A Big-Data Approach to Handle Process Variations: Uncertainty Quantification by Tensor Recovery

Zheng Zhang, Tsui-Wei Weng and Luca Daniel
Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
E-mail: z_zhang@mit.edu, twweng@mit.edu, luca@mit.edu

Abstract—**Stochastic spectral methods have become a popular technique to quantify the uncertainties of nano-scale devices and circuits. They are much more efficient than Monte Carlo for certain design cases with a small number of random parameters. However, their computational cost significantly increases as the number of random parameters increases. This paper presents a big-data approach to solve high-dimensional uncertainty quantification problems. Specifically, we simulate integrated circuits and MEMS at only a small number of quadrature samples; then, a huge number of (e.g., 1.5×10^{27}) solution samples are estimated from the available small-size (e.g., 500) solution samples via a low-rank and tensor-recovery method. Numerical results show that our algorithm can easily extend the applicability of tensor-product stochastic collocation to IC and MEMS problems with over 50 random parameters, whereas the traditional algorithm can only handle several random parameters.**

I. Introduction

Fabrication process variations can significantly decrease the yield of nano-scale chip design [1]. In order to estimate the uncertainties of chip performance, Monte Carlo [2], [3] has been used in commercial electronic design automation software for decades. Monte Carlo is easy to implement but generally requires a large number of repeated simulations due to its slow convergence rate. In recent years, stochastic spectral methods [4], [5] have emerged as a promising alternative due to their higher efficiency for certain design cases.

Stochastic spectral methods approximate a stochastic solution as a linear combination of some generalized polynomial chaos basis functions [6], and they may get highly accurate solutions without (or with only a small number of) repeated simulations. Based on intrusive (i.e., non-sampling) formulations such as stochastic Galerkin [4] and stochastic testing [7] as well as sampling-based formulations such as stochastic collocation [5], extensive results have been reported for IC [7]–[13], MEMS [14], [15] and photonic [16] applications.

Unfortunately, the computational cost of stochastic spectral methods increases very fast as the number of random parameters increases. In order to solve high-dimensional problems, several advanced algorithms have been developed. For instance, analysis of variance (ANOVA) [14] and compressed sensing [17] can exploit the sparsity of high-dimensional generalized polynomial-chaos expansion. The dominant singular vector method [18] can exploit the low-rank property of the matrix formed by all coefficient vectors. Stochastic model-order reduction [19] can remarkably reduce the number of simulations for sampling-based solvers. With devices or subsystems described by high-dimensional generalized polynomial-chaos expansions, hierarchical uncertainty quantification based on tensor-train decomposition [15] can handle more complex systems by changing basis functions.

This paper provides a big-data approach for solving the challenging high-dimensional uncertainty quantification problem. Specifically, we investigate tensor-product stochastic collocation that was only applicable to problems with a few random parameters. We represent the huge number of required solution samples as a tensor [20] (i.e., a high-dimensional generalization of matrix) and develop a low-rank and sparse recovery method to estimate the whole tensor from only a small number of solution samples. This technique can easily extend the applicability of tensor-product stochastic collocation to problems with over 50 random parameters, making it even much more efficient than sparse-grid stochastic collocation. This paper aims at briefly presenting the key idea and showing its application in IC and MEMS. Theoretical and implementation details can be found in our preprint manuscript that is focused on power system applications [21].

