

THESIS

PERFORMANCE ASSESSMENT OF MULTI-WALLED  
CARBON NANOTUBE INTERCONNECTS USING  
ADVANCED POLYNOMIAL CHAOS SCHEMES

Submitted by

Sakshi Bhatnagar

Department of Electrical and Computer Engineering

In partial fulfilment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2019

Master's Committee:

Advisor: Mahdi Nikdast

Ali Pezeshki  
Donald Estep

Copyright by Sakshi Bhatnagar 2019

All Rights Reserved

## ABSTRACT

### PERFORMANCE ASSESSMENT OF MULTI-WALLED CARBON NANOTUBE USING ADVANCED POLYNOMIAL CHAOS SCHEMES

With the continuous miniaturization in the latest VLSI technologies, manufacturing uncertainties at nanoscale processes and operations are unpredictable at chip level, packaging level and at board levels of integrated systems. To overcome such issues, simulation solvers to model forward propagation of uncertainties or variations in random processes at device level to the network response are required. Polynomial Chaos Expansion (PCE) of the random variables is the most common technique to model the unpredictability in the systems. Existing methods for uncertainty quantification have a major drawback that as the number of random variables in a system increases, its computational cost increases in a polynomial fashion.

In order to alleviate the poor scalability of standard PC approaches, predictor-corrector polynomial chaos scheme and hyperbolic polynomial chaos expansion (HPCE) scheme are being proposed in this thesis. In predictor-corrector polynomial scheme, low-fidelity meta-model is generated using Equivalent Single Conductor (ESC) approximation model and then its accuracy is enhanced using low order multi-conductor circuit (MCC) model called as corrector model. In HPCE, sparser polynomial expansion is generated based on hyperbolic criterion. These schemes result into immense reduction in CPU cost and speed. This thesis presents the novel approach to quantify the uncertainties in multi-walled carbon nanotubes using these schemes whose accuracy and validation are shown using numerical examples.

## ACKNOWLEDGEMENTS

I would like to thank all the key entities whose encouragement and support has made the completion of this thesis possible. First, I would like to express my sincere gratitude to Dr. Sourajeet Roy, without whose invaluable guidance this undertaking would not have been possible in the first place. His knowledge in this research area is unparalleled, and his expert counsel at critical junctures of the research process and incredible patience with my progress has enabled me to bring my thesis to its fruitful conclusion. I can only hope to match the energy with which he juggles a daunting workload of managing research, classes and students with a smile.

Next, I would like to give heartiest thanks to Dr. Mahdi Nikdast for supporting me as an advisor and providing essential and valuable inputs for my thesis.

Then I would like to thank Dr. Ali Pezeshki and Dr. Donald Estep for their valuable time to preside in my committee.

I also express my gratitude to my research lab mates for helping me whenever needed in every phase of my research. Apart from them, I sincerely thank my friends for their immense moral support throughout my journey of research.

Last but not the least, I would like to thank my family for their stupendous support, patience and trust that kept me motivated in my work all the time.

## TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
LIST OF PUBLICATIONS.....	viii
Chapter 1: Introduction.....	1
1.1 Problem Statement.....	1
1.2 Goals.....	4
1.3 Organization of the Thesis.....	5
Chapter 2: Existing Methods of Uncertainty Quantification.....	7
2.1 Stochastic Modified Nodal Analysis.....	7
2.2 Monte Carlo (MC).....	9
2.3 General Polynomial Chaos (gPC).....	9
2.3.1 Statistics calculation using PC coefficients.....	11
2.3.1.1 Calculation of arithmetic mean of the outputs.....	11
2.3.1.2 Calculation of standard deviation and variance of the outputs.....	12
2.3.1.3 Probability density function (PDF) and other higher order moments.....	13
2.3.2 One dimensional orthonormal polynomials.....	15
2.3.3 Multi-dimensional orthonormal polynomials generation.....	16
2.4 Non-Intrusive and Intrusive PC Methods.....	20
2.4.1 Intrusive methods.....	20
2.4.2 Non-intrusive methods.....	21
2.4.2.1 Pseudo spectral polynomial chaos approach.....	21
2.4.2.2 Standard linear regression method.....	23
2.4.2.2.1 Advantages and drawbacks of linear regression.....	26
2.4.2.2.2 Selecting regression nodes.....	27
2.4.3 D-optimality criterion.....	27

2.4.4 Fedorov search algorithm.....	29
2.4.5 Cost of computation for search algorithm.....	30
2.4.6 Accelerating search-algorithm for high dimension random spaces.....	32
2.4.6.1 Efficiency of K-worst node substitution in search algorithm.....	33
Chapter 3: Predictor-Corrector Polynomial Chaos Scheme.....	34
3.1 Introduction.....	34
3.2 Proposed Predictor-Corrector Scheme.....	35
3.2.1 Construction of low-fidelity predictor model.....	38
3.2.2 Construction of corrector metamodel.....	39
3.3 Carbon Nanotubes (CNT).....	40
3.3.1 Properties of carbon nanotubes.....	41
3.4 Numerical Example.....	42
Chapter 4: Hyperbolic Polynomial Chaos Expansion.....	49
4.1 Construction of Multidimensional Polynomial Bases.....	49
4.2 Hyperbolic PC Expansion Truncation Method.....	51
4.3 Hyperbolic Factor (u).....	53
4.4 Statistical Information using HPCE Coefficients.....	56
4.5 Numerical Example.....	57
CONCLUSION.....	64
REFERENCES.....	65

## LIST OF TABLES

Table 2.1: Weiner-Askey Polynomials and their Distributions.....	11
Table 2.2: Starting 6 Univariate Orthonormal Legendre and Hermite Polynomials.....	16
Table 2.3: First Ten 2-D Orthonormal Legendre and Hermite Polynomials.....	19
Table 3.1: Mean and Standard Deviation of the Random Variables.....	44
Table 3.2: CPU Time and Speedup Comparison of HPCE and Standard PC method.....	47
Table 4.1: Mean and Standard Deviations of Random Variables.....	58
Table 4.2: Hyperbolic Factor and Corresponding Number of Bases.....	59
Table 4.3: CPU Time and Speedup Comparison of HPCE and Standard PC Method.....	63

## LIST OF FIGURES

Figure 2.1: Skew illustration.....	14
Figure 2.2: Graphical representation of scheme for selecting multivariate polynomials.....	17
Figure 2.3: Illustration of 2-D fourth order PC Expansion.....	18
Figure 2.4: Demonstration of data-fitting to a model.....	24
Figure 3.1: MWCNT interconnect network.....	36
Figure 3.2: MCC model of the network.....	36
Figure 3.3: ESC model of MWCNT interconnect network.....	38
Figure 3.4: a) Single-walled CNT b) Multi-walled CNT.....	41
Figure 3.5: MWCNT interconnect inner view.....	43
Figure 3.6: Statistics Comparison of transient response at node N1.....	45
Figure 3.7: PDF Comparison at the time point of maximum standard deviation.....	46
Figure 4.1: a) Conventional Linear Truncation Method b) Proposed Hyperbolic method.....	50
Figure 4.2: Graphical illustration showing truncation schemes.....	53
Figure 4.3: Effect of increasing the value of $u$ from near zero to 1.....	54
Figure 4.4: MWCNT interconnect inner view.....	57
Figure 4.5: Statistics Comparison of transient response at node N1.....	61
Figure 4.6: PDF Comparison at the time point of maximum standard deviation.....	62

## LIST OF PUBLICATIONS

1. Sakshi Bhatnagar, A. Merkley, R. Berdine, Y. Li, S. Roy, “Variability-aware Performance Assessment of Multi-Walled Carbon Nanotube Interconnects using a Predictor-Corrector Polynomial Chaos Scheme”, *IEEE Conference on Electrical Design of Advanced Packaging and Systems (EDAPS)*, December 2018
2. X. Cao, Sakshi Bhatnagar, M. Nikdast, S. Roy, “Hierarchical Polynomial Chaos for Variation Analysis of Silicon Photonics Micro-resonators”, *IEEE Conference on The Applied Computational Electromagnetics Society (ACES)*, April 2019
3. Sakshi Bhatnagar, Y. Li, A. Merkley, D.Weber, S. Roy, “Predictor-Corrector Algorithms and Their Scalability Analysis for Fast Stochastic Modeling of Multi-Walled Carbon Nanotube Interconnects”, *IEEE Conference on Electromagnetic Compatibility (EMC)*, June 2019

## CHAPTER 1: INTRODUCTION

In the world of miniaturization and scaling technology, hundreds of cores are embedded on a single chip. Enhanced performance and better power consumption are the major consideration for designing the chip [1]. Nowadays, microelectronic circuits are popular and are used in many pursuits for both personal as well as professional use like communication, entertainment, spatial exploration, and many other examples. The rapid advancement in high-speed integrated circuits (ICs) is supported by the technological innovations enabling integration of various sub-devices on a single chip. However, with the development in the integration level of ICs, the complexity of these circuits will increase resulting in an exponential growth of the fabrication cost and quantifying uncertainties [3]. Manufacturing uncertainties at nanoscale processes and operations are unpredictable at chip level, packaging level and at board levels of integrated systems. These uncertainties can arise in any circuit due to fabrication process variations, unpredictable environmental factors like temperature, human error and many more and may result into the unpredictable behaviour of the circuits. To overcome such issues, simulation solvers to model forward propagation of uncertainties or variations in random processes at device level to the network response are required. At sub-nano-technology scaling, numerous challenges like CPU time, speed, accuracy and others are raised which motivates new research area. This thesis work targets to some of the stated challenges.

### 1.1 Problem Statement

With continuous miniaturization of integrated circuits to sub-nano scale system and enhancement in the packaging density, uncertainties in random processes have increased which results in unpredictable behaviour in performance of the high-speed circuits. Therefore, advanced computer-

aided design (CAD) tools are in great demand as they are flexible enough to predict the impact of parametric uncertainty in any of the network responses. Traditionally, this unpredictability has been modelled using Monte Carlo (MC) method. In MC method, the large number of inputs are being generated or created based on the probability density functions (PDF) of the random variables and then every input is simulated through some circuit solver like SPICE [2]- [7]. The generated results of the simulations are collected and statistical information about the circuit's response is obtained. However, despite being a simple approach this method has a slow convergence rate i.e. it requires a large number of simulations to attain accuracy in statistical results. Therefore, for large networks, MC is computationally infeasible and expensive.

Now-a-days, techniques based on generalized Polynomial Chaos (gPC) are robust and model uncertainty in the system response as orthogonal polynomial basis functions of the input random variables expansion. The polynomial coefficients from PC expansion are now called as system's unknowns which are then calculated and evaluated using intrusive approach or non-intrusive approach.

In intrusive approaches, the Stochastic Galerkin (SG) approach is often used because it generates the PC coefficients which are used to provide statistics with better accuracy [8]- [22]. However, the computational cost for creating coupled deterministic network model increases exponentially with the increase in number of random variables in the network. Thus, the applicability of the SG approach is limited to the problems containing only low-dimensional random spaces.

In non-intrusive approaches, most widely used approaches are pseudo-spectral collocation method [24], the linear regression, [23], [26], [27] and non-intrusive stochastic collocation approach [25].

Intrusive methods are more accurate for a fixed cost, however, non-intrusive approaches can use commercial circuit-solvers like SPICE. Since, the meta-models (model of a model or guessed of the exact model through simulation) of the original systems are created in non-intrusive methods, any value of the random variable situated within the random space can be probed into the PC expansion equation and statistical outputs of the circuit can be generated, with negligible loss of accuracy. Also, the individual simulation can be parallelized which can be an added advantage. Among non-intrusive methods, linear regression approach is widely used [28], [29], [33]. This approach analyse the PC expansion of the circuit responses at a set of over-sampled multi-dimensional nodes situated within the random space, which leads to the over determined set of linear algebraic equations which can be solved in a least-square sense to evaluate directly the PC coefficients of the responses [33]. Generally, the multi-dimensional regression nodes are selected from the grid of the tensor product of one dimensional (1D) quadrature nodes [28], [29]. As the number of nodes in the tensor product grid increases in an exponential manner with the increase in the number of random dimensions, so a subset of the nodes, also known as design of experiments (DoE), can be selected. In the work of [30], it was demonstrated that selecting the DoE blindly can result in an inaccurate evaluation of the PC coefficients. But, the current literature on linear regression based PC analysis of EM and circuit problems have not identified any specific criterion for selecting the best set of DoE [27], [28]. Lately, a stochastic testing approach has developed a reliable technique to choose possible DoE in which the number of DoE is equal to the number of unknown PC coefficients [31], [32]. However, this technique does not choose the DoE using any optimal criterion and hence does not assure the maximum accuracy of the results.

As the number of random variables increases, conventional or standard PC methods become computationally more expensive. In particular to multi-walled carbon nanotube (MWCNT), as the

number of conducting shells are increased, number of deterministic simulations increases to best fit the coefficients. Thus, CPU cost for the deterministic simulation increases. To overcome this problem, in the work of [36] an ESC model (equivalent Single Conductor) of MWCNT network is being used. Despite being numerically efficient, this method came with the cost of accuracy of its created meta-model.

## **1.2 Goals**

Advancements in VLSI technology results in the evolution of complex high-speed integrated circuits in the nanoscale regime. Due to the reduction in area and size, increased clock frequency, interconnect plays a significant role in defining the overall performance of the chip. In the prevailing situation, delay in interconnect dominates over the delay in the gate. The traditional material for interconnects at the global level like Al or Cu is sensitive to electro migration due to the high current density which extensively affects the fidelity of high-speed circuits. To avoid or overcome such problems, an alternative solution has been found by the researchers. An alternative material for interconnects can be Carbon Nanotubes (CNT) for the current nanoscale technologies [34].

Modelling of a carbon nanotube interconnect is fundamentally dependent on its parameters like diameter, length, height, metallic and semiconducting properties and many more. The most versatile numerical technique for the modelling, calculation, and verification of the performance of high-speed interconnects in the presence of manufacturing as well as fabrication variabilities is Polynomial Chaos approach. Recently, these have been applied for the performance assessment of multi-walled carbon nanotube (MWCNT) interconnects to approximate the variability in the network responses by using linear combination of polynomial basis functions. The coefficients of the linear combination form the new unknowns of the network. The linear combination behaves

as a closed-form meta-model of the network after the coefficients have been evaluated [35]. Then it can be used to obtain the statistical information for signal integrity assessment of the network.

In MWCNTs as the number of shells increases, each SPICE simulation become computationally very expensive. As discussed before, in gPC, as the number of dimensions' increases, computation is very expensive.

This thesis presents a new technique in order to construct more accurate PC meta-models for MWCNT like predictor-corrector polynomial chaos scheme and hyperbolic polynomial chaos expansion (HPCE) scheme. In predictor-corrector polynomial scheme, low-fidelity meta-model is generated using Equivalent Single Conductor (ESC) approximation model and then its accuracy is enhanced using low order multi-conductor circuit (MCC) model called as corrector model. The sparse PC metamodel is obtained from the corrector function which is generated using fewer deterministic solutions of the rigorous MCC model. Thus, the total number of deterministic MWCNT system simulations needed is the summation of a large number of compact ESC model simulations (for constructing the predictor) and relatively small number of the rigorous MCC model simulations (for constructing the corrector). As a result, CPU cost of this sum of deterministic system simulations is considerably smaller than that required compared to directly construct a standard PC metamodel from MCC simulations only [36]. In HPCE, sparser polynomial expansion is generated based on hyperbolic criterion. These schemes result into the same accuracy as in standard PC meta-model but with immense reduction in CPU cost and speed.

### **1.3 Organization of The Thesis**

Most of the state of the art PC approaches are reviewed in this thesis and does not require any previous knowledge about the topic. Exploited techniques are explained in details while novel

ideas are supported using numerical examples and their discussions. The rest of the thesis is organized as follows:

Chapter 2 provides a review of basics of the generalized PC (gPC) theory and some common nonintrusive uncertainty quantification approaches like stochastic collocation method, stochastic Galerkin and the linear regression approach. And finally, it concludes with an overview of sparse polynomial chaos schemes. A discussion of major pros and cons of these approaches are also provided in this chapter. The main aim of this chapter is to familiarize the reader with few existing methods which are used to address the complications of uncertainty quantification.

Chapter 3 provides the details of one of the novel techniques used in the thesis, i.e. Predictor-Corrector Polynomial Chaos method. The work is compared to the results generated using the Polynomial Chaos technique to show the accuracy of the proposed technique.

Chapter 4 is dedicated to improvements in the CPU time costs and speed compared to Predictor-Corrector method. For this, sparser polynomial expansion is generated based on hyperbolic criterion rather than based on linear criterion. This chapter includes the details of the method Hyperbolic Polynomial Chaos technique for uncertainty quantification. The proposed scheme is mainly compared with the standard PC technique. The validation of the contributions is done with the help of numerical examples.

## CHAPTER 2: EXISTING METHODS OF UNCERTAINTY QUANTIFICATION

This chapter explains some of the existing methods of uncertainty quantification in detail. Firstly, an overview of Monte Carlo method is discussed with some of its drawbacks. Next, review of the basics of the generalized PC (gPC) theory is presented and then some common nonintrusive uncertainty quantification approaches like the stochastic collocation method, stochastic Galerkin and the linear regression approach are discussed. Linear regression technique is used extensively in solving the problems in this thesis so we discuss this in much detail. And finally, we conclude with an overview of sparse polynomial chaos schemes. Major pros and cons of these approaches are also discussed in this chapter. These techniques assist as the basis for coming chapters in this thesis.

