Practical Implementation of Stochastic Parameterized Model Order Reduction via Hermite Polynomial Chaos *

Yi Zou† Yici Cai† Qiang Zhou† Xianlong Hong† Sheldon X.-D. Tan‡ Le Kang†

†Department of Computer Science and Technology, Tsinghua University, Beijing 10084, P.R.C
‡Department of Electrical Engineering, University of California, Riverside, CA 92521

Abstract— This paper describes the stochastic model order reduction algorithm via stochastic Hermite Polynomials from the practical implementation perspective. Comparing with existing work on stochastic interconnect analysis and parameterized model order reduction, we generalized the input variation representation using polynomial chaos (PC) to allow for accurate modeling of non-Gaussian input variations. We also explore the implicit system representation using sub-matrices and improved the efficiency for solving the linear equations utilizing block matrix structure of the augmented system. Experiments show that our algorithm matches with Monte Carlo methods very well while keeping the algorithm effective. And the PC representation of non-gaussian variables gains more accuracy than Taylor representation used in previous work[9].

I. Introduction

As we step into the nanometer design regime, circuit and device parameters are no longer deterministic and they suffer from distinct types of variations [6]. The circuit analysis and simulation methods have to consider countless variations explicitly to meet the variational design requirements in the nanometer scale. Model order reduction (MOR) technique, which change a large linear system into a small one, is an important technique to enhance the efficiency for interconnect analysis. In order to consider the variational issues in interconnect analysis and model order reduction, many approaches have been proposed.

The perturbational MOR [5, 10] focus on finding the interconnect performance under perturbational variations, which means circuit parameters are perturbed a little near the nominal value. The lower order polynomials captured via regression/fitting [10] or two-step matching[5] could accurately represent the perturbed transfer functions and capture the perturbed system performance under perturbational variations. But if the variations are modeled as stochastic variations, the lower order polynomials from the perturbational approach can only generate good results under small perturbations near the nominal value, and did not consider stochastic distributions of variations thus can not ensure good stochastic metrics. Interval or affine based MOR[2] use a specialized computer arithmetic which take place of traditional scalar arithmetic to generate the system performance under variations. Statistical/stochastic interpretation for the arithmetic is also given in [2], but error explosion and instability in long computation chains may occur thus not very scalable for large circuits, although it is fast and do not require or assume stochastic distributions.

Parametrical variations, in fact, are stochastic/statistical variations, and stochastic interconnect analysis in fact is a simulation or solving of a stochastic process. Stochastic parameterized model order reduction, which considers stochastic variational parameters during the process of MOR, is the topic of this paper. Although some theories and results of the stochastic interconnect analysis via hermite polynomials chaos have already been reported in [9] and the parametrical model order reduction based on the stochastic formulation of the augmented system has been briefly discussed in that work, there are still several issues not addressed.

First, Taylor expansions were used to generate the stochastic input variation of non-gaussian variables in [9] but the Taylor expansion itself did not take the distribution information into consideration thus may not be optimal for non-gaussian variations. Second, after the augmented system equation is built, any model order reduction technique could be used for the augmented system. But the block structure of the system equation of the augmented system provides great possibility for efficiency improvement of subsequent MOR but have not been explored in that work. In this paper, we follow the same theoretical idea in that work but we focus on the practical design and implementation issues concerning the Stochastic Model Order Reduction (SMOR) utilizing hermite polynomial expansions, including the representation of the complicated input variation using polynomial chaos, and how to express the system memory-efficiently in sub-matrix level and how to efficiently solve the governing equations utilizing the block structure of the linear equation during the model order reduction. The three issues which have not been discussed in [9], are important portions for implementing a practical algorithm of SMOR.

Our paper is organized in the following way: Section II gives an overview of the algorithm of stochastic interconnect analysis and SMOR. Section III describes the representation of variation, and how to convert non-Gaussian variation into a polynomial chaos representation. Section IV discuss how to represent the system using blocks of submatrices. Section V discuss the efficient solver utilizing matrix structures. Section VI explains the experiment results on stochastic model order reduction of interconnect analysis with Gaussian and non-Gaussian variations. Finally, conclusions and future works are given.

*This work is supported by the National Science Foundation of China 60476014.

1-4244-0630-7/07/$20.00 ©2007 IEEE.

