Stochastic Analog Circuit Behavior Modeling by Point Estimation Method

Fang Gong
University of California, Los Angeles
Electrical Engineering Department
Los Angeles, CA 90095, US
gongfang@ucla.edu

Hao Yu
Nanyang Technological University
Electrical and Electronic Engineering
haoyu@ntu.edu.sg

Lei He
University of California, Los Angeles
Electrical Engineering Department
Los Angeles, CA 90095, US
lhe@ee.ucla.edu

ABSTRACT

Stochastic device parameter variations have dramatically increased beyond the scale of 65nm and can significantly lead to large mismatch for analog circuits. To estimate unknown analog circuit behavior in performance space under the given stochastic variations in parameter space, many state-of-art approaches have been developed recently. However, either Gaussian distribution or response surface model (RSM) with analytical formulae has to be assumed when connecting performance space and parameter space. A novel point-estimation based approach has been proposed in this paper to capture arbitrary stochastic distributions for analog circuit behaviors in performance space. First, to evaluate high-order moments of circuit behavior in an accurate fashion, the point-estimation method has been applied with only a few number of simulations. Then, probability density function (PDF) of circuit behavior can be efficiently extracted by the obtained high-order moments. This method is further extended for multiple parameters under linear complexity. Extensive numerical experiments on a number of different circuits have demonstrated that the proposed point-estimation method can provide up to 181X runtime speedup with the same accuracy, when compared with Monte Carlo method. Moreover, it can further achieve up to 15X speedup over the RSM-based method such as APEX with the similar accuracy.

ACM Classification Keywords: B.7.2: - Integrated Circuits-Design Aids
General Terms: Algorithms, Performance
Authors Keywords: Behavior Modeling, Point Estimation, Circuit simulation.

1. INTRODUCTION

As semiconductor industry enters into nano-technology node, large process variations become inevitable and hence pose a serious threat to both analog circuit design and manufacturing [1, 2, 3, 4]. Device variables in parameter space, such as the effective channel length and threshold voltage of transistors, can deviate significantly from nominal values due to large uncertainties from chemical mechanical polishing (CMP), etching, lithography and etc. Under such circumstance, circuit behaviors in performance space can differ from the nominal case by a large margin, which may further lead to high loss of yield. With the process variation of device variables in parameter space, it is desirable to extract the unknown distribution of variable circuit behaviors in performance space. A robust circuit design and yield enhancement are especially important for analog circuits. One critical but missing link here is how to find an efficient yet accurate mapping between parameter space and performance space.

Note that the local random or stochastic variation is the most difficult one to be calculated, which is also called *mismatch* for the behavior modeling of analog circuits. In the past decade, many stochastic techniques had been proposed, such as Monte Carlo simulation, linear regression [1], stochastic orthogonal polynomials (SoPs) expansion [5, 3], response surface modeling based approaches [2, 4] and etc. The most general approach is to apply the Monte Carlo (MC) simulation, which samples all variable parameters and then calculate stochastic analog circuit behaviors by a large number of repeated simulations. As such, MC is too time-consuming to be afforded for the post-layout verifications beyond 65nm. To relieve the computational complexity, linear regression method [1] has been deployed to approximate the circuit behavior in performance space by a linear function of a number of normally distributed process variables in parameter space, or Gaussian distribution. This approach is efficient because of using analytical formula to obtain the circuit behavior. However, this approach cannot approximate non-normal (non-Gaussian) distributions and might lead to the loss of accuracy. With the use of different stochastic orthogonal polynomials (SoPs), SoP based methods can model process variations with non-Normally distributed random variables. The unknown distribution of circuit behaviors in performance space can be estimated by solving SoP expansion coefficients [5, 3]. However, the SoP-based methods require knowing the type of the stochastic distribution of the circuit behavior. In practice, one known parameter distribution in parameter space usually becomes unknown in performance space after the mapping.

In order to capture unknown random distribution after

ISPD'11, March 27–30, 2011, Santa Barbara, California, USA.

the mapping, response-surface-method (RSM) based methods [6, 4, 2] have been developed. One most important work developed recently is asymptotic probability extraction (APEX) [2] with the use of asymptotic waveform evaluation [7]. This approach assumes a polynomial function of all process parameters and further applies moment matching to extract the random distribution of circuit behavior (e.g. delay, gain, etc.). Nevertheless, the limitations of RSM-based approaches can be summarized in two-fold. First, the circuit behavior in performance space has become one strongly nonlinear function for random device variables. As such, the extraction by RSM has become computationally expensive. Second, it is prohibitive to evaluate high-order moments $E(f^k)$ with analytical formula of f, especially when the number of random variables and the moment order k increase. As such, the approaches based on RSM still cannot mitigate the super-linearly complexity while remaining accuracy for large-scale problems.

In this paper, a new mapping algorithm is developed to obtain the arbitrary circuit behavior in performance space from the arbitrary device variable in parameter space. High-order moments of circuit behavior f are first estimated by Point Estimation (PE) method to efficiently characterize the high-order moments $E(f^k)$ by weighted-sum of a few sampled simulations. Therefore, one can significantly improve the efficiency and accuracy of APEX [2] without the need to assume RSM inputs. As a result, the distribution of circuit behavior f in performance space can be efficiently obtained by its moments $E(f^k)$, calculated from the PE method in the parameter space. Moreover, a normalized PDF function is introduced so that to enhance the accuracy by eliminating the potential round-off error. In addition, this approach has been extended to consider the case with multiple parameters.