### 2.1 Stochastic Modified Nodal Analysis

A general non-linear network consists of distributed and lumped circuit elements characterized by the modified nodal analysis (MNA) equations. After random variables  $\lambda$  are introduced to the network, the stochastic variant of the MNA equation would be [16], [22], [27]:

$$G(\lambda) Z(t, \lambda) + C(\lambda) \frac{dZ(t, \lambda)}{dt} + F(Z(t, \lambda)) + \sum_{i=1}^N (T_i Y_i(t, \lambda) T_i^T) * Z(t, \lambda) = B(t) \quad 2.1$$

where,  $\lambda$  are the random variables, G matrix contains the stamp of all memoryless circuit elements, C matrix contains the stamp of all memory lumped circuit elements, Z is the vector of stochastic current/voltage responses, F has the stamp of all the non-linear circuit elements,  $T_i$  represents the selector matrix mapping the vector of port currents  $i_i(t)$  for the  $i^{\text{th}}$  distributed network into nodal space of the circuit,  $Y_i$  represents the time-domain Y-parameter macro model of  $i^{\text{th}}$  distributed

network,  $B$  denotes the input vector of independent current and voltage sources, ‘\*’ signifies the transient or temporary convolution performed in a recursive manner in SPICE.

Expansion of stochastic quantities in equation 2.1 is done using multi-variate orthogonal polynomial bases as following:

$$\begin{aligned} G(\lambda) &= \sum_{k=0}^P G_k \varphi_k(\lambda), \quad C(\lambda) = \sum_{k=0}^P C_k \varphi_k(\lambda), \\ Z(t, \lambda) &= \sum_{k=0}^P Z_k(t) \varphi_k(\lambda), \quad Y_i(t, \lambda) = Y_{ik}(t) \varphi_k(\lambda) \end{aligned} \quad 2.2$$

The matrices  $G_k$ ,  $C_k$  and  $Y_k$  of equation 2.2 can be obtained from the knowledge of  $G(\lambda)$ ,  $C(\lambda)$  and  $Y(\lambda)$  respectively. The expansion of equation 2.2 is then replaced in equation 2.1 thereby transforming the stochastic equations of 2.1 into a set of augmented deterministic coupled equations as [16]:

$$G_a Z_a(t) + C_a \frac{dZ_a(t)}{dt} + F_a(Z_a(t)) + \sum_{i=1}^N (T_{ia} Y_{ia}(t) T_{ia}^T) * Z_a(t) = B_a(t) \quad 2.3$$

where,  $G_a$  is the augmented matrix constructed using  $G_k$  block matrix,  $C_a$  is the augmented matrix constructed using  $C_k$  block matrix,  $T_{ia}$  denotes diagonal matrix composed of  $T_i$ ,  $Y_{ia}(t)$  is augmented time domain macro model of  $i^{\text{th}}$  distributed network constructed from  $Y_{ik}$  block matrix,  $B_a = [B, 0, \dots, 0]^T$ ,  $Z_a = [Z_0, Z_1, \dots, Z_P]^T$  and  $F_a(\cdot)$  is augmented vector of non-linear circuit elements.

The overall MNA equations of 2.3 express an augmented deterministic network which is solved within a SPICE environment. Once the PC coefficients  $Z_k(t)$  are evaluated, the statistical moments of the system can be easily obtained by the PC expansion of equation 2.2

## 2.2 Monte Carlo (MC)

Monte Carlo techniques were traditionally used to quantify uncertainty in high-speed circuit systems. In this method, pseudo-random multi-dimensional samples are collected in large number based on the probability density function of the input parameters [2]. The simulations of the network at each of these samples are done and the group of the output response is generated. Any required statistical information can be estimated from these responses. If 'N' observations of a quantity Z represented by  $\{ z_1, z_2, z_3, \dots, z_N \}$  are achieved, the mean value of Z estimated by MC method as the expected value of the set  $\{z_i\}_{i=1}^N$  is :

$$\mu_Z = \frac{1}{N} \sum_{i=1}^N z_i \quad 2.4$$

The variance of Z is given by:

$$\text{Var}_Z = \frac{1}{N} \sum_{i=1}^N (z_i - \mu_Z)^2 \quad 2.5$$

The major disadvantage of MC method is that number of required simulations for convergence are very large  $\sim O(\frac{1}{\sqrt{N}})$ . Therefore, in a network if the time taken for each simulation is very large then computational cost becomes expensive.

### 2.3 General Polynomial Chaos (gPC)

The orthogonal polynomials concept has been into existence for a long time. Polynomial Chaos (PC) Theory was initially presented for Hermite orthogonal polynomials only and was known as 'Hermite-Chaos' or also Wiener-Chaos expansion. However, due to the requirement for determining differential equations in the presence of uncertainty for extensive engineering disciplines, the polynomial chaos was extended to incorporate other orthogonal polynomials and then it was renamed as generalized Polynomial Chaos (gPC) theory.

Suppose, a network or a system has uncertainties and that input uncertainty is represented by a random variable,  $\lambda$  defined in the probability space  $\Omega$ . The uncertainty in the network response  $Z(t,\lambda)$  is modelled using gPC theory [37]; provided the variables have finite second-order moments and is expressed as an expansion of orthogonal polynomials and their coefficients.

$$Z(t, \lambda) = \sum_{k=0}^{\infty} c_k(t) \varphi_k(\lambda) \quad 2.6$$

where,  $c_k$  is the PC coefficient (scalar quantity) at  $k^{\text{th}}$  time point and  $\varphi_k$  are the bases with respect to PDF of input random variable in the expansion of orthogonal polynomial. The truncated expansion is expressed as:

$$Z(t, \lambda) = \sum_{k=0}^m c_k(t) \varphi_k(\lambda) \quad 2.7$$

Where  $m$  is the order of polynomial expansion and  $(m+1)$  terms are present in the expansion. The polynomials  $\varphi_k(\lambda)$  are orthogonal to PDF of random input variable  $\lambda$ .

$$\langle \varphi_i(\lambda) \varphi_j(\lambda) \rangle = \int_{\Omega} \varphi_i(\lambda) \varphi_j(\lambda) \rho(\lambda) d\lambda = \alpha_i^2 \delta_{ij} \quad 2.8$$

Where  $\langle \rangle$  symbolizes inner product,  $\Omega$  represents the sample space,  $\rho$  is the PDF of random variable  $\lambda$ ,  $\delta_{ij}$  is the delta function and  $\alpha_i^2$  is scalar constant. For  $i \neq j$  inner product of the bases is always zero. Here, normalization is done by the factor  $\alpha_i$  and hence they are known as orthonormal polynomials.

Table shown below shows the Weiner-Askey polynomials which are chosen according to the probability density function of the random variables.

**Table 2.1:** Weiner-Askey Polynomials and their Distributions

Distribution of $\lambda$	Orthogonal Polynomials	Support Range
Gaussian	Hermite	$(-\infty, +\infty)$
Uniform	Legendre	$[-1, 1]$
Beta	Jacobi	$[-1, 1]$
Gamma	Laguerre	$[0, +\infty)$

Weiner-Askey scheme [37] is the way of deciding which basis gives better convergence rate for certain distributions. Some common distributions with their corresponding Wiener-Askey chaos polynomials are shown, however, orthogonal polynomials can be obtained from any random distribution.

### 2.3.1 Statistics calculation using PC coefficients

Main aim of the thesis is to find the statistical moments (mean and standard deviation) of the systems using PC metamodels and these statistical moments are obtained by integration over random space and placing PC expansions in the integration formulae.

Probability Density Function (PDF) and other higher order statistics can be acquired from the evaluated PC metamodel using an extensive number of MC samples.

#### 2.3.1.1 Calculation of arithmetic mean of the outputs

Expected value or the mean value of the response is the first order statistical moment. Mean is that central value of the network response which has random outputs spread around it. For output  $Z(\lambda)$ , expected value is given as:

$$E(Z(\lambda)) = \int_{\Omega} Z(\lambda) \rho(\lambda) d\lambda \quad 2.9$$

Replacing  $Z(t, \lambda) = \sum_{k=0}^P c_k(t) \varphi_k(\lambda)$  in the above equation, we get:

$$E(Z(\lambda)) = \sum_{k=0}^P c_k \int_{\Omega} \varphi_0(\lambda) \varphi_k(\lambda) \rho(\lambda) d\lambda \quad 2.10$$

It should be noted that  $\varphi_0(\lambda)$  is 1 for all orthonormal polynomials and it is in the above equation to make further derivation simpler. Now, above equation can be written as:

$$E(Z(\lambda)) = \sum_{k=0}^P c_k \langle \varphi_0(\lambda), \varphi_k(\lambda) \rangle = c_0 \quad 2.11$$

In above equation 2.20, when  $k \neq 0$  then all the expansion terms are zero. So, mean of the output is denoted by zeroth order PC coefficient.

### 2.3.1.2 Calculation of standard deviation and variance of the outputs

Variance ( $\sigma^2$ ) is the 2<sup>nd</sup> order statistical moment which shows the deviation from the mean value of the response.

Mathematically, it can be expressed as:

$$\begin{aligned} \text{Var}(Z(\lambda)) &= E[(Z(\lambda) - E(Z(\lambda)))^2] \\ &= \int_{\Omega} (\sum_{k=0}^P c_k \varphi_k(\lambda) - c_0 \varphi_0(\lambda))^2 \rho(\lambda) d\lambda \\ &= \sum_{k=0}^P \sum_{j=0}^P c_k c_j \int_{\Omega} \varphi_k(\lambda) \varphi_j(\lambda) \rho(\lambda) d\lambda - c_0^2 \\ &= \sum_{k=0}^P c_k^2 - c_0^2 \\ &= \sum_{k=1}^P c_k^2 \end{aligned} \quad 2.12$$

In all the equations of 2.12, RHS gives all the possible combination of  $\varphi_i \varphi_j$ . However, the inner product is non-zero when  $k=j$ . As a result, variance is the summation of squares of all the coefficients except the first term.

Standard deviation is also one of the most important statistical parameter and is denoted by  $\sigma$ .

Standard deviation is calculated by square root of variance and is given as:

$$\sigma = [\sum_{k=1}^P c_k^2]^{1/2} \quad 2.13$$

Standard deviation has the same order as of mean and therefore is more favourable than variance.

In stochastic systems, most of the results are located in  $\pm 3\sigma$  of the mean.

### 2.3.1.3 Probability density function (PDF) and other higher order moments

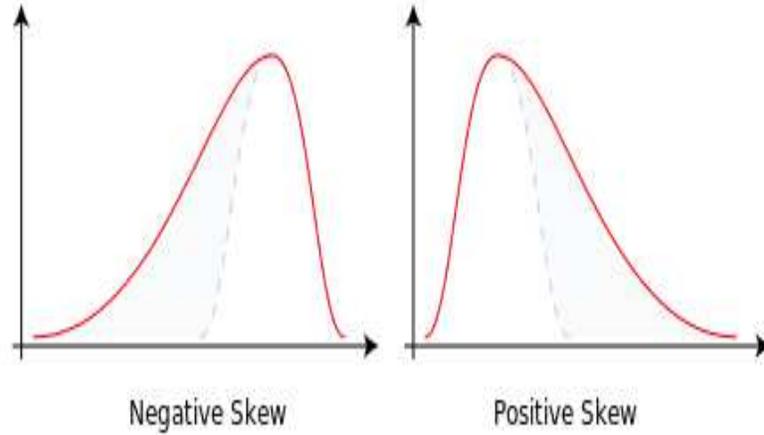
Higher-order statistical moments give more information about the random behaviour of the output of the system.

General formula for  $M^{\text{th}}$  order moment is expressed as [4]:

$$\begin{aligned} \mu_M (Z(\lambda)) &= E (Z(\lambda) - E (Z(\lambda)))^M \\ &= \int_{\Omega} (Z(\lambda) - E (Z(\lambda)))^M \rho(\lambda) d\lambda \end{aligned} \quad 2.14$$

Skewness is the third order moment which measures the asymmetry of probability distribution of output about its mean.

Figure 2.1 below is taken from [38] and it shows the asymmetry in distributions i.e. positive and negative skew:



**Figure 2.1:** Skew Illustration

Considering the two distributions in the above figure, they taper differently (known as tail) and one can observe that response distribution has positive, negative or no skew.

Kurtosis is the fourth order moment and it provides the information about the shape of the tail of the response distribution [39]. By producing a large number of Monte Carlo samples, the higher-order moment can be computed based on the distribution of the input. It is not needed to simulate the network at every sample points as the PC metamodel is known already. The PC metamodel is probed at all the samples and the values of the response  $Z(\lambda)$  is replaced in equation 2.14 to readily get any of the higher-order statistical moment.

Probability Density Function or PDF holds the information of all the other statistical moments discussed, therefore it is also one of the most important moments. It is also known as zeroth order moment. To obtain PDF from a PC expansion, one has to begin with an extensive and arbitrary set of MC samples and then probe the PC metamodel to create the responses at every sample node. After that, normalized histogram of all the data is generated to get the PDF of the response at all given time points. Conversely, one can achieve network simulations at each of those sample nodes

to get the responses; although it has to be noted that running a large number of simulations may need an ample amount of time, mainly when single simulation itself takes a while to finish.

### 2.3.2 One dimensional orthonormal polynomials

Standard normal distribution  $N(0,1)$  has a PDF  $\rho(\lambda)$  which is given as:

$$\rho(\lambda) = \frac{1}{\sqrt{2\pi}} \exp(-\lambda^2/2) \quad 2.15a$$

Now, by using equation, it can be proved that Hermite polynomials are orthogonal w.r.t distribution [3]:

$$\Phi_k(\lambda) = (-1)^k e^{\lambda^2/2} \frac{d^k}{d\lambda^k} e^{-\lambda^2/2} \quad 2.15b$$

Recursively, it can be written as:

$$\Phi_{k+1}(\lambda) = \lambda \Phi_k(\lambda) - k \Phi_{k-1}(\lambda) \quad 2.16$$

Where,  $\Phi_1(\lambda) = \lambda$ ,  $\Phi_0(\lambda) = 1$  and  $k > 1$ . Every polynomial here is normalized by  $\alpha_i^2$  factor and it is given as:

$$\alpha_i^2 = \langle \Phi_i(\lambda), \Phi_i(\lambda) \rangle = i! \quad 2.17$$

Uniform distribution  $U(-1,1)$  is given as:

$$\rho(\lambda) = \begin{cases} 0.5, & -1 \leq \lambda \leq 1 \\ 1, & \text{otherwise} \end{cases} \quad 2.18$$

Similarly, by using the above equation 2.8, it can be proved that Legendre polynomials are orthogonal w.r.t distribution [3]: They are generated analytically as:

$$\Phi_i(\lambda) = \frac{1}{k! 2^k} \frac{d^k}{d\lambda^k} (\lambda^2 - 1)^k \quad 2.19$$

Recursively, it can be written as:

$$\Phi_{k+1}(\lambda) = \frac{2k+1}{k+1} \lambda \Phi_k(\lambda) - \frac{k}{k+1} \Phi_{k-1}(\lambda) \quad 2.20$$

Here,  $\Phi_1 = \lambda$ ,  $\Phi_0 = 1$  and  $k > 1$ . Every polynomial here is normalized by  $\alpha_i^2$  factor and it is given as:

$$\alpha_i^2 = \langle \Phi_i(\lambda), \Phi_i(\lambda) \rangle = \frac{1}{2i+1} \quad 2.21$$

First 6 univariate orthonormal Legendre polynomials and Hermite polynomials are demonstrated below:

**Table 2.2:** Starting 6 Univariate Orthonormal Legendre and Hermite Polynomials

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

### 2.3.3 Multi-dimensional orthonormal polynomials generation

In real uncertainty quantification problems, there are several random variables so to analyse them, multi-dimensional polynomials are to be considered.

