THESIS

EFFICIENT MULTIDIMENSIONAL UNCERTAINTY QUANTIFICATION OF HIGH SPEED CIRCUITS USING ADVANCED POLYNOMIAL CHAOS APPROACHES

Submitted by

Majid Ahadi Dolatsara

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2016

Master's Committee:

Adviser: Sourajeet Roy

Branislav Notaros
Chuck Anderson
Ali Pezeshki

ABSTRACT

EFFICIENT MULTIDIMENSIONAL UNCERTAINTY QUANTIFICATION OF HIGH SPEED CIRCUITS USING ADVANCED POLYNOMIAL CHAOS APPROACHES

With the scaling of VLSI technology to sub-45 nm levels, uncertainty in the nanoscale manufacturing processes and operating conditions have been found to result in unpredictable circuit behavior at the chip, package, and board levels of modern integrated microsystems. Hence, modeling the forward propagation of uncertainty from the device-level parameters to the system-level response of high-speed circuits and systems forms a crucial requirement of modern computer-aided design (CAD) tools. This thesis presents novel approaches based on the generalized polynomial chaos (gPC) theory for the efficient multidimensional uncertainty quantification of general distributed and lumped high-speed circuit networks. The key feature of this work is the development of approaches which are more efficient and/or accurate comparing to recently suggested uncertainty quantification approaches in the literature. Main contributions of this thesis are development of two individual approaches for improvement of the conventional linear regression uncertainty quantification approach, and development of a sparse polynomial expansion of the stochastic response in an uncertain system. The validity of this work is established through multiple numerical examples.

ACKNOWLEDGEMENTS

I like to express my gratitude to my adviser, Dr. Sourajeet Roy, and my master's committee members, Dr. Branislav Notaros, Dr. Chuck Anderson and Dr. Ali Pezeshki. Besides I thank my collaborators and coauthors in my publications, including but not limited to Aditi Krishna Prasad, Muhammad Kabir and Dr. Roni Khazaka as this work would not be possible without their contributions. Finally, I appreciate support of my friends and family which made my experience in the Colorado State University a truly wonderful one.

DEDICATION

This work is dedicated to my parents for their endless love and support.

TABLE OF CONTENTS

CHAPTER I: INTRODUCTION

In today's fast paced environment of electrical design and fabrication, simulation tools have proved to be a great asset to electrical engineers [1]. In the past decades, we have been able to predict the behavior of a circuit through simulation before any physical implementation. These tools are essential to the industry since they save a great deal of time and money. Today instead of fabricating expensive prototypes and time consuming electromagnetic compatibility tests [2], it is possible to predict a lot of details a priori to the fabrication phase. This has been made practical due to great advancements in high speed computations and numerical analysis, which has resulted in development of advanced SPICE-like circuit simulators [3], finite element and finite volume analysis, and several advanced new techniques which provide more efficient and accurate results. However, by advancements of the fabrication technology to nanoscales, new challenges have raised which motivates new areas of research. This thesis is the report of an endeavor in addressing such problems.

1.1 Problem statement

With the scaling of VLSI technology to sub-45 nm levels, uncertainty in the nanoscale manufacturing processes and operating conditions have been found to result in unpredictable behavior of high speed circuits. As a result, contemporary computer aided design (CAD) tools need to be flexible enough to be able to predict the impact of parametric uncertainty on general circuit responses. Traditionally, uncertainty quantification of circuit networks has been performed using the brute-force Monte Carlo approach [4]-[9]. Despite the simplicity of this approach, its slow convergence translates to a prohibitively large number of deterministic simulations of the original network model in order to achieve accurate statistical results. This makes the Monte Carlo approach computationally infeasible for analyzing large networks [10].

Recently, more robust uncertainty quantification techniques based on the generalized polynomial chaos (PC) theory have been reported for various high-speed circuit, electromagnetic (EM) and electronic

packaging problems [10]-[44]. These techniques attempt to model the uncertainty in the network response as an expansion of predefined orthogonal polynomial basis functions of the input random variables. The coefficients of the expansion form the new unknowns of the system and are evaluated via intrusive or non-intrusive approaches [37].

The existing literature in circuit and EM simulation has been dominated by the highly accurate but intrusive stochastic Galerkin (SG) approach [10]-[25]. This approach requires the solution of a single but augmented coupled deterministic network model to determine the PC coefficients. This issue is further exacerbated for nonlinear networks since the pertinent inner product operations have to be approximated using a quadrature method where each quadrature term is represented using a large number of additional dependent voltage/current sources [18]. Overall, the simulation costs of such large models scale in a near-exponential manner with the number of random dimensions. While recent works such as the decoupled PC algorithm [20] and the stochastic testing method [40], [41] can mitigate the time and memory costs of the standard SG approach, both these approaches require the development of intrusive codes that preclude the direct exploitation of SPICE-like legacy circuit simulators. These bottlenecks have limited the applicability of the SG approach to problems featuring only low-dimensional random spaces [45], [46].

On the other hand, non-intrusive PC approaches such as the stochastic collocation (SC) approach, pseudo-spectral collocation approach and linear regression approach, among others, have recently been explored for circuit and EM problems as well [26]-[39]. The advantage of these non-intrusive approaches over the intrusive SG approach lies in their ability to compute the PC coefficients of the network responses by simply probing the original model at a sparse set of nodes located within the random space [45]. The deterministic simulation of the network at each node can be performed by a direct invocation of SPICE without the need for any intrusive coding. In addition, the relevant deterministic simulations can be parallelized unlike the conventional SG approach where the augmented network is always coupled.

1.2 Goals of the thesis

Among non-intrusive approaches, the linear regression approach has been found to be highly popular [27], [28], [46]. This approach probes the PC expansion of the network responses at an oversampled set of multidimensional nodes located within the random space, thereby leading to the formulation of an overdetermined set of linear algebraic equations. These equations can be solved in a least-square sense to directly evaluate the PC coefficients of the network responses [46]. Typically, the multidimensional regression nodes are chosen from the tensor product grid of one dimensional (1D) quadrature nodes [27], [28]. Since the number of nodes in the tensor product grid increases exponentially with the number of random dimensions, realistically only a sparse subset of the nodes, also referred to as design of experiments (DoE), can be chosen. In the work of [39], it was demonstrated that blindly choosing the DoE can lead to inaccurate evaluation of the PC coefficients. However, the contemporary literature on linear regression based PC analysis of EM and circuit problems have not identified any specific formal criterion for choosing the best set of DoE [27], [28]. Recently, the stochastic testing approach has developed a reliable technique to select possible DoE where the number of DoE is equal to the number of unknown PC coefficients [40], [41]. However, this technique does not choose the DoE using any optimal criterion and hence does not guarantee the maximum possible accuracy of results.

This thesis firstly presents new techniques for improvement of the linear regression methodology in order to address the above issues. The first contribution is development of a sparse linear regression (SPLINER) approach and then more improvements are suggested based on the D-optimal criterion for choosing the DoE. The proposed approaches insist that for the most accurate evaluation of the PC coefficients, the corresponding DoE have to be so chosen such that the determinant of the information matrix in the linear regression problem is maximized [39]. In other words, this thesis proposes greedy search algorithms in order to identify the D-optimal DoE from multidimensional random spaces. The proposed search algorithm begins with an arbitrary set of DoE chosen from the tensor product grid of 1D quadrature nodes and then sequentially replaces each DoE in that initial set with the best possible

substitute selected from the remaining set of quadrature nodes. The best possible substitute DoE is chosen to be the one that increases the current determinant of the information matrix by the largest amount. This step-by-step refinement of the starting set of DoE continues till all of them have been replaced at which point the new set forms the D-optimal DoE [39]. Novel numerical strategies to expedite the search of the substitute DoE for problems involving high-dimensional random spaces have been developed in this thesis.

Furthermore, in order to increase the efficiency of PC approaches a novel methodology for curbing the scaling of computational costs with respect to number of random variables is provided. In fact, in most of the state of the art PC approaches the computational cost increases in an exponential or near exponential rate with respect to number if random variables. This increase in CPU cost is called curse of dimensionality and is the main bottleneck for PC approaches which has limited their use to a relatively limited number of dimensions. In this thesis a novel approach is suggested to address the curse of dimensionality. In this approach based on sparsity of effects [47], [48] more impactful polynomial bases are selected by switching the linear criterion of the regular PC approach to a hyperbolic criterion. Coefficients of these polynomial bases can be found using any state of the art PC approach; however, in this thesis the focus is on nonintrusive approaches. Then the desired statistics are found from the coefficients with marginal loss in accuracy. Since number of new unknowns is noticeably less than the regular PC approach the CPU cost and number of deterministic simulations is only a fraction of the original amount. Moreover, an adaptive methodology is developed to determine how many impactful bases are needed based on the desired accuracy. The main advantage of this approach is reducing the increase rate of CPU costs with minimal loss in accuracy. This claim has been proved through multiple examples including distributed and nonlinear networks.

1.3 Organization of the text

This thesis tries to be self-explanatory and without a major need to previous knowledge in uncertainty quantification for the reader. Most of the state of the art PC approaches are reviewed.

Exploited techniques are explained in details, and novel ideas are supported with extensive numerical examples and discussions. The rest of text is organized as follows: Chapter II provides a review of basics of the generalized PC theory and the most common intrusive and nonintrusive uncertainty quantification approaches including stochastic Galerkin [10]-[25], stochastic collocation [49] and the linear regression approach [27], [28]. Moreover, the major advantages and disadvantages of these approaches are provided in this chapter. Chapter III is dedicated to improvements to the linear regression approach. This chapter is divided to three main sections; the first section reviews the D-optimal criterion and the Fedorov search algorithm for the linear regression approach. In the second section the proposed SPLINER approach is discussed, and in the third section more improvements on the linear regression approach are suggested. All contributions in this chapter are validated by numerical examples. In Chapter IV, firstly further details about the PC theory is provided which is used in rest of the chapter, then the improvement on PC approaches with the use of the hyperbolic criterion and its adaptive methodology are discussed. This approach is mainly compared with a nonintrusive stochastic testing based approach [37] and the contributions are validated using multiple numerical examples. Finally the thesis is ended with a conclusion in Chapter V.

CHAPTER II: EXISTING UNCERTAINTY QUANTIFICATION APPROACHES

In this chapter an extended review of currently available uncertainty quantification approaches is discussed. To do so, it is imperative to be familiar with the generalized Polynomial Chaos (gPC) theory. Therefore gPC is introduced at the beginning of this chapter. Then we discuss the uncertainty quantification problem using different intrusive and nonintrusive approaches. The considered intrusive approaches are Stochastic Galrekin (SG) [10]-[25] and Stochastic Testing [40], [41]. On the other hand, in the nonintrusive section since the focus of this work is on nonintrusive methods more methods are discussed. This includes Pseudo Spectral Stochastic Collocation [29], classical linear regression [27], [28] Stochastic Collocation [49] and Stroud cubature rules [50], [51]. At the end of each section comparative analysis of that approach with respect to other techniques is discussed. Finally, at the end of this chapter a numerical example is provided to present the application of numerical uncertainty approaches.

2.1 Generalized Polynomial Chaos Theory

The concept of the orthogonal polynomials has been in mathematics records for a long time [52], [53]. However, due to emergence of uncertainty problems and its challenges, orthogonal polynomials have become popular among engineers, especially in mechanical and electrical engineering. The work of [54] revived the idea of generalized polynomial chaos (gPC) expansion and exploits it to analyze stochastic differential equations. Afterwards the idea flourished and was expanded to different fields and applications [10], [55], [56]. To name a few, polynomial chaos is exploited in the Stochastic Galerkin [10]-[25], stochastic testing [40], [41], pseudo spectral gPC [29] and linear regression [27], [28], which will be discussed later in this chapter.

2.1.1 Basics of the gPC theory

One noticeable application of gPC is to extract statistics of stochastic systems by modeling the output as an expansion of orthogonal polynomials. Based on the [55], output of a system with a one-dimensional random variable can be written as:

$$X(\lambda) = \sum_{i=0}^{\infty} c_i \phi_i(\lambda) \tag{2.1}$$

where λ is the random variable, X is the output, c_i is a scalar coefficient and $\phi_k(\lambda)$ represents orthogonal bases with respect to probability distribution function (PDF) of λ. In order to make (2.1) practical it is truncated to:

$$X(\lambda) \approx \sum_{i=0}^{m} c_i \phi_i(\lambda) \tag{2.2}$$

where m represents the order of expansion and there are $m + 1$ terms in the expansion.

The property that makes the orthogonal polynomials attractive is that the bases are orthogonal with respect to the PDF of λ; hence,

$$\left\langle \phi_i(\lambda), \phi_j(\lambda) \right\rangle = \int_{\Omega} \phi_i(\lambda)\phi_j(\lambda)\rho(\lambda)d\lambda = \alpha_i^2 \delta_{i,j} \tag{2.3}$$

where $\langle , \rangle$ denotes the inner product, Ω is the random space, ρ represents the PDF of λ, α_i^2 is a constant scaler and $\delta_{i,j}$ represents the delta function. It is worth noting in (2.3) the inner product of bases is zero unless $i = j$. In this thesis the polynomial bases are normalized by a factor of α_i for simplicity purposes. Hence they are called orthonormal polynomials.

The above mentioned polynomials are chosen based on the Wiener-Askey scheme [55], where it can be decided which bases gives a faster convergence rate for a certain distributions. The corresponding class of orthogonal polynomials with respect to common standard distributions can be found in Table 2.1, however, theoretically orthogonal polynomials can be derived for any arbitrary distribution.

Table 2.1 Common distributions and the corresponding Wiener-Askey polynomials

Distribution of λ	Wiener-Askey chaos polynomials	Support
Gaussian	Hermite	$(-\infty, +\infty)$
Uniform	Legendre	[-1,1]
Gamma	Laguerre	$[0, +\infty)$
Beta	Jacobi	[-1,1]

2.1.2 Generation of 1-D orthonormal polynomials

Consider the standard normal distribution $N(0,1)$ where $\rho(\lambda) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\lambda^2}{2}}$. By using (2.3) it can be proven that Hermite polynomials are orthogonal to this distribution. These polynomials can be generated either analytically [10]:

$$\phi_i(\lambda) = (-1)^i e^{\lambda^2/2} \frac{d^i}{d\lambda^i} e^{-\lambda^2/2} \tag{2.4}$$

or recursively:

$$\phi_{i+1}(\lambda) = \lambda\phi_i(\lambda) - i\phi_{i-1}(\lambda) \tag{2.5}$$

where $\phi_0(\lambda) = 1$, $\phi_1(\lambda) = \lambda$ and $i > 1$. As mentioned before, all polynomials in this thesis are normalized by α_i^2 in (2.2), which can be found as:

$$\alpha_i^2 = \langle \phi_i(\lambda), \phi_i(\lambda) \rangle = i! \tag{2.6}$$

Furthermore, the uniform distribution $U(-1,1)$ is defined as:

$$\rho(\lambda) = \begin{cases} 0.5, & -1 \le \lambda \le 1 \\ 1, & \text{Otherwise} \end{cases} \tag{2.7}$$

and by using (2.3) it can be proven that Legendre polynomials are orthogonal to this distribution. These polynomials can be generated either analytically [10]:

$$\Phi_i(\lambda) = \frac{1}{2^i i!} \frac{d^i}{d\lambda^i} (\lambda^2 - 1)^i \tag{2.8}$$

or recursively:

$$\Phi_{i+1}(\lambda) = \frac{2i+1}{i+1}\lambda\Phi_i(\lambda) - \frac{i}{i+1}\Phi_{i-1}(\lambda) \tag{2.9}$$

where $\Phi_0 = 1$, $\Phi_1 = \lambda$ and $i > 1$. As mentioned before, all polynomial in this thesis are normalized by α_i^2 in (2.2), which can be found as:

$$\alpha_i^2 = \langle \phi_i, \phi_i \rangle = \frac{1}{2i+1} \tag{2.10}$$

For illustration purposes the first six univariate orthonormal Hermite and Legendre polynomials are listed in Table 2.2.

Table 2.2 The first six univariate orthonormal Hermite and Legendre polynomials

Bases	Orthonormal Hermite Polynomial	Orthonormal Legendre Polynomial
$\Phi_0(\lambda)$	1	1
$\Phi_1(\lambda)$	λ	$\sqrt{3}\,\lambda$
$\Phi_2(\lambda)$	$(\lambda^2 - 1) / \sqrt{2}$	$\sqrt{5}\,(\frac{3}{2}\lambda^2 - \frac{1}{2})$
$\Phi_3(\lambda)$	$(\lambda^3 - 3\lambda) / \sqrt{6}$	$\sqrt{7}\,(\frac{5}{2}\lambda^3 - \frac{3}{2}\lambda)$
$\Phi_4(\lambda)$	$(\lambda^4 - 6\lambda^2 + 3) / (2\sqrt{6})$	$3\,(\frac{35}{8}\lambda^4 - \frac{30}{8}\lambda^2 + \frac{3}{8})$
$\Phi_5(\lambda)$	$(\lambda^5 - 10\lambda^3 + 15\lambda) / (2\sqrt{30})$	$\sqrt{7}\,(\frac{63}{8}\lambda^5 - \frac{70}{8}\lambda^3 + \frac{15}{8}\lambda)$

In the general case, the corresponding i+1th degree orthogonal monic polynomial function can be generated via the three term recurrence relation [45]

$$\begin{aligned} f_{i+1}(\lambda) &= (\lambda - \alpha_i) f_i(\lambda) - \beta_i f_{i-2}(\lambda) \\ \alpha_i &= \frac{\int_\Omega \lambda\, f_i^2(\lambda)\rho(\lambda)\,d\lambda}{\int_\Omega f_i^2(\lambda)\rho(\lambda)\,d\lambda}; \qquad \beta_{i+1} = \frac{\int_\Omega f_{i+1}^2(\lambda)\rho(\lambda)\,d\lambda}{\int_\Omega f_i^2(\lambda)\rho(\lambda)\,d\lambda} \end{aligned} \tag{2.11}$$

where $f_{-1}(\lambda) = 0$, $f_0(\lambda) = 1$, and $\beta_0 = 1$.

2.1.3 Generation of multidimensional orthonormal polynomials

In the topic of uncertainty most of the practical problems involve more than one random variable; therefore, in order to analyze these problems it is essential to consider multidimensional polynomials. However, basics of gPC stay the same. In (2.2) λ changes to the vector $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$, with T being the transpose sign, which represents n mutually uncorrelated random variables. In other words

$$X(\boldsymbol{\lambda}) \approx \sum_{i=0}^{P} c_i \phi_i(\boldsymbol{\lambda}) \tag{2.12}$$

where $\Phi_i(\boldsymbol{\lambda})$ represents multidimensional orthonormal polynomials, and P+1 is number of polynomial bases in the multidimensional case and is equal to

$$P + 1 = \binom{m+n}{m} = \frac{(m+n)!}{m!n!} \tag{2.13}$$

with m being the common expansion degree of each dimension. Also in (2.3) in addition to polynomials, $\rho(\lambda)$ turns to the joint PDF of all random variables and the search space Ω converts to a multidimensional random space i.e. for orthonormal multidimensional polynomial bases we have

$$\left\langle \phi_i(\boldsymbol{\lambda}), \phi_j(\boldsymbol{\lambda}) \right\rangle = \int_{\Omega} \phi_i(\boldsymbol{\lambda}) \phi_j(\boldsymbol{\lambda}) \rho(\boldsymbol{\lambda}) d\boldsymbol{\lambda} = \delta_{i,j} \tag{2.14}$$

It is worth noting, in this thesis even for multidimensional cases α_i^2 is equal to one since normalized univariate polynomials are used in (2.14).

The multivariate polynomials are presented as product of 1D bases as shown in:

$$\Phi_d(\boldsymbol{\lambda}) = \prod_{j=1}^{n} \Phi_{d_j}(\lambda_j) \tag{2.15}$$

with d_j being the index of j-th 1D basis. It should be noted that 1D bases of (2.15) can be from different families. The traditional scheme to determine d_j indices of polynomials is [10]:

$$d_1 + d_2 + \cdots + d_n \leq m \tag{2.16}$$

The graphical interpretation of this scheme for the case of two random variables and the third order expansion, m = 3, is illustrated in Fig. 2.1. This illustration can be expanded to the general case of n random variables, where the polynomials are located at positive coordinates and are bounded by an n dimensional surface noted as: $\sum d_i = m$.

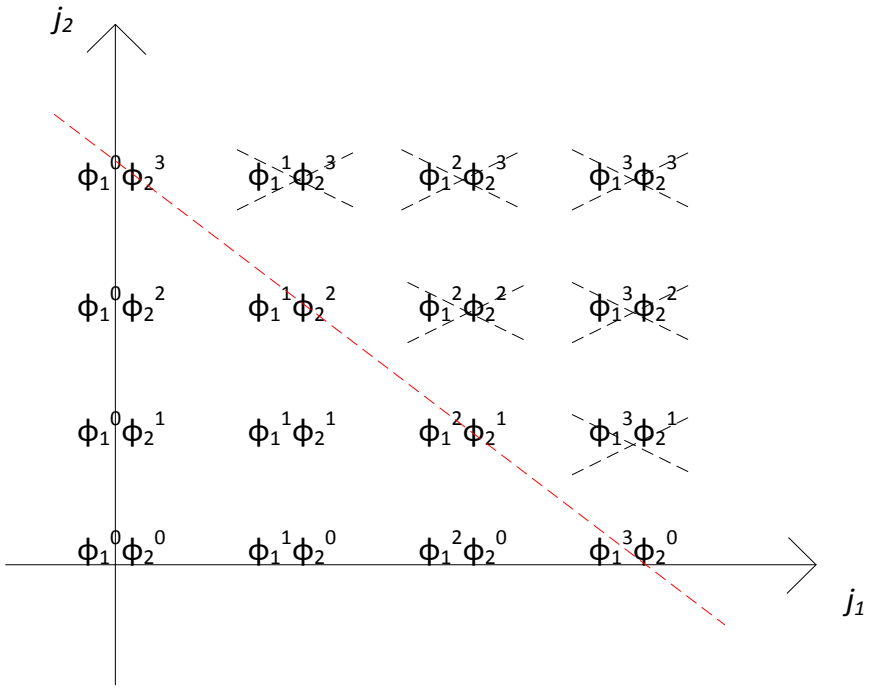


Fig. 2.1: Graphical illustration of the scheme for selection of multivariate polynomials for the case of n = 2 and m = 3.

For illustration purposes the first 10 orthonormal Hermite polynomials for two random variables with the standard normal distribution and also Legendre polynomials for two random variables with the uniform distribution are listed in Table 2.3.

2.1.4 Derivation of statistical information using PC coefficients

As stated before, the main goal of this thesis is derivation of statistical information of stochastic systems. Most of the statistical information are defined by integration over the random space, and entering PC expansions in the integration formulas, helps to derive an analytical formula to obtain them.

Table 2.3 The first ten orthonormal two-dimensional Hermite and Legendre polynomials

Bases	Orthonormal Hermite Polynomial	Orthonormal Legendre Polynomial	Total degree
$\Phi_0(\boldsymbol{\lambda})$	1	1	0
$\Phi_1(\boldsymbol{\lambda})$	λ_1	$\sqrt{3}\,\lambda_1$	1
$\Phi_2(\boldsymbol{\lambda})$	λ_2	$\sqrt{3}\,\lambda_2$	1
$\Phi_3(\boldsymbol{\lambda})$	$(\lambda_1^2 - 1)/\sqrt{2}$	$\sqrt{5}\,(\frac{3}{2}\lambda_1^2 - \frac{1}{2})$	2
$\Phi_4(\boldsymbol{\lambda})$	$\lambda_1 * \lambda_2$	$3 * \lambda_1 * \lambda_2$	2
$\Phi_5(\boldsymbol{\lambda})$	$(\lambda_2^2 - 1)/\sqrt{2}$	$\sqrt{5}\,(\frac{3}{2}\lambda_2^2 - \frac{1}{2})$	2
$\Phi_6(\boldsymbol{\lambda})$	$(\lambda^3 - 3\lambda)/\sqrt{6}$	$\sqrt{7} * (\frac{5}{2}\lambda_1^3 - \frac{3}{2}\lambda_1)$	3
$\Phi_7(\boldsymbol{\lambda})$	$\lambda_2 * (\lambda_1^2 - 1)/\sqrt{2}$	$\sqrt{15} * \lambda_2 * (\frac{3}{2}\lambda_1^2 - \frac{1}{2})$	3
$\Phi_8(\boldsymbol{\lambda})$	$\lambda_1 * (\lambda_2^2 - 1)/\sqrt{2}$	$\sqrt{15} * \lambda_1 * (\frac{3}{2}\lambda_2^2 - \frac{1}{2})$	3
$\Phi_9(\boldsymbol{\lambda})$	$(\lambda_2^3 - 3\lambda_2)/\sqrt{6}$	$\sqrt{7} * (\frac{5}{2}\lambda_2^3 - \frac{3}{2}\lambda_2)$	3

2.1.4.1 Arithmetic mean of random outputs

The first and most common statistical moment is the arithmetic mean since it indicates the central value which has random outputs spread around it. The definition of arithmetic mean for a random output is:

$$E(x(\boldsymbol{\lambda})) = \int_{\boldsymbol{\Omega}} x(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \tag{2.17}$$

where x is the random output, and the integral is multidimensional with n dimensions. By replacing (2.12) in (2.17) we would have:

$$E(x(\boldsymbol{\lambda})) = \sum_{i=0}^{P} \int_{\boldsymbol{\Omega}} c_i \phi_i(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = \sum_{i=0}^{P} \int_{\boldsymbol{\Omega}} c_i \phi_i(\boldsymbol{\lambda})\phi_0(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \tag{2.18}$$

It is worth noting $\phi_0(\lambda)$ is always equal to 1 for all orthonormal polynomials. By comparing the right hand side of (2.18) and (2.14), we can write:

$$E(x(\boldsymbol{\lambda})) = \sum_{i=0}^{P} \langle c_i \phi_i(\lambda), \phi_0(\lambda) \rangle = \sum_{i=0}^{P} c_i \delta_{i,0} = c_0 \tag{2.19}$$

Therefore, the arithmetic mean of a random output is coefficient of the first polynomial in its PC expansion disregard of the distribution and number of random variables.

2.1.4.2 Variance and standard deviation of random outputs

The second statistical moment is the variation. This parameter shows how far samples can get from the mean value; hence, a lower variation is desirable. The mathematical definition of variation is:

$$\begin{aligned} &Var(x(\boldsymbol{\lambda})) = E[(x(\boldsymbol{\lambda}) - E(x(\boldsymbol{\lambda})))^2] \\ &= \int_{\boldsymbol{\Omega}} (\sum_{i=0}^{P} c_i \phi_i(\boldsymbol{\lambda}) - c_0 \phi_0(\boldsymbol{\lambda}))^2 \rho(\boldsymbol{\lambda}) d\boldsymbol{\lambda} = \int_{\boldsymbol{\Omega}} (\sum_{i=1}^{P} c_i \phi_i(\boldsymbol{\lambda}))^2 \rho(\boldsymbol{\lambda}) d\boldsymbol{\lambda} \end{aligned} \tag{2.20}$$

Expansion of the right hand side of this equation results in all possible combinations of $\phi_i \phi_j$ which would convert the formula of variance to:

$$\text{var}(x(\boldsymbol{\lambda})) = \sum_{i=1}^{P} \sum_{j=1}^{P} \langle c_i \phi_i(\boldsymbol{\lambda}), c_j \phi_j(\boldsymbol{\lambda}) \rangle \tag{2.21}$$

However, only when $i = j$, the inner product results in a nonzero answer; therefore:

$$\text{var}(x(\boldsymbol{\lambda})) = \sum_{i=1}^{P} \langle c_i \phi_i(\boldsymbol{\lambda}), c_i \phi_i(\boldsymbol{\lambda}) \rangle = \sum_{i=1}^{P} c_i^2 \tag{2.22}$$

Or in other words summation of square of every coefficient except the first one.

The other popular statistical measurement tool is the standard deviation or σ which is the square root of variance; thus, variance is also shown as σ^2. The equation for standard deviation is:

$$\sigma = \sqrt{\sum_{i=1}^{P} c_i^2} \tag{2.23}$$

Standard deviation is favorable over the variance because it has the same order of mean; hence, it is comparable with the mean value.

2.1.4.3 Higher order moments and the probability distribution function

Higher order moments provide more information about the behavior of a random output. Although it is not very common to see these moments in the mainstream literature, they can be a noticeable help in special problems. In general statistical moments are defined as [57]:

$$\mu^M(x(\boldsymbol{\lambda})) = \int_{\Omega} (x(\boldsymbol{\lambda}) - E(x(\boldsymbol{\lambda})))^M \rho(\boldsymbol{\lambda}) d\boldsymbol{\lambda} \tag{2.24}$$

where M represents order of the moment and $\mu^M(x(\boldsymbol{\lambda}))$ is the M-th order moment. The third order moment is skewness and demonstrates asymmetry of the PDF. A high skewness means the PDF has high asymmetry. After simplification this parameter can be shown as:

$$s(x(\boldsymbol{\lambda})) = E(x(\boldsymbol{\lambda})^3) - 3\sigma(x(\boldsymbol{\lambda}))^2 - E(x(\boldsymbol{\lambda}))^3 \tag{2.25}$$

where $s(x(\boldsymbol{\lambda}))$ indicates skewness of $x(\boldsymbol{\lambda})$. The next statistical moment is called kurtosis and has an order of 4. This parameter demonstrates the form of tails of the PDF and after simplification can be shown as:

$$k(x(\boldsymbol{\lambda})) = E(x(\boldsymbol{\lambda})^4) - 4s(x(\boldsymbol{\lambda}))E(x(\boldsymbol{\lambda})) - 6\sigma(x(\boldsymbol{\lambda}))^2 E(x(\boldsymbol{\lambda}))^2 - E(x(\boldsymbol{\lambda}))^4 \tag{2.26}$$

where $k(x(\boldsymbol{\lambda}))$ indicates kurtosis of $x(\boldsymbol{\lambda})$. Unfortunately for statistical moments greater than two the inner product approach cannot be used because third degree polynomials with different random variables appear in the integral. However, knowledge of the coefficients in (2.12) allows us to find higher order moments in a different way. In order to do so, an approach similar to Monte Carlo is taken.

Statistical analysis with Monte Carlo is done by performing numerous experiments. Then statistical information of experiments' results are computed. However, this approach is not practical in the cases where one instance of the experiment takes a considerable time to perform. Nevertheless, a faster

method is generating the experiments' results by using (2.12). In this approach, first M random sample nodes are generated, where M is a great number. The number of dimensions and distribution of these samples is same as random variables $\boldsymbol{\lambda}$ in the system. By having coefficients, polynomials and random samples, the right hand side of (2.12) is known; therefore, the result of M instances of the experiment can be approximated. The final step is computing statistical information from these approximated results.

The final statistical information discussed in this section is the PDF which holds information from all previously introduced statistical moments, and in many cases is the most desirable statistical information. The approach to find PDF is same as higher order moments. M experiments' results are generated by sampling (2.12) and the PDF is constructed by drawing the normalized histogram of all that data. It is worth noting, using Monte Carlo to find higher order moments and PDF does not disvalue the PC approach because performing one single experiment often takes significantly more time comparing to calculation of (2.12).

2.2 Intrusive approaches

Intrusive approaches are the ones which require intrusive coding and development of a new circuit solver. In this section two intrusive approaches are discussed which are the Stochastic Galerkin and the stochastic testing approaches.

2.2.1 Stochastic Galerkin (SG) approach

The first PC based stochastic analysis technique that we discuss is the stochastic Galerkin approach [10]-[25], which is intrusive and requires intrusive coding and cannot be done in a black box manner. However, this effort makes the stochastic Galerkin to have a higher accuracy comparing to nonintrusive methods. In this approach, the equations governing the system are augmented with means of gPC theory. The resultant augmented equations are in fact equivalent to an expanded and coupled deterministic system. The PC coefficients can be derived after one single simulation of this system.

Before discussing this approach in details it is necessary to introduce one more PC based integration formula. This formula is explained in the next section.

2.2.1.1 Three term inner products

In previous sections we mentioned that we are interested in inner product of two polynomials because of the orthogonality condition. However, in the stochastic Galerkin approach a new expression, $\langle \phi_k(\lambda)\phi_j(\lambda), \phi_i(\lambda) \rangle$, appears, which is the inner product of multiplication of two polynomials and another polynomial. The result for Hermite polynomials would be [10]:

$$\langle \phi_k(\lambda)\phi_j(\lambda), \phi_i(\lambda) \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \phi_k(\lambda)\phi_j(\lambda)\phi_i(\lambda) e^{-\lambda^2/2} d\lambda = \frac{k!\, j!\, i!}{(m-k)!(m-j)!(m-i)!} \tag{2.27}$$

where $m = (k+j+i)/2$ is an integer and $m > k, i, j$. If m does not meet these two conditions the result is zero.

Furthermore, result of the abovementioned inner product for Legendre polynomials would be [10]:

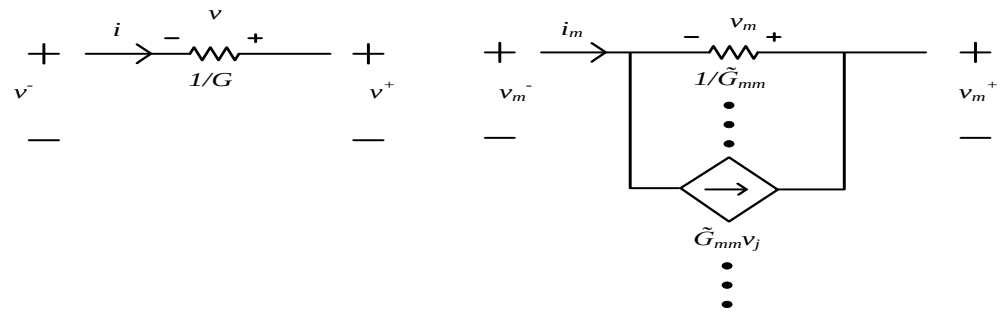
$$\begin{aligned} \langle \phi_k(\lambda)\phi_j(\lambda), \phi_i(\lambda) \rangle &= \frac{1}{2} \int_{-1}^{+1} \phi_k(\lambda)\phi_j(\lambda)\phi_i(\lambda) d\lambda \\ &= (-1)^{s-j} \frac{i!(2s-2i)!s!}{(s-k)!(s-j)!(s-1)!(2s+1)!} \sum_{t=p}^{q} (-1)^t \frac{(k+t)!(j+i-t)!}{t!(k-t)!(j-i+t)!(i-t)!} \end{aligned} \tag{2.28}$$

where s = (k+j+i)/2, p = max (0, i - j) and q = min (j+i, k, i). The above equation works only when s is even and values before factorial signs are non-negative. Otherwise the result is null.

2.2.1.2 Stochastic Galerkin approach for linear circuit elements

The SG approach is desirable in systems being governed by simple equations. Therefore, it can be easily applied to passive elements [18]. Consider the single resistor pictured in Fig. 2.2 (a). Assuming this resistor is affected by random variables $\boldsymbol{\lambda}$, the equation relating voltage and current for this element is:

$$i(t, \boldsymbol{\lambda}) = G(\boldsymbol{\lambda})(v_1(t, \boldsymbol{\lambda}) - v_2(t, \boldsymbol{\lambda})) = Gv(t, \boldsymbol{\lambda}) \tag{2.29}$$



(a) (b)

Fig. 2.2: Application of SG on a single resistor. (a) The resistor element. (b) The *m*-th spectral circuit.

where G is the conductivity. The first step is applying (2.12) to random functions in (2.29), which yields:

$$\sum_{i=0}^{P} i_k(t)\phi_i(\boldsymbol{\lambda}) = \sum_{k=0}^{P}\sum_{j=0}^{P} G_k v_j(t)\phi_k(\boldsymbol{\lambda})\phi_j(\boldsymbol{\lambda}). \tag{2.30}$$

Since random variable(s) are embedded in the resistor and $G(\boldsymbol{\lambda})$ is known, G_k in (2.30) can be found using the orthogonal projection technique:

$$\langle G(\boldsymbol{\lambda}), \phi_k(\boldsymbol{\lambda})\rangle = \int_{\boldsymbol{\Omega}} \left(\sum_{j=0}^{P} G_j \phi_j(\boldsymbol{\lambda})\right)\phi_k(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = \sum_{j=0}^{P}\int_{\boldsymbol{\Omega}} c_j \phi_j(\boldsymbol{\lambda})\phi_k(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = G_k \tag{2.31}$$

The next step is applying the Galerkin projection on (2.30) which means calculating inner product of left and right hand sides with the *m*-th polynomial.

$$\left\langle \sum_{i=0}^{P} i_k(t)\phi_i(\boldsymbol{\lambda}), \phi_m(\boldsymbol{\lambda})\right\rangle = \left\langle \sum_{k=0}^{P}\sum_{j=0}^{P} G_k v_j(t)\phi_k(\boldsymbol{\lambda})\phi_j(\boldsymbol{\lambda}), \phi_m(\boldsymbol{\lambda})\right\rangle. \tag{2.32}$$

Considering the orthogonality condition we would have:

$$i_m(t) = \sum_{j=0}^{P} \tilde{G}_{mj} v_j(t) \tag{2.33}$$

where $\tilde{\mathbf{G}}$ is the augmented conductance matrix:

$$\tilde{\tilde{G}}_{mj} = \sum_{j=0}^{P} G_k \alpha_{kjm} \tag{2.34}$$

And:

$$\alpha_{kjm} = \left\langle \phi_k(\boldsymbol{\lambda})\phi_j(\boldsymbol{\lambda}), \phi_m(\boldsymbol{\lambda}) \right\rangle = \int_{\boldsymbol{\Omega}} \phi_k(\boldsymbol{\lambda})\phi_j(\boldsymbol{\lambda})\phi_m(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda}. \tag{2.35}$$

After applying (2.32) to all $\phi_m(\boldsymbol{\lambda})$ polynomials, P+1 similar equations would be generated which can be organized as:

$$\tilde{\tilde{\mathbf{i}}}(t) = \tilde{\tilde{\mathbf{G}}}\tilde{\tilde{\mathbf{v}}}(t) \tag{2.36}$$

where $\tilde{\tilde{\mathbf{i}}}(t) = [i_0(t), \ldots i_P(t)]^T$, $\tilde{\tilde{\mathbf{v}}}(t) = [v_0(t), \ldots v_P(t)]^T$, $\tilde{\tilde{\mathbf{G}}}$ denotes the augmented conductance matrix described in (2.34), and T is the transpose sign. It is worth noting (2.36) represents a deterministic circuit which can be simulated by circuit simulator software. In this augmented circuit, there are P+1 branches and as depicted in Fig. 2.2 (b) each of them possesses a resistor equal to $1/\tilde{G}_{mm}$. Each branch is coupled to other branches with a factor of $\tilde{G}_{mj}$ modeled by parallel voltage dependent current sources. Afterwards, coefficients of output voltages or currents can be easily obtained by probing the augmented circuit. In other words, V^+_j is equal to the voltage at the end of the j-th branch.