II. Preliminaries

A. Uncertainty Quantification using Stochastic Collocation

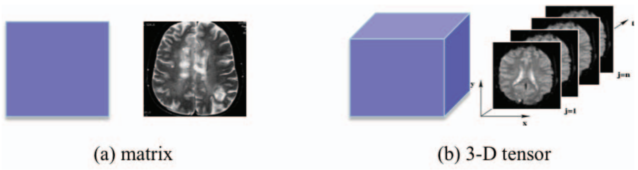
Let $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_d] \in \mathbb{R}^d$ denotes a set of mutually independent random parameters that describe process variations. We aim to estimate the uncertainty of $y(\boldsymbol{\xi})$, which is a parameter-dependent output of interest (e.g., the power consumption or frequency of a chip design). When y smoothly depends on $\boldsymbol{\xi}$ and when it has a bounded variance, a truncated generalized polynomial-chaos expansion can be applied

$$y(\boldsymbol{\xi}) \approx \sum_{|\boldsymbol{\alpha}|=0}^{p} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \text{ with } \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})\right] = \sigma_{\boldsymbol{\alpha},\boldsymbol{\beta}}. \quad (1)$$

Here $\mathbb{E}$ denotes expectation, σ denotes a Delta function, the basis functions $\{\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\}$ are some orthonormal polynomials, $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_d] \in \mathbb{N}^d$ is a vector indicating the highest polynomial order of each parameter in the corresponding basis. The total polynomial order $|\boldsymbol{\alpha}|$ is bounded by p, and thus the total number of basis functions is $(p+d)!/(p!d!)$.

The coefficient $c_{\boldsymbol{\alpha}}$ can be obtained by a projection:

$$c_{\boldsymbol{\alpha}} = \int_{\mathbb{R}^d} y(\boldsymbol{\xi}) \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \text{ with } \rho(\boldsymbol{\xi}) = \Pi_{k=1}^{d} \rho_k(\xi_k) \quad (2)$$



(a) matrix (b) 3-D tensor

Fig. 1. (a) a 2-D data array (e.g., a medical image) is a matrix, (b) a 3-D data array (e.g., multiple slices of images) is a tensor.

where $\rho(\boldsymbol{\xi})$ is the joint probability density function and $\rho_k(\xi_k)$ the marginal density of ξ_k. The above integral needs to be evaluated with some numerical techniques. This paper considers the tensor-product implementation. Specifically, let $\{(\xi_k^{i_k}, w_k^{i_k})\}_{i_k=1}^{q}$ be q pairs of quadrature points and weights for parameter ξ_k, the integral in (2) is evaluated as

$$c_{\boldsymbol{\alpha}} = \sum_{1\leq i_1,\cdots,i_d\leq q} y(\boldsymbol{\xi}_{i_1\cdots i_d})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_{i_1\cdots i_d})w_{i_1\cdots i_d} \tag{3}$$

where $\boldsymbol{\xi}_{i_1\cdots i_d} = [\xi_1^{i_1},\cdots,\xi_d^{i_d}]$ and $w_{i_1\cdots i_d} = w_1^{i_1}\cdots w_d^{i_d}$. Very often, evaluating each solution sample $y(\boldsymbol{\xi}_{i_1\cdots i_d})$ requires a time-consuming numerical simulation (e.g., a RF circuit simulation that involves periodic steady-state computation or a detailed device simulation by solving a large-scale partial differential equation or integral equation formulation).

B. Tensor and Tensor Decomposition

1) Tensor: Tensor is a high-dimensional generalization of matrix. A matrix $\mathbf{X}\in\mathbb{R}^{n_1\times n_2}$ is a 2nd-order tensor, and its element indexed by (i_1,i_2) can be denoted as $x_{i_1 i_2}$. For a general dth-order tensor $\boldsymbol{\mathcal{X}}\in\mathbb{R}^{n_1\times\cdots n_d}$, its element indexed by $(i_1,\cdots,i_d)$ can be denoted as $a_{i_1\cdots i_d}$. Fig. 1 shows a matrix and a 3rd-order tensor. Given any two tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ of the same size, their inner product is defined as

$$\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Y}}\rangle := \sum_{i_1\cdots i_d} x_{i_1\cdots i_d} y_{i_1\cdots i_d}. \tag{4}$$

The Frobenius norm of tensor $\boldsymbol{\mathcal{X}}$ is further defined as $||\boldsymbol{\mathcal{X}}||_F := \sqrt{\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{X}}\rangle}$.

