Detern	nining IP3	46
		4.3.1	Memoryless Systems	46
		4.3.2	Systems with Memory Elements	48
	4.4	Comp	utation of IP3 from Volterra Kernels	50
	4.5	Mome	nts Calculation Example	52
	4.6	Numer	rical Examples	57
		4.6.1	Example 1	57
		4.6.2	Example 2	62
		4.6.3	CPU cost comparison	67
5	Con	clusion	s and Future Research	72
	5.1	Conclu	usion	72
	5.2	Furthe	r Research	73
		5.2.1	Distortion Analysis of Mixer Circuits	73

5.2.2 Multi-tone Distortion Analysis using Single-tone Simulation 74

# **List of Tables**

1.1	Some Common Applications in the Wireless Market	2
4.1	Location of Taylor Series coefficients in Moment Vectors	47
4.2	Volterra Kernel Locations in Moment Vectors	52
4.3	Numerical Values of Moment Vectors for sample circuit	58
4.4	Circuit Element Values for Example 1	60
4.5	Distortion Analysis of Example 1 using Traditional Approach	60
4.6	Distortion Analysis of Example 1 using Proposed Method	61
4.7	Circuit Element Values for Example 2	66
4.8	Distortion Analysis of Example 2 using Traditional Approach	66
4.9	Distortion Analysis of Example 2 using Proposed Method	67
4.10	Comparison of CPU times between proposed method and HB solution	71

# **List of Figures**

1.1	Receiver's RF interface	4
2.1	Simple Nonlinear Circuit	8
2.2	a) Solution based on Transient Analysis b) Steady-State analysis	
	using Shooting method	9
2.3	Newton Raphson Iteration Flowchart	16
2.4	Frequency mapping using block truncations [1]	21
2.5	Frequency mapping using diamond truncations [1]	22
2.6	Continuation Example	23
3.1	Absolute Value of Frequency Response of a Linear Amplifier	26
3.2	Response of Amplifier to Square Wave Input	27
3.3	Concept of Nonlinear Distortion	27
3.4	Output Spectrum of Nonlinear circuit with 2 frequency excitation	28
3.5	Illustration of definitions of distortion parameters	29
3.6	Overall linear-centric circuit model [2]	33
3.7	Schematic Representation of a system characterized by a Volterra	
	Series	36
4.1	Overview of Proposed Method	40
4.2	Summary of the proposed algorithm	53
4.3	Example circuit to illustrate computation of moment vectors	53
4.4	Circuit Diagram Example 1	59
4.5	Output Power of Fundamental and IM3 tones of Example 1	61

4.6	Output Voltage Spectrum for Circuit in Example 1	62
4.7	Sparsity Pattern of HB Jacobian for Circuit in Example 1	63
4.8	Sparsity Pattern of DC Jacobian for Circuit in Example 2	64
4.9	Circuit Diagram Example 2	65
4.10	Output Power of Fundamental and IM3 Tones of Example 2	68
4.11	Output Voltage Spectrum for Example 2	68
4.12	Sparsity Pattern of HB Jacobian for Circuit in Example 2	69
4.13	Sparsity Pattern of DC Jacobian for Circuit in Example 2	70

# Notation

A	Matrix A	
$oldsymbol{A}^T$	Matrix A transposed	
$A^{-1}$	Matrix A inverted	
$\Re^n$	The set of real vectors of size n	
$\Re^{n  imes n}$	The set of real matrices of size $n \times n$	
Г	Inverse DFT matrix	
$oldsymbol{x}(t)$	Vector of unknown voltages and currents in the time domain	
G	Matrix containing MNA stamps of linear memoryless elements	
C	Matrix containing MNA stamps of linear memory elements	
f(x)	Vector of nonlinear scalar functions in MNA equations	
<b>b</b> (t)	Vector of independent sources in MNA equations	
т	Period of a signal	
$\omega_k$	Set of fundamental, harmonic and intermodulation frequencies	
X	Vector of sine and cosine components of unknown voltages and cur-	
	rents in frequency domain	
$ar{G}$	Block matrix containing stamps of linear memoryless elements in	
	expanded HB format	
$ar{C}$	Block matrix containing stamps of linear memory elements in ex-	
	panded HB format	
F(X)	Vector containing sine and cosine components of scalar nonlinear	
	functions	
Н	Highest order of harmonics considered	
$N_b$	Expanded number of variables for each original variable	

$N_h$	Total number of variables for HB system of equations		
$oldsymbol{\Phi}(oldsymbol{X})$	Objective function for HB solution		
$oldsymbol{J}_{HB}$	Harmonic Balance Jacobian matrix		
$h_n(\tau_1,\ldots,\tau_n)$	$n^{th}$ order Volterra kernel		
$H_n(j\omega_1,,j\omega_n)$	$n^{th}$ order Volterra kernel in Frequency Domain		
α	Small signal input amplitude		
$oldsymbol{A}_n$	$n^{th}$ moment vector		
$\boldsymbol{D}_n$	$n^{th}$ Taylor coefficient of $oldsymbol{F}(oldsymbol{X})$		
$\boldsymbol{T}_n$	$n^{th}$ Taylor coefficient of $\partial oldsymbol{F}(oldsymbol{X})/\partial oldsymbol{X}$		
$\boldsymbol{P}_n$	$n^{th}$ Taylor coefficient of block matrices in $T$		
${oldsymbol g}_n$	$n^{th}$ Taylor coefficient of vector of derivatives of nonlinear scalar		
	functions		

# **List of Abbreviations**

BVP	Boundary Value Problem
CAD	Computer Aided Design
CPU	Central processing unit
DFT	Direct Fourier Transform
EDA	Electronic design automation
FFT	Fast Fourier transform
GPS	Global Positioning System
HB	Harmonic Balance
HD	Harmonic Distortion
IC	Integrated circuit
IDFT	Inverse Direct Fourier Transform
IFFT	Inverse fast Fourier transform
IIP3	Input 3 <sup>rd</sup> Order Intercept Point
IM3	3 <sup>rd</sup> Order Intermodulation
IMFDR	Intermodulation Free Dynamic Range
IP3	3 <sup>rd</sup> Order Intercept Point
KCL	Kirchhoff's current law
LNA	Low Noise Amplifier
LO	Local Oscillator
LU	Lower/upper triangular decomposition.
MNA	Modified nodal analysis
MVP	Matrix vector product
NR	Newton Raphson iteration

OIP3 Output 3<sup>rd</sup> Order Intercept Point

RF Radio frequency

THD Total Harmonic Distortion

# Chapter 1

## Introduction

### **1.1 Background and Motivation**

Radio Frequency (RF) communication circuits such as Low Noise Amplifiers (LNAs) and mixers have become a staple in modern electronic products. The consumer and industry demands for wireless technologies have dramatically increased in recent years and this trend is projected to continue for the forseeable future. The need to go wireless is fueled by the need to enable people to access and communicate in any medium (voice, data, image or multimedia) at anytime, anywhere, in a timely and cost-effective way. The wireless market has significantly grown in recent years to include cellular phones, Global Positioning Systems (GPS) devices, and other applications. Some of the most common applications can be found in Table 1.1. The increasing demand for low-cost mobile communication systems has put a significant emphasis on efficient design and a reduced time to market for such applications. This can only be achieved using more effective simulation and design automation tools.

While current market pressures have underlined the importance of Electronic Design Automation (EDA), the increased complexity of modern designs has stretched the limits of current Computer Aided Design (CAD) software. In fact, the performance of a particular design is increasingly limited by the capabilities of the CAD software rather than by what can be optimally achieved using a particular

Communication Medium	Applications
Voice	Cordless Telephone
	Cellular Mobile Radio
Data	Wide Area Wireless Data
	Wireless Local Area Network and Bluetooth
	Paging/Messaging
Other	Home Satellite Network
	Global Positioning Systems (GPS)
	RF Identification (RFID)

Table 1.1: Some Common Applications in the Wireless Market

technology. The target application of the work presented in this thesis is the receiver/transmitter section which forms an integral part of any wireless communications device. Fig. 1.1 shows a typical receiver architecture (the superheterodyne receiver [3], [4]), along with its RF front end. The design requirements of the individual blocks in Fig. 1.1 typically include stringent conditions on intermodulation distortion, or in other words, on the linearity of the circuit. The main figure of merit used by RF engineers for characterizing the linearity of a circuit is the third order intercept point (IP3) which provides a measure of the third order intermodulation product. There are two classes of methods for obtaining the IP3 of a circuit. The first relies on brute force simulation and essentially mimics laboratory measurement techniques in order to obtain the value of IP3. The second class of methods are analytical techniques mostly based on Volterra series.

Using simulation based approaches, the IP3 is computed by obtaining the steady state response due to two input tones (for example  $\omega_1$  and  $\omega_2$ ) in the frequency band of the circuit under test. In this case, the third order intermodulation product is simply the component of the output tone at the frequency  $2\omega_2 - \omega_1$ . The main disadvantage of this approach is the CPU cost associated with obtaining the non-linear steady state response. It is possible to obtain the nonlinear steady-state response using a Spice-like simulator by performing a long enough transient analysis until the transients die out. Such an approach is, however, extremely inefficient due to large deviations in the time constants of the circuit and the input frequencies which result in a very large number of time steps being required before reaching steady-state [5].

The Harmonic Balance approach [1], [6] was proposed to address this difficulty. This approach expresses the circuit response as a sum of sines and cosines at the frequencies present at steady state. The problem is thus converted into that of solving a set of nonlinear algebraic equations where the unknowns are the coefficients of the sines and cosines. There are two significant limitation to this approach. The first is that the system of equations becomes one with a large number of variables, thus making it CPU intensive to solve. The second problem is the density of the matrices when solving the system, which also leads to increased CPU cost. The problem is amplified when more than one input tone is present as is the case when performing distortion analysis. When multiple input tones are present, the steady state response contains not only the harmonics of the input, but also the intermodulation products which significantly increase the number of unknown variables and thus the CPU cost.

The second class of methods for obtaining IP3 is based on analytical approaches which are mostly based on Volterra functional series [7]. Such methods have the advantage of providing some insight into the operations of the circuit such as the impact of one particular component on linearity. However, such methods are practically unusable except for very specific cases. This is due to the fact that the computation of the Volterra kernels requires complex analytical manipulations which are extremely difficult to perform except for very simple circuits and are impossible to automate [8], [9].

In this thesis a new method is proposed for calculating the IP3 of a circuit. The proposed approach is based on the automated computation of the circuit moments and is equivalent to numerically computing the values of the appropriate Volterra kernels at the frequency of interest. This approach does not require any analytical manipulation but is rather applied directly to the Modified Nodal Analysis (MNA) [10] formulation of the circuit. It can therefore be applied to circuits of arbitrary complexity. Furthermore, the computation of all the moments only requires one LU decomposition of the Jacobian evaluated at the DC point which is very sparse unlike the typical Harmonic Balance Jacobian which is usually both large and dense.



Figure 1.1: Receiver's RF interface

Finally, the computation is done numerically with the input frequencies known, and thus produces very accurate results.

### **1.2 Contributions**

In this thesis, a new efficient method for obtaining the IP3 of a circuit that was developed and tested, is presented. In order to achieve this goal, a systematic automated algorithm for computing the system moments was implemented and analytical relations between the moments and the Volterra kernels were derived. The new method was tested on common circuit topologies and was shown to be significantly faster and less memory intensive than traditional simulation based techniques.

### **1.3** Organization of the Thesis

After the Introduction in the current chapter, a background study of steady-state simulation methods in addition to a detailed description of the Harmonic Balance method is presented in Chapter 2. In Chapter 3, a description of distortion analysis methods and techniques in RF circuits in addition to definitions are given. The for-

mulation, derivation and explanation of the proposed method is presented in Chapter 4 along with a couple of examples to demonstrate the accuracy and speedup of the method. Conclusions and proposed further research are given in Chapter 5.

# Chapter 2

## **Simulation of RF Circuits**

### 2.1 Introduction

The main design specifications for Radio Frequency (RF) circuits are centered around gain, power, intermodulation distortion, noise and frequency bandwidth [1]. Such figures of merit require the computation of the frequency domain response. In other words the steady state response (after all the transients have died out) of the circuit must be computed. One way to obtain the frequency response is by using small signal analysis, or in other words by linearizing the circuit around the DC operating point using small signal analysis and using phasor type analysis to obtain the frequency response. This approximation is, however, not sufficiently accurate for the analysis of RF circuits especially when computing the intermodulation distortion. In such a case, the nonlinear steady state response is required. The steady-state response of a nonlinear circuit is the response after all the transients have died out. For circuits with constant inputs, the steady-state response is simply the DC response. When the input is periodic, the output in the steady-state, is also periodic with the same period as the input. The brute force way to obtain the steady state response is to perform a long transient analysis until all the transients die out. Such an approach is, however, very inefficient due to the very large number of time steps required, and suffers from several limitations such as the inability to determine exactly when the transients completely die out in addition to the resulting Fourier

transform noise.

A number of algorithms have been proposed in the literature [4] in order to directly solve the steady state response accurately and efficiently. These include time domain methods such as the Shooting method [1], [11], [12] which converts the problem into a boundary value problem (BVP), and frequency domain methods such as the Harmonic Balance approach [1], [6], [13].

In this chapter, a brief survey of these methods is presented with a focus on the Harmonic Balance approach because the algorithm presented in this thesis is based on the HB formulation.

### 2.2 Modified Nodal Analysis Formulation

Before addressing the various steady state simulation methods, the formulation of the Modified Nodal Analysis (MNA) [10] circuit equations is described in this section. These simply correspond to the summation of currents according to Kirchoff's Current Law (KCL) at the circuit nodes. The MNA formulation provides a simple and general approach that allows for the automated representation of circuits containing both linear and nonlinear elements as a set of equations in matrix form.

Consider a non-linear circuit excited by one or more input tones. The MNA circuit equations can be expressed in the time domain as

$$Gx(t) + C\dot{x}(t) + f(x) = b(t), \qquad (2.1)$$

where

- $\boldsymbol{x}(t) \in \Re^n$  is the vector of n unknown voltages and currents,
- $G \in \Re^{n \times n}$  is the matrix that contains the circuit stamps of the linear memoryless elements,
- $C \in \Re^{n \times n}$  is the matrix that contains the circuit stamps of the linear memory elements,



Figure 2.1: Simple Nonlinear Circuit

•  $f(x) \in \Re^n$  is a vector of nonlinear scalar functions of the form

$$\boldsymbol{f}(\boldsymbol{x}(t)) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_N(\boldsymbol{x})]^T$$
(2.2)

- b(t) is a vector that contains the independent input sources.
- *n* is the total number of variables in the circuit equations.