This process can be done likewise for stochastic capacitors and inductors, where the PC expansion, Galerkin projection and augmentation are done on $i(t,\boldsymbol{\lambda}) = C(\boldsymbol{\lambda})dv(t,\boldsymbol{\lambda})/dt$ and $v(t,\boldsymbol{\lambda}) = L(\boldsymbol{\lambda})di(t,\boldsymbol{\lambda})/dt$ respectively. A general circuit usually involves a lot of deterministic elements as well. By writing the expansion it can be easily proven these elements appear as themselves in every branch of the augmented circuit. Finally, independent deterministic sources appear as themselves in the first branch and with zero amplitude in other branches.

In the case of distributed networks, same principals apply. Consider a set of coupled transmission lines. They can be modeled by partial differential equations [10], and the PC expansion and Galrekin

projection can be applied to these equations similar to passive elements [10]. Moreover, the coupled equations describing the augmented system can be generated from Galerkin projection of expanded version of the partial differential equations and the $\phi_m(\boldsymbol{\lambda})$ basis. The result represents a deterministic augmented circuit which can be simulated by circuit simulator software. In this augmented circuit, there are P+1 times more transmission lines and all of them are coupled to others. Afterwards, coefficients of output voltages or currents can be easily obtained by probing the augmented circuit.

2.2.1.3 Stochastic Galerkin approach for nonlinear circuit elements

One major challenge in the SG approach is determination of integrals similar to the one in (2.32) which are necessary for finding coefficient in PC expansion of equations describing network's elements. In case of passive elements and even transmission lines this is a relatively easy task. However, it is different for nonlinear elements like diodes which are governed by a nonlinear function as in $i(t) = F(v(t))$ because after the Galerkin projection the result would be:

$$i_m(t) = \int_{\boldsymbol{\Omega}} F\left(\sum_{k=0}^{P} v_k(t)\phi_k(\boldsymbol{\lambda})\right)\phi_m(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \tag{2.37}$$

Therefore the work of [18] suggests a numerical approach to solve this integral by generating $Q = (m+1)^n$ companion cells where each cell includes the nonlinear element and a dependent voltage source. And the nonlinear element is replaced by a dependent current source in the m-th spectral branch of the main augmented circuit representing i_m.

This process can be done likewise for three terminal nonlinear elements such as transistors by modeling them as Q companion cells, where each cell includes the transistor and two dependent voltage sources, in addition to m spectral networks with two dependent current sources in each of them.

2.2.1.4 Stochastic Galerkin approach for a general network

Consider the general nonlinear network consisting of distributed and lumped circuit elements characterized by the modified nodal analysis (MNA) equations. After introducing the random variables $\boldsymbol{\lambda}$ to the network the stochastic version of the MNA equation would be [19], [25], [61]:

$$\mathbf{G}(\boldsymbol{\lambda})\mathbf{X}(t,\boldsymbol{\lambda})+\mathbf{C}(\boldsymbol{\lambda})\frac{d\mathbf{X}(t,\boldsymbol{\lambda})}{dt}+\mathbf{F}(\mathbf{X}(t,\boldsymbol{\lambda}))+\sum_{i=1}^{N_t}(\mathbf{T}_i\mathbf{Y}_i(t,\boldsymbol{\lambda})\mathbf{T}_i^T)*\mathbf{X}(t,\boldsymbol{\lambda})=\mathbf{B}(t) \tag{2.38}$$

where **G**, **C** matrices contain the stamp of all the memoryless and memory lumped circuit elements respectively, **X** is the vector of stochastic voltage/current responses, **F** contains the stamp of nonlinear circuit elements, $\mathbf{T}_i$ is the selector matrix mapping the vector of port currents $\mathbf{i}_i(t)$ for the *i*-th distributed network into the nodal space of the circuit, $\mathbf{Y}_i$ is the corresponding time-domain ***Y***-parameter macromodel of the *i*-th distributed network, **B** represents the input vector of independent voltage and current sources and '$*$' denotes the temporal convolution which is performed in a recursive manner in SPICE.

The intrusive SG approach begins by expanding the stochastic processes of (2.38) using multivariate orthogonal polynomial bases as [19]:

$$\begin{aligned}&\mathbf{G}(\boldsymbol{\lambda})=\sum_{k=0}^{P}\mathbf{G}_k\phi_k(\boldsymbol{\lambda}),\quad \mathbf{C}(\boldsymbol{\lambda})=\sum_{k=0}^{P}\mathbf{C}_k\phi_k(\boldsymbol{\lambda}),\\&\mathbf{X}(t,\boldsymbol{\lambda})=\sum_{k=0}^{P}\mathbf{X}_k(t)\,\phi_k(\boldsymbol{\lambda}),\ \mathbf{Y}_i(t,\boldsymbol{\lambda})=\sum_{k=0}^{P}\mathbf{Y}_{ik}(t)\,\phi_k(\boldsymbol{\lambda})\end{aligned} \tag{2.39}$$

Then the matrices $\mathbf{G}_k$, $\mathbf{C}_k$ and $\mathbf{Y}_k$ of (2.39) are obtained from the explicit knowledge of $\mathbf{G}(\boldsymbol{\lambda})$, $\mathbf{C}(\boldsymbol{\lambda})$ and $\mathbf{Y}(\boldsymbol{\lambda})$ respectively. The expansion of (2.39) is then replaced in (2.38) and a Galerkin projection of both sides of the equation is performed, thereby converting the stochastic equations of (2.38) into a set of augmented deterministic coupled equations as [19]:

$$\mathbf{G}_a\mathbf{X}_a(t)+\mathbf{C}_a\frac{d\mathbf{X}_a(t)}{dt}+\mathbf{F}_a(\mathbf{X}_a(t))+\sum_{i=1}^{N_t}(\mathbf{T}_{ia}\mathbf{Y}_{ia}(t)\mathbf{T}_{ia}{}^T)*\mathbf{X}_a(t)=\mathbf{B}_a(t) \tag{2.40}$$

where $\mathbf{G}_a$, $\mathbf{C}_a$ are augmented matrices constructed using the $\mathbf{G}_k$, $\mathbf{C}_k$ block matrices respectively, $\mathbf{T}_{ia}$ is a purely diagonal matrix composed of $\mathbf{T}_i$, $\mathbf{Y}_{ia}(t)$ is the augmented time-domain macromodel of the i-th distributed network constructed from the $\mathbf{Y}_{ik}$ block matrices of (2.39), $\mathbf{B}_a = [\mathbf{B}, \mathbf{0},\ldots, \mathbf{0}]^T$, $\mathbf{X}_a = [\mathbf{X}_0, \mathbf{X}_1 ,\ldots, \mathbf{X}_P]^T$ and $\mathbf{F}_a(.)$ denotes the augmented vector of nonlinear circuit elements.

The overall MNA equations of (2.40) represent an augmented deterministic network which can be solved within a SPICE environment. Once the PC coefficients $\mathbf{X}_k(t)$ are known, the statistical moments of the system can be easily evaluated via the PC expansion of (2.39).

2.2.1.5 Pros and cons of the stochastic Galerkin approach

With no doubt the described SG approach provides a high level of accuracy [10]-[25]. One reason is that unlike nonintrusive approaches the distribution of output is not assumed to be the joint PDF of input random variables $\boldsymbol{\lambda}$. The higher accuracy allows SG to choose lower orders of expansion which in turn saves computational costs. Moreover, SG is able to extract all PC coefficients from one single experiment or simulation, while nonintrusive and sampling approaches perform a great number of calls to the simulation software.

However the SG approach requires intrusive coding to automate the conversion of the stochastic equations describing the network into the augmented deterministic network. Thus, this approach cannot directly exploit existing sophisticated deterministic solvers such as SPICE. Moreover, the augmented deterministic equations denote the augmentation of the network by a factor of P+1. This translates to a near-exponential scaling of the CPU time and memory costs with respect to the number of random dimensions. Finally, the multidimensional integrals of nonlinear equations are computed using numerical techniques where the terms of the numerical approximation are represented in SPICE using lumped dependent sources, thus leading to further augmentation of the network [18]. Due to these factors, the application of the SG approach is only limited to problems involving small-dimensional random spaces [10]-[25].

2.2.2 The intrusive stochastic testing (ST) approach

In order to address the inefficiencies of the SG technique recently the intrusive stochastic testing approach which is able to extract PC coefficients by exploiting an intrusive solver is suggested in [40]. For simplicity in discussion of the ST approach, instead of (2.38) the following differential algebraic equation (DAE) is considered:

$$\frac{d\mathbf{q}(\mathbf{x}(t,\boldsymbol{\lambda}),\boldsymbol{\lambda})}{dt}+\mathbf{f}(\mathbf{x}(t,\boldsymbol{\lambda}),\boldsymbol{\lambda})=B\mathbf{u}(t) \tag{2.41}$$

where **u**(t) denotes the input signal, **x** represents voltages and currents in the network and **q** and **f** are stamps of memory and memoryless elements respectively. Similar to other PC approaches solution of the network is approximated as:

$$\hat{\mathbf{x}}(t,\boldsymbol{\lambda})=\sum_{i=0}^{P}\hat{\mathbf{x}}_i(t)\phi_i(\boldsymbol{\lambda}) \tag{2.42}$$

Replacing (2.42) in (2.41) yields the following residual function:

$$RES(\mathbf{X}(t,\boldsymbol{\lambda}))=(\frac{d\mathbf{q}(\hat{\mathbf{x}}(t,\boldsymbol{\lambda}),\boldsymbol{\lambda})}{dt}+\mathbf{f}(\hat{\mathbf{x}}(t,\boldsymbol{\lambda}),\boldsymbol{\lambda})-B\mathbf{u}(t) \tag{2.43}$$

where $\mathbf{X}(t,\boldsymbol{\lambda})=[\hat{x}_1(t),....\hat{x}_P(t)]$. The goal of ST is developing an intrusive solver to solve (2.43) and extract PC coefficients. In the following sections basics for developing ST's intrusive solver, selection of sample nodes required for ST analysis, an alternative nonintrusive ST approach, and finally a discussion on pros and cons of the approach are provided.

2.2.2.1 Basics of development of an intrusive solver for the ST approach

The basic idea behind the ST approach starts with calculating (2.43) for *P*+1 testing nodes denoted as $\boldsymbol{\lambda}^{(1)},\ldots.\ \boldsymbol{\lambda}^{(P+1)}$, and enforcing the residue to be zero:

$$\frac{d\mathbf{Q}(\mathbf{X}(t))}{dt}+\mathbf{F}(\mathbf{X}(t))=\tilde{\mathbf{B}}\mathbf{u}(t) \tag{2.44}$$

where:

$$\mathbf{Q}(\mathbf{X}(t))=\begin{bmatrix}\mathbf{q}\left(\hat{\mathbf{x}}(t,\boldsymbol{\lambda}^{(1)}),\boldsymbol{\lambda}^{(1)}\right)\\ .\\ .\\ .\\ \mathbf{q}\left(\hat{\mathbf{x}}(t,\boldsymbol{\lambda}^{(P+1)}),\boldsymbol{\lambda}^{(P+1)}\right)\end{bmatrix},\mathbf{F}(\mathbf{X}(t))=\begin{bmatrix}\mathbf{f}\left(\hat{\mathbf{x}}(t,\boldsymbol{\lambda}^{(1)}),\boldsymbol{\lambda}^{(1)}\right)\\ .\\ .\\ .\\ \mathbf{f}\left(\hat{\mathbf{x}}(t,\boldsymbol{\lambda}^{(P+1)}),\boldsymbol{\lambda}^{(P+1)}\right)\end{bmatrix},\widetilde{\mathbf{B}}=\begin{bmatrix}\mathbf{B}\\ .\\ .\\ .\\ \mathbf{B}\end{bmatrix} \tag{2.45}$$

The ST approach proposed in [40], solves (2.45) by developing an intrusive transient solver where adaptive time stepping and decoupling can be used. After solving (2.45), this approach directly provides PC coefficients. This solver can be implemented by using different integration schemes such as backward Euler, trapezoidal and Gear-2 techniques. Next, without any loss of generality, backward Euler is selected to demonstrate application of the ST approach; however, it can be applied to other integration schemes likewise.

Using the backward Euler integration scheme discretization of (2.44) yields in [1]

$$\mathbf{R}(\mathbf{X}_k)=\alpha_k\left(\mathbf{Q}(\mathbf{X}_k)-\mathbf{Q}(\mathbf{X}_{k-1})\right)+\mathbf{F}(\mathbf{X}_k)-\widetilde{\mathbf{B}}\mathbf{u}_k=0 \tag{2.46}$$

where:

$$\mathbf{X}_k=\mathbf{X}(t_k),\ \ \mathbf{u}_k=\mathbf{u}(t_k),\ \ \alpha_k=\frac{1}{t_k-t_{k-1}} \tag{2.47}$$

It is worth noting in this approach the local truncation error (LTE) is exploited to adaptively set the timestep. Moreover, the Newton's iterative technique is used to determine X_k in (2.47) by starting from random guess of $\mathbf{X}_k^{\,0}$. In other words the equation

$$J(\mathbf{X}_k^j)\Delta\mathbf{X}_k^j=-\mathbf{R}(\mathbf{X}_k^j), \tag{2.48}$$

where $J(\mathbf{X}_k^j)$ is the Jacobian matrix of $\mathbf{R}(\mathbf{X}_k^j)$, is solved. Then the $\mathbf{X}_k^j$ is updated:

$$\mathbf{X}_k^{j+1}=\mathbf{X}_k^j+\Delta\mathbf{X}_k^j \tag{2.49}$$

until the solution gets converged. Equation (2.48) may be solved for $\Delta \mathbf{X}_k^j$ using a matrix solver which would be costly. Therefore, [40] suggests an iterative and decoupled technique. Interested readers are encouraged to read [40] for extra details. Once the PC coefficients are extracted, the statistical moments can be generated same to previously mentioned PC approaches. Next the algorithm to find the testing node in (2.44) is discussed.

2.2.2.2 Determination of testing nodes for the ST approach

One major challenge in the ST approach is determination of testing nodes since poor selection of nodes results in ill-conditioned matrices in (2.45) which in turns makes the solution inaccurate or impossible to obtain. In order to address this issue, the ST approach of [40] starts with the $(m+1)^n$ nodes inspired by the Wiener-Askey scheme. These nodes are set to be the tensor product of roots of the $(m+1)$-th polynomial orthogonal to the joint distribution of input random variables. The ST approach tries to obtain PC coefficients from the least possible number of samples which is equal to number of coefficients i.e. $P+1$. In order to have accurate results two criteria are set for selecting nodes. The first one is giving the preference to quadrature nodes with a higher corresponding weight since they are statistically more important. The second criterion is based on the following **A** matrix generated from the selected nodes $\lambda^{(1)}$, $\lambda^{(2)}, \ldots, \lambda^{(P+1)}$:

$$\mathbf{A} = \begin{bmatrix} \phi_0(\boldsymbol{\lambda}^{(1)}) & \cdots & \phi_P(\boldsymbol{\lambda}^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^{(P+1)}) & \cdots & \phi_P(\boldsymbol{\lambda}^{(P+1)}) \end{bmatrix} \tag{2.50}$$

This criterion states the final **A** matrix needs to be full rank and well-conditioned. The algorithm for finding testing nodes starts with constructing the $(m+1)^n$ quadrature nodes and their corresponding weights. Then nodes are sorted from the highest weight to the lowest, and the first testing node, $\boldsymbol{\lambda}^{(1)}$ is selected as the one with the highest weight. In order to generate the remaining P nodes the rest of the quadrature nodes are considered in the same order as their weight. Assuming r-1 nodes are selected at step r the following vector space is constructed:

$$\mathbf{V} = \left\{\mathbf{H}(\boldsymbol{\lambda}^{(1)}),...\mathbf{H}(\boldsymbol{\lambda}^{(r-1)})\right\} \tag{2.51}$$

where $\mathbf{H}(\boldsymbol{\lambda}) = [\phi_0(\boldsymbol{\lambda}), \phi_1(\boldsymbol{\lambda}), .. \phi_P(\boldsymbol{\lambda})]^T$. Next, from the remaining $(m+1)^n - (r-1)$ nodes, the node with the highest weight, $\boldsymbol{\lambda}^{(i)}$, is tested to see if the $\mathbf{H}(\boldsymbol{\lambda}^{(i)})$ vector has large enough orthogonal component to **V**. If this condition is met $\boldsymbol{\lambda}^{(i)}$ is added to selected nodes and the algorithm moves to step *r*+1, otherwise other remaining nodes in the quadrature space are tested. Once *P*+1 nodes are obtained the algorithm stops.

2.2.2.3 An alternative nonintrusive ST approach

The ST approach introduced in [40] provides noticeable advances in accuracy comparing to nonintrusive approaches and in efficiency comparing to the intrusive SG approach. However, the intrusive nature impedes its application on many sophisticated problems. To address this issue the work of [37] suggests a nonintrusive approach. The nonintrusive stochastic testing asks for doing the deterministic simulation of the system at only *P*+1 nodes and arranges the result as:

$$\begin{bmatrix} \phi_0(\boldsymbol{\lambda}^{(1)})\mathbf{I} & \dots & \phi_P(\boldsymbol{\lambda}^{(1)})\mathbf{I} \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^{(P+1)})\mathbf{I} & \dots & \phi_P(\boldsymbol{\lambda}^{(P+1)})\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \vdots \\ \mathbf{X}_P \end{bmatrix} = \begin{bmatrix} \mathbf{X}(\boldsymbol{\lambda}^{(1)}) \\ \vdots \\ \mathbf{X}(\boldsymbol{\lambda}^{(P+1)}) \end{bmatrix} \tag{2.52}$$

where **I** is the identity matrix. Since the matrix **A** is square, the solution can be found as $\tilde{\mathbf{X}}=\mathbf{A}^{-1}\mathbf{E}$, with $\tilde{\mathbf{X}}=[\mathbf{X}_0,\dots\ \mathbf{X}_P]^T$ and $\mathbf{E}=[\mathbf{X}(\boldsymbol{\lambda}^{(1)}),\dots\ \mathbf{X}(\boldsymbol{\lambda}^{(P+1)})]^T$.

The arbitrary selection of *P*+1 nodes is not guaranteed to yield accurate results; therefore, the main contribution of [37] is selection of the *P*+1 nodes in (2.52). In order to do this task the nonintrusive stochastic testing takes advantage of the node selection algorithm described in section 2.2.2.2, claiming the ST's node selection algorithm is strong enough to yield accurate results.

2.2.2.4 Pros and cons of the stochastic testing approach

The ST approach suggests well designed technique to find the PC coefficients. Comparing to the SG method, the modified network which ST needs to solve is less complicated and also according to [40]

it can be decoupled; therefore, it is more efficient. This approach also provides a high accuracy comparing to nonintrusive approaches and although it has a higher CPU cost comparing to some PC approaches, it presents the potential to find the coefficients faster than some nonintrusive approaches which have an exponential rate with respect to number of random variables since the ST approach presents a polynomials scaling rate.

On the other hand the original ST approach is intrusive after all, and intrusive coding is not practical in all cases. This issue impedes ST's application for sophisticated networks which can only be simulated by commercial software with no open source code. Although, the nonintrusive version of ST is very efficient comparing to other PC approaches, and tries to address the issue of being intrusive, it is proven in [61] that results obtained from (2.52) may not be accurate enough because the ST's criteria for selection of the nodes is not optimal.

2.3 Nonintrusive approaches

Nonintrusive approaches are the ones which don't require intrusive coding or development of a new solver. These approaches take advantage of commercial software to find the result at a set of sample nodes. In this section four nonintrusive approaches are discussed which are the pseudo spectral PC, conventional linear regression, stochastic collocation and Stroud cubature based stochastic collocation approaches.

2.3.1 Pseudo spectral PC approach

The nonintrusive pseudo spectral approach seems to be the most straightforward among PC methods. Being nonintrusive, this approach does not need any simulation tool development, modification of the circuit or even knowledge of internal equations governing the circuit. In this approach the PC expansion of (2.12) is directly applied to the outputs of the system and by means of numerical integration methods PC coefficients are found. Then, these coefficients are used to find statistical information. In the

following sections first we introduce the used numerical integration technique which is integration with Gaussian quadrature rules and then mathematical details of this approach are discussed.

2.3.1.1 Integration with Gaussian quadrature rules

Gaussian quadrature rules are numerical integration tools used to approximate the integral of a certain function $f(\lambda)$ with distribution $\rho(\lambda)$ [58], [10]. The approximation is in form of a weighted summation of function $f(\lambda)$ at predetermined sample nodes. This can be written as:

$$\int_{\boldsymbol{\Omega}} f(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \approx \sum_{k=1}^{Q} f(\boldsymbol{\lambda}^k)\omega(\boldsymbol{\lambda}^k) \tag{2.53}$$

where Q is number of the sample nodes, $\boldsymbol{\lambda}^k = [\lambda_1^{(k)}, \lambda_2^{(k)}, \dots, \lambda_n^{(k)}]$ is the k-th discrete node in the random space, $\omega(\boldsymbol{\lambda}^k)$ is the corresponding weight to node $\boldsymbol{\lambda}^k$ and $f(\boldsymbol{\lambda}^k)$ is the value of function $f(\boldsymbol{\lambda})$ at node $\boldsymbol{\lambda}^k$.

Order of a quadrature rule is associated with number of nodes and is noted as q. In the one dimensional case Q is equal to $(q+1)$, and inspired by the Wiener-Askey scheme sample nodes are set to be the roots of the $(q+1)$-th polynomial orthogonal to the $\rho(\lambda)$ distribution. In problems with n dimensions $Q = (q+1)^n$ and sample nodes are obtained by generating tensor product of the one dimensional nodes. Finally, $\omega(\boldsymbol{\lambda}^k)$ is equal to product of the weights of one dimensional nodes λ_i^k which depends on the distribution of λ.

For the normal distribution, nodes are generated from roots of Hermite polynomials. Another way to generate nodes which also yields corresponding weights is solving an eigenvalue problem which is known as Golub-Welsch algorithm [59], [10]. In order to do so, consider the matrix $\mathbf{A}_{(q+1)x(q+1)}$ with the following entries:

$$\mathbf{A}(i,j) = \begin{cases} \sqrt{i} & j = i-1 \\ \sqrt{j} & i = j-1 \\ 0 & otherwise \end{cases} \tag{2.54}$$

nodes are equal to eigenvalues of **A** and corresponding weights are squares of first element of each eigenvector in the same order.

For the uniform distribution nodes are generated from roots of the Legendre polynomials. Corresponding weight to λ^k is obtained from the following formula [60]:

$$\omega^{(k)} = \frac{1}{\left(1-\left(\lambda^{(k)}\right)^2\right)\left(\frac{d\phi_{q+1}}{d\lambda}(\lambda^{(k)})\right)^2} \tag{2.55}$$

where ϕ_{q+1} is the (q+1)-th Legendre polynomial.

In the general case, parameters α_i and β_i of (2.12) can be used in the Golub-Welsh algorithm to compute the quadrature nodes and weights [59]. This algorithm requires first constructing the symmetric tridiagonal matrix **A** where

$$\mathbf{A}(i,j) = \begin{cases} \alpha_{i-1} & \text{if } i = j \\ \sqrt{\beta_i} & \text{if } j = i+1 \\ \sqrt{\beta_j} & \text{if } j = i-1 \\ 0 & \text{otherwise} \end{cases}; \qquad 1 \le i, j \le p+1 \tag{2.56}$$

Let $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ be the eigenvalue decomposition of **A** where **U** is the unitary matrix. This makes $\mathbf{\Lambda}(i,i)$ the i^{th} Gaussian quadrature node and $(\mathbf{U}(1,i))^2$ the corresponding weight.

2.3.1.2 Application of Gaussian quadrature rules to the pseudo spectral PC approach

As stated before, the pseudo spectral PC approach takes advantage of numerical integration to find the PC coefficients. Since polynomials used in this approach are orthonormal, coefficients can be found using the orthogonal projection technique, or in other words calculation of inner product of a function and a polynomial:

$$\left\langle X(\boldsymbol{\lambda}), \phi_i(\boldsymbol{\lambda}) \right\rangle = \int_{\boldsymbol{\Omega}} X(\boldsymbol{\lambda})\phi_i(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \tag{2.57}$$

Replacing (2.12) in (2.57) yields:

$$\left\langle X(\boldsymbol{\lambda}),\phi_i(\boldsymbol{\lambda})\right\rangle = \int_{\boldsymbol{\Omega}}\left(\sum_{j=0}^{P} c_j\phi_j(\boldsymbol{\lambda})\right)\phi_i(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = \sum_{j=0}^{P}\int_{\boldsymbol{\Omega}} c_j\phi_j(\boldsymbol{\lambda})\phi_i(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = c_i \tag{2.58}$$

Therefore, the only remaining step is numerical approximation of the integral in (2.57), which is:

$$c_i = \int_{\boldsymbol{\Omega}} X(\boldsymbol{\lambda})\phi_i(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \approx \sum_{k=1}^{Q} X(\boldsymbol{\lambda}^{(k)})\phi_i(\boldsymbol{\lambda}^{(k)})\rho(\boldsymbol{\lambda}^{(k)})\omega(\boldsymbol{\lambda}^{(k)}) \tag{2.59}$$

where $X(\boldsymbol{\lambda}^{(k)})$ are provided by performing Q experiments and the rest is done analytically.

2.3.1.3 Pros and cons of the PC pseudo spectral approach

Being nonintrusive the pseudo spectral approach is advantageous since it can be done by using commercial software. Moreover, for a moderate number of random variables it would be much faster than Monte Carlo. Comparing to other nonintrusive PC approaches, pseudo spectral demonstrates a higher accuracy because it exploits a higher number of deterministic samples.

On the other hand the scaling rate of the CPU cost is a major challenge in the pseudo spectral approach since it increases exponentially with respect to number of random variables; therefore, for problems with a higher number of random variables, which can be addressed easily by other PC methods, the pseudo spectral approach might even need more samples comparing to Monte Carlo. The exponential rate increases CPU cost so drastic that the use of intrusive techniques might be both more efficient and accurate comparing to the pseudo spectral approach.

2.3.2 Conventional linear regression approach

Another nonintrusive approach is the conventional linear regression [27], [28]. This approach takes advantage of the linear least squares technique to determine the best fit for PC coefficients. In the next section a brief review of the linear least squares method is done and then its application in the PC theory is discussed.

2.3.2.1 Linear least squares technique

The linear least squares is a common technique in statistics to fit a model to the gathered data [62]. As an example Fig. 2.3 pictures a set of data, blue dots, gathered from M experiments or simulation. It is desirable to find a model such as the red curve which represents the experiment's result, y, at an arbitrary x. In the general case, the equation describing the result for the i-th sample, $x^{(i)}$, can be written as:

$$\sum_{j=1}^{N} c_j F_j(x^{(i)}) = y^{(i)} \tag{2.60}$$

where $1 \le i \le M$, $y^{(i)}$ represents the result of experiment or simulation at the i-th sample $x^{(i)}$, $F_j(x^{(i)})$ is a function of x, $N < M$, $F_1(x^{(i)})=1$, and c_j are unknown coefficients. After doing this for M samples, the M linear equations can be written as:

$$\mathbf{Ac} = \mathbf{y} \tag{2.61}$$

where:

$$\mathbf{A} = \begin{bmatrix} F_0(x^{(1)}) & \dots & F_N(x^{(1)}) \\ \vdots & \ddots & \vdots \\ F_0(x^{(M)}) & \dots & F_N(x^{(M)}) \end{bmatrix}; \mathbf{c} = \begin{bmatrix} c_0 \\ \vdots \\ c_N \end{bmatrix}; \mathbf{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(M)} \end{bmatrix} \tag{2.62}$$

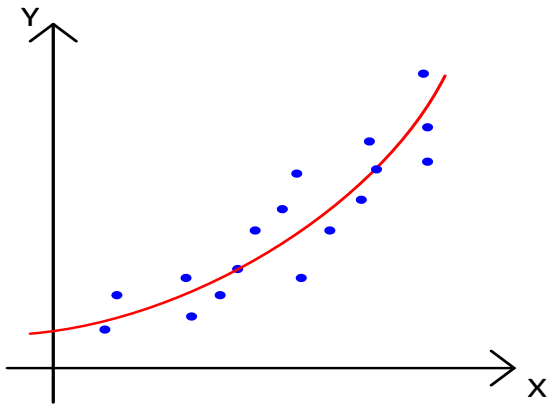


Fig. 2.3: Demonstration of fitting a model to stochastic data using the linear least square technique.

Although F_j does not have to be linear, this technique is called linear least square because (2.61) is a linear combination of F_j functions. Moreover, since $N < M$ the system is overdetermined and usually does not have an answer. Therefore, the linear least square technique tries to find coefficients c_j in the best way to fit the equations and minimize sum of squares of errors:

$$\tilde{\mathbf{c}} = \arg_{\mathbf{c}} \min S(c) \tag{2.63}$$

where:

$$S(c) = \| \mathbf{y} - \mathbf{A}\mathbf{c} \|^2 = \sum_{i=1}^{M} r_i^2 \qquad , \qquad r_i = | y^{(i)} - \sum_{j=1}^{N} c_j F_j(x^{(i)}) | \tag{2.64}$$

Equation (2.64) is minimum when its gradient is zero because it is a convex function. Therefore the following equation needs to be solved:

$$\frac{\partial S}{\partial c_j} = 2\sum_{i=1}^{M} r_i \frac{\partial r_i}{\partial c_j} = 2\sum_{i=1}^{M} \left(y_i - \sum_{k=1}^{N} \tilde{c}_k F_k(x^{(i)}) \right) \left(- F_j(x^{(i)}) \right) = 0 \tag{2.65}$$

The simplified form of (2.65) would be:

$$\sum_{i=1}^{M} \sum_{k=1}^{N} \tilde{c}_k F_k(x^{(i)}) F_j(x^{(i)}) = \sum_{i=1}^{M} F_j(x^{(i)}) y_i \tag{2.66}$$

which in the matrix form is:

$$\left(\mathbf{A}^T \mathbf{A} \right) \tilde{\mathbf{c}} = \mathbf{A}^T \mathbf{y} \tag{2.67}$$

Equation (2.67) has a solution when $\mathbf{X}^T$ is full column rank which makes $\mathbf{X}^T\mathbf{X}$ positive definite. This solution is:

$$\tilde{\mathbf{c}} = \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T \mathbf{y} \tag{2.68}$$

and yields the coefficient vector in (2.61).

2.3.2.2 Application of least squares technique to the PC theory

The conventional linear regression approach [27], [28], [46] begins by approximating the uncertainty in the network response of (2.38) using a PC expansion as

$$\mathbf{X}(t,\boldsymbol{\lambda}) = \sum_{j=0}^{P} \mathbf{X}_j(t)\,\phi_j(\boldsymbol{\lambda}) \tag{2.69}$$

As stated before the goal of a stochastic analysis approach is to find coefficients $\mathbf{X}_j(t)$; nevertheless, the polynomials ϕ_j are known beforehand and their value can be computed at any sample node $\boldsymbol{\lambda}^{(i)}$. Therefore, we can write down $\phi_0(\boldsymbol{\lambda}^{(i)})$ to $\phi_P(\boldsymbol{\lambda}^{(i)})$ in a certain vector $\mathbf{A}_i$ as:

$$\mathbf{A}_i = [\phi_0(\boldsymbol{\lambda}^{(i)})\mathbf{I}, \phi_1(\boldsymbol{\lambda}^{(i)})\mathbf{I}, \ldots \phi_P(\boldsymbol{\lambda}^{(i)})\mathbf{I}] \tag{2.70}$$

where $\mathbf{I}$ represents the identity matrix and is added to support multiple responses. Using (2.60), the summation in (2.69) can be expressed as:

$$\mathbf{A}_i \tilde{\mathbf{X}} = \mathbf{X}(t,\boldsymbol{\lambda}^{(i)}) \tag{2.71}$$

where $\tilde{\mathbf{X}} = [\mathbf{X}_0(t), \mathbf{X}_1(t), \ldots \mathbf{X}_P(t)]^T$, and $\mathbf{X}(t,\boldsymbol{\lambda}^{(i)})$ is the simulation result at $\boldsymbol{\lambda}^{(i)}$. The conventional linear regression approach suggests doing (2.71) at $M = 2(P+1)$ or $M = 3(P+1)$ nodes located within the random space Ω_n resulting in the formulation of an overdetermined system of linear algebraic equations:

$$\mathbf{A}\tilde{\mathbf{X}} = \mathbf{E} \tag{2.72}$$

where

$$\mathbf{A} = \begin{bmatrix} \phi_0(\boldsymbol{\lambda}^{(1)})\mathbf{I} & \ldots & \phi_P(\boldsymbol{\lambda}^{(1)})\mathbf{I} \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^{(M)})\mathbf{I} & \ldots & \phi_P(\boldsymbol{\lambda}^{(M)})\mathbf{I} \end{bmatrix}; \mathbf{E} = \begin{bmatrix} \mathbf{X}(t,\boldsymbol{\lambda}^{(1)}) \\ \vdots \\ \mathbf{X}(t,\boldsymbol{\lambda}^{(M)}) \end{bmatrix} \tag{2.73}$$

The vector $\mathbf{E}$ consists of the network responses obtained by probing the original stochastic network at the M multidimensional nodes $\boldsymbol{\lambda}^{(k)} = [\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots, \lambda_n^{(k)}]^T$; $1 \le k \le M$. Since the minimum number of rows in

(2.72) to have a unique answer is (P+1), this method is called oversampling. It is worth noting that (2.72) is similar to (2.61) and it can now be solved in a least-square sense to evaluate the PC coefficients of the network response:

$$\tilde{\mathbf{X}} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{E} \tag{2.74}$$

Once the PC coefficients are obtained from (2.74), all statistical moments of the network responses can be obtained from the PC expansion.

2.3.2.3 Pros and cons of the linear regression approach

As a nonintrusive method one benefit of the conventional linear regression approach is that sophisticated deterministic solvers such as SPICE can be directly used to populate the matrix **E** in (2.72) without any intrusive coding as required in the SG approach as well as the stochastic testing approach. More importantly, the M simulations of (2.72) can be easily parallelized unlike the conventional SG approach. It is appreciated that the matrix **A** is time independent and the nodes and their evaluation needs to be found only once and thereafter can be stored for future reuse. Furthermore, this approach is significantly more efficient comparing to the pseudo spectral approach because it scales as 2(P+1) or 3(P+1) with respect to number of random variables.

On the other hand the conventional linear regression holds general disadvantages of nonintrusive approaches mentioned in section 2.3.1.3. Moreover it is noted that existing works on the linear regression approach for circuit simulation have not provided any methodology to select the M regression nodes [27], [28]. Some works suggest choosing the M multidimensional nodes of (2.72) from the full tensor product grid of one dimensional (1D) Gaussian quadrature nodes. However, it is stressed that blindly choosing the M nodes from the tensor product grid is not guaranteed to give accurate results, and since the linear regression approach is less accurate because of its nonintrusive nature, it would have a poor accuracy. Nevertheless, in the field of estimation theory and data analysis various criteria to intelligently select the most important M quadrature nodes based on minimizing/maximizing some attribute of the information

matrix $\mathbf{A}^T\mathbf{A}$ of (2.74) have been reported [63]. Of these, the D-optimal criterion is the most popular and requires choosing the M nodes in such a way that the determinant of the information matrix $\mathbf{A}^T\mathbf{A}$ is maximized. Interested readers are directed to the references [63]-[65] for more details on the D-optimal criterion.