II. Overview of the stochastic analysis of interconnect

A. Notations and basic flow of the stochastic analysis of interconnect and SMOR

Here we briefly review the background and notation for stochastic interconnect analysis and stochastic MOR. A more detailed theoretical description could be found in [9].

The interconnect for analysis could be represented in the system equation written in MNA form:

$$\begin{aligned}(G+Cs)x &= Bu \\ y &= L^T x\end{aligned} \tag{1}$$

When stochastic variations are considered of the MNA form, the above equation shall become

$$\begin{aligned}(\hat{G}+\hat{C}s)\hat{x} &= \hat{B}u \\ \hat{y} &= \hat{L}^T\hat{x}\end{aligned} \tag{2}$$

in which the $\hat{G}$,$\hat{C}$ etc are represented using an affine form of random variations:

$$\hat{G} = G_0 + G_1\varepsilon_1 + G_2\varepsilon_2 + \cdots + G_n\varepsilon_n \tag{3}$$

$$\hat{C} = C_0 + C_1\varepsilon_1 + C_2\varepsilon_2 + \cdots + C_n\varepsilon_n \tag{4}$$

ε_i are random variables denoting stochastic parametrical variations, here we assume they are independent. If the random variables are correlated, a Karhunen-Loeve expansion [3] which use the covariance function of random variables could remove the correlations.

The next procedure is to generate an augmented system from the Eq.(2). Clearly, from Eq.(2), we could see $\hat{x}$ is a function of ε_i. Explicitly expand $\hat{x}$ using an affine form of a set of basis function w_i which is also function of ε_i:

$$\hat{x} = x_0w_0 + x_1w_1 + x_2w_2 + \cdots + x_kw_k + \cdots \tag{5}$$

Here x_i are deterministic coefficients while w_i are polynomial basics accounts for stochastic variations. This expansion is in fact infinite which means it should contain infinite number of terms. Practically, truncations should be performed. The typical function basis are Hermite polynomials for Gaussian random variables. The zero order Hermite polynomial is 1, the first order Hermite polynomials are ε_i, the second order Hermite polynomials are $\varepsilon_i\varepsilon_j (i \neq j)$ and $\varepsilon_i^2 - 1$. Higher order Hermite polynomials are also available. ε_i is normalized independent gaussian variables.

As the finite truncation of Eq.(5) is performed, the truncation error should be minimized. Based on Galerkin method, the minimization of the truncation error is further transformed into several equations on the inner product over the basis functions. Put Eq.(5) into Eq.(2), and perform the inner product of each functional basis with Eq.(2), or from another perspective, project the equation onto that function basis

$$\langle w_i, LHS\rangle = \langle w_i, RHS\rangle \tag{6}$$

where the inner product

$$\langle f1, f2\rangle = E(f1 * f2) \tag{7}$$

Eq.(6) holds in order to minimize the truncation error, it gives an augmented system equation of deterministic x_i, which could further been investigated via model order reduction.

Generally speaking, the stochastic interconnect analysis considers stochastic input variations and tries to found the response represented as linear combination of Hermite Polynomials as the function basis.

B. A simple illustrative example of stochastic interconnect analysis

For illustration, a simple example is shown: Suppose only one gaussian random variable is considered and a second order Hermite polynomials are used as basis:

$$((G_0 + G_1\varepsilon_1) + s(C_0 + C_1\varepsilon_1))(x_0 + x_1\varepsilon_1 + x_2(\varepsilon_1^2 - 1)) \approx b_0$$

Expand that equation, we could obtain:

$$\begin{aligned}&(G_0x_0 + G_0x_1\varepsilon_1 + G_0x_2(\varepsilon_1^2 - 1) \\ +\ &G_1x_0\varepsilon_1 + G_1x_1\varepsilon_1^2 + G_1x_2(\varepsilon_1^3 - \varepsilon_1)) \\ +\ &s(C_0x_0 + C_0x_1\varepsilon_1 + C_0x_2(\varepsilon_1^2 - 1) \\ +\ &C_1x_0\varepsilon_1 + C_1x_1\varepsilon_1^2 + C_1x_2(\varepsilon_1^3 - \varepsilon_1)) \approx b_0\end{aligned}$$