2) Tensor Decomposition: A tensor $\boldsymbol{\mathcal{X}}$ is rank-1 if it can be written as the outer product of some vectors:

$$\boldsymbol{\mathcal{X}} = \mathbf{u}_1\circ\cdots\circ\mathbf{u}_d \;\Leftrightarrow\; x_{i_1\cdots i_d} = \mathbf{u}_1(i_1)\cdots\mathbf{u}_d(i_d) \tag{5}$$

where $\mathbf{u}_k(i_k)$ denotes the i_k-th element of vector $\mathbf{u}_k\in\mathbb{R}^{n_k}$. Similar to matrices, a low-rank tensor can be written as the sum of some rank-1 tensors:

$$\boldsymbol{\mathcal{X}} = \sum_{j=1}^{r} \mathbf{u}_1^j\circ\cdots\circ\mathbf{u}_d^j. \tag{6}$$

As a demonstration, Fig. 2 shows the low-rank factorizations of a matrix and 3rd-order tensor, respectively.

III. Tensor Recovery Approach

Formulation (3) is only applicable to problems with 5 or 6 random parameters due to the q^d simulation samples required. This section describes our tensor-recovery approach that can significantly reduce the computational cost and extend (3) to design cases with many random parameters.

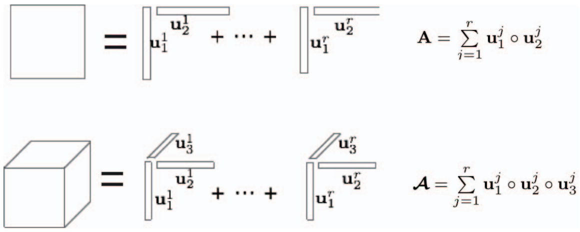

$\mathbf{u}_1^1$
$\mathbf{u}_2^1$
$+\cdots+$
$\mathbf{u}_1^r$
$\mathbf{u}_2^r$
$\mathbf{A}=\sum_{j=1}^{r}\mathbf{u}_1^j\circ\mathbf{u}_2^j$
$\mathbf{u}_3^1$
$\mathbf{u}_3^r$
$\boldsymbol{\mathcal{A}}=\sum_{j=1}^{r}\mathbf{u}_1^j\circ\mathbf{u}_2^j\circ\mathbf{u}_3^j$

Fig. 2. Low-rank factorization of a matrix (top) and of a 3rd-order tensor (bottom).

A. Stochastic Collocation Using Tensor Recovery

1) Reformulation with Tensors: We define two tensors $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{W}}_{\boldsymbol{\alpha}}$, such that their elements indexed by $(i_1,\cdots i_d)$ are $y(\boldsymbol{\xi}_{i_1\cdots i_d})$ and $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_{i_1\cdots i_d})w_{i_1\cdots i_d}$ respectively. Then, the operation in (3) can be written with tensors in a compact way:

$$c_{\boldsymbol{\alpha}} = \langle\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{W}}_{\boldsymbol{\alpha}}\rangle. \tag{7}$$

It is straightforward to show that $\boldsymbol{\mathcal{W}}_{\boldsymbol{\alpha}}$ is a rank-1 tensor. Our focus is to compute $\boldsymbol{\mathcal{Y}}$. Once $\boldsymbol{\mathcal{Y}}$ is computed, stochastic collocation can be done easily. Unfortunately, directly computing $\boldsymbol{\mathcal{Y}}$ is impossible for high-dimensional cases, since it requires simulating a design problem q^d times.

