

Efficient Stochastic Collocation based Variability Analysis using Model-Order Reduction Techniques

by

Kai Guo, B.Eng

A thesis submitted to the Faculty of Graduate Studies and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Electronics

Faculty of Engineering

Carleton University

Ottawa, Ontario, Canada

September 2016

Copyright © 2016

Kai Guo

Abstract

Predicting the effect of the variability of design parameters on the performance of high-speed integrated circuits is crucial to a successful design. The conventional Monte Carlo technique is computationally expensive due to the large number of simulations and a slow convergence rate.

To address the above difficulties, a novel method is presented in this thesis for time-domain stochastic analysis of large active/passive circuits with multiple stochastic parameters. The new approach reduces the computational cost of variability analysis by using the Stochastic Collocation technique. The Sparse Grid algorithm is applied to limit the growth of the computational cost with an increase in the number of stochastic parameters. In addition, the proposed method is based on the Model Order Reduction algorithms coupled with the Numerical Inverse Laplace Transform approach. The proposed method preserves the passivity of the originally passive circuits. It is also applicable to active circuits and preserves their stability by construction. Hence, the feasibility of the nodal reduced-order models is ensured for the transient analysis using Numerical Inverse Laplace Transform. Pertinent numerical results are presented to demonstrate the computational efficiency of the proposed method and its performance in handling the case of multiple stochastic parameters.

Acknowledgments

My deepest appreciation goes first and foremost to my supervisor, Professor Michel Nakhla, for his constant encouragement and guidance. He led me to this fascinating research area and walked me through all the stages of this work.

Second, I would like to give my gratitude to my co-supervisor, Dr. Behzad Nouri, for his consistent and illuminating instruction. This thesis could not have reached its present form without his encouragement and help.

I would also like to thank Dr. Francesco Ferranti for clarifying my ideas with his deep insight of the field, as well as helpful comments and feedback that greatly improved the quality of this work.

I also acknowledge my colleagues in our CAD group, for their valuable suggestions and help, not only in this work, but also in the 2 past years of my graduate life at Carleton. Their friendship is a precious treasure in my life.

Last but not least, I would like to give my special thanks to my family, especially my parents and my wife. I am very lucky to have their unconditional love and support, which is always the greatest spiritual power for me to overcome all difficulties. I dedicate this work to them.

Table of Contents

Abstract	ii
Acknowledgments	iii
Table of Contents	iv
List of Tables	viii
List of Figures	ix
List of Acronyms	xii
List of Symbols	xiv
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives and Contributions	3
1.3 Organization of the Thesis	4
2 Stochastic Circuit Simulation	5
2.1 Fundamentals of Circuit Formulation	6
2.1.1 Modified Nodal Analysis of Linear Circuits	6
2.1.2 Stability	8

2.1.3	Passive and Active Circuits	12
2.2	Stochastic Analysis of Parametrized Circuits	13
2.2.1	Probability Density Function	14
2.2.2	Moments of Stochastic Variables	15
2.3	The Monte Carlo Technique	17
2.3.1	Summary	18
3	Model Order Reduction Techniques	20
3.1	Projection-Based Model Order Reduction	21
3.2	Krylov-Subspace Using Moment-Matching Techniques	23
3.2.1	Arnoldi-Process	24
3.3	Krylov-Based Model Order Reduction	25
3.3.1	PRIMA	26
3.3.2	A Stability Guaranteed Algorithm for Active Circuits	29
3.4	Moder Order Reduction of Parametrized System	30
3.5	Summary and Problem Definition	33
4	Polynomial Interpolation based Variability Analysis using PMOR	35
4.1	Problem Decomposition	36
4.2	Stochastic Simulation based on Polynomial Interpolation	36
4.2.1	Polynomial Interpolation	36
4.2.2	Sampling Scheme	38
4.2.3	Lagrange Basis Function	39
4.2.4	Stochastic Analysis Using Polynomial Interpolation	40
4.3	Generation of Reduced Order Model	41
4.3.1	Generation of Two-Sided Projection Matrices	42
4.4	Transient Analysis Using Numerical Integration Techniques	43

4.5	Numerical Validation	44
4.5.1	Example 1	44
4.6	Summary	51
5	Variability Analysis Using Sparse-grid Stochastic Collocation Techniques	52
5.1	Quadrature Integration	53
5.2	Node Selection Based on Nested Quadrature Rule	57
5.2.1	Krnord-Patterson’s Nested Quadrature Rule	58
5.2.2	Generation of Nodes and Weights	59
5.2.3	Multivariate Stochastic Collocation Scheme Using Tensor Product Grids	61
5.3	Nodes Minimization Using Smolyak’s Sparse Grids	62
5.4	Numerical Validation	65
5.5	Summary	70
6	Time-domain Analysis Using Numerical Inverse Laplace Transform	71
6.1	Numerical Inverse Laplace Transform	72
6.2	High Order NILT	75
6.3	Implementation of NILT	78
6.4	Numerical Validation	80
7	Numerical Results	83
7.1	Example 1	83
7.2	Example 2	88
8	Conclusion and Further Work	92
8.1	Conclusion	92

8.2 Future Work	93
List of References	94

List of Tables

4.1	A set of data evaluated at distinct values of x	37
4.2	CPU time comparison	48
5.1	Classical orthogonal polynomials.	55
5.2	Nodes and weights of the Hermite-Gauss quadrature rule	56
5.3	CPU time comparison of example 1 (SC)	69
7.1	Sparse grid nodes and weights in the six dimensional parameter space	85
7.2	CPU time comparison of example 1	86
7.3	Sparse grid nodes and weights in the ten dimensional parameter space	89
7.4	CPU time comparison of example 2	90

List of Figures

2.1	A circuit with input $U(s)$, output $Y(s)$ and transfer function $H(s)$. . .	9
2.2	Normal Distribution.	15
4.1	A stable active circuit consisting of two coupled transmission lines and two amplifiers.	45
4.2	Cross section of the transmission lines	45
4.3	The internal circuit of the amplifiers.	46
4.4	Equidistant nodes spaced on the 2-dimensional random space.	46
4.5	Frequency analysis at V_{out1} for the circuit operating at the nominal parameter values.	47
4.6	Frequency analysis at V_{out2} for the circuit operating at the nominal parameter values.	48
4.7	Transient analysis at V_{out1} for the circuit operating at the nominal parameter values.	49
4.8	Transient analysis at V_{out2} for the circuit operating at the nominal parameter values.	49
4.9	Time-domain statistical analysis for the output voltage at V_{out1} . . .	50
4.10	Time-domain statistical analysis for the output voltage at V_{out2} . . .	50
5.1	A two-dimensional sparse grid of accuracy level 4 constructed using the combination technique.	65

5.2	Time-domain statistical analysis at Vout1 using MC technique.	66
5.3	Time-domain statistical analysis at Vout2 using MC technique.	67
5.4	Two-dimensional sparse grids based on the Hermite Genz-Keister quadrature rule.	67
5.5	Time-domain statistical analysis at Vout1 using sparse grid SC tech- niques.	68
5.6	Time-domain statistical analysis at Vout2 using sparse grid SC tech- niques.	69
6.1	An illustration of NILT that the time response at one time point is a summation of frequency responses at several complex frequencies. . .	73
6.2	Transient analysis at Vout1 using NILT0.	81
6.3	Transient analysis at Vout2 using NILT0.	81
6.4	Transient analysis at Vout1 using NILT0 and NILT1.	82
6.5	Transient analysis at Vout2 using NILT0 and NILT1.	82
7.1	A stable active circuit consisting of four coupled transmission lines and four amplifiers.	84
7.2	Cross section of the transmission lines.	84
7.3	The internal circuit of the amplifiers.	85
7.4	Time-domain statistical analysis of the output voltage at the far end of line 1.	86
7.5	Time-domain statistical analysis of the output voltage at the far end of line 2.	87
7.6	A stable active circuit consisting of eight coupled transmission lines and eight amplifiers.	88
7.7	Cross section of the transmission lines.	88

7.8	Time-domain statistical analysis of the output voltage at the far end of line 1.	90
7.9	Time-domain statistical analysis of the output voltage at the far end of line 2.	91

List of Acronyms

Acronyms	Definition
AWE	Asymptotic Waveform Evaluation
CAD	Computer Aided Design
BE	Backward Euler
CPU	Central Processing Unit
DAE	Differential-Algebraic Equations
EM	Electro-Magnetic
GLI	Generalized Lyapunov Inequality
HGK	Hermite Genz-Keister
IC	Integrated Circuit
LTI	Linear Time Invariant
MC	Monte Carlo
MNA	Modified Nodal Analysis

MMTs	Moment Matching Techniques
MOR	Model Order Reduction
MTL	Multi-conductor Transmission Line
NILT	Numerical Inverse Laplace Transform
PC	Polynomial Chaos
PDF	Probability Density Function
PMOR	Parametrized Model Order Reduction
PRIMA	Passive Reduced-order Interconnect Macro-modeling Algorithm
PSD	Positive Semi-definite
SPICE	Simulation Program with Integrated Circuit Emphasis
SC	Stochastic Collocation
SG	Sparse Grid
ROMs	Reduced Order Models
TL	Transmission Line
TR	Trapezoidal Rule
VLSI	Very Large-Scale Integration

List of Symbols

Symbols	Definition
μ	mean value
σ	standard deviation
var	variance
ξ	single stochastic variable
ξ	set of multiple stochastic variables
$H_i(\xi)$	Hermite polynomial of degree “i” where $i = 0, 1, \dots$
$p(\xi)$	weighting function
Ω	multi-dimensional random space
G	matrix describing the memoryless elements in the circuit
C	matrix describing the memory elements in the circuit
$X(s)$	vector of state variables in frequency domain
$U(s)$	vector of circuit input in frequency domain

$Y(s)$	vector of circuit output in frequency domain
$*$	complex conjugate operator
\otimes	the Kronecker product operator
Λ	polynomial basis
$P_j(\xi)$	Lagrange basis function
$\hat{\Psi}$	matrix contains the q generalized eigenvectors of the matrix pencil $(-\mathbf{G}^T, \mathbf{C}^T)$
h	step size of the transient simulation
λ	diagonal matrix whose diagonal elements are eigenvalues of matrix A
I_n	identity matrix of size $n \times n$
V_k	Krylov subspace
H_U	upper-Hessenberg matrix
$W(\xi)$	set of quadrature weights
\mathbb{E}	expectation operator
Q_l^d	a d -dimensional quadrature rule with accuracy level of l
\mathcal{F}_l^d	a d -dimensional polynomial interpolation function with accuracy level of l
Γ_q^d	support nodes set

Chapter 1

Introduction

1.1 Background and Motivation

In the world of today's technology, electronic systems, such as computers and mobile devices, have become fundamental parts of our daily lives. Moore's law predicts that the number of transistors in integrated circuits doubles approximately every two years, and this prediction has been proven accurate for decades. With the remarkable development of very large-scale integration (VLSI) and computer-aided design (CAD) technologies, the electronic systems are enabled to operate up to gigahertz frequencies and transfer high-speed signals with rise times at the nanosecond level. The ever-increasing complexity and packaging level of circuits make precision in the design and manufacturing process of vital significance.

Simulation of electronic circuits and systems allows the designer to refine the design economically and efficiently. This is done by solving the mathematical model formulated using a set of circuit equations in the time/frequency domain to obtain the response of interest. However, in practice, the circuits' models may not only have a large size, but may also be dependent on a set of parameters. This means that, in addition to the time/frequency variables, there are uncertain physical and

geometrical parameters in a broad class of designs. These parameters generally come from the design and manufacturing phases and can substantially impact the circuits' responses. Hence, predicting the effect of the variation of design parameters on the performance of high-speed integrated circuits is crucial for a successful design.

To this end, Monte Carlo analysis has been conventionally used in commercial simulators. However, it requires a large number of simulations and has a slow convergence rate. Variability analysis techniques based on the notion of Polynomial Chaos (PC) have been developed over the years [1–3]. These methods can lead to excessively large augmented systems, which are prohibitively expensive to solve.

Another class of techniques based on the Stochastic Collocation (SC) method has been developed in [4,5]. These techniques offer high accuracy, fast convergence rate, and easy implementation. However, they can be computationally expensive for an increasing number of stochastic parameters with large variations. The Sparse Grid (SG) algorithms are used to improve the efficiency of SC-based methods [6]. However, the associated CPU-cost may still be high for large circuits with multiple stochastic parameters which are generally encountered in modern designs.

To resolve these issues, a novel approach is presented in this thesis based on Sparse-grid Stochastic Collocation, Model Order Reduction (MOR), and application of Numerical Inverse Laplace Transform (NILT) algorithms. The advantage of the proposed method is threefold. Firstly, the unknown stochastic solutions are accurately approximated by a SC scheme through a polynomial interpolation function in the multidimensional parameter space. The statistical information of the circuits' responses is efficiently evaluated using the quadrature integration approach. In addition, the number of support nodes in the SC scheme is efficiently reduced by using the SG techniques. Secondly, the responses at the grid nodes of the SC scheme are efficiently obtained using reduced-order deterministic models. Thirdly, the NILT

algorithm is used for the efficient transient analysis of the reduced models.

1.2 Objectives and Contributions

This thesis presents a novel methodology for efficient time-domain variability analysis of both large active and passive circuits with multiple stochastic parameters [7]. The main contributions in this thesis can be explained in connection with the development of the following ideas:

- Decoupling the parameterized circuits by establishing a proper grid structure, and constructing the deterministic systems at the grid nodes (nodal systems);
- Application of MOR algorithms adapted to active or passive parametrized circuits;
- Adapting the Stochastic Collocation techniques to obtain the statistical information efficiently;
- Adapting Sparse Grid algorithms to minimize the number of grid nodes for the case of multiple stochastic parameters;
- Adapting the NILT algorithm to efficiently obtain the transient response of the nodal systems.

The contribution presented in this thesis directly leads to a method for variability analysis of parameterized circuits ensuring accurate results and significantly reduced computational effort. In addition, the proposed method is applicable to both active and passive circuits. It is also highly parallelizable.

1.3 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 reviews the background of the parameterized circuit formulation and the variability analysis. The difficulties of the conventional Monte Carlo approach are discussed as well. Chapter 3 presents a review of the widely used Model Order Reduction techniques. The limitation of applying these reduction techniques to parameterized circuits is discussed. Chapter 4 presents the first key feature of the proposed algorithm, which is the variability analysis of parameterized circuits using a polynomial interpolation approach. Following to this, the second feature, a scheme based on the Sparse Grid Stochastic Collocation technique is discussed in Chapter 5. An efficient transient analysis method and its application for improving the proposed method is presented in Chapter 6. The numerical results in Chapter 7 demonstrate the accuracy and efficiency of the proposed method. Finally, Chapter 8 provides concluding remarks and future research options.

Chapter 2

Stochastic Circuit Simulation

Stochastic (or variability) analysis is a widely used technique in the applied science and engineering fields. Since the physical world is full of uncertainty, purely deterministic systems are impossible to find and usually considered to be approximations of true physics. In the circuit design and manufacturing domains, parameters enter the circuit from many sources. Today's advanced technologies have remarkably minimized the variation of these parameters. However, any unnegligible impact of these random parameters on the entire circuit's performance may lead to costly re-fabrication or even a ruined design. Hence, stochastic analysis methods for predicting the circuit's performance and error scale under the variation of parameters are in great demand. A multitude of research has been done and yet this topic is still considered an open issue.

Several techniques are available in the literature to serve this purpose. Monte-Carlo based methods have been the method of choice in many commercial simulators. This chapter presents a review of the conventional Monte Carlo method.

The rest of this chapter is organized as follows: section 2.1 gives the fundamentals of the mathematical modeling method of circuits, section 2.2 discusses the problem of stochastic variability analysis, and section 2.3 reviews the traditional Monte Carlo

based technique.

2.1 Fundamentals of Circuit Formulation

Nowadays, the ever increasing scale and complexity of circuit systems in the CAD field make an appropriate modeling method of key importance. Modified Nodal Analysis (MNA) has been widely used due to its simplicity and efficiency [8]. The main idea of this method is stamping the circuit components, nodal voltages and branch currents into system matrices. The electrical information of the circuit is converted to an appropriate location and value in the matrices. Therefore, the construction of MNA circuit equations is done by self-consistent and repeatable stamps, which can be simply programmed and iterated. In addition, for SPICE simulators such as PSPICE and HSPICE, the transformation between netlists and MNA equations is effective and convenient.

Since this thesis focuses on linear circuits, we assume all the following formulations of circuits are linear in state, while from a practical point of view, nonlinear dependencies of parameters are allowed.

2.1.1 Modified Nodal Analysis of Linear Circuits

Firstly, we follow the concept of MNA for deterministic circuit networks. Given a general Linear Time Invariant (LTI) circuit, its time-domain MNA equations are expressed as follows:

$$Gx(t) + C \frac{dx(t)}{dt} = Bu(t) \tag{2.1}$$

$$y(t) = Lx(t)$$

where $x(t) \in \mathbb{R}^n$ is a vector of circuit variables, including nodal voltages and branch currents in the circuit as unknowns. The time sequence of the unknown-vector values is called the trajectory, which describes the behavior of the circuit over a certain time region. $u(t)$ and $y(t)$ are the vectors including m_i input excitations and m_o outputs, respectively. $G, C \in \mathbb{R}^{n \times n}$ are conductance and susceptance matrices, while $B \in \mathbb{R}^{n \times m_i}$ and $L \in \mathbb{R}^{m_o \times n}$ represent input and output selection matrices, respectively. n defines the number of variables, and is also called the order of the circuit.

Let variable $s \in \mathbb{C}$ denote the complex frequency, then (2.1) can be converted into the complex frequency domain by applying the Laplace transform as

$$GX(s) + sCX(s) = BU(s) + X(0) \quad (2.2)$$

$$Y(s) = LX(s)$$

where $X(0)$ is the initial condition that describes the energy stored in memory components such as capacitors and inductors. Therefore, given zero initial condition, (2.2) is rewritten as (2.3)

$$GX(s) + sCX(s) = BU(s) \quad (2.3)$$

$$Y(s) = LX(s)$$

Several important properties are preserved by the MNA equation with the structure given as

$$G = \begin{bmatrix} F & E_a \\ E_b & 0 \end{bmatrix}, C = \begin{bmatrix} C_a & 0 \\ 0 & C_b \end{bmatrix}, x = \begin{bmatrix} v \\ i \end{bmatrix} \quad (2.4)$$

where F, C_a and C_b are symmetric and diagonally dominant matrices that contain

information of the resistance, capacitance and inductance of the circuit respectively. As a result, the matrix C is symmetric and diagonally dominant, and we have $C_a \geq 0$ and $(-C_b) \geq 0$. E_a and E_b are matrices that only contain zeros and ± 1 . Note that for a passive circuit, we obtain a symmetric matrix $F \geq 0$ and $E_a = E_b^T$, which leads to a symmetric system matrix G . For the case of active circuits, G loses its overall symmetry. The concept of passive and active circuits will be reviewed in the rest of this section. Another important property is the high sparsity of the system matrices; in other words, most of the elements in these system matrices are zeros. These properties provided by the MNA formulation are of key significance for many fast simulation algorithms.

Next, we will review several important properties of the circuit system.

2.1.2 Stability

A general dynamic circuit system is said to be stable if the trajectory does not go to infinity under perturbations. In circuit analysis, the perturbations usually come from the external perturbation which is called the input (or source), and the initial condition which describes the state of the circuit at the initial time point. The infinite output of a circuit system is obviously not expected from a practical point of view. Hence, stability is the most important property of physical designs that should be preserved in the models.

A frequency domain deterministic circuit with zero initial condition is given in (2.3). To simplify the notation, we use X , Y and U instead of $X(s)$, $Y(s)$ and $U(s)$,

then, we obtain

$$\begin{aligned}(G + sC)X &= BU \\ X &= (G + sC)^{-1}BU \\ Y &= LX\end{aligned}\tag{2.5}$$

Let I denote the unity matrix. From (2.5), we have

$$\begin{aligned}X &= [G(I + sG^{-1}C)]^{-1}BU \\ Y &= LX = L[(I + sG^{-1}C)^{-1}G^{-1}B]U\end{aligned}\tag{2.6}$$

A transfer function $H(s)$ is a mathematical representation from input to output of a circuit system in frequency domain (see figure 2.1).

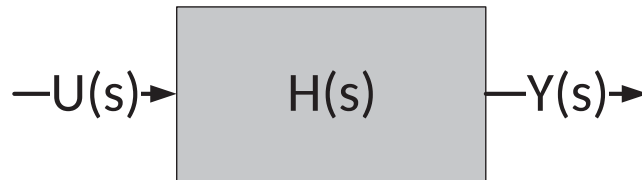


Figure 2.1: A circuit with input $U(s)$, output $Y(s)$ and transfer function $H(s)$.

From (2.6), we have

$$\begin{aligned}Y &= HU \\ H &= L(I + sG^{-1}C)^{-1}G^{-1}B\end{aligned}\tag{2.7}$$

Substituting $A = -G^{-1}C$ and $R = G^{-1}B$ into (2.7) yields

$$H = L(I - sA)^{-1}R\tag{2.8}$$

diagonalizing A as $A = P\lambda P^{-1}$, we get

$$\begin{aligned}
I - sA &= I - sP\lambda P^{-1} \\
&= PP^{-1} - sP\lambda P^{-1} \\
&= P(I - s\lambda)P^{-1} \\
&= P \left(\begin{array}{cccc} 1 - s\lambda_1 & & & \\ & \ddots & & \\ & & 1 - s\lambda_i & \\ & & & \ddots \\ & & & & 1 - s\lambda_n \end{array} \right) P^{-1} \tag{2.9}
\end{aligned}$$

where P is a regular matrix and λ is a diagonal matrix with the diagonal entries $\{\lambda_i\}_{i=1}^n$, which are called the eigenvalues of matrix A .

According to the diagonalized form in (2.9), the inverse of $(I - sA)$ is found as

$$(I - sA)^{-1} = P(I - s\lambda)^{-1}P^{-1} \tag{2.10}$$

Then substituting (2.10) into (2.8) yields

$$\begin{aligned}
 H &= LP(I - s\lambda)^{-1}P^{-1}R \\
 &= LP\left(\begin{bmatrix} 1 - s\lambda_1 & & & \\ & \ddots & & \\ & & 1 - s\lambda_i & \\ & & & \ddots \\ & & & & 1 - s\lambda_n \end{bmatrix} \right)^{-1}P^{-1}R \\
 &= LP\left(\begin{bmatrix} \frac{1}{1 - s\lambda_1} & & & \\ & \ddots & & \\ & & \frac{1}{1 - s\lambda_i} & \\ & & & \ddots \\ & & & & \frac{1}{1 - s\lambda_n} \end{bmatrix} \right)P^{-1}R
 \end{aligned} \tag{2.11}$$

Each element of the transfer function H in (2.11) is a rational function of s. The denominator polynomial is common to all elements of the transfer function H and it is the common denominator between the diagonal elements in λ denoted by D given as

$$D = (1 - s\lambda_1) \times \cdots (1 - s\lambda_i) \times \cdots (1 - s\lambda_n) \tag{2.12}$$

Therefore, the i -th pole of the transfer function H is defined by the roots of denominator polynomial D given as

$$p_i = \frac{1}{\lambda_i}, \quad i = 1, \dots, n \text{ and } \lambda_i \neq 0$$

where λ is also known as the eigenvalues of matrix $A = -G^{-1}C$. The poles represent the values of frequency s that lead to infinite solutions. Using the poles of a circuit, the following theorem defines the stability of a circuit system.

Theorem 1.