Extensive experiments on a number of different circuits are performed to demonstrate the validity and efficiency of our proposed algorithm. The contributions of this paper are further clarified as follows. First, although point estimation method has widely been applied for reliability analysis [8, 9], it can only estimate at most four moments (e.g. the mean, the variance, the skewness, the kurtosis) with empirical analytical formulae, and hence remains unclear how to estimate moments with higher order. In this paper, a modified point estimation method is developed to approximate higher order moments in a systematic manner. Moreover, unlike the observed super-linearly complexity in RSM based methods, our proposed method can be extended to deal with multiple parameters with linear complexity, which is significant for large-scale analog circuits.

The rest of this paper is organized as follows. In Section 2, we first review the mathematical formulation of the PDF estimation and the moments used in response-surface-model (RSM) methods. In Section 3, we introduce the point estimation (PE) method and further propose a new high-order moments evaluation via PE. We also discuss one normalized PDF technique in Section 4 to reduce error and further present experimental results in Section 5. This paper concludes in Section 6.

2. BACKGROUND

2.1 Mathematical Formulation

We consider circuit behavior f with multiple random variables of process variations $(x_1, x_2, \cdots, x_n)$, which can be expressed as $f(x_1, x_2, \cdots, x_n)$. As such, *parameter space* can be defined as the space $\mathbb{R}^n$ bounded by the min and max of all random variables, and *performance space* $\mathbb{R}$ consists of all possible behavior merits.

As a result of uncertainties in process technology, random variables can deviate from their nominal values and lead to variational circuit behavior. Our purpose is to extract unknown distribution (e.g. PDF/CDF functions) of circuit behavior by mapping the variable parameter distributions in parameter space into performance space.

To this end, the probabilistic moments in both spaces should be defined according to probability theory [10, 11]:

$$\begin{aligned} m_f^p &= E(f^p) = \int_{-\infty}^{+\infty} (f^p \cdot pdf(f))df \\ m_x^p &= E(x^p) = \int_{-\infty}^{+\infty} (x^p \cdot pdf(x))dx \end{aligned} \tag{1}$$

where m_f^p is the p-th moment of circuit behavior f in performance space, and m_x^p is the p-th moment of random variable x in parameter space.

LEMMA 1. *Suppose $pdf(f)$ is continuous in performance space. Then $pdf(f)$ can be determined uniquely by high order moments $E(f^k)$ $(k = 1, 2, \cdots, m)$.*

PROOF. Let $\Phi(\omega)$ is the Fourier transform [12] of $pdf(f)$ and can be written as:

$$\begin{aligned} \Phi(\omega) &= \int_{-\infty}^{+\infty} \left(pdf(f) \cdot e^{-j\omega f}\right) df \qquad (2) \\ &= \int_{-\infty}^{+\infty} \left(pdf(f) \cdot \sum_{p=0}^{+\infty} \frac{(-j\omega f)^p}{p!}\right) df \\ &= \sum_{p=0}^{+\infty} \frac{(-j\omega)^p}{p!} \cdot \int_{-\infty}^{+\infty} (f^p \cdot pdf(f))df. \\ &= \sum_{p=0}^{+\infty} \frac{(-j\omega)^p}{p!} \cdot m_f^p. \end{aligned}$$

As such, $\Phi(\omega)$ can be expanded with high order moments m_f^p, and $pdf(f)$ can be extracted from Inversion Fourier transform of $\Phi(\omega)$. In other words, there is a one-to-one correspondence between high order moments m_f^p and $pdf(f)$. □

Notice that m_x^p in (1) can be computed accurately in parameter space with known $pdf(x)$. Therefore, it is the key problem to find an efficient mapping between m_x^p and m_f^p in order to extract $pdf(f)$ in performance space.

2.2 Preliminary of PDF Calculation

The techniques to extract $pdf(f)$ with moments $E(f^k)$ have been proposed in [2, 7], which will be reviewed in what follows. First, *time moments* for f can be defined as:

$$\widehat{m}_f^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} f^k \cdot pdf(f)df. \tag{3}$$

It is clear that $\widehat{m}_f^k$ is defined in performance space but different from m_f^k in (1) due to a scaling factor $(-1)^k/k!$.

On the other hand, consider a linear time-invariant (LTI) system H, and its time moments can also be defined as[7]:

$$\widehat{m}_t^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} t^k \cdot h(t)dt. \tag{4}$$

where t is the time variable and $h(t)$ is impulse response of LTI system H. So, impulse response $h(t)$ can be an optimal approximation to $pdf(f)$ if we treat t as circuit behavior f and make $\widehat{m}_t^k$ equal to $\widehat{m}_f^k$. Furthermore, time moments in (4) can be expressed as [7]:

$$\widehat{m}_t^k = -\sum_{r=1}^{M} \frac{a_r}{b_r^{k+1}}. \tag{5}$$

Where a_r and b_r $(r = 1, \cdots, M)$ are the residues and poles of this LTI system, respectively. As such, the impulse response of the LTI system can be simplified as:

$$h(t) = \begin{cases} \sum_{r=1}^{M} a_r \cdot e^{b_r^{k+1} \cdot t} & (t \geq 0) \\ 0 & (t < 0) \end{cases} \tag{6}$$

In general, there are three steps to calculate $h(t)$ as an approximation to $pdf(f)$:

- Mapping m_x^k in parameter space into performance space as $\widehat{m}_f^k$ in (3).
- Make $\widehat{m}_f^k$ equal to $\widehat{m}_t^k$ and solve nonlinear equation system in (5) for residues a_r and poles b_r.
- Compute impulse response $h(t)$ in (6) with residues a_r and poles b_r.