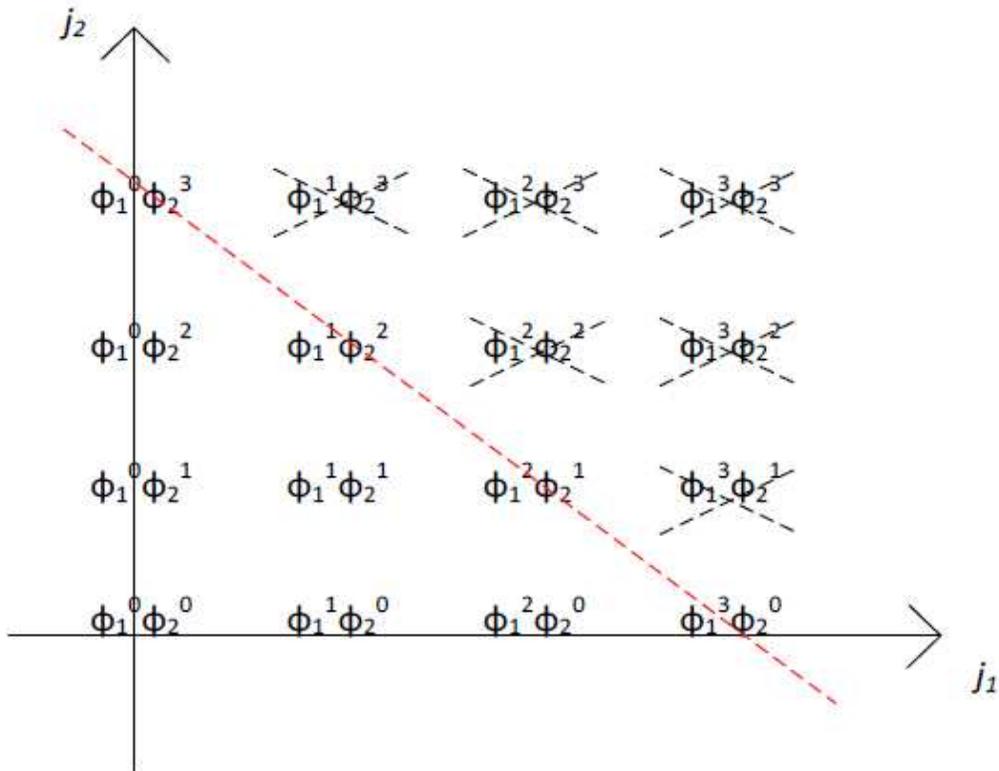
Now,  $\lambda$  will change to  $\lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n]^T$ , which signifies  $n$  mutually uncorrelated random variables. It can also be written as:

$$Z(t,\lambda) \approx \sum_{i=0}^{P+1} c_i(t) \Phi_i(\lambda) \quad 2.22$$

Here,  $\Phi_i(\lambda)$  signifies multi-dimensional orthonormal polynomials,  $P+1$  is the number of polynomial bases and is given by:

$$P+1 = \binom{m+n}{m} = \frac{(m+n)!}{m!n!} \quad 2.23$$

Where,  $m$  is the order or expansion degree of each random variable and  $n$  is the number of random variables. PDF  $\rho(\lambda)$  now is joint PDF of all the random variables and the random space  $\Omega$  converts to multi-dimensional random space. Now, for multi-D orthonormal polynomial bases we have:



**Figure 2.2:** Graphical representation of the scheme for selecting multivariate polynomials. Here  $n=2, m=3$

In this thesis,  $\alpha_i^2 = 1$  since normalization is done for univariate polynomials. Multi-variate polynomials are chosen from the tensor product of univariate polynomials.

$$\langle \varphi_i(\lambda), \varphi_j(\lambda) \rangle = \int_{\Omega} \varphi_i(\lambda) \varphi_j(\lambda) \rho(\lambda) d\lambda = \delta_{ij} \quad 2.24$$

Now each polynomial can be given as:

$$\varphi_d(\lambda) = \prod_{j=1}^n \varphi_{d_j}(\lambda_j) \quad 2.25$$

It should be noted that  $\varphi_{d_j}$  can be from different types. The standard approach to determine  $d_j$  indices of polynomials is [3]:

$$d_1 + d_2 + d_3 + \dots + d_n \leq m \quad 2.26$$

The graphical illustration of this approach for  $m=3$  and  $n=2$  is shown above in the figure 2.1. Its expansion for general ‘n’ RVs where polynomials are situated at a positive coordinates and are constrained by n-D surface is given as  $\sum d_i = m$

Illustration of 2-D fourth order PC expansion is shown below:

$$\begin{aligned} & \phi_0(\lambda_1)\phi_0(\lambda_2) \\ & \phi_1(\lambda_1)\phi_0(\lambda_2) \quad \phi_0(\lambda_1)\phi_1(\lambda_2) \\ & \phi_2(\lambda_1)\phi_0(\lambda_2) \quad \phi_1(\lambda_1)\phi_1(\lambda_2) \quad \phi_0(\lambda_1)\phi_2(\lambda_2) \\ & \phi_3(\lambda_1)\phi_0(\lambda_2) \quad \phi_2(\lambda_1)\phi_1(\lambda_2) \quad \phi_1(\lambda_1)\phi_2(\lambda_2) \quad \phi_0(\lambda_1)\phi_3(\lambda_2) \\ & \phi_4(\lambda_1)\phi_0(\lambda_2) \quad \phi_3(\lambda_1)\phi_1(\lambda_2) \quad \phi_2(\lambda_1)\phi_2(\lambda_2) \quad \phi_1(\lambda_1)\phi_3(\lambda_2) \quad \phi_0(\lambda_1)\phi_4(\lambda_2) \end{aligned}$$

**Figure2.3:** Illustration of 2-D fourth order PC Expansion

In the above Figure 2.3, the bivariate basis is shown and the total degree or order of these polynomials of the above approach in each row is consistent, and is incremented row by row [4].

In practice, the series is truncated by using a finite number of  $(P + 1)$  rows into consideration, and the total number of terms in the expansion  $Z(t, \lambda) = \sum_{k=0}^P c_k(t) \varphi_k(\lambda)$  for 'n' number of random dimensions.

Therefore, in the table 2.3 below, first ten bivariate bases and total degree is shown and the degree is incremented row by row as explained above.

**Table 2.3:** First Ten 2-D Orthonormal Legendre and Hermite Polynomials

<b>Bases</b>	<b>Legendre Polynomials</b>	<b>Hermite Polynomials</b>	<b>Total Degree</b>
$\Phi_0(\lambda)$	1	1	0
$\Phi_1(\lambda)$	$\sqrt{3}\lambda_1$	$\lambda_1$	1
$\Phi_2(\lambda)$	$\sqrt{3}\lambda_2$	$\lambda_2$	1
$\Phi_3(\lambda)$	$\sqrt{5} \left( \frac{3}{2}\lambda_1^2 - \frac{1}{2} \right)$	$(\lambda_1^2 - 1) / \sqrt{2}$	2
$\Phi_4(\lambda)$	$3 * \lambda_1 * \lambda_2$	$\lambda_1 * \lambda_2$	2
$\Phi_5(\lambda)$	$\sqrt{5} \left( \frac{3}{2}\lambda_2^2 - \frac{1}{2} \right)$	$(\lambda_2^2 - 1) / \sqrt{2}$	2
$\Phi_6(\lambda)$	$\sqrt{7} \left( \frac{5}{2}\lambda_1^3 - \frac{3}{2}\lambda_1 \right)$	$(\lambda_1^3 - 3\lambda_1) / \sqrt{6}$	3
$\Phi_7(\lambda)$	$\sqrt{15} * \lambda_2 * \left( \frac{3}{2}\lambda_1^2 - \frac{1}{2} \right)$	$\lambda_2 * (\lambda_1^2 - 1) / \sqrt{2}$	3
$\Phi_8(\lambda)$	$\sqrt{15} * \lambda_1 * \left( \frac{3}{2}\lambda_2^2 - \frac{1}{2} \right)$	$\lambda_1 * (\lambda_2^2 - 1) / \sqrt{2}$	3
$\Phi_9(\lambda)$	$\sqrt{7} \left( \frac{5}{2}\lambda_2^3 - \frac{3}{2}\lambda_2 \right)$	$(\lambda_2^3 - 3\lambda_2) / \sqrt{6}$	3

## **2.4 Non-Intrusive and Intrusive PC Methods**

PC coefficients are the key to perform uncertainty quantification and there are numerous ways to find these coefficients. They are mainly constituting of non-intrusive and intrusive methods which are discussed in details below.

### **2.4.1 Intrusive methods**

The methods which require coding and cannot be done in a black box fashion is known as Intrusive methods [8]- [22]. Different circuit solver is typically needed for the development of these methods and they manifest a higher accuracy in contrast to the non-intrusive methods. One of the widely used intrusive approaches is Stochastic Galerkin (SG) projection which generates augmented deterministic network, based on the equations governing the system. Then by one simulation of this system, PC coefficients can be determined. As the accuracy of the SG method is higher, therefore, it uses low order expansion. However, computation time and cost for this method scale exponentially when the number of random variables is increased since  $P+1$  terms are augmented. Besides, non-linear elements further add to the augmentation as they are modelled using lumped dependent sources [15]. Hence, the SG approach is generally good for small networks with less number of random variables.

The insufficiencies of the SG approach are discussed by the intrusive Stochastic Testing (ST) formulation [31], [32]. The ST approach determines the coupled system of equations of the augmented network, but in a decoupled way at every time point. These coupled equations are determined at  $P+1$  sampling nodes, however, the selection of nodes affects its accuracy. Bad choice of nodes gives ill-conditioned matrices, which results in an inaccurate solution or solution may be impossible to obtain. So, to generate a better set of sampling nodes, a node-selection algorithm is

used. The first step is to create a tensor product of  $(\mathbf{m} + \mathbf{1})^n$  nodes according to the Wiener-Askey polynomial scheme. From this tensor product, the 1st node is selected which has the highest quadrature-weight amongst all the other nodes. Similarly, the other P nodes are chosen based on their quadrature weights but with an added requirement that the node needs to have a large orthogonal component to the previously created set of nodes. Since the basis of evaluation is quadrature weights, so this algorithm does not assure the best selection of nodes, particularly for high dimensional problems.

Furthermore, ST approach can't make use of commercial solvers like SPICE as it is an intrusive formulation. The major advantage of ST approach over SG approach is that it can be solved in a decoupled fashion and hence, P+1 simulations required which can be accomplished in a parallelized manner. As ST approach is intrusive and intrusive coding is not applied in all of the cases so this problem obstructs the application of ST for sophisticated systems and tries to address the concern of being intrusive may not be precise because of the selection criteria of the nodes.

## **2.4.2 Non-intrusive methods**

Non-Intrusive methods doesn't require any necessity to design simulation tool, changes in the circuit or even understanding of any internal equations governing the network. Major benefit of non-intrusive approach over intrusive approach is that the present circuit-solvers (like SPICE etc.) can be used to obtain the output at the selected node.

### **2.4.2.1 Pseudo spectral polynomial chaos approach**

In this method, the PC expansion of  $Z(\lambda) \approx \sum_{i=0}^{P+1} c_i \Phi_i(\lambda)$  is directly utilized to the response of the system and using numerical integration techniques like Gaussian quadrature, PC coefficients are found which are then used to find any statistical information. Approximation of integral of

function  $G(\lambda)$  as a weighted sum of function  $w(\lambda)$  is done at already determined sample points using Gaussian quadrature.

$$\int_{\Omega} G(\lambda) \rho(\lambda) d\lambda = \sum_{i=1}^Q G(\lambda^i) w(\lambda^i) \quad 2.27$$

where,  $Q$  is number of sample nodes, in  $\lambda^i$ , superscript  $i$  is the  $i^{\text{th}}$  node in space and is given as  $\lambda^i = [\lambda_1^{(i)}, \lambda_2^{(i)}, \lambda_3^{(i)}, \dots, \lambda_n^{(i)}]$ ,  $w(\lambda^i)$  denotes quadrature weight to  $\lambda^i$  node,  $G(\lambda^i)$  denotes function response at  $\lambda^i$ .

$\lambda^i$  are the roots of the polynomials obtained as per Wiener-Askey scheme for 1-D problems. Consequently, there are  $m+1$  one dimensional roots generated where  $m$  is the maximum degree or order of expansion.  $Q$  number of nodes are obtained for multi-dimensional problems by forming the tensor product of all the one-dimensional nodes. Hence  $Q = (m+1)^n$ . Likewise,  $w(\lambda^i)$  is the multiplication of all the weights comparing to those one-dimensional nodes which are part of  $\lambda^i$ . From the roots of polynomials, one-dimensional nodes can be obtained.

Another way of obtaining nodes and their corresponding weights is by solving an eigenvalue problem which is called the Golub-Welch algorithm [40]. The technique to obtain nodes and weights for Hermite and Legendre polynomials using this algorithm is shown below.

For Hermite polynomials, A-matrix is constructed as:

$$A(i,j)_{(q+1) \times (q+1)} = \begin{cases} \sqrt{i} & j = i - 1 \\ \sqrt{j} & i = j - 1 \\ 0 & \text{otherwise} \end{cases} \quad 2.28$$

When Eigen-value decomposition is done on matrix;  $A = W \Lambda W^T$ , then the nodes will coincide with the Eigen values of matrix  $A$  i.e.  $\lambda^i = \Lambda(i, i)$  and their corresponding weights are the squares of 1<sup>st</sup> element of every Eigen-vector i.e.  $w(\lambda^i) = w_{1i}^2$ . Here  $W$  denotes unitary matrix.

For Legendre polynomials, A-matrix is constructed as following:

$$A(i,j)_{(q+1) \times (q+1)} = \begin{cases} \frac{0.5}{\sqrt{1-\frac{1}{(2(j-1))^2}}} & , j = i - 1 \\ \frac{0.5}{\sqrt{1-\frac{1}{(2(i-1))^2}}} & , i = j - 1 \\ 0 & otherwise \end{cases} \quad 2.29$$

In Legendre polynomials, alike Hermite polynomials, when Eigen-value decomposition is done on matrix;  $A = W \Lambda W^T$ , then the nodes will coincide with the Eigen values of matrix A which is given as  $\lambda_i = \Lambda(i, i)$  and their corresponding weights are the squares of 1st element of every Eigen-vector i.e.  $w(\lambda_i) = w_{1i}^2$ . Using orthogonal projection method, unknown PC coefficients  $c_k$  are calculated which also has inner product calculation of a function and a polynomial.

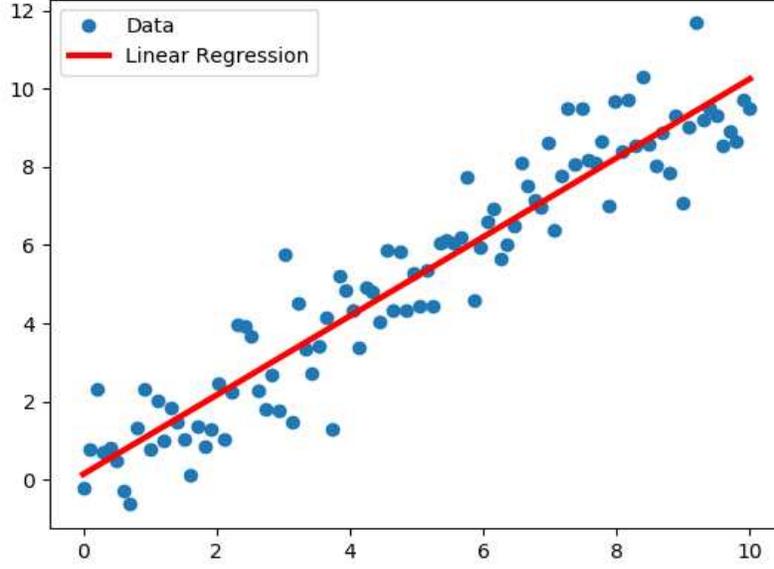
$$c_k = \langle Z, \varphi_k \rangle = \int_{\Omega} Z(t, \lambda) \varphi_k(\lambda) \rho(\lambda) d\lambda = \sum_{k=1}^Q Z(t, \lambda^i) \varphi_k(\lambda^i) w(\lambda^i) \quad 2.30$$

The pseudo-spectral method is beneficial for determining problems having a lower number of dimensions since the number of simulations needed is  $(m+1)^n$ . For higher dimensional problems, total terms in PC expansion and finally, the total number of simulations needed to compute the coefficients ascends in an exponential fashion and therefore, this approach doesn't contribute towards any notable advantages over MC method for high-dimensional problems.

#### 2.4.2.2 Standard linear regression method

Linear regression is a linear approach which is used to fit the model to a collected data. It is mainly used for predictive analysis of the systems.

Figure 2.4 below shows how the data is fitted to a model using this approach.



**Figure 2.4:** Demonstration of data-fitting to a model

The network response can be modelled using Polynomial Chaos approach and is given by:

$$Z(t, \lambda) = \sum_{k=0}^M c_k(t) \varphi_k(\lambda) \quad 2.31$$

Where,  $M = 2(P+1)$  represents the set of over-sampled nodes within the random space and for  $M$  nodes, this equation in matrix form can be expressed as:

$$Ac + \varepsilon = B \quad 2.32$$

Here,

$$A = \begin{bmatrix} \varphi_0(\lambda^{(1)}) & \dots & \varphi_N(\lambda^{(1)}) \\ \vdots & \ddots & \vdots \\ \varphi_0(\lambda^{(M)}) & \dots & \varphi_N(\lambda^{(M)}) \end{bmatrix}$$

$$c = \begin{bmatrix} c_0 \\ \vdots \\ c_N \end{bmatrix}; B = \begin{bmatrix} B^{(1)} \\ \vdots \\ B^{(M)} \end{bmatrix}; \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_M \end{bmatrix} \quad 2.33$$

$\mathcal{E}$  is the random truncation error which is ideally zero but in practical cases, it has some non-zero small value,  $B$  contains network responses generated by probing original stochastic system at multi-dimensional  $M$  nodes. Also, linear least squares method does data-modeling in a way that the maximum number of nodes are approximated to fit the model.

Once this model is created, it can be then probed to produce an output for any of the input pertaining to the random space. For instance, shown in the figure where a linear model is constructed from a discrete set of points, so as to minimize the approximation error. The linear least-squares algorithm [28], [29] doesn't give a single solution to the given problem and the final selection of nodes is partially based on the primary random selection of nodes, although it is created to assure that the most optimum model is constructed, by minimizing the terms of the sum of squares of errors.

$$\tilde{c} = \arg_{\tilde{c}} \min S(\tilde{c}) \quad 2.34$$

Where,

$$S(\tilde{c}) = \| |B - A\tilde{c}| \|^2 = \sum_{k=1}^M r_k^2 = \sum_{k=1}^M ( |y^{(k)} - \sum_{j=1}^N c_j \varphi_j(\lambda^{(k)})| )^2 \quad 2.35$$

Here,  $\mathcal{E}$  is taken as zero, assuming it to be an ideal case. Since, it is a convex function; therefore, above equation is minimum when gradient of the equation is zero. Hereafter

$$\frac{\delta S}{\delta c_j} = 2 \sum_{k=1}^M r_k \frac{\delta r_k}{\delta c_j} = 2 \sum_{k=1}^M (y^{(k)} - \sum_{j=1}^N c_j \varphi_j(\lambda^{(k)})) (-\varphi_j(\lambda^{(k)})) = 0 \quad 2.36$$

On simplification, we get

$$\sum_{k=1}^M \sum_{j=1}^N \tilde{c}_j \varphi_j(\lambda^{(k)}) \varphi_k(\lambda^{(k)}) = \sum_{k=1}^M \varphi_k(\lambda^{(k)}) B_k \quad 2.37$$

In matrix form, it can be written as:

$$(A^T A) \tilde{c} = A^T B \quad 2.38$$

This matrix form of the equation has solution when  $A^T$  is full column rank and this makes  $A^T A$  as positive definite.