2.3.3 Stochastic collocation (SC) approach based on the Lagrange interpolation

This section provides a brief review of a nonintrusive stochastic collocation approach, in general all classical sampling methods like Monte Carlo are collocation techniques [49]; nevertheless, the focus in this section is on a SC approach based on Lagrange interpolation [67]. First basics of the Lagrange interpolation and its application to SC are presented, and then pros and cons of this approach are discussed.

2.3.3.1 Basics of the SC approach with Lagrange interpolations

For the problem of a single random variable $\boldsymbol{\lambda} = \lambda$, the SC approach expands the stochastic solution of (2.38) into a sum of weighted interpolation functions as [67], [68]

$$\mathbf{X}(t,\lambda) = \sum_{i=0}^{m} \mathbf{X}(t,\lambda^{(i)})\, l_i(\lambda) \tag{2.75}$$

where m+1 is number of the collocation nodes. $l_i(.)$ represents the univariate Lagrange interpolation function expressed as

$$l_i(\lambda) = \prod_{0\le j\le m,\ j\neq i} \frac{\lambda-\lambda^{(j)}}{\lambda^{(i)}-\lambda^{(j)}} \tag{2.76}$$

It is noted that although the Lagrange polynomial of (2.76) is a highly popular interpolation function, other interpolation functions such as the piecewise multi-linear function has also been employed in the context of the SC approach [68]. In (2.76), $\lambda^{(i)}$ represents the i^{th} collocation node out of a total of m+1 nodes and $l_i(\lambda^{(j)}) = \delta_{ij}$, where δ_{ij} is the Dirac delta function. Typically, the m+1 one dimensional (1D)

collocation nodes of (2.76) are the roots of the m+1th order polynomial basis chosen from the Weiner-Askey scheme [67], [68].

Extending the above methodology to the general multivariate problem, the SC expansion of (2.75) can be rewritten as:

$$\mathbf{X}(t,\lambda) = \sum_{i=0}^{M} \mathbf{X}(t,\boldsymbol{\lambda}^{(i)})\, L_i(\boldsymbol{\lambda}) \tag{2.77}$$

where $L_i(\boldsymbol{\lambda})$ is the multivariate Lagrange interpolation function and $\boldsymbol{\lambda}^{(i)} = [\lambda_1^{(i)}, \lambda_2^{(i)}, ..., \lambda_n^{(i)}]^t$ is the i^{th} multidimensional collocation node (out a total of M+1 multidimensional nodes). This multivariate Lagrange interpolation function can be constructed from the product of univariate Lagrange interpolation functions as

$$L_i(\boldsymbol{\lambda}) = \prod_{k=1}^{n} l_i(\lambda_k); \quad l_i(\lambda_k) = \prod_{0 \le j \le m,\ j \ne i} \frac{\lambda_k - \lambda_k^{(j)}}{\lambda_k^{(i)} - \lambda_k^{(j)}} \tag{2.78}$$

Based on (2.75) and (2.77), it is noted that unlike previously mentioned PC approaches coefficients of the SC expansion are always known. Within the context of interconnect networks, these coefficients can be obtained by deterministic SPICE simulations of the original model of (2.38) at the collocation nodes $\boldsymbol{\lambda}^{(i)}$, $0 \le i \le M$. Once the coefficients of (2.77) are obtained from SPICE, the mean and variance of a particular quantity of interest $x(t,\boldsymbol{\lambda}) \in \mathbf{X}(t,\boldsymbol{\lambda})$ of (2.38) can be computed as

$$\begin{aligned} E(x(t,\boldsymbol{\lambda})) &= \int_{\Omega_n} \left(\sum_{i=0}^{M} x(t,\boldsymbol{\lambda}^{(i)}) L_i(\boldsymbol{\lambda}) \right) \rho(\boldsymbol{\lambda})\, d\boldsymbol{\lambda} \\ Var(x(t,\boldsymbol{\lambda})) &= \int_{\Omega_n} \left(\sum_{i=0}^{M} x(t,\boldsymbol{\lambda}^{(i)}) L_i(\boldsymbol{\lambda}) - E(x(t,\boldsymbol{\lambda})) \right)^2 \rho(\boldsymbol{\lambda})\, d\boldsymbol{\lambda} \end{aligned} \tag{2.79}$$

The multidimensional integrals of (72) can be easily performed numerically using suitable quadrature rules [67].

2.3.3.2 Pros and cons of the SC approach with Lagrange interpolations

One advantage of the SC approach is its simplicity and being easy to implement since the coefficients in (2.77) are obtained merely from SPICE simulations and there is no post-analysis computation cost in order to extract these coefficients. Moreover, as a nonintrusive approach SC has major benefits which are mentioned in previous sections i.e. there is no restriction on the macromodeling algorithm used in the deterministic solution at each node and hence direct exploitation of sophisticated macromodeling techniques can be supported. Moreover, these simulations do not involve augmentation of the overall circuit model caused by the need to perform cumbersome inner product computations such as the ones which are required for the SG approach. Finally, the M+1 deterministic SPICE simulations at the collocation nodes are all independent of each other and can be easily parallelized unlike the SG approach where the augmented model of (2.40) is typically coupled.

On the other hand, it is noted from the above discussion that the CPU cost of the SC approach is proportional to the number of multidimensional collocation nodes (i.e. M+1) since for each node a new deterministic simulation of the network of (2.38) is required to evaluate each coefficient of (2.77). These multidimensional nodes are typically obtained from a tensor product of the 1D nodes of (2.76) thereby leading to an exponential scaling of the number of nodes (and deterministic SPICE simulations) with respect to the number of random dimensions. As a more efficient alternative, an intelligent choice of only a sparse subset of the tensor product nodes can also be used where the choice of the sparse nodes is guided by the Smolyak algorithm [69], [67]. The Smolyak algorithm leads to a significant reduction in the number of collocation nodes from $(m+1)^n$, for full tensor product grids, to approximately $(2n)^m/m!$. However, despite the comparative benefits of the sparse grids, the number of collocation nodes for this approach still scales in a polynomial fashion with respect to the number of random dimensions, thereby providing only limited benefits for large multidimensional random spaces. In order to address this poor scalability of the classical SC approach with respect to the number of random dimensions, in the next section a Stroud cubature based approach is proposed.

2.3.4 Stroud cubature based stochastic collocation approach

This section begins with a brief review of the Stroud cubature rules and its application to stochastic collocation approach. Moreover, pros and cons of the Stroud cubature approach are presented.

2.3.4.1 Stroud cubature rules and their application to uncertainty quantification

In the work of [51], Stroud cubature rules were introduced in order to compute multidimensional integrals weighted by arbitrary weighting functions over the hypercube space $[-1,1]^n$ as

$$\int_{[-1,1]^n} f(\boldsymbol{\lambda}) w(\boldsymbol{\lambda}) dx = \sum_{i=0}^{M} \omega_i f(\boldsymbol{\lambda}^{(i)}) \tag{2.80}$$

where $w(\boldsymbol{\lambda})$ is the arbitrary weighting function, ω_i is the weight associated with the i^{th} cubature node $\boldsymbol{\lambda}^{(i)}$ and $f(\boldsymbol{\lambda})$ is any integrand that can be approximated using a second or maximum third degree polynomials. When $f(\boldsymbol{\lambda})$ is approximated using a second degree polynomial, the second degree Stroud cubature rule (also called the S2 rule) dictates that the number of multidimensional cubature points in (2.80) is equal to $M+1 = n+1$. The location of the i^{th} cubature node within the $[-1,1]^n$ hypercube space is given as [51],

$$\boldsymbol{\lambda}_{2r-1}^{(i)} = \sqrt{\frac{2}{3}} \cos\left(\frac{2ri\pi}{n+1}\right); \; \boldsymbol{\lambda}_{2r}^{(i)} = \sqrt{\frac{2}{3}} \sin\left(\frac{2ri\pi}{n+1}\right) \tag{2.81}$$

for $r = 1,2,\ldots, \lfloor n/2 \rfloor$ provided n is an even number where $\lfloor n/2 \rfloor$ represents the greatest integer less than or equal to $n/2$. If n is an odd number, the location of the i^{th} cubature node along the last (i.e. n^{th} dimension) is given by [51]

$$\boldsymbol{\lambda}_n^{(i)} = \frac{(-1)^i}{\sqrt{3}} \tag{2.82}$$

It is noted that all weights ω_i for the S2 rule of (2.81), (2.82) are equal to $1/(n+1)$.

Similarly, when $f(\boldsymbol{\lambda})$ is approximated using a third degree polynomial, the third degree Stroud cubature rule (also called the S3 rule) dictates that the number of multidimensional cubature nodes in (2.80) is equal to $M+1 = 2n$. The location of the i^{th} cubature node within the $[-1,1]^n$ hypercube space is given by [51],

$$\boldsymbol{\lambda}_{2r-1}^{(i)} = \sqrt{\frac{2}{3}}\cos\left(\frac{(2r-1)(i+1)\pi}{n}\right);\ \boldsymbol{\lambda}_{2r}^{(i)} = \sqrt{\frac{2}{3}}\sin\left(\frac{(2r-1)(i+1)\pi}{n}\right) \tag{2.83}$$

for $r = 1,2,\ldots,\lfloor n/2 \rfloor$ provided n is an even number. If n is an odd number, the location of the i^{th} cubature node along the last (i.e. n^{th} dimension) is given by [51]

$$\boldsymbol{\lambda}_{n}^{(i)} = \frac{(-1)^i}{\sqrt{3}} \tag{2.84}$$

All weights ω_i for the S3 rules of (2.83), (2.84) are equal to $1/(2n)$.

One of the drawbacks of the Stroud cubature rules of (2.81)-(2.84) was that they were originally proposed for integrals performed over a hypercube space $[-1,1]^n$ which, in the context of the SC approach, is applicable for random variables exhibiting a uniform or beta probability distribution functions. However, for other probability distribution functions such as the normal distribution, the multidimensional integral of (2.80) is performed over the $[-\infty,\infty]^n$ space instead of the $[-1,1]^n$ hypercube, and hence the results of (2.81)-(2.84) are not directly applicable for such problems. Recently, in the work of [34], the Stroud cubature nodes and weights for normal distribution functions have been reported and these results have been provided in Table 2.4. Since the uniform and normal distributions are the most commonly encountered distribution functions, the proposed Stroud cubature based SC approach of this section can directly utilize the results of Table 2.4.

By setting the collocation nodes of (2.75) as the Stroud cubature nodes of (2.81), (2.82) or (2.83), (2.84) depending on whether $\mathbf{X}(t,\boldsymbol{\lambda})$ is well approximated by a second or third degree polynomial

Table 2.4 Stroud nodes and weights for common probability distributions

Distribution	Weight		Nodes	
	S2	S3	S2	S3
Uniform	$\frac{1}{n+1}$	$\frac{1}{2n}$	$\boldsymbol{\lambda}_{2r-1}^{(i)} = \sqrt{\frac{2}{3}}\cos\left(\frac{2ri\pi}{n+1}\right)$; $\boldsymbol{\lambda}_{2r}^{(i)} = \sqrt{\frac{2}{3}}\sin\left(\frac{2ri\pi}{n+1}\right)$ where $r = 1,2,\ldots,[n/2]$ and provided n is even. If n is odd, $\boldsymbol{\lambda}_{n}^{(i)} = \frac{(-1)^{i}}{\sqrt{3}}$	$\boldsymbol{\lambda}_{2r-1}^{(i)} = \sqrt{\frac{2}{3}}\cos\left(\frac{(2r-1)(i+1)\pi}{n}\right)$; $\boldsymbol{\lambda}_{2r}^{(i)} = \sqrt{\frac{2}{3}}\sin\left(\frac{(2r-1)(i+1)\pi}{n}\right)$ where $r = 1,2,\ldots,[n/2]$ and provided n is even. If n is odd, $\boldsymbol{\lambda}_{n}^{(i)} = \frac{(-1)^{(i+1)}}{\sqrt{3}}$
Normal or Beta			$\boldsymbol{\lambda}_{2r-1}^{(i)} = \sqrt{2}\cos\left(\frac{2ri\pi}{n+1}\right)$; $\boldsymbol{\lambda}_{2r}^{(i)} = \sqrt{2}\sin\left(\frac{2ri\pi}{n+1}\right)$ where $r = 1,2,\ldots,[n/2]$ and provided n is even. If n is odd, $\boldsymbol{\lambda}_{n}^{(i)} = (-1)^{i}$	$\boldsymbol{\lambda}_{2r-1}^{(i)} = \sqrt{2}\cos\left(\frac{(2r-1)(i+1)\pi}{n}\right)$; $\boldsymbol{\lambda}_{2r}^{(i)} = \sqrt{2}\sin\left(\frac{(2r-1)(i+1)\pi}{n}\right)$ where $r = 1,2,\ldots,[n/2]$ and provided n is even. If n is odd, $\boldsymbol{\lambda}_{n}^{(i)} = (-1)^{(i+1)}$

respectively, and applying the cubature approximation of (2.80) to the multidimensional integrals of (2.79), after some algebraic manipulations, the mean can be expressed as

$$E(x(t,\boldsymbol{\lambda})) = \sum_{i=0}^{M} \omega_i x(t,\boldsymbol{\lambda}^{(i)}) \tag{2.85}$$

where the weighting function $w(\boldsymbol{\lambda})$ has been set equal to the joint probability distribution function $\rho(\boldsymbol{\lambda})$.

By using the same methodology of above and the added knowledge that multiplication of Lagrange polynomials yields $L_i(\boldsymbol{\lambda})L_j(\boldsymbol{\lambda}) = \delta_{ij}$, the expression of the variance in (2.79) can also be simplified to

$$Var(x(t,\boldsymbol{\lambda})) = \sum_{i=0}^{M} \left(\omega_i x^2(t,\boldsymbol{\lambda}^{(i)})\right) - E(x(t,\boldsymbol{\lambda}))^2 \tag{2.86}$$

where the mean term $E(x(t,\boldsymbol{\lambda}))$ in (2.86) can be obtained from (2.85). It is noticed from (2.85) and (2.86) that the mean and variance (i.e. first and second order statistical moments) of any quantity of interest can be analytically computed directly from the SPICE results of the stochastic network of (2.38) evaluated at

M+1 sparse Stroud nodes of (2.81)-(2.84). This feature of the proposed Stroud based SC approach can also be extended for any general higher order statistical moment of $x(t,\boldsymbol{\lambda})$. Equation (2.85) and (2.86) can be used for other distributions in Table 2.4 as well.

It is emphasized that since the work of Stroud [51], many other cubature rules for varying number of dimensions and degrees greater than 3 have been reported, a comprehensive list of which is provided in [70], [71]. However, none of the reported methods preserve the highly attractive linear scalability feature of the Stroud rules, though many of them still scale far better than the polynomial scalability of the SC approach based on sparse grids. Furthermore, since the gPC theory assumes that the output of interest exhibits a smooth dependence on the random variables, a third order polynomial expansion has been found to be sufficient for most contemporary interconnect problems [10]. For the above reasons, the Stroud method still remains a highly effective cubature rule for efficient uncertainty quantification.

2.3.4.2 Pros and cons of the Stroud cubature based stochastic collocation approach

The main advantage of the Stroud cubature approach is its scaling rate. Unlike other stochastic analysis methods which scale at a polynomial or exponential rate, this approach scales linearly with respect to number of random variables since number of sample nodes and SPICE simulation is n+1 for the S2 rule and 2n for the S3 rule. Therefore, the Stroud cubature rule overcomes the curse of dimensionality, mentioned in Chapter I, for random variables which can be approximated by second or third degree polynomials. Moreover, this approach is able to extract any statistical moment merely from SPICE simulations and without post analysis costs. It is worth noting in the PC approach it is necessary to analyze data after SPICE simulations to extract PC coefficients, and even after that the coefficients can provide a closed form solution for up to the second order statistical moment i.e. the variance. And for higher order moments it is necessary to generate thousands of MC samples using the PC coefficients. Finally it goes without saying that the Stroud cubature approach is nonintrusive and therefore it can use commercial software and do parallelization.

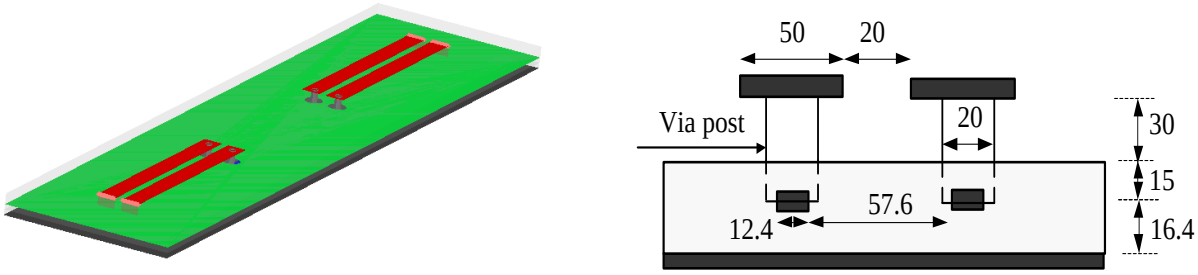
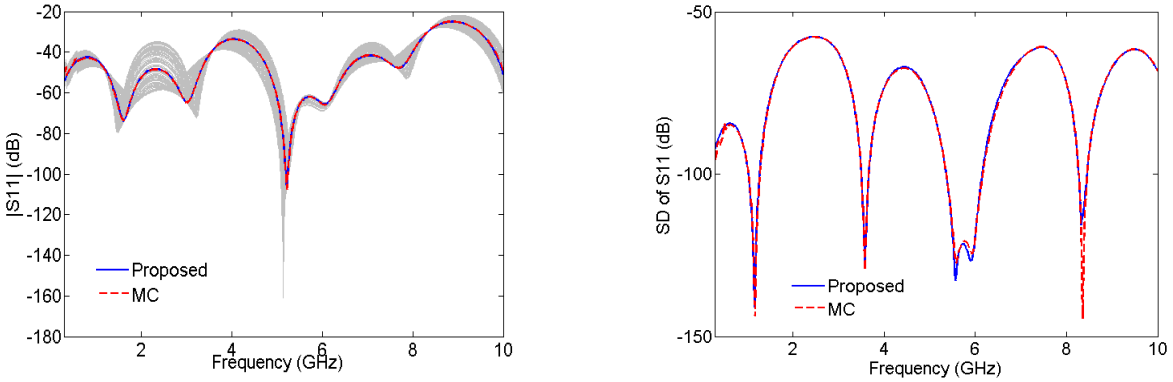


Fig. 2.4: Topology of microstrip-stripline-microstrip structure analyzed using ANSYS Designer 3D planar MOM solver. (a) 3D view. (b) Cross-section view (all dimensions are shown in mils).

On the other hand, the main disadvantage of the Stroud cubature approach is being limited to the third degree polynomials. And despite the higher order cubature rules are able to address this issue, they do not provide the optimal scaling rate of the Stroud cubature approach. In other words, although majority of stochastic variables are smooth and can be approximated with a maximum of third degree polynomials, this issue causes lack of generality for the Stroud approach. Moreover, Stroud cubature rules apply to specific distributions, while in practical examples the distribution of output might be non-standard which increases the error; therefore, the Stroud cubature rule would not be able to handle some of possible outcomes. It is worth noting in the PC approaches the mismatch in distribution can be compensated by increasing the order of expansion; however, it is not possible to increase the order more than three in Stroud cubature rules.

2.4 A numerical example

As an example, in order to exhibit the application of uncertainty quantification methods, the interconnect network of Fig. 2.4 is considered and the pseudo spectral PC and Monte Carlo approaches are taken for computation of statistical results. The microstrip traces are of length 625 mils, width 50 mils, separated by a distance of 20 mils and located 61.4 mils above the ground plane. The stripline traces are of length 625 mils, width 12.4 mils, separated by a distance of 57.6 mils and located 16.4 mils above the ground plane. The microstrip and stripline traces are connected by via posts of height 45 mils and diameter 20 mils passing through a dielectric layer of relative permittivity $\varepsilon_r = 4$. This system is



(a) (b)

Fig. 2.5: Comparison of ***S***-parameter statistics of the example of Fig. 2.4 with the width of microstrip traces as a stochastic parameter using proposed LM macromodel and Monte Carlo (10,000 samples). (a) Mean of S_{11} in dB. (b) Standard deviation of S_{11} in dB.

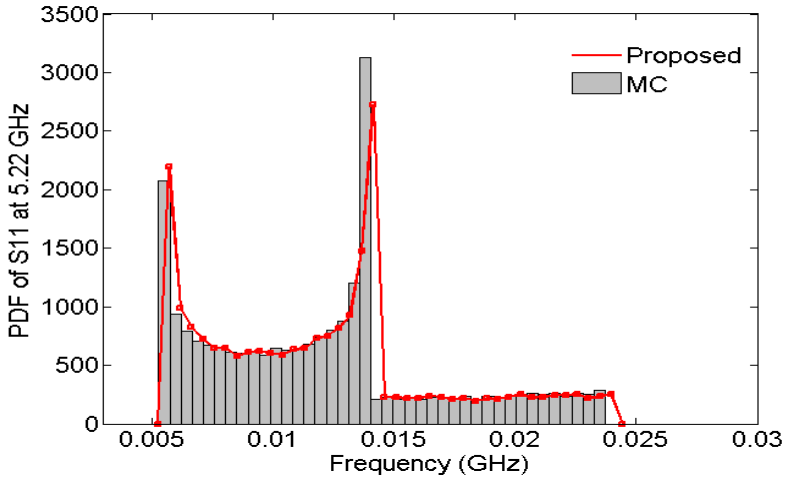
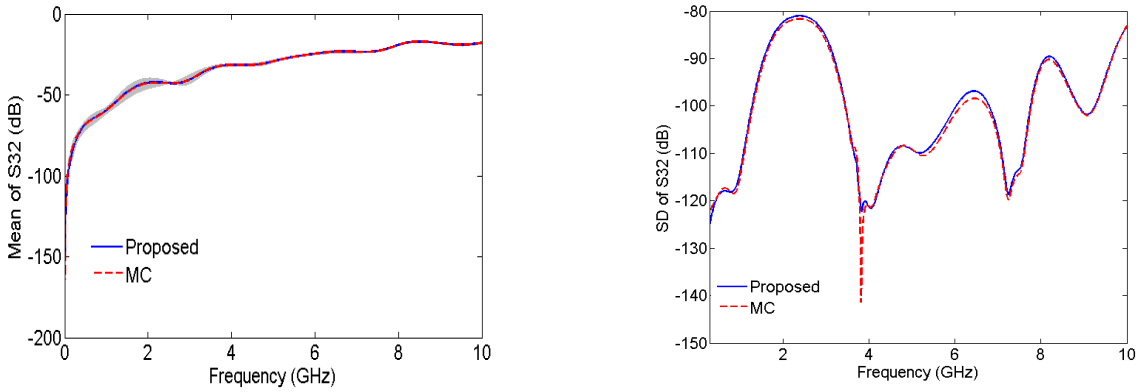


Fig. 2.6: PDF comparison between the proposed LM macromodel and Monte Carlo (20,000 samples) of S_{11} of Fig. 2.4 at 5.22 GHz.

characterized by its 4-port S-parameters computed using the commercial ANSYS Designer 3-D planar EM solver based on the method of moments (MoM). Moreover, the stochastic Loewner matrix (LM) technique [29] is used to generate the root macromodels $\mathbf{S}(s,\boldsymbol{\lambda}^{(k)})$ in (2.59) from the sampled data.

In the first test case, the width of the microstrip traces is a stochastic parameter with the nominal values of above and exhibiting a +/- 10% relative uniform variation. This requires a third order Legendre polynomial expansion of the S-parameters as in:

$$\mathbf{S}(s,\boldsymbol{\lambda}) = \sum_{i=0}^{P} \mathbf{S}_i(s)\phi_i(\boldsymbol{\lambda}),\ \mathbf{S}_i(s) \approx \sum_{k=1}^{Q} \mathbf{S}(s,\boldsymbol{\lambda}^{(k)})\phi_i(\boldsymbol{\lambda}^{(k)})\omega(\boldsymbol{\lambda}^{(k)}) \tag{2.87}$$



(a) (b)

Fig. 2.7: Comparison of ***S***-parameter statistics of the example of Fig.2.4 with the width and separation of the microstrip traces as stochastic parameters using proposed LM macromodel and Monte Carlo (10,000 samples). (a) Mean of S_{32} in dB. (b) Standard deviation of S_{32} in dB.

where $\mathbf{S}(s,\boldsymbol{\lambda}^{(k)})$ is the ***S***-parameter data at $\boldsymbol{\lambda}^{(k)}$ which can be easily obtained using deterministic full-wave EM solvers. Using a Gaussian-Legendre quadrature rule, the number of quadrature nodes is $Q = 4$ in (2.87) and are located at the roots of the fourth order Legendre polynomial. The accuracy of the pseudo spectral approach is compared against the MC simulation using 10,000 random samples of the uniform random variable. The mean and standard deviation of the S-parameter S11 is shown in Fig. 2.5 over the bandwidth 10 MHz – 10 GHz where both the above techniques show excellent agreement. The gray lines illustrate the S-parameter variation using 1,000 MC samples. Fig. 2.6 shows the probability distribution function of S11 at the minima of 5.22 GHz using the above two approaches. In order to achieve the excellent agreement of Fig. 2.6, the number of MC simulations had to be increased to 20,000.

In the next test case, in addition to the trace width, the microstrip trace separation is another stochastic parameter with the nominal value of above and a +/- 10% relative uniform variation. The order of the Legendre polynomial expansion of (2.12) is still kept at 3. In this case, the same Gaussian-Legendre quadrature rule is used leading to $Q = 16$ quadrature nodes in (2.87). The comparison between the pseudo spectral approach and MC simulations are performed similar to the previous test case. The mean and standard deviation of the S-parameter S32 is shown in Fig. 2.7 where both the above approaches show excellent agreement. The gray lines illustrate the S-parameter variation using 1,000 MC samples. It is appreciated that for this case the proposed LM based macromodeling strategy required only

16 deterministic simulations while the MC required a total of 10,000 samples leading to over two orders of magnitude in speedup.

CHAPTER III: PROPOSED FAST LINEAR REGRESSION APPROACHES

Among multidimensional uncertainty approaches introduced in Chapter II, the nonintrusive linear regression seems to have the highest potential for stochastic analysis of high-speed circuits. Firstly, being nonintrusive this approach is able to take advantage of the available commercial software and easily do parallelization on the required deterministic simulations at sample nodes. Moreover, this approach provides a desirable accuracy and by adding an optimal design of experiments it is able to extract the most critical sample nodes. The number of sample nodes is equal to $2(P+1)$ or $3(P+1)$; however, by taking some consideration this number can be further reduced, while maintaining the accuracy. Finally, it is worth noting, linear regression is available for different distributions and orders of expansion.

This chapter starts with a through explanation of the search algorithm of the linear regression approach. This includes description of the rationale behind the D-optimal criterion and the Fedorov search algorithm. Then noble ideas for improvement on the nonintrusive linear regression approach are introduced in the next two sections. The first discussed approach is called Sparse Linear Regression (SPLINER) which we have published in [61]. The contributions in this approach are developing a sparse node selection technique and then modifying it further to be faster through limiting the search space and reducing number of matrix inversions. The second approach is another modified linear regression methodology and we have published it in [72], this approach expedites the node selection technique by reducing number of iterations and exploiting a fast matrix inversion method.

In order to validate these techniques and compare their scaling rate with other PC approaches, multiple numerical examples on high-speed circuits and microwave/RF networks are presented. Although both approaches can be applied at the same time, in numerical examples we apply them separately to demonstrate their individual impact.

3.1 The search algorithm in the linear regression approach

As mentioned in Chapter II, it is noted that existing works on the linear regression approach for circuit simulation have not provided any methodology to select the *M* regression nodes [27], [28]. Some works suggest choosing the *M* multidimensional nodes of (2.72) from the full tensor product grid of one dimensional (1D) Gaussian quadrature nodes. However, it is stressed that blindly choosing the *M* nodes from the tensor product grid is not guaranteed to give accurate results, and since the linear regression approach is less accurate because of its nonintrusive nature, it would have a poor accuracy. Nevertheless, various criteria to intelligently select the most important *M* quadrature nodes based on minimizing/maximizing some attribute of the information matrix $\mathbf{A}^T\mathbf{A}$ of (2.74) have been reported in the literature [63]. Of these, the D-optimal criterion is the most popular and requires choosing the *M* nodes in such a way that the determinant of the information matrix $\mathbf{A}^T\mathbf{A}$ is maximized. The selection of the D-optimal nodes is commonly performed using the classical Fedorov search algorithm [66]. In this section, first the reasoning for selection of the D-optimal criterion is presented and then the Fedorov search algorithm for selecting the nodes is described.

3.1.1 D-optimal criterion

Based on the discussion in Section 2.3.2.2 the linear regression approach tries to solve the following system of algebraic equation:

$$\mathbf{A}\tilde{\mathbf{X}} = \mathbf{E} \tag{3.1}$$

where

$$\mathbf{A} = \begin{bmatrix} \phi_0(\boldsymbol{\lambda}^{(1)})\mathbf{I} & \dots & \phi_P(\boldsymbol{\lambda}^{(1)})\mathbf{I} \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^{(M)})\mathbf{I} & \dots & \phi_P(\boldsymbol{\lambda}^{(M)})\mathbf{I} \end{bmatrix}, \tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_0(t) \\ \vdots \\ \mathbf{X}_P(t) \end{bmatrix}, \mathbf{E} = \begin{bmatrix} \mathbf{X}(t, \boldsymbol{\lambda}^{(1)}) \\ \vdots \\ \mathbf{X}(t, \boldsymbol{\lambda}^{(M)}) \end{bmatrix} \tag{3.2}$$

with $\mathbf{I}$ being the identity matrix.

The importance of the D-optimal criterion to the accuracy of the evaluated PC coefficients of (3.1) is revealed using the following *Lemma*.

Lemma 1: Assuming that the truncation error $\varepsilon_j, 1 \le j \le M$ at all M design of experiments (DoE) of (3.1) are independent of each other and exhibit a normal distribution of zero mean and same variance σ^2, then in order to achieve the maximum accuracy of the PC coefficients the DoE must be chosen such that the determinant of the information matrix $\mathbf{A}^T\mathbf{A}$ of (3.1) is maximized.

Proof: Based on the PC expansion of the network responses of (2.69), it is understood that the presence of the random truncation error ε makes the PC coefficients themselves random variables. The variance of the evaluated PC of (3.1) can be computed as

$$Var(\tilde{\mathbf{X}}) = Var((\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{E}) = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t Var(\mathbf{E})((\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t)^t \tag{3.3}$$

Knowing that the truncation error for each DoE (i.e. ε_j) is independent and has a constant variance σ^2, $Var(\mathbf{E}) = \sigma^2\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Replacing this in (3.3) the variance of the PC coefficients of (3.3) can be compactly expressed as

$$Var(\tilde{\mathbf{X}}) = (\mathbf{A}^t\mathbf{A})^{-1}\sigma^2 \tag{3.4}$$

From (3.4) we understand that to ensure the maximum accuracy of the PC coefficients we have to reduce the uncertainty in the solution $\tilde{\mathbf{X}}$ (i.e. the variance of $\tilde{\mathbf{X}}$). Since the variance of $\tilde{\mathbf{X}}$ is inversely proportional to the determinant of the information matrix $\mathbf{A}^T\mathbf{A}$, a simple way to minimize the variance of $\tilde{\mathbf{X}}$ is to maximize the determinant. Therefore, the M DoE for the linear regression of (3.1) must be chosen so as to maximize the determinant of the information matrix. This criterion is referred to as the D-optimal criterion [63], [65]. It is noted that other optimal criterions besides the D-optimal criterion also exists although the D-optimal criterion has been deemed the most effective and popular till now [63]. The next challenge is to develop a search algorithm that can efficiently identify the D-optimal nodes from multidimensional random spaces.

3.1.2 Greedy search algorithm to identify DoE (classical Fedorov)

In this section a detailed description of the greedy search algorithm to identify the D-optimal DoE from multidimensional random spaces is presented. This greedy search algorithm is based on the Fedorov algorithm commonly used in the field of estimation theory and data analysis [66], [73]. This algorithm begins by considering a set of $M = 2(P+1)$ or $M = 3(P+1)$ starting DoE selected from the tensor product grid of $(m+1)^n$ multidimensional quadrature nodes and creating the corresponding information matrix $\mathbf{A}^T\mathbf{A}$ of (3.1). Thereafter, each DoE in the starting set is replaced by the best possible substitute DoE taken from the remaining $(m+1)^n$-M quadrature nodes such that the determinant of the information matrix increases by the maximum amount in the process. This step-by-step refinement of the starting DoE continues till all the initial set of nodes has been replaced [39].

As per the above description, at the r^{th} step it is assumed that the first r-1 nodes have been replaced by their best possible substitutes. Now if the r^{th} DoE ($\boldsymbol{\lambda}^{(r)}$) of the starting set is removed from $\mathbf{A}$, then the new determinant of the information matrix can be expressed as

$$\begin{aligned}\det(\mathbf{A}^t\mathbf{A})_{new} &= \det((\mathbf{A}^t\mathbf{A}) - \mathbf{R}(\boldsymbol{\lambda}^{(r)})\mathbf{R}^t(\boldsymbol{\lambda}^{(r)})) \\ &= \det(\mathbf{A}^t\mathbf{A})(1 - \mathbf{R}(\boldsymbol{\lambda}^{(r)})(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{R}^t(\boldsymbol{\lambda}^{(r)}))\end{aligned} \tag{3.5}$$

where $\mathbf{R}(\boldsymbol{\lambda}^{(r)})$ is the row vector contributed by the r^{th} DoE ($\boldsymbol{\lambda}^{(r)}$) in $\mathbf{A}$. Similarly, if any arbitrary k^{th} DoE ($\boldsymbol{\lambda}^{(k)}$) from the remaining $(m+1)^n$-M quadrature nodes is included into $\mathbf{A}$, the new determinant of the information matrix can be expressed as

$$\begin{aligned}\det(\mathbf{A}^t\mathbf{A})_{new} &= \det((\mathbf{A}^t\mathbf{A}) + \mathbf{R}(\boldsymbol{\lambda}^{(k)})\mathbf{R}^t(\boldsymbol{\lambda}^{(k)})) \\ &= \det(\mathbf{A}^t\mathbf{A})(1 + \mathbf{R}(\boldsymbol{\lambda}^{(k)})(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{R}^t(\boldsymbol{\lambda}^{(k)}))\end{aligned} \tag{3.6}$$

Combining the results of (3.5) and (3.6), after exchanging the r^{th} DoE ($\boldsymbol{\lambda}^{(r)}$) of the starting set with any arbitrary k^{th} DoE ($\boldsymbol{\lambda}^{(k)}$) from the remaining $(m+1)^n$-M quadrature nodes, the new determinant of the new information matrix can be mathematically expressed as a recursive function

$$\begin{aligned}\Delta_r &= \det(\mathbf{A}^t\mathbf{A})_{new} = \det(\mathbf{A}^t\mathbf{A})(1 + d_{kk} - d_{rr} + d_{kr}^2 - d_{kk}d_{rr}) \\ d_{kr} &= \mathbf{R}(\boldsymbol{\lambda}^{(k)})\boldsymbol{\Psi}^{(r-1)}\mathbf{R}^t(\boldsymbol{\lambda}^{(r)})\end{aligned} \tag{3.7}$$

where $\boldsymbol{\Psi}^{(r-1)}$ represents the inverse of the information matrix obtained after the previous (i.e. r-1$^{\text{th}}$) exchange. From (3.7) it is understood that in order to achieve D-optimality, the k^{th} node $\boldsymbol{\lambda}^{(k)}$ needs to be so chosen to satisfy the optimization criterion

$$\max(d_{kk} - d_{rr} + d_{kr}^2 - d_{kk}d_{rr}) \tag{3.8}$$

Once the best possible node $\boldsymbol{\lambda}^{(k)}$ has been found to satisfy (3.8) and the relevant exchange has been made, the new determinant can be directly updated using (3.7) and the substitution process moves on to the r+1$^{\text{th}}$ node. Once all M starting DoE have been replaced the new set of DoE will represent the D-optimal selection.