2) Tensor Recovery Approach: In order to reduce the computational cost, we estimate $\boldsymbol{\mathcal{Y}}$ using an extremely small number of its elements. Let $\mathcal{I}$ include all indices for the elements of $\boldsymbol{\mathcal{Y}}$, and its subset Ω includes the indices of a few available tensor elements obtained by circuit or MEMS simulation. With the sampling set Ω, a projection operator $\mathbb{P}$ is defined for $\boldsymbol{\mathcal{Y}}$:

$$\boldsymbol{\mathcal{B}} = \mathbb{P}_\Omega(\boldsymbol{\mathcal{Y}}) \;\Leftrightarrow\; b_{i_1\cdots i_d} = \begin{cases} y_{i_1\cdots i_d}, \text{ if } i_1\cdots i_d\in\Omega \\ 0, \text{ otherwise.}\end{cases} \tag{8}$$

We want to find a tensor $\boldsymbol{\mathcal{X}}$ such that it matches $\boldsymbol{\mathcal{Y}}$ for the elements specified by Ω:

$$\|\mathbb{P}_\Omega(\boldsymbol{\mathcal{X}}-\boldsymbol{\mathcal{Y}})\|_F^2 = 0. \tag{9}$$

However, this problem is **ill-posed**, because any value can be assigned to $x_{i_1\cdots i_d}$ if $i_1\cdots i_d\notin\Omega$.

3) Regularize Problem (9): In order to make the tensor recovery problem well-posed, we add the following constraints based on practical observations.

- **Low-Rank Constraint:** we observe that very often the high-dimensional simulation data array $\boldsymbol{\mathcal{Y}}$ has a low tensor rank. Therefore, we expect that its approximation $\boldsymbol{\mathcal{X}}$ also has a low-rank property. Thus, we assume that $\boldsymbol{\mathcal{X}}$ has a rank-r decomposition described in (6).
- **Sparse Constraint:** as shown in practical cases [14], [17], most of the coefficients in a high-dimensional generalized polynomial-chaos expansion have very small magnitude. This implies that ℓ_1-norm of all coefficients

$$\sum_{|\boldsymbol{\alpha}|=0}^{p}|c_{\boldsymbol{\alpha}}| \approx \sum_{|\boldsymbol{\alpha}|=0}^{p}|\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{W}}_{\boldsymbol{\alpha}}\rangle| \tag{10}$$

should be very small.

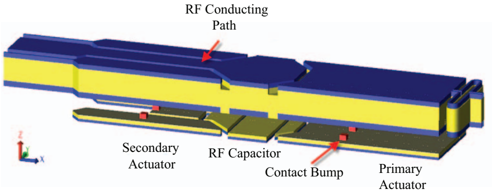


Fig. 3. Schematic of the RF MEMS capacitor [22].

TABLE I
COMPARISON OF SIMULATION COST FOR THE MEMS CAPACITOR.

method	tensor product	sparse grid	proposed
simulation samples	8.9×10^{21}	4512	300

Combining the low-rank and sparse constraints together, we suggest the following optimization problem to compute $\boldsymbol{\mathcal{X}}$ as an estimation (or approximation) of $\boldsymbol{\mathcal{Y}}$:

$$\min_{\{\mathbf{u}_k^1,\cdots,\mathbf{u}_k^r\}_{k=1}^d} \frac{1}{2}\|\mathbb{P}_\Omega\left(\sum_{j=1}^r \mathbf{u}_1^j \circ \cdots \circ \mathbf{u}_d^j - \boldsymbol{\mathcal{Y}}\right)\|_F^2$$
$$+\lambda \sum_{|\boldsymbol{\alpha}|=0}^{p} |\left\langle \sum_{j=1}^r \mathbf{u}_1^j \circ \cdots \circ \mathbf{u}_d^j, \boldsymbol{\mathcal{W}}_{\boldsymbol{\alpha}} \right\rangle|. \quad (11)$$

In this formulation, we compute the vectors that describe the low-rank decomposition of $\boldsymbol{\mathcal{X}}$. This treatment has a significant advantage: the number of unknown variables is dqr, which is only a linear function of parameter dimensionality d. We pick λ and the size of Ω by empirical cross validations.