Hence, coefficients vector can be presented as:

$$\tilde{c} = (A^T A)^{-1} A^T B \quad 2.39$$

Here,  $A^T A$  is information matrix (also known as Fisher information matrix). On solving the equation above, it gives PC coefficients. From these coefficients, information regarding statistical moments can be determined.

#### **2.4.2.2.1 Advantages and drawbacks of linear regression**

The primary benefits of the linear regression method are that several popular circuit-solvers can be applied to obtain the values of the outputs needed in  $\tilde{c} = (A^T A)^{-1} A^T B$  without any requirement of extensive coding. Furthermore, simulations can also be parallelized as the simulation of a specific node is independent of the other. The number of simulations also scale in a polynomial manner with the number of random numbers  $M=2(P+1)$ . It is important to note that the matrix  $A$  can be evaluated and saved once, and can be used again for the same or any other problem later, having the same number of random variables and maximum expansion order. These factors play a vital role in speedup over any intrusive method or MC method. Also, by over-sampling the primary set of nodes, it is guaranteed that this method will give the results with adequate accuracy.

In contrast, standard linear regression [28],[29] approach has some disadvantages as well i.e. it does not present a way of optimizing the subset of chosen nodes. Randomly picking  $M$  number of

nodes from the tensor product without any optimization, may or may not produce good results depending on the selection of nodes. Therefore, there are several techniques like D-optimal criterion and many others are used in addition to the linear regression method that acts upon optimizing some attributes of the equation mentioned above.

#### 2.4.2.2.2 Selecting regression nodes

Linear regression method starts by approximating the uncertainty in the system's output using PC expansion and then coefficients are computed at the sample node  $\lambda^{(k)}$  using known polynomials  $\varphi_k$ .

The equations are expressed as:

$$Z(t, \lambda) = \sum_{k=0}^P Z_k(t) \varphi_k(\lambda) \quad 2.40$$

$$A_j = [ \varphi_0(\lambda^{(j)}) I, \varphi_1(\lambda^{(j)}) I, \varphi_2(\lambda^{(j)}) I, \dots, \varphi_P(\lambda^{(j)}) I ] \quad 2.41$$

Here, identity matrix is denoted as I and the above equation can be simplified and expressed as:

$$A_j \tilde{Z} = Z(t, \lambda^{(j)}) \quad 2.42$$

Where,  $\tilde{Z} = [ Z_0(t), Z_1(t), \dots, Z_P(t) ]^T$ ,  $Z(t, \lambda^{(j)})$  represents the result of the simulation

at  $\lambda^{(j)}$

Standard linear regression method uses  $M=2(P+1)$  or  $3(P+1)$  nodes situated within the random space which results into the construction of overdetermined system of linear algebraic equations

i.e.  $A\tilde{Z} = Z$

#### 2.4.3 D-optimality criterion

As discussed above that selection of M regression nodes plays a vital role in the accuracy of response of the system. One of the most prevalent technique for selecting optimized nodes is D-

optimality criteria [41],[42]. This method is used to minimize  $(A^T A)^{-1}$  term so as to maximize the determinant of the information matrix.

As discussed above, linear regression method solves  $Ac + \mathcal{E} = Z$  system of algebraic equation. Following lemma shows the accuracy of calculated PC coefficients using D-optimal technique:

*Lemma:* Assuming that the truncation error  $\mathcal{E}_j$ ,  $1 \leq j \leq M$  at all M DoE of (2.30) are independent of each other and exhibit a normal distribution of zero mean and same variance  $\sigma^2$ , 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  $A^T A$  is maximized.

Proof: Due to the presence of random truncation error, PC coefficients based on the PC expansion of the system responses make themselves random dimensions. Variance is evaluated as:

$$\text{Var}(\tilde{c}) = \text{var}((A^T A)^{-1} A^T B) = (A^T A)^{-1} A^T \text{var}(B) ((A^T A)^{-1} A^T)^T \quad 2.43$$

As the truncation error  $\mathcal{E}_j$  for every Design of Experiments (DoE) is independent and has constant variance  $\sigma^2$ . Replacing  $\text{var}(B) = \sigma^2 I$  in the above equation, we get:

$$\text{Var}(\tilde{c}) = (A^T A)^{-1} \sigma^2 \quad 2.44$$

It is important to decrease the uncertainty in the solution  $\tilde{c}$  (i.e. variance of  $\tilde{c}$ ) to guarantee maximum accuracy of PC coefficients. As variance of  $\tilde{c}$  is inversely proportional to the determinant of the information matrix  $A^T A$ , so to minimize the value of the variance of  $\tilde{c}$ , the best way is to maximize the determinant. Hence, M DoE for the linear regression must be picked in such a way that the maximized determinant of the information matrix can be achieved. This criterion is known as the D-optimal criterion. Numerous optimality criteria do exist, yet the D-optimal criterion has been considered the most efficient and widespread till date. The next

challenge is to generate a search-algorithm that can effectively distinguish the D-optimal nodes from the tensor product space of nodes.

#### 2.4.4 Fedorov search algorithm

D-optimal criterion is based on Fedorov Search algorithm which is greedy-search algorithm to select D-optimal nodes [43], [44]. Estimation theory, data analysis, and many such fields use greedy search algorithm quite often. In this algorithm, first, a set of  $M=2(P+1)$  nodes are considered which are picked from the tensor product grid of  $(m+1)^n$  multi-dimensional quadrature nodes and then corresponding information matrix  $A^T A$  is constructed. Consequently, from the remaining  $(m+1)^n - M$  quadrature nodes, each DoE in the initial set is replaced by the best possible substitute DoE in such a way that the determinant of the information matrix increases by the maximum amount in the procedure. This refinement process of the starting DoE continues step-by-step till all the starting set of nodes has been replaced [26], [30].

As per the earlier explanation, at the  $j$ th step, it is assumed that the first  $j-1$  nodes have been replaced by their best possible substitutes. Now if the  $j$ th DoE of the starting set is eliminated from  $A$ , then the new determinant of the information matrix can be represented as:

$$\begin{aligned} \det(A^T A)_{\text{new}} &= \det((A^T A) - R(\lambda^{(j)}) R'(\lambda^{(j)})) \\ &= \det((A^T A)(1 - R(\lambda^{(j)})(A^T A)^{-1} R'(\lambda^{(j)}))) \end{aligned} \quad 2.45$$

Where,  $R(\lambda^{(j)})$  represents a row vector added by  $j$ th DoE in matrix  $A$ . Similarly, for any  $k$ th arbitrary DoE from remaining  $(m+1)^n - M$  quadrature nodes, new determinant of the new information matrix is evaluated and it is expressed as:

$$\begin{aligned}\det(A^T A)_{\text{new}} &= \det((A^T A) + R(\lambda^{(j)}) R^T(\lambda^{(j)})) \\ &= \det((A^T A)(1 + R(\lambda^{(j)})(A^T A)^{-1} R^T(\lambda^{(j)})))\end{aligned}\quad 2.46$$

Adding the results of above 2 equations and then exchanging  $j^{\text{th}}$  DoE of the initial set with any random  $k^{\text{th}}$  DoE from the remaining  $(m+1)^n - M$  quadrature nodes, recursive function of new determinant is given as:

$$\det(A^T A)_{\text{new}} = \det(A^T A)(1 + d_{kk} - d_{jj} + d_{kj}^2 - d_{kk}d_{jj})\quad 2.47$$

$$d_{kj} = R(\lambda^{(k)}) \psi^{(j-1)} R^T(\lambda^{(j)})\quad 2.48$$

Where,  $\psi^{(j-1)}$  denotes inverse of information matrix generated after preceding exchange (j-1). Now,  $k^{\text{th}}$  node  $\lambda^{(k)}$  needs to be selected such to satisfy optimization criteria:

$$\text{Max}(d_{kk} - d_{jj} + d_{kj}^2 - d_{kk}d_{jj})\quad 2.49$$

Using above mentioned steps, best possible node  $\lambda^{(k)}$  is selected to satisfy above optimization criteria and then replacement is done. New determinant obtained can be updated directly using equation 2.43 and this replacement-substitution process moves on to the  $(j+1)^{\text{th}}$  node. When all the initial DoE are replaced, then new set of DoE will represent D-optimal selection.

#### 2.4.5 Cost of computation for search algorithm

Computation cost of the search algorithm is mainly due to 2 reasons:

- a. It requires searching through  $(m+1)^n - M$  quadrature nodes in initial set for each DoE i.e. total searches would be  $M((m+1)^n - M)$ . So the related CPU cost would be:

$$\begin{aligned}C_a &= 2(P+1) ((m+1)^n - 2(P+1)) C_1 \\ &\approx 2(P+1)(m+1)^n C_1\end{aligned}\quad 2.50$$

Where,  $C_1$  represents CPU cost of computing the terms  $(d_{kk} - d_{jj} + d_{kj}^2 - d_{kk}d_j)$  with an assumption that  $\psi^{(j-1)}$  is already known.  $C_1$  is expressed as:

$$C_1 = 3k ((P+1)^2 + (P+1)) \quad 2.51$$

Here, first term is the cost due to matrix-vector multiplication i.e.  $\psi^{(j-1)} R^T(\lambda^{(j)})$  and second term is the cost due to vector-vector multiplication i.e.  $R(\lambda^{(k)})$  and  $\psi^{(j-1)} R^T(\lambda^{(j)})$  and the above operations are to be performed for three scalar quantities  $d_{jj}$ ,  $d_{kk}$ ,  $d_{kj}$  so factor 3 is used in the cost computation. Here,  $k$  is assumed to be the cost due to each floating point operation. After combining 2 equations i.e. 2.48 and 2.49, overall cost of the search algorithm ( $C_a$ ) scales in an exponential way with the number of random variables ( $n$ ) and it is computed as:

$$O((P+1)^3(m+1)^n) \approx O(n^{3m} (m+1)^n) \quad 2.52$$

- b. Another factor for computational cost is due to each substitution as information matrix changes every time and therefore, inverse  $\psi^{(j-1)}$  has to be evaluated again. CPU cost is expressed as:

$$C_b = 2(P+1) C_2 \quad 2.53$$

Where,  $C_2$  represents CPU cost of each matrix inversion.  $C_2$  scales as  $O((P+1)^3)$  for direct inversion methods; thereby confirming that collective cost of matrix inversion methods ( $C_b$ ) scales as  $O((P+1)^4) \approx O(n^{4m})$  w.r.t number of random variables ( $n$ ). When  $2 \leq m \leq 5$ , for typical PC problems, then CPU cost scales to near exponential.

The above two peculiarities of the search algorithm significantly slow down its performance for high-dimensional problems and can even render it infeasible for some of the problems.

The cost of executing the search algorithm can usually become an important part of the cost of performing the M deterministic SPICE simulations.

#### 2.4.6 Accelerating search-algorithm for high dimension random spaces

This approach is based on the basis that once a fairly large determinant of the information matrix has been accomplished, any further enrichment of the determinant will change to negligible improvement in the accuracy of the calculated PC coefficients. Therefore, here only the K worst DoE in the initial set of nodes will be identified and replaced instead of replacing all M DoE as proposed in previous approach. The replacement of the K worst DoE will result in an adequately great increase of the determinant of the information matrix, through eliminating the requirement for replacing the remaining M-K DoE. Therefore, this approach will diminish the number of searches from  $M((m+1)^n - M)$  to  $K((m+1)^n - M)$ . Due to this, cost computation will have reduction by a factor of M/K. Here, K is originally set to  $\lceil M/5 \rceil$  where  $\lceil \cdot \rceil$  is the ceiling function. Next, from equation 2.45 it is seen that the reduction in the value of the determinant caused by removing the  $j^{\text{th}}$  DoE ( $\lambda^{(j)}$ ) is proportional to the term  $d_{jj}$ .

$$d_{jj} = \mathbf{R}(\lambda^{(j)}) \boldsymbol{\psi}^{(0)} \mathbf{R}(\lambda^{(j)})^T \quad 2.54$$

where, inverse of original information matrix [26] containing starting M DoE is represented by  $\boldsymbol{\psi}^{(0)}$ . Consequently, the K worst DoE are distinguished as those DoE in the starting set that have the smallest plausible value of the scalar quantity  $d_{rr}$ . It is acknowledged that estimation of the term  $d_{rr}$  of all  $2(P+1)$  DoE can be achieved inexpensively since the matrix inverse  $\boldsymbol{\psi}^{(0)}$  requires to be computed just once and once the K worst DoE have been distinguished, the search algorithm of the previous subsection is meant for only these DoE. Consequently, an analysis is made to determine if the determinant of the information matrix  $\mathbf{A}^T \mathbf{A}$  is fairly high. If not, then the next

worst DoE (i.e.,  $K+1^{\text{th}}$  worst DoE) is identified using equation 2.54 and replaced as in earlier case. This sequential method proceeds until the determinant of the information matrix  $A^T A$  is considered to be large enough. It is observed from various examples that  $K = [M/5]$  is a reliable starting assumption for  $K$  and rarely does this value need to be extended further [26].

#### 2.4.6.1 Efficiency of K-worst node substitution in search algorithm

Substitution of  $K$ -worst DoE has two major advantages. First, total number of searches will reduce from  $[M(m+1)^n - M]$  to  $[k((m+1)^n - M)]$ ; where  $k = \frac{M}{5}$ . Due to this reason, cost of computation is reduced by factor 5 and is given as:

$$C_a \approx k((m+1)^n - 2(P+1)) C_1 = \left[ \frac{2(P+1)}{5} \right] (m+1)^n C_1 \quad 2.55$$

Second, CPU cost due to matrix inversion performance will also be reduced significantly according to the following lemma:

*Lemma 2:* Utilization of the Sherman-Morrison-Woodbury formula of 2.51 and 2.53 will ensure that the total CPU costs to perform the matrix inversions in the  $k$ -worst node substitution search algorithm will scale as  $O((P+1)^3) \approx O(n^{3m})$  with respect to the number of random variables ( $n$ ).

## CHAPTER 3: PREDICTOR-CORRECTOR POLYNOMIAL CHAOS SCHEME

This chapter demonstrates a novel predictor-corrector scheme to expedite the construction of polynomial chaos (PC) metamodel for the performance assessment of multi-walled carbon nanotube (MWCNT) interconnects. The proposed method is divided into two main steps. Firstly, a low-fidelity predictor PC metamodel of the MWCNT system is formed using the equivalent single conductor (ESC) estimate model. Consequently, the accuracy of the predictor model is adequately enhanced by using low-order corrector function based on the multi-conductor circuit (MCC) model. The total CPU costs by combining the predictor and corrector functions come out to be much smaller than the CPU costs for creating a traditional PC metamodel of similar accuracy.

### 3.1 Introduction

Polynomial Chaos schemes have been employed for the performance assessment of multi-walled carbon nanotube (MWCNT) interconnects [12], [36] to approximate the variability in the MWCNT network responses using a linear combination of orthonormal polynomial basis functions. These basis functions, in particular, are orthonormal with respect to the joint probability density function of the network's input random variables [47]. Then the coefficients of the linear combination have been evaluated which are the new unknowns of the network. This linear combination acts as a closed-form metamodel of the network, which is then probed in an analytic way to do an assessment of the network.

Despite, the advantage of being fast convergent and reliable, the chief limitation of PC approach is that it normally requires an excessive number of deterministic simulations of the MWCNT network to best fit the coefficients [36]. Moreover, this problem is worsened by the fact that the

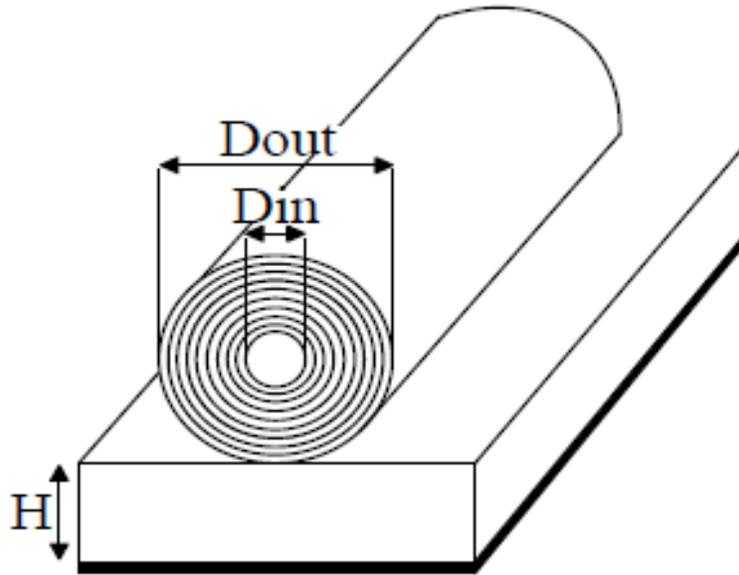
number of conducting shells in MWCNT network is usually very large, and thus, the CPU time cost for even a single deterministic simulation can be too expensive. Therefore, in order to avoid this issue, an equivalent single conductor (ESC) approximation of the more rigorous multi-conductor circuit (MCC) model [48] of the MWCNT network was used in the work of [36]. With the use of ESC model, simulation of each deterministic network model was significantly expedited, however, this efficiency came at the cost of accuracy of the created metamodel.