It is noted that the total computational cost of the search algorithm is due to two main factors. Firstly, identifying the D-optimal DoE requires searching through $(m+1)^n$-M quadrature nodes for each DoE in the starting set – in other words, a total of $M((m+1)^n$-$M)$ searches. The associated CPU cost can be expressed as

$$C_a = 2(P+1)((m+1)^n - 2(P+1))C_1 \approx 2(P+1)(m+1)^n C_1 \tag{3.9}$$

where C_1 is the CPU cost of computing the terms in the brackets of (3.7) assuming that the inverse $\boldsymbol{\Psi}^{(r-1)}$ is known. It is noted that based on (3.5) and (3.6) C_1 can be expressed as

$$C_1 = 3k((P+1)^2 + (P+1)) \tag{3.10}$$

where the first term is the cost of the matrix-vector multiplication $\boldsymbol{\Psi}^{(r-1)}\mathbf{R}^t(\boldsymbol{\lambda}^{(r)})$, the second term is the cost of the vector-vector multiplication of $\mathbf{R}(\boldsymbol{\lambda}^{(r)})$ with $\boldsymbol{\Psi}^{(r-1)}\mathbf{R}^t(\boldsymbol{\lambda}^{(r)})$, and the factor 3 is due to the fact that the above operations needs to be performed for three scalars d_{rr}, d_{kk}, and d_{kr} of (3.7). Also k is assumed to be the cost of each floating point operation. Combining (3.9) and (3.10), it can be concluded that the overall

search cost (C_a) scales in an exponential manner with the number of random dimensions (n), quantified as $O((P+1)^3(m+1)^n)) \approx O(n^{3m}(m+1)^n)$.

The other source of computational effort arises from the fact that for each substitution, the information matrix changes and the inverse $\Psi^{(r-1)}$ has to be reevaluated. This CPU cost is expressed as

$$C_b = 2(P+1)C_2 \tag{3.11}$$

where C_2 is the CPU cost of each matrix inversion. Note that for direct inversion methods C_2 scales as $O((P+1)^3)$ thereby ensuring that the cumulative cost of the matrix inversions (C_b) scales as $O((P+1)^4)$ $\approx O(n^{4m})$ with respect to the number of random dimensions (n). Given that for typical PC problems $2 \le m \le 5$, this suggests a near exponential scaling of the associated CPU costs.

The above two features of the search algorithm significantly slow down its performance for high-dimensional problems and may even render it infeasible for some problems. In fact, the cost of implementing the search algorithm can often become a significant fraction of the cost of performing the M deterministic SPICE simulations of (3.1) as will be demonstrated in the numerical examples section. In order to address these computational constraints of the search algorithm, in coming sections two main approaches to expedite the search algorithm for problems involving high-dimensional random spaces is presented.

3.2 Development of the proposed SPLINER approach

In this section the proposed SPLINER approach based on sparse node selection is described. Thereafter, the modified Fedorov search algorithm for expeditiously locating the sparse nodes from a high-dimensional random space is presented. Moreover an in-depth analysis of the computational cost of the SPLINER approach compared against that of conventional non-intrusive PC approaches, is provided. Finally, this section concludes with numerical examples in order to validate the proposed approach and study its computational cost scaling rate with respect to number of random variables.

3.2.1 SPLINER with sparse node selection

It is noted that the matrix **A** in (3.1) being time independent, the optimal nodes needs to be constructed only once and can be stored for future reuse. However, the conventional linear regression approach oversamples the PC expansion of (2.69) resulting in 2(P+1) or 3(P+1) deterministic SPICE simulations [46]. This may lead to an unsustainably high CPU time and memory costs in order to extract the PC coefficients of large circuit networks. In order to curb the CPU costs of the SPICE simulations, in this work only the minimum number of nodes required for a self-consistent evaluation of the PC coefficients of (2.69) is selected. This means that the number of nodes for the SPLINER approach is set to $M = P+1$ whereby the matrix **A** of (3.1) becomes a square matrix. Now, (3.1) can be solved directly to accurately evaluate the PC coefficients of the network provided the matrix **A** is full rank, well-conditioned, and composed of the most important P+1 quadrature nodes. In this work, the D-optimal P+1 nodes extracted by the Fedorov search algorithm is considered to represent the most important quadrature nodes. It is appreciated that the Fedorov search algorithm is sufficiently robust such that selecting this sparse P+1 D-optimal nodes (instead of the 2(P+1) or 3(P+1) D-optimal nodes required by the conventional linear regression approach) will still lead to a very high determinant of the information matrix $\mathbf{A}^T\mathbf{A}$. Since the determinant of $\mathbf{A}^T$ is equal to that of **A**, this in turn means that the Fedorov algorithm will automatically ensure that the determinant of **A** itself is very large. This indicates that the matrix **A** is full rank and well-conditioned.

3.2.2 Modified Fedorov algorithm

In the previous section, the characteristics of the proposed SPLINER approach (i.e. use of only a sparse set of P+1 nodes for the linear regression) was presented. While using a sparse set of nodes reduces the number of deterministic SPICE simulations required, it is noted from Section 3.1.2 that locating these nodes from a high-dimensional random space using the classical Fedorov search algorithm still remains a time-intensive process. In order to expedite the search process, a modified Fedorov search algorithm based on the following two novel features is proposed next.

3.2.2.1 Pruning the search space

Based on the discussion of Section 3.1.2, it is appreciated that one major challenge facing the classical Fedorov algorithm is the fact that the search space for locating the D-optimal nodes consists of $(m+1)^n$ Gaussian quadrature nodes. This means that the search space grows exponentially with the number of random dimensions, thereby slowing down the Fedorov search algorithm for even small-dimensional problems. In the modified Fedorov algorithm, initially a subset consisting of $10(P+1)$ Gaussian quadrature nodes with the largest quadrature weights will be chosen. If the number of random dimensions (n) is small enough, then the total number of Gaussian quadrature nodes may be less than $10(P+1)$ nodes (i.e. $(m+1)^n < 10(P+1)$). e.g., if $n = 4$, $m = 3$. In such cases, the total set of quadrature nodes is small enough to be directly probed to locate the best possible substitutes. Once the $10(P+1)$ nodes are extracted, from this reduced set of nodes the starting set of $M = P+1$ nodes will be randomly selected to populate the **A** matrix. The remaining $9(P+1)$ quadrature nodes will represent the new reduced search space from which the D-optimal nodes will be selected. Therefore, in this methodology, for each starting node, the best possible substitute node will be selected from the subset of $9(P+1)$ nodes rather than the massive set of $(m+1)^n$-M nodes thereby significantly expediting the Fedorov algorithm.

3.2.2.2 Constrained exchange criterion

It is further noted from Section 3.2.1 that for each refinement of the starting nodes the information matrix changes and its inverse needs to be recomputed for use in (3.7). This translates to a maximum of $P+1$ inversions of the information matrix for the proposed SPLINER. Since $P+1$ scales in a polynomial fashion with the number of random dimensions, for high-dimensional problems a large number of matrix inversions needs to be evaluated thereby once again slowing down the classical Fedorov search algorithm. In the modified Fedorov algorithm, rather than simply exchanging a node from the starting set with one from the reduced search space that yields the maximum increase in the determinant, an additional constraint criterion

$$\Delta_r > 11\Delta_{r-1} \tag{3.12}$$

is considered before making an exchange. Based on (3.7), the constraint of (3.12) translates to the new criterion for the substitution of nodes expressed as

$$d_{kk} - d_{rr} + d_{kr}^2 - d_{kk}d_{rr} > 10 \tag{3.13}$$

This criterion will ensure that unless and until the substitute node is able to increase the determinant of the information matrix significantly (i.e. by more than 10 times) then that node is not considered to be a viable replacement. In other words, by adding the constraint of (3.12), the exchange criterion of (3.13) becomes stricter thereby substantially reducing the number of possible substitutions, and consequently, the number of matrix inversions than what would be required by the classical Fedorov algorithm. Nevertheless, the constraint of (3.12) will still ensure that the improvement in the determinant of the information matrix for every valid exchange is large enough so that the determinant of matrix **A** ultimately reaches a very large value.

Overall, the proposed modified Fedorov algorithm will address both the two major sources of inefficiency of the classical Fedorov algorithm for high-dimensional problems – the exponential increase in the search space and the large number of matrix inversions. This along with the reduction in number of deterministic SPICE simulations will results in a more CPU-efficient approach to evaluate the PC coefficients.

3.2.3 Computational cost analysis

This section quantifies the overall CPU time costs incurred by the proposed SPLINER approach and contrasts that against conventional non-intrusive PC approaches. It is observed that no comparative analysis with respect to the intrusive stochastic Galerkin (SG) approach is performed since it is well-established that for high-dimensional problems most non-intrusive approaches will outperform the SG approach [45]. This is particularly true for nonlinear circuits where the multidimensional integral of (2.37)

using the SG approach has to be represented in SPICE using massive number of additional voltage/current dependent sources.

Based on the discussion of Section 3.2.1 and 3.2.2, it is appreciated that the CPU time required for the proposed SPLINER approach can be divided into two parts – the cost incurred during selecting the P+1 regression nodes using the modified Fedorov algorithm and the cost incurred while performing the P+1 deterministic SPICE simulations. Each of these two parts of the algorithm is studied separately.

3.2.3.1 Cost of modified Fedorov search algorithm

The cost of the modified Fedorov search algorithm can be further divided into the cost of finding the suitable exchange nodes and that required for re-evaluating the inverse of the information matrix. Beginning with the cost of finding the suitable exchange nodes, for each of the P+1 starting nodes, the value of Δ_r of (3.7) is computed 10(P+1) times. This makes a total of 10(P+1)^2 computations of Δ_r and the associated time costs can be mathematically quantified as:

$$C_{ex} = 10(P+1)^2 C_1 \tag{3.14}$$

where C_1 is the costs of performing all the matrix-vector and vector-vector multiplications of (3.7) assuming that the inverse of the information matrix is known. Next, assuming that due to the constraint condition of (3.12), (3.13) only R matrix inversions ($R \leq P+1$) are required, the total time costs required to implement the modified Fedorov algorithm can be quantified as the sum

$$C_t = 10(P+1)^2 C_1 + RC_2 \tag{3.15}$$

where C_2 is the time cost to perform one matrix inversion. It is appreciated that the cost C_1 scales as $O(n^{2m})$ with respect to the random dimensions (n). On the other hand, the cost C_2 scales as $O(n^{3m})$ or $O(n^{2m})$ depending on whether a direct or indirect approach for matrix inversion is used.

This overall CPU time cost is contrasted with that incurred by the classical Fedorov algorithm expressed as

$$C_t = K(P+1)((m+1)^n - K(P+1))C_1 + K(P+1)C_2$$
$$\approx K(P+1)(m+1)^n C_1 + K(P+1)C_2; \quad K \in \{2,3\} \tag{3.16}$$

where the first term on the right hand side of (3.16) represents the CPU cost required to compute Δ_r of (3.7) $(m+1)^n$-$K(P+1)$ times for each node of the starting set while the second term represents the CPU cost of the requisite matrix inversions. By comparing (3.15) with (3.16) and recalling that $(m+1)^n >> 10(P+1)$ for even moderate number of random dimensions and always $R < K(P+1)$, it is concluded that the modified Fedorov algorithm is substantially more efficient than its classical counterpart.

3.2.3.2 SPICE simulation cost

From the discussion of Section 3.2.1, it is understood that the proposed SPLINER approach requires only $P+1$ SPICE simulations. This is smaller than the $2(P+1)$ or $3(P+1)$ SPICE simulations required by the conventional linear regression approach [27], [28], [46]. Hence, the SPLINER approach incurs only a fraction of the cost of the conventional linear regression approach when performing the deterministic SPICE simulations.

Overall, the SPLINER approach is demonstrated to be computationally far more efficient than the conventional linear regression approach both in implementation of the Fedorov search algorithm and the number of SPICE simulations required, especially for high-dimensional problems. It is further emphasized that the proposed SPLINER approach is significantly more efficient than the rigorous pseudo-spectral collocation approach for even moderate dimensional problems [29]. This is due to the fact that although the pseudo-spectral collocation does not involve the cumbersome Fedorov search algorithm, the nodes of the SPICE simulations represent the full tensor product grid of 1D Gaussian quadrature nodes [45]. Thus, the number of deterministic SPICE simulations required by the pseudo-spectral collocation approach scales in an exponential manner as $(m+1)^n$,where even for moderate number of random dimensions $(m+1)^n >> P+1$. This massive number of SPICE simulations renders this approach too computationally expensive compared to the SPLINER approach even after considering the

CPU costs incurred for implementing the modified Fedorov algorithm. In fact, given that the SPLINER approach requires the minimum P+1 number of SPICE simulations to solve for the PC coefficients in (3.1), it is appreciated that this approach is expected to be more efficient than even the SC approach based on sparse grids [30]-[33] which requires only a fraction of the total tensor product nodes.

Among other non-intrusive approaches, that based on the ST approach introduced in Section 2.2.2.3 is highly popular and powerful [37]. This approach does not involve any matrix inversions (i.e. R = 0). Moreover, it is based on a non-optimal node selection criterion which requires a smaller number of matrix-vector multiplications than the proposed modified Fedorov algorithm, thereby making it relatively more time-efficient. However, since the modified Fedorov algorithm is still based on a D-optimal criterion, it is more accurate than the work of [37] where a non-optimal (relaxed) selection criterion is used to expedite the ST search algorithm. In addition, for most problems the SPICE simulation costs dominate over that of the search algorithm. Since both the proposed SPLINER and nonintrusive ST based approaches perform the same number of SPICE simulations, their difference in overall CPU time costs is usually small. The comparison of the computational complexity of the SPLINER approach with existing non-intrusive PC approaches is further illustrated using numerical examples in the next section.

3.2.4 Numerical examples regarding the SPLINER approach

In this section, three examples are presented to compare the accuracy and scalability of the proposed SPLINER approach against existing non-intrusive PC approaches. All relevant computations are performed using MATLAB 2013b while the deterministic transient simulations are performed using HSPICE [3]. In particular, the transmission line networks of the presented examples are modeled using the W-element transmission line model provided by HSPICE which can consider frequency dependent per-unit-length parameters [3]. The above simulations are run on a workstation with 8 GB RAM, 500 GB memory and an Intel i5 processor with 3.4 GHz clock speed.

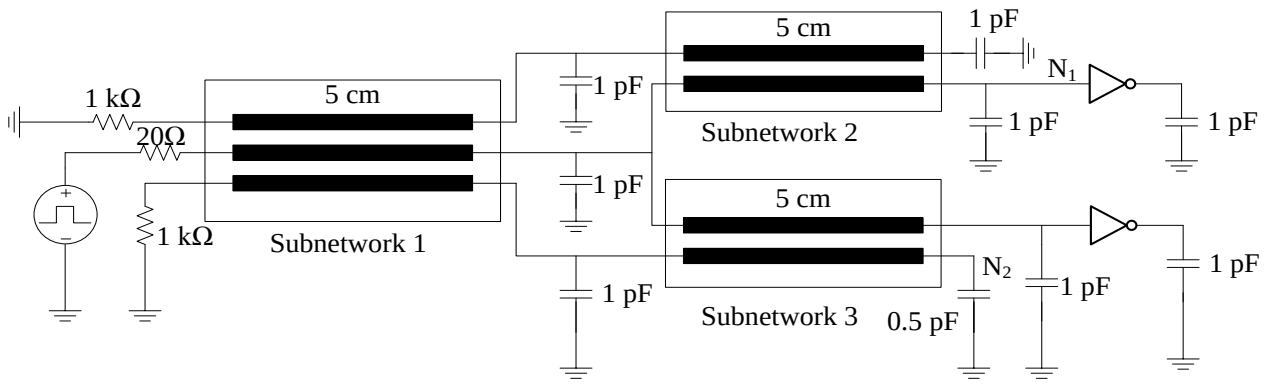
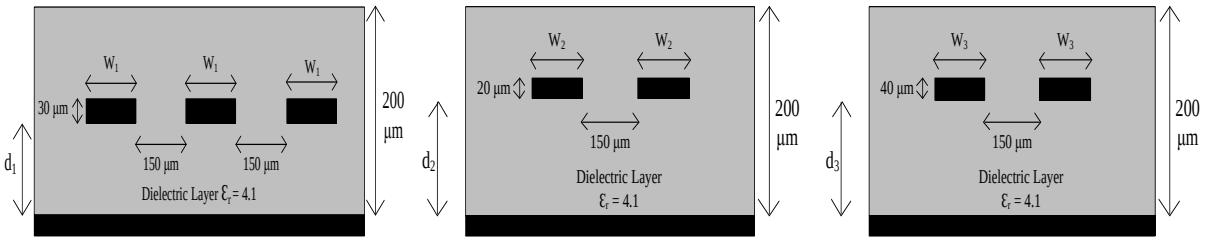


Fig. 3.1: Multiconductor transmission line (MTL) network for Example 1 in Section 3.2.4.1.



(a) (b) (c)

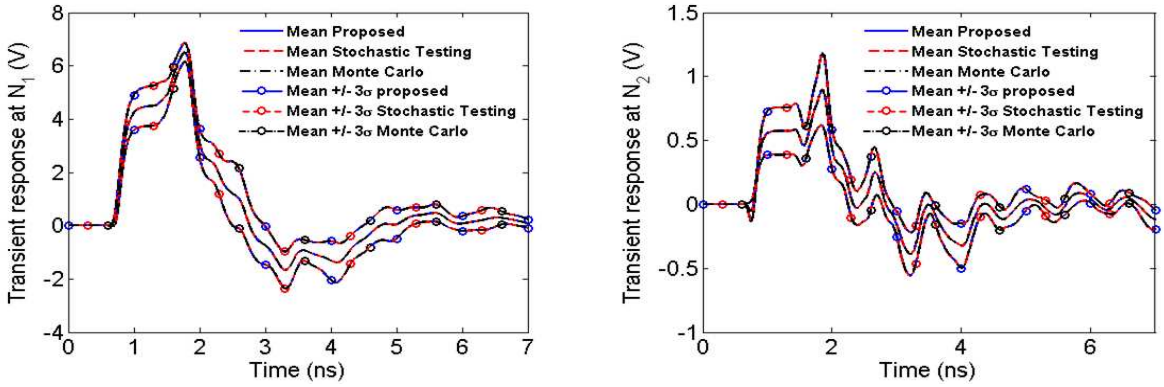
Fig. 3.2: Geometrical and physical layout of transmission lines of Example 1 in Section 3.2.4. (a) Subnetwork 1. (b) Subnetwork 2. (c) Subnetwork 3

3.2.4.1 Example 1

The objective of this example is to compare the performance of the proposed SPLINER approach with the nonintrusive ST based approach [37], and the conventional linear regression approach [27], [28]. For this purpose, the multiconductor transmission line (MTL) network of Fig. 3.1 terminated by inverters made up of SPICE level 49 CMOS transistor models is considered. The lengths of the MTL networks are set to 5 cm and their layout and geometric dimensions are shown in Fig. 3.2. This network is driven by a voltage source with a trapezoidal waveform of rise/fall time T_r = 0.1 ns, pulse width T_w = 1 ns and amplitude of 5V. The uncertainty in the network is introduced via six random variables (n = 6) whose characteristics are listed in Table 3.1 and a Legendre PC expansion of degree m = 3 is considered.

Table 3.1 Characteristics of random variables of example 1 in Section 3.2.4 (Fig. 3.1)

Random Variables	Mean	% Standard Deviation (Uniform Distribution)
d_1	100 μm	+/- 20 %
d_2	140 μm	
d_3	70 μm	
W_1	150 μm	
W_2	130 μm	
W_3	170 μm	



(a) (b)

Fig. 3.3: Comparison of the statistics of the transient response of Example 1 in Section 3.2.4 obtained using the proposed SPLINER approach, the nonintrusive ST based approach, and the Monte Carlo approach. (a) Statistical results of the transient response at N_1. (b) Statistical results of the transient response at N_2.

In order to demonstrate the accuracy of the proposed SPLINER approach, the mean and standard deviation (σ) of the transient responses at the output nodes N_1 and N_2 of Fig. 3.1 are computed using the SPLINER approach where a total of P+1 = 84 regression nodes are required. These results are compared against those obtained using the nonintrusive ST based approach [37] and the brute-force Monte Carlo approach where 20,000 samples are considered. The comparison of the above results is shown in Fig. 3.3 where both PC approaches exhibit good agreement with the Monte Carlo approach. Furthermore, the SPLINER approach exhibits good agreement with the Monte Carlo approach when comparing the probability distribution function of the transient response at node N_2 evaluated at the time point of maximum mean crosstalk (t = 1.83 ns) in Fig. 3.4.

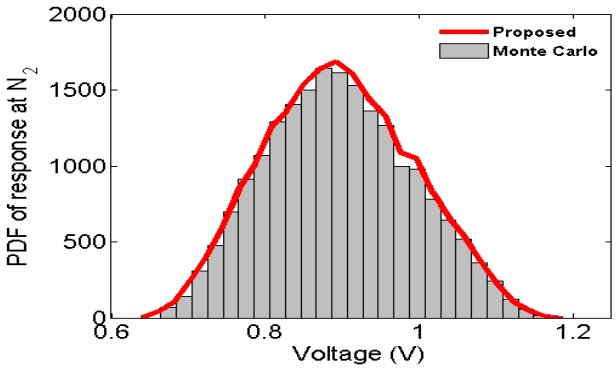


Fig. 3.4: Probability distribution at the time point of maximum mean crosstalk (t = 1.83 ns) obtained using 20,000 samples of Example 1 in Section 3.2.4.

Next, the third order statistical moment (skew) at the output node N_1 of Fig. 3.1 is computed using the proposed SPLINER approach and the nonintrusive ST based approach [37]. These results are compared against the Monte Carlo results as shown in Fig. 3.5. It is worth noting in this example the classical Fedorov (2(P+1) nodes) approach exhibits near-perfect agreement with Monte Carlo. More importantly, it is observed from Fig. 3.5 that the SPLINER approach is more accurate than the nonintrusive ST based approach. This is reflected in the fact that the SPLINER approach incurs an L_2 error norm of $1.27\text{x}10^{-6}$ which is nearly 4 times as small as that incurred by the nonintrusive ST based approach (L_2 error norm of $5.04\text{x}10^{-6}$) when compared against Monte Carlo.

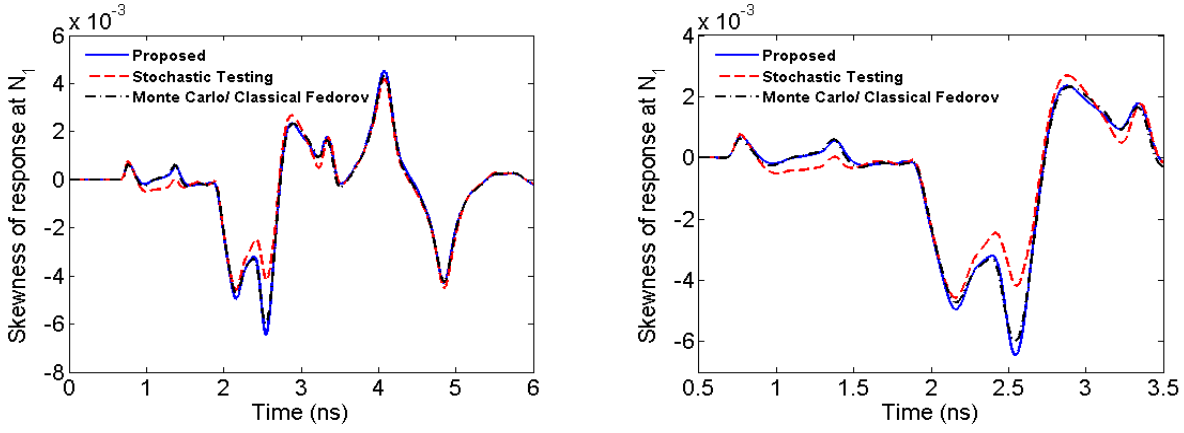


Fig. 3.5: Accuracy comparison between the proposed SPLINER and the nonintrusive ST based approach for Example 1 in Section 3.2.4. (a) Third statistical moment (skew) of the transient response of node N_1. (b) Enlarged plot showing the relative error of the nonintrusive ST based approach.

Finally, the CPU times taken by the SPINER approach, the nonintrusive ST based approach, and the conventional linear regression approach are listed in Table 3.2. It is observed from Table 3.2 that both the SPLINER and the nonintrusive ST based approach incur very similar overall CPU time costs although the nonintrusive ST approach can identify the nodes faster. This is due to the fact that the SPICE simulation costs dominate over the cost of the search algorithm as noted in Section 3.2.3. On the other hand, the SPLINER approach is more than 2 times faster than the conventional linear regression approach. This speedup is due to both the efficiency provided by the modified Fedorov algorithm as well as the smaller number of SPICE simulations involved.

Table 3.2 CPU Time costs using SPLINER, conventional linear regression and stochastic testing for example 1 in Section 3.2.4.

Number of RVs	CPU Time (SPLINER)		CPU Time (Conventional Linear Regression)		CPU Time (Stochastic Testing [33])	
	Modified Fedorov (s)	SPICE Simulations (s)	Classical Fedorov (s)	SPICE Simulations (s)	Stochastic Testing (s)	SPICE Simulations (s)
6	1.5	56.28	14	112.56	0.012	56.28

3.2.4.2 Example 2

The objective of this example is to compare the performance of the proposed SPLINER approach with the ST based approach [37], and the conventional linear regression approach [27], [28] for an RF amplifier. For this purpose, network of Fig. 3.6 is considered [41]. This LNA utilizes an NXP BFG425W wideband BJT, which is represented as a level-1 (Gummel-Poon) SPICE model. This network is driven by a voltage source with a sinusoidal waveform of frequency 2 GHz and amplitude of 1V. The supply voltage of the network is set to V_s = 4.5 V. The uncertainty in the network is introduced via eight random variables (n = 8). These random variables are the first eight variables (out of the total 12 variables) whose characteristics are listed in Table 3.3. A Hermite PC expansion of degree m = 3 is considered for this example.

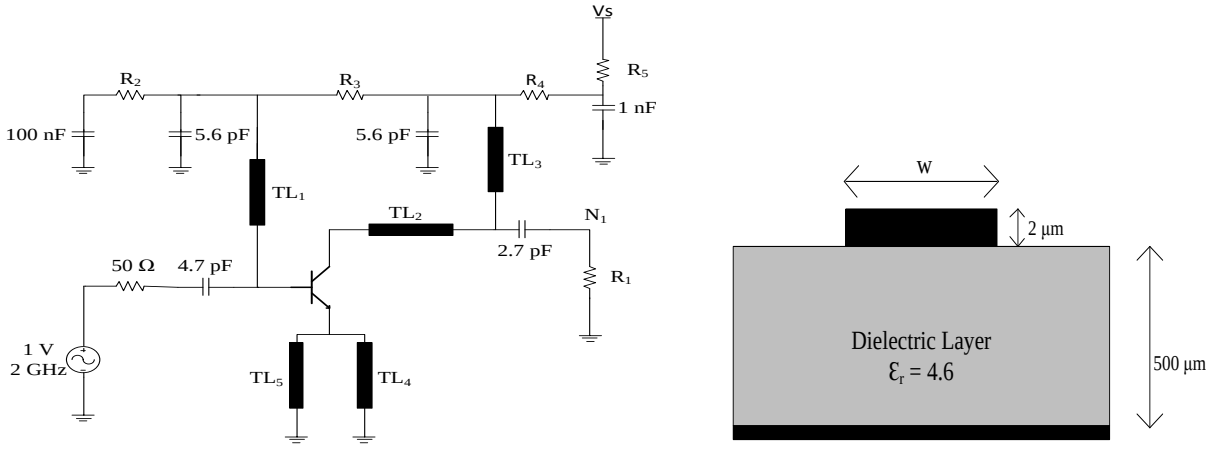
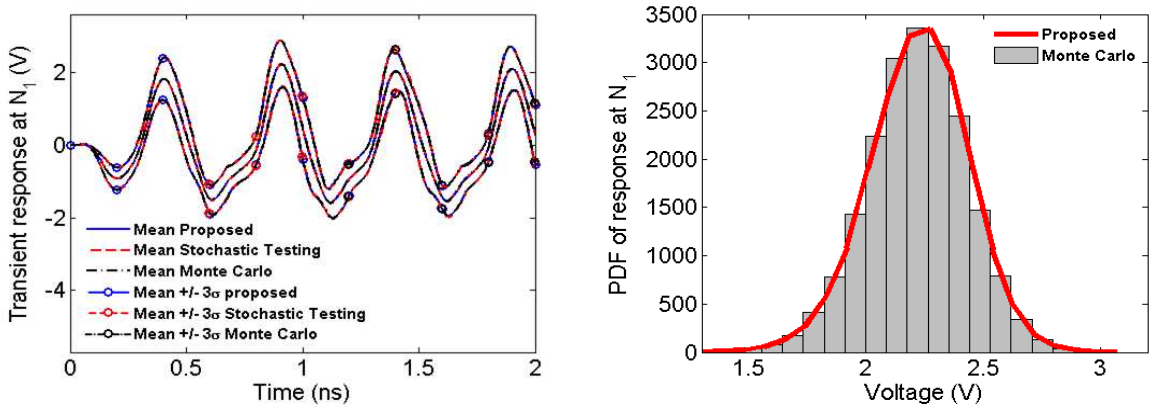


Fig. 3.6: LNA network of Example 2 in Section 3.2.4. (a) Circuit schematic. (b) Cross-section view of the transmission lines.

Table 3.3 Characteristics of random variables of example 2 and 3 in Section 3.2.4 (Fig. 3.6)

Random Variables	Mean	% Standard Deviation (Normal Distribution)
w_1	200 µm	+/- 20 %
w_2	250 µm	
w_3	300 µm	
w_4	700 µm	
w_5	900 µm	
B_f	145	
C_{js}	667.5 fF	
R_1	50 Ω	
R_2	100 Ω	
R_3	15 KΩ	
R_4	22 Ω	
R_5	84 Ω	

In order to demonstrate the accuracy of the proposed SPLINER approach, the mean and standard deviation (σ) of the transient response at the output node N_1 of Fig. 6 is computed using the SPLINER approach where a total of P+1 = 165 regression nodes are required. These results are compared against those obtained from the nonintrusive ST based approach [37] and the Monte Carlo approach using 20,000 samples in Fig. 3.7(a). It is observed from Fig. 3.7(a) that both PC approaches exhibit good agreement with the Monte Carlo approach. In addition, , the SPLINER approach exhibits good agreement with the Monte Carlo approach for the probability distribution of the transient response at node N_1 evaluated at the time point of maximum standard deviation (t = 0.91 ns) as shown in Fig. 3.7(b).



(a) (b)

Fig. 3.7: Comparison of the statistics of the transient response of Example 2 in Section 3.2.4 obtained using the proposed SPLINER approach, the ST based approach, and the Monte Carlo approach. (a) Statistical results of the transient response at N_1. (b) Probability distribution of the response at N_1 at the time point of maximum standard deviation (t = 0.91 ns) using 20,000 samples.

Next, the third order statistical moment (skew) at the output node N_1 of Fig. 3.6 is computed using the proposed SPLINER approach and the nonintrusive ST based approach [37]. Both these results are compared against the Monte Carlo results as shown in Fig. 3.8. Similar to Example 1, even for this example the SPLINER approach is found to be more accurate than the ST based approach. This is reflected in the fact that the SPLINER approach incurs an L_2 error norm of 1.07×10^{-5} which is nearly 3 times as small as that incurred by the nonintrusive ST based approach (L_2 error norm of 3.23×10^{-5}) when compared against Monte Carlo.

Finally, the CPU times taken by the SPLINER approach, the nonintrusive ST based approach, and the conventional linear regression approach are listed in Table 3.4. It is observed from Table 3.4 that both the SPLINER and the nonintrusive ST based approach incur very similar overall CPU time costs although once again the nonintrusive ST approach can identify the nodes far faster. This is as expected from the discussion in Section 3.2.3. Additionally, for this example the SPLINER approach is roughly 18 times faster than the conventional linear regression approach.

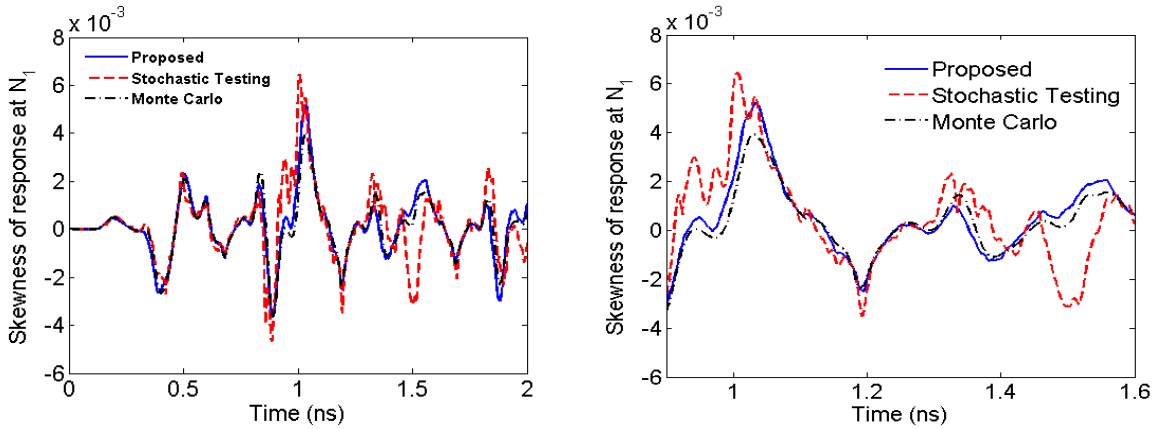


Fig. 3.8: Accuracy comparison between the proposed SPLINER and the nonintrusive ST based approach for Example 2 in Section 3.2.4. (a) Third statistical moment (skew) of the transient response of node N1. (b) Enlarged plot showing the relative error of the nonintrusive ST based approach.

Table 3.4 CPU time costs using SPLINER, conventional linear regression and nonintrusive ST for example 2 in Section 3.2.4

Number of RVs	CPU Time (SPLINER)		CPU Time (Conventional Linear Regression)		CPU Time (Stochastic Testing [33])	
	Modified Fedorov (s)	SPICE Simulations (s)	Classical Fedorov (s)	SPICE Simulations (s)	Stochastic Testing (s)	SPICE Simulations (s)
8	13	38	838	76	0.031	38

3.2.4.3 Example 3

The objective of this example is to compare the scalability of the proposed SPLINER approach against that of conventional non-intrusive PC approaches. For this purpose the same RF low noise amplifier (LNA) network of Fig. 3.6 is considered [41]. The uncertainty in the network is expanded to include twelve random variables (n = 12) whose characteristics are listed in Table 3.3 and a Hermite PC expansion of degree m = 3 is considered.

In order to demonstrate the scalability of the proposed work the number of random dimensions is progressively increased from 3 to 12 as shown in Table 3.5. For the same test cases of Table 3.5, the total PC problem is solved using three methods – the proposed SPLINER approach, the conventional linear regression approach [27], [28], and the nonintrusive ST based approach [37]. The CPU time incurred by each approach for each test case is noted in Table 3.5 and scaling of the overall CPU costs is demonstrated in Fig. 3.9. It is observed from Fig. 3.9 that the SPLINER and the nonintrusive ST based

approach are significantly more efficient than the conventional linear regression approach. This is due to both the relatively slow classical Fedorov algorithm as well as the fact that conventional linear regression approach requires twice as many SPICE simulations as the other approaches. Moreover, while the ST based approach is much faster than the modified Fedorov algorithm in locating the nodes (see Table 3.5), due to the very large SPICE simulation costs, their overall CPU costs turn out to be very similar. For example, when considering n = 12, the nonintrusive ST approach is nearly 118 times faster than the modified Fedorov algorithm, although after taking the SPICE simulation into account this speedup reduced to less than 2 times. Besides, the improved efficiency of the nonintrusive ST based approach is counterbalanced by the lower accuracy as demonstrated in the previous examples. These results are as expected from the discussion of Section 3.2.3.

Table 3.5 Scaling of CPU time costs using proposed, conventional linear regression and nonintrusive ST for example 3 in Section 3.2.4

Number of RVs (Random Variables)	CPU Time (SPLINER)		CPU Time (Conventional Linear Regression)		CPU Time (Stochastic Testing [33])	
	Modified Fedorov (s)	SPICE Simulations (s)	Classical Fedorov (s)	SPICE Simulations (s)	Stochastic Testing (s)	SPICE Simulations (s)
3 (w_1, w_2, w_3)	0.014	4.60	0.016	9.20	0.007	4.60
4 (w_1, w_2, w_3, w_4)	0.18	8.10	0.69	16	0.007	8.10
5 (w_1, w_2, w_3, w_4, w_5)	0.52	13	2.40	26	0.011	13
6 (w_1, w_2, w_3, w_4, w_5, B_f)	1.50	19	14	38	0.012	19
7 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js})	3.40	28	87	56	0.017	28
8 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_1)	13	38	838	76	0.031	38
9 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_1, R_2)	30	51	3031	102	0.069	51
10 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_1, R_2, R_3)	52	66	7432	132	0.179	66
11 w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_1, R_2, R_3, R_4)	76	84	14803	168	0.417	84
12 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_1, R_2, R_3, R_4, R_5)	99	105	25911	210	0.839	105

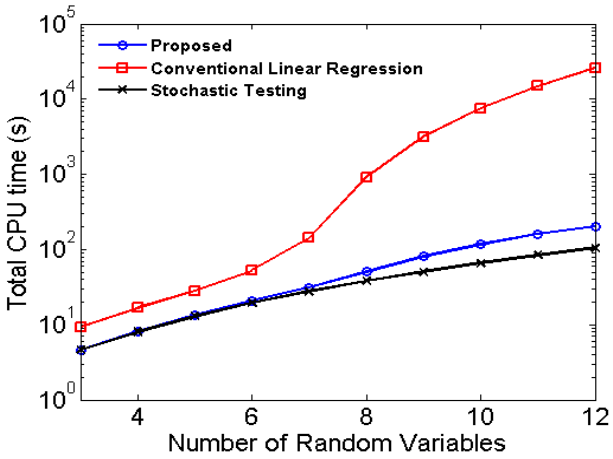


Fig. 3.9: Scaling of overall CPU time costs for SPLINER, conventional linear regression, and the nonintrusive ST based approach with increasing number of random variables for Example 3 in Section 3.2.4.