Clearly, one needs to find an efficient mapping between parameter space and performance space to obtain the stochastic circuit behavior. Within this mapping, the most challenging step is the evaluation of high-order moments. Although the RSM based methods, such as APEX, assume that one nonlinear function can be found to approximate the mapping, they might become unaffordable to calculate the high-order moments for large-scale stochastic problems. To this end, we have developed a modified point estimation (PE) method to perform the mapping for the calculation of high-order moments.

3. HIGH ORDER MOMENTS ESTIMATION

In this section, we discuss how to evaluate high order moments of circuit behavior m_f^k by mapping parameter moments m_x^k from parameter space into performance space via Point Estimation (PE) method.

3.1 Moments via Point Estimation

For illustration purpose, we consider circuit behavior $f(x)$ with single variable parameter x. Usually it is impractical to compute m_f^k as (1) because $pdf(f)$ is unknown. The other straightforward way is to use Taylor expansion, which involves high order derivatives. But there is no way to guarantee the existence of high order derivatives of $f(x)$.

As such, we propose to leverage the Point Estimation method to compute high order moments[8, 9], which approximates m_f^k with a weighted sum of sampling values of $f(x)$. Assume $\tilde{x}_j$ $(j = 1, \cdots, p)$ are estimating points of random variable, and P_j are corresponding weights. In this way, the k-th order moment of $f(x)$ can be approximated as:

$$m_f^k = \int_{-\infty}^{+\infty} f^k \cdot pdf(f)df \approx \sum_{j=1}^{p} P_j \cdot f(\tilde{x}_j)^k. \tag{7}$$

However, [8, 9] only provide empirical analytical formulae of $\tilde{x}_j$ and P_j for first four moments. Therefore, it is significant but remains unknown how to determine $\tilde{x}_j$ and P_j for higher order moments systematically.

3.2 Estimating Points and Weights

To this end, we start with (7) in performance space, but it is impossible to compute $\tilde{x}_j$ and P_j since both sides are unknown. Thus, we need to reformulate the problem in parameter space, where random variable x and its distribution (e.g. PDF function $pdf(x)$) are known beforehand.

According to classic probability theory[10, 11], we have following theorem:

Theorem 1. *Let x and $f(x)$ are both continuous random variables, and their PDFs are $pdf(x)$ and $pdf(f)$, respectively. Suppose $\int f^k(x) \cdot pdf(x)dx$ exists. Then*

$$E(f^k(x)) = \int f^k(x) \cdot pdf(f)df = \int f^k(x) \cdot pdf(x)dx$$

As such, the moments of circuit behavior $f(x)$ in performance space can be calculated in parameter space. For example, the k-th order moment of $f(x)$ in equation (7) becomes:

$$m_f^k = \int f(x)^k \cdot pdf(x)dx \approx \sum_{j=1}^{m} P_j \cdot f(\tilde{x}_j)^k. \tag{8}$$

On the other hand, the k-th order moments of random variable x can be written as:

$$m_x^k = \int x^k \cdot pdf(x)dx \approx \sum_{j=1}^{m} P_j^{'} \cdot (\tilde{x}_j^{'})^k. \tag{9}$$

It is obvious that estimating points $\tilde{x}_j$ and corresponding weights P_j in (8) are the *same* as $\tilde{x}_j^{'}$ and $P_j^{'}$ in (9) because they are all defined in parameter space. Therefore, we can calculate $\tilde{x}_j^{'}$ and $P_j^{'}$ from equation (9) with m_x^k obtained from (1), and then estimate m_f^k using (8).

Now, the problem is how to solve for $\tilde{x}_j^{'}$ and $P_j^{'}$ systematically. Since there are total $2m$ unknowns in (9), we need to build $2m$ equations using first $2m$ moments of random variable, which can be rewritten as:

$$\begin{aligned}
&\sum_{j=1}^{m} P_j^{'} = 1 = m_x^0 \\
&\sum_{j=1}^{m} P_j^{'} \cdot \tilde{x}_j^{'} = E(x) = m_x^1 \\
&\sum_{j=1}^{m} P_j^{'} \cdot (\tilde{x}_j^{'})^2 = E(x^2) = m_x^2 \\
&\cdots \\
&\sum_{j=1}^{m} P_j^{'} \cdot (\tilde{x}_j^{'})^{2m-1} = E(x^{2m-1}) = m_x^{2m-1}
\end{aligned} \tag{10}$$

Note that the right-hand-side of above nonlinear system are first $2m$ moments of x in the behavior domain and can be calculated exactly with known $pdf(x)$ and definition in (1).

This nonlinear system (10) can be solved using algorithm proposed in [7]. In what follows, we briefly describe this algorithm.

Assume residues $a_j = P_j^{'}$ and poles $b_j = 1/\tilde{x}_j^{'}$, the equations (10) can be reformulated as:

$$\begin{bmatrix} a_1 + a_2 + \cdots a_m \\ \frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots \frac{a_m}{b_m} \\ \frac{a_1}{b_1^2} + \frac{a_2}{b_2^2} + \cdots \frac{a_m}{b_m^2} \\ \vdots \\ \frac{a_1}{b_1^{2m-1}} + \frac{a_2}{b_2^{2m-1}} + \cdots \frac{a_m}{b_m^{2m-1}} \end{bmatrix} = \begin{bmatrix} m_x^0 \\ m_x^1 \\ m_x^2 \\ \vdots \\ m_x^{2m-1} \end{bmatrix} \tag{11}$$