In this thesis, a novel and even more effective approach is presented to create very accurate PC metamodels for MWCNT networks. In this approach, firstly, a numerically reasonable predictor PC metamodel is constructed using the ESC model, also called as low-fidelity metamodel because of the ESC approximation of the MWCNT network. Next, the accuracy of this low-fidelity predictor metamodel is enhanced by combining a corrector function to it. The corrector function takes the form of a very sparse PC metamodel which is constructed using a very little number of deterministic solutions of the rigorous MCC model. Therefore, the total number of deterministic MWCNT network simulations needed is the sum of a large number of compact ESC model simulations (for constructing the predictor) and a small number of the rigorous MCC model simulations (for constructing the corrector). The CPU cost of this sum is also considerably smaller than the CPU cost of directly constructing a traditional PC metamodel from MCC simulations only. The proposed predictor-corrector scheme still presents similar accuracy as the traditional PC metamodel.

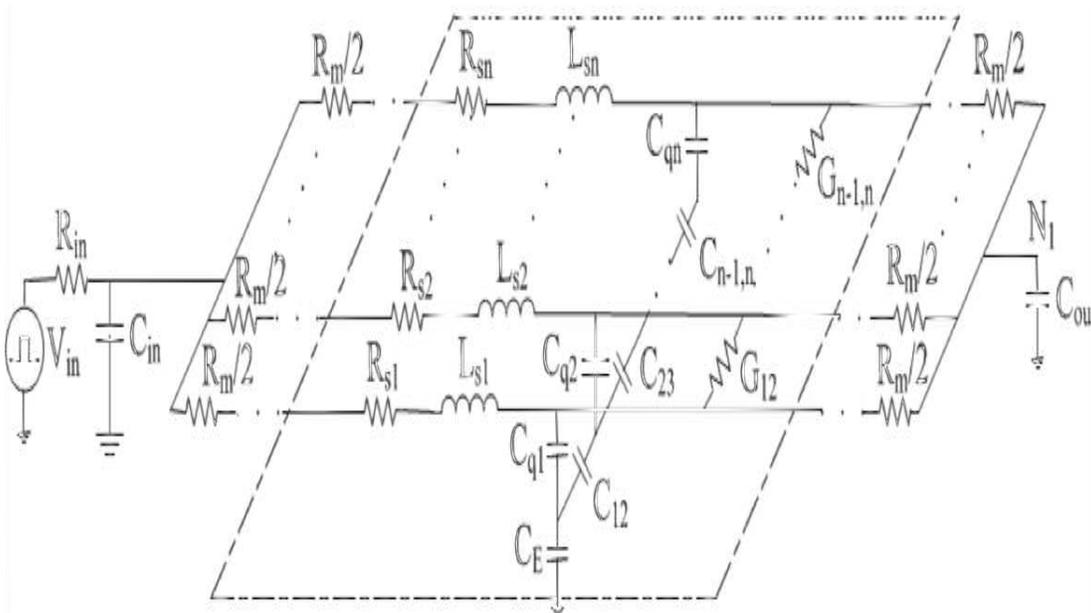
### **3.2 Proposed Predictor-Corrector Scheme**

Consider a common MWCNT network which consists of 'n' number of concentric shells as shown in Figure 3.1 below. In the network, each shell can be represented by an equivalent resistance-inductance-conductance-capacitance (RLGC) based lumped SPICE model as demonstrated in

Figure 3.2. Here,  $R_m$  is the imperfect metal contact resistance of the shells. Figure 3.2 represents the rigorous multiconductor circuit (MCC) model of the complete network.



**Figure 3.1:** MWCNT interconnect network



**Figure 3.2:** MCC model of the network

MCC model is the hi-fidelity model for MWCNT network. Here, for 1 line it is represented by multiple conducting lines which is much more realistic characterization of the circuit. So this model actually generates an accurate result but it takes a while to simulate as each line has R-C components so to simplify this, we approximate the circuit by considering the input voltage  $V_{in}$  and voltage at end to be the same. So in this way we can compress all these lines into one and which is known as ESC model which is much faster with less accuracy than MCC model. The performance of this MCC model is described by the stochastic modified nodal analysis (MNA) equations as:

$$G(\lambda)Z(t, \lambda) + C(\lambda) \frac{dZ(t, \lambda)}{dt} + F(Z(t, \lambda)) = B(t) \quad 3.1$$

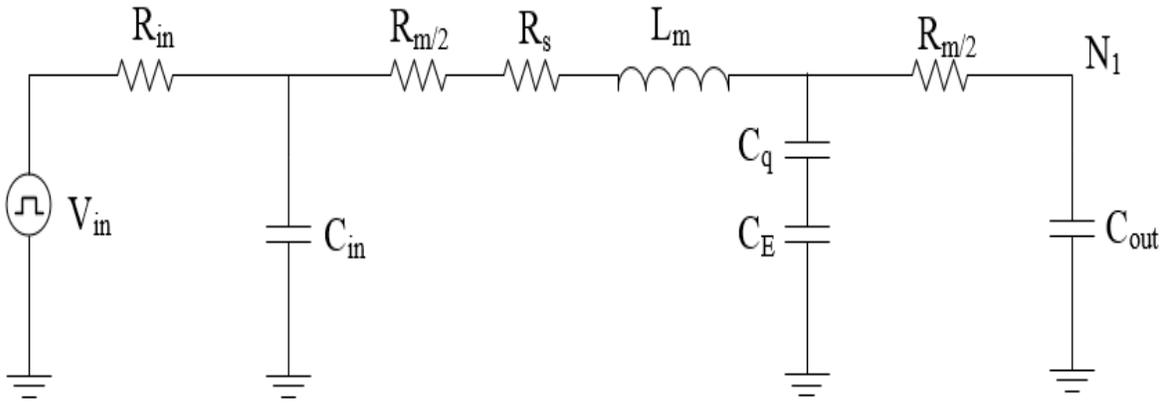
where, C and G are the matrices which contain RLGC stamps of lumped circuit, Z represents the vector of stochastic current or voltage responses and vector of independent current or voltage sources is denoted by B. Due to N mutually uncorrelated random parameters  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_N]$ , the variability is introduced into the model located within the multi-dimensional space  $\Omega$ . The foremost purpose of PC approaches is to approximate the total or resultant variability in the network responses using a linear combination of orthonormal basis functions [12], [36] as:

$$Z(t, \lambda) \approx \sum_{k=0}^P Z_k(t) \phi_k(\lambda) \quad 3.2$$

where,  $\phi_k(\lambda)$  is the  $k^{\text{th}}$  degree N-dimensional polynomial,  $Z_k(t)$  is the coefficient and the number of terms in the expansion is truncated to  $P+1 = \frac{(N+m)!}{N!m!}$ , here m is the maximum degree of the expansion of equation 3.2. Normally, a large  $O(P+1) = O(N^m)$  number of deterministic SPICE simulations of the MCC model of Figure 3.2 is needed to accurately estimate the coefficients of equation 3.2 [12], [36]. To reduce the excessive CPU time costs required for these many SPICE simulations, a more efficient predictor-corrector method is proposed.

### 3.2.1 Construction of low-fidelity predictor model

In the predictor-corrector approach, firstly, the MCC model as shown in Figure 3.2 is replaced by its equivalent single conductor (ESC) model which is shown below in Figure 3.3 [48]. The ESC model is based on the hypothesis that the voltage on each shell positioned at the same longitudinal distance from one end of the network is always equal. This hypothesis, while not specifically true, however, gives a more compact SPICE model without incurring many errors [48].



**Figure 3.3** ESC model of MWCNT interconnect network

Non-intrusive linear regression approach is being used in constructing PC metamodel of the network. In this approach, ESC model is used for each deterministic SPICE simulation of the network needed to estimate the PC coefficients. This PC metamodel is also called as the low fidelity predictor metamodel on account of the ESC approximation and is represented as:

$$Z_{\text{pred}}(t, \lambda) = \sum_{k=0}^P Z_k^{(ESC)}(t) \phi_k(\lambda) \quad 3.3$$

Where,  $Z_k^{(ESC)}(t)$  is the  $k^{\text{th}}$  predictor coefficient,  $\phi_k(\lambda)$  is the  $k^{\text{th}}$  degree  $N$ -dimensional polynomial.

In equation 3.2 and 3.3, the number of coefficients are same, however, for each ESC simulation, CPU cost is very small when compared to each MCC simulation.

Therefore, predictor metamodel of equation 3.3 can be constructed very fast compared to standard PC metamodel of equation 3.2. Beside this, it has a drawback that predictor metamodel of equation 3.3 is less accurate than standard PC metamodel of equation 3.2.

### 3.2.2 Construction of corrector metamodel

Enrichment of the low-fidelity predictor metamodel is the next step after it is constructed as expressed in equation 3.3. Enrichment is done till it reaches close enough in accuracy to standard PC metamodel. For this purpose, the difference between the actual network solution  $Z(t,\lambda)$  and the predictor metamodel of equation 3.3 is defined which is known as corrector factor. It is expressed as:

$$F(t,\lambda) = Z(t, \lambda) - Z_{\text{pred}}(t, \lambda) \quad 3.4$$

As ESC model is a reasonable approximation of MCC model so the Euclidean norm of the corrector function in equation 3.4 is small. Therefore, the corrector function  $F(t,\lambda)$  is approximated using sparse PC metamodel as:

$$Z(t, \lambda) - Z_{\text{pred}}(t, \lambda) = \sum_{k=0}^Q Z_k^{(MCC)}(t) \phi_k(\lambda) \quad 3.5$$

In the above equation,  $Z_k^{(MCC)}(t)$  denotes  $k^{\text{th}}$  coefficient of the corrector function,  $Z_{\text{pred}}(t, \lambda)$  is the predictor approximation nodes calculated in equation 3.3 above. Here,  $Q+1$  is the number of terms which is lesser than  $P+1$  terms i.e.  $Q+1 \ll P+1$ . The reason for this is that  $Q+1$  basis functions contain all 1-D basis functions and only those 2-D basis functions of order two present in original metamodel of equation 3.2.

Here also,  $Q+1$  coefficients are calculated using non-intrusive linear regression method. In this linear regression method, for each regression node  $\lambda = \lambda^{(j)} = [\lambda_1^{(j)}, \lambda_2^{(j)}, \lambda_3^{(j)}, \dots, \lambda_n^{(j)}]$ , the

predictor solution  $Z_{\text{pred}}(t, \lambda = \lambda^{(j)})$  is directly calculated from equation 3.3 and the network solution  $Z(t, \lambda = \lambda^{(j)})$  is evaluated from SPICE simulations of MCC model.

Lastly, once the coefficients of equation 3.5 are calculated, the accurate and high-fidelity PC metamodel of the response can be recovered easily by adding the predictor and corrector function from equation 3.4 as:

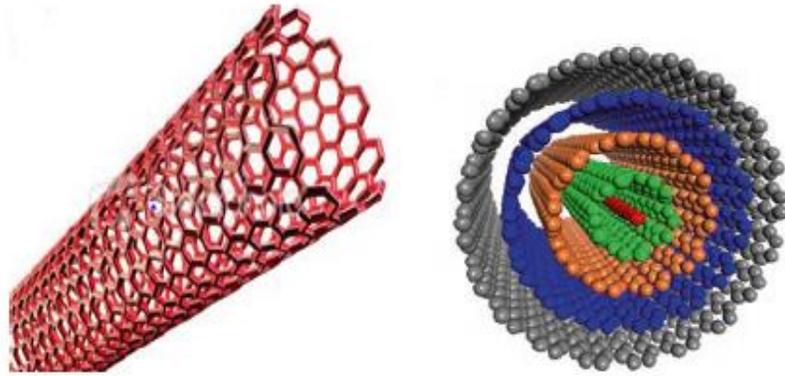
$$\begin{aligned}
 Z(t, \lambda) &= Z_{\text{pred}}(t, \lambda) + F(t, \lambda) \\
 &\approx \sum_{k=0}^Q (Z_k^{(ESC)}(t) + Z_k^{(MCC)}(t)) \phi_k(\lambda) + \sum_{k=Q+1}^P (Z_k^{(ESC)}(t) \phi_k(\lambda)
 \end{aligned} \tag{3.6}$$

It is noted that the total number of deterministic SPICE simulations needed for the predictor-corrector method are  $2(P+1)$  ESC simulations plus the  $2(Q+1)$  MCC simulations. The CPU costs for these multiple SPICE simulations are typically much smaller than that of the  $2(P+1)$  MCC simulations required to estimate the PC coefficients of 3.2. This is verified using a numerical example in further section.

### 3.3 Carbon Nanotubes (CNT)

Numerical example to validate predictor-corrector method is based on carbon nanotubes (CNT). Therefore, before jumping to the example, parameters and properties of CNT are important. Carbon nanotubes are very small tubes, about ten-thousand times finer than a human hair and they are made up of rolled-up sheets of carbon hexagons.

Generally, there are two kinds of CNTs with high structural perfection viz single-walled CNTs (SWNTs) and multi-walled CNTs (MWNTs). SWNTs are made up of just a single graphite sheet seamlessly encased into a cylindrical tube. Multi-walled CNTs (MWNTs) consist of an array of such nanotubes which are concentrically nested like rings of a trunk of a tree.



**Figure 3.4** a) Single-walled CNT b) Multi-walled CNT

Due to such small size, parameters of CNTs are more prone to uncertainties. Several physical parameters like diameter, length, tunnelling conductivity, metallic, and semiconducting properties are also essential for modeling of CNT interconnects [50].

### 3.3.1 Properties of carbon nanotubes

1. Electrical Conductivity: The inter-wall interactions of MWNTs non-uniformly spread the current over each tube, hence making their electrical conductivity pretty complex. The resistivity of SWNTs are generally in the order of  $10^{-4} \Omega \text{ cm}$  at 27 degree C.
2. Elasticity and Strength: Every carbon atom in one individual film of graphite is attached via a robust chemical bond to 3 neighbouring atoms. Therefore, CNTs can manifest the strongest basal plane elastic modulus and consequently are supposed to be an ultimate high-strength fibre. CNTs return to their original state as soon as the force is removed from them, therefore they are very useful for high resolution scanning probe microscopy.
3. Thermal Conductivity: Due to the substantial in-plane C–C bonds of graphene, CNTs can show super-conductivity below 20 K (approx.  $-253 \text{ }^\circ\text{C}$ ). This C-C bond gives exceptional strength and robustness against axial strains. Therefore, CNTs has interesting prospects in nanoscale

electronics, sensing and actuating devices, strengthening additive fibres in composite materials, etc.

4. Field Emission: Due to the strong electric field, tunnelling of electrons from metallic tip to vacuum leads to field emission phenomenon. High aspect ratio and small diameter of CNTs causes field emission. For MWNTs, the field emission properties happen because of the emission of electrons and light. With no applied potential, the light emission happens through the electron field emission and the visible part of the spectrum

5. Aspect Ratio: CNTs have a high aspect ratio, indicating that lower CNT load is expected compared to different conductive additives to attain comparable electrical conductivity. The high aspect ratio of CNTs holds novel electrical conductivity when compared to the standard additive materials like carbon black, or stainless steel fibre.

6. Absorbent: CNTs have been rising as perspective absorbing materials due to their light weight, super flexibility, high mechanical strength, and excellent electrical properties. Hence, CNTs appear to be the ideal applicant for use in gas, water, and air filtration.

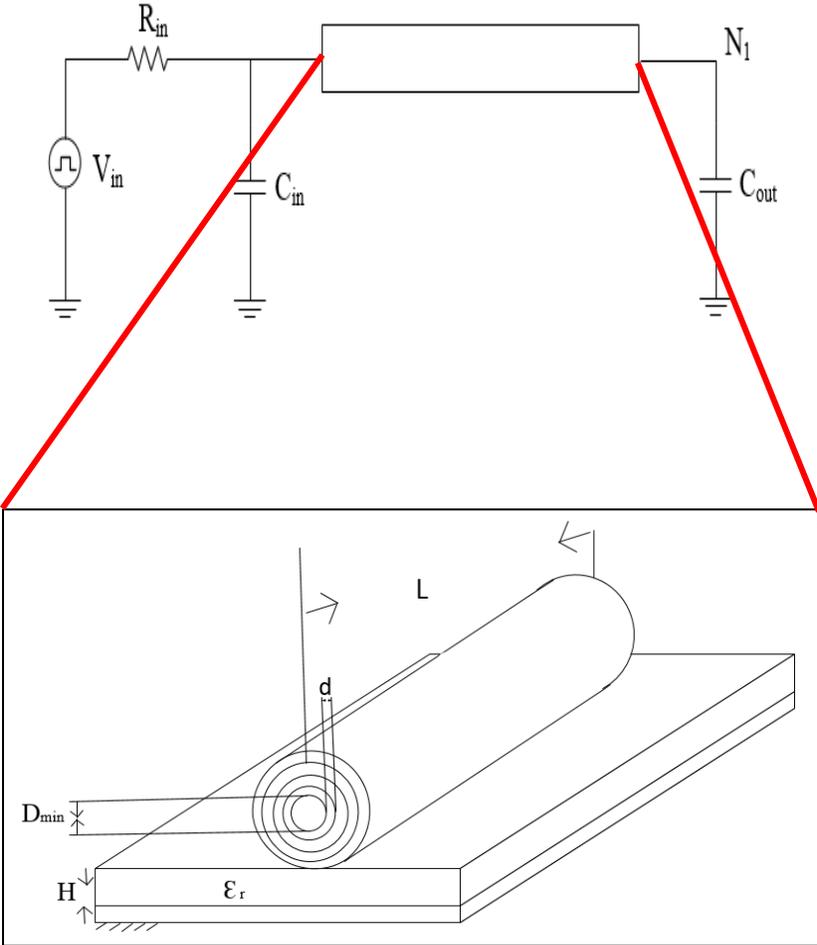
### **3.4 Numerical Example**

To verify the accuracy and efficiency of the proposed predictor-corrector method, MWCNT network shown in figure 3.2 is considered for MCC model and Figure 3.3 is considered for ESC model. The goal is to develop hi-fidelity PC model based on predictor PC model which captures coarse features of the response and a corrector PC model which captures finer details.