As an example, consider the simple non-linear circuit shown in Fig. 2.1. The MNA equations for this circuit are:

$$\begin{bmatrix} g & -g & 1 \\ -g & g & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_E \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{I}_E \end{bmatrix} + \begin{bmatrix} 0 \\ I_s(e^{V_2/V_t} - 1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_{in} \end{bmatrix}$$
(2.3)

Note that one of the key advantages of MNA formulation is that these MNA systems of equations can be automatically generated by a computer with the use of component signatures, or stamps [14], for each circuit element. Several stamps for common circuit elements can be found in [15]. Another advantage is that the resulting system of equations is typically very sparse.



Figure 2.2: a) Solution based on Transient Analysis b) Steady-State analysis using Shooting method

### 2.3 Shooting Methods

The shooting method [1], [11], [12] is a time domain method for finding the steady state response of a nonlinear circuit excited with a periodic input. Standard circuit simulators such as SPICE solve the initial value problem by integrating the differential algebraic equations (DAEs) representing the circuit in the time domain from a known initial condition x(0). This results in the transient response of the circuit as shown in Fig. 2.2 a). In order to directly find the steady state response (also called the periodic solution) the shooting method relies on the fact that for a periodic input, the output in the steady state is also periodic and recasts the problem as a boundary value problem. Thus, in the new formulation the initial condition is an unknown and it is computed subject to the boundary condition x(T) = x(0) [16] where T is the period as is illustrated in Fig. 2.2 b).

By representing x(T) as x(T, x(0)) (since it is a function of x(0)) the function that must be solved to satisfy the boundary condition is expressed as:

$$x(0) - x(T, x(0)) = 0$$
(2.4)

This expression is then solved using Newton Iteration [11], [16], or using other iterative methods such as conjugate gradient [17] or Krylov techniques [18].

There are two problems that arise when trying to apply this method to analog and microwave circuits. Firstly, shooting methods find the periodic steady-state response of a circuit by assuming that the periodicity constraint of x(t) = x(t+T) for all t holds as a two-point boundary-value constraint [19]. However, the steady-state response of a mixer circuit for example is, in general, not periodic but rather quasiperiodic which makes the method inappropriate for such circuits which has led to the development of envelope techniques [4] which extend the applicability of this method. The second problem is that being a time domain method, it has difficulties with distributed elements such as transmission lines which are best described in the frequency domain [20].

### **2.4 Harmonic Balance Method**

The Harmonic Balance (HB) method [1], [6], [13] is a technique for obtaining the steady-state response due to a periodic input directly in the frequency domain. A number of variations on this approach designed to improve the CPU efficiency have been proposed such as piecewise Harmonic Balance [6], [13], domain decomposition [21], Harmonic Balance using inexact Newton [22], and model order reduction based methods [23]. All these models are based on the one fundamental concept behind Harmonic Balance, that is, given a periodic input, the steady-state output will also be periodic and can thus be expressed as a Fourier series. The HB algorithm reformulates the system of nonlinear differential algebraic equations into a system of nonlinear algebraic equations where the unknowns are the Fourier coefficients. In this section the HB approach is explained in detail since the formulation of the HB equations is used as a basis for the proposed algorithm in this thesis.

#### 2.4.1 HB Formulation

Consider a circuit containing linear and nonlinear elements described by its MNA equations as shown in (2.1). Given a periodic input b(t), the response is also known to be periodic in the steady-state. The Harmonic Balance approach expresses the periodic solution as a truncated series of sine and cosine functions at the frequencies of the harmonics of the inputs as well as the intermodulation products. For a sin-

gle tone simulation, the Harmonic Balance solution vector simplifies to a truncated Fourier Series including the first H harmonics since there would be no intermodulation frequencies present. In general, the solution vector can then be represented as

$$\boldsymbol{x}(t) = \boldsymbol{A}_0 + \sum_{k=1}^{H} (\boldsymbol{A}_k \cos(\omega_k t) + \boldsymbol{B}_k \sin(\omega_k t))$$
(2.5)

where  $\omega_k$  are the harmonic and intermodulation frequencies present in the circuit,  $A_0 \in \Re^n$  is a vector containing all the DC amplitudes of the *n* variables,  $A_k \in \Re^n$ is a vector containing all the amplitudes of the cosine terms at frequency  $\omega_k$  and  $B_k \in \Re^n$  is a vector containing all the amplitudes of the sine terms at frequency  $\omega_k$ . Substituting this expression into (2.1) and equating the coefficients of the sine and cosine terms results in a set of nonlinear algebraic equations in the form of

$$\bar{\boldsymbol{G}}\boldsymbol{X} + \bar{\boldsymbol{C}}\boldsymbol{X} + \boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{B}_{\boldsymbol{dc}} + \boldsymbol{B}_{\boldsymbol{ac}}, \qquad (2.6)$$

where

- X ∈ ℜ<sup>N<sub>h</sub></sup> is a vector of unknown cosine and sine coefficients for each of the variables in x(t).
- $B_{dc} \in \Re^{N_h}$  and  $B_{ac} \in \Re^{N_h}$  represent the contributions of the DC and AC independent sources respectively.
- $\bar{G} \in \Re^{N_h \times N_h}$  is a block matrix  $\bar{G} = [G_{ij}]$  representing the contribution of the linear memoryless elements of the network to the frequency components. The blocks  $G_{ij} \in \Re^{N_b \times N_b}$  are diagonal matrices given by

$$\boldsymbol{G}_{ij} = \operatorname{diag}(g_{ij}, \cdots, g_{ij}) \tag{2.7}$$

with  $g_{ij}$  being the corresponding entry in the G matrix in (2.1).

•  $\bar{C} \in \Re^{N_h \times N_h}$  is a block matrix  $\bar{C} = [C_{ij}]$  representing the contribution of the linear memory elements of the network to the frequency components. The blocks  $C_{ij} \in \Re^{N_b \times N_b}$  are diagonal matrices given by

$$\boldsymbol{C}_{ij} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \omega_1 & \cdots & 0 & 0 \\ 0 & -\omega_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \omega_k \\ 0 & 0 & 0 & \cdots & -\omega_k & 0 \end{bmatrix}$$
(2.8)

with  $c_{ij}$  being the corresponding entry in the C matrix in (2.1).

F(X) contains the sine and cosine coefficients of the nonlinear vector f(x) defined in (2.1) and is expressed as

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} F_1(\boldsymbol{X}) \\ F_2(\boldsymbol{X}) \\ \vdots \\ F_N(\boldsymbol{X}) \end{bmatrix}$$
(2.9)

- N<sub>b</sub> is the expanded number of variables for each original unknown variable in x(t) and is equal to 2H + 1.
- $N_h$  is total number of variables for the HB system of equations.  $N_h = n \cdot N_b = n(2H + 1)$ . This quantity is usually very large even for modest size systems.

The relation between the vectors X, f(x(t)) and F(X) is established through the use of the Fast Fourier Transform (FFT) and the Inverse Fast Fourier Transform (IFFT). These transforms are utilized while performing Newton Iterations to obtain the solution of the HB equations and are explained in the following section.

To illustrate how much bigger the HB matrices become relative to the original system, the simple 2-node circuit shown in Fig. 2.1 is taken as an example. Taking only 2 Harmonics (*H*=2), then  $\bar{G}$  becomes a 15×15 matrix, as seen in equation (2.10). If 32 harmonics are needed for example, then the size of the matrix becomes 195×195.



(2.10)

This dramatic increase in size presents a significant CPU cost problem. Despite the fact that these matrices ( $\bar{G}$  and  $\bar{C}$ ) are sparse, this is not the case with the HB Jacobian matrix which is usually very dense, and has to be manipulated at each Newton Iteration, as will be explained more clearly in the following sections.

### 2.4.2 Solution of HB Equations

The solution to the set of nonlinear algebraic HB equations where the unknowns are the Fourier coefficients of the steady state solution, is obtained by applying iterative numerical techniques such as Newton Iteration or the Conjugate Gradient method. In this chapter, Newton Iteration [24], [25] which benefits from quadratic convergence near the solution, will be presented.

In order to apply the Newton Raphson iteration, the HB equations in (2.6) are reformulated as

$$\Phi(\mathbf{X}) = \bar{\mathbf{G}}\mathbf{X} + \bar{\mathbf{C}}\mathbf{X} + \mathbf{F}(\mathbf{X}) - \mathbf{B} = 0$$
(2.11)

where  $\Phi(X)$  is referred to as the objective function. In this expression, the terms  $B_{ac}$  and  $B_{dc}$  have been combined into one vector such that  $B = B_{ac} + B_{dc}$ . The target solution vector X is found iteratively using Newton Raphson Iteration by starting with an initial guess and then updating the solution at each iteration until convergence occurs. At each iteration, the solution vector is updated using

$$\boldsymbol{X}^{(i+1)} = \boldsymbol{X}^{(i)} - \boldsymbol{J}_{HB}(\boldsymbol{X}^{(i)})^{-1} \boldsymbol{\Phi}(\boldsymbol{X}^{(i)})$$
(2.12)

where *i* is the iteration number,  $X^{(i)}$  is the old guess and  $J_{HB}(X^{(i)})$  is the Jacobian Matrix defined as

$$\boldsymbol{J}_{HB}(\boldsymbol{X}^{(i)}) = \frac{\partial \boldsymbol{\Phi}(\boldsymbol{X})}{\partial \boldsymbol{X}} |_{\boldsymbol{X}^{(i)}} = \bar{\boldsymbol{G}} + \bar{\boldsymbol{C}} + \frac{\partial \boldsymbol{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} |_{\boldsymbol{X}^{(i)}}$$
(2.13)

To begin the process of solving the system, a good initial guess for the solution vector must be selected and used in (2.12) in the first iteration. A good initial guess is one that is close to the solution vector, which will lead to faster convergence and therefore result in lower CPU cost. As shown in the flowchart that highlights the main steps of the Newton Raphson (NR) algorithm in Fig. 2.3, once an initial guess is selected, the next step is to compute the objective function using the current value of the vector X which includes the evaluation of the nonlinear vector F(X). The Jacobian matrix must then also be evaluated using the current value of the vector X which is then used along with  $\Phi(X_{old})$  to determine the error  $\Delta X$ . This error is then added to  $X_{old}$  to obtain the new updated solution vector  $X_{new}$ . If the error between  $X_{new}$  and  $X_{old}$  is large, then the latest value of  $\kappa$ , in which case

the solution would have converged to a value and the iteration loop is stopped.

#### 2.4.3 Computation of Objective Function

In this section, the procedure for evaluating the value of the objective function,  $\Phi(\mathbf{X})$ , defined in (2.11) is explained. Specifically, it is the evaluation of the nonlinear vector  $F(\mathbf{X})$  which is of particular interest. The remaining terms in (2.11) are expansions of the MNA equations into block matrix forms as highlighted earlier.

To evaluate the vector F(X) given by (2.9), we need to make use of the Direct Fourier Transform (DFT). Thus, an overview of the DFT is provided before proceeding with the evaluation.

#### **Direct Fourier Transform**

In this section, a brief outline of the Direct Fourier Transform (DFT) is presented. The Fourier Transform is used in the calculation of the objective function and the Jacobian matrix at each Newton iteration of the HB algorithm. Consider a periodic signal x(t) with a period  $T = \frac{2\pi}{\omega}$  expressed as a Fourier Series given by

$$\boldsymbol{x}(t_n) = a_0 + \sum_{k=1}^{H} (a_k \cos(k\omega t_n) + b_k \sin(k\omega t_n))$$
(2.14)

This signal is then sampled at  $N_b$  time points  $[t_0, t_1, \dots, t_{N_b-1}]$  that are equally spaced across the interval [0, T] with

$$t_n = n \frac{T}{N_b}; n = 0, 1, \cdots, N_b - 1$$
 (2.15)



Figure 2.3: Newton Raphson Iteration Flowchart

The Fourier Series expressions at each sampled time point results in

$$\boldsymbol{x}(t_0) = a_0 + \sum_{k=1}^{H} (a_k \cos(k\omega t_0) + b_k \sin(k\omega t_0))$$
(2.16)

$$\boldsymbol{x}(t_1) = a_0 + \sum_{k=1}^{H} (a_k \cos(k\omega t_1) + b_k \sin(k\omega t_1))$$
(2.17)

$$\boldsymbol{x}(t_{N_{b}-1}) = a_{0} + \sum_{k=1}^{n} (a_{k} \cos(k\omega t_{N_{b}-1}) + b_{k} \sin(k\omega t_{N_{b}-1}))$$
(2.19)

which can be re-written in matrix form as

$$\begin{bmatrix} x(t_0) \\ x(t_1) \\ \vdots \\ x(t_{N_{b}-1}) \end{bmatrix} = \begin{bmatrix} 1 & \cos(\omega t_0) & \sin(\omega t_0) & \cdots & \cos(H\omega t_0) & \sin(H\omega t_0) \\ 1 & \cos(\omega t_1) & \sin(\omega t_1) & \cdots & \cos(H\omega t_1) & \sin(H\omega t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 \cos(\omega t_{N_{b}-1}) \sin(\omega t_{N_{b}-1}) \cdots \cos(H\omega t_{N_{b}-1}) \sin(H\omega t_{N_{b}-1}) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_H \\ b_H \end{bmatrix}$$

This relation is that of the Inverse Direct Fourier Transform (IDFT), with the times samples obtained by multiplying the vector of Fourier coefficients with the IDFT matrix, which will be referred to as  $\Gamma$ . Similarly the DFT can be performed with the use of  $\Gamma^{-1}$ . The arguments in matrix  $\Gamma$  of the form  $k\omega t_n$  can be re-written as

$$k\omega t_n = k\left(\frac{2\pi}{T}\right)n\left(\frac{T}{N_b}\right) = kn\left(\frac{2\pi}{N_b}\right)$$
(2.20)

which shows that the arguments are independent of frequency. In fact, matrix  $\Gamma$  can be expressed independently of frequency as

$$\boldsymbol{\Gamma} = \begin{bmatrix} 1 & \cos(\Theta_{0,1}) & \sin(\Theta_{0,1}) & \cdots & \cos(\Theta_{0,H}) & \sin(\Theta_{0,H}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\Theta_{n,1}) & \sin(\Theta_{n,1}) & \cdots & \cos(\Theta_{n,H}) & \sin(\Theta_{n,H}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\Theta_{N_b-1,1}) & \sin(\Theta_{N_b-1,1}) & \cdots & \cos(\Theta_{N_b-1,H}) & \sin(\Theta_{N_b-1,H}) \end{bmatrix}$$

with

$$\Theta_{n,k} = kn\left(\frac{2\pi}{N_b}\right) \tag{2.21}$$

#### Evaluation of the Nonlinear Vector F(X)

To simplify matters, only one nonlinear scalar function from the vector f(x) shown in (2.2) is considered. The evaluation of the other functions is done in a similar fashion. We also assume that f(x) is a function of one variable  $x_1$ .