The system matrix of (11) is the well-known Vandermonde matrix and can be divided into two parts:

$$M \cdot v = rhs_{low}; \; M \cdot \Lambda^{-q} \cdot v = rhs_{upper}$$

where rhs_{low} consists of the low order moments (k = 0,1, $\cdots$, $m-1$), and rhs_{upper} contains the high order moments ($k = m, m+1, \cdots, 2m-1$). Λ^{-1} is a diagonal matrix of $\{1/b_j\}$ ($j = 1, \cdots, m$). And M matrix ($m \times m$) can be expressed as:

$$M = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ b_1^{-1} & b_2^{-1} & \cdots & b_{m-1}^{-1} \\ b_1^{-2} & b_2^{-2} & \cdots & b_{m-1}^{-2} \\ \vdots & \vdots & \vdots & \vdots \\ b_1^{-(m-1)} & b_2^{-(m-1)} & \cdots & b_{m-1}^{-(m-1)} \end{bmatrix}$$

Therefore, the linear system $M \cdot v = rhs_{low}$ can be solved as $v = M^{-1} \cdot rhs_{low}$, and $rhs_{upper} = M \cdot \Lambda^{-q} \cdot M^{-1} \cdot rhs_{low}$. Since M is also a Vandermonde matrix that is the modal matrix for a system matrix in companion form, rhs_{upper} can be simplified as $rhs_{upper} = \hat{M}^{-q} \cdot rhs_{low}$, where $\hat{M}^{-1}$ is

$$\hat{M}^{-1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_0 & s_1 & s_2 & \cdots & s_{m-1} \end{bmatrix}. \tag{12}$$

In this way, the eigenvalues of $\hat{M}^{-1}$ are $\{1/b_j\}$ in (11). To calculate $\hat{M}^{-1}$, we need to compute $\{s_t\}$ ($t = 0, \cdots, m-1$) with following equation system:

$$\begin{bmatrix} m_x^0 & m_x^1 & \cdots & m_x^{m-1} \\ m_x^1 & m_x^2 & \cdots & m_x^m \\ \vdots & \vdots & \vdots & \vdots \\ m_x^{m-1} & m_x^m & \cdots & m_x^{2m-2} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{m-1} \end{bmatrix} = \begin{bmatrix} m_x^m \\ m_x^{m+1} \\ \vdots \\ m_x^{2m-1} \end{bmatrix}. \tag{13}$$

When $\{1/b_j\}$ are available, the $\{a_j\}$ can be calculated from Equation (11). Therefore, the weights $\{P_j^{'}\}$ and estimating points $\{\tilde{x}_j^{'}\}$ can be computed systematically and used to compute m_f^k in equation (8).

3.3 Extension to multiple parameters

It is usually necessary to handle multiple variable parameters simultaneously in real-world problems. Thus, we discuss how to extend aforementioned techniques to deal with multiple parameters.

Existing methods [8, 9] model $m_{f(x_1,x_2,\cdots,x_n)}^k$ as a linear combination of moments $m_{f(x_i)}^k$, where $f(x_i)$ is a function of single variable x_i with other variables set equal to mean values. However, [8, 9] can only estimate first four moments using explicit analytical formulae as linear combination.

Consider circuit behavior $f(x_1, x_2, \cdots, x_n)$ where x_1, x_2, $\cdots, x_n$ are independent random variables, it is desirable to estimate $m_{f(x_1,x_2,\cdots,x_n)}^k$ that is the k-th order moment of $f(x_1, x_2, \cdots, x_n)$.

To estimate higher order moments of $f(x_1, x_2, \cdots, x_n)$ systematically, we derive the equation for $m_{f(x_1,x_2,\cdots,x_n)}^k$ as:

$$m_{f(x_1,x_2,\cdots,x_n)}^k = \sum_{i=1}^{n} g_i m_{f(x_i)}^k. \tag{14}$$

Moreover, g_i can be calculated as follows and detailed derivation can be referred to technical report:

$$g_i = c \cdot \frac{\partial\,(f(x_i))}{\partial x_i}$$

For each parameter x_i, we have its estimating points $\tilde{x}_j$ and corresponding $f(\tilde{x}_j)$. Hence, it is possible to calculate $\partial\,(f(x_i))/\partial x_i$ with finite difference method numerically.

Besides, the constant c can be computed with:

$$m_{f(x_1,x_2,\cdots,x_n)}^0 = \sum_{i=1}^{n} g_i m_{f(x_i)}^0 = \sum_{i=1}^{n} g_i \tag{15}$$

$$= \sum_{i=1}^{n} c \cdot \frac{\partial\,(f(x_i))}{\partial x_i} = 1 \Rightarrow c = 1 \Bigg/ \sum_{i=1}^{n} \frac{\partial\,(f(x_i))}{\partial x_i}.$$

As such, high order moments of multi-variable function $m_{f(x_1,x_2,\cdots,x_n)}^k$ can be evaluated with moments of univariate function $m_{f(x_i)}^k$ efficiently. Extensive experiments can demonstrate its validity and efficiency.

3.4 Error Estimation

Theoretical maximum approximation error of point estimation method is analyzed in [13]. For the univariate case, the maximum approximation error to exact integral value in equation (1) can be governed by:

$$\left| \sum_{j=1}^{m} P_j \cdot f^k(\tilde{x}_j) - \int_{-\infty}^{+\infty} f^k(x) \cdot pdf(f) df \right| \le \alpha \cdot k^{1/m}. \tag{16}$$

where α is a constant, and k is the order of moments. m is the number of estimating points $\tilde{x}_j$ for order k. As such, it implies that more estimating points for each variable should be used to reduce the estimation error of higher order moments.