1. *A system is stable if and only if all poles are on the closed left-half-plane of the complex domain, and the pure imaginary poles do not have multiplicity.*
2. *A system is asymptotically stable (A-stable) if and only if all poles of the system are on the open left-half-plane of the complex domain.*

2.1.3 Passive and Active Circuits

A physical system is denoted as passive when it is unable to generate energy. The precise mathematical definition of passivity depends on the representation adopted for an n-port network. For impedance or admittance representations we have [9]

Definition 1. An n-port network is said to be passive if

$$\int_{-\infty}^t i(\tau)v^T(\tau)d\tau \geq 0 \tag{2.13}$$

for all t and all admissible port voltages $v(t)$ and currents $i(t)$.

Passivity is a stricter notion compared to stability. Stability is a necessary but not sufficient condition for a passive system, in other words, a passive circuit must be stable, however, a stable circuit may be active. Given a circuit described with MNA equations, the following properties can be obtained for passive circuits, where their

conductance and susceptance matrices are respectively constructed as:

$$G = \begin{bmatrix} F & E_a \\ E_a^T & 0 \end{bmatrix}, C = \begin{bmatrix} C_a & 0 \\ 0 & C_b \end{bmatrix} \quad (2.14)$$

1) $C_a \geq 0, -C_b \geq 0$;

2) $G + G^T \geq 0$.

The latter is also referred to as the contaction property which ensures the stability of the system. The above mentioned properties are of great importance for passivity preservation in MOR techniques such as PRIMA.

However, a practical circuit usually contains components such as dependent sources that lead to active systems. A circuit is said to be active if the integral in (2.13) is less than zero, which means the circuit produces energy. The MNA formulation of active circuits does not hold the second condition above. This is due to the fact that, in the presence of the dependent sources, the lower left block of matrix G will not be E_a^T anymore. This leads to difficulties in most of the model reduction techniques and will be further discussed in chapter 3.

In the next section, we will provide the problem definition of stochastic analysis of parametrized circuits.

2.2 Stochastic Analysis of Parametrized Circuits

Theories and technologies for analyzing deterministic systems have been well developed in the past decades. However, in practice, uncertain parameters from the design and fabrication phases effect the circuit's transient behavior, leading to a noticeable

deviation from the intended deterministic trajectory. In MNA formulation, this uncertainty is modeled by including additional variables in the system matrices in (2.1).

Given a parameterized circuit dependent on d design parameters $\boldsymbol{\xi} = (\xi_1, \dots, \xi_d)$, the notion of uncertainty in the circuit response due to the uncertain parameters is captured by adapting the MNA formulation as follows

$$\begin{aligned} G(\boldsymbol{\xi})x(t, \boldsymbol{\xi}) + C(\boldsymbol{\xi})\frac{dx(t, \boldsymbol{\xi})}{dt} &= Bu(t) \\ \mathbf{y}(t, \boldsymbol{\xi}) &= Lx(t, \boldsymbol{\xi}) \end{aligned} \tag{2.15}$$

Similarly by using the Laplace transform, the complex frequency domain MNA equation of the parametric linear circuit in (2.15) is obtained as

$$\begin{aligned} G(\boldsymbol{\xi})X(s, \boldsymbol{\xi}) + sC(\boldsymbol{\xi})X(s, \boldsymbol{\xi}) &= BU(s) \\ Y(s, \boldsymbol{\xi}) &= LX(s, \boldsymbol{\xi}) \end{aligned} \tag{2.16}$$

In stochastic circuit analysis, we are interested in solving a set of parametrized differential equations to obtain the state trajectories under the variation of parameters.

2.2.1 Probability Density Function

Unlike the time or frequency variable, the design parameters in (2.16) are stochastic (or random) variables whose values are subject to variations due to probability. For instance, the width of a transmission line may vary around an expected value (nominal value) due to the fabrication precision. In probability theory, the function describing the probability of a continuous stochastic variable distributed over a certain range is known as a probability density function (PDF) [10]. An example is a stochastic variable ξ with the normal (Gaussian) distribution $\xi \in N(\mu, \sigma^2)$. The

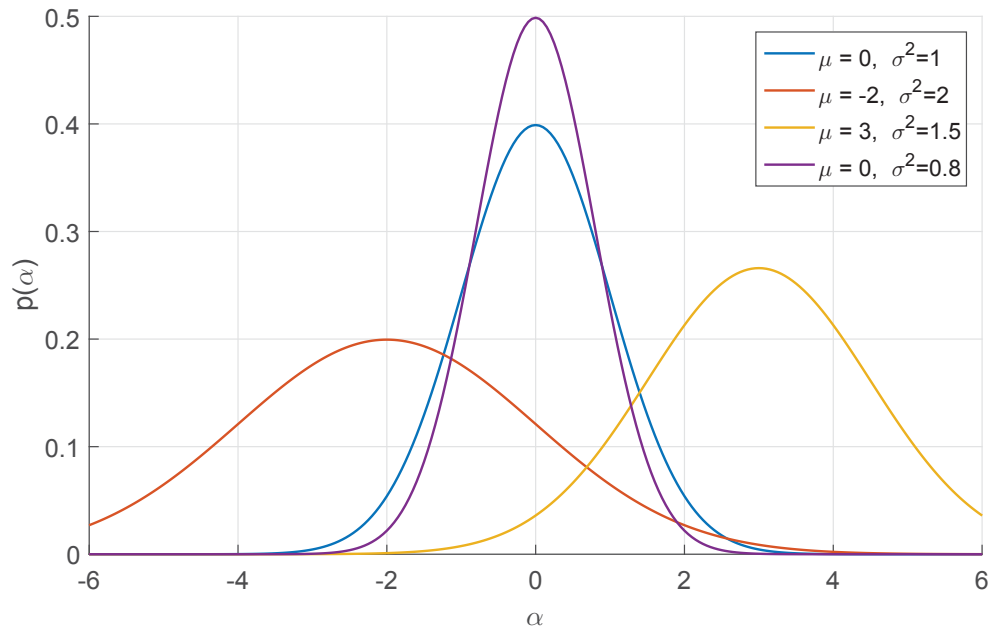


Figure 2.2: Normal Distribution.

density function $p(\xi)$ of ξ is given as

$$p(\xi; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\xi - \mu)^2}{2\sigma^2}\right)$$

where μ and σ are the first and second order moments that describe the statistical information of ξ . Figure (2.2) gives examples of normally distributed random variables with different μ and σ .

2.2.2 Moments of Stochastic Variables

In statistics, a moment is a specific quantitative measure of the distribution shape of a set of random values. The first order stochastic moment (also called expected value or mean) describes the center value of the variable, with the notation of μ given as

follows

$$\mu(\xi) = \int_{-\infty}^{\infty} \xi p(\xi) d\xi \quad (2.17)$$

If $f(\xi)$ is a measurable function of ξ , the expectation of this function is

$$\mu(f) = \int_{-\infty}^{\infty} f(\xi) p(\xi) d\xi \quad (2.18)$$

Let \mathbb{E} denote the expectation operator, then the general $m - th$ order ($m \geq 2$) non-central moment is represented as

$$\mathbb{E}(\xi^m) = \int_{-\infty}^{\infty} [\xi^m] p(\xi) d\xi \quad (2.19)$$

and the central moment is evaluated as

$$\mathbb{E}(\xi^m) = \int_{-\infty}^{\infty} [\xi - \mathbb{E}(\xi)]^m p(\xi) d\xi \quad (2.20)$$

The second order moment describes the variance (\mathbf{Var}) and its square root (σ) describes the standard deviation.

$$\begin{aligned} \mathbf{Var}(\xi) &= \mathbb{E}[(\xi - \mathbb{E}(\xi))^2] \\ &= \mathbb{E}(\xi^2) - \mathbb{E}(\xi)^2 \end{aligned} \quad (2.21)$$

$$\sigma(\xi) = \sqrt{\mathbf{Var}(\xi)} \quad (2.22)$$

The moments of the stochastic simulation results provide valuable information for designers to predict the circuit's input-output behavior and error scale. For instance,

the widely used empirical rule is that, for a normally distributed variable, about 68%, 95% and 99.7% of its values are within σ , 2σ , and 3σ away from the mean value, respectively.

The Monte Carlo technique presented in the next section is known as the conventional approach for solving parametrized circuit equations and evaluating the stochastic moments of the responses.

2.3 The Monte Carlo Technique

Monte Carlo simulation is one of the most commonly used methods for statistically analyzing the parametrized circuits. It provides solid results and simplicity of implementation, which makes it a preferred approach. This popularity is despite the fact that, this method is computationally expensive because it requires a large number of samples and has a slow convergence rate. The main idea of the MC approach is creating a discrete sample set of the parameters, then forming the deterministic systems at all the sample values and simulating each system to collect the output of interest as a set of response data. The statistical information is obtained through computing the moments of the sample set of responses. We should note that the number of the samples must be sufficiently large such that the results are statistically valid.

Consider a univariate parameterized circuit sampled at a sample set $\xi = \{\xi_i\}_{i=1}^N$, and by simulating all the samples, we obtain the data $\mathbf{y}(t, \xi) = \{\mathbf{y}(t, \xi_i)\}_{i=1}^N$. Then, the sample mean and standard deviation is calculated as follows

$$\mu[\mathbf{y}] \approx \frac{1}{N} \sum_{i=1}^N \mathbf{y}(t, \xi_i) \quad (2.23)$$

$$\sigma(\mathbf{y})^2 \approx \frac{1}{N} \sum_{i=1}^N [\mathbf{y}(t, \xi_i) - \mu(\mathbf{y})]^2 = \frac{1}{N} [\mu(\mathbf{y}^2) - \mu(\mathbf{y})^2] \quad (2.24)$$

where $\mu(\mathbf{y}^2)$ is the mean value of \mathbf{y}^2 . Note that equation (2.24) is also known as the uncorrected or biased standard deviation. This is because the number of the samples is always insufficient in the real world. Hence, we select the corrected or unbiased formula to evaluate the sample standard deviation given as

$$\sigma(\mathbf{y})^2 \approx \frac{1}{N-1} \sum_{i=1}^N [\mathbf{y}(t, \xi_i) - \mu(\mathbf{y})]^2 \quad (2.25)$$

A typical Monte Carlo based method for stochastic circuit analysis basically takes the following steps:

1. Set the number of the random simulations N (MC runs); this number should be statistically large.
2. For each $i = 1, \dots, N$, generate the N random values of the physical parameter vector $\{\xi_i\}_{i=1}^N$ according to the associated probability density function of the parameters.
3. For each $\{\xi_i\}_{i=1}^N$, construct and solve the corresponding MNA equations to obtain the output trajectory $\{y(t, \xi_i)\}_{i=1}^N$
4. Evaluate the stochastic information using solutions from step 3.

2.3.1 Summary

The above procedures clearly show the true-sampling based Monte Carlo technique is the most straightforward and simple approach which provides accurate statistical information for stochastic circuit analysis. As a result, Monte Carlo simulation is commonly used in commercial simulators due to its validity, reliability and simplicity. In this thesis, the result from MC is used as an authoritative reference in comparisons

with the proposed method.

However, the drawbacks of the MC approach are quite obvious as well. Firstly, the accuracy of the solutions is highly dependent on the large number of the samples, especially for the case of multiple parameters. Moreover, for circuit simulation under the variation of random parameters, the answer to the question “how many runs of MC is statistically enough” is always unknown. The solution converges slowly as the number of samples increases, which usually leads to an extra step to verify the convergence using more samples. These drawbacks usually result in a huge number of realizations and simulations of large-size circuit networks.

To address the above difficulties, it is natural to select the powerful tool known as Model Order Reduction techniques. A review of these methods is provided in the next chapter.

Chapter 3

Model Order Reduction Techniques

In the past several decades, the Model Order Reduction (MOR) approach has demonstrated its effectiveness and flexibility for application in the simulation of large scale circuit networks. It has a multitude of applications in the circuit design and analysis fields. The delightful idea of these techniques is to approximate the input-output behavior of a given large-scale circuit by using a quite smaller model [11]. Hence, the computational effort is significantly reduced and a high level of accuracy is preserved as well. It is to be noted that, the resulting Reduced Order Models (ROMs) must be efficient enough. This implies, under an acceptable error scale, the size of the ROMs should be as small as possible. In addition, the stability and accuracy of the original model should be preserved in the order reducing process.

Model order reduction of linear systems has been extensively studied in the last two decades. As a result, the MOR techniques for macro-modeling and efficient analysis of non-parametrized linear circuits have been very well established [12]. For the case of originally passive circuits, there are efficient MOR algorithms for preserving the passivity of the reduced models. The stability is also guaranteed by these methods due to the fact that passivity is a stronger notion compared to stability. On the other hand, these algorithms fail in constructing stable ROMs for active circuits. Since

stability is crucial for the feasibility of the ROMs for transient analysis, there are efforts that have been reported in the literature to ensure the stability of the reduced model. They are mostly based on post-processing and optimization [13–16]. However, these methods are inefficient due to the sacrificed accuracy or additional computation effort required for stability preservation of active networks. To address these issues of accuracy and model-generation efficiency degradation, recently, a new algorithm is proposed in [17], which preserves the stability by construction.

As an extension to parametrized circuits, the application of MOR techniques to circuits’ variability analysis has been widely studied and yet is still considered an active topic. This area will be further explored in the next chapters of this thesis.

In this chapter, a review of the popular projection-based MOR methods for deterministic circuits with a focus on the Krylov-based Moment-Matching Techniques (MMTs) is presented. Next, the existing methods for parametrized MOR are reviewed. Then, the main objective, which is applying parameterized MOR techniques to stochastic analysis is discussed.

3.1 Projection-Based Model Order Reduction

First, we consider a linear circuit network with no parameter-dependency to provide a review of the MOR techniques. Most of the existing reduction approaches share the common idea of projection. Given the circuit formulated using MNA equations as

$$Gx(t) + C \frac{dx(t)}{dt} = Bu(t) \tag{3.1}$$

$$y(t) = Lx(t)$$

where $x(t) \in \mathbb{R}^n$ describes a trajectory in a n -dimensional state-space, the number of dimensionality n is also known as the system's order, which usually exceeds the thousands-scale in today's circuit design field. As a result, the large size of the system equations leads to high computational effort, which makes an effective approximation necessary. To address this problem, projection based MOR methods approximate the state trajectory by point-wise projecting the trajectory onto a precomputed low-dimensional subspace $z(t) \in \mathbb{R}^q$ using a right projection matrix $V \in \mathbb{R}^{n \times q}$, $q \ll n$ (reduced basis)

$$\begin{aligned} x(t) &\approx Vz(t) \\ GVz(t) + C\frac{dVz(t)}{dt} &= Bu(t) \\ y(t) &= LVz(t) \end{aligned} \tag{3.2}$$

Defining the residual error as

$$e = GVz(t) + C\frac{dVz(t)}{dt} - Bu(t) \tag{3.3}$$

Then, a left projection matrix M is used to minimize the residual error through enforcing the Petrov-Galerkin condition

$$M^T e = M^T [GVz(t) + C\frac{dVz(t)}{dt} - Bu(t)] = 0 \tag{3.4}$$

Therefore, the reduced model is constructed by a Petrov-Galerkin projection [18]

$$\begin{aligned} \tilde{G}z(t) + \tilde{C}\frac{dz(t)}{dt} &= \tilde{B}u(t) \\ y(t) &= \tilde{L}z(t) \end{aligned} \tag{3.5}$$

where

$$\begin{aligned}\tilde{G} &= M^T G V & \tilde{C} &= M^T C V \\ \tilde{B} &= M^T B & \tilde{L} &= L V\end{aligned}\tag{3.6}$$

We should note that matrices M and V can be identical. The selection of the two-sided projection matrices varies between different techniques.

The next section reviews a widely-used technique known as the implicit moment-matching based method.

3.2 Krylov-Subspace Using Moment-Matching Techniques

As a well-established method for the construction of subspaces, Moment-Matching Techniques (MMTs) are based on the notion of using the Taylor expansion to approximate the response of the original circuit [19, 20]. First, the transfer function of a system $H(s)$ is expanded with respect to complex frequency s

$$H(s) = \sum_{j=0}^{\infty} m_j(s_0)(s - s_0)^j\tag{3.7}$$

where s_0 is the expansion point (normally $s_0 = 0$) and m_j is the j -th moment. Therefore, to approximate the circuit's response, we can construct a reduced order model that matches the first q -th moments of the original model given as

$$H(s) \approx H_r(s) = \sum_{j=0}^{q-1} m_j(s_0)(s - s_0)^j\tag{3.8}$$

Using the modified equation of the transfer function in (2.8) from chapter 2, and selecting the expansion frequency point $s_0 = 0$, the approximation in (3.8) can be

rewritten as

$$H_r = \tilde{L}(I_r - s\tilde{A})^{-1}\tilde{R} = \tilde{L}[\tilde{R} + s\tilde{A}\tilde{R} + \dots + s^{q-1}(\tilde{A}^{q-1})\tilde{R}] \quad (3.9)$$

Hence, the subspace constructed using the moments in (3.9) is also known as the Krylov-subspace. The resulting ROM after projection matches the first q-th moments of the transfer function of the original model. The definition of the Krylov-subspace is given as

Definition 2.

$$V_k = Kr(A, R, q) = \text{colspan}\{R, AR, \dots, A^{q-1}R\}$$

There are several algorithms available for generating the Krylov-subspace in the literature, and we consider an efficient one among them called Arnoldi [21]. Next, we will review the concept of the Arnoldi approach.

3.2.1 Arnoldi-Process

A numerical algorithm to obtain the subspace is important in the moment-matching based reduction method. The Arnoldi approach introduces the concept of an 'Orthogonal Basis'. A basis of a subspace contains the minimum number of linearly independent vectors that span this subspace. Letting $V_k \in \mathbb{R}^{n \times q}$ represents the orthogonal basis of a Krylov-subspace $Kr(A, R, q)$, we have

$$V_k^T V_k = I \quad (3.10)$$

$$AV_k = V_k H_U$$

where $A = -G^{-1}C$ and H_U is an upper-Hessenberg matrix with zero entries below the first subdiagonal.

For further illustration, we can project a 3-dimensional trajectory onto a 2-dimensional surface (subspace), then an orthogonal (perpendicular) coordinate system (basis) can be found to describe any point in this 2-D space. The procedure of generating the orthogonal basis is given as follows

Algorithm 3.1: Arnoldi Algorithm

Input: A , R and N

Output: $V_k = [q_1, \dots, q_N]$, H_{sn} , where H_{sn} is an upper Hessenberg matrix.

begin Arnoldi

$$q_1 = \frac{R}{\|R\|}$$

for j = 1 to N - 1 do

$$z = Aq_j$$

for i = 1 to j do

$$h_{ij} = q_i^T z$$

$$z = z - h_{ij}q_i$$

end for

$$h_{j+1,j} = \|z\|$$

$$q_{j+1} = \frac{z}{h_{j+1,j}}$$

end for

end Arnoldi

The above algorithm provides a simple and efficient way to generate the Krylov basis for the case of a single input. Extending the idea to multi-input cases is straightforward, and is known as Block Arnoldi [22].

3.3 Krylov-Based Model Order Reduction

This section provides a review of existing efficient reduction algorithms based on the Krylov-subspace.

3.3.1 PRIMA

Passive Reduced-Order Interconnect Macromodeling Algorithm (PRIMA) [21] is one of the state-of-the-art methods for reducing passive circuits. The core of this algorithm is the Krylov-subspace obtained by using the Arnoldi process, with a slight modification employed when formulating the MNA equations. The resulting ROMs are guaranteed to be stable for all passive circuits.

One of the key procedures of PRIMA is modifying the MNA equation by multiplying the rows of the stamps of inductances and independent voltage sources by -1, resulting in

$$G = \begin{bmatrix} F & E \\ -E^T & 0 \end{bmatrix}, \quad C = \begin{bmatrix} C_a & 0 \\ 0 & -C_b \end{bmatrix}, \quad x = \begin{bmatrix} v \\ -i \end{bmatrix} \quad (3.11)$$

Note that with this modification, we obtain symmetric and diagonally dominant matrices $G + G^T$ and C . To perform the projection, the Krylov-subspace V_k from Arnoldi's method is chosen to be both of the two-sided projection matrices, which leads to the reduced model $[\tilde{G}, \tilde{C}, \tilde{B}, \tilde{L}]$ of order q with the passivity preserved and therefore to stability preserved, where

$$\begin{aligned} z(t) &= V_k^T x(t) \\ \tilde{G} &= V_k^T G V_k & \tilde{C} &= V_k^T C V_k \\ \tilde{B} &= V_k^T B & \tilde{L} &= L V_k \end{aligned} \quad (3.12)$$

The proof of passivity preservation is given as follows [23]

Theorem 2. *The necessary and sufficient conditions for a circuit transfer-function*

matrix $\tilde{H}(s) = \tilde{L}(\tilde{G} + s\tilde{C})^{-1}\tilde{B}$ to be passive (and hence stable) are

1. $\tilde{H}(s^*) = \tilde{H}^*(s)$, $\forall s \in \mathbb{C}$, where $*$ is the complex conjugate operator;
2. $\tilde{H}(s)$ is a positive matrix such that, $z^{*T}(\tilde{H}(s) + \tilde{H}^{*T}(s))z \geq 0$ for all complex values of s satisfying $\Re(s) > 0$ for any $z \in \mathbb{C}^q$.

Due to the fact that the reduced system matrices $[\tilde{G}, \tilde{C}, \tilde{B}, \tilde{L}]$ and the projection basis V_k are all real matrices, the first condition is automatically satisfied. To prove the second condition, suppose $\tilde{L} = \tilde{B}^T$, which implies the output ports are the same as the input ports. Let $\tilde{H}_h(s) = [\tilde{H}(s) + \tilde{H}^{*T}(s)]$, then we obtain:

$$\begin{aligned}
z^{*T}\tilde{H}_h(s)z &= z^{*T}[\tilde{B}^T(\tilde{G} + s\tilde{C})^{-1}\tilde{B} + \tilde{B}^T(\tilde{G} + s^*\tilde{C})^{-1}\tilde{B}]z \\
&= z^{*T}\tilde{B}^T[(\tilde{G} + s\tilde{C})^{-1} + (\tilde{G} + s^*\tilde{C})^{-1}]\tilde{B}z \\
&= z^{*T}\tilde{B}^T(\tilde{G} + s\tilde{C})^{-1}[(\tilde{G} + s\tilde{C}) + (\tilde{G} + s^*\tilde{C})^T](\tilde{G} + s^*\tilde{C})^{-T}\tilde{B}z
\end{aligned} \tag{3.13}$$

where z is an arbitrary complex vector, substituting $\kappa = (\tilde{G} + s^*\tilde{C})^{-T}\tilde{B}z$ and $s = j\omega + \sigma$ in (3.13), we obtain

$$\begin{aligned}
z^{*T}\tilde{H}_h(s)z &= \kappa^{*T}[(\tilde{G} + (\sigma + j\omega)\tilde{C}) + (\tilde{G} + (\sigma - j\omega)^*\tilde{C})^T]\kappa \\
&= \kappa^{*T}[(\tilde{G} + \tilde{G}^T + \sigma(\tilde{C} + \tilde{C}^T)]\kappa \\
&= \kappa^{*T}V_k^T[(\tilde{G} + \tilde{G}^T + \sigma(\tilde{C} + \tilde{C}^T)]V_k\kappa
\end{aligned} \tag{3.14}$$

replacing $V_k\kappa$ by δ yields

$$z^{*T}\tilde{H}_h(s)z = \delta^{*T}(\tilde{G} + \tilde{G}^T)\delta + \delta^{*T}[\sigma(\tilde{C} + \tilde{C}^T)]\delta \tag{3.15}$$

Since in the case of passive circuits, the modified C and $G + G^T$ in (3.11) are known

to be symmetric and diagonally dominant with non-negative real diagonal entries, then, according to the following theorem:

Theorem 3.

A symmetric diagonally dominant real matrix with non-negative diagonal entries is positive semi-definite.