To evaluate the nonlinear vector  $F(X_1)$ , as a function of  $X_1$ , which is the vector containing the sine and cosine coefficients of  $x_1(t)$ , we use the following relation

$$\boldsymbol{F}(\boldsymbol{X}_1) = \boldsymbol{\Gamma}^{-1} \boldsymbol{F}_s \tag{2.22}$$

with  $F_s$  being the vector that contains the time samples of f(x(t)) and is given by  $F_s = [f(x_0), f(x_1), \dots, f(x_{N_h})]$  and  $\Gamma$  being the inverse DFT matrix. The vector of time samples  $X_s = [x_0, x_1, \dots, x_{N_h}]$  needed to determine  $F_s$  is also evaluated using the DFT, namely

$$\boldsymbol{X}_s = \boldsymbol{\Gamma} \boldsymbol{X}_1 \tag{2.23}$$

#### 2.4.4 Harmonic Balance Jacobian

The computation, storage and inversion of the HB Jacobian matrix at each iteration of the Newton Raphson algorithm constitutes the bulk of the CPU cost for obtaining the HB solution as this matrix is usually very large and dense, especially when simulating circuits with multi-tone inputs. The number of iterations needed to obtain the final solution vector also varies and can be quite large, therefore requiring the expensive process of manipulating the Jacobian matrix to be repeated several times.

The HB Jacobain matrix is evaluated using the expression shown in (2.13). The matrices  $\bar{G}$  and  $\bar{C}$  are the same as those defined in (2.6). The remaining term is found using

$$\frac{\partial F(X)}{\partial X} = \begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \cdots & \frac{\partial F_1}{\partial X_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial X_1} & \cdots & \frac{\partial F_n}{\partial X_n} \end{bmatrix}$$
(2.24)

where  $F_n$  and  $X_n$  contain the Fourier coefficients of  $f_n$  and  $x_n$  respectively. Each of the terms in (2.24) is a matrix in itself, forming a block matrix structure. Each term is evaluated using

$$\frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{X}_n} = \Gamma^{-1} \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{X}_n} = \Gamma^{-1} \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{X}_s} \frac{\partial \boldsymbol{X}_s}{\partial \boldsymbol{X}_n} = \Gamma^{-1} \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{X}_s} \Gamma$$
(2.25)

$$\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{X}_s} = \operatorname{diag}[\frac{df(x(t_0))}{dx(t_0)}, \cdots, \frac{df(x(t_{N_b-1}))}{dx(t_{N_b-1})}]$$
(2.26)

where  $\Gamma$  is the DFT matrix. In practice, the multiplication by  $\Gamma$  and  $\Gamma^{-1}$  is not done, and is replaced by the FFT and IFFT algorithms.

Although  $\frac{\partial F_s}{\partial X_s}$  is a diagonal matrix, the multiplication by  $\Gamma$  and  $\Gamma^{-1}$  makes  $\frac{\partial F_n}{\partial X_n}$  a full matrix, which then makes the Jacobian matrix more dense. A comparison of the sparsity patterns between the HB Jacobian matrix and the Jacobian matrix used in the proposed method is shown for each of the examples given in this thesis.

#### 2.4.5 Fourier Transform for Almost Periodic Input Signals

When a circuit is excited with an input signal that contains multiple tones, the signal is usually quasi-periodic, i.e. the two input frequencies are non-commensurate, and are therefore not multiples of each other [15]. However, these frequency values now create problems when applying the Fourier Transforms and also when calculating the Jacobian matrix [15]. A simple and efficient way to address this problem is

through the use of frequency mapping techniques presented in [1] which are known as the Diamond and Block Truncation methods. These methods are used to map actual harmonic frequencies (including intermodulation terms) to arbitrary artificial frequencies. In particular, the fundamentals of the input signal are chosen to be multiples of some arbitrary frequency so that the resulting signals will be periodic and can also be expressed as shown in (2.5). This trick is best illustrated with an example. For simplicity, only circuits that are excited with two input tones are considered, although this analysis could also be extended for the case of greater input tones. Consider a nonlinear circuit with an input-output relation as follows

$$f(v(t)) = v(t)^2$$
 (2.27)

Assuming that the input signal consists of a voltage signal with two input tones as follows

$$v(t) = A\cos(\alpha t) + B\cos(\beta t)$$
(2.28)

Substituting (2.28) into (2.27) and expanding using trigonometric identities the output function becomes

$$f(v(t)) = \frac{A}{2} + \frac{B}{2} + \frac{A}{2}\cos(2\alpha t) + \frac{B}{2}\cos(2\beta t) + \frac{AB}{2}\cos(\alpha t + \beta t) + \frac{AB}{2}\cos(\alpha t - \beta t)$$
(2.29)

It is important to note how the coefficients of the cosine terms are independent of the actual values of the frequencies  $\alpha$  and  $\beta$ , therefore allowing the possibility of mapping these frequencies to convenient values.

#### **Block Truncation**

The Block Truncation algorithm is the first of the two frequency mapping algorithms presented and is applied when the two frequencies  $\omega_1$  and  $\omega_2$  are well spaced out from each other on the frequency spectrum i.e.  $\omega_1 >> \omega_2$ . Considering the set


Figure 2.4: Frequency mapping using block truncations [1]

of frequencies given by

$$\omega_k = k_1 \omega_1 + k_2 \omega_2; 0 \le k_1 \le H_1, |k_2| \le H_2, k_1 \ne 0 \text{ if } k_2 < 0 \tag{2.30}$$

where  $H_1$  and  $H_2$  are the number of harmonics of  $\omega_1$  and  $\omega_2$  respectively. The new artificial set of frequencies that are equally spaced and do not overlap is given by  $k\omega = \alpha_1 k_1 \omega_1 + \alpha_2 k_2 \omega_2$ , with the scaling factors for the two fundamental frequencies ( $\alpha_1$  and  $\alpha_2$ ) being

$$\alpha_1 = 1; \alpha_2 = \frac{\omega_1}{\omega_2(2H_2 + 1)} \tag{2.31}$$

The mapping of the quasiperiodic frequencies to periodic frequencies using Block truncations is shown in Fig. 2.4.

#### **Diamond Truncation**

For the more frequent case in RF circuits where the two fundamental frequencies of the input signal are very close to each other on the output spectrum, known as almost periodic signals, the Diamond Truncation algorithm is used. Considering the set of frequencies given by

$$\omega_k = k_1 \omega_1 + k_2 \omega_2; |k_1| + |k_2| \le H, k_1 + k_2 \ge 0, k_1 \ne k_2 \text{ if } k_2 > 0 \qquad (2.32)$$



Figure 2.5: Frequency mapping using diamond truncations [1]

where H is the highest order of harmonics of  $\omega_1$  and  $\omega_2$  that is accounted for. The new artificial set of frequencies that are equally spaced and do not overlap is given by  $k\omega = \alpha_1 k_1 \omega_1 + \alpha_2 k_2 \omega_2$ , with the scaling factors for the two fundamental frequencies ( $\alpha_1$  and  $\alpha_2$ ) being

$$\alpha_1 = 1; \alpha_2 = \frac{H\omega_1}{\omega_2(H+1)}$$
(2.33)

The mapping of the quasiperiodic frequencies to periodic frequencies using Block truncations is shown in Fig. 2.5.

#### 2.4.6 Convergence

The Harmonic Balance method for steady state analysis comes with significant problems. Firstly there is the problem of convergence. Since it relies on numerical iterations to achieve the solutions, a good initial guess is required for two main reasons, the first is to reduce the number of iterations by choosing an initial guess that is closer to the solution, and the second is to improve the chances of converging to the correct solution. Although Newton Iteration benefits from quadratic convergence near the solution, convergence is not guaranteed. A number of techniques such as Gauss-Jacobi Newton harmonic relaxation [6] and inexact Newton Iteration [22] have been presented to improve convergence. The main challenge is that there are no rules for selecting a good initial guess. Continuation or homotopy



Figure 2.6: Continuation Example

methods [26]–[28] have been introduced to circuit simulation to address the issue of convergence in locally convergent iterative methods such as Newton Iteration. The main idea behind continuation methods is to augment the system of equations  $\Phi(\mathbf{X})$  shown in (2.11) with a new variable  $\beta$  as a continuation parameter to form a new set of equations  $\Psi(\mathbf{X}, \beta)$  with a trivial solution when  $\beta = 0$ , and the solution of the original system when  $\beta = 1$ , such that  $\Psi(\mathbf{X}, 1) = \Phi(\mathbf{X})$ . The continuation parameter is swept from 0 to 1 using a discrete number of values, and the solution at each value is obtained and tracked, such that the solution at each value is used as the initial guess for the obtaining the next solution. To illustrate the use of continuation methods let us consider an example of a nonlinear DC circuit excited by a 5V DC source as shown in the diagram of Fig. 2.6. Continuation is applied to this problem by ramping up the value of the input voltage source linearly from 0 to 5V. The solution for the first point when the source is 0 volts is trivial. For the intermediate steps between 0 and 5, the solution to the previous step is taken as the initial guess, and the solution at 5V is the solution to the original problem.

## 2.5 Conclusion

The main limitation of the HB method is the size and the typically dense structure of the Jacobian matrix, as was shown in the previous section, which must be evaluated and inverted at each iteration. The number of variables,  $N_h$ , of an HB system easily becomes very large due to the large number of harmonics and intermodulation products present in the case of multi tone inputs. If the system is solved using H harmonics, then the value of  $N_h$  is equal to n(2H + 1). Furthermore, the dense Jacobian results in a large CPU cost in the computation of the system solution, and thus, the value of IP3. The methods that have been mentioned in the previous section for improving convergence also add to the CPU cost of obtaining the solution, therefore further amplifying the problem.

The proposed method that is presented in Chapter 4 of this thesis addresses these limitations directly. Newton Iteration is no longer required for obtaining the solution, and despite the systems being of similar size, the Jacobian matrix used in the proposed method is very sparse and static, therefore it needs only be computed and factorized once.

Now that an overview of steady-state simulation has been presented in addition to a detailed analysis of the Harmonic Balance method, the next step is to discuss distortion in RF circuits and the existing methods that are currently used to analyze it. This is the topic of the next chapter.

24

# Chapter 3

# **Distortion Analysis of RF Circuits**

Nonlinear distortion is an important bottleneck which limits the performance of Radio Frequency (RF) communication circuits. It is thus an important requirement which must be accounted for during the design process. In this chapter, nonlinear distortion is introduced along with its associated figures of merit. In addition, the main techniques used for calculating these figures of merit are introduced.

## **3.1** Linear Versus Nonlinear Distortion

Distortion is defined in general as the deviation of the output signal from the desired waveform [29]. There are many types of distortion resulting from non-ideal circuit behavior. These can be divided into two main classes: linear and non-linear distortion. Linear distortion results from a non-ideal frequency response. For example, consider a linear amplifier that has the frequency response shown in Fig. 3.1. If a square wave signal with a frequency  $f_2$ , that is smaller than the corner frequency  $f_1$ , is applied at the input of this amplifier, the response will be distorted as shown in Fig. 3.2. The output will deviate considerably from a square wave even though the circuit is assumed to contain only linear elements. This distortion can be explained through the use of a Fourier Series. Since a square wave consists of a summation of cosine and sine signals at  $f_2$  and the harmonics of  $f_2$  ( $2f_2$ ,  $3f_2$ , etc.), the gain at the higher harmonics is lower than the lower ones due to the frequency response



Figure 3.1: Absolute Value of Frequency Response of a Linear Amplifier

and thus the shape of the output waveform is distorted. Another example of linear distortion is dispersive effects on lossy transmission lines where different frequency components of a signal travel at different speeds. Although linear distortion affects signals with complex spectral distributions such as square waves, a monotone signal containing only one frequency is not impacted. The second class of distortion, nonlinear distortion, affects any type of signal, including single tone sinusoids. One example of nonlinear distortion is crossover distortion at the output of a Class B amplifier [30]. In general, any nonlinearity in a circuit causes the output to contain the harmonics of the input signal as well as the fundamental tone. This type of nonlinear distortion is typically characterized in the frequency domain using figures of merit such as the 3rd order intercept point (IP3) and is discussed in detail in the remainder of this chapter.

# 3.2 Importance of Nonlinear Distortion

Consider the diagram shown in Fig. 3.3. When a nonlinear circuit is excited with one sinusoidal source at a frequency  $\omega_1$  with an amplitude that is small enough, then the output spectrum contains one frequency component above the noise floor



Figure 3.2: Response of Amplifier to Square Wave Input



Figure 3.3: Concept of Nonlinear Distortion

which is the response corresponding to the circuit's linear behavior. This frequency is the same as that of the input signal and is known as the fundamental frequency. Under these conditions the circuit is essentially linear and can be analyzed using small signal models. If the input amplitude is increased, the small signal model is no longer valid and nonlinearities must be accounted for. The output signal will thus contain signals at multiples of the fundamental frequency, which will cause the output signal to be distorted. The *nth* harmonic is the term given to the signal at  $n\omega_1$  of the output spectrum. The linear response is typically the wanted response while the harmonics are usually not desired.

When a nonlinear circuit is excited with two sinusoids at two frequencies ( $\omega_1$  and  $\omega_2$ ) that are both applied at the same input port, then the output spectrum will contain harmonics at combinations of the two frequencies ( $|n\omega_1 \pm m\omega_2|$ ) in addition to the harmonics at multiples of the input frequencies, as shown in Fig. 3.4.



Figure 3.4: Output Spectrum of Nonlinear circuit with 2 frequency excitation

These additional signals in the output spectrum are known as intermodulation products and also result in distortion.