4. PDF CALCULATION WITH MOMENTS

4.1 PDF/CDF Estimation with Moments

With high order moments m_f^k available, the next step is to compute residues $\{a_r\}$ as well as poles $\{b_r\}$ in (5) and impulse response $h(t)$ [2, 7] in (6). To do so, we calculate $\widehat{m}_f^k$ in (3) and make them equal to $\widehat{m}_t^k$ in (5). The nonlinear equation system becomes:

$$
-\begin{bmatrix}
\frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots \frac{a_M}{b_M} \\
\frac{a_1}{b_1^2} + \frac{a_2}{b_2^2} + \cdots \frac{a_M}{b_M^2} \\
\frac{a_1}{b_1^3} + \frac{a_2}{b_2^3} + \cdots \frac{a_M}{b_M^3} \\
\vdots \\
\frac{a_1}{b_1^{2M}} + \frac{a_2}{b_2^{2M}} + \cdots \frac{a_M}{b_M^{2M}}
\end{bmatrix}
=
\begin{bmatrix}
\overset{\frown}{m}_f^{0} \\
\overset{\frown}{m}_f^{1} \\
\overset{\frown}{m}_f^{2} \\
\vdots \\
\overset{\frown}{m}_f^{M}
\end{bmatrix}. \qquad (17)
$$

which has a Vandermonde matrix similar to (11) and can be solved with the same technique in [7]. With poles $\{b_r\}$ and residues $\{a_r\}$ available, PDF function can be approximated with equation (6).

Notice that impulse response $h(t)$ is zero for $t < 0$, but the PDF in real-life problems can be nonzero for $f \leq 0$. In this case, PDF function can be shifted as [2] which can be demonstrated with our experiments.

4.2 Normalized PDF for Error Prevention

From (17), it is obvious that the accuracy of PDF approximation mainly depends on the accuracy of residues a_r, poles b_r and moments estimation. However, there are roundoff error within moment estimation and the PDF approximation, which can lead to instability issue. In order to prevent potential error, we propose to normalize PDF calculated from equation (6) to cancel out the potential roundoff error.

For illustration purpose, we take the roundoff error in moments as an example, and other roundoff error can be eliminated with the same way. Assume $\overset{\frown}{m}_f^{k}$ is the exact value of k-th time moment in equation (3), and $\tilde{m}_f^k$ is the estimated value of k-th time moment. Also, we assume $\tilde{m}_f^k = const \cdot \overset{\frown}{m}_f^{k}$ due to roundoff error, where *const* is a scaling constant.

As such, when we use the direct solution in section 2, the scaling constant in both system matrix and right-hand-side vector of (13) can be canceled out. Hence, s_t $(t = 0, \cdots, m-1)$ and thus eigenvalues of equation (12) (that is poles $\{1/b_j\}$) are both exact values.

Next, the nonlinear equation system (17) becomes:

$$
-\begin{bmatrix}
\frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots \frac{a_M}{b_M} \\
\frac{a_1}{b_1^2} + \frac{a_2}{b_2^2} + \cdots \frac{a_M}{b_M^2} \\
\frac{a_1}{b_1^3} + \frac{a_2}{b_2^3} + \cdots \frac{a_M}{b_M^3} \\
\vdots \\
\frac{a_1}{b_1^{2M}} + \frac{a_2}{b_2^{2M}} + \cdots \frac{a_M}{b_M^{2M}}
\end{bmatrix}
= const \cdot
\begin{bmatrix}
\overset{\frown}{m}_f^{0} \\
\overset{\frown}{m}_f^{1} \\
\overset{\frown}{m}_f^{2} \\
\vdots \\
\overset{\frown}{m}_f^{M}
\end{bmatrix} \qquad (18)
$$

Which leads to $\tilde{a}_j = const \cdot a_j$, where a_j are exact values of residues. Therefore, PDF of $f(x)$ is approximated with:

$$
pdf(f) = \sum_{r=1}^{M} \tilde{a}_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f} = \sum_{r=1}^{M} const \cdot a_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f} \qquad (19)
$$

In order to eliminate the scaling constant, we propose to normalize PDF of $f(x)$ as follows: First, we discretize $f(x)$ into discrete points $\{f_p(x)\}$ $(p = 1, \cdots, N)$. As such, PDF on p-th discrete point $\{f_p(x)\}$ can be expressed as:

$$
pdf(f_p) = \sum_{r=1}^{M} const \cdot a_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f_p} \qquad (20)
$$

To normalize it, we can divide it with the total value of PDF on all discrete points as:

$$
\begin{aligned}
pdf_{norm}(f_p) &= \frac{\sum_{r=1}^{M} const \cdot a_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f_p}}{\sum_{p=1}^{N} \sum_{r=1}^{M} const \cdot a_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f_p}} \qquad (21) \\
&= \frac{\sum_{r=1}^{M} a_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f_p}}{\sum_{p=1}^{N} \sum_{r=1}^{M} a_r \cdot e^{\overset{\frown}{b}_r^{k+1} \cdot f_p}}
\end{aligned}
$$

In this way, the scaling constant can be eliminated from the approximation of PDF and thus normalization improves the numerical stability of proposed algorithm.