Then we have the following equation to conclude the proof:

$$\begin{aligned} \delta^{*T}(G + G^T)\delta &= \delta^{*T} \left(\begin{bmatrix} F & E \\ -E^T & 0 \end{bmatrix}^T + \begin{bmatrix} F & E \\ -E^T & 0 \end{bmatrix} \right) \delta \\ &= \delta^{*T} \begin{bmatrix} 2F & 0 \\ 0 & 0 \end{bmatrix} \delta \geq 0 \end{aligned} \quad (3.16)$$

$$\delta^{*T}[\sigma(C + C^T)]\delta = 2\sigma\delta^{*T}C\delta \geq 0 \quad (3.17)$$

PRIMA preserves the passivity of the ROMs through a slight modification of the MNA formulation of passive circuits as shown in (3.11). However, the application of this technique is restricted within passive circuits. This is because PRIMA takes advantage of the peculiar structural properties of passive system matrices, which are not preserved by active circuits with the MNA formulation. For this reason, studies on stability-preserving MOR techniques for deterministic linear active circuits are attracting attention. Next, we will review a recently proposed MOR algorithm in this area.

3.3.2 A Stability Guaranteed Algorithm for Active Circuits

As explained in the previous section, PRIMA applies a Petrov-Galerkin projection by selecting the two-sided projection matrices to be $M = V = V_k$, where V_k is the Krylov-subspace efficiently obtained from the Arnoldi approach. However, PRIMA fails to construct stable ROMs when dealing with active circuits. This is because the symmetry of MNA equations is ruined while stamping active components such as dependent sources.

There exist several stabilizing MOR techniques based on optimization or post-processing, but we will not focus on them due to their low efficiency. Recently, a new projection based MOR approach is proposed in [17], this method generates stable ROMs for active systems by construction. It first performs an Arnoldi operation to generate the Krylov subspace as the right projection matrix, such that the first $q - th$ moments of the original model are preserved. Additionally, a left projection matrix is found through modifying the Krylov subspace with the solution of the Lyapunov stability constraint. The resulting ROM is therefore guaranteed to be stable. Here, we follow this method to gain the necessary background. Given a right projection matrix $V_k \in \mathbb{R}^{n \times q}$ found by Arnoldi's algorithm, the left projection $M \in \mathbb{R}^{n \times q}$ is expressed as

$$M^T = V_k^T C^T J \quad (3.18)$$

where $J \in \mathbb{R}^{n \times n}$ is an arbitrary symmetric positive-definite matrix obtained by solving the Lyapunov stability equation:

$$C^T J(-G) + (-G)^T J C = -Z \quad (3.19)$$

where C and G are the system matrices of the original model and $Z \in \mathbb{R}^{n \times n}$ is an

arbitrary positive definite matrix which is usually selected to be the unity matrix $I \in \mathbb{R}^{n \times q}$. Then the Generalized Lyapunov Inequality (GLI) can be defined as

$$C^T J(-G) + (-G)^T J C < 0 \quad (3.20)$$

By solving the generalized eigenvalue problem in (3.20), we obtain

$$J = (-G)^{-T} \Psi \Psi^{*T} (-G)^{-1}, \quad \Psi \in \mathbb{C}^{n \times q} \quad (3.21)$$

where the column vectors of Ψ are the q eigenvectors corresponding to the q stable eigenvalues of $(-G^T C)^T$. Hence with the left projection matrix M obtained using the above procedures, the reduced model is found through the Petrov-Galerkin projection in (3.5)

$$M^T G V_k z(t) + M^T C V_k \frac{dz(t)}{dt} = M^T B u(t) \quad (3.22)$$

$$y(t) = L V_k z(t)$$

The proof of stability preservation and an efficient algorithm for solving GLI can be found in [17]. MOR techniques for deterministic circuits are well-developed with a number of applications. However, difficulties arise when we look for extensions of these algorithms in the case of parametrized circuits, which are discussed in the next section.

3.4 Moder Order Reduction of Parametrized System

As a more practical extension, the idea of applying MOR on parametrized circuits is not new and research on this can be found in the literature [24–27]. The problem

in this case is much more complicated due to the fact that, in addition to the time or frequency variables, the random parameters affect the state trajectories as well. The response of a parametric system falls within a certain region instead of a deterministic trajectory in the state space. Hence, the ROMs are required to approximate the parameter-dependency of the original system as well, resulting in difficulties for developing efficient reduction algorithms. Here, we first review the method known as Parametrized Model Order Reduction (PMOR) [24].

Given a parametric circuit in (2.15),

$$\begin{aligned} G(\xi)x(t, \xi) + C(\xi)\frac{dx(t, \xi)}{dt} &= Bu(t) \\ y(t, \xi) &= Lx(t, \xi) \end{aligned} \tag{3.23}$$

where $\xi = [\xi_1, \dots, \xi_d]$ represents d random parameters and t is the time variable. As an extension of the Moment-Matching based MOR methods, one natural selection is to expand the transfer function of the original system with respect to all variables. For simplification of the notation, let E be a descriptor matrix, and $\eta = [s, \dots, \xi_n]$ represents a vector containing all the variables. Then, in the frequency domain we have

$$\begin{aligned} E(\eta)X(s) &= BU(s) \\ Y(s) &= LX(s) \end{aligned} \tag{3.24}$$

Expanding $E(\eta)$ in power series, the first equation in (3.24) can be rewritten as

$$[E_0 + \sum_{i=0} \eta_i E_i + \sum_{i,j=0} \eta_i \eta_j E_{ij} + \sum_{i,j,k=0} \eta_i \eta_j \eta_k E_{ijk} + \dots]X(s) = BU(s)$$

$$[I - (\sum_{i=0} \eta_i (-E_0^{-1} E_i) + \sum_{i,j=0} \eta_i \eta_j (-E_0^{-1} E_{ij}) + \sum_{i,j,k=0} \eta_i \eta_j \eta_k (-E_0^{-1} E_{ijk}) + \dots)] X(s) = E_0^{-1} B U(s)$$

$$[I - (\sum_{i=0} \eta_i A_0 + \sum_{i,j=0} \eta_i \eta_j A_1 + \dots + \sum_{i,j,k=0} \eta_i \eta_j \dots \eta_k A_n)] X(s) = R U(s) \quad (3.25)$$

where $A_0 = -E_0^{-1} E_i$, $A_1 = -E_0^{-1} E_{ij}$, \dots . Then, we obtain the expression of the multivariate transfer function as

$$H(\eta) = L \left(I - (\sum_{i=0} \eta_i A_0 + \sum_{i,j=0} \eta_i \eta_j A_1 + \dots + \sum_{i,j,k=0} \eta_i \eta_j \dots \eta_k A_n) \right)^{-1} R \quad (3.26)$$

According to Definition 2, a Krylov-basis obtained from the Arnoldi algorithm is given as

$$V_k = \text{colspan}\{R, A_0 R, A_1 R, \dots, A_n R, \\ A_0^2 R, (A_0 A_1 + A_1 A_0) R, \dots, (A_0 A_n + A_n A_0) R, A_n^2 R, \dots\} \quad (3.27)$$

Next, the ROM is generated using the projection approach. For instance, using PRIMA, the model reduction process for a passive circuit is expressed as

$$\begin{aligned} \tilde{G}(\xi) z(t, \xi) + \tilde{C}(\xi) \frac{dz(t, \xi)}{dt} &= \tilde{B} u(t) \\ y(t, \xi) &= \tilde{L} z(t, \xi) \end{aligned} \quad (3.28)$$

where $z(t, \xi) = V_k^T x(t, \xi)$, $\tilde{G} = V_k^T G V_k$, $\tilde{C} = V_k^T C V_k$, $\tilde{B} = V_k^T B$ and $\tilde{L} = L V_k$.

From the above procedure, we can find the difficulties lie in the direct construction of a Krylov-basis for parameterized circuits. First, this method suffers from the curse of dimensionality. Since there are both frequency and stochastic parameters in the Taylor expansion, the order of the Krylov basis is usually very large, resulting

in inefficient ROMs. In addition, the order of the ROMs grows drastically as the number of matched moments increases. Second, in order to achieve the multivariate Taylor expansion of the original system, the above method requires a closed-form representation of the parameter dependency in the formulation. Usually, linearity is assumed to describe the parameter-dependency.

3.5 Summary and Problem Definition

In this section, we summarize the problem discussed in chapters 2-3. There are two main difficulties in the Monte Carlo analysis of parametrized circuits. The first one is the large number of MC simulations and the slow convergence rate of the statistical results. The second problem is the large-size systems of the MC samples.

In order to achieve a fast simulation, one simple idea is constructing ROMs for the entire MC sample set, which is supported by the well-developed MOR algorithms for non-parametrized circuits. As a result, the computational burden reduces to simulating a large number of small systems. However, with this method, the first problem above is still unsolved. Although the simulation of the ROMs is generally much faster than simulating the original models, we must note that, the generation of ROMs is not free. The advantages of applying MOR techniques could be undermined as the number of the ROM constructions increases. For this reason, the PMOR approach is considered a better alternative. As explained in chapter 3, the original PMOR method based on the multivariate Taylor expansion cannot be a good choice due to the efficiency and the failure of stability preservation for active circuits. Moreover, the closed-form assumption is inaccurate in capturing the exact electrical information of parametric circuits. Improvements of the PMOR method have been done in [27] by eliminating trivial column-vectors in the Krylov-subspace of PMOR. These

algorithms still fail in generating stable ROMs for active parametrized circuits.

To address the above problems, an efficient algorithm for time-domain stochastic circuit analysis is proposed in the rest of this thesis.

Chapter 4

Polynomial Interpolation based Variability Analysis using PMOR

In the following three chapters, we propose a novel method for the variability analysis of parametric circuits using Model Order Reduction techniques. It obtains accurate stochastic results through a polynomial interpolation of the transient responses in a properly selected grid instead of solving a large number of MC samples. Moreover, the responses at the grid nodes are accurately approximated using the ROMs for a further speed-up. Hence, the proposed method overcomes the two main problems in the Monte Carlo simulation. It should be noted that the proposed approach can be applied to both passive and active circuits.

This chapter is organized as follows: sections 4.1-4.2 describe how to address the parametrized MOR problems through a sampling and interpolation scheme. Next, sections 4.3-4.4 provide the procedure for generating local transient responses through the MOR techniques and the numerical integration approach. A numerical example in section 4.5 is given to illustrate the proposed method.

4.1 Problem Decomposition

As previously discussed, there are two different types of variables in the MNA equation of a parametrized circuit. One is a class of stochastic parameters arising from the design and fabrication phases. These parameters are subject to certain probability distributions. The other type, the time (or frequency) variable only depends on the time (or frequency) region of the simulation under study. In addition, the response variation resulting from the random parameters is relatively smooth compared to the response variation due to the time or frequency available.

In order to address the difficulties associated with direct application of the parameterized MOR and to evaluate the stochastic response of interest efficiently, we first exclude the time/frequency variables and construct a sampling grid in the random space. This scheme leads to a set of samples at specific parameter values (so called nodes). Hence, there is no parameter-dependency in the system equations of the grid nodes, which can be interfered as decoupled systems. Then, any well-established MOR technique for non-parametrized circuits can be easily applied to achieve a fast simulation. Once the responses are obtained at all the nodes, the stochastic responses over the parameter space can be approximated by using polynomial interpolation. Then, statistical analysis can be performed through calculating the moments of the interpolated responses of the MC samples.

4.2 Stochastic Simulation based on Polynomial Interpolation

4.2.1 Polynomial Interpolation

Given a set of $n+1$ data points as shown in Table (4.1);

Table 4.1: A set of data evaluated at distinct values of x .

x_0	x_1	\cdots	x_n
$f(x_0)$	$f(x_1)$	\cdots	$f(x_n)$

a polynomial $g(x)$ of degree m ($m \leq n$) can be found such that the trajectory of this polynomial passes through all the data points.

$$g(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_2 x^2 + a_1 x + a_0 \quad (4.1)$$

$$g(x_i) = f(x_i), \quad i = 0, \cdots, n \quad (4.2)$$

where $\{x_i\}_0^n$ is the set of $n+1$ distinct points which are called interpolation points or nodes. Substituting (4.2) in (4.1), we get

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^m \\ 1 & x_1 & x_1^2 & \cdots & x_1^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \quad (4.3)$$

The matrix on the left side is also known as the Vandermonde matrix. Due to the fact that the $n+1$ points are distinct ($x_i \neq x_j, \forall i \neq j$), the Vandermonde matrix must be nonsingular. The following theorem presents the existence and uniqueness of the polynomial.

Theorem 4.

1. An n -degree non-zero polynomial has exactly n real or complex roots (including multiplicities).

2. Given $n+1$ distinct interpolation nodes $[x_0, \dots, x_n]$ and arbitrary function values $[f(x_0), \dots, f(x_n)]$, one and only one polynomial of degree at most n can be obtained that satisfies the interpolation.

Explicitly solving the linear equation in (4.3) for coefficients $[a_0, \dots, a_m]$, where $m \leq n$ is computationally expensive. Alternatively, the interpolating polynomials can be obtained directly by using the interpolation basis functions. Common basis functions include Lagrange basis functions, Piecewise linear basis functions and Chebyshev polynomials [28]. Let $\{\Lambda_i(x)\}_0^m$ denote the basis of the polynomial space of $g(x)$, then the polynomial of interest can be rewritten as a linear combination of function values $[f(x_0), \dots, f(x_n)]$

$$g(x) = \Lambda_0(x)f(x_0) + \Lambda_1(x)f(x_1) + \dots + \Lambda_n(x)f(x_n) \quad (4.4)$$

Hence, to obtain the function value at any point in the random space, we can directly compute the basis functions (weights) and combine the responses evaluated at each node. In general, the accuracy and computational complexity of the interpolation depend on the selection of the basis functions and the number of nodes. In the next section, we will construct a sampling-scheme based on equidistant nodes for polynomial interpolation.

4.2.2 Sampling Scheme

Consider the MNA equation of a univariate parametric circuit in (2.15)

$$G(\xi)x(t, \xi) + C(\xi)\frac{dx(t, \xi)}{dt} = Bu(t) \quad (4.5)$$

$$\mathbf{y}(t, \xi) = Lx(t, \xi)$$

We first select a set of n equidistant nodes $\{\xi_j\}_{j=1}^n$ in the parameter space. Note that two conditions must be satisfied by the chosen nodes:

1. The number of nodes should be as small as possible to ensure efficiency;
2. Nodes should be selected such that an accurate approximation can be attained through polynomial interpolation using certain basis.

4.2.3 Lagrange Basis Function

The interpolating polynomial can be constructed using Lagrange basis functions $P_j(\xi)$ expressed as

$$\mathbf{y}(t, \xi) \approx \mathcal{F}(\mathbf{y}) = \sum_{j=1}^n \mathbf{y}(t, \xi_j) P_j(\xi) \quad (4.6)$$

where the Lagrange basis function $P_j(\xi)$ is given as

$$P_j(\xi) := \prod_{i=1, i \neq j}^{n-1} \frac{\xi - \xi_i}{\xi_j - \xi_i} = \frac{(\xi - \xi_1) \dots (\xi - \xi_{j-1}) (\xi - \xi_{j+1}) \dots (\xi - \xi_n)}{(\xi_j - \xi_1) \dots (\xi_j - \xi_{j-1}) (\xi_j - \xi_{j+1}) \dots (\xi_j - \xi_n)} \quad (4.7)$$

The second condition of the nodal selection can be verified by introducing another set of samples called validation grids [29], e.g. the middle points of the intervals between every pair of neighbor nodes. Then, we can determine the effectiveness of the sampling scheme by comparing the interpolated results with their counterparts of actual samples in the validation grid.

In the case of multiple stochastic parameters $\boldsymbol{\xi} = (\xi^{(1)}, \dots, \xi^{(d)}) \subset \boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ defines a d -dimensional random space, a generalization can be easily obtained by using the tensor-product of one-dimensional grids.

$$\mathbf{y}(t, \xi^{(1)}, \dots, \xi^{(d)}) \approx \mathcal{F}(\mathbf{y}) = \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} \mathbf{y}(t, \xi_{j_1}^{(1)}, \dots, \xi_{j_d}^{(d)}) P_{j_1}(\xi^{(1)}) \cdots P_{j_d}(\xi^{(d)}) \quad (4.8)$$

where n_1, \dots, n_d denote the number of nodes along each dimension, and $P_{j_1}(\xi^{(1)}) \times \cdots \times P_{j_d}(\xi^{(d)})$ are the multivariate Lagrange interpolants. Provided that the nodal selection is appropriate, the polynomial interpolation in (4.8) gives an accurate approximation of the circuit response throughout the parameter space Ω .

4.2.4 Stochastic Analysis Using Polynomial Interpolation

In order to evaluate the statistical information of a parametrized circuit, we first generate a set of random values according to the parameters' distribution as explained for the Monte Carlo approach. However, unlike the standard MC method, instead of realizing and solving the MNA equation of all the samples, the responses of these random samples can be directly obtained by a polynomial interpolation using the nodal responses in the grid.

Given a set of N samples required by the MC method $\{\zeta_k\}_{k=1}^N = \{\zeta_k^{(1)}, \dots, \zeta_k^{(d)}\}_{k=1}^N \subset \Omega$, by substituting (4.7) in (4.8), we have the interpolated stochastic response expressed as

$$\begin{aligned} \mathbf{y}(t, \zeta_k^{(1)}, \dots, \zeta_k^{(d)}) &\approx \mathcal{F}_k(\mathbf{y}) = \\ &\sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} \mathbf{y}(t, \xi_{j_1}^{(1)}, \dots, \xi_{j_d}^{(d)}) P_{j_1}(\zeta_k^{(1)}) \cdots P_{j_d}(\zeta_k^{(d)}) \\ &= \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} \mathbf{y}(t, \xi_{j_1}^{(1)}, \dots, \xi_{j_d}^{(d)}) \prod_{i=1, i \neq j}^{n_1} \frac{\zeta_k^{(1)} - \xi_i^{(1)}}{\xi_j^{(1)} - \xi_i^{(1)}} \cdots \prod_{i=1, i \neq j}^{n_d} \frac{\zeta_k^{(d)} - \xi_i^{(d)}}{\xi_j^{(d)} - \xi_i^{(d)}} \quad (4.9) \end{aligned}$$

where $k = 1, \dots, N$. Using the interpolated responses of the MC sample set, the statistical information such as mean value μ_m and standard deviation σ_m are calculated using (2.23) and (2.25)

$$\mu_m[\mathcal{F}(\mathbf{y})] = \frac{1}{N} \sum_{k=1}^N \mathcal{F}_k(\mathbf{y}) \quad (4.10)$$

$$\sigma_m[\mathcal{F}(\mathbf{y})]^2 \approx \frac{1}{N-1} \sum_{k=1}^N [\mathcal{F}_k(\mathbf{y}) - \mu_m[\mathcal{F}(\mathbf{y})]]^2 \quad (4.11)$$

By using the above procedures, the stochastic analysis problem is simplified to interpolation and data acquisition. The latter will be discussed in the next section.

4.3 Generation of Reduced Order Model

In the previous section, the problem of repetitive MC simulations is reduced to simulating the deterministic systems only in the tensor product grid. We can further reduce the computational effort by using MOR techniques to approximate the transient responses of the deterministic systems at the grid-points. PRIMA has been proven as an efficient approach to reduce passive circuits while preserving the passivity (and hence stability) of the original circuits. On the other hand, the stability-preserving method in [17] can be applied to active circuits for efficient construction of stable ROMs. Therefore, we employ these methods for the construction of ROMs when dealing with different type of circuits.

Given the MNA formulation of a nodal system at a grid node $\boldsymbol{\xi}_j = (\xi_{j_1}^{(1)}, \dots, \xi_{j_d}^{(d)})$ as

$$G(\boldsymbol{\xi}_j)x(t, \boldsymbol{\xi}_j) + C(\boldsymbol{\xi}_j)\frac{dx(t, \boldsymbol{\xi}_j)}{dt} = Bu(t) \quad (4.12)$$

$$\mathbf{y}(t, \boldsymbol{\xi}_j) = Lx(t, \boldsymbol{\xi}_j)$$

where $C, G \in \mathbb{R}^{n \times n}$ are parameter-dependent conductance and susceptance matrices, respectively; $x \in \mathbb{R}^n$ is the vector of MNA variables, $u(t) \in \mathbb{R}^{n_{in}}$ is the vector of input signals, $\mathbf{y}(t, \boldsymbol{\xi}_j) \in \mathbb{R}^{n_{out}}$ is the parameter-dependent output response, and $B \in \mathbb{R}^{n \times n_{in}}$ and $L \in \mathbb{R}^{n_{out} \times n}$ are constant input and output selection matrices, respectively. The original circuit (either active or passive) presented by (4.12) is assumed to be asymptotically stable.

Next, the left and right projection matrices M and V are applied to form the reduced model at any grid node ξ_j as

$$\begin{aligned} \tilde{G}(\boldsymbol{\xi}_j)z(t, \boldsymbol{\xi}_j) + \tilde{C}(\boldsymbol{\xi}_j)\frac{dz(t, \boldsymbol{\xi}_j)}{dt} &= \tilde{B}u(t) \\ \tilde{\mathbf{y}}(t, \boldsymbol{\xi}_j) &= \tilde{L}z(t, \boldsymbol{\xi}_j) \end{aligned} \quad (4.13)$$

where $x = Vz$, $z \in \mathbb{R}^q$, $\tilde{C} = M^T C V$, $\tilde{G} = M^T G V \in \mathbb{R}^{q \times q}$, $\tilde{B} = M^T B \in \mathbb{R}^{q \times n_{in}}$, and $\tilde{L} = L V \in \mathbb{R}^{n_{out} \times q}$ with $q \ll n$.

4.3.1 Generation of Two-Sided Projection Matrices

The right orthogonal projection matrix $V \in \mathbb{R}^{n \times q}$ is formed through implicitly matching the first q -th moments of the transfer function in (4.13). One possible selection is the Krylov-subspace from the Block-Arnoldi algorithm. For a circuit excited by m inputs, the Block-Arnoldi algorithm finds an orthogonal projection matrix $V_k \in \mathbb{R}^{q \times n}$, $q = m \times q_b$, whose columns span the Krylov-subspace with the first $q_b - th$ moments matched for each input.

$$A = -G^{-1}C, \quad R = G^{-1}B \quad (4.14)$$

$$V = Kr(A, R, q_b) = \text{colspan}\{R, AR, \dots, A^{q_b-1}R\}$$

For a passive circuit, we select the Krylov-subspace from the above procedure as both of the two-sided projection matrices. In the case of active circuits, we follow the stability-preserving algorithm described in chapter 3 to generate the left projection matrix $M \in \mathbb{R}^{n \times q}$.