4) Solve Problem (11): The optimization problem (11) is solved iteratively in our implementation. Specifically, starting from a provided initial guess of the low-rank factors $\{\mathbf{u}_k^1, \cdots, \mathbf{u}_k^r\}_{k=1}^d$, alternating minimization is performed recursively using the result of previous iteration as a new initial guess. Each iteration of alternating minimization consists of d steps. At the k-th step, the r vectors $\{\mathbf{u}_k^1, \cdots, \mathbf{u}_k^r\}$ corresponding to parameter ξ_k are updated by keeping all other factors fixed and by solving (11) as a convex optimization problem.

IV. NUMERICAL RESULTS

In order to verify our tensor-recovery uncertainty quantification algorithm, we show the simulation results of two high-dimensional IC and MEMS examples. All codes are implemented in MATLAB and run on a Macbook with 2.5-GHz CPU and 16-G memory.

A. MEMS Example (with 46 Random Parameters)

We consider the MEMS device in Fig. 3, and we intend to approximate its capacitance a 2nd-order generalized polynomial-chaos expansion of 46 process variations. As shown in Table I, using 3 Gauss-quadrature points for each parameter, a tensor-product integration requires $3^{46} \approx 8.9 \times 10^{21}$ simulation samples, and the Smolyak sparse-grid technique requires 4512 simulation samples.

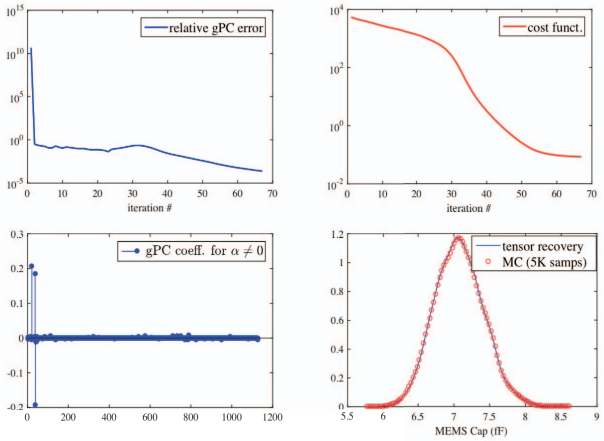


Fig. 4. Numerical results of the MEMS capacitor, with $\lambda = 0.01$. Top left: relative error of the generalized polynomial-chaos coefficients in iterations; top right: decrease of the cost function in (11); bottom left: sparsity of the obtained generalized polynomial-chaos expansion; bottom right: obtained probability density function compared with that from Monte Carlo.

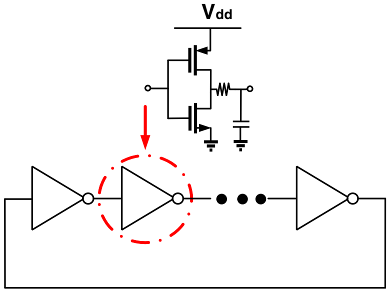


Fig. 5. Schematic of the CMOS ring oscillator.

We simulate this device using only 300 quadrature samples randomly selected from the tensor-product integration rules, then our tensor recovery method estimates the whole tensor $\boldsymbol{\mathcal{Y}}$ [which contains all 3^{46} samples for the output $y(\boldsymbol{\xi})$]. The relative approximation error for the whole tensor is about 0.1% (measured by cross validation). As shown in Fig. 4, our optimization algorithm converges with less than 70 iterations, and the generalized polynomial-chaos coefficients are obtained with a small relative error (below 10^{-4}); the obtained model is very sparse, and the obtained density function of the MEMS capacitor is almost identical with that from Monte Carlo. Note that the number of repeated simulations in our algorithm is only about $1/4$ of the total number of basis functions.

B. 7-Stage CMOS Ring Oscillator (with 57 Parameters)

We further consider the CMOS ring oscillator in Fig. 5. This circuit has 57 random parameters describing the variations of threshold voltages, gate-oxide thickness, and effective gate length/width. We intend to obtain a 2nd-order polynomial-chaos expansion for its frequency by calling a periodic steady-state simulator repeatedly. The required number of simulations for different algorithms are listed in Table II, which clearly shows the superior efficiency of our approach for this example.