4.3 Error Estimation

Since the Fourier transform in equation (3) is unique, it is equivalent to evaluate the error of $\Phi(\omega)$ in order to investigate the accuracy of PDF approximation with qth order moments. It is ideally to compare the difference between Fourier transform of estimated PDF and that of exact PDF. However, the exact PDF is usually not available. Instead, we use approximation with $n+1$ order moments as the exact value, and proceed to estimate the error.

$$
\begin{aligned}
Error &= \left| \frac{\Phi^{q+1}(\omega) - \Phi^{q}(\omega)}{\Phi^{q+1}(\omega)} \right| \qquad (22) \\
&= \left| \frac{\sum_{p=0}^{q+1} \frac{(-j\omega)^p}{p!} \cdot m_f^p - \sum_{p=0}^{q} \frac{(-j\omega)^p}{p!} \cdot m_f^p}{\sum_{p=0}^{q+1} \frac{(-j\omega)^p}{p!} \cdot m_f^p} \right| \\
&= \left| \frac{(-j\omega)^{q+1}}{(q+1)!} \cdot \left(\sum_{p=0}^{q+1} \frac{(-j\omega)^p}{p!} \cdot \frac{m_f^p}{m_f^{q+1}} \right)^{-1} \right|.
\end{aligned}
$$

When $|m_f^p| \geq |m_f^{q+1}|$ ($p \leq q+1$), above error estimation can become:

$$
Error \leq \left| \frac{(-j\omega)^{q+1}}{(q+1)!} \cdot \left(\sum_{p=0}^{q+1} \frac{(-j\omega)^p}{p!} \right)^{-1} \right|. \qquad (23)
$$

The same error estimation can be obtained for $|m_f^p| \leq |m_f^{q+1}|$ by taking reciprocal of circuit behavior to shift the behavioral distribution.

As such, the error estimation can be used to measure the accuracy of the approximation with first q-th order moments. When the approximation order increases, the error estimation should move to higher order as required.

4.4 Complexity Analysis

The Monte Carlo simulation requires to generate massive samples to cover the entire parameter space evenly, so number of simulations is p^n where p is the number of samplings for every single variable and n is the total number of random variables.

As for RSM based methods, we take APEX as an example where RSM formulation is the most time-consuming part. For example, when RSM uses n random variables and k order polynomial function, it has total $C_n^1 + C_n^2 + \cdots + C_n^k$ terms and thus APEX requires $C_n^1 + C_n^2 + \cdots + C_n^k$ simulation samples. In other words, the complexity of APEX is $O(n^k)$.

When proposed algorithm handles total n variables and needs m estimating points for each variable, the total complexity is $(m-1)*n+1$ (usually $m \ll n$) or $O(n)$. $m-1$ denotes that estimating points of each variable includes the nominal point and nominal circuit simulation can be shared by all variables. Therefore, it has linear complexity.

5. EXPERIMENTAL RESULTS

We have implemented proposed algorithm in the MATLAB environment, and all experiments are carried out on a Linux server with a 2.4GHz Xeon processor and 4GB memory. We use a six-transistor SRAM cell and a two-stage operational amplifier to compare the accuracy and efficiency of proposed algorithm with APEX [2] and Monte Carlo simulation. As an illustration, we consider the threshold voltages of MOSFETs as independent random variables subject to process variations, but our algorithm can also handle other variation sources.

5.1 SRAM Cell

We first consider a typical design of 6T SRAM cell in Fig.(1) and investigate the access time failure of the SRAM cell during reading operation, which is determined by the voltage difference between BL_B and BL.

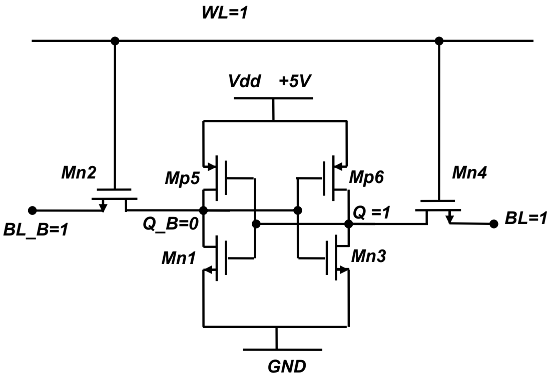

Figure 1: Schematic of SRAM 6-T Cell

Initially, both BL_B and BL are pre-charged to Vdd, while Q_B stores zero and Q stores one. When reading the SRAM cell, BL_B starts to discharge from Vdd and produces a voltage difference ΔV between it and BL. The time it takes BL_B to produce a large enough voltage difference is called access time. Since process variations are inevitable, the access time of manufactured SRAM cells can deviate from nominal value. When access time is larger than acceptable maximum value T_{max}, this leads to an access time failure.

In our experiment, we consider threshold voltages of all MOSFETs as independent variables which are normally distributed with 30% perturbation from nominal values. As such, there are perturbations to the nominal discharge trajectory on BL_B as shown in Fig.(2). Therefore, we can investigate the access time failure by capturing the random distribution of voltage on BL_B at T_{max} time-step: when voltage of BL_B at T_{max} is larger than its nominal value, the access time failure happens.

Note that both PEM and APEX can not capture high precision in the tail region of CDF/PDF, which is required to deal with rare event in an SRAM. Therefore, we focus on reducing average error of the performance distribution in the entire range. We start from univariate case to validate proposed algorithm and then extend it to multi-variable case. We have implemented three other methods for comparison:

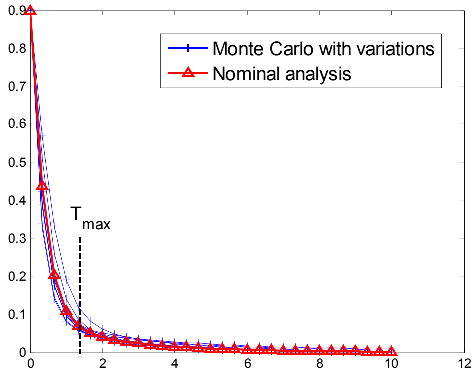

Figure 2: BL_B discharge behavior with V_{th} variations

- **Monte Carlo simulation (MC):** This is direct Monte Carlo simulation.
- **APEX:** Implementation of asymptotic probability extraction algorithm proposed in [2].
- **Point Estimation Method (PEM):** Proposed algorithm that leverages the point estimation method.