4.4 Transient Analysis Using Numerical Integration Techniques

Once the stable ROMs at all the nodes in the tensor product grids are constructed, the transient responses can be obtained by simulating these ROMs using the Numerical Integration methods. The solution of the time-domain differential equation in (4.13) is then approximated by a set of linear difference equations. Common numerical integration formulas include Backward Euler (BE) and Trapezoidal Rule (TR). For Backward Euler, suppose a trajectory is expressed using a linear equation of states at time points n and $n+1$ as $x_{n+1} = x_n + h \frac{dx_{n+1}}{dt}$, where h is the step-size. Then, we have

$$G(\boldsymbol{\xi}_j)x_{n+1} + C(\boldsymbol{\xi}_j)\frac{dx_{n+1}}{dt} = Bu_{n+1} \quad (4.15)$$

$$\frac{dx_{n+1}}{dt} = \frac{1}{h}[x_{n+1} - x_n] \quad (4.16)$$

Substituting (4.16) into (4.15) yields the formula of BE as

$$\left[G(\boldsymbol{\xi}_j) + \frac{C(\boldsymbol{\xi}_j)}{h}\right]x_{n+1} = \frac{C(\boldsymbol{\xi}_j)}{h}x_n + Bu_{n+1} \quad (4.17)$$

For the case of Trapezoidal Rule (TR), using the following assumption

$$\begin{aligned} x_{n+1} &= x_n + \frac{1}{2}h\left[\frac{d}{dt}(x_{n+1} + x_n)\right] \\ \frac{d}{dt}[x_{n+1} + x_n] &= \frac{2}{h}[x_{n+1} - x_n] \end{aligned} \quad (4.18)$$

Similarly, we can obtain the formula of TR as

$$\left[G(\boldsymbol{\xi}_j) + \frac{2}{h}C(\boldsymbol{\xi}_j)\right]x_{n+1} = \left[\frac{2}{h}C(\boldsymbol{\xi}_j) - G(\boldsymbol{\xi}_j)\right]x_n + B[u_{n+1} + u_n] \quad (4.19)$$

With the integration formulas shown above, the response at t_{n+1} is found using the response from the previous time point t_n through solving a linear difference equation. Hence, given the initial condition at t_0 , we can evaluate the trajectory through a “marching in time” implementation. It is well-known that the accuracy of these integration methods is highly dependent on the step size h .

4.5 Numerical Validation

In this section, we present an illustrative example to demonstrate the validity of the underlying ideas of the proposed method. Simulation results are presented in comparison with the conventional Monte Carlo approach.

4.5.1 Example 1

In this example, we consider a stable active circuit shown in Figure (4.1). The circuit is composed of two coupled transmission lines (TL) with op-amp amplifiers at the far-end. Each line has a length of 5cm and it is modeled using 100 sections of lumped RLGC segmentation. The internal model of the amplifiers is shown in Figure (4.3). The input signal $u(t)$ is chosen as a 2V pulse signal with 0.2ns fall/rise time and

2.3ns pulse-width.

The sources of uncertainty in this example arise from the geometric structure of the coupled transmission lines. Two stochastic physical parameters are considered whose nominal values are given in Figure (4.2) and are assumed to have Gaussian distributions with standard deviation of $\pm 30\%$.

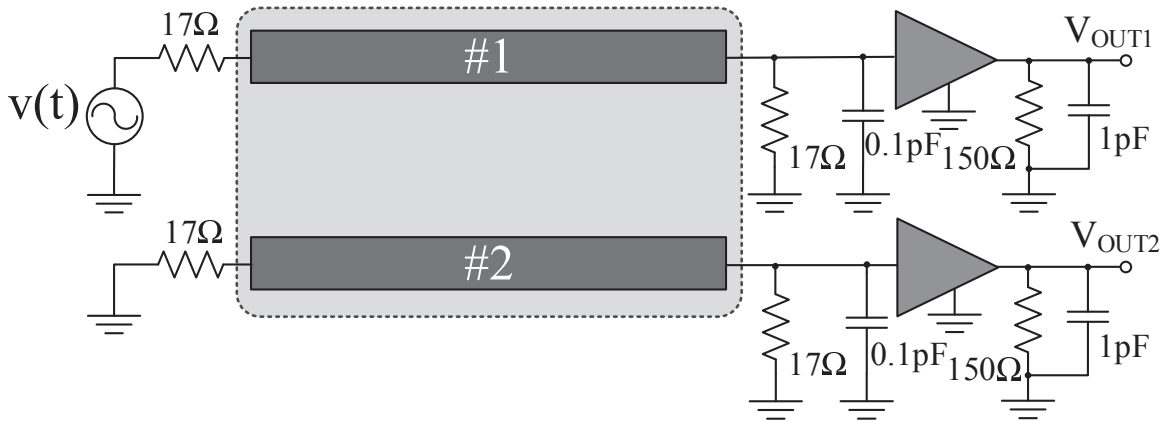
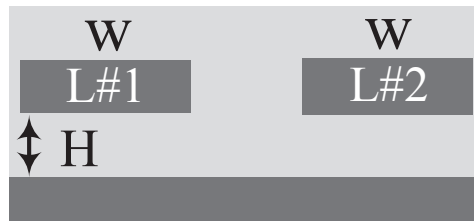


Figure 4.1: A stable active circuit consisting of two coupled transmission lines and two amplifiers.



Conductivity: $\sigma = 5.4e7 \text{ S/m}$
 Relative Permittivity: $\epsilon_r = 4.65$
 Line Width: $W = 180\mu\text{m}$
 Line Height: $H = 100\mu\text{m}$

Figure 4.2: Cross section of the transmission lines

Figure (4.4) shows the sampling scheme using the tensor-product of equidistant nodes in the two-dimensional parameter space. The grid covers the entire region of parameter variation between $\pm 3\sigma$, which is $\pm 90\%$ of the nominal value in this example. The nodes are chosen at every 15% of the nominal value, resulting in a

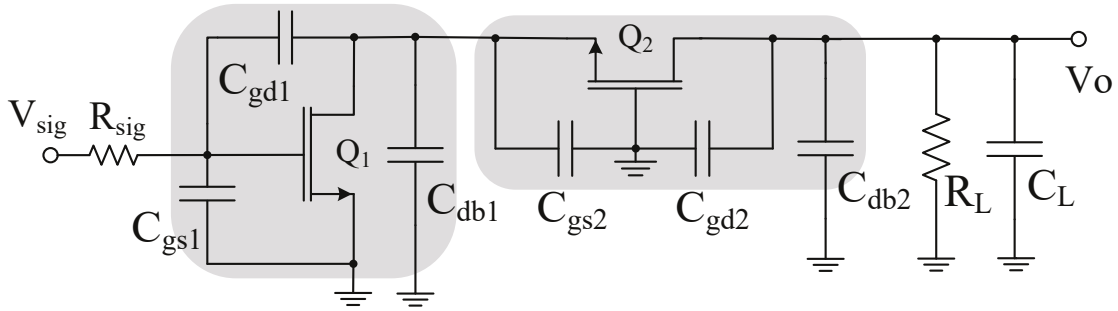


Figure 4.3: The internal circuit of the amplifiers.

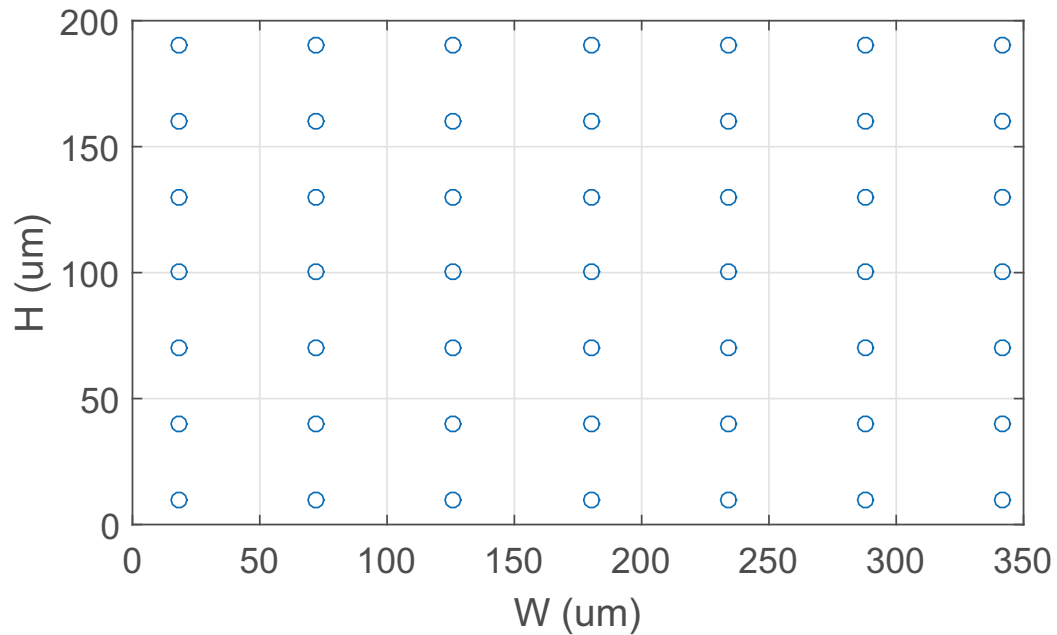


Figure 4.4: Equidistant nodes spaced on the 2-dimensional random space. 7-node one-dimensional grid and overall 49 nodes in the scheme.

Next, stable ROMs are constructed at each node in the sampling scheme by using the stability-preserving algorithm. The reduced order is predefined (14 in this example) such that the agreement of transient responses between original and reduced models is guaranteed. To verify the accuracy of the ROMs, we select the circuit operating at the nominal values. Figures (4.5) and (4.6) compare the frequency responses from the ROM at the output of the two lines with the responses from the full model. It can be seen that the frequency responses of the reduced model (order 14)

is accurate up to 2.5GHz, compared with the full model of order 612.

The corresponding transient response is generated using Trapezoidal Rule with step-size $h = 0.0201\text{ns}$ and is given in Figures (4.7) and (4.8). The results from the full model are accurately matched by the ROM. Moreover, the simulation result of the full model in HSPICE is provided in the comparison to validate the MATLAB implementation.

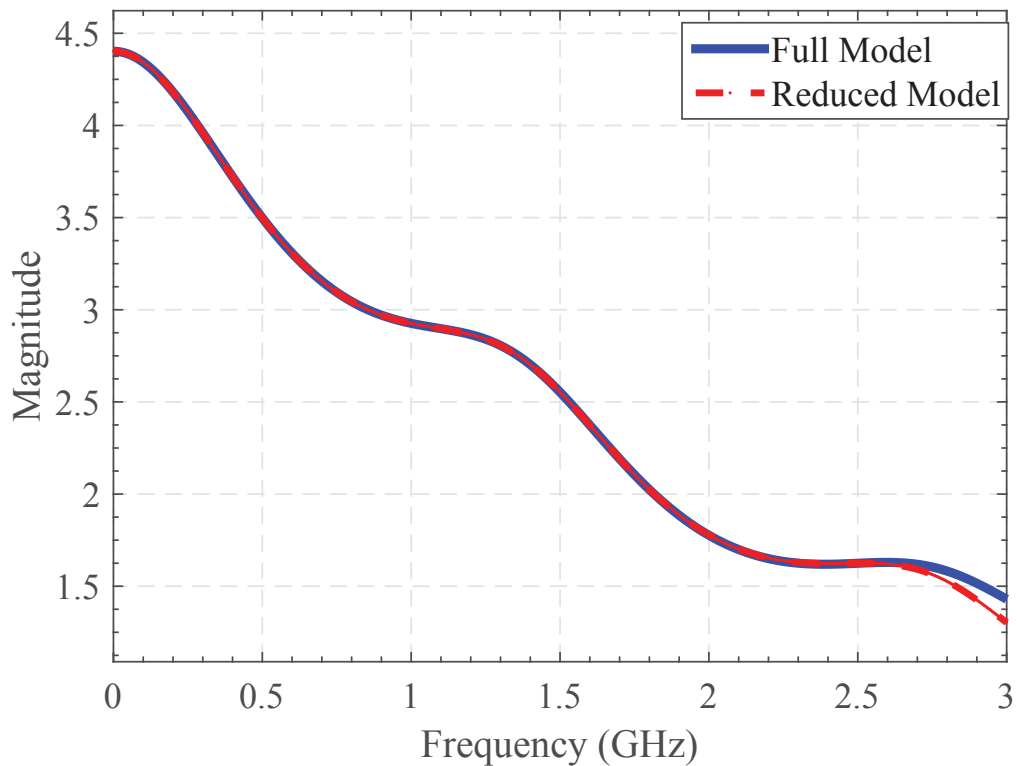


Figure 4.5: Frequency analysis at V_{out1} for the circuit operating at the nominal parameter values.

Once the transient responses of all the grid nodes are collected, the Lagrange Interpolation based Monte Carlo simulation is performed for statistical analysis. Figures (4.9) and (4.10) provide the mean value and $\pm 3\sigma$ tolerance of the output at each line using the proposed method in comparison with the standard Monte Carlo simulation. A good agreement between the results obtained using the two methods

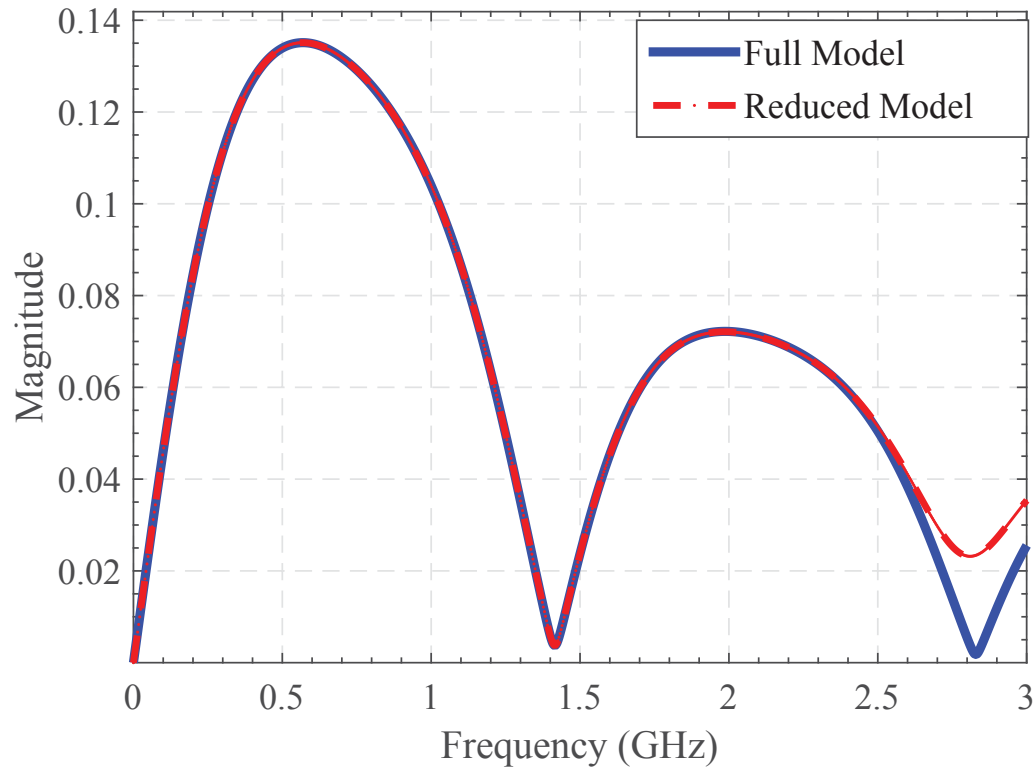


Figure 4.6: Frequency analysis at Vout2 for the circuit operating at the nominal parameter values.

is achieved.

Table (4.2) shows the CPU time comparison and the speed-up ratio between the two methods.

Table 4.2: CPU time comparison

	Time (s)
Monte Carlo	124.35
- Original system size: 612	
- 500 MC transient simulations	
Proposed interpolation approach	1.98
- Number of grid-nodes: 49	
- Reduced model size: 14	
- Random samples of parameters: 500	
Speed-up Ratio	≈ 63

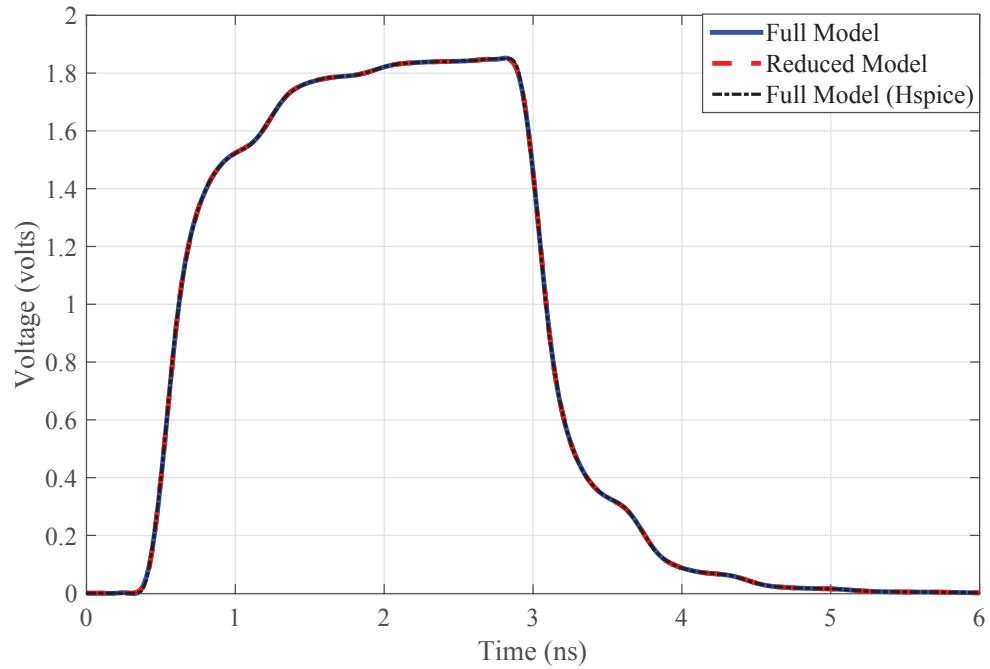


Figure 4.7: Transient analysis at Vout1 for the circuit operating at the nominal parameter values.

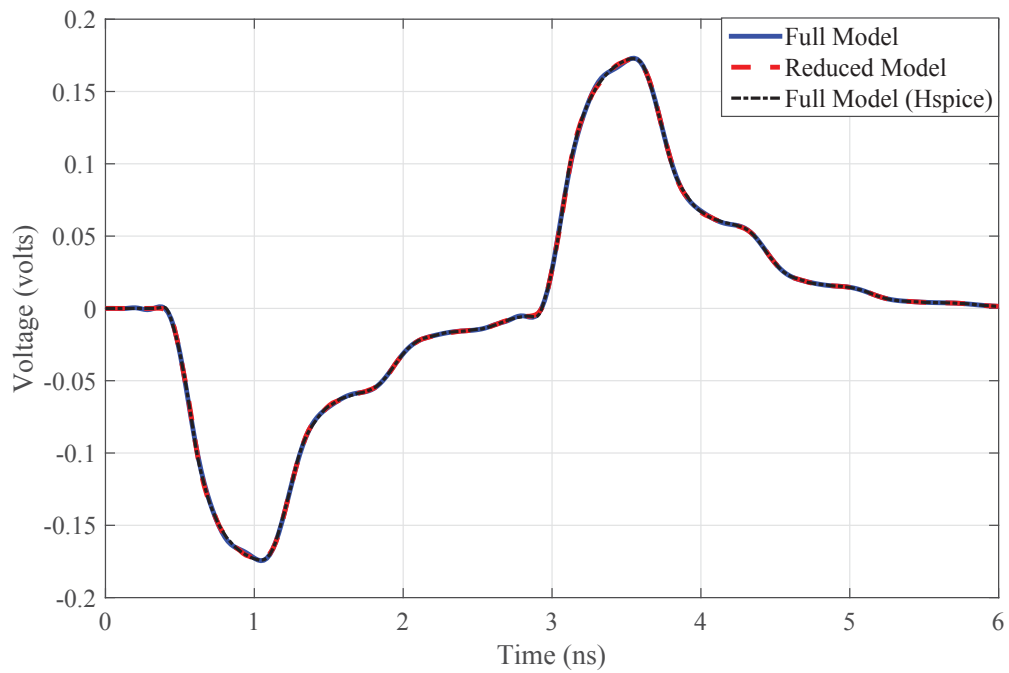


Figure 4.8: Transient analysis at Vout2 for the circuit operating at the nominal parameter values.

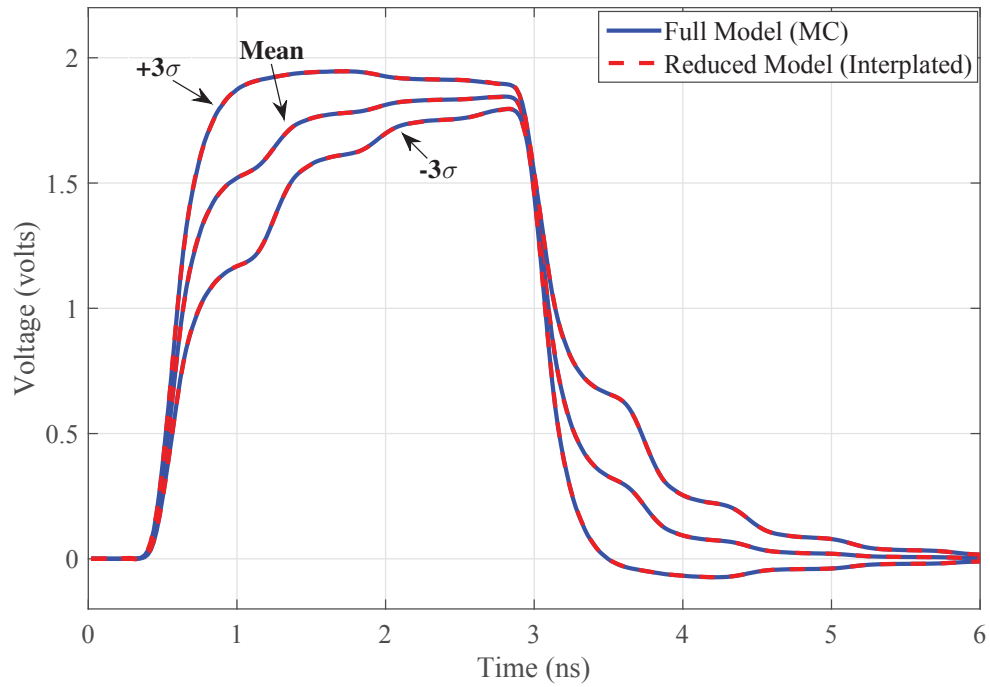


Figure 4.9: Time-domain statistical analysis for the output voltage at Vout1.

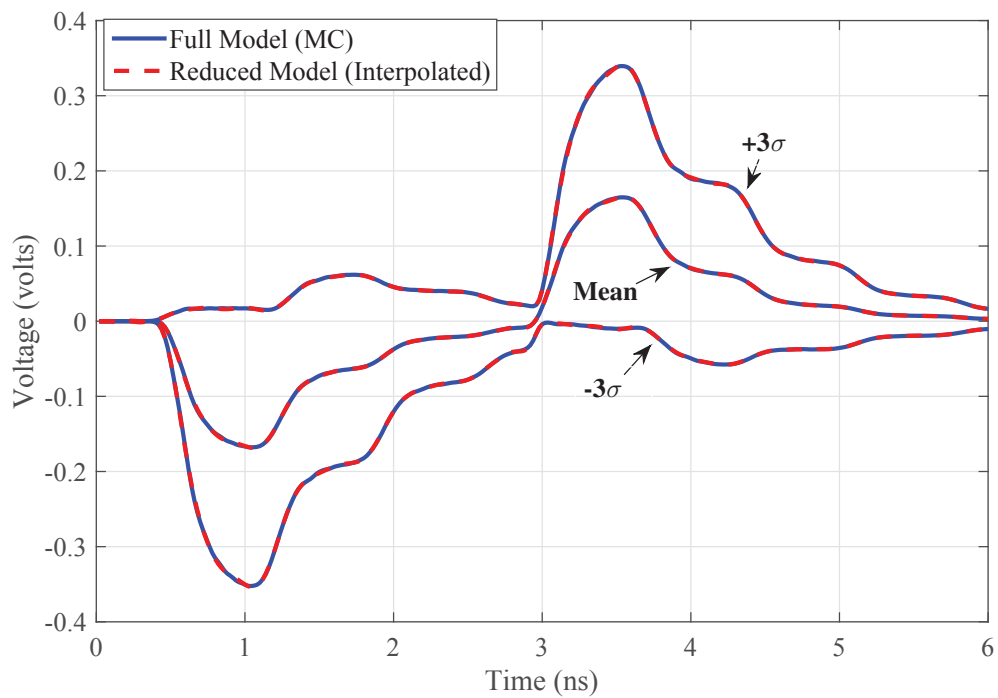


Figure 4.10: Time-domain statistical analysis for the output voltage at Vout2.