Perform the inner product of $w_0 : 1$; $w_1 : \varepsilon_1$ and $w_2 : \varepsilon_1^2 - 1$ with the LHS and RHS respectfully, we could get three block linear equations:

$$\begin{aligned}&((G_0x_0 + G_1x_1) + s(C_0x_0 + C_1x_1)) = b_0 \\ &((G_1x_0 + G_0x_1 + 2G_1x_2) + s(C_1x_0 + C_0x_1 + 2C_1x_2) = 0 \\ &(2G_1x_1 + 2G_0x_2) + s(2C_1x_1 + 2C_0x_2) = 0\end{aligned}$$

. Thus the augmented system could be obtained from the equations above:

$$\left(\begin{bmatrix} G_0 & G_1 & 0 \\ G_1 & G_0 & 2G_1 \\ 0 & 2G_1 & 2G_0 \end{bmatrix} + s\begin{bmatrix} C_0 & C_1 & 0 \\ C_1 & C_0 & 2C_1 \\ 0 & 2C_1 & 2C_0 \end{bmatrix}\right)\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_0 \\ 0 \\ 0 \end{bmatrix}$$

Then traditional MOR algorithm may be further be applied to obtain the stochastic response. Even from this small example, we could see that the system equations of the augmented system are intrinsically represented in block matrices, thus block algorithms may be available to enhance the efficiency of MOR algorithm.

In the next sections, we present algorithm improvements and practical implementing techniques to utilize the special matrix property of stochastic MOR to give a more widely applicable while efficient SMOR algorithm.

III. Polynomial chaos representation of input variations

A. Reasons and methods to support non-gaussian variations

The first issue we discuss in SMOR is the support of non-gaussian stochastic input variations in Eq.(3,4). Although, in most cases, manufacturing variations are modeled as Gaussian variations, there are still several reasons that non-Gaussian variations needs to be considered and supported. First, it is likely some statistical information of the variation collected

from measured data, we should perform the variational simulation according to the measured statistics rather than empirical Gaussian assumption. Another possibility is that Gaussian geometrical variations may still generate non-gaussian variations of RLC electrical parameters due to some non-linearities in extraction process. Also, variations due to thermal effects or leakage effects could not be modeled as Gaussian. So the variational simulator and underlying variational model order reduction should be able to cope with variations modeled as non-gaussian to be widely applicable.

The askey scheme [1, 9] change the basis functions series w_i for projection according to the stochastic distribution of the variations ε_i and it can guarantee faster stochastic convergence. However, to implement several different functional basis for stochastic simulation is not an easy job. Also the askey scheme did not provide solution for arbitrary stochastic input variations.

Another scheme, which is also the method used in this paper, only implements hermite polynomial chaos representation (solely using polynomials of Gaussian variables similar to the example in the previous section) in the framework of SMOR. So in order to support non-Gaussian input variables, transformation from non-Gaussian input variations into functions of Gaussian variables is needed.

B. Demonstrations and comparisons of two types of transformations

Two types of such transformation are available: Taylor representation and polynomial chaos representation. In [9], Taylor expansions are used to cope with non-Gaussian variations in an example for coping lognormal input variations. In order to demonstrate the difference between the two approach, here, we also make the illustration of lognormal variations as an example.

Consider a lognormal random variable k, $k = e^{\varepsilon}$, here ε is a random variable with normal distribution. The transformation using Taylor models using the Taylor expansions:

$$e^{\varepsilon} = 1 + \varepsilon + \varepsilon^2/2 + \varepsilon^3/6 + \cdots$$

If finite truncation is used e.g a second order truncation, the equations become

$$e^{\varepsilon} \approx 1 + \varepsilon + \varepsilon^2/2 \tag{8}$$

Later this equations is further substituted into the equation of input variations e.g Eq.(3,4), and the SMOR based on gaussian hermite polynomials still could be used subsequently.

The polynomial chaos (PC) representation, on the other hand, seek the truncated polynomials which match the inner product of the function basis selected to minimize the stochastic truncation error, e.g, for the second order PC representation:

$$e^{\varepsilon} \approx a * 1 + b * \varepsilon + c * (\varepsilon^2 - 1)$$

$1, \varepsilon, \varepsilon^2 - 1$ are zero order, first order, second order Hermite polynomials respectfully, and they are orthogonal with each other. Similarly, using these Hermite polynomials to conduct inner product with LHS and RHS, we could obtain:

$$\begin{aligned} a &= < e^{\varepsilon}, 1 > / < 1, 1 > = < e^{\varepsilon}, 1 > \\ b &= < e^{\varepsilon}, \varepsilon > / < \varepsilon, \varepsilon > = < e^{\varepsilon}, \varepsilon > \\ c &= < e^{\varepsilon}, \varepsilon^2 - 1 > / < e^{\varepsilon}, \varepsilon^2 - 1 > = < e^{\varepsilon}, \varepsilon^2 - 1 > /2 \end{aligned}$$