In most RF systems, the unwanted harmonics can be removed by passing the output signal through a band-pass filter. The problem arises when the two frequencies of interest ( $\omega_1$  and  $\omega_2$ ) are situated close to each other, which results in the 3rd order intermodulation tones at  $|2\omega_1 \pm \omega_2|$  also being situated very close to the fundamental tones, thus requiring filters of very high orders to remove. For this reason, the accurate measurement of distortion due to these 3rd order intermodulation terms is of particular interest to RF circuit designers.

# 3.3 Terminology and Figures of Merit

Once distortion is present at the output spectrum of an RFIC, engineers then rely on computing and measuring several well known parameters to analyze the impact of it on circuit performance. In this section, a brief description of these parameters is given. The diagram in Fig. 3.5 shows the wanted response at the fundamental frequency of  $\omega_1$  and one of the unwanted third-order intermodulation (IM3) products at  $|2\omega_1 - \omega_2|$ .



Figure 3.5: Illustration of definitions of distortion parameters

• Intercept Points

When the power of the input signal is swept starting from zero, the output signal at the fundamental frequency and the output at  $|2\omega_1 - \omega_2|$  will increase linearly. The amplitude of the IM3 response increases with a slope of 3 on a dB scale while the amplitude at the fundamental will increase with a slope of 1 on a dB scale. This linear increase will continue until the output begins to compress. However, if the extrapolation of the linear increases is made, the point at which the two responses meet is defined as the intercept point. In this case, since it's the intersection of the fundamental with the IM3 tone, it is the 3rd order intercept point (IP3). If a second order intermodulation product is considered, then a second order intercept point (IP2) can be defined in a similar way to IP3. The input IP3 power level is referred to as IIP3 whereas the output power level is referred to as OIP3.

• Total Harmonic Distortion

Total Harmonic Distortion (THD) is a measure that indicates how closely the output waveform resembles a pure sine wave. It is calculated by measuring the amount of energy in the harmonics relative to the energy in the fundamental, using the following expression:

$$THD = \sqrt{\frac{\sum_{n=2}^{\infty} |V_{out,n}|^2}{|V_{out,1}|^2}}$$
(3.1)

where  $|V_{out,n,0}|$  is the amplitude of the *nth* harmonic

• Intermodulation Free Dynamic Range

The intermodulation free dynamic range (IMFDR) is the ratio of the largest and the smallest signal levels the circuit can handle without the appearance of an intermodulation component. The diagram in Fig. 3.5 illustrates the IMFDR for the 3rd order intermodulation component.

• Gain Compression and Expansion

At higher amplitudes of input power, higher-order odd numbered nonlinear components give rise to components at the fundamental frequency. For example, the third order nonlinearity gives rise to a component at the fundamental frequency which increases with the 3rd power of the input amplitude. As a result, the fundamental response no longer increases linearly. The increase therefore becomes faster, and this leads to Gain Expansion. If the 3rd order contribution is negative, the increase becomes slower and this leads to Gain Compression.

• 1dB Compression Point

When the power of the input signal is swept starting from zero, the output signal at the fundamental is observed to increase linearly with the input amplitude at first, until it reaches a point where the gain compression occurs and the output signal starts to exhibit compression as shown in Fig. 3.5. The 1dB compression point is defined as the input level for which the fundamental response is 1dB lower than the extrapolation of the linear response.

# 3.4 Distortion Analysis Methods

In the literature, there are several methods that have been employed to analyze distortion. There are two classes of techniques for analyzing the distortion of a circuit. The first is based on analytical approaches such as the Volterra functional series [7]. Such methods, while they provide useful analytical insight into the operation of the circuit, are typically very complex for practical applications [8], [9]. The other class of methods are simulation based and consist of algorithms that are employed by simulators to try and measure distortion in the most efficient way possible. In this section, a brief overview of some of the methods available is presented, with an emphasis on Volterra Series which will be used later.

#### 3.4.1 Distortion Analysis using Harmonic Balance

The Harmonic Balance method for obtaining the steady-state solution of a system has already been presented in the previous chapter. In this section, a brief description of how to use the HB method in analyzing distortion is presented.

The solution vector of the HB system of equations consists of the sine and cosine components at each frequency on the output spectrum of the signal. The values of these components are then used to determine any one of the several figures of merit of distortion such as the 1dB compression point and IP3. To determine the value of IP3, the steady-state solution for a circuit with power gain G, is obtained using the HB method, with the input power  $(P_i)$  and the output powers at the fundamental  $(P_1)$  and IM3  $(P_3)$  frequencies noted. Since power division amounts to subtraction on a log scale, the gain of the circuit can be expressed as

$$G = OIP3 - IIP3 = P_1 - P_i$$
(3.2)

This relation is then used to determine the value of IIP3 knowing that the slope of the fundamental power is 1 on a dB scale, and that of the IM3 power is 3 on a dB scale. These relations are solved to give the following expression which is used to determine the value of IIP3 [31]

$$IIP3 = P_1 + \frac{1}{2}[P_1 - P_3] - G$$
 (3.3)

$$= P_i + \frac{1}{2}[P_1 - P_3] \tag{3.4}$$

This concept is illustrated in the diagram of Fig. 3.5. This method is later used as a basis for comparison with the results of the proposed method presented in this thesis.

#### **3.4.2** Linear Centric Models

One of the most recent methods for simulation-based distortion analysis employs linear centric circuit models for the nonlinear elements to determine the individual distortion contributions. This method is presented in detail in [2], [32]. The linear centric models approach makes it possible to solve for each distortion contribution via straight-forward device evaluations and one linear circuit solution in the post evaluation step. It is motivated by the successive chord method, which is an iterative method, in which constant linearizations of the nonlinear elements are used to construct the Jacobian matrix which in turn is used for determining the non-linear system solution [33], [34]. This Jacobian matrix remains constant and is used in all iterations thus reducing CPU cost.

The first step in this approach involves replacing each non-linear circuit element by its linear centric model. The new circuit diagram will then resemble that shown in Figure 3.6. This linear-centric model can be used to perform distortion analysis with any steady-state simulation method as a post-simulation processing step. A good example of this is in conjunction with the Harmonic Balance method. The advantages of using this method are firstly that no higher order derivatives are required to be evaluated explicitly as is the case for computing the Volterra kernels when performing a Volterra Series analysis (presented later). Therefore, this method can be applied directly when given the device models of the nonlinear elements in the system. In addition, this method presents low post-simulation CPU cost since



Figure 3.6: Overall linear-centric circuit model [2]

only one linear system solution needs to be evaluated and the rest of the computations are device model evaluations. The main disadvantage in using this method is that it cannot distinguish distortion contributions due to different orders of nonlinearities in a non-linear element, which can be done using Volterra Series Analysis. Thus, only the sum of all the distortion contributions of all orders per nonlinearity is extracted. Also, linear centric device models must be derived for each nonlinear element in the circuit which also presents the need to distinguish between linear and nonlinear elements before any computations are performed. This can be challenging in the case of large complex systems. In addition, there is still a dependence on traditional steady state simulation methods, such as Harmonic Balance, in order to obtain the steady state solution and the distortion contributions. The accuracy of the obtained results is quite high and is similar to the accuracy of general circuit-level simulation.

#### 3.4.3 Simplified Newton Method

Another one of the more recent simulation-based methods of analyzing distortion in nonlinear circuits is one which works around the principle of applying a simplified version of the Newton Method (iteration) on the nonlinear system to obtain the periodic steady-state solution and evaluate the intermodulation distortion products from the solution. This method is presented and described in [35]. Similar to the Linear Centric model approach presented earlier, this method also bypasses the need to compute the higher order device model derivatives that are required in Volterra Series analysis and greatly improves the CPU cost when compared to complete multi-tone nonlinear steady state analysis using Harmonic Balance. The simplified Newton method, when applied to solving a nonlinear set of equations of the form f(x) = 0, results in the following iterative formula

$$J(x^{(0)})\Delta x^{(i)} = -f(x^{i}); \quad x^{(i+1)} = x^{(i)} + \Delta x^{(i)}$$
(3.5)

where J is the Jacobian matrix which does not require updating under the simplified Newton method. After formulating the HB equations in the frequency domain, the differential algebraic equations set that describe the system are solved using a single-tone input to determine a periodic steady state Jacobian matrix ( $J^{PSS}$ ), which is smaller than the regular HB Jacobian matrix. The solution is then obtained by solving

$$J^{PSS}(\omega)\Delta X^{(j)}(\omega) = -H^{(j)}(\omega); \quad X^{(j+1)} = X^{(j)} + \Delta X^{(j)}$$
(3.6)

where  $H(\omega)$  is the formulation of the function f(x) of the system in the frequency domain. During the solution process, switching between the time and frequency domains occurs using the DFT and the IDFT to evaluate the nonlinear components of the system in order to compute the right-hand-side of equation (3.6), at each iteration. Because distortion analysis typically requires computing only up to 3rd order nonlinearities, this method can only involve as little as two steps of the simplified Newton method. What makes this method efficient is the need to only compute one fixed Jacobian matrix which is also of a smaller size and order than the HB Jacobian matrix. The drawback with this method is that it is inaccurate when compared to HB simulation. Also, since Newton iteration must be performed, the problems associated with such methods, such as convergence and the varying number of iterations required for different systems, also apply.

## **3.5 Volterra Series**

The Volterra functional series is a weakly non-linear system theory that has been widely adopted in the research literature of analog circuit analysis, where the weakly nonlinear system response is decomposed into responses of different orders. The use of Volterra series for distortion analysis has been traditionally adopted for RF circuits because not only are figures of merit such as IP3 determined, but it also gives designers insight into which circuit parameters or elements result in such behavior and how they can be modified to reduce their effects. Unfortunately, Volterra series analysis is rarely used by IC designers since they are limited to weakly non-linear system behavior and because they require complex analytical computations even for modest size systems. Volterra series describe the output of a nonlinear system as the sum of the response of a first order, second order, third order, etc. operators as shown in the diagram in Fig. 3.7. A Volterra series approximates an analytic function. Each operator can be expressed in either the time or frequency domains with an expression similar to a transfer function known as a Volterra kernel.

#### 3.5.1 Volterra Series Formulation

A Volterra series combines the representation of a linear, causal system with memory given by

$$y(t) = \int_{-\infty}^{\infty} h(\sigma)v(t-\sigma)d\sigma$$
(3.7)



Figure 3.7: Schematic Representation of a system characterized by a Volterra Series

with the Taylor series representation of a nonlinear system without memory given by

$$y(t) = \sum_{n=1}^{\infty} a_n [v(t)]^n$$
 (3.8)

to give the Volterra Series representation of a nonlinear system with memory which is given by

$$y(t) = H_1(v(t)) + H_2(v(t)) + \dots + H_n(v(t))$$
(3.9)

where

$$H_1(v(t)) = \int_{-\infty}^{\infty} h_1(\tau_1) v(t - \tau_r) d\tau_1$$
(3.10)

$$H_2(v(t)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) v(t - \tau_1) v(t - \tau_2) d\tau_1 d\tau_2$$
(3.11)

$$H_n(v(t)) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \cdots, \tau_n) \prod_{r=1}^n v(t - \tau_r) d\tau_1 \cdots d\tau_n \quad (3.12)$$

In the above relations y(t) is the output of a system H(t), with v(t) being the input.  $H_i(v(t))$  is the  $i^{th}$  Volterra operator while  $h_i(\tau_1, \dots, \tau_i)$  is the  $i^{th}$  Volterra kernel and are both of the  $i^{th}$  order [9]. The Volterra kernel  $h_i(\tau_1, \dots, \tau_i)$  is also considered to be the  $i^{th}$  order impulse response of the system [36]. The relation given by (3.9) is also consistent with the representation shown by Fig. 3.7.

The frequency domain representation of an *n*th order Volterra kernel, found using the multidimensional Laplace or Fourier transforms of the kernel, is commonly referred to as the *n*th order nonlinear transfer function. The Laplace transform  $H_n(s_1, \dots, s_n)$  of the kernel  $h_n(\tau_1, \dots, \tau_n)$  is found using

$$H_n(s_1,\cdots,s_n) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1,\cdots,\tau_n) e^{-(s_1\tau_1+\cdots+s_n\tau_n)} d\tau_1\cdots d\tau_n \quad (3.13)$$

with  $s_i = (\sigma_i + j\omega_i)$  being complex variables. If  $\sigma_i = 0$  then these would simplify to the Fourier Transform of the kernel.

#### 3.5.2 Limitations of Volterra Series

The Volterra Series approach has several notable limitations. Namely, the main problem is the need for computing complex analytical solutions for the system. The computation of the Volterra Kernels requires the need to determine higher-order derivatives of the circuit equations which can be quite difficult to obtain. Moreover, the Volterra series approach is also only applicable to circuits that assume weakly nonlinear system behavior. Several algorithms have been proposed to extend Volterra series analysis to circuits that exhibit strong nonlinear behavior. One such method is time varying Volterra series and is covered in the following section.

ı

#### 3.5.3 Time Varying Volterra Series

Since the Volterra Series analysis presented in the previous section applies only to weakly non-linear system behavior, we now consider the case for circuits that exhibit strong non-linear behavior such as switching mixers and switch capacitor circuits. If we assume the input is periodic, we can apply the method of periodically time-varying Volterra series for distortion analysis. For the more general case of large input signals that are not periodic, more general time-varying Volterra series can be applied [2].

The time varying network function  $H(t, \omega)$  first introduced by Zadeh [37] and

extended to multi-frequency networks  $(H_n(t, \omega_1, \dots, \omega_n))$  in [38], is an effective means to describe linear time varying systems and is given by

$$H_n(t,\omega_1,\ldots,\omega_n) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(t,\tau_1,\cdots,\tau_n) e^{-j\omega_n(t-\tau_n)} d\tau_1 \cdots d\tau_n \quad (3.14)$$

Since the network function is periodic, it can also be expressed in terms of its Fourier Series

$$H_n(t,\omega_1,\ldots,\omega_n) = \sum_{k=-\infty}^{\infty} H_{n,k}(\omega_1,\ldots,\omega_n) e^{jk\omega_s t}$$
(3.15)

where  $H_{n,k}(\omega_1, \ldots, \omega_n)$  are the coefficients of the Fourier Series. It is important to note that the main difference in the response between these systems and that of time invariant systems using regular Volterra Series, is that due to the time-varying nature, harmonic and intermodulation frequency components will appear at both the base bands and the side bands.