So, predictor PC model is based on the simulations from ESC model which is a low-fidelity model, thus it runs really fast and captures the broad output. Corrector PC model is based on the simulations from MCC model so it takes longer (here, for specific CNT problem it is around 100

times slower than ESC model). So here, this has an advantage is that both of them are working with relatively small CPU cost which is better than directly developing PC model from MCC model.

The output that we are testing is the voltage at Node 1 denoted by N1, so we will find the transient response at that node for 0.2ns. Here, the total number of shells taken are  $n = 30$ . The uncertainty in the network is described using the  $N = 9$  random variables and maximum order of 4 i.e.  $m=4$



**Figure 3.5:** MWCNT interconnect inner view

Here, in Figure 3.5,  $D_{\min}$  is the inner diameter of CNT,  $d$  is the inter-shell distance,  $H$  is the height of the dielectric,  $L$  is the length of the conductor,  $\epsilon_r$  is the dielectric constant.

Random variables with their mean values and standard deviation (SD) are shown below in Table 3.1:

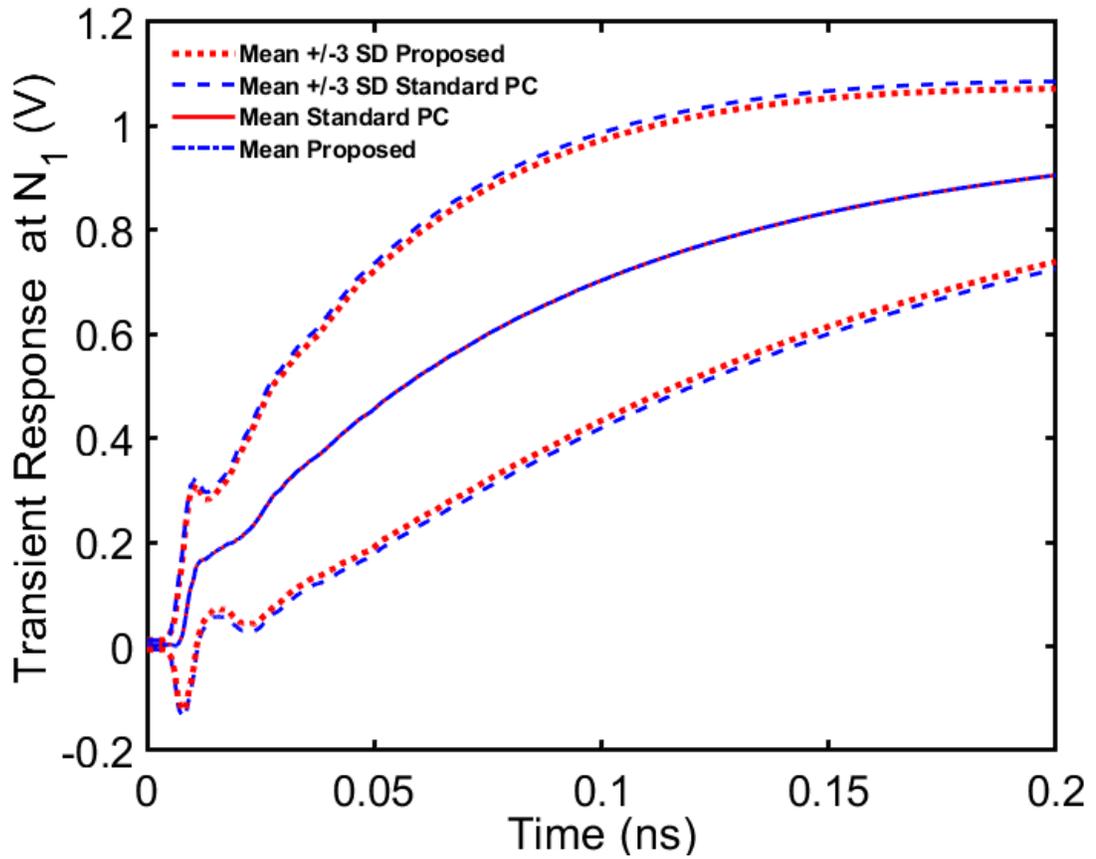
**Table 3.1** Mean and Standard Deviation of the Random Variables

Random Parameter	Mean	SD
$D_{\text{in}}$ (inner diameter of CNT)	2.28 nm	20%
$d$ (inter-shell distance)	0.34 nm	
$\sigma$ (tunneling conductivity)	20	
$C_{\text{in}}$ (driver capacitance)	0.14 fF	
$C_{\text{out}}$ (load capacitance)	0.049 fF	
$H$ (height of the dielectric)	50nm	
$\epsilon_r$ (dielectric constant)	2	
$R_m$ (contact resistance)	1000 Ohms	
$L$ (length of conductor)	100 $\mu\text{m}$	5%

In the example, we have considered nine random parameters which have their own mean values and standard deviation. As we are using normal distribution in the example, so hermite polynomials are being used in the expansion as per the Weiner-Askey scheme, discussed in chapter 2.

Voltage source with a saturated ramp waveform of rise/fall time  $T_r = 0.1$  ps and an amplitude of 1V is used to excite the network. Two methods i.e. predictor-corrector scheme and PC full-blown metamodel are used for the performance assessment of the network.

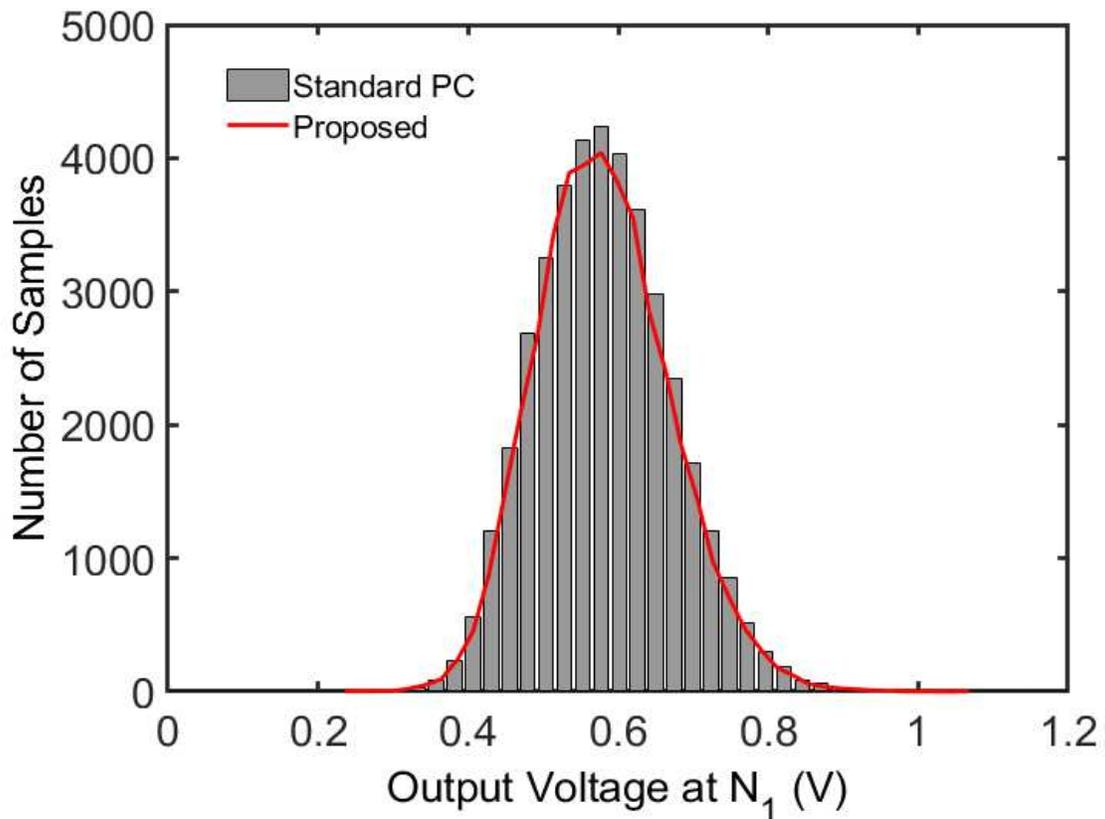
Statistics calculated are plotted in MATLAB and are shown below in figure 3.6 and 3.7.



**Figure 3.6:** Statistics Comparison of transient response at node N1

In the above Figure 3.6, middle curve is the mean of standard PC and the mean from the proposed predictor-corrector method.

The mean calculated from both the methods overlap with each other. Then, mean  $\pm$  3 standard deviation curves are shown for both of the methods which are very similar and are pretty close to each other. Probability density function (PDF) of this output is also computed at the time point of maximum standard deviation i.e.  $SD_{\max} = 0.071\text{ns}$  with 40,000 samples and 40 number of bins. The PDF of proposed predictor-corrector method is shown in red outline curve is compared with PDF of standard PC shown in grey bins. The PDF curves are almost similar to each other depicting the accuracy of the method.



**Figure 3.7:** PDF comparison at the time point of maximum standard deviation

After plotting the statistics of the MWCNT network response, CPU time and speedup is being calculated. Important formulae to calculate CPU time and speedup is given below:

$$\begin{aligned}
 (\text{CPU Time})_{\text{proposed}} &= \text{Time for Predictor} + \text{Time for Corrector} \\
 &= [2(P+1)] * [T1 + T0]
 \end{aligned}
 \tag{3.7}$$

$$(\text{CPU Time})_{\text{standard}} = 2(P+1) T0
 \tag{3.8}$$

Where, T1 is the time for 1 MCC simulation, T0 is the time for 1 ESC simulation. Speedup can be expressed as:

$$\text{Speedup} = \frac{\text{CPU Time for Standard PC}}{\text{CPU Time for Proposed PC}}
 \tag{3.9}$$

Each SPICE simulation of the ESC model is approximately 700 times faster than the MCC model in the proposed predictor-corrector method, therefore, the total CPU time to construct the predictor-corrector metamodel takes only 3199.9 seconds i.e.  $2(P+1) = 1430$  ESC simulations takes 14.3 seconds and the remaining  $2(Q+1) = 440$  MCC simulations takes 3185.6 seconds.

On the other hand, the CPU time to construct the standard PC metamodel of equation 3.2 needs  $2(P+1) = 1430$  MCC simulations which take 10353.2 seconds to finish.

**Table 3.2:** CPU Time and Speedup Comparison of HPCE and Standard PC method

Approach	# of SPICE simulations	CPU time (s)	Speedup
Monte Carlo	40000 (hi)	135200	-
Standard PC	1430 (hi)	10353.2	13x faster than Monte Carlo
Predictor-Corrector	1430 (low) + 440 (hi)	3199.9	3x faster than Standard PC

Hence, the proposed predictor-corrector method is approximately 3 times faster than the standard PC metamodel for the MWCNT example explained here. The accuracy of the statistical responses

achieved using the proposed predictor-corrector method gives a good agreement with that accomplished using the standard PC metamodel of equation 3.2.

## CHAPTER 4: HYPERBOLIC POLYNOMIAL CHAOS EXPANSION (HPCE)

This chapter presents a novel methodology for implementing the PC theory to uncertainty quantification problems. In a conventional PC approach, orthogonal polynomial bases are picked based on a linear criterion, however, in HPCE, the hyperbolic criterion is proposed [51]. This criterion determines the most significant polynomial bases based on the sparsity of effects [52], [49]; therefore, it gives a sparse set of polynomial bases which is much smaller than the original set of bases with no or negligible loss of accuracy.

The chapter begins with a discussion on constructing the multidimensional polynomial bases which is required for a better understanding of the proposed method. Afterward, the criterion for the hyperbolic PC method is presented. Furthermore, the CPU cost of the proposed method and its scaling with respect to the number of random variables is assessed and it is then compared with other PC schemes. Lastly, the accuracy and CPU costs scaling of the proposed HPCE method is validated with a numerical example.

### 4.1 Construction of Multidimensional Polynomial Bases

Formation of multidimensional orthonormal polynomial bases are obtained as a product of 1-D polynomial bases:

$$\Phi_k(\lambda) = \prod_{j=1}^n \Phi_{k_j}(\lambda_j) \quad 4.1$$

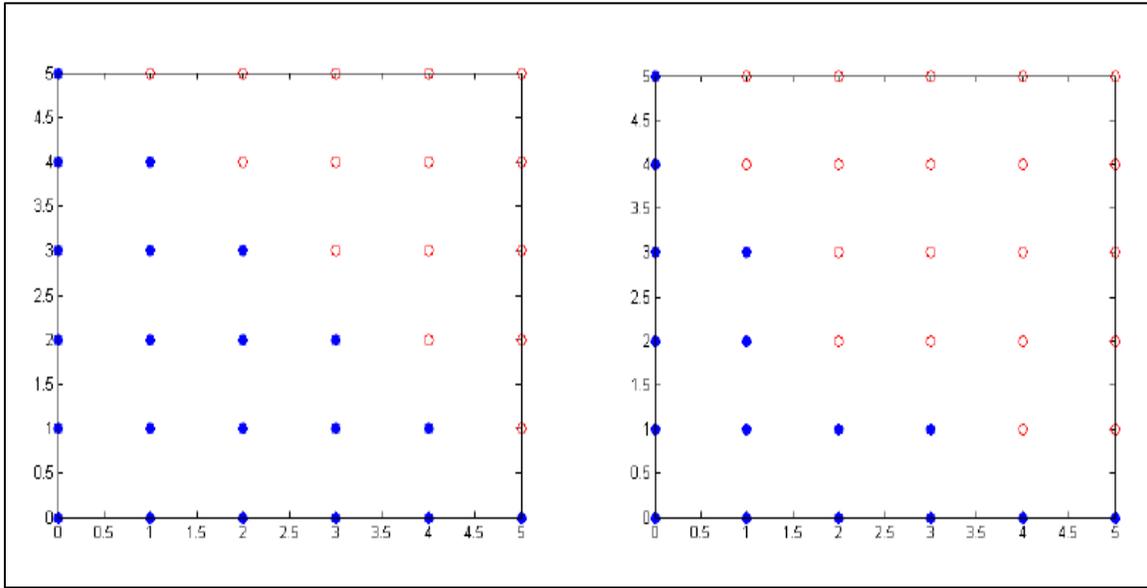
where,  $j^{\text{th}}$  random variable is denoted by  $\lambda_j$ ,  $n$  represents number of random variables,  $\lambda$  shows all random variables in a vector,  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$ ,  $k_j$  is the order of the  $j^{\text{th}}$  1-D polynomial,  $\Phi_{k_j}$  is

the  $j^{\text{th}}$  1-D polynomial,  $d$  denotes all  $d_j$  values in a vector as  $V = [k_1, k_2, \dots, k_n]^T$ , and  $\Phi_k$  is the multidimensional polynomial basis made up of 1-D polynomials whose order is specified in  $V$ .

In standard methods the criterion used to determine the order of each 1-D polynomial in a multidimensional polynomial basis is given by:

$$\|V\|_1 = k_1 + k_2 + \dots + k_n \leq m \quad 4.2$$

where  $\|\cdot\|_1$  denotes the  $L_1$  norm, and  $m$  is the maximum expansion order for 1-D polynomials.



**Figure 4.1:** a) Conventional Linear Truncation Method b) Proposed Hyperbolic method

In figure 4.1, number of random variable  $n=2$  and order  $m =5$  is considered for graphical presentation. Here, blue dots represent indices of chosen polynomial bases and red empty dots represents indices of polynomial bases which are not selected.  $\Phi_{[3,1]}(\lambda) = \Phi_3(\lambda_1) \Phi_1(\lambda_2)$  is chosen as  $\|[3,1]\|_1 = 4 \leq 5$ . Moreover,  $\Phi_{[2,4]}(\lambda) = \Phi_2(\lambda_1) \Phi_4(\lambda_2)$  is not chosen as  $\|[2,4]\|_1 = 6 > 5$ . The number of polynomial bases given by equation 4.2 and illustrated in Fig 4.1 (a) is equal to  $P+1$  in equation 2.14

The standard method provides a moderately good scaling of computational cost with respect to the number of random variables, however, it still bears cumbersome computational costs for problems with a reasonably high number of random variables. According to  $P+1 = \frac{(m+n)!}{m!n!}$ , the scaling rate is  $O(P+1) \approx O(n^m)$ . The near-exponential increment in computational cost is known as the curse of dimensionality and in order to address this issue, a novel method for selection of multidimensional bases is proposed in the next section.