4.6 Summary

In this chapter, one of the key features of the proposed method is discussed, which is resolving the problem of repetitive simulations of large circuits in the Monte Carlo technique to simulating a small number of specific ROMs and then obtaining the MC results through polynomial interpolation. In addition, the stochastic results are guaranteed to be stable for both passive and active circuits. Numerical results of the illustrative example have confirmed the efficiency and accuracy of the proposed technique in comparison with the standard Monte Carlo approach. It is to be noted that, the advantage from the proposed method based on the scheme using equidistant grid-nodes and tensor-product extensions may be rapidly undermined as the number of parameters increases. To address this issue, in the next chapter, I propose the application of sparse-grid Stochastic Collocation (SC) techniques to parsimoniously select the grid-nodes and ensure the efficiency of variability analysis in the presence of multiple stochastic parameters.

Chapter 5

Variability Analysis Using Sparse-grid Stochastic Collocation Techniques

As presented in the previous chapter, the construction of an appropriate grid with a small number of nodes is of vital importance in the proposed method. The previous grid-construction using equidistant one-dimensional nodes and the tensor-product extension to the high-dimensional space is expensive. In this chapter, we propose to construct the grids using sparse-grid Stochastic Collocation (SC) techniques. Recently, these methods have become an attractive alternative in the stochastic analysis of differential equations with stochastic parameters [4, 30]. A key feature of the SC-based approach is evaluating the statistical quantities through the Gaussian Quadrature rules instead of the brute-force Monte Carlo approach. Moreover, SC methods require a small number of nodal system simulations and provide a fast convergence rate of the statistical responses.

The SC-based approach constructs the grid using the nodes of the quadrature rules as an optimal one-dimensional sampling scheme. In the case of multiple parameters, Smolyak's Sparse Grid (SG) techniques are used to construct a multi-dimensional grid [6, 31]. The Smolyak-based techniques generate the multivariate interpolants

using a weighted algebraic sum of low-order tensor products based on appropriately selected univariate interpolation basis functions. Hence, the efficiency of stochastic analysis is maintained in the high-dimensional case due to the significantly reduced nodal growth-rate in the sparse grid SC scheme.

5.1 Quadrature Integration

In addition to the stochastic solution of the parametrized circuit equations, we are also interested in evaluating the moments of the responses for statistical analysis. A quadrature rule is an effective method for numerical integration due to its efficiency and accuracy [32]. The main idea of this approach is to approximate the integrals using a weighted sum of the function values evaluated at specific nodes in the parameter space. An n-point quadrature rule is expressed as

$$\int_{\Gamma} \mathbf{y}(\xi)p(\xi)d\xi = \sum_{j=1}^n \mathbf{y}(\xi_j)w_j \quad (5.1)$$

where Γ is the support set where the PDF $p(\xi)$ is nonzero valued, and $\{w_j\}_{j=1}^n$ is a set of weights that can be pre-computed. Recall that a univariate stochastic solution can be approximated by a polynomial interpolation function given as

$$\mathbf{y}(t, \xi) \approx \mathcal{F}(\mathbf{y}) = \sum_{j=1}^N \mathbf{y}(t, \xi_j)\Lambda_j(\xi) \quad (5.2)$$

where $\Lambda_j(\xi)$ represents the interpolating basis functions. In general, the m-th moment is expressed by a weighted integral given as

$$\mathbb{E}(\mathbf{y}^m) = \int_{\Gamma} \mathbf{y}^m(t, \xi)p(\xi)d\xi \quad (5.3)$$

Substituting (5.2) in (5.3) yields

$$\begin{aligned}\mathbb{E}(\mathbf{y}^m) &= \sum_{j=1}^n \mathbf{y}^m(t, \xi_j) \int_{\Gamma} \Lambda_j(\xi) p(\xi) d\xi \\ &= \sum_{j=1}^n \mathbf{y}^m(t, \xi_j) W_j\end{aligned}\tag{5.4}$$

where the set of weights $\{W_j\}_{j=1}^n$ can be calculated using quadrature integration as

$$W_j = \int_{\Gamma} \Lambda_j(\xi) p(\xi) d\xi\tag{5.5}$$

$$= \sum_{i=1}^{n_i} \Lambda_j(\xi_i) w_i\tag{5.6}$$

The quadrature nodes are selected to be the zeros (roots) of the orthogonal polynomials, such that $\Lambda_j(\xi_i) = 0, \forall i \neq j$. The orthogonal polynomials for interpolation are selected according to the probability density function of the stochastic parameter used as a weighting function. The multiple options lead to different types of quadrature rules as shown in Table (5.1).

Therefore, instead of explicitly simulating or interpolating all the MC solutions, the stochastic collocation procedure reduces the problem to solving a set of n decoupled deterministic systems at each node $\xi_i, i = 1, \dots, n$, and computing the moments directly using quadrature rules. Due to the roots of the classical orthogonal polynomials are fixed, the corresponding nodes (roots) and weights are consistent as well and can be precomputed.

For further illustration, consider a circuit parameterized by a normally distributed

Table 5.1: Classical orthogonal polynomials.

Quadrature Rule	Distribution	$p(x)$	Interval	Orthogonal Polynomials
Gauss-Lagrange	Uniform	1	$[-1, 1]$	Legendre (P_n)
Gauss-Hermite	Normal	e^{-x^2}	$(-\infty, \infty)$	Hermite (H_n)
Chebyshev-Gauss	Chebyshev I	$\frac{1}{\sqrt{1-x^2}}$	$(-1, 1)$	Chebyshev I (T_n)
Chebyshev-Gauss	Chebyshev II	$\sqrt{1-x^2}$	$[-1, 1]$	Chebyshev II (U_n)
Gauss-Laguerre	Exponential	e^{-x}	$[0, \infty)$	Laguerre (L_n)

parameter $\xi \in N(\mu, \sigma^2)$, then the m-th moments of the response $\mathbb{E}[\mathbf{y}]$ are approximated by a weighted integral. Then, the quadrature integration based on the Hermite-Gauss rule is expressed as

$$\begin{aligned} \mathbb{E}[\mathbf{y}^m(t, \xi)] &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathbf{y}^m(\mathbf{t}, \xi) e^{\xi^2} d\xi \\ &= \sum_{j=1}^n \mathbf{y}^m(t, \xi_j) W_{Hj} = \mathcal{Q}_n \end{aligned} \quad (5.7)$$

where \mathcal{Q}_n denotes an n-point Hermite-Gauss quadrature rule. The accuracy of the result depends on the number of nodes used to perform the quadrature integration. The formula for computing the weights W_{Hj} is given as follows

$$W_{Hj} = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(\xi_j)]^2} \quad (5.8)$$

Table (5.2) shows the nodes and corresponding weights precomputed for an n-point (from n=2 to n=8) Hermite-Gauss quadrature rule [33].

Table 5.2: Nodes and weights of the Hermite-Gauss quadrature rule

n	Nodes (x_j)	Weights (W_j)
2	± 0.70710678118655	0.88622692545276
3	0 ± 1.22474487139159	1.18163590060368 0.295408975150919
4	± 0.52464762327529 ± 1.65068012388578	0.804914090005513 0.081312835447245
5	0 ± 0.95857246461382 ± 2.02018287045609	0.945308720482942 0.393619323152241 0.019953242059046
6	± 0.43607741192761 ± 1.33584907401370 ± 2.35060497367449	0.724629595224392 0.157067320322857 0.004530009905509
7	0 ± 0.81628788285897 ± 1.67355162876747 ± 2.65196135683523	0.810264617556807 0.425607252610127 0.054515582819127 0.000971781245099
8	± 0.38118699020732 ± 1.15719371244678 ± 1.98165675669584 ± 2.93063742025724	0.661147012558241 0.207802325814892 0.017077983007414 0.000199604072211

It should be noted that, since the chosen nodes are with respect to the normalized variable x , the exact nodes for sampling ξ_j in the parameter space are given as

$$\xi_j = \mu(\xi) + x_j \sqrt{2}\sigma(\xi) \quad (5.9)$$

The next step is to construct the system at the nodes set ξ and simulate the nodal system. Once we obtain the responses at each quadrature node, the moments used

for statistical analysis such as mean and standard deviation are evaluated as follows

$$\mu(\mathbf{y}) = \mathbb{E}(\mathbf{y}) \quad (5.10)$$

$$\sigma^2(y) = \mathbb{E}[(\mathbf{y} - \mu)^2] = \mathbb{E}(\mathbf{y}^2) - \mathbb{E}^2(\mathbf{y}) \quad (5.11)$$

As shown above, the variability analysis using quadrature integration requires a fewer number of nodal system simulations as compared with the Monte Carlo method. To calculate the moments of responses, the repetitive evaluation of interpolants for the MC samples is avoided as well. Moreover, the result of the quadrature rules converges faster than the MC method [34], which further ensures the accuracy and efficiency.

The SC scheme constructed using the sparse-grid techniques is presented in the following sections.

5.2 Node Selection Based on Nested Quadrature Rule

Nested quadrature rules are of critical significance in SC techniques. As shown in the last section, given two quadrature rules \mathcal{Q}_n and \mathcal{Q}_m ($n < m$) of the same type, the roots of the non-equal degree classical orthogonal polynomials are in general different. The resulting one-dimensional grid is not nested, which is denoted by $\Gamma_n \not\subset \Gamma_m$. As a result, when we increase the number of nodes in (5.3) for a higher accuracy level, the location of all sample points is changed, and the previously evaluated data is not reusable. This fact leads to a large number of trivial simulations in the SC-based approach. Hence, the quadrature rules for statistical analysis are generally required

to be nested.

Kronrod-Patterson's method [35] provides an effective way to construct a nested rule using a given n -point rule and additional nodes. The numerically stable approach to obtain these new nodes is given in [36].

5.2.1 Krnord-Patterson's Nested Quadrature Rule

We start with an n -point Gauss-type quadrature rule. The following theorem 5 justifies the existence and uniqueness of the set of weights $\{W_j\}_{j=1}^n$ [35, 37].

Theorem 5. *For every random variable ξ with the PDF $p(\xi)$ and every set of nodes $\{\xi_j\}_{j=1}^n \subset \Gamma$, there exists a set $\{W_j\}_{j=1}^n$ of n weights such that the quadrature integration with respect to $p(\xi)$ has a polynomial degree of exactness $m \geq n - 1$. These weights are the unique solution of the following equations:*

$$\sum_{j=1}^n \xi_j^m W_j = \int_{\Gamma} \xi^m p(\xi) d\xi, \quad m = 0, \dots, n - 1. \quad (5.12)$$

Assume there exists a new polynomial $\hat{\Lambda}_k(\xi)$ of degree k ($k \geq 1$) such that it satisfies

$$\int_{\Gamma} \Lambda_n(\xi) \hat{\Lambda}_k(\xi) p(\xi) \xi^m d\xi = 0, \quad m = 0, \dots, k - 1 \quad (5.13)$$

where the roots of $\hat{\Lambda}_k$ are all real, distinct, and non-equal to the roots from Λ_n . Then, according to Theorem 5, there exists a new quadrature formula expressed as

$$\mathcal{Q}_{n+k} = \sum_{j=1}^n \mathbf{y}(t, \xi_j)^m W_j + \sum_{i=n+1}^{n+k} \mathbf{y}(t, \xi_i)^m W'_i \quad (5.14)$$

The proof can be found in [37]. Studies on Kronrod-Patterson's extensions of Gauss quadrature rules can be found in [38–40].

To obtain the nested quadrature rule, the Kronrod-Patterson's method basically takes the following steps:

1. Select the polynomial basis $\Lambda_n(\xi)$ and construct an n -point quadrature rule with support nodes and corresponding weights.
2. Find the new monic (the highest degree coefficient is 1) polynomials $\hat{\Lambda}_k$ that satisfy equation (5.13).
3. Compute the roots of the new extension $\hat{\Lambda}_k$. Note that the nodes must be within the region of Γ .
4. For the new quadrature rule with $n+k$ nodes, evaluate the corresponding weights $\{W_j\}_{j=1}^{n+k}$ using (5.12). Note that due to the additional nodes, the weights of the original n -point rule is changed as well. Therefore, all the $n+k$ weights require re-evaluation.

5.2.2 Generation of Nodes and Weights

Given a monic polynomial extension $\hat{\Lambda}_k$ expressed as,

$$\hat{\Lambda}_k(x) = x^k + \sum_{i=0}^{k-1} c_i x^i \quad (5.15)$$

Substituting (5.15) in (5.13), we obtain

$$\int_{\Gamma} \Lambda_n(x) \left(x^k + \sum_{i=0}^{k-1} c_i x^i \right) p(x) x^m dx = 0 \quad (5.16)$$

$$\sum_{i=0}^{k-1} c_i \int_{\Gamma} \Lambda_n(x) p(x) (x^i x^m) dx = - \int_{\Gamma} \Lambda_n(x) p(x) x^k x^m dx \quad (5.17)$$

where the first index $i = 0, \dots, k-1$ and the second index $m = 0, \dots, k-1$ form a $k \times k$ matrix on the left side of the linear system. Theorem 5 ensures the exactness of

the moments up to degree $k - 1$. On the right hand side, we have a vector containing up to $k - 1$ degree moments of the probability density function $p(x)$. The matrix form of the linear equations are given as

$$\begin{bmatrix} \vdots \\ \cdots \int_{\Gamma} \Lambda_N(x)p(x)(x^{i+m})dx \cdots \\ \vdots \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{bmatrix} = \begin{bmatrix} \vdots \\ - \int_{\Gamma} \Lambda_N(x)p(x)x^{k+m}dx \\ \vdots \end{bmatrix} \quad (5.18)$$

The right hand side vector can be easily obtained using closed-form expressions. For instance, the closed-form of the moments for uniformly distributed ξ with the PDF $p(\xi) = 1$ is given as

$$\int_{-1}^1 \xi^m d\xi = \frac{1 + (-1)^m}{m + 1} \quad (5.19)$$

For parameters with other distributions, the PDF is used as a weighting function of the rational moments. In the case of normal distribution with Hermite polynomials

$$\int_{-\infty}^{\infty} \xi^m e^{-\xi^2} d\xi = \begin{cases} 0 & m \text{ odd} \\ \frac{\sqrt{\pi}}{2^{m/2}} \prod_{j=1}^{m/2} (2j - 1) & m \text{ even} \end{cases} \quad (5.20)$$

Then, by solving the linear equation in (5.18), the coefficients $\{c_i\}_{i=0}^{k-1}$ can be easily found and the extension polynomials are then constructed using (5.15). The additional nodes are chosen as the roots of the new polynomials and the corresponding

weights are obtained using (5.12). Hence, a nested quadrature rule is built by iteratively adding new pairs of nodes from extension polynomials. It is noted that, since the nodes and weights of generalized Gauss quadrature is fixed, their counterparts in the extended quadrature rule are also fixed and can be precomputed.

5.2.3 Multivariate Stochastic Collocation Scheme Using Tensor Product Grids

Based on the nested quadrature rules, a generalization of the SC scheme to multiple stochastic parameters $\boldsymbol{\xi} = (\xi^{(1)}, \dots, \xi^{(d)})$ can be obtained using tensor product polynomial interpolation [6]

$$\mathbf{y}(t, \xi^{(1)}, \dots, \xi^{(d)}) \approx \mathcal{F}^d(\mathbf{y}) = \sum_{j_1=1}^{N_1} \cdots \sum_{j_d=1}^{N_d} \mathbf{y}(t, \xi_{j_1}^{(1)}, \dots, \xi_{j_d}^{(d)}) \Lambda_{j_1}(\xi^{(1)}) \cdots \Lambda_{j_d}(\xi^{(d)}) \quad (5.21)$$

The general m-th stochastic moment of $\mathbf{y}(t, \boldsymbol{\xi})$ is expressed as

$$\begin{aligned} \mathbb{E}[\mathbf{y}^m(t, \xi^{(1)}, \dots, \xi^{(d)})] &= \\ &= \int_{\Gamma_1} \cdots \int_{\Gamma_d} \mathbf{y}^m(t, \xi^{(1)}, \dots, \xi^{(d)}) p(\xi^{(1)}, \dots, \xi^{(d)}) \approx \\ &= \sum_{j_1=1}^{N_1} \cdots \sum_{j_d=1}^{N_d} \mathbf{y}^m(t, \xi_{j_1}^{(1)}, \dots, \xi_{j_d}^{(d)}) W_{j_1}^{(1)}, \dots, W_{j_d}^{(d)} = \mathcal{Q}^d(\mathbf{y}^m) \end{aligned} \quad (5.22)$$

$$W_{j_k}^{(r)} = \int_{\Gamma_r} \Lambda_{j_k}(\xi^{(r)}) p(\xi^{(r)}) d\xi^{(r)} \approx \sum_{k=1}^n \Lambda_{j_k}(\xi^{(r)}) w_k \quad (5.23)$$

where $\mathbf{p}(\boldsymbol{\xi}) = p(\xi^{(1)}) \cdots p(\xi^{(d)})$ is a joint probability density function with support $\Gamma_1 \times \cdots \times \Gamma_d$. The nodes are chosen based on the one-dimensional nested quadratures

according to the distribution of the stochastic parameters.

In the next section, we provide an efficient multivariate scheme constructed using the Sparse Grid technique.

5.3 Nodes Minimization Using Smolyak's Sparse Grids

In this section, we present the application of the multidimensional grid construction based on Smolyak's Sparse Grid technique. Basically, tensor products of the univariate case are considered as a natural choice for multi-dimensional problems. Compared to the sampling scheme with manually selected equidistant nodes, the SC scheme minimizes the number of nodes at each dimension. Moreover, the procedure of interpolating the center-point of intervals between each pair of nodes for accuracy validation is not necessary. However, the computational effort of tensor-product generalization is still cumbersome due to the geometric growth of the grids as the number of parameters increases. Alternatively, Smolyak's technique [31] can overcome this curse of dimensionality by using sparse grids constructed from a linear combination of nodes in the tensor product grid. This approach starts from a univariate case and provides an straightforward extension to multi-dimensional problems. Moreover, it preserves a high level of accuracy while the number of grid points is relatively small. The construction of Smolyak's Sparse Grid can be approached by several methods. Here we focus on the widely used one, known as the combination technique [41, 42].

Given a univariate stochastic solution $\mathcal{F}_l^1(\mathbf{y})$ obtained from the polynomial interpolation in (5.2)

$$\mathcal{F}_l^1(\mathbf{y}) = \sum_{j=1}^n \mathbf{y}(t, \xi_j) \Lambda_j(\xi) \quad (5.24)$$

where l defines the accuracy level (depth) and n_l is the corresponding number of support nodes. Defining the difference formula of the polynomial interpolation with different accuracy level as

$$\Delta_l^1(\mathcal{F}) = (\mathcal{F}_l^1 - \mathcal{F}_{l-1}^1), \quad l = 1, 2, \dots \quad (5.25)$$

$$\text{and } \mathcal{F}_0^1 = 0$$

The nodes used for interpolation are essentially required to be nested, then we have $\Gamma_{l-1}^1 \subset \Gamma_l^1$. Therefore, the difference formula is considered on the support set of $\Gamma_{l-1}^1 \cup \Gamma_l^1 = \Gamma_l^1$. Then for the d -dimensional case, the Smolyak's formula is given as

$$\mathcal{F}_l^d(\mathbf{y}) = \sum_{|\mathbf{i}| \leq l+d-1} (\Delta_{i_1}^1 \otimes \dots \otimes \Delta_{i_d}^1) \mathcal{F} \quad (5.26)$$

where \mathbf{i} is a multi-index $\{i_j\}_{j=1}^d \in \mathbb{N}^d$ and $|\mathbf{i}| = i_1 + i_2 + \dots + i_d$, where i_j is the accuracy level along each dimension. The tensor-product formula can be written as

$$(\mathcal{F}_l^1 \otimes \dots \otimes \mathcal{F}_l^1) \mathbf{y} = \sum_{j=1}^d \sum_{1 \leq |\mathbf{i}|_j \leq l} (\Delta_{i_1}^1 \otimes \dots \otimes \Delta_{i_d}^1) \quad (5.27)$$

where $|\mathbf{i}|_j = \max\{i_1, \dots, i_d\}$. Then the original tensor-product grid is reduced to a combination of tensor-product expansions of the one-dimensional multilevel grid along each dimension. Another form of Smolyak's formula known as the combination

technique is expressed as

$$\mathcal{F}_l^d(\mathbf{y}) = \sum_{l \leq |i| \leq l+d-1} (-1)^{l+d-1-|i|} \binom{d-1}{|i|-l} (\mathcal{F}_{i_1}^1 \otimes \dots \otimes \mathcal{F}_{i_d}^1) \quad (5.28)$$

where $l \geq 1$. This is denoted as the global accuracy level (depth) of the polynomial interpolation.

Similarly, the sparse grid quadrature formula can be obtained as

$$\int_{\Gamma^d} \mathcal{F}_l^d = \sum_{l \leq |i| \leq l+d-1} (-1)^{l+d-1-|i|} \binom{d-1}{|i|-l} (\mathcal{Q}_{i_1}^1 \otimes \dots \otimes \mathcal{Q}_{i_d}^1) = \mathcal{Q}_l^d(\mathbf{y}) \quad (5.29)$$

where the one-dimensional quadrature is given as

$$\mathcal{Q}_i^1(\mathbf{y}) = \sum_{j=1}^{n_i^1} \mathbf{y}(t, \xi_{ij}) W_{ij} \quad (5.30)$$

To achieve the quadrature integration for statistical analysis, we should evaluate the systems at all the nodes in the sparse grid

$$\Gamma_l^d = \bigcup_{l \leq |i| \leq l+d-1} (\Gamma_{i_1}^1 \otimes \dots \otimes \Gamma_{i_d}^1) \quad (5.31)$$

Note that the d quadrature rules do not necessarily have to be identical, the node-number and type of the one-dimensional rule can be selected according to the parameters' features. This is known as anisotropic sparse grids [43]. For instance, if the a parameter's impact on the response is known to be smoother than others, we can select a quadrature with less nodes at this dimension. In addition, different types

of quadrature rules can be mixed in the construction of the multidimensional sparse grid.

Hereby, the multidimensional tensor-product quadrature is converted into a combination of tensor-product expansions of a one-dimensional multilevel quadrature rule along each dimension. Figure (5.1) shows a simple example of a 2-dimensional sparse grid constructed using the combination technique, where the accuracy level $l = 4$ and Γ denotes the set of support nodes.

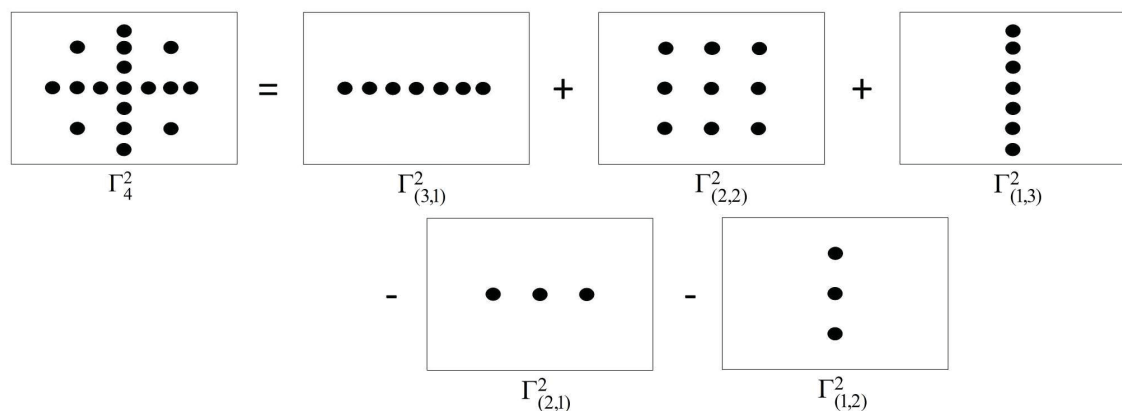


Figure 5.1: A two-dimensional sparse grid of accuracy level 4 constructed using the combination technique.