# 3.6 Conclusion

In this chapter, an overview of the significance of distortion analysis as a means for RF design engineers to analyze linearity has been discussed. In addition, a literature survey of some of the traditional and more recent approaches for computing IP3 and other distortion parameters for different systems has been presented. A special emphasis was placed on Volterra Series since it is directly related to some of the analysis of the method that is proposed in Chapter 4 of this thesis.

# **Chapter 4**

# Direct Numerical Computation of Intermodulation Distortion

# 4.1 Introduction

In this thesis, a new method for obtaining the IP3 is presented. Using the proposed approach the required Volterra kernels are computed numerically from the moments of the MNA system of equations of the system. It can therefore handle circuits with arbitrary complexity and nonlinearity. Furthermore it is to be noted that, IP3 is in fact obtained without the need to compute the steady-state solution of the system which requires a large CPU cost due to the dense nature of the Jacobian matrix and the number of Newton iterations required. Instead, the value of IP3 is computed from the system moments which only requires one LU decomposition of a sparse matrix. This method is therefore efficient and accurate as is presented in this chapter. A basic overview of the method is shown in the flowchart of Fig 4.1. The first step is to calculate the system moments in an efficient manner from the formulation of the equations in the frequency domain. The system moments are the Taylor series coefficients of the solution vector as a function of the input amplitude. The next step is to extract the Volterra kernels for analyzing distortion from the moment vectors. Finally the value of IP3 is found using the relevant Volterra kernels.



Figure 4.1: Overview of Proposed Method

# 4.2 Calculation of the Moments

The system moments are essentially the derivatives of the unknown solution vector X with respect to the input voltage [15]. Equation (2.6) is now expressed in the form shown below

$$\bar{\boldsymbol{G}}\boldsymbol{X} + \bar{\boldsymbol{C}}\boldsymbol{X} + \boldsymbol{F}(\boldsymbol{X}) - \boldsymbol{B}_{\boldsymbol{dc}} - \alpha \boldsymbol{B}_{\boldsymbol{ac}} = 0, \qquad (4.1)$$

In this expression,  $\alpha$  is the amplitude of the input signals and  $B_{ac}$  represents a vector with the only non-zero entries being entries of value '1' at the frequencies of interest. The system moments  $A_0 \dots A_q$  are then defined as the coefficients of the Taylor series of X as a function of  $\alpha$  in

$$X = A_0 + A_1 \alpha + A_2 \alpha^2 + A_3 \alpha^3 + \dots$$
  
= 
$$\sum_{k=0}^{q} A_k \alpha^k$$
 (4.2)

where  $A_k$  is the  $k^{th}$  moment of the system. Once the moments are determined, the distortion analysis parameters and the solution of the system can be obtained efficiently.

For the purpose of analyzing distortion efficiently, the moments vectors must be computed efficiently by the simulator. Next a method is presented to show how this task is performed [15]. By substituting (4.2) into (4.1), the following expression is obtained:

$$\bar{\boldsymbol{G}}\sum_{k=0}^{q}\boldsymbol{A}_{k}\alpha^{k} + \bar{\boldsymbol{C}}\sum_{k=0}^{q}\boldsymbol{A}_{k}\alpha^{k} + \sum_{k=0}^{q}\boldsymbol{D}_{k}\alpha^{k} - \boldsymbol{B}_{\boldsymbol{dc}} - \alpha\boldsymbol{B}_{\boldsymbol{ac}} = 0 \qquad (4.3)$$

where  $D_k$  are the Taylor coefficients of F(X)

$$\boldsymbol{F}(\boldsymbol{X}) = \sum_{k=0}^{q} \boldsymbol{D}_{k} \alpha^{k}$$
(4.4)

To solve for the zeroth moment  $A_0$ , the value of  $\alpha$  in (4.3) is set to zero.

Setting  $\alpha = 0$ :

$$\bar{\boldsymbol{G}}\boldsymbol{A}_0 + \bar{\boldsymbol{C}}\boldsymbol{A}_0 + \boldsymbol{F}(\boldsymbol{A}_0) = \boldsymbol{B}_{\boldsymbol{dc}}$$
(4.5)

Solving the above system to obtain  $A_0$  is simply obtaining the DC solution of the system.

To solve for the remaining moments ( $A_n$ ;  $n \ge 1$ ), the powers of alpha in (4.3) are equated. Equating the first power of  $\alpha$  results in

$$\bar{\boldsymbol{G}}\boldsymbol{A}_1 + \bar{\boldsymbol{C}}\boldsymbol{A}_1 + \boldsymbol{D}_1 = \boldsymbol{B}_{\boldsymbol{ac}} \tag{4.6}$$

Applying the chain rule to  $D_1 = \frac{\partial F}{\partial \alpha}|_{\alpha=0} = \frac{\partial F}{\partial X} \cdot \frac{\partial X}{\partial \alpha}|_{\alpha=0} = T_0 A_1$ , and substituting this expression into (4.6) results in,

$$(\bar{\boldsymbol{G}} + \bar{\boldsymbol{C}} + \boldsymbol{T}_0)\boldsymbol{A}_1 = \boldsymbol{B}_{ac} \tag{4.7}$$

The first moment can now be obtained using one LU Decomposition to solve (4.7). It is important to note that the matrix  $(\bar{G} + \bar{C} + T_0)$  is simply the sparse Jacobian matrix which is already computed when obtaining the DC solution. To obtain the remaining moments, the  $n^{th}$  powers of  $\alpha$  in (4.3) are equated to obtain:

$$\bar{\boldsymbol{G}}\boldsymbol{A}_n + \bar{\boldsymbol{C}}\boldsymbol{A}_n + \boldsymbol{D}_n = 0 \qquad n > 1 \tag{4.8}$$

To solve the system given in (4.8) efficiently for each value of n,  $D_n$  must be expressed in a different manner. Using the chain rule it can be expressed as

$$\frac{\partial F}{\partial \alpha} = \frac{\partial F}{\partial X} \cdot \frac{\partial X}{\partial \alpha} = T \frac{\partial X}{\partial \alpha}$$
(4.9)

where

$$T(\alpha) = \frac{\partial F(X)}{\partial X} = \sum_{k=0} T_k \alpha^k$$
(4.10)

Substituting (4.2), (4.4) and (4.10) into (4.9) results in

$$\sum_{i=1}^{q} i \boldsymbol{D}_{i} \alpha^{i-1} = \sum_{i=0}^{q} \boldsymbol{T}_{i} \alpha^{i} \sum_{i=1}^{q} i \boldsymbol{A}_{i} \alpha^{i-1}$$
(4.11)

To solve for  $D_n$ , the  $n^{th}$  derivative of (4.11) is taken and  $\alpha$  is set to zero.  $D_n$  can then be expressed as

$$\boldsymbol{D}_n = \boldsymbol{T}_0 \boldsymbol{A}_n + \frac{1}{n} \sum_{j=1}^{n-1} (n-j) \boldsymbol{T}_j \boldsymbol{A}_{n-j}$$
(4.12)

substituting (4.12) into (4.8) and rearranging yields

$$(\bar{\boldsymbol{G}} + \bar{\boldsymbol{C}} + \boldsymbol{T}_0)\boldsymbol{A}_n = -\frac{1}{n}\sum_{j=1}^{n-1}(n-j)\boldsymbol{T}_j\boldsymbol{A}_{n-j}$$
 (4.13)

This recursive relationship is used to calculate the remaining moments. The righthand side of equation (4.13) is calculated using the values of the previous moments  $(A_{n-j})$  that have already been obtained, in addition to the values of the partial derivatives of the non-linear vector functions with respect to the solution vector  $(T_j)$ . All that remains is to show how to obtain these terms. Since F(X) and Xare vectors, the term  $T(\alpha)$  in (4.10) will be a matrix of the form

$$\boldsymbol{T}(\alpha) = \frac{\partial \boldsymbol{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \cdots & \frac{\partial F_1}{\partial X_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial X_1} & \cdots & \frac{\partial F_n}{\partial X_n} \end{bmatrix}$$
(4.14)

where each  $\frac{\partial F_j}{\partial X_i}$  term is a block matrix in itself. To simplify the presentation of calculating these terms, only one of the terms in the  $T(\alpha)$  matrix shown in (4.14) is considered;  $\frac{\partial F_1}{\partial X_1}$ . If  $\frac{\partial F_1}{\partial X_1}$  is represented as matrix  $T_{11}$  and expressed as a Taylor series expansion, then the following expression is obtained

$$\frac{\partial F_1}{\partial X_1} = \boldsymbol{T}_{11} = \boldsymbol{P} = \sum_{j=0} \boldsymbol{P}_j \alpha^j$$
(4.15)

where the Taylor coefficient  $P_j$  is entered in  $T_j$  at the location corresponding to  $\frac{\partial F_1}{\partial X_1}$ . The  $P_j$  matrices are computed using

$$\boldsymbol{P}_{j} = \boldsymbol{\Gamma}^{-1} \begin{bmatrix} \frac{\partial f_{1}(x_{1}(t_{1}))}{\partial x_{1}} & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & \frac{\partial f_{1}(x_{1}(t_{s}))}{\partial x_{1}} \end{bmatrix} \boldsymbol{\Gamma}$$
(4.16)

where  $t_1$  to  $t_s$  are the time sample points equally spaced over the fundamental period (note that frequency mapping and diamond truncation [1] is used in order to handle quasi-periodic inputs efficiently using FFT), and  $\Gamma$  is the Inverse DFT matrix defined as

$$\Gamma = \begin{bmatrix} 1 & \cos(\Theta_{0,1}) & \sin(\Theta_{0,1}) & \cdots & \cos(\Theta_{0,H}) & \sin(\Theta_{0,H}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\Theta_{n,1}) & \sin(\Theta_{n,1}) & \cdots & \cos(\Theta_{n,H}) & \sin(\Theta_{n,H}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 \cos(\Theta_{N_h-1,1}) \sin(\Theta_{N_h-1,1}) \cdots \cos(\Theta_{N_h-1,H}) \sin(\Theta_{N_h-1,H}) \end{bmatrix}$$

with

$$\Theta_{n,k} = kn\left(\frac{2\pi}{N_h}\right) \tag{4.17}$$

Note that the matrix vector multiplication with  $\Gamma$  can be done efficiently by taking advantage of the Fast Fourier Transform (FFT) algorithm. It is also to be noted that the Jacobian is the same for all moments as can be seen from (4.7) and (4.13). Furthermore, since the expansion is at the DC operating point ( $A_0$  is the DC solution), this Jacobian is very sparse. The only expressions that remain to be determined are the derivatives of the nonlinear functions shown in (4.16). This is shown in the next section

#### **4.2.1** Obtaining the Derivatives

The derivatives of the nonlinear functions with respect to each variable in the solution vector must be determined in order to compute the system moments effectively. Once again, for simplicity of presentation only one of the terms from equation (4.14),  $\frac{\partial F_1}{\partial X_1}$ , is considered. The term  $g(x_1)$  is defined as,

$$g(x_1) = \frac{\partial f_1(x_1)}{\partial x_1} = \sum_{j=0} g_j \alpha^j$$
(4.18)

while noting the Taylor Series expansion of  $g(x_1)$ . The variable  $x_1(t)$  is also represented in Taylor Series form [15] as

$$x_1(t) = \sum_{i=0}^{\infty} a_i(t) \alpha^i$$
 (4.19)

The coefficients  $a_i(t)$  are the time domain versions of the moment vectors  $A_i(t)$ . Substituting (4.18) into (4.16) gives

$$\boldsymbol{P}_{j} = \boldsymbol{\Gamma}^{-1} \begin{bmatrix} g_{j}(x_{1}(t_{1})) & 0 \\ & \ddots & \\ 0 & g_{j}(x_{1}(t_{s})) \end{bmatrix} \boldsymbol{\Gamma}$$
(4.20)

The best way to show how to obtain the  $g_j$  terms is by taking an example. Consider the nonlinear diode current equation and its derivative, defined as

$$f_1(x_1) = I_s(e^{x_1/v_t} - 1)$$
(4.21)

$$g(x_1) = \frac{\partial f_1(x_1)}{\partial x_1} = \frac{I_s}{v_t} e^{x_1/v_t}$$
 (4.22)

It is also useful to express  $f_1(x_1)$  and  $\frac{\partial f_1}{\partial \alpha}$  in terms of  $g(x_1)$  at this point, which essentially becomes

$$f_1(x_1) = I_s(e^{x_1/v_t} - 1)$$
 (4.23)

$$= v_t g(x_1) - I_s \tag{4.24}$$

$$\Rightarrow \frac{\partial f_1(x_1)}{\partial \alpha} = v_t \frac{\partial g_1(x_1)}{\partial \alpha}$$
(4.25)

Now applying the chain rule of  $\frac{\partial f_1}{\partial \alpha} = \frac{\partial f_1}{\partial x_1} \frac{\partial x_1}{\partial \alpha}$  and substituting (4.18),(4.19) and (4.25) into this expression [39] yields

$$v_t \sum_{i=1}^n ig_i \alpha^{i-1} = \sum_{i=0}^{n-1} g_i \alpha^i \sum_{i=1}^n ia_i \alpha^{i-1}$$
(4.26)

By equating like powers of  $\alpha$ , this expression then simplifies to

$$g_n = \frac{1}{nv_t} \sum_{i=0}^{n-1} g_i a_{n-i}(n-i)$$
(4.27)

Similar expressions can be derived for analytic nonlinear function [15], [39]

Now that the algorithm to efficiently obtain the moments,  $A_k$ , of the system has been presented, the process showing how to extract the parameters necessary to determine the value of IP3 from these vectors is presented.

# 4.3 Numerical Computation of Volterra Kernels for Determining IP3

The parameter that has been selected to measure the distortion in the circuit is that of IP3. In this section, the method to compute the input 3rd order intercept point (IIP3) from the moment vectors is presented. To compute the value of the output 3rd order intercept point (OIP3), simply multiply the value of IIP3 by the gain of the circuit [40].