Thus $a = e^{1/2}, b = e^{1/2}, c = e^{1/2}/2$. so that

$$e^{\varepsilon} \approx e^{1/2}/2 + e^{1/2}\varepsilon + e^{1/2}\varepsilon^2/2 \tag{9}$$

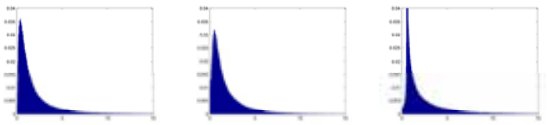
There are several prosperities of the truncated PC representation. First, it keeps the stochastic expectation:

$$E(e^{\varepsilon}) = E(e^{1/2}/2 + e^{1/2}\varepsilon + e^{1/2}\varepsilon^2/2)$$

Second, it is the polynomial (here means the second order polynomial in this example, a higher order could give better results obviously) which minimized the stochastic truncations error $E((e^{\varepsilon} - (e^{1/2}/2 + e^{1/2}\varepsilon + e^{1/2}\varepsilon^2/2)^2)$. While the polynomials generated from Taylor models do not have these properties. Both the p.d.f of the PC representation and that of the Taylor representation are close to that provided by Monte-Carlo (See Fig.1). A function evaluation comparison of e^x and third order Taylor and third order PC is shown in Fig.2. We could see from the figure that Taylor model may be accurate near the mean value while PC model prevail in stochastic statistics and is more accurate in the far ends.Some statistics are also compared and PC models are more accurate in both mean value and variance (See Table I). So the PC representation of input stochastic variation is a more preferable choice.

TABLE I
STATISTICS COMPARISON OF DIFFERENT REPRESENTATION OF A LOGNORMAL VARIABLE

Monte Carlo		Taylor (3rd Order)		PC (3rd Order)	
mean	std	mean	std	mean	std
1.6482	2.1570	1.4999	1.7072	1.6482	2.1257



(a) Monte Carlo (b) Taylor (3rd order) (c) PC (3rd order)

Fig. 1. p.d.f comparison of different representation of a lognormal random variable

C. Support for arbitrary stochastic input variations

The lognormal variation demonstrated in the previous subsection could be explicitly represented as a function over a gaussian variables: e^{ε}. Not every variation could be written in that obviously explicit form such as uniform distribution or distributions represented in histogram form. However, this form could be obtained via cumulative probability function CDF.

Let k denote that random variable,

$$CDF(k) = CDF(\varepsilon) \tag{10}$$

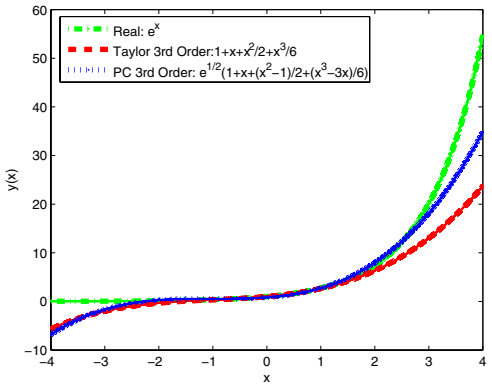


Fig. 2. Function evaluation comparison of the lognormal variable

defines a transformation or mapping from gaussian variable ε to that random variable k. An explicit from is:

$$k = CDF^{-1}(CDF(\varepsilon)) \tag{11}$$

Then we could perform numerical integration of the inner products to calculate the coefficients of Hermite polynomial function basis and represent that variation with polynomials of gaussian variables similar to examples of lognormal variable in previous subsection, and perform SMOR subsequently.