TABLE II

COMPARISON OF SIMULATION COST FOR THE RING OSCILLATOR.

method	tensor product	sparse grid	proposed
simulation samples	1.6×10^{27}	6844	500

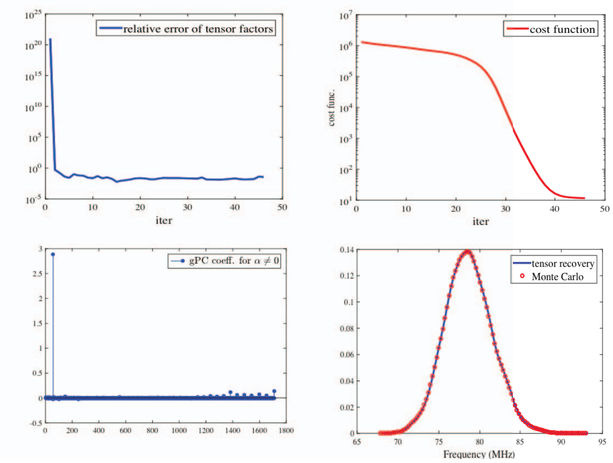


Fig. 6. Numerical results of the ring oscillator, with $\lambda = 0.1$. Top left: relative error of the tensor factors for each iteration; top right: decrease of the cost function in (11); bottom left: sparsity of the obtained generalized polynomial-chaos expansion; bottom right: obtained density function compared with that from Monte Carlo using 5000 samples.

We simulate this circuit using only 500 samples randomly selected from the $3^{57} \approx 1.6 \times 10^{27}$ tensor-product integration samples, then our algorithm estimates the whole tensor $\boldsymbol{\mathcal{Y}}$ with a 1% relative error. As shown in Fig. 6, our optimization algorithm converges after 46 iterations, and the tensor factors are obtained with less than 1% relative errors; the obtained model is very sparse, and the obtained density function of the oscillator frequency is almost identical with that from Monte Carlo. Note that the number of our simulations (i.e., 500) is much smaller than the total number of basis functions (i.e., 1711) in the generalized polynomial-chaos expansion.

V. CONCLUSION

This paper has presented a big-data approach for solving the challenging high-dimensional uncertainty quantification problem. Our key idea is to estimate the high-dimensional simulation data array from an extremely small subset of its samples. This idea has been described as a tensor-recovery model with low-rank and sparse constraints. Simulation results on integrated circuits and MEMS show that our algorithm can be easily applied to problems with over 50 random parameters. Instead of using a huge number of (e.g., about 10^{27}) quadrature samples, our algorithm requires only several hundreds which is even much smaller than the number of basis functions.

ACKNOWLEDGMENT

This work was supported by the NSF NEEDS program and by AIM Photonics under Project MCE_EPDA004.

REFERENCES

[1] D. S. Boning, “Variation,” *IEEE Trans. Semiconductor Manufacturing*, vol. 21, no. 1, pp. 63–71, Feb 2008.

[2] S. Weinzierl, “Introduction to Monte Carlo methods,” NIKHEF, Theory Group, The Netherlands, Tech. Rep. NIKHEF-00-012, 2000.

[3] A. Singhee and R. A. Rutenbar, “Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design,” *IEEE Trans. on CAD of Integrated Circuits and systems*, vol. 28, no. 8, pp. 1176–1189, Aug. 2009.

[4] R. Ghanem and P. Spanos, *Stochastic finite elements: a spectral approach*. Springer-Verlag, 1991.

[5] D. Xiu and J. S. Hesthaven, “High-order collocation methods for differential equations with random inputs,” *SIAM J. Sci. Comp.*, vol. 27, no. 3, pp. 1118–1139, Mar 2005.