#### 4.3.1 Memoryless Systems

In order to simplify the presentation, we first consider a memoryless system. The output variable y is expressed as a power series of the input v. The input output relationship can be written as,

$$y = k_0 + k_1 v + k_2 v^2 + k_3 v^3 + \dots = \sum_n k_n v^n$$
(4.28)

Frequency	$A_0$	$A_1$	$A_2$	$A_3$
DC	$k_0$	0	$k_2$	0
$2\omega_1 \pm \omega_2$	0	0	0	$\frac{3}{4}k_3$
$\omega_1$	0	k1	0	$\frac{9}{4}k_{3}$
$\omega_2$	0	k <sub>1</sub>	0	$\frac{9}{4}k_{3}$
$2\omega_2 \pm \omega_1$	0	0	0	$\frac{3}{4}\mathbf{k_3}$
$2\omega_1$	0	0	$\frac{1}{2}k_2$	0
$2\omega_2$	0	0	$\frac{1}{2}k_{2}$	0

Table 4.1: Location of Taylor Series coefficients in Moment Vectors

Substituting  $v = \alpha(\cos(\omega_1 t) + \cos(\omega_2 t))$  into (4.28), truncating after  $k_3$ , and expanding using trigonometric identities we obtain

$$y = k_{0} + (k_{1}cos(\omega_{1}t) + k_{1}cos(\omega_{2}t))\alpha + (k_{2} + \frac{k_{2}}{2}cos(2\omega_{1}t) + k_{2}cos((\omega_{1} + \omega_{2})t) + k_{2}cos((\omega_{1} - \omega_{2})t) + \frac{k_{2}}{2}cos(2\omega_{2}t))\alpha^{2} + (\frac{9k_{3}}{4}cos(\omega_{1}t) + \frac{k_{3}}{4}cos(3\omega_{2}t) + \frac{9k_{3}}{4}cos(\omega_{2}t) + \frac{k_{3}}{4}cos(3\omega_{1}t) + \frac{3k_{3}}{4}cos((2\omega_{2} - \omega_{1})t) + \frac{3k_{3}}{4}cos((2\omega_{1} + \omega_{2})t) + \frac{3k_{3}}{4}cos((2\omega_{2} + \omega_{1})t) + \frac{3k_{3}}{4}cos((2\omega_{1} - \omega_{2})t))\alpha^{3}$$

$$(4.29)$$

From (4.29) and (4.2) the relationship between  $k_n$  and the system moments can be deduced since the solution vector X in (4.2) is essentially the output variable y in (4.29). By equating the same powers of  $\alpha$  in these two equations and noting the frequencies, we are able to determine the location of the  $k_n$  terms in the  $A_k$  vectors which are located as shown in Table 4.1 [31]. For example, if we equate the first power of  $\alpha$ , we can see that the vector  $A_1$  consists of  $k_1$  at frequency  $\omega_1$  and also another  $k_1$  at frequency  $\omega_2$ . Table 4.1 shows the first 3 moment vectors in addition to the zeroth moment vector, with the entries at the frequencies of interest for the computation of IP3 shown in bold.

#### 4.3.2 Systems with Memory Elements

In the presence of memory elements such as capacitors and inductors both sine and cosine components will be present in the moment vectors, and the computation of the system moments can be shown to be equivalent to the computation of the Volterra kernels at the corresponding frequencies. In fact, the magnitudes of the Volterra kernels of interest, represented as  $H_n(j\omega_1, j\omega_2, \ldots, j\omega_n)$ , are found at the appropriate entry in the first and third moments as shown in Table 4.2 [29]. To prove that the parameters in the moment vectors are equivalent to the numerical computation of Volterra kernels we start by writing the Volterra Series representation of a nonlinear system with memory, which is given by

$$y(t) = H_1(v(t)) + H_2(v(t)) + \dots + H_n(v(t))$$
(4.30)

where y(t) is the output of the system H(t), with v(t) being the input.  $H_i(t)$  is the  $i^{th}$  Volterra operator and is of the  $i^{th}$  order [9]. To determine the location of the kernels in the moment vectors, we substitute the expression for an input function with two tones of the same amplitude, similar to the case of memoryless systems, given by

$$v = \alpha(\cos(\omega_1 t) + \cos(\omega_2 t)) \tag{4.31}$$

$$= \frac{\alpha}{2}e^{j\omega_{1}t} + \frac{\alpha}{2}e^{-j\omega_{1}t} + \frac{\alpha}{2}e^{j\omega_{2}t} + \frac{\alpha}{2}e^{-j\omega_{2}t}$$
(4.32)

$$= v_a(t) + v_b(t) + v_c(t) + v_d(t)$$
(4.33)

into equation (4.30). Truncating the resulting equations after n = 3, we obtain the following expressions for each Volterra operator

$$\begin{split} H_1(v(t)) &= H_1(v_a) + H_1(v_b) + H_1(v_c) + H_1(v_d) \\ H_2(v(t)) &= H_2(v_a) + H_2(v_b) + H_2(v_c) + H_2(v_d) + 2H_2(v_a, v_b) + 2H_2(v_a, v_c) \\ &+ 2H_2(v_a, v_d) + 2H_2(v_b, v_c) + 2H_2(v_b, v_d) + 2H_2(v_c, v_d) \\ H_3(v(t)) &= H_3(v_a) + H_3(v_b) + H_3(v_c) + H_3(v_d) + 2H_3(v_a, v_b, v_c) + \\ &2H_3(v_a, v_c, v_b) + 2H_3(v_b, v_a, v_c) + 2H_3(v_b, v_c, v_a) + \\ &2H_3(v_c, v_a, v_b) + 2H_3(v_c, v_b, v_a) + 2H_3(v_a, v_a, v_d) + \\ &2H_3(v_c, v_b, v_c) + 2H_3(v_d, v_a, v_a) + 2H_3(v_c, v_c, v_b) + \\ &2H_3(v_c, v_a, v_d) + 2H_3(v_d, v_c, v_a) + 2H_3(v_d, v_a, v_c) + \\ &2H_3(v_c, v_a, v_d) + 2H_3(v_d, v_c, v_a) + 2H_3(v_d, v_a, v_c) + \\ &2H_3(v_a, v_c, v_d) + 2H_3(v_c, v_a, v_a) + 2H_3(v_c, v_c, v_a) + \\ &2H_3(v_c, v_a, v_c) + 2H_3(v_c, v_a, v_a) + 2H_3(v_c, v_c, v_a) + \\ &2H_3(v_c, v_a, v_c) + 2H_3(v_c, v_a, v_a) + 2H_3(v_c, v_c, v_d) + \\ &2H_3(v_c, v_a, v_c) + 2H_3(v_b, v_c, v_c) + 2H_3(v_c, v_c, v_d) + \\ &2H_3(v_a, v_b, v_a) + 2H_3(v_b, v_a, v_a) + 2H_3(v_c, v_c, v_d) + \\ &2H_3(v_a, v_b, v_a) + 2H_3(v_b, v_a, v_a) + 2H_3(v_c, v_c, v_d) + \\ &2H_3(v_c, v_d, v_c) + 2H_3(v_b, v_c, v_c) \end{pmatrix}$$

Each of the above kernels are then evaluated and expressed in the frequency domain using the expression

$$H_n(v_1) = \int_{-\infty}^{\infty} d\tau_1 \cdots \int_{-\infty}^{\infty} d\tau_n h_n(\tau_1, \cdots, \tau_n) \prod_{r=1}^n v_1(t - \tau_r)$$
(4.35)

for a single input function  $v_1$ , and also the following expression for multiple input functions  $v_n$ 

$$H_n(v_1,\cdots,v_n) = \int_{-\infty}^{\infty} d\tau_1 \cdots \int_{-\infty}^{\infty} d\tau_n h_n(\tau_1,\cdots,\tau_n) \prod_{r=1}^n v_r(t-\tau_r)$$
(4.36)

Using these relations to determine all the terms in (4.34) and by substituting into (4.28) then rearranging by grouping like powers of  $\alpha$ , the following input output relation is obtained:

$$y = H_{0} + (Re(H_{1}(j\omega_{1})e^{j\omega_{1}t} + H_{1}(j\omega_{2})e^{j\omega_{2}t}))\alpha + (\frac{1}{2}Re(H_{2}(j\omega_{1}, -j\omega_{1})) + \frac{1}{2}Re(H_{2}(j\omega_{2}, -j\omega_{2})) + \frac{1}{2}Re(H_{2}(j\omega_{1}, j\omega_{1})e^{j2\omega_{1}t}) + Re(H_{2}(j\omega_{1}, j\omega_{2})e^{j(\omega_{1}-\omega_{2})t}) + \frac{1}{2}Re(H_{2}(j\omega_{2}, j\omega_{2})e^{j(\omega_{1}+\omega_{2})t}) + Re(H_{2}(j\omega_{1}, -j\omega_{2})e^{j(\omega_{1}-\omega_{2})t}) + \frac{1}{2}Re(H_{2}(j\omega_{2}, j\omega_{2})e^{j2\omega_{2}t}))\alpha^{2} + (\frac{3}{4}Re(H_{3}(j\omega_{1}, j\omega_{1}, j\omega_{2})e^{j(2\omega_{1}+\omega_{2})t}) + \frac{3}{4}Re(H_{3}(j\omega_{1}, j\omega_{1}, -j\omega_{2})e^{j(2\omega_{1}-\omega_{2})t}) + \frac{3}{4}Re(H_{3}(j\omega_{1}, j\omega_{1}, -j\omega_{2})e^{j(2\omega_{2}-\omega_{1})t}) + \frac{3}{2}Re(H_{3}(j\omega_{1}, j\omega_{2}, -j\omega_{2})e^{j\omega_{1}t}) + \frac{3}{4}Re(H_{3}(j\omega_{1}, j\omega_{1}, -j\omega_{1})e^{j\omega_{1}t}) + \frac{3}{2}Re(H_{3}(j\omega_{1}, -j\omega_{1}, j\omega_{2})e^{j\omega_{2}t}) + \frac{3}{4}Re(H_{3}(j\omega_{3}, j\omega_{2}, -j\omega_{2})e^{j\omega_{3}t}) + \frac{1}{4}Re(H_{3}(j\omega_{1}, j\omega_{1}, j\omega_{2}, j\omega_{2})e^{j(2\omega_{2}-\omega_{1})t}))\alpha^{3}$$

$$(4.37)$$

The Volterra Series expression in (4.37) is similar to the expression shown in (4.29) which proves that the proposed method is essentially the numerical computation of the Volterra operators. In a similar fashion to memoryless systems, the location of the parameters to compute the value of IP3 are the entries in bold found in the moment vectors at the locations shown in Table 4.2 [9], [41]. For the case of memoryless systems, the expressions for these kernels simplify to the terms  $k_1$  and  $k_3$  in Table 4.1.

# 4.4 Computation of IP3 from Volterra Kernels

The third order intercept point (IP3) is defined as the power at which the fundamental frequency is equal to the 3rd order intermodulation frequency, which can now be found by obtaining the appropriate entries from  $A_1$  and  $A_3$  and using the following relation [29]

$$|H_1(j\omega_1)|IP_{3i} = \frac{3}{4}|H_3(j\omega_1, j\omega_1, -j\omega_2)|IP_{3i}^3$$
(4.38)

which then simplifies to

$$IP_{3i} = \frac{2}{\sqrt{3}} \sqrt{\frac{|H_1(j\omega_1)|}{|H_3(j\omega_1, j\omega_1, -j\omega_2)|}}$$
(4.39)

The above relation is for the Low Side IIP3, to obtain the High Side IIP3 we replace  $|H_1(j\omega_1)|$  by  $|H_1(j\omega_2)|$  and  $|H_3(j\omega_1, j\omega_1, -j\omega_2)|$  by  $|H_3(j\omega_1, -j\omega_2, -j\omega_2)|$ . For the case of circuits with memoryless elements, the value of IIP3 will be the same regardless of whether the low side or high side intermodulation term was considered.

A summary of the main steps of the algorithm can be found in Fig. 4.2. It is to be noted that the above process for obtaining IP3 only required the computation of the first 3 moments. The CPU cost required is one LU decomposition of the Jacobian matrix, and one DC solution. Furthermore, the Jacobian used here is at the DC operating point, and it is therefore very sparse unlike a typical Harmonic Balance Jacobian matrix. In contrast, the brute force simulation based approach requires the solution of a system of equations with a dense Jacobian at each newton iteration which requires high CPU cost. With regrads to the accuracy, for most RF circuits with low power signals, it is sufficient to compute only the first 3 moments to obtain very accurate results. However, for the case of greater input RF power, higher order odd-numbered responses will be large enough to affect the magnitude of the responses at the fundamental and IM3 frequencies, thus affecting the value of IP3. In this case, more accurate solutions can be obtained by computing higher moments and utilizing their values in the computation of IP3. The computation of each additional moment requires very little CPU cost as it simply consists of forward/backward substitutions using (4.13) since the left-hand-side Jacobian matrix does not change. The accuracy of the proposed approach using only 3 moments is shown in the Examples presented in section IV. It is also important to note that other terms used for analyzing distortion, such as the Second Order Intercept

Point (*IP2*), Harmonic Distortion (*HD*) and Intermodulation Free Dynamic Range (*IMFDR*<sub>3</sub>), can be easily computed from the moments by using the appropriate values of  $k_n$  [29].