IV. Representation and Calculation of the Augmented System

Now, we discuss the techniques in forming the augmented system. Because of the effect of allowing for non-gaussian variations, the system matrices $\hat{G}$, $\hat{C}$ are also represented in a truncated PC form:

$$\hat{G} = G_0 + G_1 w_1 + G_2 w_2 + \cdots + G_{N_w} w_{N_w} \tag{12}$$

$$\hat{C} = C_0 + C_1 w_1 + C_2 w_2 + \cdots + C_{N_w} w_{N_w} \tag{13}$$

Here the N_w is the number of the basis functions. Since $w_0 = 1$, the system equation: $(\hat{G} + s\hat{C})\hat{x} = \hat{b}$ could be rewritten as

$$\sum_{i=0}^{N_w}\sum_{j=0}^{N_w} G_i x_j w_i w_j + s\sum_{i=0}^{N_w}\sum_{j=0}^{N_w} C_i x_j w_i w_j = \sum_{i=0}^{N_w} b_i w_i \tag{14}$$

and then the block equation generated from the inner product of w_k should be

$$\begin{aligned} & \sum_{i=0}^{N_w}\sum_{j=0}^{N_w} G_i x_j \langle w_i w_j, w_k\rangle + s\sum_{i=0}^{N_w}\sum_{j=0}^{N_w} C_i x_j \langle w_i w_j, w_k\rangle \\ = & \sum_{i=0}^{N_w}\sum_{j=0}^{N_w} G_i x_j E(w_i w_j w_k) + s\sum_{i=0}^{N_w}\sum_{j=0}^{N_w} C_i x_j E(w_i w_j w_k) \\ = & \sum_{i=0}^{N_w} b_i E(w_i w_k) \qquad (k = 0, 1, \cdots, N_w) \end{aligned} \tag{15}$$

Because the orthogonality of the hermite polynomials $E(w_i w_k) = 0 (i \neq k)$, and $E(w_i^2)$ could be easily computed and put in a table for look-up further. The LHS, on the other hand, requires some special treatment. Rewritten Eq.(15) into a block matrix form:

$$\begin{cases} (G_{aug} + sC_{aug})* = b_{aug} \\ y = L_{aug}^T x_{aug} \end{cases} \tag{16}$$

where

$$G_{aug} = \begin{bmatrix} A_{00} & \cdots & A_{0N_w} \\ \vdots & \ddots & \vdots \\ A_{N_w 0} & \cdots & A_{N_w N_w} \end{bmatrix} \tag{17}$$

$$C_{aug} = \begin{bmatrix} B_{00} & \cdots & B_{0N_w} \\ \vdots & \ddots & \vdots \\ B_{N_w 0} & \cdots & B_{N_w N_w} \end{bmatrix} \tag{18}$$

$$b_{aug} = \begin{bmatrix} b_0 \\ \vdots \\ 0 \end{bmatrix} \; x_{aug} = \begin{bmatrix} x_0 \\ \vdots \\ x_{N_w} \end{bmatrix} \; L_{aug} = \begin{bmatrix} Lw_0 \\ \vdots \\ Lw_{N_w} \end{bmatrix} \tag{19}$$

$$A_{jk} = \sum_{i=0}^{N_w} G_i E(w_i w_j w_k) \tag{20}$$

$$B_{jk} = \sum_{i=0}^{N_w} C_i E(w_i w_j w_k) \tag{21}$$

So we got the matrix form of the system equation of the augmented system. There are some observations from that equation. First, the matrix G_{aug} and C_{aug} are block symmetric, which means $A_{jk} = A_{kj}$, $B_{jk} = B_{kj}$. If the system G_i is symmetric, e.g the matrices for RC case, the matrices of the augmented system G_{aug}, C_{aug} are also symmetric. Second, for a memory efficient implementation, we do not need to represent matrix G_{aug} and C_{aug} explicitly using $N_w * N_w$ submatrices, but they could be determined using the matrix G_i,C_i and the inner product $E(w_i w_j w_k)$, it would cost only about $1/N_w$ memory of the explicit form.

And $E(w_i w_j w_k)$ could be computed using symbolic or numerical integration before the augmented system is built. In [3], some tables for the $E(w_i w_j w_k)$ is already given in advance.

V. Efficient Solving of Linear Equations of the Augmented System

A. Model order reduction via PRIMA

After the augmented system is built, we then perform the model order reduction on the augmented system. The model order reduction method we use is PRIMA [4] which is a widely used model order reduction algorithm. Fig.3 shows the main algorithmic flow of PRIMA. From the flow we could see the most costly process of the PRIMA is to solving linear equations $Gx = (G^{-1}C)^i R$.