5.1.1 Univariate Case

First, we consider one random variable (e.g. threshold voltage variation on Mn_2) to compare the accuracy of APEX and PEM against MC. The random distributions (PDF function) from these methods are plotted in Fig.(3). Note that the histogram from Monte Carlo simulation has been normalized to eliminate the effect of total number of samplings.

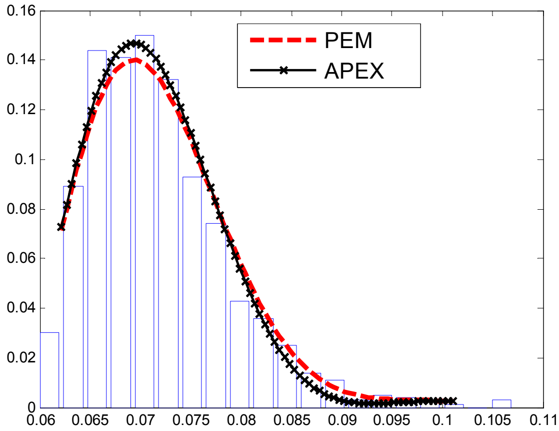

Figure 3: random distributions of BL_B voltage at T_{max}(Monte Carlo result has been normalized)

When compared with Monte Carlo results, PEM can provide better accuracy than APEX, especially in the peak and right tail regions. However, APEX has better efficiency even if the same order of moments are used: APEX needs only 8.59 second with quadratic function in response surface model, while PEM requires 17.71 seconds with seven estimating points for one variable.

It is because APEX use analytical formula which is suitable for low dimensions. It should be noticed that the number of required simulation samples in APEX will increases exponentially when more variables or a strongly nonlinear RSM required.

5.1.2 Multiple Variable Parameters

Next, we consider all threshold voltages of six transistors are independent random variables, which are normally distributed with 30% perturbation from nominal values. Similarly, we attempt to compare all three methods, but it is prohibitive to implement APEX as response surface model becomes very complicated, especially when high order response surface model is required.

Instead of original APEX, we calculate high order moments numerically using results from Monte Carlo simulations and extract PDF with technique in APEX, which is denoted as MMC+APEX. The random distributions from all methods are compared in Fig.(4).

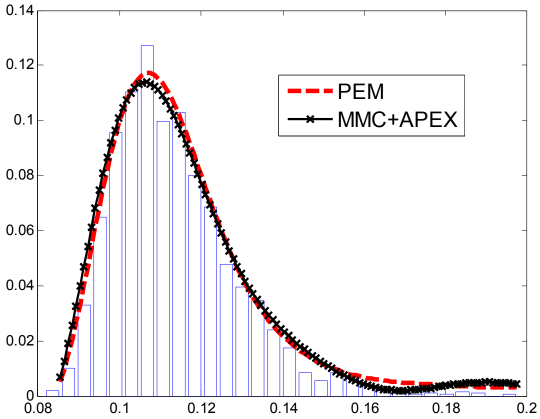

Figure 4: random distributions of BL_B voltage at T_{max}(Monte Carlo result has been normalized)

It is obvious that PEM can capture the exact distribution of BL_B voltage at T_{max}, which fits well with Monte Carlo simulation and MMC+APEX method. This can shows that PEM can achieve very high accuracy in the multi-variable problems. Also, we compare only the runtime of MC and PEM in Table 1, since runtime of MMC+APEX is almost the same as MC. In this table, PEM uses 5 estimating points for each variable parameter, and can achieve the same accuracy with 119.2X speedup over Monte Carlo method.

Table 1: Runtime Comparison of three methods

Method	Time (second)	Speedup
Monte Carlo (3×10^3)	7644	1x
PEM (5 point)	64.12	119.2x

To compare the efficiency with APEX, we can consider a SRAM cell under commercial 65nm CMOS process where 10 independent variables are used to model random variation for each transistor [14]. As such, RSM using quadratic function has 1830 coefficients and thus APEX requires 1830 simulation samples as discussed in Section 4.4. However, PEM only needs 121 simulation samples when 3 estimating points are used for each independent variable, and achieves 15X speedup over APEX.

5.2 Operational Amplifier

We further consider a two-stage operational amplifier in Fig. (5) and a negative feedback circuit in Fig.(6). We use this example to show that proposed method can estimate random distributions where circuit behavior is negative.

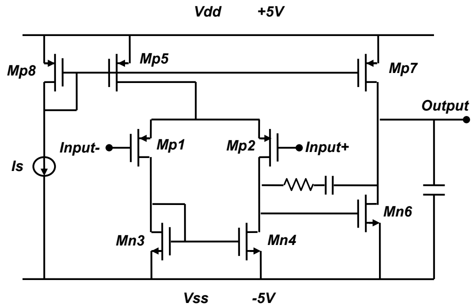

Figure 5: Schematic of Operational Amplifier

Similar to SRAM cell example, we consider threshold voltages of all MOSFETs as independent variables which are normally distributed with 30% perturbation from nominal values. Note that the threshold voltages of input transistor pair (Mp_1, Mp_2) should be kept the same to ensure the convergence of nonlinear system solver in circuit simulators.