3.3 Additional enhancements to the linear regression approach

This section proposes another novel approach for improving the linear regression method. It begins by the numerical strategies to accelerate the search algorithm for problems involving high-dimensional random spaces. Then a detailed analysis of the improvements provided by the said strategies over the original search algorithm presented in [39] is provided. Finally, this section concludes with numerical examples in order to validate the proposed approach and study its computational cost scaling rate with respect to number of random variables.

3.3.1 Expediting the search algorithm for high dimensional random spaces

In this section, two modifications to the original Fedorov search algorithm are presented. This approach is applied to the original linear regression approach with $2(P+1)$ sample nodes [27], [28], in order to observe the individual improvement caused by the suggested strategies.

3.3.1.1 Substituting the *K* worst DoE

This strategy is based on the rationale that once a reasonably large determinant of the information matrix has been reached, any further enrichment of the determinant will translate to only marginal improvement in the accuracy of the evaluated PC coefficients. Thus, instead of substituting all M DoE (as

suggested previously), in this strategy only the K worst DoE in the starting set will be identified and substituted. The substitution of the K worst DoE will result in a sufficiently large increase of the determinant of the information matrix, thereby eliminating the need for exchanging the remaining $M-K$ DoE. Thus, this strategy will reduce the number of searches from $M((m+1)^n-M)$ searches to $K((m+1)^n-M)$ – a reduction of the search cost (C_a) of (3.9) by a factor of M/K.

In this work, K is initially set to $[M/5]$ where [.] is the ceiling function. Next, from (3.5) it is noted that the depreciation in the value of the determinant caused by removing the r^{th} DoE ($\boldsymbol{\lambda}^{(r)}$) is proportional to the term

$$d_{rr} = \mathbf{R}(\boldsymbol{\lambda}^{(r)})\boldsymbol{\Psi}^{(0)}\mathbf{R}(\boldsymbol{\lambda}^{(r)})^t \tag{3.17}$$

where $\boldsymbol{\Psi}^{(0)}$ is the inverse of the original information matrix consisting of the starting M DoE. Thus, the K worst DoE are identified as those DoE in the starting set that have the smallest possible value of the scalar d_{rr}. It is appreciated that computation of the d_{rr} term of all $2(P+1)$ DoE can be performed cheaply since the matrix inverse $\boldsymbol{\Psi}^{(0)}$ needs to be computed only once. Once the K worst DoE have been identified, the Fedorov search algorithm is run for only these DoE. Thereafter, a check is made to ascertain if the determinant of the information matrix $\mathbf{A}^T\mathbf{A}$ is reasonably high. If not, then the next worst DoE (i.e. $K+1^{\text{th}}$ worst DoE) is identified using (3.17) and substituted as before. This sequential process continues until the determinant of the information matrix $\mathbf{A}^T\mathbf{A}$ is deemed to be sufficiently large. It is observed from numerous examples that $K = [M/5]$ is a good starting guess for K and rarely does this value need to be increased further.

3.3.1.2 Implicit matrix inversion

Although the above strategy will reduce the number of searches, and consequently, the number of matrix inversions, given that the cost to directly invert the information matrix even once scales in a near-exponential manner with the number of random dimensions (see (3.11)), the overall cost of matrix inversions may still remain prohibitively large for high-dimensional random spaces.

In this section, a new strategy is adopted whereby $\boldsymbol{\Psi}^{(0)}$ (i.e. inverse of the information matrix for the starting set of DoE) is computed once and stored. Thereafter, the substitution of any r^{th} DoE ($\boldsymbol{\lambda}^{(r)}$) ($r < K$) will change the information matrix. The inverse of the new information matrix (i.e. $\boldsymbol{\Psi}^{(r)}$ of (3.7)) will now be expressed as a P+1 rank correction to the previous inverse $\boldsymbol{\Psi}^{(r-1)}$ using the Sherman-Morrison-Woodbury formula as [58].

$$\begin{aligned}\boldsymbol{\Psi}^{(r)} &= ((\mathbf{A}^t\mathbf{A}) + \mathbf{R}^t(\lambda^{(k)})\mathbf{R}(\lambda^{(k)}) - \mathbf{R}^t(\lambda^{(r)})\mathbf{R}(\lambda^{(r)}))^{-1} \\ &= \boldsymbol{\Psi}^{(r-1)} + w_k\mathbf{V}_k^t\mathbf{V}_k - w_r\mathbf{U}_r^t\mathbf{U}_r\end{aligned} \tag{3.18}$$

where

$$\begin{aligned} w_k &= \frac{-1}{(1+\mathbf{R}(\boldsymbol{\lambda}^{(k)})\mathbf{V}_k)}; \quad \mathbf{V}_k = \boldsymbol{\Psi}^{(r-1)}\mathbf{R}^t(\boldsymbol{\lambda}^{(k)}) \\ w_r &= \frac{1}{(1-\mathbf{R}(\boldsymbol{\lambda}^{(r)})\mathbf{U}_k)}; \quad \mathbf{U}_k = \left(\boldsymbol{\Psi}^{(r-1)} + w_k\mathbf{V}_k^t\mathbf{V}_k\right)\mathbf{R}^t(\boldsymbol{\lambda}^{(r)}) \end{aligned} \tag{3.19}$$

Based on the recursive expressions of (3.18) and (3.19), it is observed that for the substitution of any r^{th} DoE ($\boldsymbol{\lambda}^{(r)}$) ($r < K$) the new inverse $\boldsymbol{\Psi}^{(r)}$ will be updated efficiently using numerically cheap matrix-vector and vector-vector multiplications as opposed to the direct matrix inversions.

3.3.1.3 Numerical efficiency of the modified search algorithm

In this section numerical efficiency of the proposed approach is discussed where only effect of substituting the K worst DoE and implicit matrix inversion on the classical linear regression are considered and improvements of the SPLINER approach are not considered. It is emphasized that the modified search algorithm described above offers two clear benefits. Firstly, the number of searches will decrease from $M((m+1)^n\text{-}M)$ searches to $K((m+1)^n\text{-}M)$ where $K = [M/5]$. This automatically reduces the search cost of (3.9) approximately by a factor of 5 to

$$C_a \approx K((m+1)^n - 2(P+1))C_1 = \left[\frac{2(P+1)}{5}\right](m+1)^n C_1 \tag{3.20}$$

Secondly, the CPU cost of performing the matrix inversions for the modified search algorithm will also decrease significantly as quantified using the following lemma.

Lemma 2: Utilization of the Sherman-Morrison-Woodbury formula of (3.18) and (3.19) will ensure that the total CPU costs to perform the matrix inversions in the modified search algorithm will scale as $O((P+1)^3) \approx O(n^{3m})$ with respect to the number of random dimensions (n).

Proof: Based on (3.19) it is noted that the main computations required in order to evaluate $\mathbf{V}_k$ is a matrix-vector multiplication of dimensions P+1 where $\mathbf{\Psi}^{(r-1)}$ is assumed to be known. The cost of this operation will be $C_3 = k(P+1)^2$ where k is the cost of each floating point operation. Next, to compute the matrix $\mathbf{V}_k^t\mathbf{V}_k$ a vector-vector multiplication of dimensions P+1 is required. This will incur an additional CPU cost of $C_4 = k(P+1)^2$. Finally, to compute the denominator of the scalar term w_k, another matrix-vector multiplication of dimensions P+1 will be required at the cost $C_3 = k(P+1)^2$. Thus, the overall CPU cost to evaluate the second term in (3.18) (i.e. $w_k \mathbf{V}_k^t\mathbf{V}_k$) will be

$$C_3 + C_4 + C_3 = 3k(P+1)^2 \tag{3.21}$$

It is observed that computing the third term in (3.18) (i.e. $w_r \mathbf{U}_r^t\mathbf{U}_r$) proceeds exactly in the same way as that of the second term. Hence, the associated cost too will equal to that of (3.21). Adding all the above CPU costs for K substitutions along with the cost of directly inverting the starting information matrix (i.e. computing $\mathbf{\Psi}^{(0)}$), the total CPU cost incurred to perform the matrix inversions in the modified search algorithm can be quantified as

$$C_b = C_2 + 6k\,K(P+1)^2 \tag{3.22}$$

It is observed that for K = [M/5] both the first and the second terms of (3.22) will scale as $O((P+1)^3) \approx O(n^{3m})$ with respect to the number of random dimensions (n). Thus, it is concluded that the overall costs of performing the matrix inversions for the modified search algorithm will also scale as $O((P+1)^3)$

$\approx O(n^{3m})$. Comparing this result with that of (3.11) reveals a distinct improvement in the CPU cost to the order of $O(n)$ – a major numerical benefit for high-dimensional problems (i.e. for large n).

Finally, it is remarked that the cost of identifying the K worst nodes will require the additional computation of the scalar d_{rr} in (3.17) for the entire $2(P+1)$ starting DoE. This cost can be expressed as

$$C_c = 2(P+1)\frac{C_1}{3} \tag{3.23}$$

where C_1 is defined in (3.10). Comparing (3.20) and (3.23) it is evident that the cost of identifying the K worst DoE is a negligible fraction of the search cost C_a and can be safely ignored. Overall, it is appreciated that the proposed modified search algorithm will be able to significantly reduce both the search cost C_a of (3.9) and the cost of the matrix inversions C_b of (3.11), thereby substantially accelerating the original search algorithm of Section 3.1.2. This benefit will be validated in the numerical examples section.

3.3.2 Comparative analysis of overall CPU costs

In this section, the CPU cost of the proposed D-optimal linear regression approach is compared against that of conventional non-intrusive PC approaches. Comparison with the intrusive SG approach is not included since the relative efficiency of non-intrusive approaches over the SG approach is well-documented in the literature [26], [46]. This is particularly true for nonlinear circuits where the multidimensional integral of (2.37) using the SG approach have to be represented in SPICE using massive number of additional voltage/current dependent sources (also demonstrated in Example 2 of Section 3.3.3.2).

3.3.2.1 Proposed linear regression approach

Based on the discussion of Section 3.3.1, it is appreciated that the CPU cost required for the proposed linear regression approach can be divided into two parts – the cost incurred by the modified

search algorithm and the cost incurred to perform the M deterministic SPICE simulations to extract the **E** matrix of (3.2). The cost of the modified search algorithm is expressed using (3.20) and (3.22) as

$$C_t \approx K(m+1)^n C_1 + C_2 + 6k\,K(P+1)^2 \tag{3.24}$$

where typically $K = [M/5]$. As for the SPICE simulation cost, it is assumed that each of the M simulations requires the same CPU cost which is a reasonable assumption since the variation in the unknowns of the MNA equations of (2.38) from one DoE to another will be typically small. Thus, the SPICE simulation cost can be quantified as

$$C_S = 2(P+1)C_0 \tag{3.25}$$

where C_0 is the cost of each deterministic SPICE simulation. Thus, the overall cost of the SPICE simulations scales as $O(2n^m)$ with respect to the number of random dimensions (n). It is pointed out that the substantial numerical advantages of this proposed algorithm over the work of [36] lies in the acceleration of the modified search algorithm as explained in details in Section 3.3.1.3.

3.3.2.2 Other approaches

The proposed linear regression approach has some interesting features compared to the stochastic testing algorithm of [40], [41]. Firstly, the modified search algorithm allows only K substitutions as opposed to the significantly larger P+1 substitutions required by the stochastic testing algorithm. Further, the proposed linear regression approach can directly utilize the SPICE results of the M DoE without the need of any intrusive coding or access to the internals of the SPICE engine, thereby making the proposed approach truly non-intrusive in nature. These benefits are offset by the fact that the stochastic testing algorithm requires only P+1 sample nodes as opposed to the 2(P+1) sample nodes required by the proposed D-optimal linear regression approach. However, by combining this approach with the one in Section 3.2, this issue can be addressed too. However, it is assumed the proposed D-optimal approach is applied to the conventional linear regression approach of Section 2.3.2.

Furthermore, stochastic collocation (SC) has been a very popular nonintrusive PC approach [32], [33], [45], [46]. In this approach, if the non-intrusive multidimensional nodes are selected to be the full tensor product of 1D quadrature nodes, then $M = (m+1)^n$. These nodes can be analytically identified at negligible computational costs (i.e. $C_t = 0$). Thus, the cumulative costs of the entire SC approach is equal to that of the SPICE simulations and is expressed as

$$C_S = (m+1)^n C_0 \tag{3.26}$$

This corresponds to an exponential scaling of the time costs with respect to the number of random dimensions (n), quantified as $O((m+1)^n)$. This means that for even moderate dimensional problems, the massive cost of SPICE simulations in (3.26) will make this approach highly cost intensive compared to the proposed linear regression approach.

In order to mitigate this prohibitive scaling, an intelligent choice of only a sparse subset of the tensor product nodes guided by the Smolyak algorithm has been proposed [32], [33], [45], [69]. Once again, this method allows the fast identification of the sparse nodes (i.e. $C_t = 0$ compared to the proposed linear regression approach). This approach results in a decrease in the number of multidimensional nodes from $M = (m+1)^n$ to approximately $M = (2n)^m/m!$, thereby improving the CPU time costs of the SC algorithm from that of (3.26) to

$$C_S = \frac{(2n)^m}{m!} C_0 \tag{3.27}$$

For this approach, it is observed that the number of deterministic SPICE simulations required scales as $O(2^m n^m)$ which is still 2^{m-1} times more than that required for the proposed linear regression approach (see (3.25)). Thus, for large variation in the random dimensions requiring high degrees of PC expansion (m), the proposed linear regression approach may still be more cost effective than even this sparse collocation approach.

Among other existing non-intrusive approaches, the pseudo-spectral collocation has been recently reported for full-wave EM problems [29]. However, this approach suffers from the same exponential scaling of the SPICE simulation costs as the classical SC approach. Other methods based on the Stroud low order cubature methods has also recently been explored for packaging problems [26], [34], [36], [38]. This approach can easily locate the multidimensional nodes using simple analytic formulas and exhibits only a linear scaling of the number of SPICE simulations with number of random dimensions (i.e. O(n)). However, this excellent scaling with the number of random dimensions only exists for a second and third degree PC expansion and cannot be extended to higher degree expansions [26].

Finally, we remark the promising non-intrusive stochastic testing based approach which has been proposed in [37]. In this approach the selection of the non-intrusive nodes is determined exactly as proposed in the stochastic testing approach of [40]. As a result, this approach too relies on the costly P+1 substitutions as opposed to the relative smaller K substitutions used in the modified search algorithm. On the other hand, this approach requires only P+1 SPICE simulations as opposed to the 2(P+1) simulations required by the proposed D-optimal linear regression approach. However, this decrease in number of SPICE simulations comes at the cost of loss of accuracy [61].

From the above analysis, it is observed that the proposed linear regression approach offers clear benefits over the state-of-the-art non-intrusive PC approaches and this is validated through multiple lumped and distributed microwave network examples in the next section.

3.3.3 Numerical examples

In this section, three examples are presented to compare the accuracy and scalability of the proposed D-optimal linear regression approach against existing intrusive and non-intrusive PC approaches. All relevant PC computations are performed using MATLAB 2013b while the deterministic transient simulations, whether using intrusive or non-intrusive PC approaches, are performed using HSPICE [3]. In particular, for Example 2 and 3 the transmission line networks are modeled using the W-

element transmission line model provided by HSPICE which can automatically consider frequency dependent per-unit-length parameters [3]. The above simulations are run on a workstation with 8 GB RAM, 500 GB memory and an Intel i5 processor with 3.4 GHz clock speed.

3.3.3.1 Example 1

The objective of this example is to demonstrate the accuracy of the proposed linear regression approach. For this purpose, the RF low noise amplifier (LNA) network of Fig. 3.10 comprising of three SPICE level-49 CMOS transistor models are considered. The RF input to the network is a sinusoidal wave with amplitude of 1V and a frequency of 1 GHz. The uncertainty in the network is introduced via six normally distributed random variables (n = 6) whose characteristics are listed in Table 3.6. A Hermite PC expansion of degree m = 3 is required for this example.

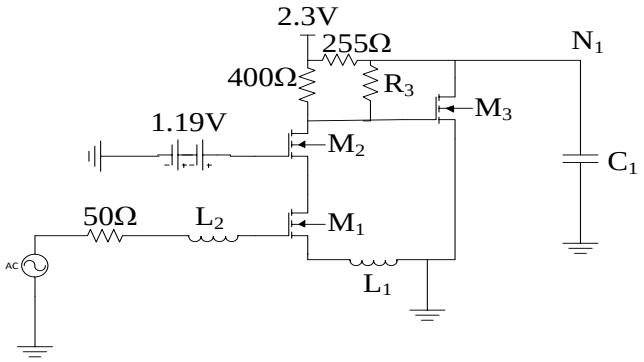


Fig. 3.10: Circuit schematic of the CMOS low-noise amplifier network of Example 1 in Section 3.3.3.

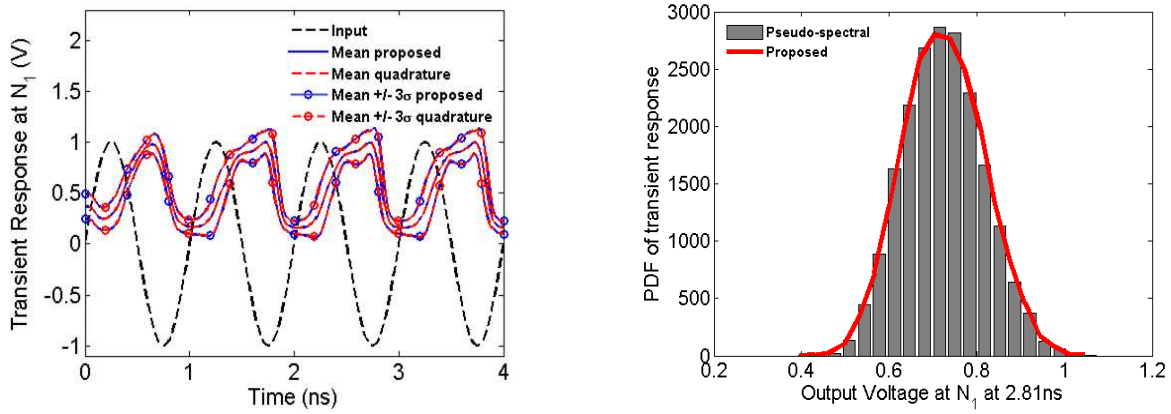
Table 3.6 Characteristics of random variables of example 1 in Section 3.3.3 (Fig. 3.10)

Random Variables	Mean	% Standard Deviation (Normal Distribution)
w_1 (width of M_1)	7.5µm	+/- 10 %
w_2 (width of M_2)	7.5µm	
w_3 (width of M_3)	7.5µm	
L_1	13 nH	
L_2	0.9 nH	
R_3	120 Ω	

In order to evaluate the accuracy of the proposed approach, the mean and standard deviation (σ) of the transient response at the output node N_1 of Fig. 3.10 is computed using two methods – the proposed

D-optimal linear regression approach described in Section 3.3.1 and the pseudo-spectral collocation approach based on Gauss-Hermite quadrature techniques [29]. For the proposed approach, only K = [2(P+1)/5] = 33 substitutions are performed as described in Section 3.3.1. The comparison of the above results is shown in Fig. 3.11(a) where the proposed linear regression approach is found to exhibit good agreement with the pseudo-spectral collocation approach.

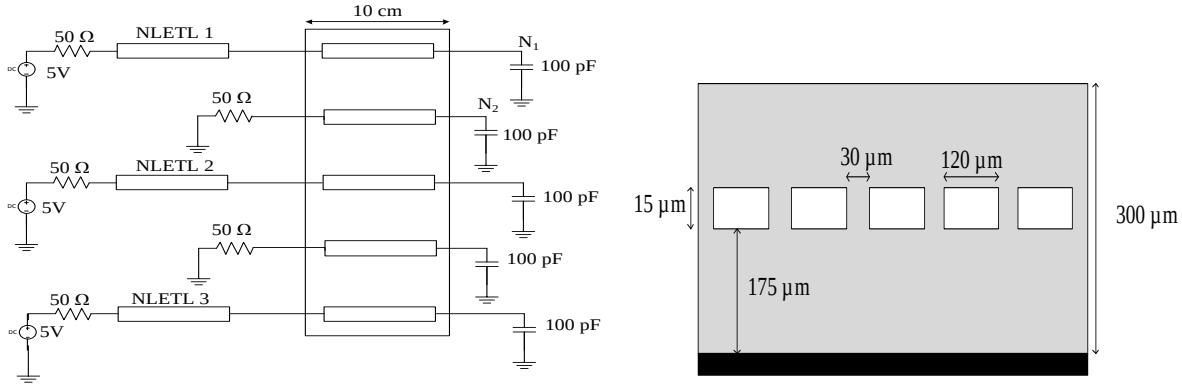
Next, in order to test the accuracy for higher order statistical moments, the probability distribution function of the transient response at node N_1 evaluated at the time point of maximum standard deviation (t = 2.81 ns) is computed using the above two approaches and the results are displayed in Fig. 3.11(b). As expected, the probability distribution results for 20,000 analytically generated samples exhibit good agreement demonstrating the accuracy of the proposed linear regression approach.



(a) (b)

Fig. 3.11: Comparison of mean, standard deviation and PDF of the transient response of Example 1 in Section 3.3.3 computed using the proposed linear regression approach and pseudo-spectral collocation approach. (a) Input, mean and statistical corners (+/- 3σ) of the transient response at N_1. (b) Probability distribution function of transient response at N_1 at the time point of maximum standard deviation (t = 2.81ns) using 20,000 samples.

Finally, it is noted that the proposed approach requires 15 seconds for completing the K substitutions and another 13.44 seconds for the 2(P+1) = 84 SPICE simulations. On the other hand, the pseudo-spectral collocation approach requires 327.68 seconds to perform the necessary 4096 SPICE simulations at the Gauss-Hermite quadrature nodes. This amounts to a speedup of 11.5 provided by the proposed algorithm over the pseudo-spectral collocation approach. This is as expected from the discussion of Section 3.3.2.



(a) (b)

Fig. 3.12: Transmission line network of Example 2 in Section 3.3.3. (a) Circuit schematic. (b) Geometry of coupled transmission lines.

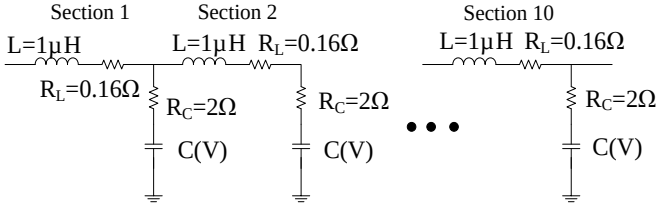


Fig. 3.13: Equivalent lumped RLGC model of nonlinear transmission lines of Fig. 3.12.

3.3.3.2 Example 2

The objective of this example is to compare the performance of the proposed linear regression approach with the existing intrusive SG approach for a large distributed network. For this purpose, the multiconductor stripline network driven by nonlinear transmission lines shown in Fig. 3.12(a) is considered. The layout of the stripline network is illustrated in Fig. 3.12(b). The nonlinear transmission lines are represented using cascaded lumped nonlinear RLGC segments as shown in Fig. 3.13 where the nonlinear capacitance $C(V)$ is represented as

$$C(V) = C_a(b + (1-b)e^{-V/a})) \tag{3.28}$$

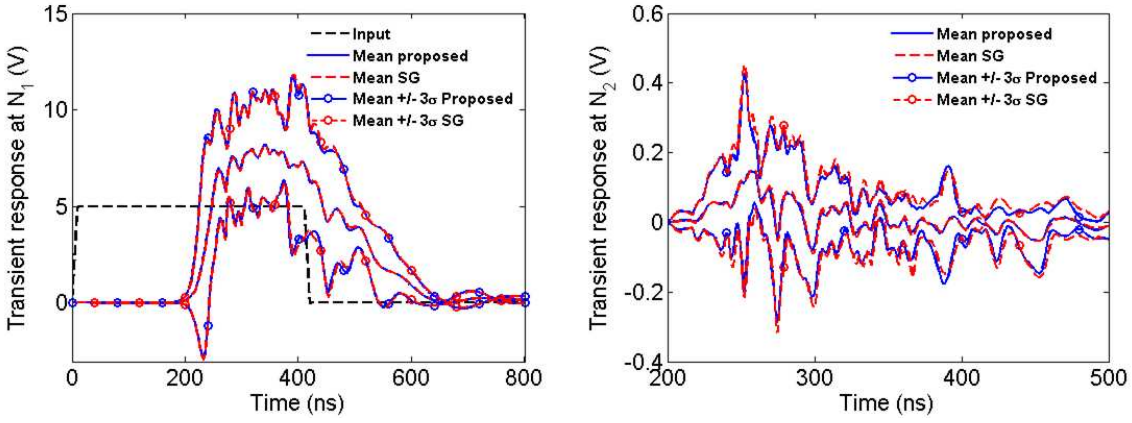
and V is the potential difference across the capacitor [74]. It is noted that (3.28) is a strongly nonlinear function used for benchmarking the proposed approach. For the network in Fig. 3.12(a), 10 nonlinear RLGC segments are considered. The input to the network is a trapezoidal waveform with rise/fall time T_r = 10 ns, pulse width T_w = 400 ns and amplitude equal to 5V. The uncertainty in the network is introduced

via five random variables (n = 5) whose characteristics are listed in Table 3.7. A mixed Legendre-Hermite PC expansion of degree m = 2 is required for this example.

Table 3.7 Characteristics of random variables of example 2 in Section 3.3.3 (Fig. 3.12)

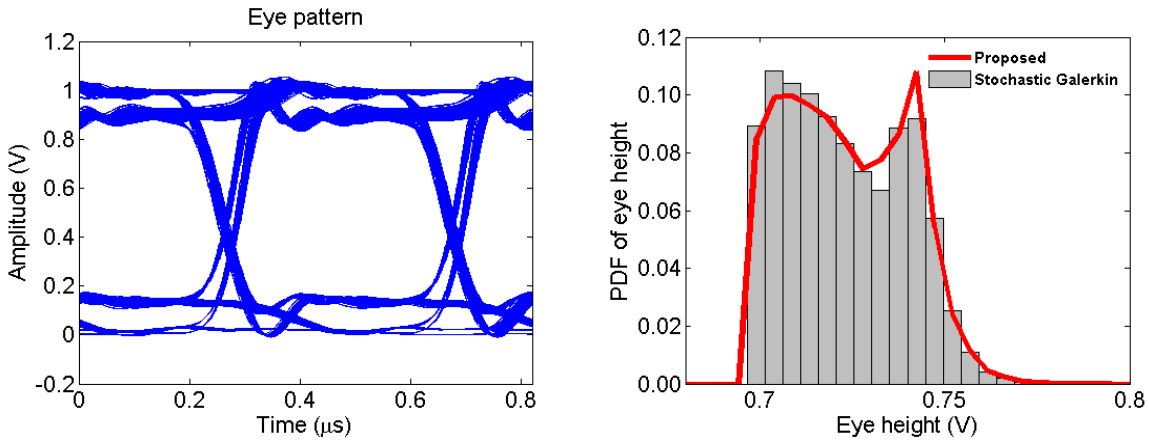
Random Variables	Mean	Distribution	% Relative Variation
"a" parameter in NL capacitors of NLETL 1	2.137	Normal	+/- 10 %
"a" parameter in NL capacitors of NLETL 2	2.637		
"a" parameter in NL capacitors of NLETL 3	1.637		
"b" parameter in NL capacitors of NLETL 1	6.037 e-3	Uniform	
"b" parameter in NL capacitors of NLETL 2	9.108 e-3		

In order to evaluate the accuracy of the proposed approach, the mean and standard deviation (σ) of the transient response at the output nodes N_1 and N_2 of Fig. 3.12(a) is computed using two methods – the proposed D-optimal linear regression approach described in Section 3.3.1 and the SG approach [18]. For the proposed approach, only K = [2(P+1)/5] = 8 substitutions are performed as described in Section 3.3.1. The comparison of the above results is shown in Fig. 3.14 where the proposed linear regression approach is found to exhibit good agreement with the SG approach.



(a) (b)

Fig. 3.14: Comparison of mean and standard deviation of the transient response of Example 2 in Section 3.3.3 computed using the proposed linear regression approach and the stochastic Galerkin approach. (a) Input, mean and statistical corners (+/- 3σ) of the transient response at N_1. (b) Mean and statistical corners (+/- 3σ) of the transient response at N_2.



(a) (b)

Fig. 3.15: Eye diagram comparison for Example 2 in Section 3.3.3 computed using the proposed linear regression approach and the stochastic Galerkin approach. (a) 100 eye diagram samples analytically produced at N_1. (b) Probability distribution function of the eye height.

To further probe the accuracy of the proposed approach, the PC expansion of the voltage response at node N_1 is computed using the same above two methods where the input sources in Fig. 3.12 are changed to random pulse trains with rise/fall time T_r = 10 ns, pulse width T_w = 410 ns and amplitude equal to 1V. From the PC expansion the eye diagram is generated as shown in Fig. 3.15 (a). The probability distribution function of the eye height is then extracted using both the above methods and compared in Fig. 3.15 (b). It is observed from Fig. 3.15 (b) that the proposed linear regression approach matches very well the result of the SG approach.

It is appreciated that the proposed approach requires 1.022 seconds for completing the K substitutions and another 23.94 seconds for the $2(P+1) = 42$ SPICE simulations. On the other hand, the SG approach requires 4722.45 seconds to perform the solitary augmented SPICE simulation. This is because the augmented network is 21 times larger than the original model and includes an additional $(m+1)^n = 243$ companion circuits to model the uncertainty in each nonlinear capacitor [18]. Thus, the proposed approach provides a speedup of roughly 189 over the SG approach. If the size of the network and/or the number of random dimensions is increased, the achieved speedup will be even higher, thereby clearly validating the advantage of the proposed linear regression approach over the intrusive SG approach.

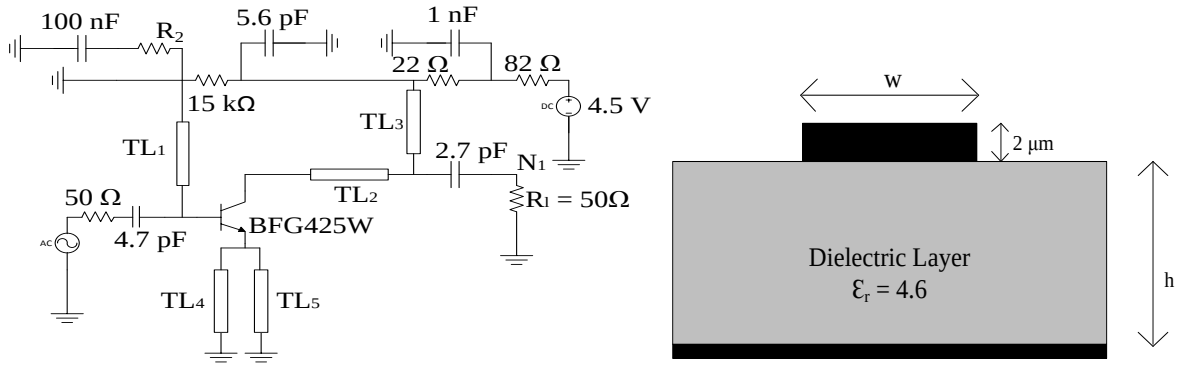


Fig. 3.16: Circuit schematic of the BJT low-noise amplifier network of Example 3 in Section 3.3.3. (a) Circuit schematic. (b) Geometry of transmission lines.

3.3.3.3 Example 3

The objective of this example is to compare the computational complexity of the proposed linear regression approach against that of conventional non-intrusive PC approaches as the number of random dimensions increases. For this purpose the example of Section 3.2.4.2 is considered once again. Because of changes in random variables, the low noise amplifier (LNA) network is presented in Fig. 3.16 again[41]. As a reminder, this LNA utilizes an NXP BFG425W wideband BJT, which is represented as a level-1 (Gummel-Poon) SPICE model. This network is driven by a voltage source with a sinusoidal waveform of frequency 2 GHz and amplitude of 1V. The supply voltage of the network is set to V_s = 4.5 V. The uncertainty in the network is introduced via twelve random variables (n = 12) whose characteristics are listed in Table 3.8 and a Hermite PC expansion of degree m = 4 is considered.

First, the number of random dimensions is set to n = 8 represented as the first eight dimensions of Table 3.8. For this case, in order to demonstrate the accuracy of the proposed linear regression approach, the mean and standard deviation (σ) of the transient response and output power at the output node N_1 is computed using two methods – the proposed linear regression approach described in Section 3.3.1 and the Monte Carlo approach. For the proposed approach, only $K = [2(P+1)/5] = 198$ substitutions are performed as described in Section 3.3.1. The comparison of the above results is shown in Fig. 3.17 where the proposed linear regression approach is found to exhibit good agreement with the Monte Carlo approach.

Table 3.8 Characteristics of random variables of Example 3 in section 3.3.3 (Fig. 3.16)

Random Variables	Mean	% Relative Variation
w_1 (width of TL_1)	0.2 mm	+/- 25% Normal Distribution
w_2 (width of TL_2)	0.25 mm	
w_3 (width of TL_3)	0.3 mm	
w_4 (width of TL_4)	0.7 mm	
w_5 (width of TL_5)	0.9 mm	
B_f (Current gain of BJT)	145	
C_{js} (Junction Cap. Of BJT)	667.5 fF	
R_l	50 Ω	
h_1 (height of TL_1)	0.4 mm	
h_2 (height of TL_2)	0.45 mm	
h_3 (height of TL_3)	0.5 mm	
h_4 (height of TL_4)	0.55 mm	
h_5 (height of TL_5)	0.6 mm	
R_2	100 Ω	

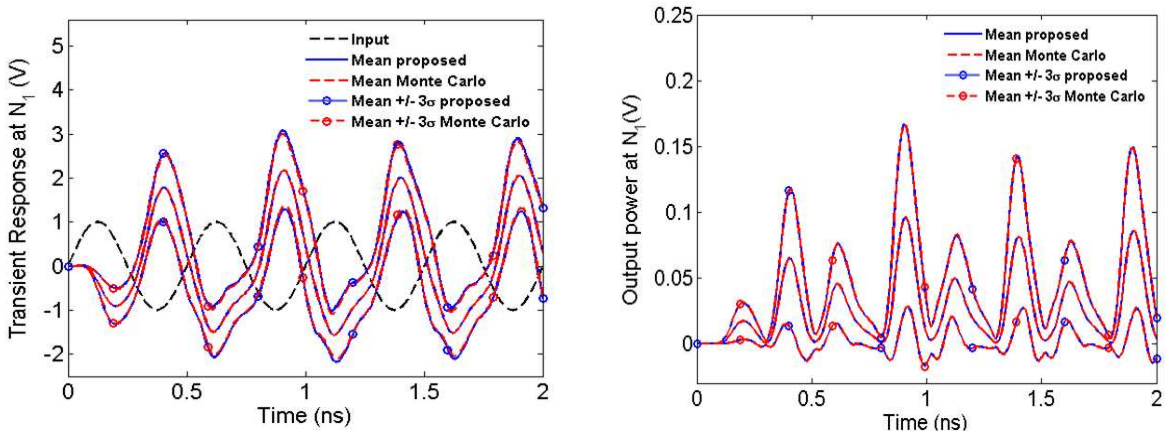


Fig. 3.17: Comparison of mean and standard deviation of the transient response of Example 3 in Section 3.3.3 computed using the proposed linear regression approach and the Monte Carlo approach. (a) Input, mean and statistical corners (+/- 3σ) of the transient response at N_1. (b) Mean and statistical corners (+/- 3σ) of the output transient power at N_1.

Next, to demonstrate the efficiency offered by the modified search algorithm over the classical Fedorov search algorithm of [39], the number of random dimensions is progressively increased from 4 to 14. For each test case, the original Fedorov search algorithm of [39] and the proposed modified search algorithm described in Section 3.3.1 are implemented. The total time cost for each algorithm is decomposed into two parts – the CPU cost incurred in identifying the regression nodes (compared in Table 3.9) and the CPU cost incurred in the matrix inversion after every node exchange (compared in

Table 3.10). From Table 3.9 it is observed that the speedup achieved using the K worst nodes compared to the full set of $2(P+1)$ regression nodes are consistently 5. This is exactly as expected from (3.20) in Section 3.3.1.3. Similarly, from Table 3.10 it is observed that the speedup provided by the Sherman-Morrison-Woodbury method compared to the explicit matrix inversion method closely matches the expected scaling of $O(n^m)$ as illustrated in Fig. 3.18(a). This too is as expected from the comparison of (21) with (15) in Section 3.3.1.3.