Frequency	$A_0$	$A_1$	$A_2$	$A_3$
DC	$H_0$	0	$\frac{1}{2} H_2(j\omega_1,-j\omega_1) +$	0
			$\frac{1}{2} H_2(j\omega_2,-j\omega_2) $	
$2\omega_1 \pm \omega_2$	0	0	0	$rac{3}{4} \mathbf{H_3}(\mathbf{j}\omega_1,\mathbf{j}\omega_1,-\mathbf{j}\omega_2) $
$\omega_1$	0	$ \mathbf{H_1}(\mathbf{j}\omega_1) $	0	$ rac{3}{2} H_3(j\omega_1,j\omega_2,-j\omega_2) + $
				$rac{3}{4} H_3(j\omega_1,j\omega_1,-j\omega_1) $
$\omega_2$	0	$ \mathbf{H_1}(\mathbf{j}\omega_2) $	0	$\frac{3}{2} H_3(j\omega_2,j\omega_1,-j\omega_1) +$
				$rac{3}{4} H_3(j\omega_2,j\omega_2,-j\omega_2) $
$2\omega_2 \pm \omega_1$	0	0	0	$ rac{3}{4} \mathbf{H_3}(\mathbf{j}\omega_1,-\mathbf{j}\omega_2,-\mathbf{j}\omega_2)  $
$2\omega_1$	0	0	$rac{1}{2} H_2(j\omega_1,j\omega_1) $	0
$2\omega_2$	0	0	$  rac{1}{2}   H_2(j\omega_2, j\omega_2)  $	0

Table 4.2: Volterra Kernel Locations in Moment Vectors

# 4.5 Moments Calculation Example

The best method to clarify the algorithm for computing the moments is through the use of an example. Consider the example circuit shown in Fig. 4.3 which is driven with a voltage source with a waveform  $V_{in} = 0.85 + 0.02 \cos(200\pi \times 10^6 t)V$  and contains a resistance of 1 $\Omega$  in addition to a diode with a current characteristic of  $I_D = I_S(e^{(V_A - V_B)}) - 1)A$ , with  $I_S = 10^{-15}$ ,  $v_T = 0.025$  and  $V_A - V_B$  being the voltage drop across the diode. The MNA equations for this circuit are:

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ I_E \end{bmatrix} + \begin{bmatrix} 0 \\ I_s(e^{v_2/v_T} - 1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_{in} \end{bmatrix}$$
(4.40)

- 1. Set up the system equations in the frequency domain according to the formulation shown in (2.6)
- 2. Calculate the zeroth moment  $A_0$  as defined in (4.2) by finding the DC solution of the system as shown in (4.5)
- 3. Calculate the first moment  $A_1$  by solving the formulation in (4.7)
- 4. Calculate the remaining moments  $(A_n; n > 1)$  by solving (4.13) recursively
- 5. Obtain the Volterra kernels from the entries in the moment vectors at the fundamental frequencies ( $\omega_1$  and  $\omega_2$ ) and the 3rd order intermodulation frequencies ( $2\omega_1 \omega_2$  and  $2\omega_2 \omega_1$ ) as outlined in Table 4.1 for memoryless systems, and in Table 4.2 for systems with memory elements.
- 6. Determine the distortion by calculating IP3 according to (4.39)

#### Figure 4.2: Summary of the proposed algorithm



Figure 4.3: Example circuit to illustrate computation of moment vectors

Taking three harmonics (H = 3), the number of variables in the expanded system now becomes  $N_h = 3 \times (2H + 1) = 21$ . The matrices in equation (4.1) for this circuit are found to be of dimensions  $21 \times 21$  ( $21 \times 1$  for vectors), with each block matrix of dimension  $7 \times 7$  ( $7 \times 1$  for vectors), and are given by

$$\bar{\boldsymbol{G}} = \begin{bmatrix} 1 & -1 & 1 \\ \ddots & \ddots & \ddots \\ 1 & -1 & 1 \\ \hline -1 & 1 & & \\ \hline -1 & 1 & & \\ \hline & \ddots & \ddots & \\ \hline & -1 & 1 & \\ \hline 1 & & & \\ & \ddots & & \\ 1 & & & & \\ \hline & 1 & & \\ \end{bmatrix}; \boldsymbol{X} = \begin{bmatrix} a_0(v_1) \\ \vdots \\ b_3(v_1) \\ \hline a_0(v_2) \\ \vdots \\ b_3(v_2) \\ \hline a_0(I_E) \\ \vdots \\ b_3(I_E) \end{bmatrix}$$
(4.41)

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_{s}(e^{v_{2}(t_{0})/v_{T}} - 1) \\ I_{s}(e^{v_{2}(t_{0})/v_{T}} - 1) \end{bmatrix} \boldsymbol{B}_{dc} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \boldsymbol{B}_{ac} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
(4.42)

where  $\Gamma^{-1}$  is obtained from (4.17) and the vector of time samples given by

$$\boldsymbol{X}_{s} = [0, \dots, 0 | v_{2}(t_{0}), \dots, v_{2}(t_{N_{b}-1}) | 0, \dots, 0]^{T}$$
(4.43)

is obtained using  $X_s = \Gamma X$ . Since there are no memory elements in the circuit, the matrix  $\bar{C} = [0]$  and the amplitude of the input signal is  $\alpha = 0.02$ .

The zeroth moment vector,  $A_0$ , can now be obtained by solving (4.5) with the given matrices and the solution is shown in Table 4.3. To obtain the remaining moments, we need to obtain the DC Jacobian matrix  $(\bar{G} + \bar{C} + T_0)$ , for which we need to obtain the value of  $T_0$ . Since there is only one nonlinear function  $(f_2)$ , which in turn is only a function of one variable  $(v_2)$ , the matrix T in (4.14) is of the block matrix form

$$\boldsymbol{T}(\alpha) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\partial F_2}{\partial X_2} & 0 \\ 0 & 0 & 0 \end{bmatrix}; \qquad \frac{\partial F_2}{\partial X_2} = \boldsymbol{T}_{22} = \boldsymbol{P} = \sum_{j=0} \boldsymbol{P}_j \alpha^j$$
(4.44)

Now, given that  $f_2(x_2) = 10^{-15}(e^{v_2/0.025} - 1)$ , then  $g(x_2) = \frac{\partial f_2(x_2)}{\partial x_2} = 4 \times 10^{-14}e^{v_2/0.025}$ . To obtain matrix  $T_0$ , we need to compute matrix  $P_0$ , which at DC (since  $\alpha = 0$ ) means that the time samples  $(t_1 - t_s)$  are all the same value. Therefore we have, according to the relation given by (4.20)

$$\boldsymbol{P}_{0} = \boldsymbol{\Gamma}^{-1} \begin{bmatrix} g_{0}(x_{2}(t_{1})) & 0 \\ & \ddots & \\ 0 & g_{0}(x_{2}(t_{s})) \end{bmatrix} \boldsymbol{\Gamma}$$
(4.45)

For this we have  $g(x_2) = 4 \times 10^{-14} e^{v_{DC}/0.025} = 4 \times 10^{-14} e^{0.85/0.025} = 23.338$ . The matrix  $P_0$  now becomes

$$\boldsymbol{P}_{0} = \boldsymbol{\Gamma}^{-1} \begin{bmatrix} 23.338 & 0 \\ & \ddots & \\ 0 & 23.338 \end{bmatrix} \boldsymbol{\Gamma}$$
(4.46)

In turn, matrices  $T_0$  and the DC Jacobian ( $ar{G} + ar{C} + T_0$ ) are also found to be



The first moment  $A_1$  can now obtained by solving (4.7) using the given matrices and the solution is shown in Table 4.3. It is important to note how sparse is the DC Jacobian matrix that was computed in (4.48). This matrix will also remain static and be used for the computation of the remaining moment vectors.

The next step is to calculate the remaining moment vectors recursively using the relation given by (4.13). All the terms in this relation have already been obtained except  $T_j$  which has a similar structure to  $T_0$  obtained earlier except now to com-
pute the terms  $g_j(x_2)$  we use the relation given by (4.27). Taking the first 3 terms we have

$$\boldsymbol{g}_{1} = \frac{1}{v_{t}}\boldsymbol{g}_{0}\boldsymbol{a}_{1} = \begin{bmatrix} 23.338 & 0 \\ & \ddots & \\ 0 & 23.338 \end{bmatrix} \boldsymbol{\Gamma} \begin{bmatrix} 0 \\ 0.3019 \\ 0 \\ \vdots \end{bmatrix}$$
(4.49)

$$\boldsymbol{g}_{2} = \frac{1}{2v_{t}} (2\boldsymbol{g}_{0}\boldsymbol{a}_{2} + \boldsymbol{g}_{1}\boldsymbol{a}_{1})$$
(4.50)

$$\boldsymbol{g}_{3} = \frac{1}{3v_{t}} (3\boldsymbol{g}_{0}\boldsymbol{a}_{3} + 2\boldsymbol{g}_{1}\boldsymbol{a}_{2} + \boldsymbol{g}_{2}\boldsymbol{a}_{1})$$
(4.51)

The terms  $a_i$  are the time domain versions of the moment vectors, namely  $a_i(t) = \Gamma A_i$ . Also it is important to mention that the vector  $g_i$  must be made into a diagonal matrix when computing the product of  $g_j a_k$  in (4.49-4.51) so that the matrix indexes match.

### 4.6 Numerical Examples

In this section, numerical results of simulations performed on two example circuits are shown in order to illustrate the proposed approach. The value of IP3 obtained using the proposed method, which does not require a harmonic balance solution, is compared with that obtained using the brute force method which is based on multitone harmonic balance simulation. As was expected the results were identical.

#### **4.6.1** Example 1

The first example considered is a common emitter Low Noise Amplifier (LNA) with an LC tank centered at 1GHz, and input and output matching networks as shown in Fig. 4.4. In order to measure the linearity of the circuit the brute force method was first used by applying two -50 dBm input tones,  $f_1 = 1$  GHz and  $f_2 = 1.01$  GHz, and performing a standard harmonic balance analysis. The values of the circuit components used in the simulation are summarized in Table 4.4 while the results are

Variable	Frequency	$A_0$	$A_1$	$A_2$	$A_3$
$v_1$	DC	0.85	0	0	0
	$\cos(\omega_1)$	0	1	0	0
	$\sin(\omega_1)$	0	0	0	0
	$\cos(2\omega_1)$	0	0	0	0
	$\sin(2\omega_1)$	0	0	0	0
	$\cos(3\omega_1)$	0	0	0	0
	$\sin(3\omega_1)$	0	0	0	0
$v_2$	DC	0.7922	0	-0.6364	0
	$\cos(\omega_1)$	0	0.3019	0	4.2049
	$\sin(\omega_1)$	0	0	0	0
	$\cos(2\omega_1)$	0	-0.6364	0	0
	$\sin(2\omega_1)$	0	0	0	0
	$\cos(3\omega_1)$	0	0	0	1.4016
	$\sin(3\omega_1)$	0	0	0	0
$I_E$	DC	-0.0578	0	-0.6364	0
	$\cos(\omega_1)$	0	-0.6981	0	4.2049
	$\sin(\omega_1)$	0	0	0	0
	$\cos(2\omega_1)$	0	0	-0.6364	-10.2786
	$\sin(2\omega_1)$	0	0	0	0
	$\cos(3\omega_1)$	0	0	0	1.4016
	$\sin(3\omega_1)$	0	0	0	0

ţ

Table 4.3: Numerical Values of Moment Vectors for sample circuit



Figure 4.4: Circuit Diagram Example 1

shown in Table 4.5 and Fig. 4.5 and Fig. 4.6. The calculated IIP3 in this case was -12.02 dBm and the Output IP3 (OIP3) was found to be -7.79 dBm. This simulation was run with 10 harmonics, therefore the size of the dense Jacobian which had to be solved was  $3315 \times 3315$  due to 10 harmonics of the fundamental tones in addition to the intermodulation tones truncated using diamond truncation [1].

The distortion was then analyzed using the proposed approach by computing the moments of the system and extracting the Volterra kernels at the appropriate frequencies. Note that only three moments were required, and the computation of these moments was done at the DC biasing point, thus resulting in a sparse Jacobian. Also note that the same Jacobian is used to compute all three moments. Finally, these moments are obtained from linear equations and Newton Iteration is therefore not required. The results obtained are summarized in Table 4.6. The resulting IIP3 was found to be also -12.02 dBm.

The results of the simulation clearly show the accuracy of the method with only a very small error present. To achieve greater accuracy and for larger input signal powers, more moments can be utilized to obtain the IM3 tone components. The more moments that are utilized in the calculation, the more accurate the results will be without requiring significant additional CPU cost.

Circuit Element	Value	
R <sub>T</sub>	2000Ω	
$\mathbf{L}_{\mathbf{T}}$	$0.1 \mu H$	
$\mathbf{C}_{\mathbf{T}}$	0.2508 pF	
$\mathbf{R_s}$	$50\Omega$	
$\mathbf{R}_{\mathbf{L}}$	$50\Omega$	
$L_{M1}$	12.71nH	
$C_{M1}$	5.642 pF	
$C_{M2}$	9.195 pF	
$C_{M3}$	2.903 pF	
C <sub>C</sub>	0.1F	
I <sub>DC</sub>	$10\mu A$	
V <sub>CC</sub>	5V	
Vin	$cos(\omega_1 t) + cos(\omega_2 t)mV$	

Table 4.4: Circuit Element Values for Example 1

Table 4.5: Distortion Analysis of Example 1 using Traditional Approach

	<b>Brute Force</b>
$P_{out}$ at $\omega_1$	-45.79 dBm
$P_{out}$ at $2\omega_1 - \omega_2$	-121.77 dBm
P <sub>in</sub>	-50  dBm
IIP3	−12.02 dBm
OIP3	-7.79 dBm

	<b>Proposed Method</b>
$ H_1(j\omega_1) $	1.6244 V
$ H_3(j\omega_1,j\omega_1,-j\omega_2) $	343.749 V
IIP3	-12.02 dBm
OIP3	−7.79 dBm

Table 4.6: Distortion Analysis of Example 1 using Proposed Method



Figure 4.5: Output Power of Fundamental and IM3 tones of Example 1



Figure 4.6: Output Voltage Spectrum for Circuit in Example 1

The diagram in Fig. 4.7 shows the sparsity pattern of the HB Jacobian while Fig. 4.8 shows the sparsity pattern of the DC Jacobian matrix used to find the moments. The reduction in the number of non-zero elements between the two matrices is significant thus resulting in significant CPU cost saving when performing LU decomposition on each matrix.

### 4.6.2 Example 2

The second example considered is the differential cascode LNA circuit shown in Fig. 4.9. This amplifier is biased using a DC Current Source in the emitter and also with DC voltage biasing at the bases of the transistors. The biasing circuitry is not shown in the diagram and neither are the matching networks to  $50\Omega$  source and load impedances. The current source is implemented using a current mirror topology. This amplifier has a differential voltage gain of 18.3dB. The distortion at the output is analyzed by computing IP3. Linearity was measured first using the brute force approach by applying two -70dBm input tones,  $f_1 = 1$  GHz and  $f_2 =$ 



Figure 4.7: Sparsity Pattern of HB Jacobian for Circuit in Example 1



Figure 4.8: Sparsity Pattern of DC Jacobian for Circuit in Example 2



Figure 4.9: Circuit Diagram Example 2

1.01 GHz, and performing a standard harmonic balance analysis. The values of the circuit components used in the simulation are summarized in Table 4.7 while the results are shown in Table 4.8 and Fig. 4.10 and Fig. 4.11. The measured IIP3 in this case was found to be -13.89 dBm, and the OIP3 was 4.63 dBm. This simulation was run with 10 harmonics, therefore the size of the dense Jacobian which had to be solved was  $9061 \times 9061$  due to 10 harmonics of the fundamental tones in addition to the diamond truncation tones.

The distortion was then analyzed using the proposed approach by computing the moments of the system and extracting the Volterra kernels at the appropriate frequencies. Note that only three moments were required, and the computation of these moments was done at the DC biasing point, thus resulting in a sparse Jacobian.