[6] D. Xiu and G. E. Karniadakis, “The Wiener-Askey polynomial chaos for stochastic differential equations,” *SIAM J. Sci. Comp.*, vol. 24, no. 2, pp. 619–644, Feb 2002.

[7] Z. Zhang, T. A. El-Moselhy, I. A. M. Elfadel, and L. Daniel, “Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, Oct. 2013.

[8] P. Manfredi, D. V. Ginste, D. D. Zutter, and F. Canavero, “Stochastic modeling of nonlinear circuits via SPICE-compatible spectral equivalents,” *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 61, no. 7, pp. 2057–2065, July 2014.

[9] I. S. Stievano, P. Manfredi, and F. G. Canavero, “Parameters variability effects on multiconductor interconnects via hermite polynomial chaos,” *IEEE Trans. Compon., Packag., Manufacut. Tech.*, vol. 1, no. 8, pp. 1234–1239, Aug. 2011.

[10] K. Strunz and Q. Su, “Stochastic formulation of SPICE-type electronic circuit simulation with polynomial chaos,” *ACM Trans. Modeling and Computer Simulation*, vol. 18, no. 4, pp. 15:1–15:23, Sep 2008.

[11] Z. Zhang, T. A. El-Moselhy, P. Maffezzoni, I. A. M. Elfadel, and L. Daniel, “Efficient uncertainty quantification for the periodic steady state of forced and autonomous circuits,” *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 60, no. 10, Oct. 2013.

[12] R. Pulch, “Modelling and simulation of autonomous oscillators with random parameters,” *Mathematics and Computers in Simulation*, vol. 81, no. 6, pp. 1128–1143, Feb 2011.

[13] M. Rufuie, E. Gad, M. Nakhla, R. Achar, and M. Farhan, “Fast variability analysis of general nonlinear circuits using decoupled polynomial chaos,” in *Workshop Signal and Power Integrity*, May 2014, pp. 1–4.

[14] Z. Zhang, X. Yang, G. Marucci, P. Maffezzoni, I. M. Elfadel, G. Karniadakis, and L. Daniel, “Stochastic testing simulator for integrated circuits and MEMS: Hierarchical and sparse techniques,” in *Proc. IEEE Custom Integrated Circuits Conf.* San Jose, CA, Sept. 2014, pp. 1–8.

[15] Z. Zhang, I. Osledets, X. Yang, G. E. Karniadakis, and L. Daniel, “Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition,” *IEEE Trans. CAD of Integrated Circuits and Systems*, vol. 34, no. 1, pp. 63 – 76, Jan 2015.

[16] T.-W. Weng, Z. Zhang, Z. Su, Y. Marzouk, A. Melloni, and L. Daniel, “Uncertainty quantification of silicon photonic devices with correlated and non-Gaussian random parameters,” *Optics Express*, vol. 23, no. 4, pp. 4242 – 4254, Feb 2015.

[17] X. Li, “Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 11, pp. 1661–1668, Nov 2011.

[18] T. Moselhy and L. Daniel, “Stochastic dominant singular vectors method for variation-aware extraction,” in *Proc. Design Auto. Conf.*, Jun. 2010, pp. 667–672.

[19] T. A. El-Moselhy and L. Daniel, “Variation-aware interconnect extraction using statistical moment preserving model order reduction,” in *Design, Automation and Test in Europe*, 2010, pp. 453–458.

[20] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[21] Z. Zhang, H. D. Nguyen, K. Turitsyn, and L. Daniel, “Probabilistic power flow computation via low-rank and sparse tensor recovery,” *arXiv preprint*, vol. arXiv:1508.02489, 2015.

[22] Z. Zhang, M. Kamon, and L. Daniel, “Continuation-based pull-in and lift-off simulation algorithms for microelectromechanical devices,” *J. Microelectromech. Syst.*, vol. 23, no. 5, pp. 1084–1093, Oct. 2014.