## 4.2 Hyperbolic PC Expansion Truncation Method

As low order polynomial bases have more impact on the network's response, so HPCE selection scheme results in the selection of  $M < P+1$  bases which have the ability to approximate the response as:

$$Z(\lambda) = \sum_{i=0}^{M-1} c_i \Phi_i(\lambda) \quad 4.3$$

In this method, the major constraint is to put on the  $L_u$ <sup>th</sup> norm of the indices vector  $V$ , where  $u \leq 1$ , and the  $L_u$ <sup>th</sup> norm is randomly assigned as the  $u$ <sup>th</sup> root of summation of each term of  $V$  to the power of  $u$ . Replacing the  $L_1$  norm in equation 4.2 with  $L_u$ <sup>th</sup> gives:

$$\|V\|_u = (k_1^u + k_2^u + \dots + k_n^u)^{1/u} \leq m \quad 4.4$$

Where,  $0 \leq u \leq 1$ , here  $k_1$  varies from 0 to  $m$ ,  $k_2$  varies from 0 to  $m$  and so on. Therefore, there would be  $(m+1)^n$  total possible combinations. Now, we have to vary  $k_1$  to  $k_n$  to get  $(m+1)^n$  bases such that it satisfies the equation 4.4. By selecting  $u < 1$ , we automatically guarantee that when constructing the HPCE higher priority is given to the low-degree interactions when compared to their higher-degree equivalents. It is important to note that by increasing the number of random variables, the number of eliminated polynomials in Figure 4.3(b) would also increase very fast,

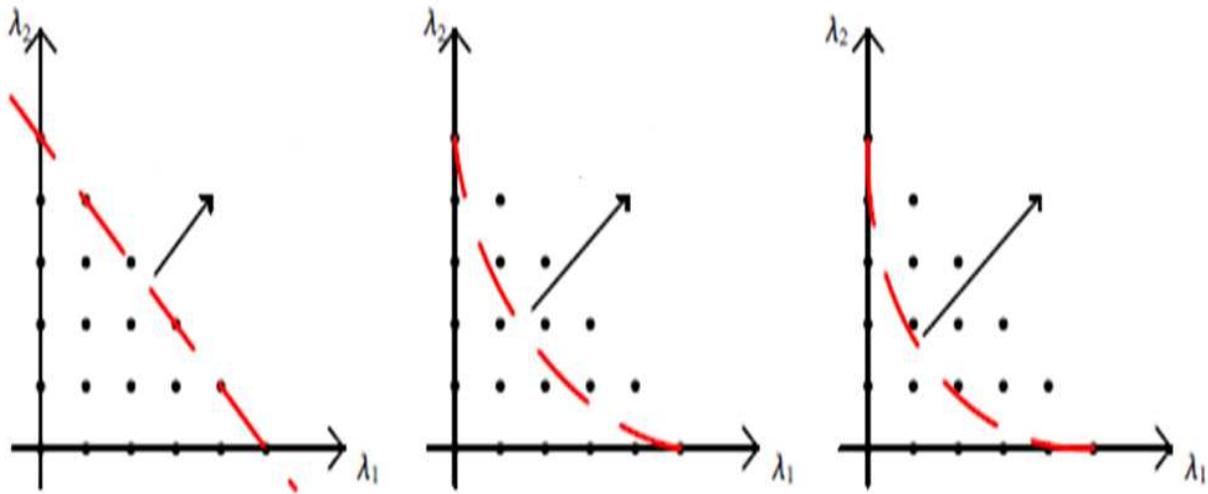
thus enhancing the sparsity of the HPCE. Therefore, HPCE is highly recommended for high-dimensional problems.

Here polynomial bases are restricted between hyperbolic function  $\|V\|_u = m$  and positive axis and is given by:

$$(k_1^u + k_2^u + \dots + k_n^u)^{1/u} = m \quad 4.5$$

Thus, this method is known as hyperbolic and  $u$  is known as hyperbolic factor. This method would significantly decrease the number of chosen indices while maintaining accuracy in PC approaches. To demonstrate the HPCE scheme, Figure 4.1 (b) illustrates the graphical presentation of an example where  $n=2$ ,  $m=5$  and  $u=0.7$ . Here, in the figure filled blue dots denote indices of chosen polynomial bases and empty red dots denote indices of not chosen polynomial bases. For example, in Fig. 4.1(b),  $\Phi_{[3,1]}(\lambda) = \Phi_3(\lambda_1) \Phi_1(\lambda_2)$  is chosen because  $\| [3,1] \|_{0.7} = 4.98 \leq 5$ . Moreover,  $\Phi_{[2,2]}(\lambda) = \Phi_2(\lambda_1) \Phi_2(\lambda_2)$  is not chosen as  $\| [2,2] \|_{0.7} = 5.17 > 5$ . It is important to note that  $\Phi_{[3,1]}(\lambda)$  and  $\Phi_{[2,2]}(\lambda)$  both are of the equivalent degree of 4, and they also have the similar rank of 2; hence they cannot be distinguished using methods discussed in previously which has constraint on the degree or the rank. It is worth to note that the proposed HPCE method does not depend on which uncertainty quantification scheme is being used.

In fact, all PC methods discussed previously are either intrusive or non-intrusive, can be used. This is mere because HPCE is another orthonormal expansion which does not influence the process of determining the coefficients and obtaining statistics.



**Figure 4.2:** Graphical illustration showing standard linear truncation, proposed hyperbolic truncation method and effect of reducing hyperbolic factor on HPCE method (Left to Right)

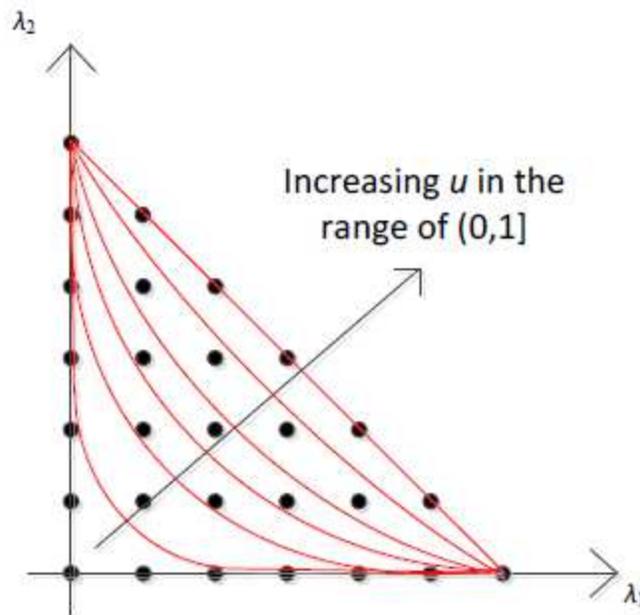
Different values of the hyperbolic factor  $u$  results into a different number of polynomials. This is illustrated in Figure 4.2 for an instance, having  $n=5$  and  $m=4$ , where the standard truncation method is described on the left side of the figure while the two other plots explain that decreasing  $u$  gives a sparser set of polynomial bases. We can also say that  $u=1$  is equal to the standard PC expansion since with  $u=1$  in equation 4.4 converts to equation 4.2; furthermore, a larger value of  $u$  results in higher number of polynomial bases and much better accuracy whereas a smaller value of  $u$  results in lower number of polynomials and also lower accuracy. Based on this, the foremost purpose of HPCE is to obtain the hyperbolic factor  $u$  as it determines the accuracy and efficiency. Hence, a novel technique to obtain the different values of  $u$  discussed in the next section.

### 4.3 Hyperbolic Factor ( $u$ )

The hyperbolic factor  $u$  has a great impact on the accuracy-sparsity trade-off in creating the HPCE. When  $u = 1$ , the HPCE converges to the full-blown PCE which has very high accuracy but also

computationally expensive to construct. However, on the other hand, if the value of  $u$  is very small then all the multi-dimensional bases are ignored and only the 1-D bases are considered.

This results in the sparsest expansion yet is limited in its predictive accuracy. Here, by 1-D bases, it means polynomial bases which have a rank up to 1 which are located on the multidimensional axis in the graphical presentation. As the order of expansion is  $m$  and  $n$  random variables, so the total number of 1-D bases are  $((m*n) + 1)$ , where the single basis at the centre of the graphical presentation is loosely described as 1-D too. Hence, the range of the hyperbolic factor  $u$  is equal to  $(0,1]$ , where 0 is not included because equation 4.4 would not have any answer at  $u=0$ . The variation of  $u$  on its spectrum is presented in Figure 4.3.



**Figure 4.3:** Effect of increasing the value of  $u$  from near zero to 1 on selection of polynomial bases

Therefore, the foremost challenge of the proposed PC method is to tune the hyperbolic factor  $u$  for a general circuit problem. So, a greedy iterative approach is used to increase the hyperbolic factor

u from a very small initial value  $u_0$  till  $u_n$ , where PC expansion is enriched sufficiently to satisfy a predefined error tolerance. Therefore, for any  $j^{\text{th}}$  iteration, the value of u is increased as:

$$u_j = u_{j-1} + \Delta u \quad 4.6$$

Where,  $\Delta u$  is the step size which is fixed and subscript j is the iteration count. When the value of u is increased, it enriches the expansion from previous iteration  $(j-1)^{\text{th}}$  by adding new PC bases terms. The new bases terms have a constraint:

$$\|V\|_{u_{j-1}} > m ; \|V\|_{u_j} \leq m \quad 4.7$$

The coefficients of new bases terms are such selected to minimize the residual error from the previous iteration; so that can be calculated using linear regression. Furthermore, the coefficients calculated from the previous iterations are retained always and do not have to be recomputed. Once the coefficients of the current iteration are calculated, then the variance enrichment the circuit response due to the added new bases is measured as:

$$S_j(t) = \sum Z_i^{(j)^2}(t) \quad 4.8$$

The enrichment should be greater than a prescribed tolerance, for the iteration to continue. Once the enrichment of the above equation falls below that tolerance, it is then believed that the point of diminishing return has reached and the iterations will stop. The resultant PC expansion is known as the HPCE. It is important to note that given  $u < 1$  when the iterations are stopped, it is guaranteed to have sparsity in the HPCE. Furthermore, by increasing the value of hyperbolic factor u in the iterative process, the HPCE is guaranteed to converge to the full-blown PC expansion.

#### 4.4 Statistical Information using HPCE Coefficients

Statistical information is the most important part to obtain. HPCE task is done similarly to standard PC methods which are discussed in previous chapters; nonetheless, in this method, a more confined set of polynomial bases are used. The first statistical moment also known as an arithmetic mean is given by:

$$E(z(\lambda)) = \int_{\Omega} z(\lambda) \rho(\lambda) d\lambda = \sum_{i=0}^M c_i \phi_i(\lambda) \rho(\lambda) d\lambda = \sum_{i=0}^M \langle c_i \phi_i(\lambda), \phi_0(\lambda) \rangle = c_0 \quad 4.9$$

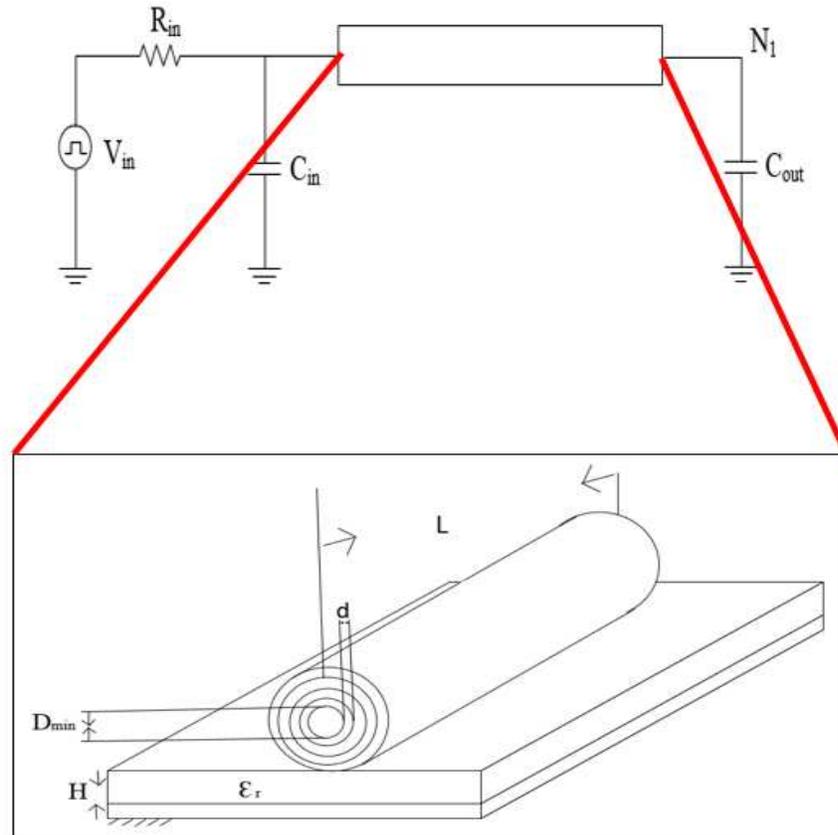
Standard deviation is the square root of variance and variance is the sum of square of all the coefficients upto  $c_M$  except the first one and is expressed as:

$$\begin{aligned} \text{Var}(z(\lambda)) &= E[z(\lambda) - E(z(\lambda))]^2 = \int_{\Omega} (\sum_{i=0}^M c_i \phi_i(\lambda))^2 \rho(\lambda) d\lambda \\ &= \sum_{i=1}^M \sum_{j=1}^M \langle c_i \phi_i(\lambda), c_j \phi_j(\lambda) \rangle \\ &= \sum_{i=1}^M \langle c_i \phi_i(\lambda), c_i \phi_i(\lambda) \rangle \\ &= \sum_{i=1}^M c_i^2 \end{aligned} \quad 4.10$$

To find the PDF and other higher order statistical moments, like skewness and kurtosis, the technique similar to Monte Carlo is used. In this approach, the 1st Q random sample nodes are generated. The number of dimensions and distribution of the sample nodes is similar to the random variables  $\lambda$  in the system. Once we have the coefficients, polynomials  $\phi_0$  to  $\phi_{M-1}$ , and random samples, the right hand side of equation 4.4 is known; so, after finding the coefficients  $c_0$  to  $c_{M-1}$  the result of Q instances of the experiment can be approximated. The final step is to compute the PDF and other higher-order statistical moments from these approximated results.

## 4.5 Numerical Example

In this section, the same MWCNT network is used for validating HPCE results which was used for proposed predictor-corrector method. Then proposed HPCE scheme results are compared with standard PC approach.



**Figure 4.4:** MWCNT interconnect inner view

In this example, number of random variables  $n = 9$ , order  $m = 4$  and number of shells = 50. Random variables for uncertainty quantification with their mean values and standard deviation (SD) are shown below:

**Table 4.1:** Mean and Standard Deviations of Random Variables

Random Parameter	Mean	SD
$D_{in}$ (Inner diameter of CNT)	2.28 nm	20%
$d$ (inter-shell distance)	0.34 nm	
$\sigma$ (tunneling conductivity)	20	
$C_{in}$ (driver capacitance)	0.14 fF	
$C_{out}$ (load capacitance)	0.049 fF	
H (height of the dielectric)	50nm	
$\epsilon_r$ (dielectric constant)	2	
$R_m$ (Contact resistance)	1000 Ohms	
L (length of conductor)	100 $\mu$ m	5%

Voltage source with a saturated ramp waveform of rise/fall time  $T_r = 0.1$ ps and an amplitude of 1V is used to excite the network. In the example, we have considered nine random parameters similar to the previous example which have their own mean values and standard deviation. As we are using normal distribution in the example, so hermite polynomials are being used in the expansion as per the Weiner-Askey scheme

Firstly, values of hyperbolic factor  $u$  are calculated. If the step size of hyperbolic factor i.e.  $\Delta u$  is very small then there would be no difference in the number of polynomial bases at some steps, and if  $\Delta u$  is very big then there would be a significant change in the number of polynomial bases at some steps which could be split into 2 steps by taking finer steps.

Values and corresponding number of bases for  $n=9$  and  $m=4$  are given as:

**Table 4.2:** Hyperbolic Factor and Corresponding Number of Bases

Hyperbolic Factor, $u$	Number of Bases
$u \approx 0$	37
0.5	73
0.69	145
0.79	229
1	715

Here, in this case, there are 5 hyperbolic factors  $u$  and these  $u$  values can be arranged in a vector as  $u = [u_0, 0.5, 0.69, 0.79, 1]$  where  $u_0 \approx 0$  and greater than zero, this will lead to  $L = [37, 73, 145, 229, 715]$  bases respectively. By taking a very fine  $\Delta u$ , it can be shown that it is not possible to distinguish polynomial bases to groups smaller than what is stated in the second column of Table. 4.3. Hence, if  $\Delta u < 0.1$  there would be iterations with negligible change in the expansion; furthermore, if  $\Delta u > 0.1$ , number of chosen polynomials may directly jump from 73 to 229 and thereby forcing the algorithm to do extra computations because multidimensional bases are located on the integer coordinates. Hence, it is feasible to satisfy all bases coordinates using limited number of hyperbolas. Here, only 5 hyperbolas are required, and they can be expressed using equation 4.5 and hyperbolic factors of  $u = [u_0, 0.5, 0.69, 0.79, 1]$ ; therefore, we call these  $u$  values as critical hyperbolic factors.