5.4 Numerical Validation

In this section, we provide numerical results to illustrate the proposed method using the sparse grid SC scheme. We select the same example as the one in section (4.5.1) which serves the purpose of comparing the proposed method with Monte Carlo technique. Moreover, we will show that the quadrature integration evaluates the statistical information of the response without explicitly interpolating the MC results.

First, the number of Monte Carlo simulations is increased to ensure convergence of the statistical result. It is illustrated in Figures (5.2) and (5.3) that in this example,

the statistical responses at output nodes Vout1 and Vout2 converge at 10000 times MC simulation.

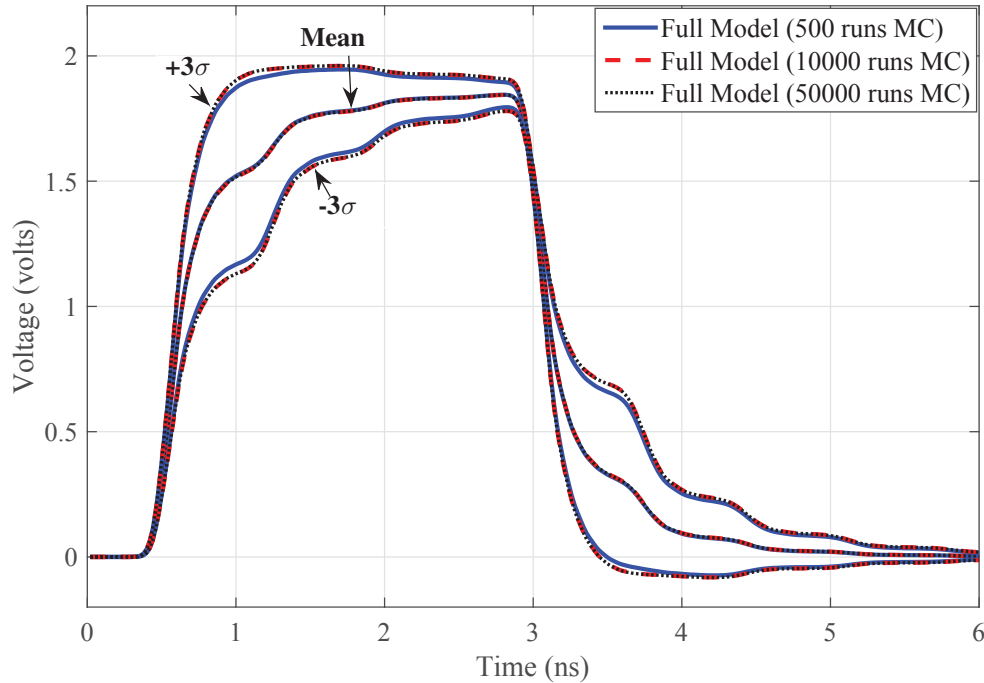


Figure 5.2: Time-domain statistical analysis at Vout1 using MC technique.

Next, using the Stochastic Collocation technique, the sparse-grid for quadrature integration is constructed for the two stochastic parameters W and H with both standard deviations of $\pm 30\%$. The nested Hermite Genz-Keister (HGK) quadrature [44] is selected to generate the one-dimensional nodes and weights. Then by using the sparse-grid method, a multi-dimensional sparse-grid is constructed with a set of nested nodes and the corresponding weights. Figure (5.4) shows the isotropic sparse-grids of accuracy level 2 and 3 whose node numbers are 5 and 9, respectively.

Using the same stability-preserving MOR technique, we obtain the stable ROMs of order 14 at each sparse-grid node. Then the transient responses of the ROMs are evaluated using Trapezoidal Rule with the previously selected step-size $h = 0.0201\text{ns}$.

Next, the moments are calculated directly by using a weighted summation of the

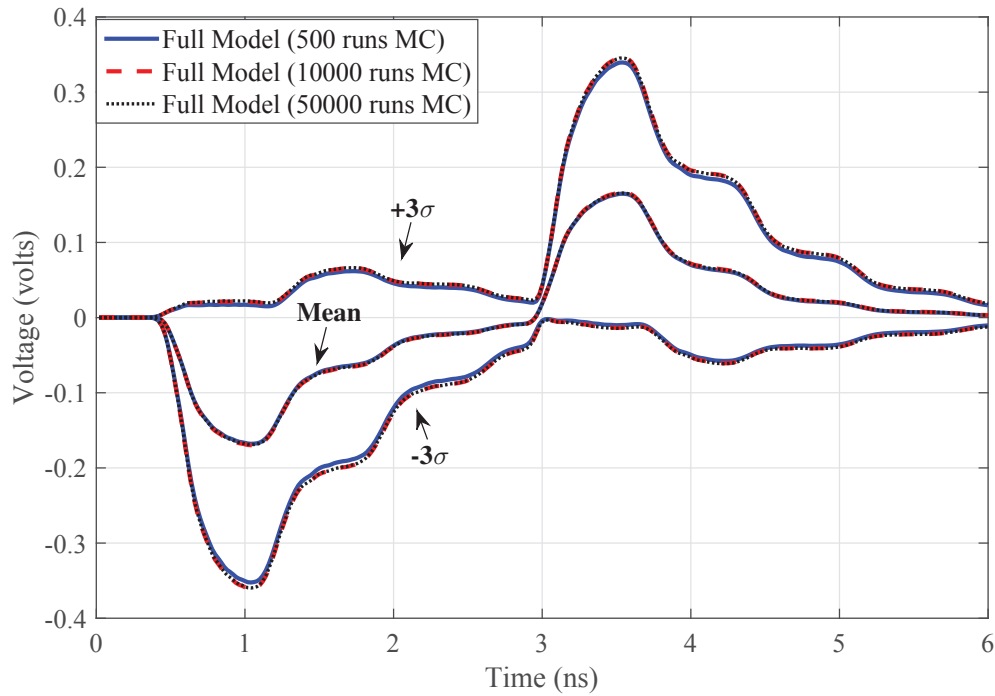


Figure 5.3: Time-domain statistical analysis at Vout2 using MC technique.

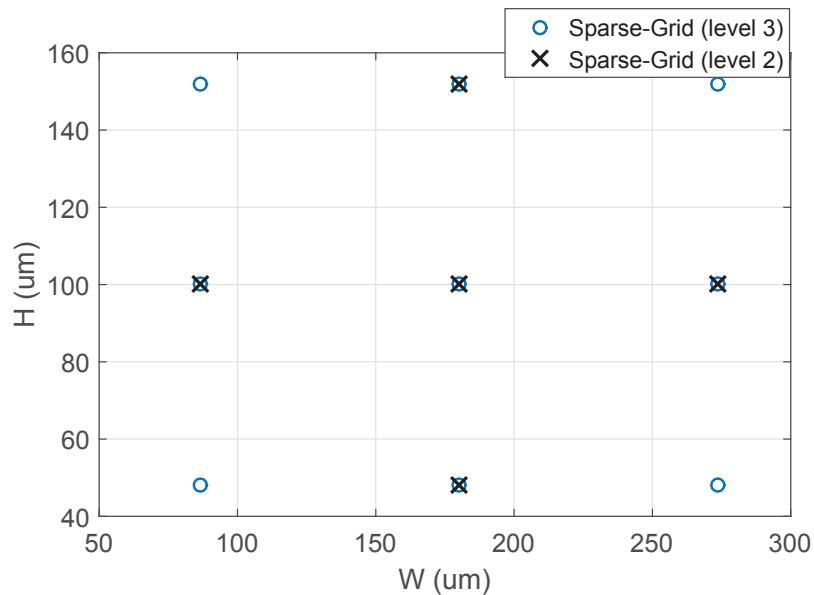


Figure 5.4: Two-dimensional sparse grids based on the Hermite Genz-Keister quadrature rule.

nodal responses in the sparse grid. Figures (5.5) and (5.6) provide the mean value and $\pm 3\sigma$ tolerance of output at each output node using the proposed method in

comparison with the standard Monte Carlo simulation. A good agreement is shown between the results obtained from the two approaches. It should be noted that the SC-based method converges remarkably faster than MC simulation. It is shown that we achieve accurate results with a level-3 sparse-grid (9 nodes), which significantly improves the efficiency of statistical analysis.

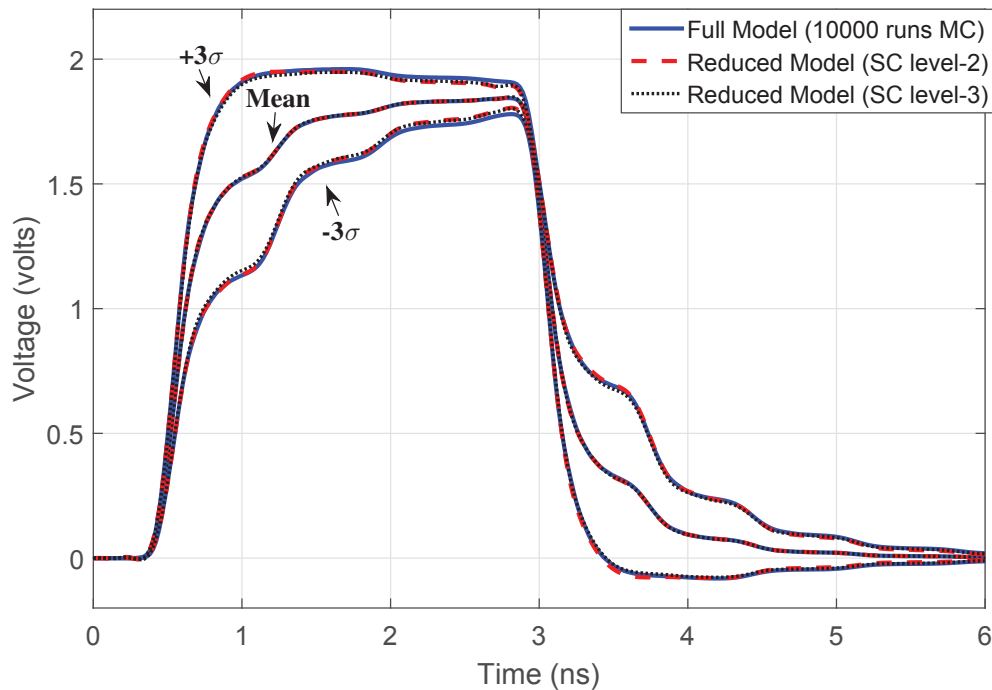


Figure 5.5: Time-domain statistical analysis at Vout1 using sparse grid SC techniques.

Table (5.3) shows the CPU time comparison between the Monte Carlo approach and the proposed method. For illustrative purposes, similar to Table (4.2) (in the previous chapter), the CPU-times are presented here only for 500 MC simulations. It is apparent that the speed-up will be noticeably improved as the number of MC simulations increases.

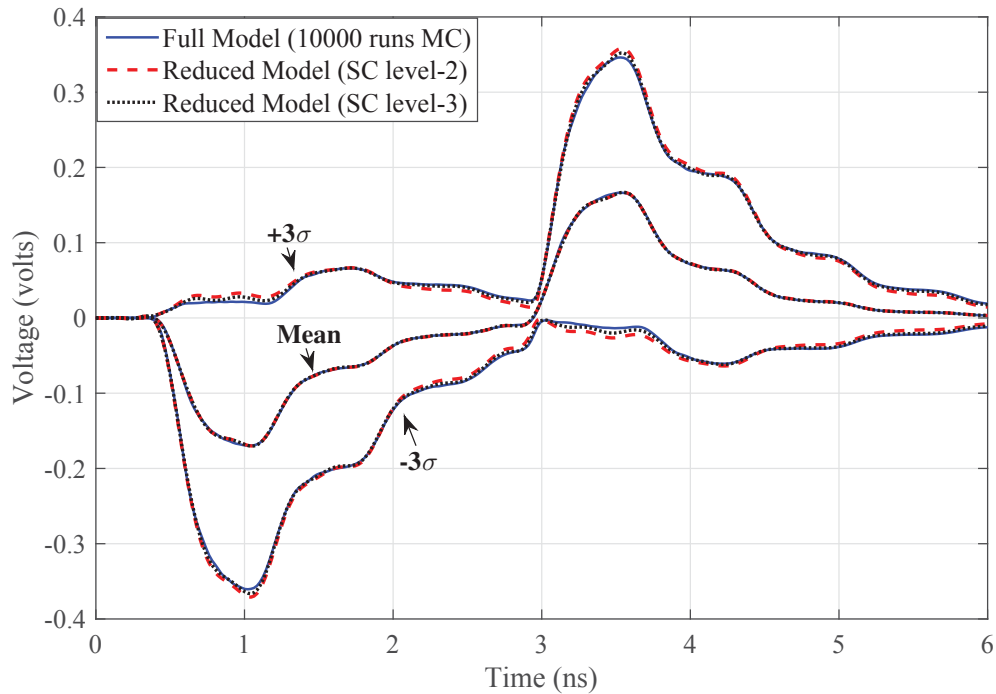


Figure 5.6: Time-domain statistical analysis at Vout2 using sparse grid SC techniques.

Table 5.3: CPU time comparison of example 1 (SC)

	Time (s)
Monte Carlo	124.35
- Original system size: 612	
- 500 MC transient simulations	
Proposed SC approach	0.37
- Number of sparse-grid nodes: 9 (level 3)	
- Reduced model size: 14	
- Random samples of parameters: 500	
Speed-up Ratio	≈ 336

5.5 Summary

This chapter presented the second key feature of the proposed method, which is evaluating the statistical simulation results using sparse-grid SC and quadrature integration techniques. Hence, the procedure of explicitly interpolating the MC results using ROMs is avoided. Moreover, it is illustrated by the numerical results that the proposed method provides excellent accuracy and a fast convergence rate with a relatively small number of sample nodes.

In the next chapter, a class of efficient transient analysis techniques known as Numerical Inverse Laplace Transform techniques is discussed to achieve further improvements of the CPU-time required by the proposed method.

Chapter 6

Time-domain Analysis Using Numerical Inverse Laplace Transform

In this chapter, an efficient method for generating the transient responses of the decoupled deterministic systems in the sparse grid is presented. As discussed in chapter 4, the time-domain analysis of the grid-nodes systems can be performed using numerical integration methods. However, the main difficulty is that using these methods, the responses at a new time point must be obtained based on the previous responses. Additionally, in order to obtain accurate results, the step-size h has to be relatively small, especially for circuits with high-speed signals propagating. Hence, numerical integration with a small step-size usually leads to a large computational effort and non-parallelizable implementations. For example, one may be interested in the system's response only at $t = 0.5ns$, however, the simulation has to start from time $t = 0s$ and steps forwards slowly.

There exist alternative techniques for time-domain simulation in the literature. To further improve the efficiency of the proposed method, we consider the Numerical Inversion of Laplace Transform (NILT) [45, 46] which introduces the Padé approximation into the transient response evaluation of linear circuits. The NILT method is

equivalent to an absolutely stable, high-order integration method. Hence, it is more accurate than the commonly used integration formulas (e.g. BE or TR) in commercial simulators. The high level of accuracy provided by NILT does not depend on the small step-size, and the information from previous responses is not necessary for the next response evaluation.

Due to its high efficiency, accuracy and parallelizability, we employ the NILT method in the time-domain analysis of the grid nodes systems. Next section presents a review of the Numerical Inverse Laplace Transform and the derivation of its numerical formulas.

6.1 Numerical Inverse Laplace Transform

Given a parameterized circuit operating at a node in the sparse grid, the s -domain MNA deterministic equation is expressed as

$$\begin{aligned} Y(s) &= L(G + sC)^{-1}BU(s) \\ &= H(s)U(s) \end{aligned} \tag{6.1}$$

where s is the complex frequency. The response $Y(s)$ is solved using LU decomposition.

Given a Laplace domain output of a linear system $Y(s)$, the time-domain response $y(t)$ may be obtained by applying the inversion of the Laplace transform on the function $Y(s)$ by evaluating the following integral

$$y(t) = \frac{1}{j2\pi} \int_{a'-j\infty}^{a'+j\infty} Y(s)e^{st} ds \tag{6.2}$$

where a' must be greater than the real part of any poles of $Y(s)$. For a stable system

it is known that, all the poles of $Y(s)$ exist in the left-half of the complex plane. Hence, application of NILT to the stable circuits which are the focus of this thesis allows for a' to be zero or greater.

Exact inversion in (6.2) may only be executed once all the poles of $Y(s)$ are known. However, this presents a challenge as large circuits may possess numerous poles and it proves to be prohibitively computationally expensive to apply root finding procedures to $Y(s)$. By denoting a new variable $z = st$, the inversion of Laplace transform becomes,

$$y(t) = \frac{1}{j2\pi t} \int_{a-j\infty}^{a+j\infty} Y\left(\frac{z}{t}\right) e^z dz \quad (6.3)$$

where a is an arbitrary constant such that it is greater than the real part of any poles of $Y(s)$. Alternately, the integral in (6.3) can be evaluated numerically using Numerical Inversion of Laplace Transform (NILT) presented in [46, 47] and its main idea is illustrated in Figure (6.1).

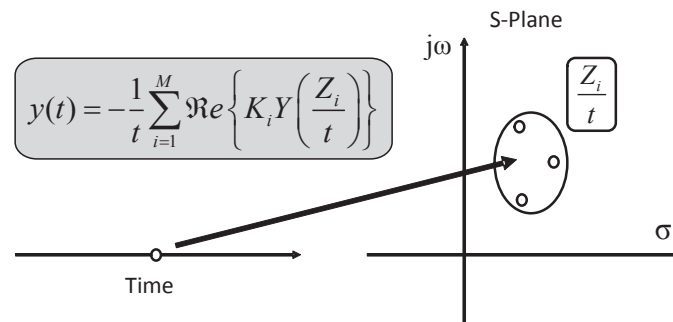


Figure 6.1: An illustration of NILT that the time response at one time point is a summation of frequency responses at several complex frequencies.

Approximating the exponential function e^z using the Padé approximation we get

$$y(t) \approx \frac{1}{2\pi j t} \int_{a-j\infty}^{a+j\infty} Y\left(\frac{z}{t}\right) \mathcal{P}_{N,M}(z) dz \quad (6.4)$$

where the Padé approximation is given as

$$\mathcal{P}_{N,M} = \frac{\alpha_N(z)}{\beta_M(z)} = \frac{\sum_{i=0}^N (M+N-i)! \binom{N}{i} z^i}{\sum_{i=0}^M (-1)^i (M+N-i)! \binom{M}{i} z^i} \quad (6.5)$$

where N and M determines the order of the numerator and the denominator of the Padé approximation, respectively. Note that we should define the order such that $\mathcal{P}_{N,M}$ has more poles than zeros ($N < M$). Formula (6.5) can be rewritten in the pole-and-residue form

$$\mathcal{P}_{N,M} = \sum_{i=1}^M \frac{K_i}{z - z_i} \quad (6.6)$$

where z_i and K_i are the poles and residues of the Padé approximation. It is to be noted that the function $Y(\frac{z}{t}) \mathcal{P}_{N,M}(z)$ in (6.4) only has a small number of poles on the right half plane. Therefore, substituting (6.6) in (6.4) and taking the contour integration in the right half complex plane about all poles of $\mathcal{P}_{N,M}$ in (6.4) is considered. In accordance with the residue theorem the integral may be evaluated by closing the path of integration between $a - j\infty$ and $a + j\infty$ either to the left or to the right. By the Cauchy theorem only the residues within the closed contour will contribute to the integral. Here, the choice is made to close the path to the right as there are numerous residues in the left half plane arising from the system's poles in the left half plane. In order that the path along the infinite arc not contribute to the integral, it is necessary that M, N be chosen such that the function $Y(z/t)\mathcal{P}_{N,M}(z)$ has at least

two more finite poles than zeroes.

$$\begin{aligned}
 y(t) &\approx \frac{1}{2\pi jt} \int_{a+j\infty}^{a-j\infty} Y\left(\frac{z}{t}\right) \sum_{i=1}^M \frac{K_i}{z - z_i} dz \\
 &= -\frac{1}{t} \sum_{i=1}^M K_i Y\left(\frac{z_i}{t}\right)
 \end{aligned} \tag{6.7}$$

M is usually taken to be even due to the poles of Padé approximation always appear in complex-conjugate pairs. Then (6.7) is rewritten as

$$y(t) \approx -\frac{1}{t} \sum_{i=1}^{M'} \Re[K'_i Y\left(\frac{z_i}{t}\right)] \tag{6.8}$$

where $M' = \frac{M}{2}$ when M is even, and $K'_i = 2K_i$; $M' = \frac{M+1}{2}$ when M is odd, and in this case $K'_i = K_i$ for the residue corresponding to the real pole. \Re denotes the real part of a complex number.

By using the above procedure, exactly $N+M+1$ terms of the Taylor expansion of $y(t)$ are preserved [45]. Therefore, the NILT method provides a good approximation as compared with the widely used Backward Euler and Trapezoidal Rule with the order of one and two, respectively. In addition, the truncation error expression is derived in [48] which shows that the truncation error decreases rapidly as M and N increase. It should be noted that K_i and z_i can be precomputed and stored as a lookup table [49].