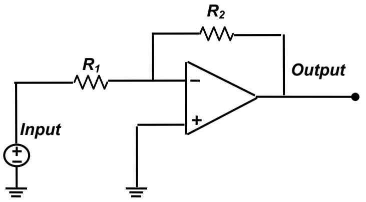

Figure 6: Schematic of a unity gain feedback circuit

There are a number of op am specifications in time-domain and frequency-domain, such as slew rate, settling time, phase margin, input offset voltage and etc. In our experiment, we investigate the input offset voltage variation due to threshold voltage variations of MOSFETs.

In this experiment, we implement following methods for comparison purpose:

- **Monte Carlo simulation (MC):** This is direct Monte Carlo simulation.
- **Moments from Monte Carlo (MMC+APEX):** Estimate high order moments of input offset voltage from Monte Carlo simulation, and extract the PDF using techniques in [2, 7].
- **Point Estimation Method (PEM):** Proposed algorithm that leverages the point estimation method.

First, we validate the accuracy of PEM by comparing the random distributions of input offset voltage from different methods in Fig.(7). PEM employs 5 estimating points for each variable and achieve the correct PDF function by shifting distribution of circuit behavior into positive region and moving it back.

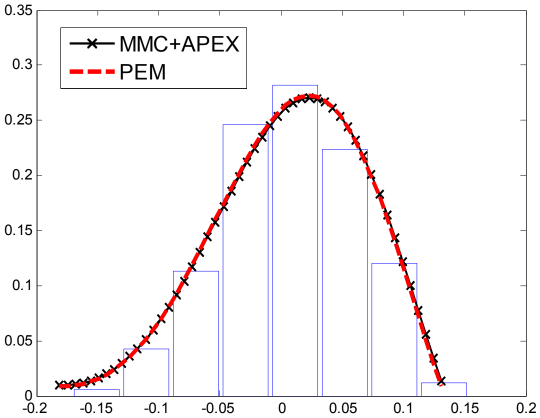

Figure 7: random distributions of input offset voltage (Monte Carlo result has been normalized.)

Since MMC+APEX method has the exact moment values from Monte Carlo simulations, it can provide very high accuracy. Also, PEM can offer the same accuracy with MMC+APEX method, which implies that PEM can achieve high accuracy of high order moments. On the other hand, we further validate accuracy of PDF function from PEM against the histogram from Monte Carlo simulation in Fig.(7), which fit with each other very well.

Moreover, we compare the runtime of Monte Carlo method and PEM in Table (2), and MMC+APEX is omitted because it has almost the same runtime as Monte Carlo method. Besides, we list PEM method with different number of estimating points for each variable to demonstrate it has linear scalability with the estimating points at the same time.

Table 2: Runtime Comparison between Monte Carlo simulation and PEM

Method	Time (second)	Speedup
Monte Carlo (3×10^3)	27765 (7.71 hours)	1x
PEM (3 point)	153.1 (0.04 hours)	181.5x
PEM (9 point)	525.84 (0.15 hours)	52.8x

From Table (2), PEM can provide up to hundreds times speedup over MC method. Moreover, the computational cost increases linearly with the number of estimating points for each variable.

6. CONCLUSION

In this paper, we have proposed one efficient point estimation (PE) based algorithm to extract the stochastic circuit behavior in performance space from parameter space. Our approach can perform an efficient evaluation of high-order moments of circuit behavior, and thus circumvent the use of response surface model (RSM) methods. This can dramatically reduce the computational cost seen in APEX. Moreover, the proposed method can be extended to deal with multiple parameters under linear complexity. Experiments on a few different circuits have shown that the proposed method can provide up to 181X more runtime speedup with the same accuracy when compared with the Monte Carlo method. Also, it can achieve up to 15X speedup over the RSM based method such as APEX with the similar accuracy.

7. REFERENCES

[1] S. Nassif, "Modeling and analysis of manufacturing variations," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 223–228, 2001.

[2] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," in *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, pp. 2–9, 2004.

[3] S. Vrudhula, J. M. Wang, and P. Ghanta, "Hermite polynomial based interconnect analysis in the presence of process variations," *IEEE Tran. on Computer-aided-design (TCAD)*, pp. 2001–2011, 2006.

[4] X. Li, Y. Zhan, and L. Pileggi, "Quadratic statistical MAX approximation for parametric yield estimation of analog/RF integrated circuits," *IEEE Tran. on Computer-aided-design (TCAD)*, vol. 27, pp. 831–843, 2008.

[5] D. Xiu and G. E. Karniadakis, "The wiener-askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, pp. 619–644, 2002.

[6] G. E. P. Box and N. R. Draper, "Empirical model building and response surfaces," *Wiley series In Probability and Mathematical Statistics, New York: John Wiley and Sons*, 1987.

[7] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Tran. on Computer-aided-design (TCAD)*, vol. 9, no. 4, pp. 352–366, 1990.

[8] E. Rosenblueth, "Point estimation for probability moments," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 72, no. 10, pp. 3812–3814, 1975.

[9] Y.-G. Zhao and T. Ono, "New point estimation for probability moments," *Journal of Engineering Mechanics*, vol. 126, no. 4, pp. 433–436, 2000.

[10] A. Papoulis and S. Pillai, "Probability, random variables and stochastic processes," *McGraw-Hill*, 2001.

[11] M. H. DeGroot and M. J. Schervish, "Probability and statistics," *Addison Wesley*, 2011.

[12] A. V. Oppenheim, A. S. Willsky, and S. Hamid, "Signals and systems," *Prentice Hall*, 1996.

[13] S. Haber, "Numerical evaluation of multiple integrals," *SIAM Review*, vol. 12, pp. 481–525, 1970.

[14] X. Li and H. Liu, "Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, pp. 38–43, 2008.