Table 3.9 Scaling of CPU time for identifying D-optimal nodes using proposed and classical search algorithms for example 3 in Section 3.3.3 (Fig. 3.16)

Random Variables	CPU Time for Proposed Algorithm using K Worst Nodes (sec)		CPU Time for Classical Search Algorithm using $2(P+1)$ nodes (sec)		Speedup
	Simulation Cost	Analytic Cost	Simulation Cost	Analytic Cost	
4 (w_1, w_2, w_3, w_4)	0.0005	0.0003	0.0026	0.0014	5
6 (w_1, w_2, w_3, w_4, w_5, B_f)	0.0055	0.0077	0.0277	0.0383	
8 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_l)	0.0615	0.1001	0.3074	0.5008	
10 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_l, h_1, h_2)	0.6806	0.8274	3.4034	4.1373	
12 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_l, h_1, h_2, h_3, h_4)	6.3336	4.9714	31.668	24.8570	
14 (w_1, w_2, w_3, w_4, w_5, B_f, C_{js}, R_l, h_1, h_2, h_3, h_4, h_5, R_2)	23.6232	23.6219	118.1160	118.1098	

It is further noted that the simulation CPU cost for both the modified search algorithm and the classical Fedorov search algorithm for Tables 3.9 and 3.10 have been compared with an analytic estimation of the CPU cost possible using (13)-(15) and (19)-(21). The analytic and simulation CPU cost results of Table 3.9 show good agreement. Even for Table 3.10, for higher dimensional problems (i.e. $n >$ 6) the analytic CPU costs are within a factor of 1.6 – 2 times the simulation CPU costs for the proposed approach and within a factor of 0.57 – 0.80 times the simulation CPU costs for the classical search algorithm.

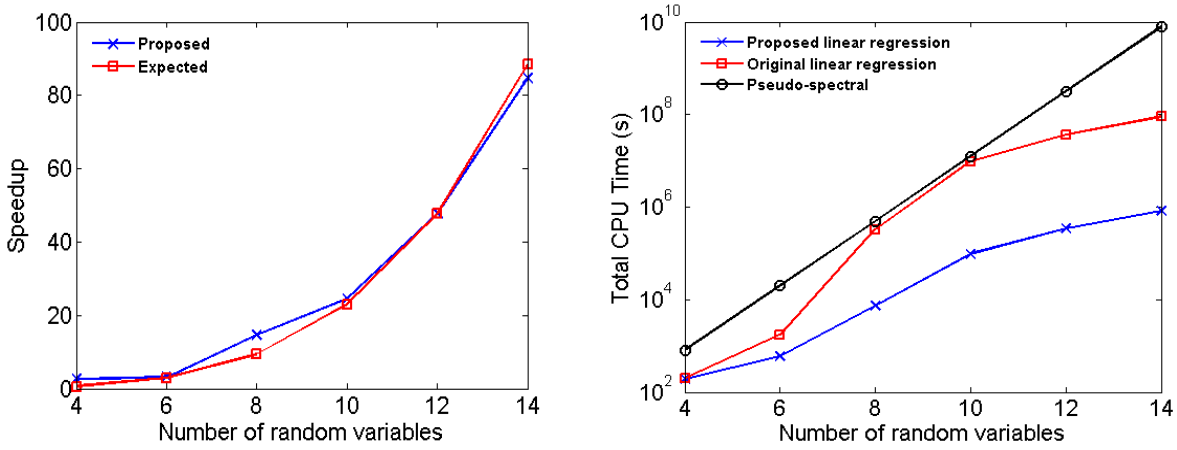


Fig. 3.18: Comparison of CPU time required by proposed linear regression approach with state-of-the-art approaches for example 3 in section 3.3.3 (Fig. 3.16). (a) Speedup achieved during matrix inverse computation by proposed algorithm compared to expected speedup. (b) Scaling of total CPU time cost of proposed linear regression algorithm compared against the pseudo-spectral approach and the original linear regression approach.

Finally, for the same test cases of Table 3.9 and 3.10, the total PC problem is solved using three methods – the proposed linear regression approach, the original linear regression approach of [39], and the pseudo-spectral collocation approach [29]. The total CPU time incurred by each approach is noted in Table 3.11 and plotted in Fig. 3.18 (b). For all methods, the time costs of the corresponding search algorithms have been added with the time cost for the $2(P+1)$ SPICE simulation costs. It is observed from Table 3.11 that the pseudo-spectral collocation exhibits an exponential scalability with respect to the number of random dimensions [29]. Thus, the pseudo-spectral collocation approach runs out of memory for more than 8 random variables. Similarly, for the linear regression approach of [39], the cost of the original search algorithm quickly becomes very large and also runs out of memory for more than 8 random variables. The CPU costs for n = 10, 12, and 14 for these methods is estimated via extrapolation and added in Fig. 3.18 (b) for completeness. As seen from Fig. 3.18 (b), the proposed linear regression approach which uses the more efficient modified search algorithm provides far superior scalability of the total CPU costs with respect to the number of random dimensions than the original linear regression approach of [39] or the pseudo-spectral collocation approach [29]. Interestingly, here too the total savings in CPU times increases with the number of random dimensions, thereby validating the benefits of the proposed linear regression approach for high-dimensional problems.

CHAPTER IV: HYPERBOLIC POLYNOMIAL CHAOS EXPANSION (HPCE)

In this chapter a new methodology for applying the gPC theory to uncertainty quantification problems is proposed. In this approach unlike the traditional gPC theory where orthogonal polynomial bases are selected based on a linear criterion introduced in Chapter II, a new hyperbolic criterion is suggested [75]. This criterion determines the most important polynomial bases based on the sparsity of effects [47], [48]; thus, it provides an sparse set of polynomial bases which is much shorter than the original set of bases; meanwhile, the loss of accuracy in results would be negligible.

This chapter starts with a detailed discussion on generation of multidimensional polynomial bases which is necessary for better understanding of the proposed approach. Then the new criterion for the hyperbolic approach is introduced and the new pattern for polynomial bases is presented graphically. Afterwards, an adaptive methodology for applying this approach on nonintrusive methods and in particular the nonintrusive ST based approach is proposed. Moreover, an upper bound on the computational cost of the proposed approach and its scaling with respect to number of random variables is evaluated and it is compared with other PC methods. Finally the accuracy and computational costs scaling of the proposed approach is validated through multiple numerical examples.

4.1 Generation of multidimensional polynomial bases

In Section 2.1.3 we discussed generation of multidimensional orthonormal polynomial bases which are obtained as a product of one-dimensional polynomial bases:

$$\phi_d(\boldsymbol{\lambda}) = \prod_{j=1}^{n} \phi_{d_j}(\lambda_j) \tag{4.1}$$

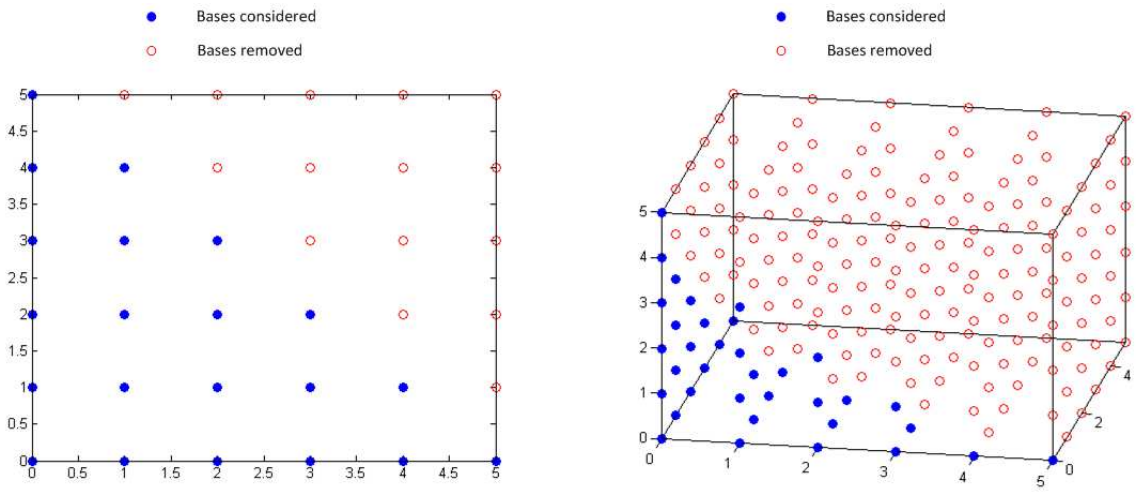
where λ_j is the j-th random variable, n is number of random variables, $\boldsymbol{\lambda}$ represents all random variables in a vector as $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^t$, d_j is the order of the j-th one-dimensional polynomial, ϕ_{d_j} shows the j-th one-dimensional polynomial, $\mathbf{d}$ represents all d_j values in a vector as $\mathbf{d} = [d_1, d_2, \ldots, d_n]^t$, and $\phi_{\mathbf{d}}$ is the

multidimensional polynomial basis made up of one dimensional polynomials whose order is mentioned in *d*.

In conventional methods the following criterion from the traditional gPC is used to determine order of each one-dimensional polynomial in a multidimensional polynomial basis:

$$\| \mathbf{d} \|_1 = d_1 + d_2 + ... + d_n \leq m \tag{4.2}$$

where $\|.\|_1$ represents the L_1 norm, and m is the common maximum expansion order for one dimensional polynomials.



(a) (b)

Fig. 4.1: Graphical illustration of the traditional gPC scheme for selection of multivariate polynomials. (a) The case of n = 2 and m = 5. (b) The case of n = 3 and m = 5.

For illustration purposes, Fig. 4.1 (a) demonstrates the graphical presentation of a simple example where m=5 and n=2, and Fig. 4.1 (b) demonstrates the graphical presentation of another example where m=5 and n=3. In these figures filled blue circles represent indices of selected polynomial bases and empty red circles represent indices of not selected polynomial bases. For instance in Fig. 4.1 (a) $\phi_{[3,1]}(\boldsymbol{\lambda}) = \phi_3(\lambda_1)\phi_1(\lambda_2)$ is selected because $\| [3,1] \|_1 = 4 \leq 5$. However, $\phi_{[2,4]}(\boldsymbol{\lambda}) = \phi_2(\lambda_1)\phi_4(\lambda_2)$ is not selected because $\| [2,4] \|_1 = 6 > 5$. More examples of the two dimensional case are presented in Table 2.3.

Likewise, in Fig. 4.1 (b) $\phi_{[2,2,1]}(\boldsymbol{\lambda}) = \phi_2(\lambda_1)\phi_2(\lambda_2)\phi_1(\lambda_3)$ is selected because $\|[2,2,1]\|_1 = 5 \le 5$. However, $\phi_{[3,2,2]}(\boldsymbol{\lambda}) = \phi_3(\lambda_1)\phi_2(\lambda_2)\phi_2(\lambda_3)$ is not selected because $\|[3,2,2]\|_1 = 7 > 5$. It is worth noting the number of polynomial bases provided by (4.2) and demonstrated in Fig. 4.1 is equal to P+1 in (2.13) which can be proved to be right using the following lemma. This *Lemma* and its proof are provided for comparison with the proposed hyperbolic scheme in following sections.

Lemma 4.1: The number of possible combinations in $\|\mathbf{d}\|_1 = d_1 + d_2 + \ldots + d_n \le m$, where d_j are integer values is:

$$P+1 = \binom{m+n}{n} = \frac{(m+n)!}{m!n!} = O\left(\frac{n^m}{m!}\right) \tag{4.3}$$

Proof: The equation $\|\mathbf{d}\|_1 \le m$ can be written as:

$$\|\mathbf{d}\|_1 = m \text{ OR } \|\mathbf{d}\|_1 = m-1 \text{ OR } .... \text{ OR } \|\mathbf{d}\|_1 = 0 \tag{4.4}$$

Number of possible combinations for each individual condition in (4.4) is easy to find. It is similar to the problem where there are m balls and they should be divided between n people. In order to divide m balls between n people, n-1 barriers are needed, and number of possible permutations for m balls and n-1 barriers is $(m+n-1)!$. Since all balls are similar to each other and all barriers are similar to each other too, number of possible combinations in this problem is

$$\frac{(m+n-1)!}{m!(n-1)!} = \binom{m+n-1}{n-1} \tag{4.5}$$

therefore number of possible combinations in (4.4) is

$$P+1 = \binom{m+n-1}{n-1} + \binom{m+n-2}{n-1} + ... + \binom{m-(m-1)+n-1}{n-1} + \binom{m-m+n-1}{n-1} \tag{4.6}$$

The last tem in the right hand side of (4.6) is equal to 1; hence, (4.6) can be written as

$$P+1=\binom{m+n-1}{n-1}+\binom{m+n-2}{n-1}+....+\binom{m-(m-1)+n-1}{n-1}+\binom{n}{n} \tag{4.7}$$

In order to continue this proof we take advantage of the Pascal's rule in combinatorics theory [76]

$$\binom{n}{k}=\binom{n-1}{k-1}+\binom{n-1}{k} \tag{4.8}$$

The Pascal's rule can be interpreted as a problem where k elements are selected out of n elements. The result can be partitioned into two groups, the ones including a particular element X and the ones without this element. Numbers of possible combinations for each group are the terms on the right hand side of (4.8) respectively [76].

Using (4.8), (4.3) can be written as

$$P+1=\binom{m+n}{n}=\binom{m+n-1}{n-1}+\binom{m+n-1}{n} \tag{4.9}$$

It is worth noting the first term on the right hand side of (4.9) is equal to the first term on the right hand side of (4.7). The Pascal's rule can be applied on the second term on the right hand side of (4.9)

$$P+1=\binom{m+n}{n}=\binom{m+n-1}{n-1}+\binom{m+n-2}{n-1}+\binom{m+n-2}{n} \tag{4.10}$$

where the second term on the right hand side is equal to the second term on the right hand side of (4.7). By continuing this pattern and applying the Pascal's rule for m times, the produced equation would be same as (4.7); therefore, *Lemma* 4.1 is proved.

Although the conventional scheme provides a relatively good scaling of computational cost with respect to number of random variables, it still faces cumbersome computational costs for problems with moderately high number of random variables since rate of scaling is $O(P+1) \approx O(n^m/m!)$ according to (4.3). The near-exponential or polynomial increase in computational cost is referred to as curse of

dimensionality and in order to address this issue a new scheme for selection of multidimensional bases is suggested in this chapter. This scheme is inspired by the sparsity of effects which is reviewed in the next section.

4.2 Sparsity of effects

Sparsity of effects is a common observation in *factorial experiments*, which are experiments involving the effect of two or more factors. In other words, in a factorial experiment there are more than one element such as A and B with different levels which can affect the output in different ways. The primary change caused in the response by the individual change in a factor is called a *main effect*, and variation caused in impact of a factor by changes in other factors is called an *interaction*. Furthermore, if A has a levels and B has b levels, the output can be impacted in ab ways, which is number of possible combinations of A and B. The study of factorial experiments is called *factorial design* [48].

Uncertainty problems can be treated as factorial experiments since every random variable is a factor which affects the response individually and also in interaction with other random variables. Likewise, in the PC expansion, each one-dimensional polynomial basis is a factor and its order of expansion is the level of that factor. In addition, the interaction between factors is equal to multidimensional polynomials bases.

A general rule in factorial experiments which can be proven heuristically is the sparsity of effects. Based on this rule, in a factorial experiment the number of relatively important effects is small, and lower effects are more likely to be important than higher order effects; moreover, effects with the same order are equally likely to be important [47]. In other words, in a system with several random variables, it is more likely for main effects and low order interactions to primarily impact the response since most of the high order interactions are possibly negligible [47]. This phenomenon is validated through several examples in the literature [48],[47].

The sparsity effect applies to the PC theory too. It is appreciated that in the PC expansion a lot of interactions are already discarded. In other words, in an example with n random variables and the common order of expansion of m, there are $(m+1)^n$ possible combinations. However, the PC expansion only considers $P+1$ polynomial bases. Although this number of polynomial bases is practical to implement in problems with a moderate number of random variables, it increases in a near exponential rate when the number of random variables is further increased. This problem is called the curse of dimensionality and prohibits application of the gPC theory in problems with relatively high number of dimensions. Therefore, further reduction in number of polynomial bases is desired.

Using the sparsity of effects, it is possible to discard more polynomial bases without a major loss in accuracy. In general lower degree polynomials have a higher impact on the output. For instance, in a case with 10 dimensions, n=10, and fourth order of expansion, m=4, where $P+1$=1001, Table 4.1 shows number of polynomial bases with degrees 0 to 4. However, if only the polynomials falling into the first two or three rows of this table are used, it is not possible to accurately report statistics of outputs with a fourth degree behavior.

Table 4.1 Number of polynomial bases with degrees 0 to 4 for the case of n=10 and m=4

Degree	**Number of polynomial bases**
0	1
1	10
2	55
3	220
4	715
Total	**1001**

In order to reduce number of polynomial bases without loss of accuracy, the works of [77] and [78] suggest considering the rank of polynomials, where rank is defined as the number of dimensions in a polynomial basis, for example $\phi_{[1,2,1]}(\boldsymbol{\lambda}) = \phi_1(\lambda_1)\phi_2(\lambda_2)\phi_1(\lambda_3)$ is a rank 3 while $\phi_{[3,0,0]}(\boldsymbol{\lambda}) = \phi_3(\lambda_1)$ is a rank 1. Table 4.2 shows number of polynomial bases with ranks 0 to 4. Although, this scheme provides an improvement in efficiency of the gPC theory, for a general nonintrusive approach since higher order

interactions will be missing, it fails to converge to the true model response [75]. Inspired by these approaches and [75] another sparse selection scheme is introduced in the next section.

Table 4.2 Number of polynomial bases with ranks 0 to 4 for the case of n=10 and m=4

Rank	Number of polynomial bases
0	1
1	40
2	270
3	480
4	210
Total	**1001**

4.3 Hyperbolic scheme for truncation of the PC expansion

As stated before, based on the sparsity of effects lower interaction polynomial bases have a higher impact on the response; therefore, in this section an alteration on the conventional scheme of (4.2) for truncation of the PC expansion is suggested to obtain the most important polynomial bases. This scheme results in selection of $M < P+1$ polynomial bases which are able to approximate the response as

$$X(\boldsymbol{\lambda}) = \sum_{i=0}^{M-1} c_i \phi_i(\boldsymbol{\lambda}) \tag{4.11}$$

In this approach, the constraint is put on the L_u^{th} norm of the indices' vector d, where $u \leq 1$, and the L_u^{th} norm is casually defined as the u-th root of summation of each member of d to the power of u. Replacing the L_1 norm in (4.2) with L_u^{th} would result in

$$\|d\|_u = \left(d_1^{\,u} + d_2^{\,u} + \ldots + d_n^{\,u}\right)^{1/u} \leq m \tag{4.12}$$

In this scheme, the indices are limited between positive axis and a hyperbola which is represented by

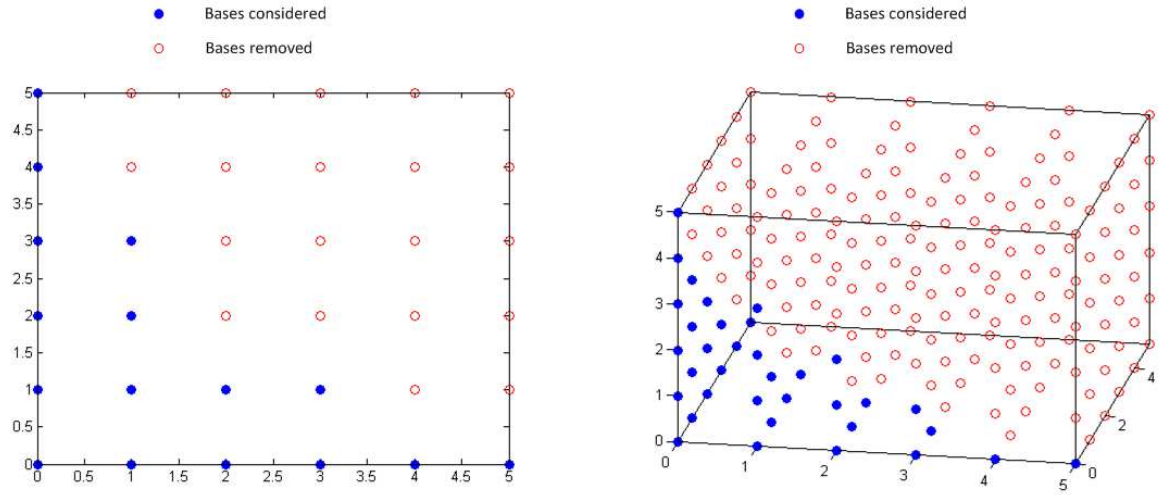
$$\left(d_1^{\,u} + d_2^{\,u} + \ldots + d_n^{\,u}\right)^{1/u} = m \tag{4.13}$$

Hence, this scheme is called hyperbolic, and u is called the hyperbolic factor. This technique would significantly reduce number of selected indices while preserving the accuracy in PC approaches. For

instance, in the case of 10 dimensions, n=10, and fourth order of expansion, m=4, where P+1=1001, Table 4.3 shows number of added polynomial bases when increasing the hyperbolic factor u.

Table 4.3 Number of added polynomial bases when increasing the hyperbolic factor u for the case of n=10 and m=4

Hyperbolic factor u	**Number of added polynomial bases**
$u \approx 0$, $u > 0$	41
0.5	45
0.7	90
0.8	120
1	705
Total	**1001**



(a) (b)

Fig. 4.2: Graphical illustration of the proposed hyperbolic scheme with u=0.76 for selection of multivariate polynomials. (a) The case of n = 2 and m = 5. (a) The case of n = 3 and m = 5.

In order to further illustrate this truncation scheme, Fig. 4.2 (a) demonstrates the graphical presentation of a simple example where m=5, n=2, and u=0.73. Fig. 4.2 (b) demonstrates the graphical presentation of another example where m=5, n=3, u=0.76. In these figures filled blue circles represent indices of selected polynomial bases and empty red circles represent indices of not selected polynomial bases. For instance in Fig. 4.2 (a) $\phi_{[3,1]}(\boldsymbol{\lambda}) = \phi_3(\lambda_1)\phi_1(\lambda_2)$ is selected because $\|[3,1]\|_{0.73} = 4.98 \leq 5$. However, $\phi_{[2,2]}(\boldsymbol{\lambda}) = \phi_2(\lambda_1)\phi_2(\lambda_2)$ is not selected because $\|[2,2]\|_{0.73} = 5.17 > 5$. It is worth noting that

Table 4.4 Equivalent orthonormal two-dimensional Hermite and Legendre polynomials of specified indices in Fig. 4.1 (b)

Bases	**Orthonormal Hermite Polynomial**	**Orthonormal Legendre Polynomial**	**Total degree**	**Rank**	$L_{0.73}^{th}$ **norm**
$\Phi_0(\boldsymbol{\lambda})$	1	1	0	0	0
$\Phi_1(\boldsymbol{\lambda})$	λ_1	$\sqrt{3}\,\lambda_1$	1	1	1
$\Phi_2(\boldsymbol{\lambda})$	λ_2	$\sqrt{3}\,\lambda_2$	1	1	1
$\Phi_3(\boldsymbol{\lambda})$	$(\lambda_1^2 - 1)/\sqrt{2}$	$\sqrt{5}\,(\frac{3}{2}\lambda_1^2 - \frac{1}{2})$	2	1	2
$\Phi_4(\boldsymbol{\lambda})$	$\lambda_1 * \lambda_2$	$3*\lambda_1 * \lambda_2$	2	2	2.58
$\Phi_5(\boldsymbol{\lambda})$	$(\lambda_2^2 - 1)/\sqrt{2}$	$\sqrt{5}\,(\frac{3}{2}\lambda_2^2 - \frac{1}{2})$	2	1	2
$\Phi_6(\boldsymbol{\lambda})$	$(\lambda_1^3 - 3\lambda_1)/\sqrt{6}$	$\sqrt{7} * (\frac{5}{2}\lambda_1^3 - \frac{3}{2}\lambda_1)$	3	1	3
$\Phi_7(\boldsymbol{\lambda})$	$\lambda_2 * (\lambda_1^2 - 1)/\sqrt{2}$	$\sqrt{15} * \lambda_2 * (\frac{3}{2}\lambda_1^2 - \frac{1}{2})$	3	2	3.82
$\Phi_8(\boldsymbol{\lambda})$	$\lambda_1 * (\lambda_2^2 - 1)/\sqrt{2}$	$\sqrt{15} * \lambda_1 * (\frac{3}{2}\lambda_2^2 - \frac{1}{2})$	3	2	3.82
$\Phi_9(\boldsymbol{\lambda})$	$(\lambda_2^3 - 3\lambda_2)/\sqrt{6}$	$\sqrt{7} * (\frac{5}{2}\lambda_2^3 - \frac{3}{2}\lambda_2)$	3	1	3
$\Phi_{10}(\boldsymbol{\lambda})$	$(\lambda_1^4 - 6\lambda_1^2 + 3)/2\sqrt{6}$	$\frac{105}{8}\lambda_1^4 - \frac{45}{4}\lambda_1^2 + \frac{9}{8}$	4	1	4
$\Phi_{11}(\boldsymbol{\lambda})$	$(\lambda_1^3\lambda_2 - 3\lambda_1\lambda_2)/\sqrt{6}$	$\sqrt{21} * (\frac{5}{2}\lambda_1^3\lambda_2 - \frac{3}{2}\lambda_1\lambda_2)$	4	2	4.98
$\Phi_{12}(\boldsymbol{\lambda})$	$(\lambda_2^3\lambda_1 - 3\lambda_2\lambda_1)/\sqrt{6}$	$\sqrt{21} * (\frac{5}{2}\lambda_2^3\lambda_1 - \frac{3}{2}\lambda_2\lambda_1)$	4	2	4.98
$\Phi_{13}(\boldsymbol{\lambda})$	$(\lambda_2^4 - 6\lambda_2^2 + 3)/2\sqrt{6}$	$\frac{105}{8}\lambda_2^4 - \frac{45}{4}\lambda_2^2 + \frac{9}{8}$	4	1	4
$\Phi_{14}(\boldsymbol{\lambda})$	$(\lambda_1^5 - 10\lambda_1^3 + 15\lambda_1)/2\sqrt{30}$	$\sqrt{11} * (\frac{63}{8}\lambda_1^5 - \frac{70}{8}\lambda_1^3 + \frac{15}{8}\lambda_1)$	5	1	5
$\Phi_{15}(\boldsymbol{\lambda})$	$(\lambda_2^5 - 10\lambda_2^3 + 15\lambda_2)/2\sqrt{30}$	$\sqrt{11} * (\frac{63}{8}\lambda_2^5 - \frac{70}{8}\lambda_2^3 + \frac{15}{8}\lambda_2)$	5	1	5

$\phi_{[3,1]}(\boldsymbol{\lambda})$ and $\phi_{[2,2]}(\boldsymbol{\lambda})$ both are of the same degree of four, and they also have the same rank of two; therefore they cannot be distinguished using schemes suggested in previous sections which put the constraint on the degree or rank. The equivalent orthonormal Hermite and legendre polynomials of filled blue circles are presented in Table 4.4. Likewise, in Fig. 4.2 (b) $\phi_{[2,2,0]}(\boldsymbol{\lambda}) = \phi_2(\lambda_1)\phi_2(\lambda_2)$ is selected because $\|[2,2,0]\|_{0.76} = 4.98 \le 5$. However, $\phi_{[2,1,1]}(\boldsymbol{\lambda}) = \phi_2(\lambda_1)\phi_1(\lambda_2)\phi_1(\lambda_3)$ is not selected because $\|[2,1,1]\|_{0.76} = 5.58 > 5$. It is worth noting that $\phi_{[2,2,0]}(\boldsymbol{\lambda})$ and $\phi_{[2,1,1]}(\boldsymbol{\lambda})$ both are of the same degree of four;

therefore, they cannot be distinguished by putting the constraint on the degree. Moreover, $\phi_{[1,1,1]}(\boldsymbol{\lambda}) = \phi_1(\lambda_1)\phi_1(\lambda_2)\phi_1(\lambda_3)$ is selected because $\|[1,1,1]\|_{0.76} = 4.24 \leq 5$. However, $\phi_{[1,2,1]}(\boldsymbol{\lambda}) = \phi_1(\lambda_1)\phi_2(\lambda_2)\phi_1(\lambda_3)$ is not selected because $\|[1,2,1]\|_{0.76} = 5.58 > 5$. In this case, both $\phi_{[1,1,1]}(\boldsymbol{\lambda})$ and $\phi_{[1,2,1]}(\boldsymbol{\lambda})$ have the same rank of 3; hence, they cannot be distinguished using the approach which puts the constraint on the rank of polynomials.

It is worth noting that the proposed HPCE approach does not depend on which uncertainty quantification approach is used. In fact all PC approaches mentioned in Chapter II, either intrusive or nonintrusive, could be used. This is merely because HPCE is another orthonormal expansion and does not affect the process of finding the coefficients and extracting statistics. However, the rest of this thesis is focused on nonintrusive approaches and particularly the linear regression and the nonintrusive stochastic based approaches because of their superior characteristics explained in previous chapters.

It can be interpreted from (4.12) and examples provided in this section that different values of the hyperbolic factor *u* result in different number of polynomials. This is demonstrated in Fig. 4.3 for the case with *n*=5 and *m*=4, where the conventional truncation scheme is depicted on the left side of the figure and the two other plots show that decreasing *u* results in a sparser set of polynomial bases. In other words *u*=1 is equivalent to the conventional PC expansion since with *u*=1 (4.12) converts to (4.2); besides, a larger *u* results in a higher number of polynomial bases and better accuracy while a smaller *u* results in a lower number of polynomials and lower accuracy.

Based on this argument, the main goal in HPCE is obtaining the hyperbolic factor *u* beacuse it determines accuracy and efficiency. Since the effect of *u* in a particular example is not known beforehand, this is not a trivial task. Therefore, this thesis proposes a novel adaptive approach for obtaining *u* which is described in the next section.

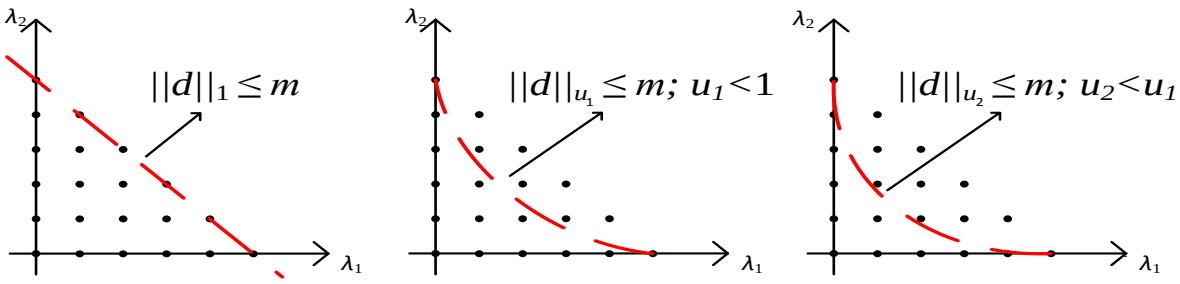


Fig. 4.3 Graphical interpretation of the truncation schemes using a 2D example (n = 2, m = 5), showing the traditional linear truncation scheme, the proposed hyperbolic truncation scheme, and the effect of decreasing the hyperbolic factor on the proposed approach, from left to right.

4.4 Derivation of statistical information using HPCE coefficients

The main goal in this thesis is the derivation of statistical information of the response. When using the HPCE this task is done similar to conventional PC approaches which is discussed in Section 2.1.4; nevertheless, in this approach a more limited set of polynomial bases is used. As for the first statistical moment or the arithmetic mean, (4.11) is entered in the integration formula of (2.17) and by considering the definition of inner product and the orthonormality condition it is simplified to the very first coefficient

$$E(x(\boldsymbol{\lambda})) = \int_{\boldsymbol{\Omega}} x(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = \sum_{i=0}^{M} \int_{\boldsymbol{\Omega}} c_i \phi_i(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} = \sum_{i=0}^{M} \left\langle c_i \phi_i(\lambda), \phi_0(\lambda) \right\rangle = c_0 \tag{4.14}$$

In order to find variance and standard deviation (4.11) is entered into the variance's integration formula of (2.20), and again by considering the definition of inner product and the orthonormality condition it is simplified to summation of square of every coefficient up to c_M except the first one.

$$\begin{aligned} Var(x(\boldsymbol{\lambda})) &= E[(x(\boldsymbol{\lambda}) - E(x(\boldsymbol{\lambda})))^2] = \int_{\boldsymbol{\Omega}} (\sum_{i=1}^{M} c_i \phi_i(\boldsymbol{\lambda}))^2 \rho(\boldsymbol{\lambda})d\boldsymbol{\lambda} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M} \left\langle c_i \phi_i(\boldsymbol{\lambda}), c_j \phi_j(\boldsymbol{\lambda}) \right\rangle = \sum_{i=1}^{M} \left\langle c_i \phi_i(\boldsymbol{\lambda}), c_i \phi_i(\boldsymbol{\lambda}) \right\rangle = \sum_{i=1}^{M} c_i^2 \end{aligned} \tag{4.15}$$

For finding the PDF and higher order statistical moments, like skewness and kurtosis, the technique similar to Monte Carlo, introduced in Section 2.1.4.3 is taken. In this method, first Q random

sample nodes are generated. The number of dimensions and distribution of these samples is same as random variables $\boldsymbol{\lambda}$ in the system. By having coefficients, polynomials ϕ_0 to ϕ_{M-1}, and random samples, the right hand side of (4.11) is known; therefore, after finding coefficients c_0 to c_{M-1} the result of Q instances of the experiment can be approximated. The final step is computing the PDF and higher order statistical moments from these approximated results.

4.5 Development of the adaptive HPCE approach

This section takes advantage of the scheme introduced in Section 4.3 to develop an efficient uncertainty quantification methodology to derive statistical information of the stochastic MNA equation of (2.38). One of the main contributions of this approach is the adaptive selection of the hyperbolic factor u; hence, it starts with discussing the possible range of u and then proposes techniques to find candidate hyperbolic factors which play an important role in this range. Afterwards, an iterative approach for application of the HPCE scheme on the nonintrusive ST based approach is introduced, and then an accuracy constraint in the form of HPCE coefficients' enrichment for stopping the algorithm is suggested. Having all the required tools, the overall program flow of the HPCE approach is discussed next. And at the end the computational cost of the proposed approach is computed and compared with other state of the art uncertainty quantification approaches.

4.5.1 Range of the hyperbolic factor *u*

The hyperbolic factor u has a great influence on the accuracy-sparsity tradeoff in constructing the HPCE. As it is mentioned before, when $u = 1$ the HPCE converges to the full-blown PCE which is highly accurate but computationally expensive to construct. On the other end of the spectrum, if the value of u is very small then only the one-dimensional bases are considered and all multidimensional bases are neglected. This leads to the sparsest expansion but is limited in its predictive accuracy. Here by one-dimensional bases we mean polynomial bases which have a rank up to 1. These bases are the ones located on the multidimensional axis in the graphical presentation. Since the common order of expansion is m and

there are n dimensions, the total number of one-dimensional bases is $m*n+1$, where the single basis at the center of the graphical presentation is loosely defined as one-dimensional too. Therefore, the spectrum of the hyperbolic factor u is equal to (0,1], where 0 is excluded because (4.12) would have no answer at $u=0$. The variation of u on its spectrum is presented in Fig. 4.4. Based on this demonstration and the previous explanation it is safe to say the main challenge of the proposed PC approach is how to adaptively tune the hyperbolic factor u for a general circuit problem. In the next section a simple approach is suggested.

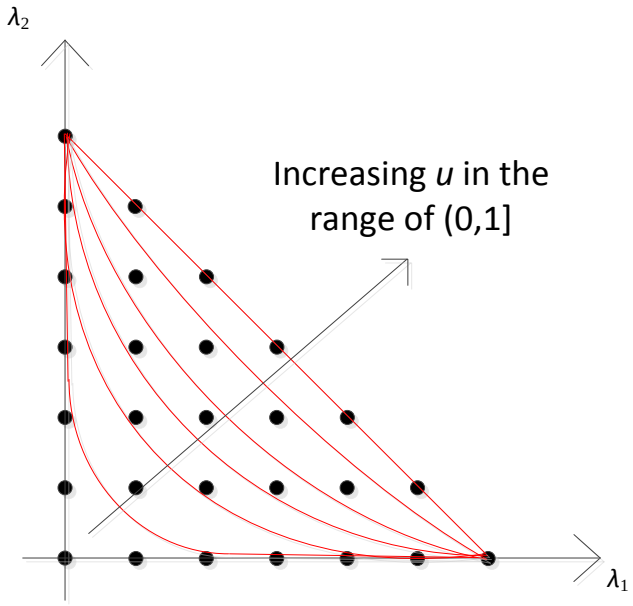


Fig. 4.4 The effect of increasing the hyperbolic factor u on selection of polynomial bases, when u is increased from a near zero value to 1.