6.2 High Order NILT

The NILT method discussed in the last section is also known as NILT of order zero (NILT0). The high-order forms (NILTn) [50] introduce the $(n+1)$ -th degree of Padé

approximation and n -th degree derivatives to improve the accuracy of transient analysis. Substituting $t = (n + 1)t'$ in (6.3), we have

$$y(t') = \frac{1}{2\pi j} \int_{a-j\infty}^{a+j\infty} Y(s) e^{(n+1)st'} ds \quad (6.9)$$

Then, by substituting $z = st'$ and $m = n+1$, and using the Padé approximation, (6.9) is rewritten as

$$\begin{aligned} y(t) &\approx \frac{m}{2\pi jt} \int_{a-j\infty}^{a+j\infty} Y\left(\frac{z}{t'}\right) \mathcal{P}_{N,M}^m(z) dz \\ &= \frac{m}{2\pi jt} \int_{a-j\infty}^{a+j\infty} Y\left(\frac{mz}{t}\right) \mathcal{P}_{N,M}^m(z) dz \end{aligned} \quad (6.10)$$

Here, we take $n=1$ to follow the derivation of the discrete formula of NILT1. The Padé approximation of degree $m=2$ is expressed as

$$\begin{aligned} \mathcal{P}_{N,M}^2(z) &= \left(\sum_{j=1}^M \frac{K_j}{z - z_j} \right)^2 \\ &= \sum_{j=1}^M \sum_{i=1}^M \frac{K_j K_i}{(z - z_j)(z - z_i)} \\ &= \sum_{\substack{j=1 \\ i=j}}^M \frac{K_j^2}{(z - z_j)^2} + \sum_{\substack{j=1 \\ i \neq j}}^M \sum_{i=1}^M \frac{K_j K_i}{(z - z_j)(z - z_i)} \end{aligned} \quad (6.11)$$

Using (6.10) and (6.11), and taking the contour integration in the entire right half complex plane, we have

$$\begin{aligned}
y(t) &\approx \frac{1}{\pi j} \int_{a+j\infty}^{a-j\infty} Y\left(\frac{2z}{t}\right) \left(\sum_{\substack{j=1 \\ i=j}}^M \frac{K_j^2}{(z-z_j)^2} + \sum_{\substack{j=1 \\ i \neq j}}^M \sum_{i=1}^M \frac{K_j K_i}{(z-z_j)(z-z_i)} \right) dz \\
&= -\frac{2}{t} \left(\frac{2}{t} \sum_{\substack{j=1 \\ i=j}}^M K_j^2 Y'\left(\frac{2z_j}{t}\right) + 2 \sum_{\substack{j=1 \\ i \neq j}}^M \sum_{i=1}^M \frac{K_j K_i}{(z_j - z_i)} Y\left(\frac{2z_j}{t}\right) \right) \quad (6.12)
\end{aligned}$$

where Y' denotes the first degree derivative of the response Y . The values of M and N are selected to be even such that the imaginary part of (6.12) is zero. Selecting $M' = \frac{M}{2}$ and $K'_i = 2K_i$ yields

$$\begin{aligned}
y(t) &\approx -\frac{2}{t} \left(\frac{1}{t} \sum_{\substack{j=1 \\ i=j}}^{M'} K_j'^2 Y'\left(\frac{2z_j}{t}\right) + 2 \sum_{\substack{j=1 \\ i \neq j}}^{M'} \sum_{i=1}^M \frac{K'_j K_i}{(z_j - z_i)} Y\left(\frac{2z_j}{t}\right) \right) \\
&= -\frac{2}{t} \sum_{\substack{j=1 \\ i=j}}^{M'} \Re \left(\frac{1}{t} K_j'^2 Y'\left(\frac{2z_j}{t}\right) + 2K'_j \sum_{\substack{i=1 \\ i \neq j}}^M \frac{K_i}{(z_j - z_i)} Y\left(\frac{2z_j}{t}\right) \right) \quad (6.13)
\end{aligned}$$

Higher order form of NILT ($n > 1$) can be derived using similar procedures. We focus on NILT1 and NILT0 in this thesis. It should be noted that the evaluation of the derivative $Y'(s)$ can be easily obtained. Consider a frequency domain MNA equation given as

$$(G + sC)X(s) = BU(s)$$

$$Y(s) = LX(s)$$

It is well known that the main effort in solving the above equation is the LU decomposition for inverting the matrix $G + sC$. To evaluate the first degree derivative, we

have

$$\begin{aligned}\frac{d}{ds}[(G + sC)X(s)] &= \frac{d}{ds}[BU(s)] \\ \frac{d}{ds}X(s) &= (G + sC)^{-1}[B\frac{d}{ds}U(s) - CX(s)] \\ Y'(s) = L\frac{d}{ds}X(s) &= L(G + sC)^{-1}[B\frac{d}{ds}U(s) - CX(s)]\end{aligned}\quad (6.14)$$

As shown in (6.14), the calculation of the derivative $Y'(s)$ requires one operation of the LU decomposition, which is already computed while solving for $Y(s)$. As a result, the additional derivative used in NILT1 does not introduce high computational effort compared to NILT0.

6.3 Implementation of NILT

The implementation of NILT mainly takes the following steps:

1. Calculate poles and residues of Padé approximation corresponding to the different values of M and N ($N = M - 2$) as a lookup table. In this thesis, we select from $M=6$ to $M=16$ and K_i and z_i are generated using Mathematica.
2. Select a set of time points $\{t_j\}_{j=0}^n$ for the transient simulation under study, and it should be noted that the step-size is not necessarily to be fixed. Due to the required frequency response $Y(\frac{z_i}{t})$, the first time point is usually taken to be a small value instead of $t_0 = 0$.
3. Define the value of M according to the required accuracy level, select the corresponding pairs of poles $\{z_i\}_{i=1}^{M/2}$ and generate the frequency vector using $s = \frac{z_i}{t}$.
4. Set the initial condition of the state vector X_0 and evaluate the frequency

response at each frequency point.

$$Y\left(\frac{z_i}{t}\right) = H\left(\frac{z_i}{t}\right)U\left(\frac{z_i}{t}\right)$$

Note that for high order formulas NILT_n (n>0), the derivatives $\frac{dY(z_i/t)}{ds}$ require evaluation as well.

5. Sum the frequency response and complete the inversion using corresponding NILT formulas.

We should note that in the time-domain simulation using NILT techniques, the Laplace transform $U(s)$ of some typical inputs such as a trapezoidal pulse are difficult to find. To simplify the implementation, we can use the superposition of one step function to form the pulse as follows

$$v(t) = v'(t) - v'(t - t_r) - v'(t - t_r - t_p) + v'(t - t_r - t_p - t_f) \quad (6.15)$$

where t_r , t_f and t_p represent the rising, falling and pulse-width time. Hence, one can only calculate the time response of rising and falling edge (only rising edge if $t_r = t_f$).

The Laplace transform of a step function is simply expressed as

$$U(s) = \frac{A}{t_r s^2} \quad (6.16)$$

where A is the pulse's amplitude. Therefore, by properly selected 'break-points' for superposition, we can simply obtain the response of the entire time region under study, which is equivalent to exact inversion. In the next section, we provide numerical results to show the application of NILT in the proposed method.

6.4 Numerical Validation

In this section, we provide numerical results to show the accuracy and efficiency of the Numerical Inverse Laplace Transform method for transient simulation of nodal systems in the sparse grid. We select the parameterized circuit in section (4.5.1) operating at the nominal value of the stochastic parameters.

Due to the fact that the NILT methods are parallelizable, in contrast to the stepping forward Numerical Integration methods, we have flexibility in selecting the time-points. For NILT methods, the step-size does not necessarily need to be fixed. Here, we select 60 time-points for the NILT and compare with the results from HSPICE.

Figure (6.2) and (6.3) shows the transient response at the far-end amplifier output node. The accuracy of the NILT0 method is improved by increasing the value of M . Figure (6.4) and (6.5) present the results obtained by using HSPICE, NILT0 with $M = 6$ and NILT1 with $M = 6$. It is demonstrated that using the same order of the Padé approximation, NILT1 provides better accuracy than NILT0.

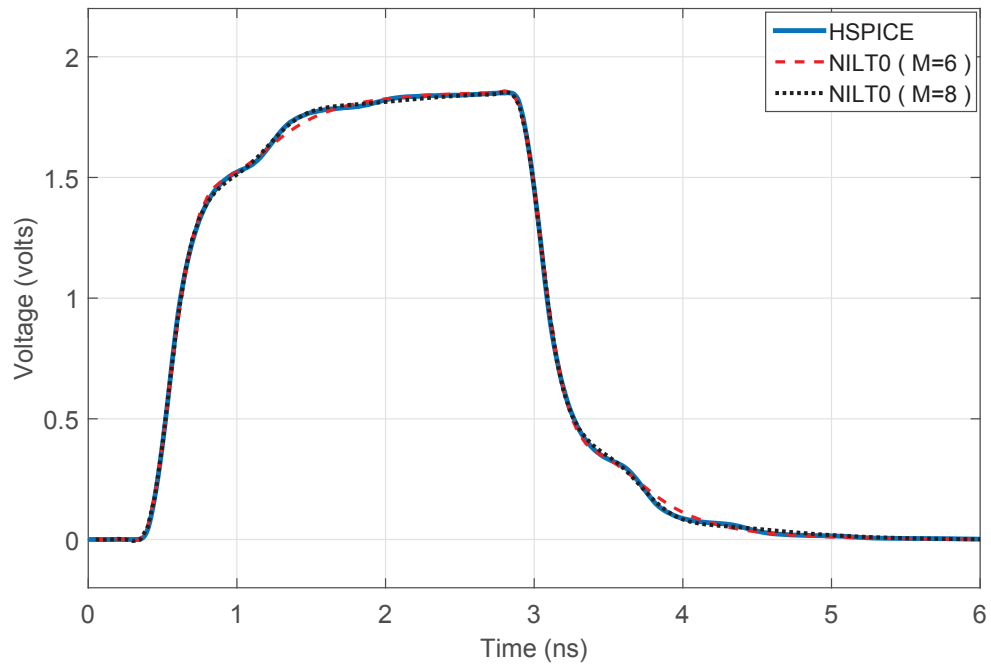


Figure 6.2: Transient analysis at Vout1 using NILT0.

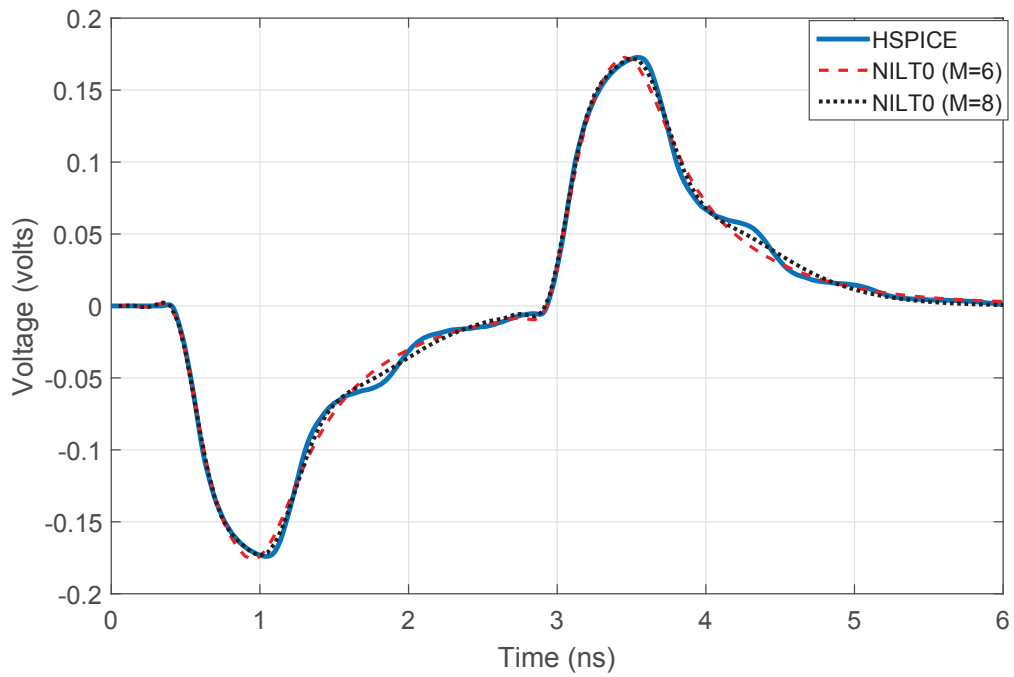


Figure 6.3: Transient analysis at Vout2 using NILT0.

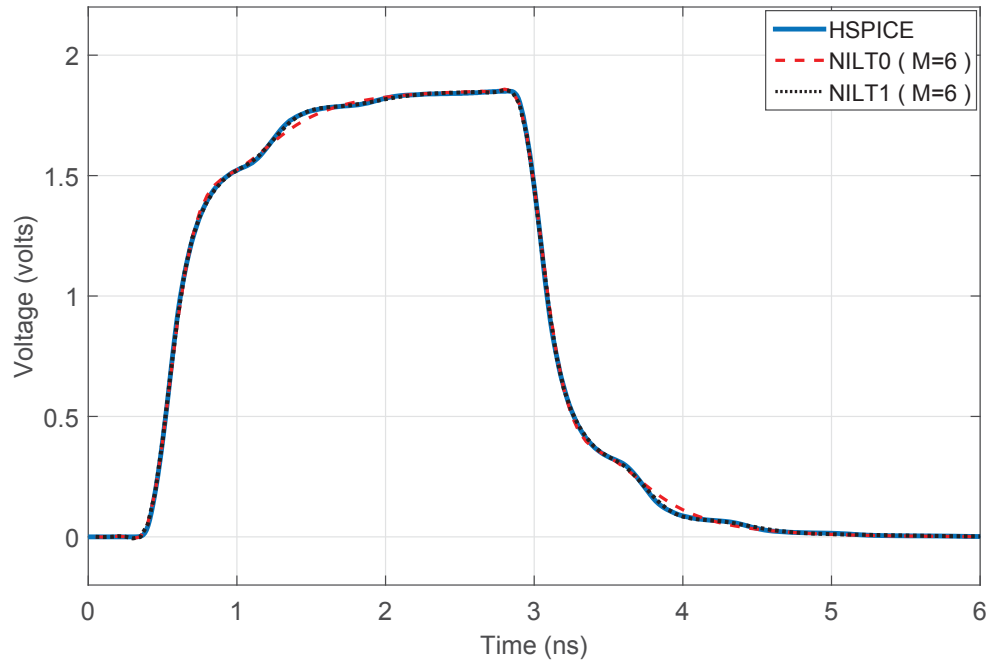


Figure 6.4: Transient analysis at Vout1 using NILT0 and NILT1.

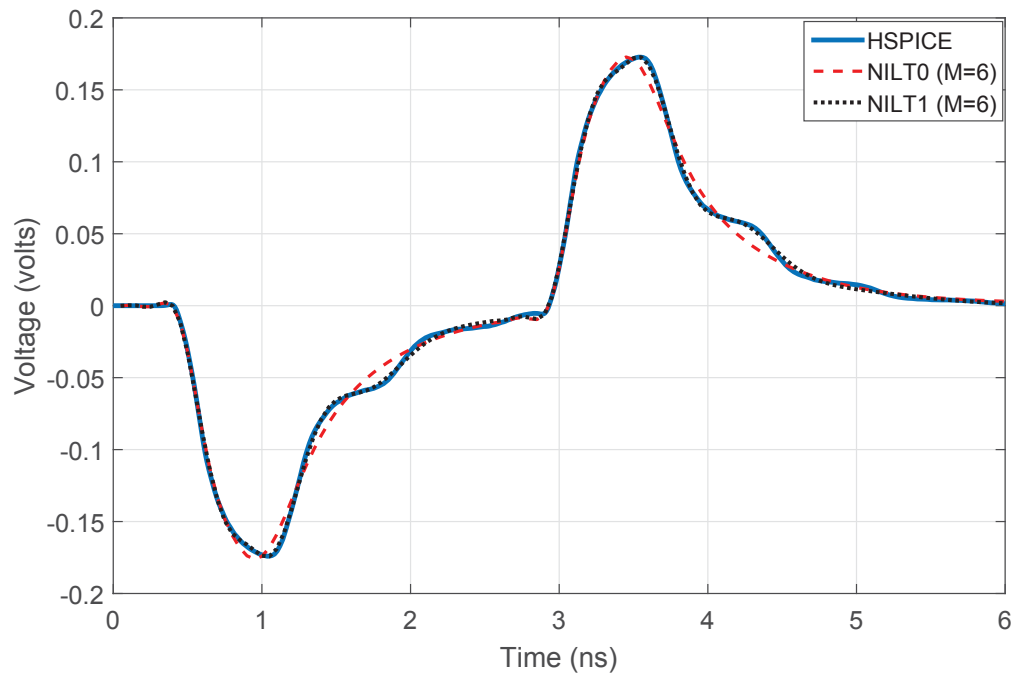


Figure 6.5: Transient analysis at Vout2 using NILT0 and NILT1.

Chapter 7

Numerical Results

An illustrative example has been presented in the previous chapters to demonstrate the validity, accuracy and efficiency of the proposed method as the steps of the method were being developed.

In this chapter, two numerical examples are presented to further demonstrate the accuracy and speedup obtained by the proposed method. The CPU times reported here are based on a PC platform with 8-GB RAM and 2-GHz Intel processor, and the computation is executed in the Matlab 8.6.0 (R2015b) environment.

7.1 Example 1

In this example, we consider a practical parameterized circuit shown in Figure (7.1). The stable active circuit is composed of four coupled transmission lines with the stages of the cascade amplifiers at the far-end of each transmission line. The single input signal $u(t)$ is chosen as a $2V$ pulse signal with $0.1ns$ fall/rise time and $4ns$ pulse-width. Each transmission line has a length of 10cm and it is modeled using 100 sections of lumped RLGC segmentation. The internal model of the amplifiers is shown in Figure (7.3).

The uncertainty in this example arises from the geometric structure of the coupled transmission lines. Six stochastic physical parameters are considered whose nominal values are given in Figure (7.2) and are assumed to have Gaussian distributions with standard deviation of $\pm 20\%$.

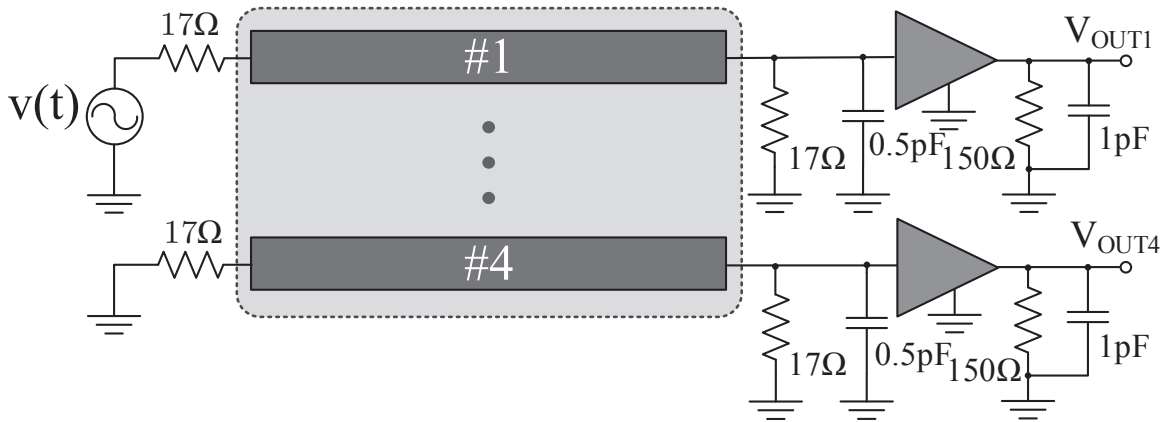
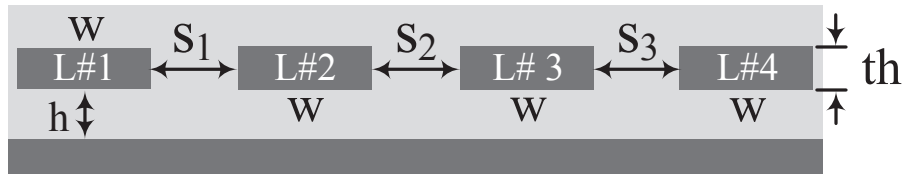


Figure 7.1: A stable active circuit consisting of four coupled transmission lines and four amplifiers.



Conductivity: $\sigma = 5.4e7 \text{ S/m}$
 Relative Permittivity: $\epsilon_r = 4.65$
 Separation: $s_1 = s_2 = s_3 = 60\mu\text{m}$
 Line Width: $w = 180\mu\text{m}$
 Line Height: $h = 180\mu\text{m}$
 Line Thickness: $th = 25\mu\text{m}$

Figure 7.2: Cross section of the transmission lines.

The SC scheme is constructed by using the proposed method with a 13-node sparse grid (accuracy level 2). Table (7.1) shows the exact sampled parameter values and the corresponding weights in the 6-D parameter space.

The statistical distribution of the output response was obtained using the NILT1

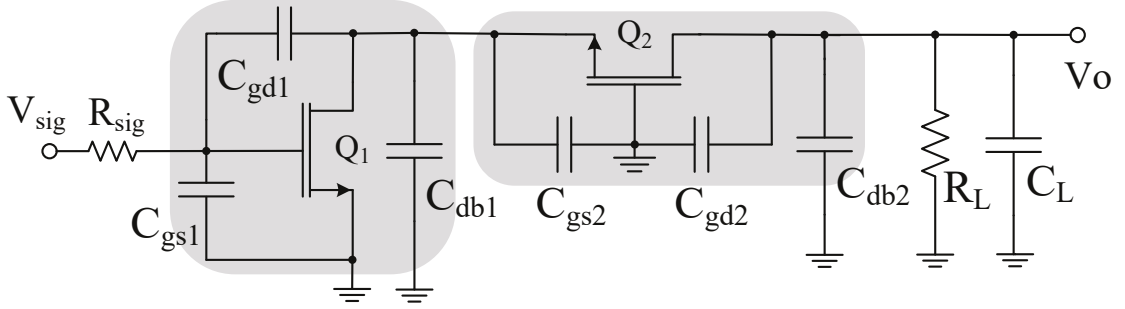


Figure 7.3: The internal circuit of the amplifiers.

Table 7.1: Sparse grid nodes and weights in the six dimensional parameter space

$[H, Th, W, S_1, S_2, S_3]$ (μm)	Weights
[65.36, 25, 180, 60, 60, 60]	0.1667
[100, 16.34, 180, 60, 60, 60]	0.1667
[100, 25, 117.65, 60, 60, 60]	0.1667
[100, 25, 180, 39.22, 60, 60]	0.1667
[100, 25, 180, 60, 39.22, 60]	0.1667
[100, 25, 180, 60, 60, 39.22]	0.1667
[100, 25, 180, 60, 60, 60]	-1
[134.64, 25, 180, 60, 60, 60]	0.1667
[100, 33.66, 180, 60, 60, 60]	0.1667
[100, 25, 242.35, 60, 60, 60]	0.1667
[100, 25, 180, 80.78, 60, 60]	0.1667
[100, 25, 180, 60, 80.78, 60]	0.1667
[100, 25, 180, 60, 60, 80.78]	0.1667

method with $M = 6$. Figures (7.4) and (7.5) show the *mean* and the 3σ tolerances of the output voltage at the far end of lines 1 and 2 based on 2000 MC simulations obtained using the original model and the proposed approach with reduced models. The figures demonstrate an excellent agreement between the results obtained using the two methods.

Table (7.2) shows the CPU-time and the speed-up ratio between these two methods.

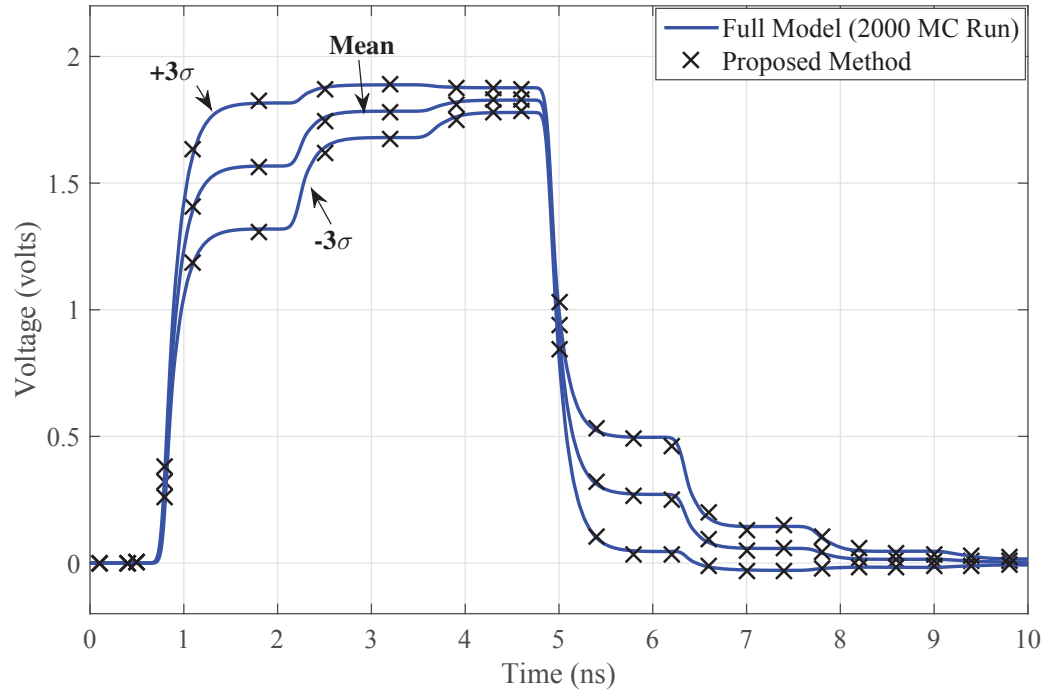


Figure 7.4: Time-domain statistical analysis of the output voltage at the far end of line 1.