As the dimension of the augmented system is much larger than the original system, direct methods via LU decomposition

1.start from the augmented system $G_{aug}, C_{aug}, b_{aug}$ in (16).
2.Solve $G_{aug}X_0 = b$.
3.Orthogonalize and normalize each column in X_0.
For $k = 1, 2, \cdots$
    4.Solve $G_{aug}X_k = C_{aug}X_{k-1}$
    5.Orthogonalize and normalize each column in X_k
end for
6.$X = [X_0 X_1 \cdots X_k]$.
7.Obtain the reduced system matrix from
$\tilde{G} = X^T G_{aug} X, \tilde{C} = X^T C_{aug} X$,
$\tilde{b} = X^T b_{aug}, \tilde{L} = X^T L_{aug}$.
8.Perform the eigen decomposition $\tilde{G}^{-1}\tilde{C} = S\Lambda S^{-1}$
9.Obtain the poles, residues and transfer functions

Fig. 3. The Main algorithmic flow of MORstat, conducting PRIMA on augmented system

may be prohibitive. Krylov space iterative methods which even do not need the explicit matrix form but only need the Matrix-vector product may be a better choice.

The matrix-vector product could be constructed implicitly in sub-matrix level with the help of the sub-matrices representation of the block equations equation (20,21). Suppose we need to compute the matrix-vector multiplication in the Krylov space iterative methods:

$$\begin{bmatrix} A_{00} & \cdots & A_{0N_w} \\ \vdots & \ddots & \vdots \\ A_{N_w 0} & \cdots & A_{N_w N_w} \end{bmatrix} * \begin{bmatrix} x_0 \\ \vdots \\ x_{N_w} \end{bmatrix} = \begin{bmatrix} r_0 \\ \vdots \\ r_{N_w} \end{bmatrix} \quad (22)$$

We could obtain the results using the implicict representation of the matrix (20,21), thus we have:

$$r_i = \sum_{j=0}^{N_w} A_{ij} * x_j = \sum_{i=0}^{N_w}\sum_{j=0}^{N_w} G_i * x_j * E(w_i w_j w_k) \quad (23)$$

The matrix-vector multiplication using the implicit sub-matrix representation (20,21) may seem to be more complicated than direct matrix-vector representation. However, as there are a lot of zero in the table of $E(w_i w_j w_k)$, implicit matrix-vector multiplication is also very efficient. Besides, the results of sub-matrix vector multiplication $G_i * x_j$ could be reused for efficiency.

B. Efficient solving of the system equation via block Jacobi preconditioners

Although the krylov space method such as Conjugated Gradient(CG) and GMRES could be used to solve the system expressed in an implicit matrix form. The convergence speed may be slow. Preconditioner should be used to speed up the krylov space method.

However, as the system is expressed in an implicit form, we use a block Jacobi preconditioner on higher block level and an ILUTP preconditioner on lower sub-matrix level. As the scale of the variations are one or two magnitudes less than the nominal value, thus the whole augmented system is block diagonal dominant, but at sub-matrix level, the matrix may have many zero entries on the diagonal, the MNA matrix may be non-symmetric or non-definite (for RLC case), the ILUTP preconditioner is robust and the preconditioner greatly speeds up the krylov space method for solving the linear equations for PRIMA. Moreover, the block diagonals are mostly the scalar multiple of G_0 (see the small examples in the section II), we only need to generate the ILUTP preconditioner of G_0 once to form the preconditioner of the augmented system, which is also much faster than to built a ILUTP preconditioner for the large augmented system.

VI. Experiment results

We implemented the algorithms of stochastic model order reduction in a software prototype called MORstat using C/C++ language. The sparse matrix library MORstat use is GMM++[7] which also has SuperLU and LAPACK interface. We tested our algorithms on a PC work station with AMD 2000+, 768M memory running GNU Linux.

We test several RC/RLC circuits under normal stochastic variations or log-normal variations. The circuit size ranges from 113 nodes to 5452 nodes. We use second order polynomial expansion in our experiments, and four random variables are incorporated in the experiment. Table II shows the running-time comparison and memory consumption comparison of the Monte-Carlo simulation using PRIMA, Morstat using explicit matrix formulation and direct solver, Morstat using implicit matrix formulation and preconditioned iterative solver, respectfully. It is not clear which MOR method the work in [9] use and how the linear equations concerning the augmented system is solved. But we found that our block-jacobi preconditioned krylov method needs roughly about $1/N_w$ memory while have the same or even faster speed for solving the augmented system. In RC case, we use the block Jacobi plus incomplete cholsky preconditioned CG method and it runs 2-5 times faster then direct method using SuperLU. While in RLC case, we use block Jacobi plus ILUTP preconditioned GMRES method, it only slightly faster than direct method. But the implicit iteration method saves a great memory than the direct method.