Next, for  $u = u_0$ , we find optimal nodes  $B_0$  using linear regression method. Here,  $B_0$  is a  $37 \times 1$  row vector and evaluate PC coefficients  $X_0$

$$A_0 X_0 = B_0 \quad 4.11$$

$$PC_0 = X(t, \lambda) - X^{(ESC)}(t, \lambda)$$

$$= \sum_{i=0}^P X_i(t) \phi_i(\lambda) \quad 4.12$$

Similarly, for  $u = u_1$ , we find optimal nodes  $B_1$  using linear regression method  $A_1 X_1 = B_1$ . Here,  $B_1$  is a  $73 \times 1$  row vector and again evaluate PC coefficients

$$X(t, \lambda) = \sum_{i=0}^P X_i(t) \phi_i(\lambda) \quad 4.13$$

Here,  $P+1 = 73$  i.e. 37 old bases + 36 new bases.

$$PC_1 = X(t, \lambda) - X^{(ESC)}(t, \lambda)PC_0 + \sum_{i=37}^{73} X_i(t) \phi_i(\lambda) \quad 4.14$$

$$A_1 = \begin{bmatrix} \left[ \begin{array}{ccc} & & \\ & A_0 & \\ & & \end{array} \right] & \begin{array}{ccc} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{array} & \begin{array}{c} \phi_{M_1-1}(\lambda^{(1)}) \\ \vdots \\ \phi_{M_1-1}(\lambda^{(M_0)}) \end{array} \\ \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} & \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} & \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \\ \phi_0(\lambda^{(M_1)}) \bullet \bullet \bullet \phi_{M_0-1}(\lambda^{(M_1)}) & \begin{array}{ccc} \bullet & \bullet & \bullet \end{array} & \phi_{M_1-1}(\lambda^{(M_1)}) \end{bmatrix}$$

Similarly, the process will continue for  $u_2, u_3, \dots, u_n$ , by expanding information matrix, more SPICE simulations and solving for the new set of HPCE coefficients. Therefore, the general form of the system of linear algebraic equations at step  $j$  can be formulated as:

$$A_j X_j = B_j \quad 4.15$$

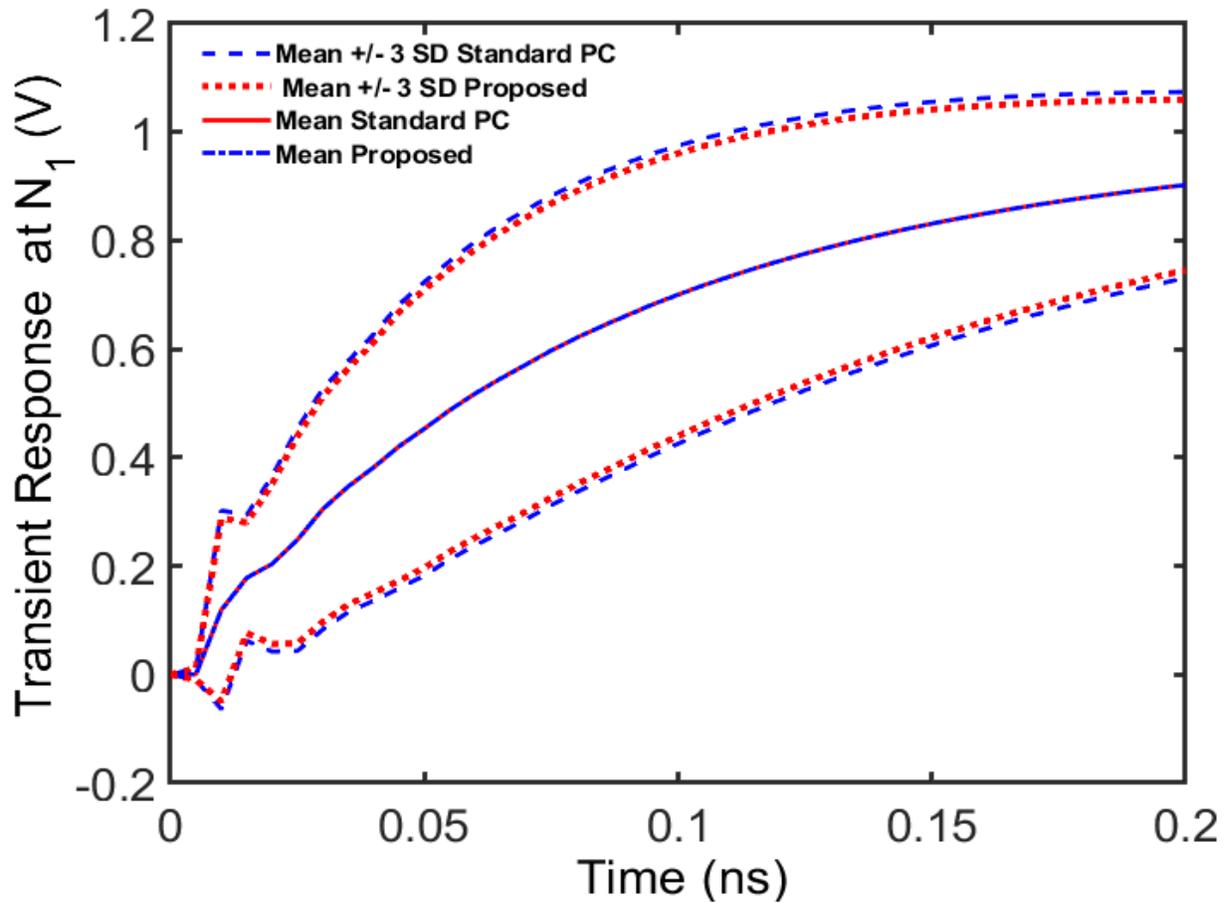
where,

$$X_j = [X_0(t), \dots, X_{M_j-1}]$$

$$B_j = [[B_{j-1}]^T, \dots, X(t, \lambda^{(M_j)})]^T$$

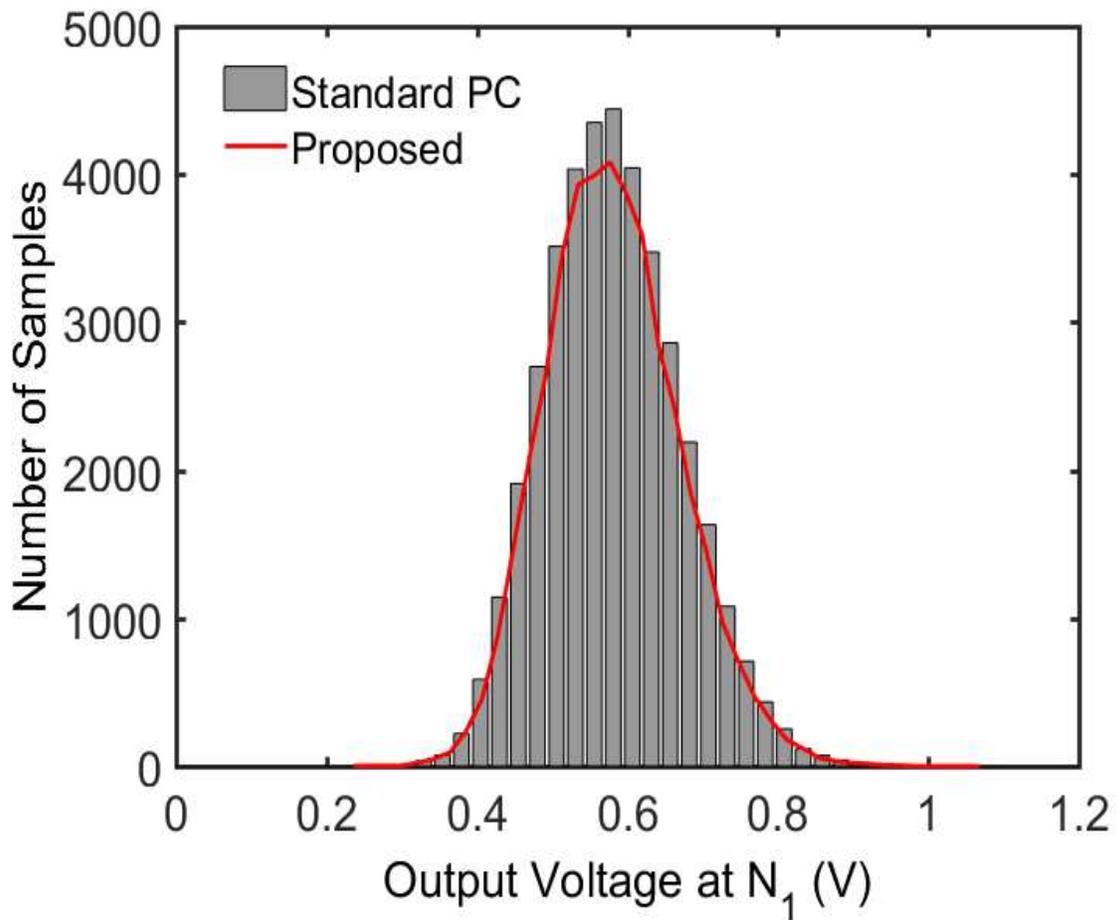
$$\mathbf{A}_j = \begin{bmatrix} \left[ \begin{array}{ccc} & & \\ & \mathbf{A}_{j-1} & \\ & & \end{array} \right] & \cdot & \cdot & \cdot & \phi_{M_{j-1}}(\lambda^{(1)}) \\ & & & & \vdots \\ & & & & \phi_{M_{j-1}}(\lambda^{(M_{j-1})}) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_0(\lambda^{(M_j)}) & \cdot & \cdot & \cdot & \phi_{M_{j-1}}(\lambda^{(M_j)}) \end{bmatrix}$$

Statistics for  $u=0.69$  are calculated and plotted in MATLAB and are shown below in Figure 4.5 and 4.6.



**Figure 4.5:** Statistics Comparison of transient response at node N1

In the above Figure 4.5, middle curve is the mean of standard PC and the mean from the proposed predictor-corrector method. The mean obtained from both of the methods overlap with each other. Then, mean  $\pm$  3 standard deviation curves are shown for HPCE (dotted red) and Standard PC scheme (dotted blue) which are very similar and are pretty close to each other. Probability density function (PDF) of this output is also computed at the time point of maximum standard deviation i.e.  $SD_{max} = 0.072ns$  with 40,000 samples and 40 number of bins. The PDF of proposed HPCE scheme is shown in red outline curve is compared with PDF of standard PC shown in grey bins. The PDF curves are almost similar to each other depicting the accuracy of the method.



**Figure 4.6:** PDF Comparison at the time point of maximum standard deviation

After plotting the statistics of the MWCNT network response, CPU time and speedup is being calculated with the formulae stated in previous chapter. Standard deviation error is expressed as:

$$SD_{\text{error}} = \frac{\sqrt{\sum (SD_{MC} - SD_{Proposed})^2}}{\text{Number of time points}} \quad 4.16$$

Similarly, HPCE model is constructed for different number of shells in MWCNT network for  $u=0.69$  and then compared to standard PC method. This is shown below in the following table:

**Table 4.3:** CPU Time and Speedup Comparison of HPCE and Standard PC Method

Shells	Standard PC		Proposed		Speedup
	CPU Time (s)	SD Error	CPU Time (s)	SD Error	
30	10353.2	1.19E-04	839.35	1.03E-04	12.3x
35	10624.9	1.19E-04	951.01	1.03E-04	11.2x
40	11311.3	1.18E-04	1036.55	1.06E-04	10.9x
45	11640.2	1.21E-04	1081.50	1.01E-04	10.8x
50	12326.6	1.20E-04	1152.55	1.04E-04	10.7x

Here, the proposed HPCE scheme combined with predictor-corrector method is approximately 12 times faster for 30 shells than standard PC scheme which is even faster than proposed predictor-corrector method for same number of shells and same MWCNT example. As the number of shells of MWCNT will increase, the speedup will be saturated to around 10 times than standard PC scheme. The accuracy of the statistical responses achieved using the proposed HPCE method gives a good agreement with that accomplished using the standard PC metamodel.

## CONCLUSION

With the continuous miniaturization in the latest VLSI technologies, the impact of random fabrication process variations and unpredictable operating requirements on the performance of integrated circuits can no longer be neglected. In order to overcome such issues, simulation solvers to model forward propagation of uncertainties or variations in random processes at the device level to the network response are required. The generalized polynomial chaos (PC) theory has evolved as a very robust and versatile method for the statistical analysis of high-speed circuits. Typically, Polynomial Chaos Expansion (PCE) of the random variables is the most common technique to model the unpredictability in the systems.

In this thesis, an overview of standard and contemporary schemes to analyse parametric uncertainty is presented. Then, predictor-corrector polynomial chaos scheme and hyperbolic polynomial chaos expansion (HPCE) scheme are being proposed in order to alleviate the poor scalability of standard PC approaches which are validated using multi-walled carbon nanotubes network as an example. The proposed predictor-corrector method is approximately 3 times faster and HPCE method combined with predictor-corrector method is predictor-corrector method 12 times faster than the standard PC metamodel for MWCNT network having 30 shells. If the conducting shells are increased further, then the proposed technique will still be faster than standard PC scheme and the speedup will be saturated to approximately 10 times. Furthermore, the accuracy of the statistical responses achieved using the proposed methods gives a good agreement with that accomplished using the standard PC metamodel.

## REFERENCES

- [1] F. N. Najm, *Circuit Simulation*. Hoboken, NJ: Wiley, 2010
- [2] G.S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*. New York, NY: Springer-Verlag, 1996
- [3] Q. Zhang, J. J. Liou, J. McMacken, J. Thomson and P. Layman, “Development of robust interconnect model based on design of experiments and multiojective optimization,” *IEEE Trans. Electron Devices*, vol. 48, no. 9, pp. 1885–1981, Sept. 2001
- [4] 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
- [5] 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
- [6] 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
- [7] 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
- [8] 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
- [9] 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
- [10] 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
- [11] 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

- [12] 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
- [13] 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
- [14] 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
- [15] 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
- [16] 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
- [17] 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
- [18] D. Spina, F. Ferranti, G. Antonini, T. Dhaene and L. Knockaert, "Efficient variability analysis of electromagnetic systems via polynomial-chaos and model order reduction," *IEEE Trans. Comp., Packag. and Manuf. Technol.*, vol. 4, no. 6, pp. 1038–1051, June 2014
- [19] 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
- [20] 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
- [21] 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
- [22] 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

- [23] A. K. Prasad, and S. Roy, "Global sensitivity based dimension reduction for fast variability analysis of nonlinear circuits," in Proc, 24th IEEE Conf. Electric. Perform. Electron. Packag., October 2015, pp. 97-100.
- [24] 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
- [25] A. K. Prasad, and S. Roy, "Multidimensional variability analysis of complex power distribution networks via scalable stochastic collocation approach," IEEE Trans. Comp., Packag. and Manuf. Technol., vol. 5, no. 11, pp. 1656-1668, Nov. 2015
- [26] A. K. Prasad, M. Ahadi, and S. Roy, "Multidimensional uncertainty quantification of microwave/RF networks using linear regression and optimal design of experiments," IEEE Trans. Microwave Theory Tech, July 2016
- [27] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits," IEEE Trans. Computer Aided Design, Jan. 2015
- [28] 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
- [29] 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
- [30] 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
- [31] 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
- [32] 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
- [33] 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
- [34] B. K. Kaushik and M. K. Majumder, "Carbon nanotube based VLSI interconnects analysis and design," Springer briefs in Applied Sciences and Technology, 2015

- [35] S. Bhatnagar, A. Merkley, R. Berdine, Y. Li, S. Roy, “Variability-aware performance assessment of multi-walled carbon nanotube interconnects using a predictor-corrector polynomial chaos scheme,” in IEEE Electrical Design of Advanced Packaging and Systems Symposium, Dec. 2018
- [36] A. K. Prasad, D. Zhou, and S. Roy, “Reduced dimension polynomial chaos approach for efficient uncertainty analysis of multi-walled carbon nanotube interconnects,” in Proc. IEEE MTT-S 64th International Microwave Symposium, May 2016, pp. 1-3
- [37] D. Xiu and G. E. Karniadakis, “The Wiener-Askey polynomial chaos for stochastic differential equations,” SIAM Journal on Scientific Computing, vol. 24, no. 2, pp. 619–644, 2002
- [38] Skewness [<https://en.wikipedia.org/wiki/Skewness>]
- [39] Kurtosis [<https://en.wikipedia.org/wiki/Kurtosis>]
- [40] G. H. Golub and J. H. Welsch, “Calculation of Gauss Quadrature Rules”, Math. Comp., vol. 23, pp. 221–230, 1969.
- [41] 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
- [42] 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
- [43] V. V. Fedorov, “Theory of Optimal Experiments”, NY: Academic Press, New York, 1972.
- [44] R. D. Cook and C. J. Nachtsheim, “A comparison of algorithms for constructing exact D-optimal designs”, Amer. J. Math., vol. 60, pp. 897–936, 1938.
- [45] Y. Ye, D. Spina, P. Manfredi, D. Vande Ginste, and T. Dhaene, “A comprehensive and modular stochastic modeling framework for the variability-aware assessment of signal integrity in high-speed links,” IEEE Trans. Electromagnetic Compatibility, vol. 60, no. 2, pp. 459-467, April 2018
- [46] Y. Tao, B. Nouri, M. S. Nakhla, M. A. Farhan, and R. Achar, “Variability analysis via parameterized model order reduction and numerical of Laplace transform,” IEEE Trans. Comp., Packag. and Manuf. Technol., vol. 7, no. 5, pp. 678–686, May 2017
- [47] D. Xiu, “Fast numerical methods for stochastic computatios: a review,” Comm. In Comput. Phys. vol. 5, no. 2-4, pp. 242-272, Feb. 2009
- [48] M. S. Sarto and A. Tamburrano, “Single-conductor transmission line model of multiwall carbon nanotubes,” IEEE Trans. Nanotechnol., vol. 9, no. 1, pp. 82–92, Jan. 2010
- [49] D.C. Montgomery, Design and Analysis of Experiment. 8th ed. John Wiley, NY, 2012

[50] B.K Kaushik, M.K Majumdar, Carbon Nanotube Based VLSI Interconnects Analysis and Design, SPRINGER, 2015

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

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