<b>Circuit Element</b>	Value	
R <sub>c1</sub>	$10 \ \text{k}\Omega$	
$R_{c2}$	10 kΩ	
$L_{E1}$	$10 \ nH$	
$L_{E2}$	$10 \ nH$	
$\mathbf{R}_{\mathbf{L}}$	$50\Omega$	
$\mathbf{R}_{\mathbf{S}}$	$50\Omega$	
$C_{c}$	0.1 <i>F</i>	
I <sub>DC</sub>	$0.425 \ mA$	
V <sub>CC</sub>	5V	
$\mathbf{V}_{in}$	$0.1cos(\omega_1 t) + 0.1cos(\omega_2 t) mV$	

Table 4.7: Circuit Element Values for Example 2

 Table 4.8: Distortion Analysis of Example 2 using Traditional Approach

	Harmonic Balance Results
$P_{out}$ at $\omega_1$	-51.48 dBm
$P_{out}$ at $2\omega_1 - \omega_2$	-163.81 dBm
$P_{in}$	-70 dBm
IIP3	-13.89 dBm
OIP3	4.63 dBm

	<b>Proposed Method</b>
$ H_1(j\omega_1) $	4.2262 V
$ H_3(j\omega_1,j\omega_1,-j\omega_2) $	1374.48 V
IIP3	−13.87 dBm
OIP3	4.63 dBm

Table 4.9: Distortion Analysis of Example 2 using Proposed Method

The results obtained are summarized in Table 4.9. The resulting IIP3 was found to be also -13.89 dBm.

The diagram in Fig. 4.12 shows the sparsity pattern of the HB Jacobian while Fig. 4.13 shows the sparsity pattern of the DC Jacobian matrix used to find the moments. The reduction in the number of non-zero elements between the two matrices is even more significant than that in Example 1, thus resulting in even greater significant CPU cost saving.

### 4.6.3 CPU cost comparison

The data in Table 4.10 shows a comparison of the CPU times and the speed-up between the proposed method and the HB solution. The speed-up over a harmonic balance simulation was 19.8 times for Example 1, and 64.6 times for Example 2. This speed-up is due to three main reasons. Firstly, the moments used in the proposed method are found by solving a linear equation without the need for any Newton Iteration. Secondly, the left-hand-side matrix in (4.13) for finding the moments is the same for all moments, while the Harmonic Balance Jacobian is different at each Newton Iteration. Finally the HB Jacobian is significantly more dense than the Jacobian used for solving for the moments as was shown earlier. For Example 2, the 9061  $\times$  9061 HB Jacobian contains 2,033,113 non-zero elements while the 9061  $\times$ 9061 matrix for finding the moments contains only 36,875 non-zeros. As a result, the CPU time needed for 1 LU decomposition of the HB Jacobian was 1.14 seconds for Example 1 and 19.98 seconds for Example 2 as opposed to 0.016 seconds and



Figure 4.10: Output Power of Fundamental and IM3 Tones of Example 2



Figure 4.11: Output Voltage Spectrum for Example 2



Figure 4.12: Sparsity Pattern of HB Jacobian for Circuit in Example 2



Figure 4.13: Sparsity Pattern of DC Jacobian for Circuit in Example 2

	<b>HB</b> Solution	<b>Proposed Method</b>	Speed-up
Example 1	11.14	0.562	19.8
Example 2	197.80	3.063	64.6

Table 4.10: Comparison of CPU times between proposed method and HB solution

٠

0.078 seconds respectively, for 1 LU decomposition of the matrix for finding the moments. It is also important to note that the greater the number of non-linear elements present in the system, the more significant is the speed-up between the two approaches as can be seen when comparing the results of Example 1 to those of Example 2.

# **Chapter 5**

# **Conclusions and Future Research**

## 5.1 Conclusion

In this thesis, a new simulation method for measuring distortion at the output of a non-linear system based on the calculation of the system moments was presented and described. It has been demonstrated that by using this new simulation based approach to compute IP3 from the moments, distortion analysis of RF circuits becomes significantly more efficient. The proposed model is more flexible than other existing methods and works for many types of systems.

The main advantages that the proposed method enjoys over the Harmonic Balance method are summarized as follows:

- The DC Jacobian matrix used to obtain the moment vectors from the expanded set of MNA equations is very sparse. This is in contrast to the HB Jacobian matrix which is dense and must be recomputed, stored and manipulated at each iteration of the solution. This difference presents a significant reduction in CPU cost.
- The DC Jacobian matrix that is used is also static and does not change throughout the algorithm. For this reason, it needs to be computed, stored and inverted only once. The HB Jacobian matrix changes at each iteration of the solution, which could be a significant number of times. This means it has to

be computed and manipulated at each iteration, which when you also consider its dense structure, leads to increased CPU cost.

• The proposed method is essentially equivalent to numerically computing the Volterra Series kernels, which can be used to analyze distortion using several parameters. This avoids the need to perform complex analytical manipulations to compute the Volterra kernels.

## 5.2 Further Research

The research that has been presented in this thesis can be further extended to cover more types of RF circuits in addition to the possibility of increasing efficiency. A selection of such future work is highlighted in this section

### **5.2.1** Distortion Analysis of Mixer Circuits

In the previous chapter it was shown how the value of IP3 can be obtained efficiently for RF circuits in general. Mixer circuits (e.g. Gilbert Cell [40]), however, are a special type of RF circuit due to the presence of a high power Local Oscillator (LO) frequency signal in addition to the RF signals. In the presence of two RF input tones, the vector of intermodulation frequencies will therefore consists of 3 local frequencies and their harmonics. For these reasons, the method for obtaining IP3 of Mixer circuits differs to that presented in this thesis and is the subject of future research work.

Due to the presence of 3 input frequencies (two at the RF input and 1 at the LO input), a new truncation algorithm must be developed that maps the possible combinations and permutations of intermodulation frequencies to equally spaced artificial frequencies such that the DFT and inverse DFT operations can be performed. The existing Block and Diamond truncations are only limited to two frequencies.

The calculation of the moments is done in a similar fashion to that for general RF circuits with some significant differences. Firstly, since the moment vectors

are essentially the derivatives of the solution vector with respect to the RF input amplitudes, the zeroth moment vector,  $A_0$ , is no longer simply the DC solution of the system as this vector will contain non-zero components at the harmonics of the LO frequencies. Another significant difference is the noticeable increase in the number of variables and more importantly in the size of the Jacobian matrix, even with only a small order for the number of harmonics (H). Also the Jacobian matrix, due to the presence of the LO frequencies and their harmonics, will no longer be of similar sparsity as the DC Jacobian that was used in the method proposed in this thesis. To increase efficiency, this Jacobian must be pruned in order to sparsify a significant portion of nonzero elements that are not exactly zeros, but are quite small.

## 5.2.2 Multi-tone Distortion Analysis using Single-tone Simulation

One of the ways to further reduce CPU cost and to dramatically increase efficiency when analyzing distortion is to perform the necessary analysis using only a singletone simulation. The aim is to compute the moment vectors of a single-tone excited system and extract the parameters necessary for determining IP3 of the multi-tone response of the same circuit, assuming the amplitudes of the multiple tones are the same. Work on this approach is currently in progress and initial results have been promising.

# Bibliography

- K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*. Boston: Kluwer Academic, 1990.
- [2] P. Li and L. T. Pileggi, "Efficient per-nonlinearity distortion analysis for analog and rf circuits," *IEEE Transactions on Computer Aided Design of Inte*grated Circuits and Systems, vol. 10, pp. 1297–1309, 2003.
- [3] E. H. Armstrong, "A study of heterodyne amplification by the electron relay," *Proceedings of the IRE*, vol. 5, pp. 145–159, 1917.
- [4] K. Kundert, "Introduction to rf simulation and its application," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 1298–1319, 1999.
- [5] K. Mayaram, D. C. Lee, S. Moinian, D. Rich, and J. Roychowdhury, "Overview of computer-aided analysis tools for RFIC simulation: Algorithms, features, and limitations," in *Proc. IEEE Custom Integrated Circuits Conf.*, 1997, pp. 505–512.
- [6] K. Kundert and A. Sangiovanni-Vincentelli, "Simulation of nonlinear circuits in the frequency domain," *IEEE Trans. Computer-Aided Design*, vol. 5, no. 4, pp. 521–535, Oct. 1986.
- [7] S. Maas, *Nonlinear Microwave and RF Circuits*. Upper Saddle River, NJ: Artech House Publishers, 2003.

- [8] W. Reiss, "Nonlinear distortion analysis of p-i-n diode attenuators using volterra series representation," *Circuits and Systems*, 1984.
- [9] R. Baki, C. Beainy, and M. N. El-Gamal, "Distortion analysis of high frequency log-domain filters using volterra series," *Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, 2003.
- [10] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, June 1975.
- [11] T. J. Aprille, Jr. and T. N. Trick, "Steady-state analysis of nonlinear circuits with periodic inputs," *Proceedings IEEE*, vol. 60, no. 1, pp. 108–114, Jan. 1972.
- [12] S. Skelboe, "Computation of the periodic steady-state response of nonlinear networks by extrapolation methods," *IEEE Trans. Circuits Syst.*, vol. 27, no. 3, pp. 161–175, Mar. 1980.
- [13] M. S. Nakhla and J. Vlach, "A piecewise harmonic-balance technique for determination of periodic response of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. 23, no. 2, pp. 85–91, Feb. 1976.
- [14] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Norstrand Reinolds, 1983.
- [15] R. Khazaka, "Projection based techniques for the simulation of rf circuits and high speed interconnects," PhD Thesis, Carleton University, Ottawa ON, Canada, 2002.
- [16] Y. Hu, X. Duan, and K. Mayaram, "A comparison of time-domain and harmonic balance steady-state analyses for coupled device and circuit simulation," in *NEWCAS*, 2004, pp. 93–96.

- [17] M. S. Nakhla and F. H. Branin, "Determining the periodic response of nonlinear systems by a gradient method," *Int. J. Circuit Theory Appl.*, vol. 5, no. 3, pp. 225–273, July 1977.
- [18] P. K. Gunupudi, M. S. Nakhla, and R. Achar, "Efficient simulation of highspeed distributed interconnects using krylov-subspace techniques," in *Proc. IEEE conf. on Electrical Performance of Electronic Packaging*, Oct. 1998.
- [19] K. Kundert, "Simulation methods for rf integrated circuits," in *ICCAD*, 1997, pp. 752–765.
- [20] K. S. Kundert and A. Sangiovanni-Vincentelli, "Finding the steady-state response of analog and microwave circuits," in *Custom Integrated Circuits Conference*, 1988, pp. 6.1/1–6.1/7.
- [21] M. Vouvakis, K. Zhao, and J.-F. Lee, "Modeling large almost periodic structures using a non-overlapping domain decomposition method," in Antennas and Propagation Society International Symposium, 2004, pp. 343–346.
- [22] V. Rizzoli, F. Mastri, F. Sgallari, and G. Spaletta, "Harmonic-balance simulation of strongly nonlinear very large-size microwave circuits by inexact newton methods," in *IEEE MTT-S Int. Microwave Symp. Dig.*, 1996, pp. 1357– 1360.
- [23] E. Gad, R. Khazaka, M. Nakhla, and R. Griffith, "A circuit reduction technique for finding the steady-state solution of nonlinear circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 12, pp. 2389–2396, Dec. 2000.
- [24] S. Egami, "Nonlinar, linear analysis and computer-aided design of resistive mixers," *IEEE Trans. Microwave Theory Tech.*, vol. 22, no. 3, pp. 270–275, Mar. 1973.
- [25] P. Pai, E. Gad, R. Khazaka, R. Achar, and M. Nakhla, "Parametric circuit reduction for steady-state simulation with multi-tone inputs," 2004.

- [26] R. C. Melville, L. Trajković, S.-C. Fang, and L. T. Watson, "Artificial parameter homotopy methods for the DC operating point problem," *IEEE Trans. Computer-Aided Design*, vol. 12, no. 6, pp. 861–877, June 1993.
- [27] K. S. Chao and R. Saeks, "Continuation methods in circuit analysis," Proceedings IEEE, vol. 65, pp. 1187–1194, 1977.
- [28] S. L. Richter and R. A. DeCarlo, "Continuation methods: theory and applications," *IEEE Trans. Circuits Syst.*, vol. 30, pp. 347–352, 1983.
- [29] P. Wambacq and W. Sansen, *Distortion Analysis of Analog Integrated Circuits*. The Netherlands: Kluwer Academic Publishers, 1998.
- [30] A. S. Sedra and K. Smith, *Microelectronic Circuits*. 198 Madison Avenue, New York, NY: Oxford University Press, 2004.
- [31] J. Rogers and C. Plett, *Radio Frequency Integrated Circuit Design*. Norwood, MA: Artech House, Inc., 2003.
- [32] P. Li and L. T. Pileggi, "Nonlinear distortion analysis via linear-centric models," in DAC, 2003, pp. 897–903.
- [33] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1970.
- [34] W. J. McCalla, Fundamentals of Computer-Aided Circuit Simulation. Norwell, MA: Kluwer, 1988.
- [35] M. Gourary, S. Rusakov, S. Ulyanov, M. Zharov, K. Gullapalli, and B. Mulvaney, "New computational technique for periodic distortion analysis of communication circuits," in *IEEE MTT-S Int. Microwave Symp. Dig.*, 2004.
- [36] J. Cherry, "Distortion analysis of weakly nonlinear filters using volterra series," Masters Thesis, Carleton University, Ottawa ON, Canada, 1994.
- [37] L. A. Zadeh, "Frequency analysis of variable networks," *Proceedings IRE*, vol. 32, pp. 291–299, 1950.

- [38] F. Yuan and A. Opal, "Distortion analysis of periodically switched nonlinear circuits using time-varying volterra series," *IEEE Transactions on Circuits and Systems-I*, vol. 48, pp. 726–738, 2001.
- [39] R. Griffith and M. Nakhla, "A new high order absolutely stable explicit numerical integration algorithm for the time domain simulation of nonlinear circuits," in *Proc. ACM ICCAD*, 1997, pp. 504–509.
- [40] B. Razavi, RF Microelectronics. New Jersey: Prentice-Hall, 1998.
- [41] P. Wambacq, G. G. E. Gielen, P. R. Kinget, and W. Sansen, "High frequency distortion analysis of analog integrated circuits," *Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, 1999.