As many results of the the stochatic interconnect analysis with Gaussian variations was reported in [9]. We omit the results about SMOR on the pure Gaussian case. Table III shows the accuracy comparison of the mean and variance comparison on the circuit 50% delay under log-normal stochastic parametrical variations. We compare the results of the Monte-Carlo simulation using PRIMA, Morstat using Taylor expansion for input variation, Morstat using using Polynomial expansion for input variation. The input variations are assumed log-normal, four random variations are incorporated, and the number of sampling in Monte Carlo is 4000. Clearly we can see using PC representation of the input stochastic variations can improve the accuracy of MOR than the Taylor representation.

TABLE II
RUNNING-TIME AND MEMORY CONSUMPTION COMPARISON OF DIFFERENT METHODS

Monte Carlo		Morstat(Direct)		Morstat(Iterative)	
time(s)	mem(M)	time(s)	mem(M)	time(s)	mem(M)
78m35s	0.7	2m11s	8.7	1m37s	2.2
325m44s	1.8	14m11s	84.7	5m15s	13.2
956m9s	6.4	31m24s	231.8	11m47s	45.1

TABLE III
ACCURACY COMPARISON OF DIFFERENT METHODS

Monte Carlo		Morstat(Taylor)		Morstat(PC)	
mean(ns)	std(ns)	mean(ns)	std(ns)	mean(ns)	std(ns)
12.22	1.719	12.13	1.724	12.23	1.721
19.75	1.921	19.55	1.911	19.77	1.923
25.86	2.412	25.82	2.414	25.88	2.409

VII. CONCLUSIONS AND FUTURE REMARKS

We present specific implementation techniques of stochastic model order reduction algorithm via stochastic Hermite Polynomials expansions. Comparing with existing work on stochastic parameterized model order reduction, we generalize the input variation representation using polynomial chaos to allow for accurate modeling of non-Gaussian input variations. We also provide an implicit representation using sub-matrices for the augmented system and improved the efficiency for solving the linear equations using the special matrix structure of the of the augmented system. Experiments show that our algorithm matches with Monte Carlo methods very well while keeping the algorithm effective, and the PC representation of non-gaussian variables gains more accuracy than Taylor representation used in previous work.

REFERENCES

[1] D.Xiu and G.E.Karniadakis, "Modeling uncertainty in flow simulations via generalized polynomial chaos", *Journal of Computational Physics*, no.187, pp. 137-167, 2003.

[2] J.D.Ma and R.A.Rutenbar, "Fast interval valued statistical interconnect modeling and reduction", *Proc. ISPD*, pp. 159-166, 2005.

[3] R.G.Ghanem and P.D.Spanos, "Stochastic finite elements: A spectral approach", Springer-Verlag, 1993.

[4] A.Odabasioglu, M.Celik and L.T.Pileggi,"PRIMA: passive reduced-order interconnect macromodeling algorithm", *Proc of ICCAD*, pp. 58-65, 1997.

[5] X.Li, P.Li and L. T. Pileggi, "Parameterized Interconnect Order Reduction with Explicit-and-Implicit Multi-Parameter Moment Matching for Inter/Intra-Die Variations", *Proc. ICCAD*, pp. 806-812, 2005.

[6] V.Mehrotra, "Modeling the effects of systematic processvariation on circuit performance", Ph.D. Thesis, MIT, May, 2001.

[7] Gmm++, "www-gmm.insa-toulouse.fr/getfem/gmm.html"

[8] K.Agarwal, D.Sylvester and D.Blaauw, "Variational delay metrics for interconnect timing analysis ", *Proc. DAC*, pp. 381-384, 2004.

[9] J.Wang, P.Ghanta and S.Vrudhula, "Stochastic Analysis of Interconnect Performance in the Presence of Process Variations", *Proc. ICCAD*, pp. 880-886, 2004.

[10] Y.Liu, L.T.Pileggi and A.J.Strojwas, "Model Order-Reduction of RC(L) Interconnect including Variational Analysis", *Proc. DAC*, pp. 201-206, 1999.