Table 7.2: CPU time comparison of example 1

	Size	Sample Number	Time (s)
MC Full-Model Simulation	1222	2000	2953.6
Proposed Method (including the ROM generation time)	36	13	1.75
Speed-up Ratio			≈ 1688

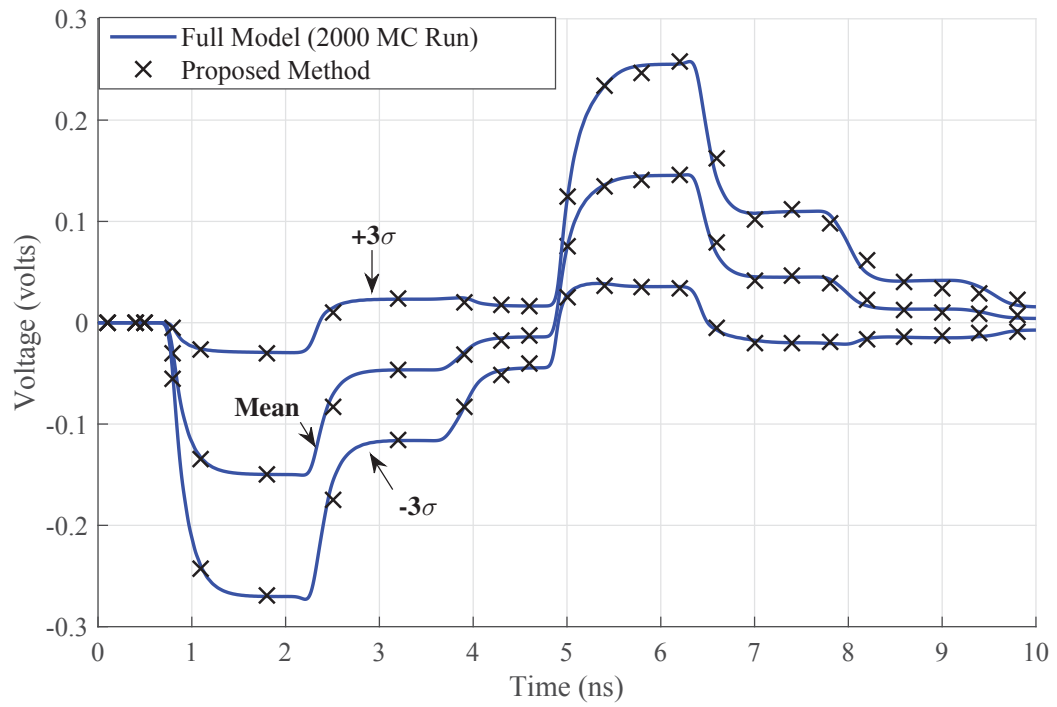


Figure 7.5: Time-domain statistical analysis of the output voltage at the far end of line 2.

7.2 Example 2

In this example, we apply the proposed algorithm to an eight coupled transmission line active circuit shown in Figure (7.6). 100 RLGC sections are used to model each transmission line with a length of 8cm. The internal model of the amplifiers is given in Figure (7.3). Ten stochastic physical parameters are considered whose nominal values are given in Figure (7.7) and are assumed to have Gaussian distributions with standard deviation of $\pm 20\%$

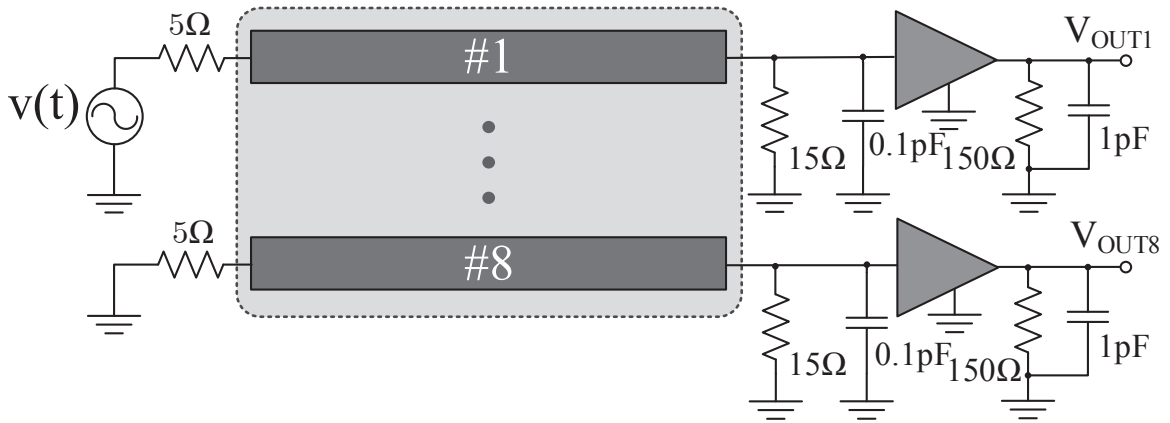
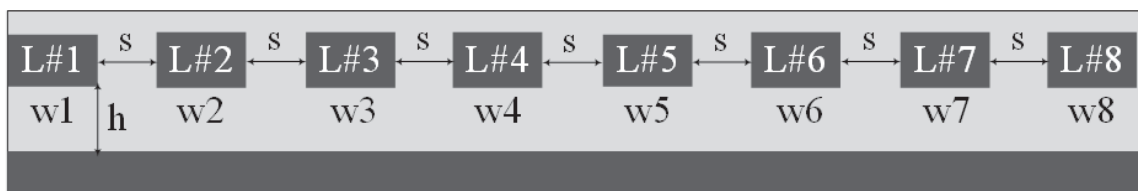


Figure 7.6: A stable active circuit consisting of eight coupled transmission lines and eight amplifiers.



Conductivity: $\sigma = 5.4e7 \text{ S/m}$

Relative Permittivity: $\epsilon_r = 4.65$

Line Width: $w1 = w2 = w3 = w4 = w5 = w6 = w7 = w8 = 180\mu\text{m}$

Separation: $s = 60\mu\text{m}$

Line Height: $h = 180\mu\text{m}$

Figure 7.7: Cross section of the transmission lines.

By using the proposed method, a 21-point sparse grid is constructed. Table (7.3)

Table 7.3: Sparse grid nodes and weights in the ten dimensional parameter space

$[H, W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8, S]$ (μm)	Weights
[65.36, 180, 180, 180, 180, 180, 180, 180, 180, 80]	0.1667
[100, 117.65, 180, 180, 180, 180, 180, 180, 180, 80]	0.1667
[100, 180, 117.65, 180, 180, 180, 180, 180, 180, 80]	0.1667
[100, 180, 180, 117.65, 180, 180, 180, 180, 180, 80]	0.1667
[100, 180, 180, 180, 117.65, 180, 180, 180, 180, 80]	0.1667
[100, 180, 180, 180, 180, 117.65, 180, 180, 180, 80]	0.1667
[100, 180, 180, 180, 180, 180, 117.65, 180, 180, 80]	0.1667
[100, 180, 180, 180, 180, 180, 180, 117.65, 180, 80]	0.1667
[100, 180, 180, 180, 180, 180, 180, 180, 117.65, 80]	0.1667
[100, 180, 180, 180, 180, 180, 180, 180, 180, 52.29]	0.1667
[100, 180, 180, 180, 180, 180, 180, 180, 180, 80]	-2.33
[100, 180, 180, 180, 180, 180, 180, 180, 180, 107.71]	0.1667
[100, 180, 180, 180, 180, 180, 180, 180, 242.35, 80]	0.1667
[100, 180, 180, 180, 180, 180, 180, 242.35, 180, 80]	0.1667
[100, 180, 180, 180, 180, 180, 242.35, 180, 180, 80]	0.1667
[100, 180, 180, 180, 180, 242.35, 180, 180, 180, 80]	0.1667
[100, 180, 180, 180, 242.35, 180, 180, 180, 180, 80]	0.1667
[100, 180, 180, 242.35, 180, 180, 180, 180, 180, 80]	0.1667
[100, 180, 242.35, 180, 180, 180, 180, 180, 180, 80]	0.1667
[100, 242.35, 180, 180, 180, 180, 180, 180, 180, 80]	0.1667
[134.64, 180, 180, 180, 180, 180, 180, 180, 180, 80]	0.1667

shows the exact sampled parameter values and the corresponding weights in the ten-dimensional random space.

The statistical distribution of the output response was obtained using the NILT1 method and $M = 6$. Figures (7.8) and (7.9) present the *mean* and the 3σ tolerances of the output responses at the outputs Vout1 and Vout2 based on 2000 MC simulations obtained using the original models and the proposed approach with reduced models. The excellent agreement proves the accuracy of the proposed method for the large

circuits with multiple parameters. The speedup ratio is given in Table (7.4).

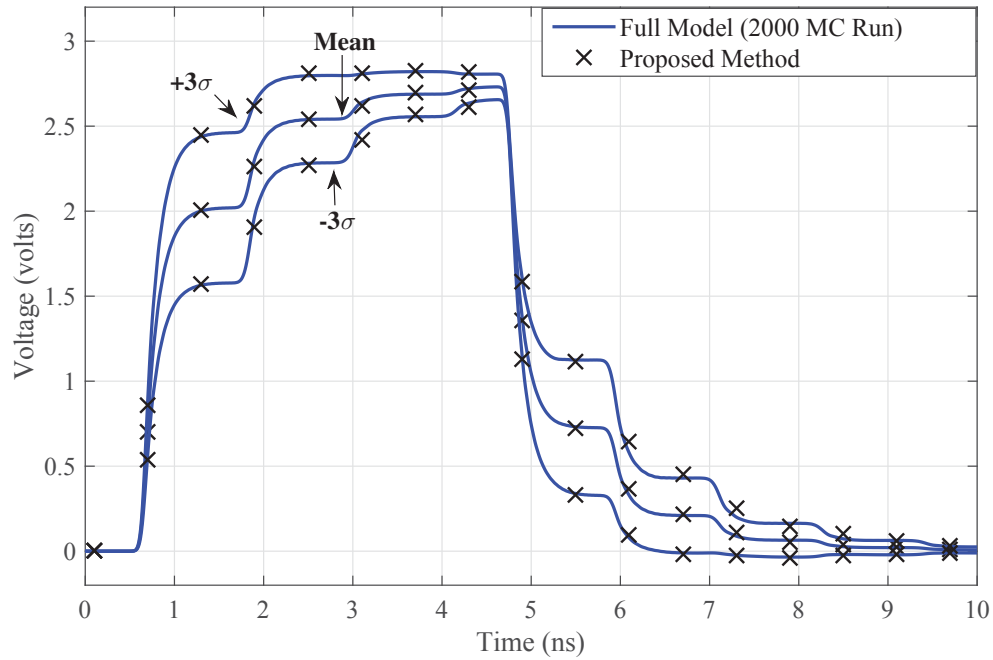


Figure 7.8: Time-domain statistical analysis of the output voltage at the far end of line 1.

Table 7.4: CPU time comparison of example 2

	Size	Sample Number	Time (s)
MC Full-Model Simulation	2442	2000	8047.6
Proposed Method (including the model generation time)	46	21	5.32
Speed-up Ratio			≈ 1513

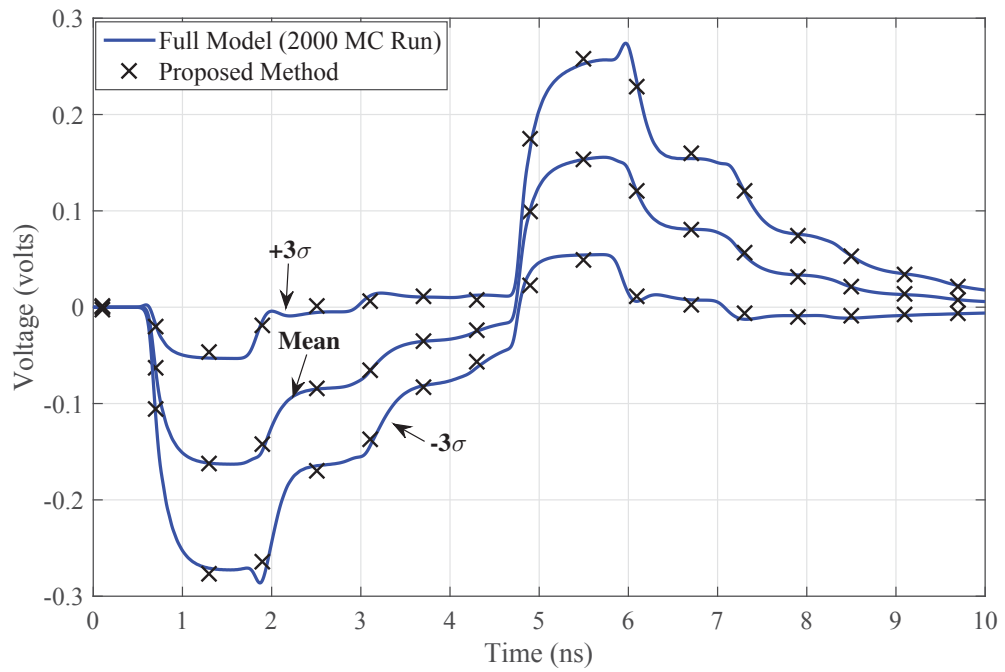


Figure 7.9: Time-domain statistical analysis of the output voltage at the far end of line 2.

Chapter 8

Conclusion and Further Work

8.1 Conclusion

Variability analysis of parameterized circuits based on conventional Monte Carlo technique is computationally expensive. This is mainly due to the following facts: (1) MC approach requires a large number of simulations and the convergence rate of the statistical results is slow; (2) the size of the local deterministic circuits in the MC sample set is very large.

In this thesis, an efficient method has been presented for time-domain variability analysis of large circuits with multiple stochastic parameters. The three main key features of this method have been discussed. The first problem associated with the conventional MC approach is solved using sparse-grid Stochastic Collocation based techniques. The second problem is addressed by constructing stable ROMs for the nodal circuit systems in the SC scheme. The proposed method can be applied to both passive and active circuits. Additionally, NILT is applied to further improve the efficiency of time-domain analysis in the proposed method. Numerical results have confirmed the efficiency and accuracy of the proposed technique in comparison with standard Monte Carlo approach.

8.2 Future Work

For future research, the following directions are foreseen:

- Applying anisotropic sparse grid technique in the proposed method to address the difficulties associated with the stochastic parameters subject to different type of distributions and different level smoothness of the impact on the response.
- Parallel implementation of the proposed method.
- Selecting the Taylor expansion point considering the location of complex frequency points required by the NILT methods, such that the size of the ROMs is further reduced.

List of References

- [1] D. Xiu and G. E. Karniadakis, “The wiener–askey polynomial chaos for stochastic differential equations,” *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.
- [2] D. Spina, F. Ferranti, T. Dhaene, L. Knockaert, G. Antonini, and D. V. Ginsteste, “Variability analysis of multiport systems via polynomial-chaos expansion,” *IEEE Trans. Microw. Theory Tech.*, vol. 60, pp. 2329–2338, Aug. 2012.
- [3] T. A. Pham, E. Gad, M. S. Nakhla, and R. Achar, “Decoupled polynomial chaos and its applications to statistical analysis of high-speed interconnects,” *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 4, pp. 1634–1647, Oct. 2014.
- [4] D. Xiu and J. S. Hesthaven, “High-order collocation methods for differential equations with random inputs,” *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2005.
- [5] I. Babuska, F. Nobile, and R. Tempone, “A stochastic collocation method for elliptic partial differential equations with random input data,” *SIAM Journal on Numerical Analysis*, vol. 45, no. 3, pp. 1005–1034, 2007.
- [6] N. Agarwal and N. R. Aluru, “Weighted smolyak algorithm for solution of stochastic differential equations on non-uniform probability measures,” *International Journal for Numerical Methods in Engineering*, vol. 85, no. 11, pp. 1365–1389, 2011.
- [7] K. Guo, F. Ferranti, B. Nouri, and M. Nakhla, “A stochastic collocation technique for time-domain variability analysis of active circuits.” Proc. 25th IEEE Conference on Electrical Performance of Electronic Packaging and Systems, Oct. 2016. to be published.

- [8] C.-W. Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” *IEEE Transactions on Circuits and Systems*, vol. 22, pp. 504–509, Jun 1975.
- [9] P. Triverio, S. Grivet-Talocia, M. Nakhla, F. Canavero, and R. Achar, “Stability, causality, and passivity in electrical interconnect models,” *IEEE Transactions on Advanced Packaging*, vol. 30, pp. 795–808, Nov. 2007.
- [10] L. Castaneda, V. Arunachalam, and S. Dharmaraja, *Introduction to Probability and Stochastic Processes with Applications*. Wiley, 2012.
- [11] H. A. v. d. V. Wilhelmus H. A. Schilders and J. Rommes, *Model order reduction: theory, research aspects and applications.*, vol. 13. Springer Berlin Heidelberg, 2008.
- [12] A. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, 2005.
- [13] B. N. Bond and L. Daniel, “Guaranteed stable projection-based model reduction for indefinite and unstable linear systems,” in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, ICCAD '08*, (Piscataway, NJ, USA), pp. 728–735, IEEE Press, 2008.
- [14] R. C. Selga, B. Lohmann, and R. Eid, “Stability preservation in projection-based model order reduction of large scale systems,” *European Journal of Control*, vol. 18, no. 2, pp. 122 – 132, 2012.
- [15] I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, and M. Barone, “Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment,” *Computer Methods in Applied Mechanics and Engineering*, vol. 272, pp. 251 – 270, 2014.
- [16] I. M. Jaimoukha and E. M. Kasenally, “Implicitly restarted krylov subspace methods for stable partial realizations,” *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 3, pp. 633–652, 1997.
- [17] X. Deng, B. Nouri, and M. S. Nakhla, “Stability preserving algorithm for model order reduction of active networks,” in *Electrical Performance of Electronic Packaging and Systems (EPEPS)*, pp. 181–184, Oct. 2015.

- [18] K. Carlberg, C. Bou-Mosleh, and C. Farhat, “Efficient non-linear model reduction via a least-squares petrov-galerkin projection and compressive tensor approximations,” *International Journal for Numerical Methods in Engineering*, vol. 86, no. 2, pp. 155–181, 2011.
- [19] Z. Bai, “19th dundee biennial conference on numerical analysis krylov subspace techniques for reduced-order modeling of large-scale dynamical systems,” *Applied Numerical Mathematics*, vol. 43, no. 1, pp. 9 – 44, 2002.
- [20] R. W. Freund, “Passive reduced-order modeling via krylov-subspace methods,” in *Computer-Aided Control System Design, 2000. CACSD 2000. IEEE International Symposium on*, pp. 261–266, 2000.
- [21] A. Odabasioglu, M. Celik, and L. T. Pileggi, “Prima: passive reduced-order interconnect macromodeling algorithm,” in *Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on*, pp. 58–65, Nov 1997.
- [22] A. Odabasioglu, M. Celik, and L. T. Pileggi, “Prima: passive reduced-order interconnect macromodeling algorithm,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 645–654, 1998.
- [23] P. B. Ulrike Baur and L. Feng, “Model order reduction for linear and nonlinear systems: a system-theoretic perspective,” in *Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on*, pp. 58–65, Nov 1997.
- [24] B. Bond and L. Daniel, “Parameterized model order reduction of nonlinear dynamical systems,” in *Proceedings of the 2005 IEEE/ACM International Conference on Computer-aided Design, ICCAD '05*, (Washington, DC, USA), pp. 487–494, IEEE Computer Society, 2005.
- [25] P. K. Gunupudi, R. Khazaka, M. S. Nakhla, T. Smy, and D. Celso, “Passive parameterized time-domain macromodels for high-speed transmission-line networks,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, pp. 2347–2354, Dec 2003.
- [26] U. Baur, C. Beattie, P. Benner, and S. Gugercin, “Interpolatory projection methods for parameterized model reduction,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2489–2518, 2011.

- [27] P. Benner and L. Feng, "A Robust Algorithm for Parametric Model Order Reduction Based on Implicit Moment Matching", pp. 159–185. Cham: Springer International Publishing, 2014.
- [28] W. Gautschi, *Numerical Analysis*. SpringerLink : Bücher, Birkhäuser Boston, 2011.
- [29] F. Ferranti, M. Nakhla, G. Antonini, T. Dhaene, L. Knockaert, and A. E. Ruehli, "Interpolation-based parameterized model order reduction of delayed systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, pp. 431–440, March 2012.
- [30] I. Babuka, F. Nobile, and R. Tempone, "A stochastic collocation method for elliptic partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 45, no. 3, pp. 1005–1034, 2007.
- [31] S. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Dokl. Akad. Nauk SSSR*, vol. 4, p. 240243, 1963.
- [32] J. Ma, V. Rokhlin, and S. Wandzura, "Generalized gaussian quadrature rules for systems of arbitrary functions," *SIAM Journal on Numerical Analysis*, vol. 33, no. 3, pp. 971–996, 1996.
- [33] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, October 1992.
- [34] T. Zhou and T. Tang, "Convergence Analysis For Spectral Approximation To A Scalar Transport Equation With A Random Wave Speed," *Journal Of Computational Mathematics*, vol. 30, no. 6, pp. 643–656, 2012.
- [35] A. Kronrod, *Nodes and weights of quadrature formulas: sixteen-place tables*. Consultants Bureau, 1965.
- [36] T. N. L. Patterson, "The optimum addition of points to quadrature formulae," *Mathematics of Computation*, vol. 104, no. 22, pp. 847–856, 1968.
- [37] S. Mehrotra and D. Papp, "Generating nested quadrature formulas for general weight functions with known moments," *CoRR*, vol. abs/1203.1554, 2012.
- [38] T. J. R. Walter Gautschi, "A family of gauss-kronrod quadrature formulae," *Mathematics of Computation*, vol. 51, no. 184, pp. 749–754, 1988.

- [39] G. Monegato, *An Overview of Results and Questions Related to Kronrod Schemes*, pp. 231–240. Basel: Birkhäuser Basel, 1979.
- [40] R. Bourquin, “Exhaustive search for higher-order kronrod-patterson extensions,” Tech. Rep. 2015-11, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2015.
- [41] M. Griebel, M. Schneider, and C. Zenger, “A combination technique for the solution of sparse grid problems,” in *Iterative Methods in Linear Algebra* (P. de Groen and R. Beauwens, eds.), pp. 263–281, IMACS, Elsevier, North Holland, 1992.
- [42] T. Gerstner and M. Griebel, “Numerical integration using sparse grids,” *Numerical Algorithms*, vol. 18, no. 3, pp. 209–232, 1998.
- [43] F. Nobile, R. Tempone, and C. G. Webster, “An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data,” *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2411–2442, 2008.
- [44] A. Genz and B. Keister, “Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight,” *J. Comput. Appl. Math.*, vol. 71, no. 2, pp. 299–309, 1996.
- [45] K. Singhal and J. Vlach, “Computation of time domain response by numerical inversion of the laplace transform,” *Journal of the Franklin Institute*, vol. 299, no. 2, pp. 109 – 126, 1975.
- [46] J. Griffith and M. Nakhla, “Time-domain analysis of lossy coupled transmission lines,” vol. 38, pp. 1480–1487, Oct. 1990.
- [47] R. Griffith, “Mixed frequency/time domain analysis of high-speed interconnects,” m.a.sc. thesis, Carleton University, Ottawa, Canada, July 1993.
- [48] K. Singhal, J. Vlach, and M. Nakhla, “Absolutely stable, high order method for time domain solution of networks,” *Archiv fr Elektronik und bertragungstechnik*, vol. 30, pp. 157–166, 1976.
- [49] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Electrical Engineering, Springer, 1994.
- [50] R. Griffith and M. Nakhla, “A new high-order absolutely-stable explicit numerical integration algorithm for the time-domain simulation of nonlinear circuits,”

in *Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on*, pp. 276–280, Nov 1997.