4.5.2 Determining the hyperbolic factor by gradually increasing it

In the work of [79] we have suggested a greedy iterative approach to adaptively increase the hyperbolic factor from a very small starting value u_0 till the HPC expansion becomes enriched enough to satisfy a predefined error tolerance. Thus, for any j^{th} iteration, the value of u is increased as

$$u_j = u_{j-1} + \Delta u \tag{4.16}$$

where the subscript j is the iteration count and Δu is a fixed step size. This increase in u is intended to translate to the enrichment of the expansion from the previous $(j\text{-}1)^{th}$ iteration by adding new PC basis terms.

Although this technique is proven to be practical in [79], it is not the most efficient approach. The main reason is if Δu is very small there would be no difference in number of polynomial bases at some steps, and if Δu is too big there would be a big change in number of polynomial bases at some steps which could be broken down into two steps by taking finer steps. For instance, in the case of $n = 10$ and $m = 4$, five hyperbolic factors and their corresponding number of polynomial bases are presented in Table 4.3, these u values can be placed in a vector as $\mathbf{u} = [u_0, 0.5, 0.7, 0.8, 1]$ where $u_0 \approx 0$ and greater than zero, this will result in $\mathbf{M} = [41, 86, 176, 296, 1001]$ polynomial bases respectively. By considering a very fine Δu, it can be proved that it is impossible to distinguish polynomial bases to groups smaller than what is mentioned in the second column of Table 4.3. Therefore, if Δu is smaller than 0.1 there would be iterations with no change in the expansion; moreover, if Δu is greater than 0.1, number of selected polynomials might directly jump from 86 to 296 and force the algorithm to do extra computations. This is due to the fact that multidimensional indices are located on integer coordinates. Hence, it is possible to meet all indices' coordinates using a limited number of hyperbolas. In this example only five hyperbolas are needed, and they can be described using (4.13) and hyperbolic factors of $\mathbf{u} = [u_0, 0.5, 0.69, 0.79, 1]$; thus, we call these u values critical hyperbolic factors.

Since u values are unknown, the safest practice is to choose a very small Δu and check for enrichment only when there is a change in number of polynomials. However, this approach results in unnecessary computational costs and it fails to provide a strong argument to prove it has not missed any critical hyperbolic factor. In order to address this issue a closed-form technique for finding these critical hyperbolic factors is suggested in the next section.

4.5.3 Closed-form technique for determination of the critical hyperbolic factors

In this section a closed-form technique is suggested to initially find critical hyperbolic factors i.e. u values which by crossing them there would be a change in number of polynomial bases. As mentioned in the previous section, all integer coordinates showing multidimensional bases can be met with a limited number of hyperbolas; therefore, all critical hyperbolic factors can be found by solving (4.13) for u. It can be done since in a specific example n and m are known and d_i values are selected from the P+1 indices, which are selected by the conventional PC scheme i.e. (4.2). For instance when n = 3, m = 5 and d = [1,2,1] we solve the following equation

$$\left(d_1^{\,u} + d_2^{\,u} + d_3^{\,u}\right)^{1/u} = 5 \tag{4.17}$$

Where the answer is u = 0.8248. Therefore, the polynomial basis $\phi_{[1,2,1]}(\boldsymbol{\lambda}) = \phi_1(\lambda_1)\phi_2(\lambda_2)\phi_1(\lambda_3)$ is discarded when u = 0.76, but it is selected when u = 0.83. Inspired by this characteristic of multidimensional indices, we solve (4.13) P+1 times for all **d** values which were originally selected by the conventional PC scheme i.e. (4.2). The result is a limited set of u values which is common for many of the equations. These u values are the critical hyperbolic factors, they can be organized in a new vector **u**

$$\mathbf{u} = [u_0, u_1, ..., u_k] \tag{4.18}$$

where k is number of critical hyperbolic factors, and it is guaranteed when moving from u_{j-1} to u_j, with 1 $\leq j \leq k$, there would be a change in the number of selected polynomial bases. In other words, if every polynomial basis is tagged with an index vector $\mathbf{d}_i$, by moving to the next critical hyperbolic factor there would be one or more polynomial basis index where

$$\left\|\mathbf{d}_i\right\|_{u_{j-1}} > m; \quad \left\|\mathbf{d}_i\right\|_{u_j} \leq m \tag{4.19}$$

Therefore, unlike the scheme in the previous section the computational cost for classification of polynomial bases is minimized and it is proved the algorithm does not miss any critical hyperbolic factor.

4.5.4 Application of HPCE on the nonintrusive ST based approach

As mentioned before the HPCE scheme can be applied to any PC approach; however, in this chapter the focus is on the nonintrusive ST based approach [37], introduced in Section 2.2.2.3, because of its efficiency. This technique starts with setting the hyperbolic factor to $u_0 \approx 0$ and $u_0 > 0$. Based on the discussion in Section 4.5.1 this will result in $M_0 = m*n+1$ polynomial bases ϕ_0 to ϕ_{m*n}. Then, the system of linear algebraic equations in (2.52) is modified for this number of polynomial bases:

$$\mathbf{A}_0 \tilde{\mathbf{X}}_0 = \mathbf{E}_0 \tag{4.20}$$

with

$$\mathbf{A}_0 = \begin{bmatrix} \phi_0(\boldsymbol{\lambda}^{(1)})\mathbf{I} & \dots & \phi_M(\boldsymbol{\lambda}^{(1)})\mathbf{I} \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^{(M)})\mathbf{I} & \dots & \phi_M(\boldsymbol{\lambda}^{(M)})\mathbf{I} \end{bmatrix}; \tilde{\mathbf{X}}_0 = \begin{bmatrix} \mathbf{X}_0(t) \\ \vdots \\ \mathbf{X}_M(t) \end{bmatrix}; \mathbf{E}_0 = \begin{bmatrix} \mathbf{X}(t, \boldsymbol{\lambda}^{(1)}) \\ \vdots \\ \mathbf{X}(t, \boldsymbol{\lambda}^{(M)}) \end{bmatrix} \tag{4.21}$$

where $\mathbf{I}$ represents the identity matrix, M is the length of the PC expansion which in this case is $m*n+1$, $\mathbf{X}_i$ is PC coefficient of the i-th polynomial basis, and $\mathbf{X}(t,\boldsymbol{\lambda}^{(i)})$ is the response probed at the sample node $\boldsymbol{\lambda}^{(j)}$. These sample nodes are selected by exploiting the ST node selection technique introduced in Section 2..2.2.2, where instead of the conventional PC expansion with $P+1$ bases only $m*n+1$ one-dimensional bases are used in the selection process for $m*n+1$ nodes.

In the next step, the hyperbolic factor is set to u_1, which in turns results in M_1 polynomial bases ϕ_0 to ϕ_{M_1-1}, where $M_1 > M_0$. Afterwards, M_1 sample nodes are generated using the ST technique of Section 2.2.2.2 with one small change. This time instead of starting with one single node with the highest quadrature weight, the node selection technique starts with the M_0 nodes that it had chosen in the previous step and generates the vector space of (2.51) using the M_1 new polynomial bases

$$\mathbf{V} = \left\{\mathbf{H}(\boldsymbol{\lambda}^{(1)}),...\mathbf{H}(\boldsymbol{\lambda}^{(M_0)})\right\} \tag{4.22}$$

where

$$\mathbf{H}(\boldsymbol{\lambda}^{(i)}) = [\phi_0(\boldsymbol{\lambda}^{(i)}), \phi_1(\boldsymbol{\lambda}^{(i)}),....\phi_{M_1}(\boldsymbol{\lambda}^{(i)})]^T \tag{4.23}$$

Then the remaining $M_1 - M_0$ are selecting by continuing the ST node selection approach in the regular way and selecting nodes which have a $\mathbf{H}(\boldsymbol{\lambda}^{(i)})$ vector with a large enough component orthogonal to $\mathbf{V}$. Next, (4.20) is written for M_1 polynomial bases and nodes. It is worth noting this time we already have the first M_0 elements in $\mathbf{E}$, which means only $M_1 - M_0$ new SPICE simulations are needed since the first M_0 SPICE simulations are already done in the previous step. Besides the first M_0 columns in the first M_0 rows of matrix $\mathbf{A}$ are in common with the previous step and the rest needs to be computed. Moreover, all coefficients in $\mathbf{X}$ are recomputed and the first M_0 coefficients are not necessarily the same as coefficients in the previous step. Therefore, the new version of system of linear algebraic equations can be written as

$$\mathbf{A}_1\tilde{\mathbf{X}}_1 = \mathbf{E}_1 \tag{4.24}$$

where

$$\tilde{\mathbf{X}}_1 = \left[\mathbf{X}_0(t),....\mathbf{X}_{M_1-1}(t)\right],\ \mathbf{E}_1 = \left[[\mathbf{E}_0]^T,....\mathbf{X}(t,\boldsymbol{\lambda}^{(M_1)})\right]^T \tag{4.25}$$

$$\mathbf{A}_1 = \begin{bmatrix} \begin{bmatrix} \\ \mathbf{A}_0 \\ \\ \end{bmatrix} & \cdots & \phi_{M_1-1}(\boldsymbol{\lambda}^{(1)}) \\ & & \vdots \\ & \cdots & \phi_{M_1-1}(\boldsymbol{\lambda}^{(M_0)}) \\ \vdots \quad \vdots & & \vdots \quad \vdots \\ \phi_0(\boldsymbol{\lambda}^{(M_1)}) \cdots \phi_{M_0-1}(\boldsymbol{\lambda}^{(M_1)}) & \cdots & \phi_{M_1-1}(\boldsymbol{\lambda}^{(M_1)}) \end{bmatrix}$$

This process can continue for u_2, u_3, … u_k, by expanding the $\mathbf{A}$ matrix, doing more SPICE simulations and solving for the new set of HPCE coefficients, in the same way. In other words the general format of system of linear algebraic equations at step j can be written as

$$\mathbf{A}_j \tilde{\mathbf{X}}_j = \mathbf{E}_j \tag{4.26}$$

where

$$\tilde{\mathbf{X}}_j = \left[\mathbf{X}_0(t),....\mathbf{X}_{M_j-1}(t)\right], \quad \mathbf{E}_1 = \left[\left[\mathbf{E}_{j-1}\right]^T,....\mathbf{X}(t,\boldsymbol{\lambda}^{(M_j)})\right]^T \tag{4.27}$$

$$\mathbf{A}_j = \begin{bmatrix} \begin{bmatrix} & \mathbf{A}_{j\text{-}1} & \end{bmatrix} & \cdots & \phi_{M_j-1}(\boldsymbol{\lambda}^{(1)}) \\ & & \vdots \\ & & \phi_{M_j-1}(\boldsymbol{\lambda}^{(M_{j-1})}) \\ \vdots & \vdots & \vdots \\ \phi_0(\boldsymbol{\lambda}^{(M_j)}) \cdots \phi_{M_{j-1}-1}(\boldsymbol{\lambda}^{(M_j)}) & \cdots & \phi_{M_j-1}(\boldsymbol{\lambda}^{(M_j)}) \end{bmatrix}$$

However, between each two steps an accuracy check needs to be done to stop the algorithm if the desired accuracy is met. This is done by computing the normalized enrichment of PC coefficients gained by increasing the hyperbolic factor from u_{j-1} to u_j. This process is explained in the next section.

4.5.5 Enrichment of the adaptive HPCE approach

Knowing the critical hyperbolic factors, we have a strong tool to distinguish between polynomial bases and divide them to smaller groups. The decision on how many of these groups should be exploited in a particular example is based on the desired accuracy. In this section, the accuracy measures used in making this decision are introduced.

Adding each group of the nodes to the process, results in an increase in the accuracy. This is due to the fact that, by solving (4.26) and using the least square technique the new HPCE coefficients are so chosen to minimize the residual error of the approximation. In other words

$$\tilde{\mathbf{X}}_j(t) = \arg\min \left\| \mathbf{X}(t,\boldsymbol{\lambda}) - \sum_{i=0}^{M_j} \mathbf{X}_i(t)\Phi_i(\boldsymbol{\lambda}) \right\|_2 \tag{4.28}$$

when (4.28) converges to zero it results in the error between two set of coefficients from two subsequent steps to be very small too, which in practice translates to coefficients of new polynomial bases be close to zero and the change in coefficients of old polynomial bases be trivial.

This thesis suggests putting the accuracy constraint on the variation of the response, since this statistical moment is usually requested and is more sensitive to changes in HPCE coefficients comparing the arithmetic mean. Once coefficients of step j, $j \geq 1$, are evaluated, the normalized enrichment of the variance of the circuit responses due to the additional new bases can be analytically measured as

$$En(t) = \frac{\sum_{k=M_{j-1}+1}^{M_j} X_j^{(k)}(t)^2}{(\sum_{k=1}^{M_{j-1}} X_j^{(k)}(t)^2 + \sum_{k=M_{j-1}+1}^{M_j} X_j^{(k)}(t)^2) * q_j} \tag{4.29}$$

where $q_j = M_j - M_{j-1}$ is the number of added polynomial bases at step j and it is placed in the enrichment formula to show the average value of added polynomial bases. It is worth noting that at later steps of the algorithm, q_j increases and $En(t)$ decreases since on average the contribution of each added polynomial basis reduces. Provided the time average of this enrichment is greater than a prescribed tolerance, the iteration will continue. Once the enrichment measure of (4.29) falls below the tolerance, it is assumed that the point of diminishing return has been reached and then the iterations are stopped. The resultant PC expansion is referred to as the HPCE.

4.5.6 The program flow of the adaptive HPCE approach

In this section a summary of the overall program flow of the algorithm which controls the adaptive HPCE approach is presented. Flowchart of this process is demonstrated in Fig. 4.5, this approach starts with setting the hyperbolic factor as $u_0 \approx 0$ and finding the corresponding polynomial bases, then using the nonintrusive ST approach M_0 sample nodes are selected and SPICE simulations are done at these nodes. Afterwards by replacing the simulation results in (4.20) and solving the equation in the least square manner, the first set of coefficients i.e. $\mathbf{X}_0$ are found. In the next step, the hyperbolic factor is

updated to u_1, M_1 corresponding polynomial bases and sample nodes are found and by doing $M_1 - M_0$ SPICE simulations and placing them in (4.24) the new set of coefficients are calculated. Then, the normalized enrichment from step one to step two is computed using (4.29) and the result is compared against a prescribed scalar accuracy threshold which is set by user and depends on the example. If time average of the normalized enrichment is greater than the threshold, the process is stopped and $\mathbf{X}_0$ is reported as the HPCE coefficients. However, if the normalized enrichment is less than the threshold, the algorithm continues to update the hyperbolic factor to the next u value. This process continues until time average of the normalized enrichment falls below the threshold or the hyperbolic factor reaches 1 which means HPCE is converted to the conventional PC expansion. However, through multiple numerical examples at the end of this chapter it is observed that if the threshold is set realistically, it will be met before the hyperbolic factor reaches 1. It is noted that provided $u < 1$ when the iterations are stopped, sparsity in the HPCE is guaranteed. Moreover, since all experiments' results are reused the total number of SPICE simulations stays as a fraction of the conventional PC approach.

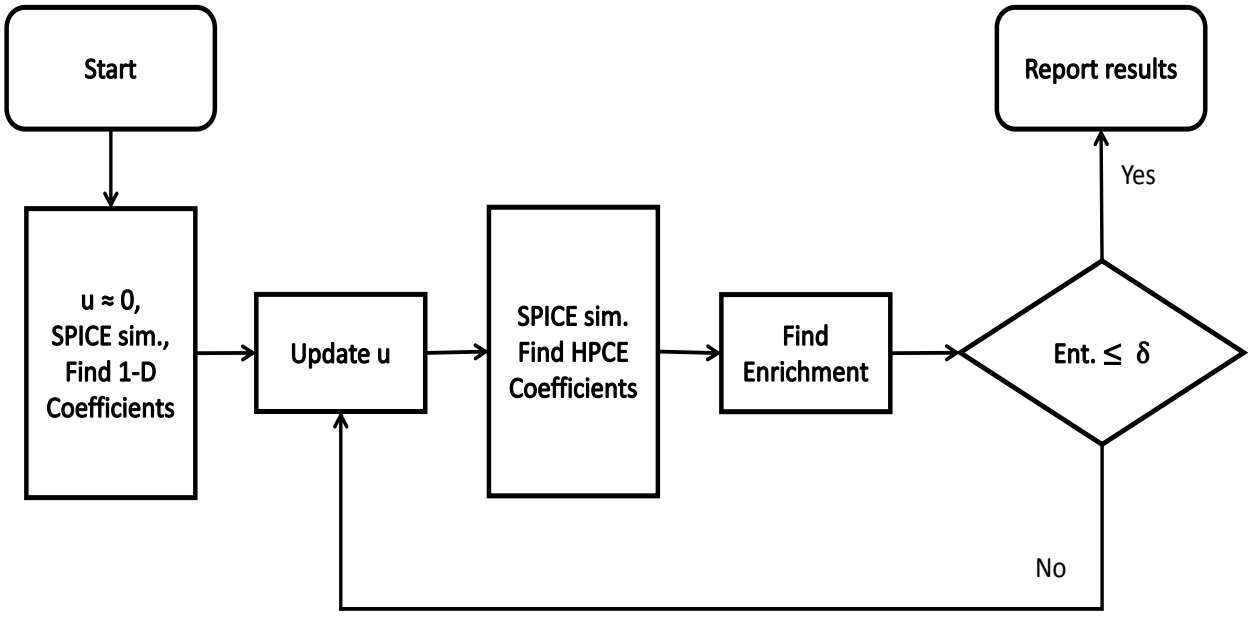


Fig. 4.5 Flowchart of the algorithm controlling the adaptive HPCE approach.

4.5.7 Computational cost of the proposed adaptive HPCE approach

This section starts with finding an upper bound for number of polynomial bases in the adaptive HPCE approach, and then it quantifies the CPU time cost of this approach and contrasts that against conventional non-intrusive PC approaches. Based on the discussion of Section 4.5.4 it is appreciated that the CPU time required for determination of nodes is negligible compared to computations cost of SPICE simulations since the nonintrusive ST based approach is used for node selection. This node selection technique is very time efficient because it does not involve any matrix inversion when studying candidate nodes. This can also be verified from numerical examples of Chapter III and particularly CPU times of the nonintrusive ST approach presented in Table 3.5 which shows the CPU time of the node selection approach to be less than one percent of SPICE simulations. Therefore, only the CPU cost of SPICE simulations is considered in this section.

4.5.7.1 Upper bound on number of polynomial bases in the HPCE approach

Based on the discussion of Section 4.1 and *Lemma* 4.1 we proved that the number of polynomial bases in the conventional PC approach is equal to $P+1=C(m+n, n)$. However, the same technique cannot be used to find number of polynomial bases as a function of u because the left hand side of (4.12) might involve non-integer values. Therefore, assuming $u < 1$, in this section we propose an upper bound on number of polynomial bases in the proposed approach, which still shows a great speed up comparing to the conventional PC. This upper bound is found by exploiting the following *Lemma*.

Lemma 4.2: An upper bound on number of possible combinations in $\|\mathbf{d}\|_u = (d_1^{\,u} + d_2^{\,u} + \ldots + d_n^{\,u})^{1/u} \leq m$, where d_j are integer values and $u < 1$, is:

$$\tilde{M} = \binom{m+n-1}{n} + n = \frac{(m+n-1)!}{(m-1)!n!} + n \tag{4.30}$$

Proof: By observing the graphical presentation of two and three dimensional indices of the conventional PC approach, demonstrated in Fig. 4.1, it is noted that the nodes located on the surface, except the ones

which are also on one-dimensional axes, cannot be distinguished using the HPCE approach, and in general this is true for any n dimensional PC expansion. It is merely because the hyperbola described in (4.13) with u=1, is the first hyperbola to cross all these nodes which means in the HPCE approach they are considered to have the same value and by slightly decreasing u all of them would be discarded from HPCE. The nodes which are also on the axes are exception because they represent the one-dimensional polynomials; hence, they would be selected even with a near zero hyperbolic factor.

Since when u is less than one all nodes on the surface, except the ones in common with one-dimensional axes, are discarded, and in a successful HPCE u is always less than 1, we can argue by subtracting number of these nodes from P+1 it is possible to find an upper bound on the number of polynomial bases in the HPCE approach.

Number of nodes on the surface can be determined by finding number of possible combinations for **d** in

$$\| \mathbf{d} \|_1 = d_1 + d_2 + ... + d_n = m \tag{4.31}$$

Thus, based on (4.5) number of nodes on the surface is

$$\frac{(m+n-1)!}{m!(n-1)!} = \binom{m+n-1}{n-1} \tag{4.32}$$

And since there are n nodes in common with one-dimensional axes, number of discarded nodes for a u which is slightly less than 1 would be

$$\frac{(m+n-1)!}{m!(n-1)!} - n \tag{4.33}$$

Therefore, after subtracting number of discarded nodes from number of total nodes in the conventional PC, the upper bound can be found as

$$\tilde{M} = P + 1 - \frac{(m+n-1)!}{m!(n-1)!} + n = \frac{(m+n)!-n(m+n-1)!}{m!n!} + n$$
$$= \frac{(m+n-n)(m+n-1)!}{m!n!} + n = \frac{(m+n-1)!}{(m-1)!n!} + n = \binom{m+n-1}{n} + n \tag{4.34}$$

Hence, *Lemma* 4.2 is proved.

It is worth noting, (4.33) also shows number of polynomial bases for the penultimate critical hyperbolic factor in **u** since based on the discussion in Section 4.5.3 there would be no difference in HPCE if u is greater than or equal to u_{k-1} and smaller than 1.

4.5.7.2 Comparing computational cost of HPCE with other nonintrusive PC approaches

This section quantifies the CPU time cost of the proposed adaptive HPCE approach, and compares it with conventional non-intrusive PC approaches. It is observed that no comparative analysis with respect to the intrusive SG approach is performed since it is well-established that for high-dimensional problems most non-intrusive approaches will outperform the SG approach [45]. This is particularly true for nonlinear circuits where the multidimensional integral of (2.37), using the SG approach, has to be represented in SPICE using massive number of additional voltage/current dependent sources.

4.5.7.2.1 Total computational cost of the adaptive HPCE approach

As mentioned before, the major computational cost in the adaptive HPCE approach is spent on SPICE simulations and the CPU time for node selection is negligible. In this section we calculate the CPU cost in the upper bound case of HPCE with $\tilde{M}$ polynomial bases. Based on the discussion in Section 4.5.7.1 it is known that the CPU cost is always equal or less than this upper bound case.

For the SPICE simulation cost, it is assumed that each of the $\tilde{M}$ simulations requires the same CPU cost which is a reasonable assumption since the variation in the unknowns of the MNA equations of

(2.38) from one SPICE simulation to another will be typically small. Thus, the SPICE simulation cost or in other words the total computational cost can be quantified as

$$C_T = \tilde{M} C_0 \tag{4.35}$$

where C_0 is the cost of each deterministic SPICE simulation. Thus, the overall cost of the proposed approach scales as $O(n^{m-1})$ with respect to the number of random dimensions (n). This scaling can be easily derived by simplification of (4.34).

4.5.7.2.2 Stochastic collocation approach

Stochastic collocation (SC) has been a very popular nonintrusive PC approach [32], [33], [45], [46]. In this approach, if the non-intrusive multidimensional nodes are selected to be the full tensor product of 1D quadrature nodes it will result in $M = (m+1)^n$ sample nodes. These nodes can be analytically identified at negligible computational costs (i.e. CPU time of node selection = 0). Thus, the cumulative costs of the entire SC approach is equal to that of the SPICE simulations and is expressed as

$$C_T = (m+1)^n C_0 \tag{4.36}$$

This corresponds to an exponential scaling of the time costs with respect to the number of random dimensions (n), quantified as $O((m+1)^n)$. This means that for even moderate dimensional problems, the massive cost of SPICE simulations in (4.36) will make this approach highly cost intensive compared to the proposed adaptive HPCE approach.

In order to mitigate this prohibitive scaling, an intelligent choice of only a sparse subset of the tensor product nodes guided by the Smolyak algorithm has been proposed [32], [33], [45], [69]. Once again, this method allows the fast identification of the sparse nodes. This approach results in a decrease in the number of multidimensional nodes from $M = (m+1)^n$ to approximately $M = (2n)^m/m!$, thereby improving the CPU time costs of the SC algorithm from that of (4.36) to

$$C_T = \frac{(2n)^m}{m!} C_0 \tag{4.37}$$

For this approach, it is observed that the number of deterministic SPICE simulations required scales as $O(2^m n^m)$ which is still $2^m n$ times more than that required for the proposed adaptive HPCE approach. Thus, adaptive HPCE still remains more cost effective than even this sparse collocation approach.

4.5.7.2.3 The conventional linear regression approach

The conventional linear regression approach, introduced in Section 2.3.2, takes advantage of the traditional Fedorov search algorithm and needs to do SPICE simulations at $2(P+1)$ or $3(P+1)$ sample nodes [27], [28]. In this approach the computational cost of finding the nodes cannot be neglected because of the high number of the matrix inversions and matrix vector multiplications. In fact this process might cause a computational cost comparable to the CPU time of SPICE simulations in a moderately high number of random variables.

Based on the discussion of Section 3.1.2, the computational cost of the node selection process for the conventional linear regression approach is

$$\begin{aligned} C_t &= 2(P+1)((m+1)^n - 2(P+1))C_1 + k(P+1)C_2 \\ &\approx 2(P+1)(m+1)^n C_1 + k(P+1)C_2; \quad k \in \{2,3\} \end{aligned} \tag{4.38}$$

where the first term on the right hand side of (4.38) represents the CPU cost required to compute Δ_r of (3.1) $(m+1)^n - k(P+1)$ times for each node of the starting set with C_1 as the costs of performing all the matrix-vector and vector-vector multiplications of (3.1) assuming that the inverse of the information matrix is known, the second term represents the CPU cost of the requisite matrix inversions with C_2 as the time cost to perform one matrix inversion. It is appreciated that the cost C_1 scales as $O(n^{2m})$ with respect to the random dimensions (n). On the other hand, the cost C_2 scales as $O(n^{3m})$ or $O(n^{2m})$ depending on whether a direct or indirect approach for matrix inversion is used.

Furthermore the cost of $K(P+1)$ SPICE simulations would be

$$C_S = K(P+1)C_0 \tag{4.39}$$

It is clear from (4.38) and (4.39) that the proposed adaptive HPCE approach is far more efficient than the conventional linear regression approach since its CPU time for node selection is negligible compared to (4.38) and in SPICE simulations it is *Kn* times more efficient than the conventional linear regression approach.

4.5.7.2.4 Nonintrusive stochastic testing based approach

The promising nonintrusive stochastic testing based approach which has been proposed in [37] and reviewed in Section 2.2.2.3 is one of the most efficient available approaches. However, as described in Section 4.5.4, the proposed adaptive HPCE approach has improved the efficiency of this method even further.

The nonintrusive ST based approach requires only P+1 SPICE simulations as opposed to the 2(P+1) simulations required by the linear regression approach, and since the cost for selecting the nodes is negligible the total computational cost of this approach would be

$$C_T = (P+1)C_0 \tag{4.40}$$

This in turns translates to a total scaling rate of $O(n^m)$ for computational cost of the nonintrusive ST based approach which can be derived by simplification of (4.3). As it can be seen, the proposed adaptive HPCE approach is about n times faster than the nonintrusive ST based approach.

Among other existing non-intrusive approaches, there is the pseudo-spectral collocation approach which is introduced in Section 2.3.1. However, this approach suffers from the same exponential scaling of the SPICE simulation costs as the classical SC approach. Other methods based on the Stroud low order cubature methods, introduced in Section 2.3.4, have also been reported. This approach can easily locate the multidimensional nodes using simple analytic formulas and exhibits only a linear scaling of the number of SPICE simulations with number of random dimensions (i.e. $O(n)$). However, this excellent

scaling with the number of random dimensions only exists for a second and third degree PC expansion and cannot be extended to higher degree expansions [26].

From the above analysis, it is observed that the proposed adaptive HPCE approach offers clear benefits over the state-of-the-art PC approaches and this is validated through multiple examples in the next section.

4.6 Numerical examples

In this section, three examples are presented to compare the accuracy and scalability of the proposed adaptive HPCE approach against state of the art nonintrusive PC approaches. All relevant computations are performed using MATLAB 2013b while the deterministic transient simulations are performed using HSPICE [3]. In particular, the transmission line networks of the presented examples are modeled using the W-element transmission line model provided by HSPICE which can consider frequency dependent per-unit-length parameters [3]. The above simulations are run on a workstation with 8 GB RAM, 500 GB memory and an Intel i5 processor with 3.4 GHz clock speed.

4.6.1 Example 1

The objective of this example is to compare the performance of the proposed adaptive ST based approach with the nonintrusive ST based approach [37]. For this purpose, the multiconductor transmission line (MTL) network of Fig. 4.6 terminated by inverters made up of SPICE level 49 CMOS transistor models is considered. The lengths of the MTL networks are set to 5 cm and their layout and geometric dimensions are shown in Fig. 4.7. This network is driven by two voltage sources with a trapezoidal waveform of rise/fall time T_r = 0.1 ns, pulse width T_w = 1 ns and amplitude of 5V. The uncertainty in the network is introduced via ten random variables (n = 10) whose characteristics are listed in Table 4.5 where the last four random variables are related to characteristics of NPN and PNP transistors used to construct inverters in the termination. For this example a Legendre PC expansion of degree m = 4 is considered.

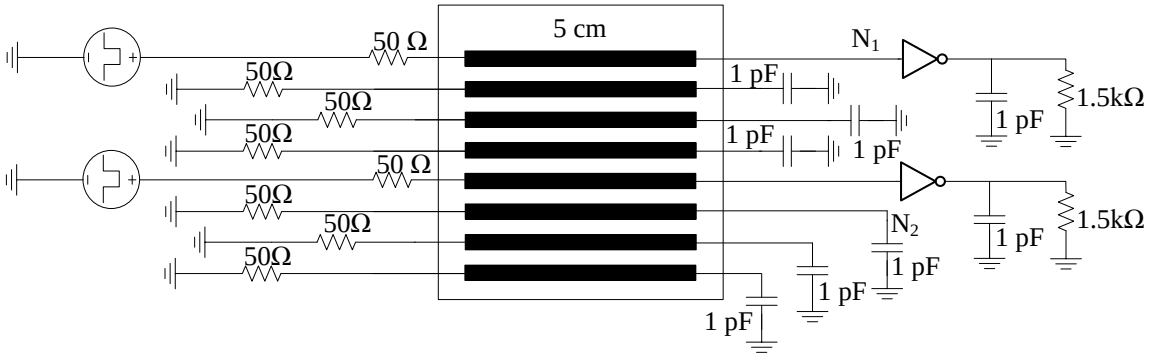


Fig. 4.6: Multiconductor transmission line (MTL) network for Example 1 in Section 4.6.

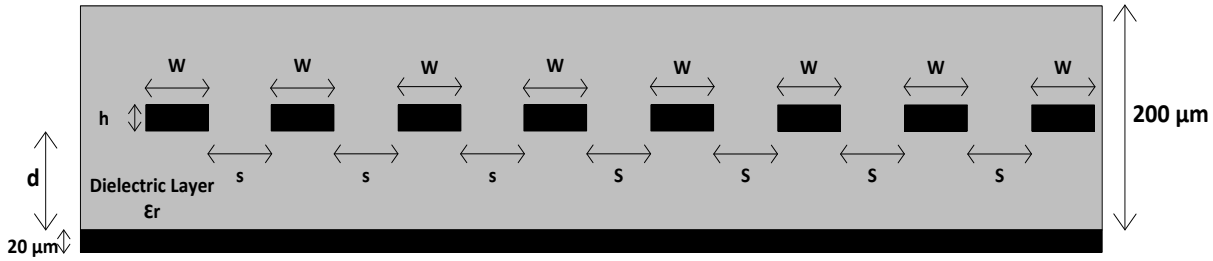


Fig. 4.7: Geometrical and physical layout of transmission lines of Example 1 in Section 4.6

Table 4.5 Characteristics of random variables of example 1 in Section 4.6 (Fig. 4.6 and Fig. 4.7)

Random Variables	Mean	% Standard Deviation
h	100 µm	+/- 30 % Uniform Distribution
W	90 µm	
d	30 µm	
s	200 µm	
Transmission line conductivity	5.8e7	
Permittivity	4.1	
NPN channel length	0.1 µm	
NPN channel width	10 µm	
PNP channel length	0.2 µm	
PNP channel width	20 µm	

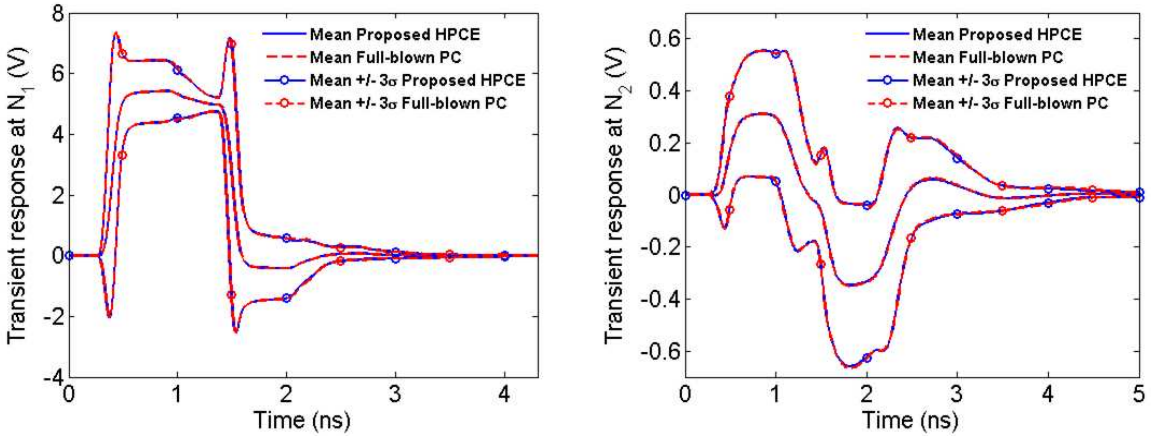
In order to demonstrate the accuracy of the proposed adaptive HPCE approach, the mean and standard deviation (σ) of the transient responses at the output nodes N_1 and N_2 of Fig. 4.6 are computed using the methodology described in Section 4.5. By running the closed-form technique of Section 4.5.3, five critical hyperbolic factors are obtained, these values are **u=[**$u_0 \approx 0$, 0.5, 0.69, 0.79, 1]. In this example, the threshold on the accuracy constraint of standard deviation is set as 5e-3; hence, the adaptive HPCE approach runs until it meets this accuracy. Table 4.6 shows the enrichment at different steps. This table is

filled by finding the time average of (4.29) at different steps. As it can be seen, the accuracy is met when the hyperbolic factor is equal to the penultimate critical hyperbolic factor i.e. u = 0.79. Nevertheless, we have continued the approach for u = 1 for illustrations purposes. The penultimate critical hyperbolic factor u=0.79 is results in selection of $\tilde{M}$ = 296 sample nodes while the nonintrusive based ST approach needs P+1 = 1001 sample nodes.

Table 4.6 Time average of enrichment of responses of example 1 in Section 4.6 at different steps.

	After step 2 W. 86 bases	After step 2 W. 176 bases	After step 2 W. 296 bases	After step 2 W. 1001 bases
Enrichment of response at N_1	0.0134	0.0064	0.0038	9.04e-4
Enrichment of response at N_2	0.0146	0.0057	0.0033	9.14e-4

Next the mean and standard deviation results are compared against those obtained using the nonintrusive ST based approach. The comparison of the above results is shown in Fig. 4.8 where both PC approaches exhibit good agreement.

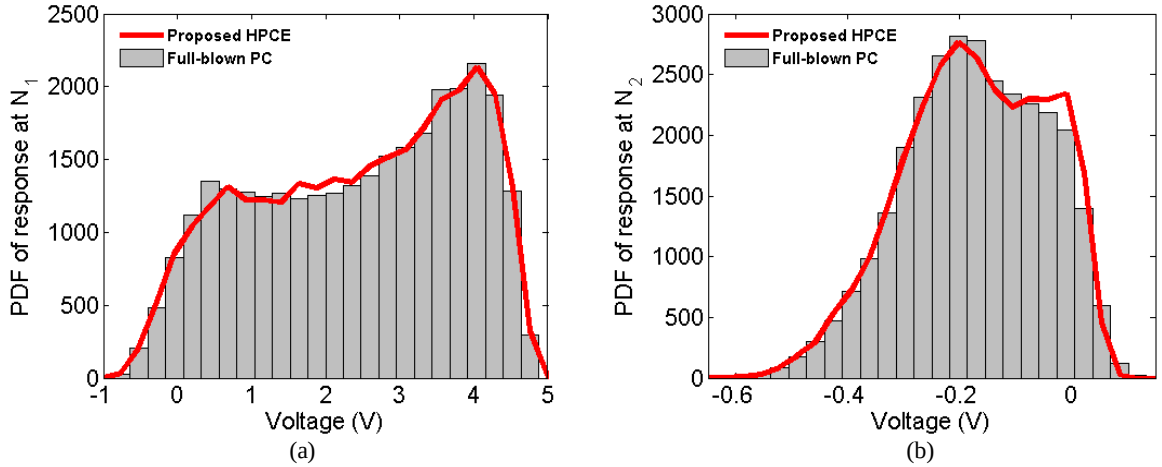


(a) (b)

Fig. 4.8: Comparison of the statistics of the transient response of Example 1 in Section 4.6 obtained using the proposed adaptive HPCE approach and the nonintrusive ST based approach. (a) Statistical results of the transient response at N_1. (b) Statistical results of the transient response at N_2.

Next the PDFs of the transient response at outputs N_1 and N_2 are obtained at the time point of maximum standard deviation of transient response at N_1 (t = 1.5ns) and at the time point of maximum standard deviation of transient response at N_2 (t = 2.26ns) respectively. The PDF at these time points is obtained using resultant coefficients of the proposed adaptive HPCE approach and the nonintrusive ST

based approach and by taking advantage of the analytical technique explained in Section 2.1.4.3. Using this technique, 30000 analytical samples are generated and the PDFs are generated from them. The result is presented in Fig. 4.9, where both approaches demonstrate a good agreement.



(a) (b)

Fig. 4.9: PDFs of Example 1 in Section 4.6 obtained using 30,000 samples. (a) PDF of response at N_1 at the time point of maximum standard deviation of response at N_1 (t = 1.5 ns). (b) PDF of response at N_2 at the time point of maximum standard deviation of response at N_2 (t = 2.26 ns).

Finally, the CPU times taken by the proposed adaptive HPCE approach and the nonintrusive ST based approach are 65 seconds and 220 seconds respectively. It is observed that the HPCE approach is more efficient compared to the nonintrusive ST approach and represents a speedup greater than 3 over this method. This speedup is due to the smaller number of SPICE simulations involved.

4.6.2 Example 2

The objective of this example is to compare the performance of the proposed adaptive HPCE approach with the nonintrusive ST based approach. For this purpose, the multiconductor transmission line (MTL) network of Fig. 4.10 terminated by inverters made up of SPICE level 49 CMOS transistor models is considered. The lengths of the MTL networks are set to 6 cm and their layout and geometric dimensions are shown in Fig. 4.11. This network is driven by a voltage source with a trapezoidal waveform of rise/fall time T_r = 0.1 ns, pulse width T_w = 1 ns and amplitude of 5V. The uncertainty in the network is introduced via thirteen random variables (n = 13) whose characteristics are listed in Table 4.7. For this example a Legendre PC expansion of degree m = 4 is considered.

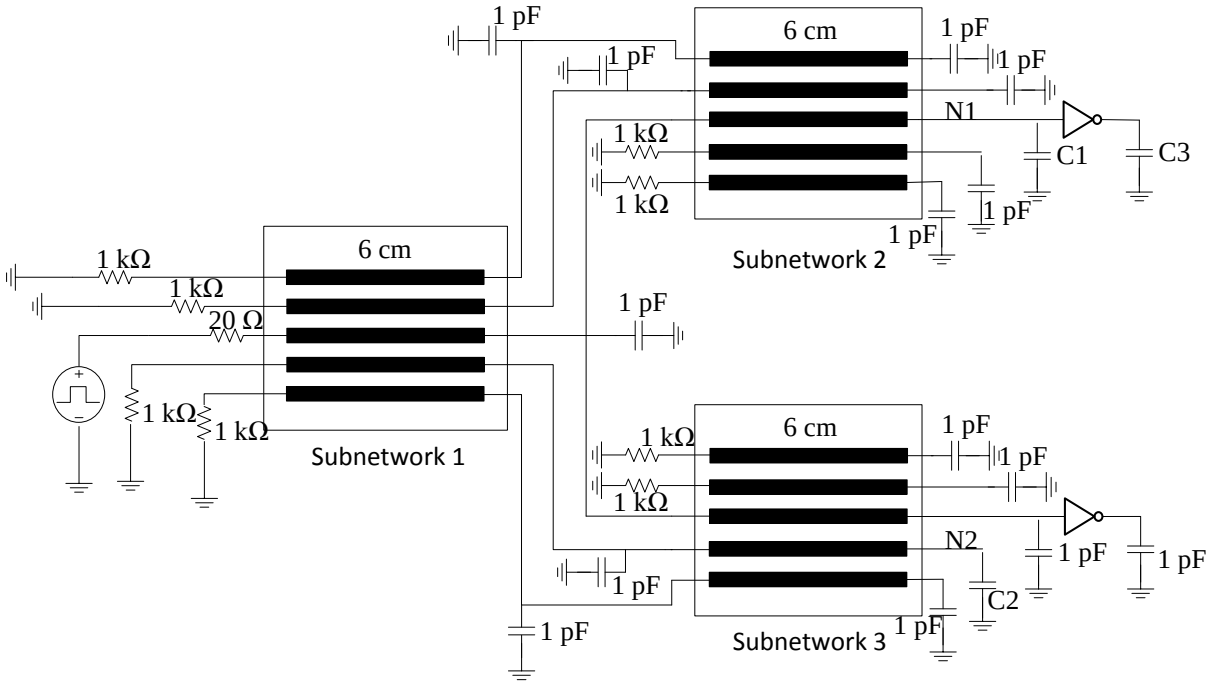


Fig. 4.10: Multiconductor transmission line (MTL) network for Example 2 in Section 4.6.

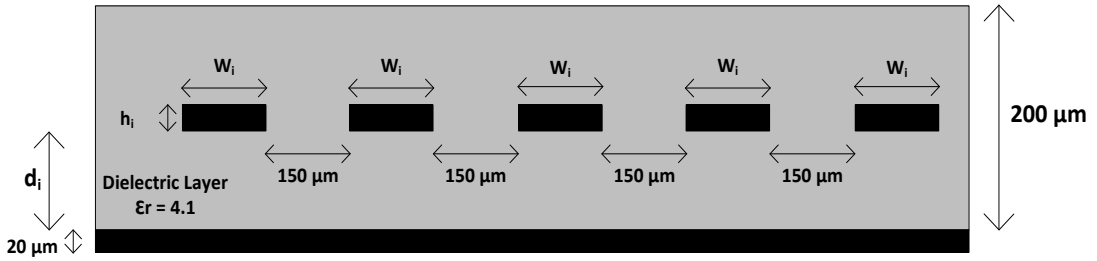


Fig. 4.11: Geometrical and physical layout of transmission lines of Example 2 in Section 4.6 where i = [1,2,3] refers to the i-th subnetwork.

In order to demonstrate the accuracy of the proposed adaptive HPCE approach, the mean and standard deviation (σ) of the transient responses at the output nodes N_1 and N_2 of Fig. 4.10 are computed using the methodology described in Section 4.5. By running the closed-form technique of Section 4.5.3, five critical hyperbolic factors are obtained, these values are same as the case of ten random variables and equal to $\mathbf{u}=[u_0 \approx 0, 0.5, 0.69, 0.79, 1]$. In this example, the threshold on the accuracy constraint of standard deviation is set as 1e-4; hence, the adaptive HPCE approach runs until it meets this accuracy. Table 4.8 shows the enrichment at different steps. This table is filled by finding the time average of (4.29) at different steps. As it can be seen, the accuracy is met when the hyperbolic factor is equal to the

penultimate critical hyperbolic factor i.e. u = 0.79. Nevertheless, we have continued the approach for u = 1 for illustrations purposes. The penultimate critical hyperbolic factor u=0.79 results in selection of $\tilde{M}$ = 573 sample nodes while the nonintrusive based ST approach needs P+1 = 2380 sample nodes.

Table 4.7 Characteristics of random variables of example 2 in Section 4.6 (Fig. 4.10 and Fig 4.11)

Random Variables	Mean	% Standard Deviation
d_1	100 μm	+/- 30 % Uniform Distribution
d_2	140 μm	
d_3	70 μm	
W_1	150 μm	
W_2	130 μm	
W_3	170 μm	
h_1	30 μm	
h_2	20 μm	
h_3	40 μm	
Transmission line conductivity	5.8e7	
C_1	1 pF	
C_2	0.5 pF	
C_3	1.5 pF	

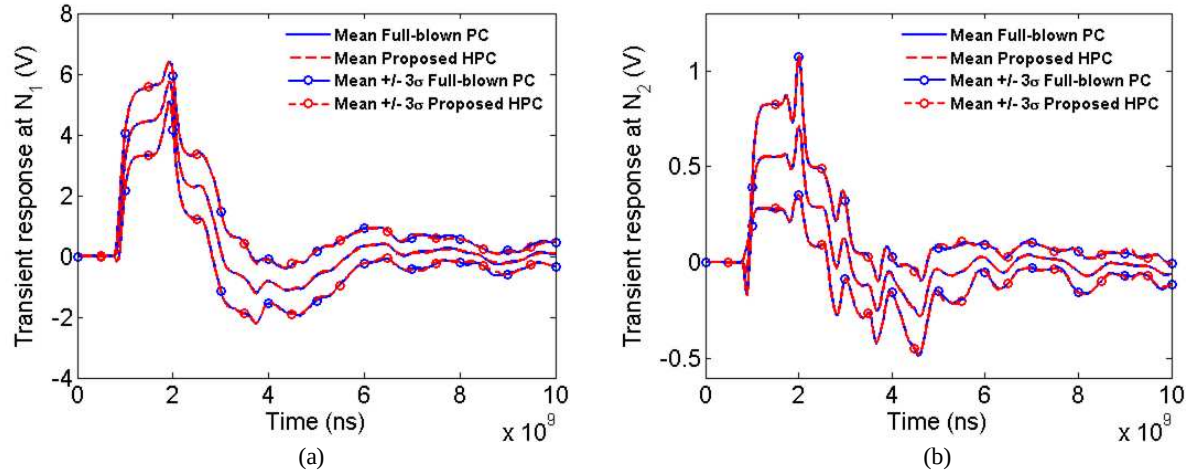
Table 4.8 Time average of enrichment of responses of example 2 in Section 4-6 at different steps.

	After step 2 W. 131 bases	After step 2 W. 287 bases	After step 2 W. 573 bases	After step 2 W. 2380 bases
Enrichment of response at N_1	7.52E-04	3.36E-04	6.95E-05	2.55E-05
Enrichment of response at N_2	9.18E-04	4.11E-04	7.67E-05	1.91E-05

Next the mean and standard deviation results are compared against those obtained using the nonintrusive ST based approach. The comparison of the above results is shown in Fig. 4.12 where both PC approaches exhibit good agreement.

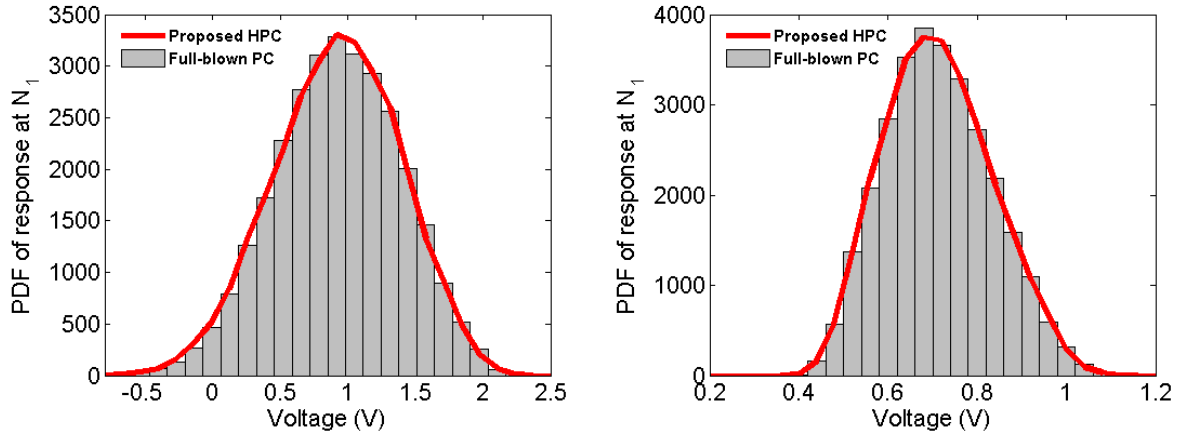
Next the PDFs of the transient response at outputs N_1 and N_2 are obtained at the time point of maximum standard deviation of transient response at N_1 (t = 2.89ns) and at the time point of maximum standard deviation of transient response at N_2 (t = 2.01ns) respectively. The PDF at these time points is obtained using resultant coefficients of the proposed adaptive HPCE approach and the nonintrusive ST based approach and by taking advantage of the analytical technique explained in Section 2.1.4.3. Using

this technique, 30000 analytical samples are generated and the PDFs are generated from them. The result is presented in Fig. 4.13, where both approaches demonstrate a good agreement.



(a) (b)

Fig. 4.12: Comparison of the statistics of the transient response of Example 2 in Section 4.6 obtained using the proposed adaptive HPCE approach and the nonintrusive ST based approach. (a) Statistical results of the transient response at N_1. (b) Statistical results of the transient response at N_2.



(a) (b)

Fig. 4.13: PDFs of Example 2 in Section 4.6 obtained using 30,000 samples. (a) PDF of response at N_1 at the time point of maximum standard deviation of response at N_1 (t = 2.89ns). (b) PDF of response at N_2 at the time point of maximum standard deviation of response at N_2 (t = 2.01ns).

Finally, the CPU times taken by the proposed adaptive HPCE approach and the nonintrusive ST based approach are 974 seconds and 4046 seconds respectively. It is observed that the HPCE approach is more efficient compared to the nonintrusive ST approach and represents a speedup greater than 4 over this method. This speedup is due to the smaller number of SPICE simulations involved.

Table 4.9 Scaling of CPU time costs using adaptive HPCE and nonintrusive ST for example 3 in Section 4.6

Number of RVs (Random Variables)	CPU Time (Proposed adaptive HPCE approach) (s)	CPU Time (Nonintrusive ST based approach) (s)
5 (d_1, d_2, d_3,w_1, w_2)	104	214
6 (d_1, d_2, d_3,w_1, w_2, w_3)	153	357
7 (d_1, d_2, d_3,w_1, w_2, w_3,h_1)	216	561
8 (d_1, d_2, d_3,w_1, w_2, w_3,h_1,h_2)	294	841
9 (d_1, d_2, d_3,w_1, w_2, w_3,h_1,h_2,h_3)	389	1215
10 (d_1, d_2, d_3,w_1, w_2, w_3,h_1,h_2,h_3,g^*)	503	1701
11 (d_1, d_2, d_3,w_1, w_2, w_3,h_1,h_2,h_3,g,C_1)	637	2320
12 (d_1, d_2, d_3,w_1, w_2, w_3,h_1,h_2,h_3,g,C_1,C_2)	794	3094
13 (d_1, d_2, d_3,w_1, w_2, w_3,h_1,h_2,h_3,g,C_1,C_2,C_3)	974	4046

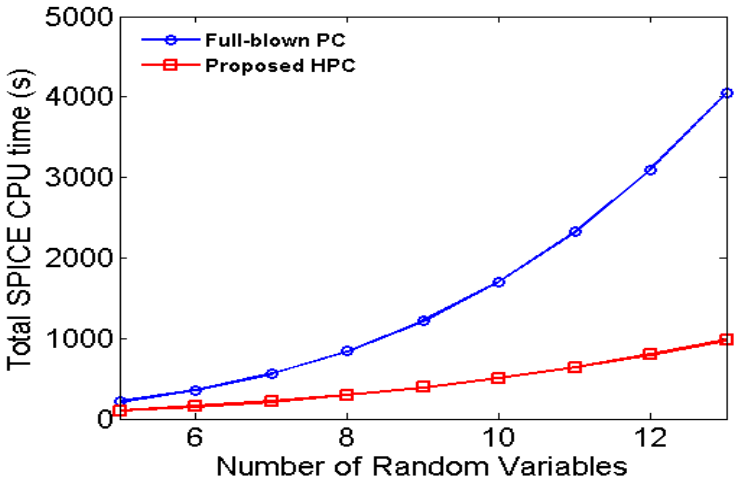


Fig. 4.14: Scaling of overall CPU time costs for example 3 in Section 4.6 for the proposed adaptive HPCE approach, and the full-blown PC nonintrusive ST based approach with increasing number of random variables.

4.6.3 Example 3

The objective of this example is to compare the scalability of the proposed adaptive HPCE approach against that of the nonintrusive ST based approach. For this purpose the same multiconductor transmission line (MTL) network of Fig. 4.10 is considered. The uncertainty in the network is expanded to include between five and thirteen random variables (n = 5…13) whose characteristics are listed in Table 4.7 and a Legendre PC expansion of degree m = 4 is considered.

In order to demonstrate the scalability of the proposed work the number of random dimensions is progressively increased from 5 to 13 as shown in Table 4.9. For the same test cases of Table 4.9, the total PC problem is solved using both of the adaptive HPCE approach and the nonintrusive ST based approach.

The CPU time incurred by each approach for each test case is noted in Table 4.9 and scaling of the overall CPU costs is demonstrated in Fig. 4.14. It is observed from Fig. 4.14 that the proposed HPCE approach outperforms the nonintrusive ST based approach when number of random variables is relatively high. This is due fact that based on (4.35) and (4.40) the proposed approach is in the order of $O(n)$ faster than the ST based approach, which is more visible when number of random variables increases.

CHAPTER V: CONCLUSION

With the advent of sub-90 nm technology, the effect of random manufacturing process variations and unpredictable operating conditions on the performance of microwave and radio-frequency (RF) circuits can no longer be neglected. In order to include the impact of such parametric uncertainty in the design optimization process, numerical methods to predict the effect of this uncertainty on the circuit response are required. The generalized polynomial chaos (PC) theory has emerged as a highly robust and versatile approach for the statistical analysis of high-speed circuits and electromagnetic (EM) systems. Typically, PC approaches model the uncertainty in network responses as a polynomial expansion of predefined orthonormal functions of the input random variables.

In this thesis, a number of techniques for advancement of PC approaches are suggested. The first major contributions is a novel sparse linear regression approach for the fast multidimensional uncertainty quantification of high-speed circuits within a SPICE environment. The next contribution is another approach which exploits and expedited version of D-optimal design of experiments to accurately evaluate the PC coefficients of the network responses. This approach provides fast search algorithms to identify the design of experiments from large multi-dimensional random spaces. Both approaches are based on modified versions of the Fedorov search algorithm that quickly locates a small set of regression nodes within the random space. By probing the PC expansion of the circuit current and voltage quantities at these sparse nodes the uncertainty in the circuit response can be efficiency quantified. Overall, the proposed approaches provides excellent CPU savings for high-dimensional circuit problems compared to contemporary intrusive and nonintrusive PC methods.

Next a novel improvement on truncation of the polynomial bases in the generalized PC theory is suggested. This approach promises to alleviate the exponential or near exponential scaling of computational costs with respect to number of random variables, by adopting a new hyperbolic truncation scheme in place of the conventional linear truncation scheme used for PC expansions. This scheme is

based on the sparsity of effects principle which states that the low-degree interactions between the random variables are statistically more significant than the higher-degree interactions. This suggests that if the high-degree multidimensional bases can be pruned from the full-blown PC expansion, a sparser expansion can be achieved without significantly sacrificing the accuracy. In order to adaptively generate such a sparse PC expansion based on only the low-degree interactions, an adaptive methodology is proposed in this thesis. All contributions in this thesis are validated through multiple numerical examples.

REFERENCES

[1] F. N. Najm, *Circuit Simulation*. Hoboken, NJ: Wiley, 2010

[2] C. R. Paul, *Introduction to Electromagnetic Compatibility*. 2nd edition, Hoboken, NJ: Wiley, 2006

[3] *HSPICE Reference Manual Version E 2010.12*, Synopsis Inc., Santa Clara, CA, 2010

[4] G.S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*. New York, NY: Springer-Verlag, 1996

[5] Q. Zhang, J. J. Liou, J. McMacken, J. Thomson and P. Layman, "Development of robust interconnect model based on design of experiments and mulitobjective optimization," *IEEE Trans. Electron Devices*, vol. 48, no. 9, pp. 1885–1981, Sept. 2001

[6] Y. Massoud and A. Nieuwoudt, "Modeling and design challenges and solutions for carbon-nanotube based interconnect in future high performance integrated circuits," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 3, pp. 155–196, July 2006

[7] A. Nieuwoudt and Y. Massoud, "On the impact of process variations for carbon nanotube bundles for VLSI interconnect," *IEEE Trans. Electron. Devices*, vol. 54, no. 3, pp. 446–455, March 2007

[8] A. Nieuwoudt and Y. Massoud, "On the optimal design, performance, and reliability of future carbon nanotube-based interconnect solutions," *IEEE Trans. Electron. Devices*, vol. 55, no. 8, pp. 2097–2110, Aug. 2008

[9] J. F. Swidzinski and K. Chang, "Nonlinear statistical modeling and yield estimation technique for use in Monte Carlo simulations [microwave devices and ICs]," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2316-2324, Dec. 2000

[10] P. Manfedi, "High-speed interconnect models with stochastic parameter variability," Ph.D. dissertation, Informat. Comm. Tech., Politecnico di Torino, Turin, Italy, 2013

[11] S. Vrudhula, J. M. Wang and P. Ghanta, "Hermite polynomial based interconnect analysis in the presence of process variations," *IEEE Trans. Computer Aided Design*, vol. 25, no. 10, pp. 2001–2011, Oct. 2006

[12] I. S. Stievano, P. Manfredi and F. G. Canavero, "Stochastic analysis of multiconductor cables and interconnects," *IEEE Trans. Electromagn. Compatibility*, vol. 53, no. 2, pp. 501–507, May 2011

[13] I. S. Stievano, P. Manfredi and F. G. Canavero, "Parameters variability effects on multiconductor interconnects via Hermite polynomial chaos," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 1, no. 8, pp. 1234–1239, Aug. 2011

[14] D. Vande Ginste, D. De Zutter, D. Deschrijver, T. Dhaene, P. Manfredi and F. Canavero, "Stochastic modeling-based variability analysis of on-chip interconnects," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 3, no. 7, pp. 1182–1192, Jul. 2012

[15] I. S. Stievano, P. Manfredi and F. G. Canavero, "Carbon nanotube interconnects: process variation via polynomial chaos," *IEEE Trans. Electromagn. Compatibility*, vol. 54, no. 1, pp. 140–148, Feb. 2012

[16] A. Biondi, D. Vande Ginste, D. De Zutter, P. Manfredi and F. Canavero, "Variability analysis of interconnects terminated by general nonlinear loads," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 2, no. 7, pp. 1244–1251, Jul. 2013

[17] P. Manfredi, D. Vande Ginste, D. De Zutter and F. Canavero, "Uncertainty assessment in lossy and dispersive lines in SPICE-type environments," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 2, no. 7, pp. 1252–1258, Jul. 2013

[18] P. Manfredi, D. Vande Ginste, D. De Zutter and F. Canavero, "Stochastic modeling of nonlinear circuits via SPICE-compatible spectral equivalents," *IEEE Trans. Circuits Syst.*, vol. 61, no. 7, pp. 2057–2065, Jul. 2014

[19] M. R. Rufuie, E. Gad, M. Nakhla R. Achar, "Generalized Hermite polynomial chaos for variability analysis of macromodels embedded in nonlinear circuits," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 4, no. 4, pp. 673–684, April 2014

[20] 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. Comp., Packag. and Manuf. Technol.*, vol. 4, no. 10, pp. 1634–1647, Oct. 2014

[21] D. Spina, F. Ferranti, G. Antonini, T. Dhaene and L. Knockaert, "Efficient variability analysis of electromagentic systems via polynomial-chaos and model order reduction," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 4, no. 6, pp. 1038–1051, June 2014

[22] D. Spina, F. Ferranti, T. Dhaene, L. Knockaert, G. Antonini, and D. Vande Ginste, "Variability analysis of multiport systems via polynomial-chaos expansion," *IEEE Trans. Microwave Theory Tech.*, vol. 60, no. 8, pp. 2329–2338, Aug. 2012

[23] A. Rong and A. C. Cangellaris, "Transient analysis of distributed electromagnetic systems exhibiting stochastic variability in material parameters," in *Proc. XXXth General Assembly and Scientific Symp.*, Aug. 2011, pp. 1-4

[24] Md. A. H. Talukder, M. Kabir, S. Roy and R. Khazaka, "Efficient generation of macromodels via the Loewner matrix approach for the stochastic analysis of high-speed passive distributed networks," in *Proc. IEEE 18th Workshop on Signal and Power Integrity*, May 2014, pp. 1-4

[25] Md. A. H. Talukder, M. Kabir, S. Roy and R. Khazaka, "Efficient stochastic transient analysis of high-speed passive distributed networks using Loewner matrix macromodels," in *Proc. IEEE Intl. Conf. Signal and Power Integrity*, August 2014, pp. 1-4

[26] A. K. Prasad, and S. Roy, "Multidimensional variability analysis of complex power distribution networks via scalable stochastic collocation approach," to appear in *IEEE Trans. Comp., Packag. and Manuf. Technol.*, 2015

[27] D. Spina, F. Ferranti, G. Antonini, T. Dhaene and L. Knockaert, "Non-intrusive polynomial chaos-based stochastic macromodeling of multiport systems," in *Proc. IEEE 18th Workshop on Signal and Power Integrity*, May 2014, pp. 1-4

[28] D. Spina, D. De Jonghe, D. Deschrijver, G. Gielen, L. Knockaert, and T. Dhaene, "Stochastic macromodeling of nonlinear systems via polynomial-chaos expansion and transfer function trajectories," *IEEE Trans. Microwave Theory Tech*, vol. 62, no. 7, pp. 1454–1460, July 2014

[29] M. Ahadi, M. Kabir, E. Chobanyan, A. Smull, Md. A. H. Talukder, S. Roy, R. Khazaka and B. M. Notaros, "Non-intrusive pseudo spectral approach for stochastic macromodeling of EM systems using deterministic full-wave solvers," in *Proc. 23rd IEEE Conf. Electric. Perform. Electron. Packag.*, October 2014, pp. 235-238

[30] A. Rong and A. C. Cangellaris, "Interconnect transient simulation in the presence of layout and routing uncertainty," in *Proc. 20th IEEE Conf. Electric. Perform. Electron. Packag.*, October 2011, pp. 157-160

[31] J. S. Ochoa and A. C. Cangellaris, "Analysis of the impact of interconnect routing variability on signal degradation," in *Proc. 21st IEEE Conf. Electric. Perform. Electron. Packag.*, October 2012, pp. 315-318

[32] P. Sumant, H. Wu, A. Cangellaris, and N. Aluru, "Reduced-order models of finite element approximations of electromagnetic devices exhibiting statistical variability," *IEEE Trans. Antennas Propag.*, vol. 60, no. 1, pp. 301–309, Jan. 2012

[33] X. Chen, J. S. Ochoa, J. E. Schutt-Aine and A. C. Cangellaris, "Optimal relaxation of I/O electrical requirements under packaging uncertainty by stochastic methods," in *Proc. 64th Electronic Comp. Tech. Conf.*, May 2014, pp. 717–722

[34] A. C. Yucel, H. Bagci, J. S. Hesthaven and E. Michielssen, "A fast Stroud-based collocation for statistically characterizing EMI/EMC phenomena on complex platforms", *IEEE Trans. Electromagn. Compatibility*, vol. 51, no. 2, pp. 301-311, May 2009

[35] A. C. Yucel, H. Bagci and E. Michielssen, "An adaptive multi-element probabilistic collocation method for statistical EMC/EMI characterization," *IEEE Trans. Electromagn. Compatibility*, vol. 55, no. 6, pp. 1154–1168, Dec. 2013

[36] M. Ahadi, M. Vempa, and S. Roy, "Efficient multidimensional statistical modeling of high speed interconnects in SPICE via stochastic collocation using Stroud cubature," in *Proc. IEEE Symp. Electromagn. Compatibility and Signal Integrity*, March 2015, pp. 300-305

[37] P. Manfredi, D. Vande Ginste, D. De Zutter and F. Canavero, "Generalized decoupled polynomial chaos for nonlinear circuits with many random parameters," *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 8, pp. 505–507, Aug. 2015

[38] M. Ahadi, M. Kabir, S. Roy and R. Khazaka, "Fast multidimensional statistical analysis of microwave networks via Stroud cubature approach," to appear in *Proc. IEEE International Conf. on Numerical Electromagn., Multiphysics Modeling and Optimiz.*, August 2015

[39] A. K. Prasad, M. Ahadi, B. S. Thakur and S. Roy, "Accurate polynomial chaos expansion for variability analysis using optimal design of experiments," *IEEE International Conf. on Numerical Electromagn., Multiphysics Modeling and Optimiz.*, August 2015

[40] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel and L. Daniel, "Stochastic testing method of transistor level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Computer Aided Design*, vol. 32, no. 10, pp. 1533-1545, Oct. 2013

[41] P. Manfredi and F. Canavero, "Efficient statistical simulation of microwave devices via stochastic testing-based circuit equivalents of nonlinear components," *IEEE Trans. Microw. Theory Tech.*, vol. 63, no. 5, pp. 1502-1511, May 2015

[42] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel and L. Daniel, "Calculation of generalized polynomial-chaos basis functions and Gauss quadrature rules in hierarchical uncertainty quantification," *IEEE Trans. Computer Aided Design*, vol. 33, no. 5, pp. 728-740, May 2014

[43] Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel, "Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition," *IEEE Trans. Computer Aided Design*, vol. 34, no. 1, pp. 63-76, Jan. 2015

[44] A. C. Yucel, H. Bagci, and E. Michielssen, "An ME-PC enhanced HDMR method for efficient statistical analysis of multiconductor transmission line networks," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 5, no. 5, pp. 685-696, May 2015

[45] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. New Jersey: Princeton University Press, 2010

[46] M. S. Eldred, "Recent advance in non-intrusive polynomial-chaos and stochastic collocation methods for uncertainty analysis and design," in *Proc. 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conf.*, May 2010, pp. 1-37

[47] C. F. Jeff Wu; M. Hamada, *Experiments: Planning, analysis, and parameter design optimization*. New York: Wiley, p. 112, 2000

[48] D.C. Montgomery, *Design and Analysis of Experiment*. 4th ed. New York, NY: Wiley, 1997

[49] D. Xiu, "Fast numerical methods for stochastic computations: a review." *Communications in computational physics*, vol. 5, no. 2-4, pp. 242-272, 2009

[50] C. Chauviere, H. S. Jan , C. W. Lucas. "Efficient computation of RCS from scatterers of uncertain shapes." *Antennas and Propagation, IEEE Transactions on* 55.5 : 1437-1448, 2007

[51] A. H. Stroud, "Remarks on the disposition of points in numerical integration formulas." *Mathematical Tables and Other Aids to Computation* 11.60, 257-261, 1957

[52] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, pp. 897–936, 1938

[53] R. Cameron and W. Martin, "The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals," *The Annals of Mathematics*, vol. 48, no. 2, pp. 385–392,1947

[54] R. Askey and J. Wilson, "Some basic hypergeometric polynomials that generalize Jacobipolynomials," *Memoirs of the American Mathematical Society*, vol. 54, no. 319, Mar. 1985.

[55] D. Xiu and G. E. Karniadakis, "TheWiener-Askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002

[56] X.Wan, G. E. Karniadakis, "Multi-element generalized polynomial chaos for arbitrary probabilitymeasures," *SIAM Journal on Scientific Computing*, vol. 28, no. 3, pp. 901–928, 2006

[57] P. M. Congedo, G. Gianluca, I. Gianluca. "On the use of high-order statistics in robust design optimization." *6th European Conference on Computational Fluid Dynamics (ECFD VI)*. 2014

[58] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. 3rd edition, New York: Cambridge University Press, 2007

[59] G. H. Golub and J. H. Welsch, "Calculation of Gauss quadrature rules," *Mathematics of Computation*, vol. 23, pp. 221–230, 1969

[60] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover,1972

[61] Ahadi, Majid, and Sourajeet Roy. "Sparse Linear Regression (SPLINER) Approach for Efficient Multidimensional Uncertainty Quantification of High-Speed Circuits." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, 2015

[62] A. S. Goldberger, *Classical Linear Regression Econometric Theory*, New York, NY: Wiley, pp. 156–212, 1964

[63] N.-K. Nguyen and A. J. Miller, "A review of some exchange algorithms for constructing discrete D-optimal designs," *Comput. Statistics and Data Analysis*, vol. 14, pp. 489-498, 1992

[64] S. Zein, B. Colson, and F. Glineur, "An efficient sampling method for regression-based polynomial chaos expansion," *Commun. Comput. Physics*, vol. 13, no. 4, pp. 1173-1188, April 2013

[65] A. J. Miller and N.-K. Nguyen, "A Fedorov exchange algorithm for D-optimal design," J. Royal Stat. Society, vol. 43, no. 4, pp. 669-677, 1994

[66] V. V. Fedorov, *Theory of Optimal Experiments*, New York, NY: Academic Press, 1972

[67] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM J. Scientific Comput.*, vol. 27, no. 3, pp. 1118-1139, March 2006

[68] N. Agarwal and N. R. Aluru, "Stochastic analysis of electrostatic MEMS subjected to parameter variations", *J. Microelec. Systs.*, vol. 18, no. 6, pp. 1454–1468, Dec. 2009

[69] S. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Math. Dokl.*, vol. 4, pp. 240–243, 1963

[70] R. Cools and P. Rabinowitz, "Monomial cubature rules since Stroud: A compilation," *J. Comput. Appl. Mathematics*, vol. 48, no. 3, pp. 309-326, Nov. 1993

[71] R. Cools, “Monomial cubature rules since Stroud: a compilation – part 2,” *J. Comput. Appl. Mathematics*, vol. 112, pp. 21-27, 1999

[72] A. K. Prasad, M. Ahadi and S. Roy, “Multidimensional Uncertainty Quantification of Microwave/RF Networks using Linear Regression and Optimal Design of Experiments,” to appear in *IEEE Trans. on Microwave Theory and Techniques*, 2016

[73] [48] R. D. Cook and C. J. Nachtsheim, “A comparison of algorithms for constructing exact D-optimal designs,” *Technometrics*, vol. 22, no. 3, pp. 315-324, Aug. 1980

[74] N. S. Kuek, A. C. Liew, E. Shamiloglu, and J. Rossi, “Circuit modeling of nonlinear lumped element transmission lines including hybrid lines,” *IEEE Trans. Plasma Science*, vol. 40, no. 10, pp. 2523–2534, Oct. 2012

[75] G. Blatman, “Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis,” Ph.D. dissertation. Clermont-Ferrand 2, 2009

[76] R. Merris, *Combinatorics*, 2nd ed. Hoboken, NJ: Wiley, p. 11, 2003

[77] J. An, and O. Art, "Quasi-regression." *Journal of complexity* 17.4, pp. 588-607, 2001

[78] R. A. Todor, and S. Christoph, "Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients." *IMA Journal of Numerical Analysis*, 27.2, pp. 232-261, 2007

[79] M. Ahadi, A. K. Prasad, S. Roy “Hyperbolic Polynomial Chaos Expansion (HPCE) and its Application to Statistical Analysis of Nonlinear Circuits,” *IEEE Intl. Conf. Signal and Power Integrity*, 2016

APPENDIX A: LIST OF PUBLICATIONS

Conference Papers:

- M. Ahadi, M. Kabir , et al. “Non-Intrusive Pseudo Spectral Approach for Stochastic Macromodeling of EM Systems using Deterministic Full-wave Solvers”, *IEEE Conference on Electrical Performance of Electronic Packaging and Systems,* 2014
- A. K. Prasad, M. Ahadi, S. Roy “Polynomial Chaos Based Variability Analysis of Power Distribution Networks Using a 3D Topology of Multiconductor Transmission Lines”, *IEEE Conference on Electrical Performance of Electronic Packaging and Systems,* 2014
- M. Ahadi, M. Vempa, S. Roy “Efficient Multidimensional Statistical Modeling of High Speed Interconnects in SPICE via Stochastic Collocation using Stroud Cubature”, *IEEE International Symposium on Electromagnetic Compatibility* (Invited paper to special session), 2015
- M. Ahadi, M.Kabir, S. Roy, R. Khazaka “Fast Multidimensional Statistical Analysis of Microwave Networks via Stroud Cubature Approach”, *IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization*, 2015
- A. K. Prasad, M. Ahadi, B. S. Thakur, S. Roy “Accurate Polynomial Chaos Expansion for Variability Analysis using Optimal Design of Experiments”, *IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization*, 2015
- M. Ahadi, A. K. Prasad, S. Roy “Hyperbolic Polynomial Chaos Expansion (HPCE) and its Application to Statistical Analysis of Nonlinear Circuits,” *IEEE International Conference on Signal and Power Integrity,* 2016

Journal Papers:

- M. Ahadi, S. Roy, "Sparse Linear Regression (SPLINER) Approach for Efficient Multidimensional Uncertainty Quantification of High-Speed Circuits", *IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016
- A.K.Prasad, M. Ahadi, S.Roy, "Multidimensional Uncertainty Quantification of Microwave/RF Networks using Linear Regression and Optimal Design of Experiments", *IEEE Transactions on Microwave Theory and Techniques* (special issue), Accepted in June 2016
- M. Ahadi, S. Roy, "Adaptive Hyperbolic polynomial chaos expansion (HPCE)", *IEEE Transactions on components packaging and manufacturing technology*, Submitted in June 2016 (Under review)