University of California

Los Angeles

A Moment Matching Based Fitting Algorithm for High Sigma Distribution Modeling

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Electrical Engineering

by

Rahul Krishnan

2015

Abstract of the Thesis

A Moment Matching Based Fitting Algorithm for High Sigma Distribution Modeling

by

Rahul Krishnan

Master of Science in Electrical Engineering

University of California, Los Angeles, 2015

Professor Lei He, Chair

The impact of process variations continue to grow as transistor feature size shrinks. Such variations in transistor parameters lead to variations and unpredictability in circuit output, and may ultimately cause them to violate specifications leading to circuit failure. In fact, timely failures in critical circuits may lead to catastrophic failures in the entire chip. As such, statistical modeling of circuit behavior is becoming increasingly important. However, existing statistical circuit simulation approaches fail to accurately and efficiently analyze the high sigma behavior of probabilistic circuit output. To this end, we propose PDM (Piecewise Distribution Model) - a piecewise distribution fitting approach via moment matching using maximum entropy to model the high sigma behavior of analog/mixed-signal (AMS) circuit probability distributions. PDM is independent of the number of input dimensions and matches region specific probabilistic moments which allows for significantly greater accuracy compared to other moment matching approaches. PDM also utilizes Spearman's rank correlation coefficient to select the optimal approximation for the tail of the distribution. Experiments on a known mathematical distribution and various circuits obtain accurate results up to 4.8 sigma with more than 2 orders of speedup relative to Monte Carlo.

The thesis of Rahul Krishnan is approved.

Sudhakar Pamarti

Lara Dolecek

Lei He, Committee Chair

University of California, Los Angeles

2015

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Lei He, for his guidance and advice as I worked on this project. I would like to thank my colleague Wei Wu of UCLA and Srinivas Bodapati of Intel Corp. whose immense help and insight made my work possible. Finally, I would like to thank my parents and friends for their support and love.

CHAPTER 1

Introduction

As transistor feature size continues to shrink, the impact of process variations on circuit behavior grows and cannot be neglected [1, 2, 3]. Under these variations, circuit behavior is transformed from a deterministic value to a random variable with an unknown distribution. These variations can cause significant circuit performance degradation that may violate constraints and fail. As such, circuit reliability has become an area of growing concern. For critical circuits that are repeated millions of times, a single failure may lead to catastrophic results. Consequently, such "rare event" failures must be accurately and efficiently modeled to maximize the effective yield of a circuit.

As society and industry move towards more energy efficient chips, minimizing power consumption becomes increasingly important. In such designs, low supply voltages (VDD) are often used to reduce power. However, while VDD is explicitly reduced the gate over drive ($V_{gs} - V_{th}$) is implicitly reduced [4]. In the presence of large V_{th} variations from the manufacturing process, transistors may enter the subthreshold operation region causing a strongly non-linear circuit behavior. This non-linear behavior translates to circuit behavior distributions becoming strongly non-Gaussian (see Figure 4.4). Consequently, when modeling this behavior for yield analysis, it is necessary to consider the inherent non-linearity that arises due to the aforementioned reasons.

Although there are many methods that attempt to model overall circuit behavior [1, 3, 5], there are very few methods that efficiently model the high sigma behavior of strongly non-Gaussian distributions. One brute force method is Monte Carlo (MC), which is considered to be the gold standard approach; it involves repeated sampling and simulation to extract an approximate distribution of circuit behavior [6]. Although Monte Carlo is highly accurate, it is infeasible for yield analysis because it requires millions of samples/simulations making it runtime prohibitive. Moreover, if any design changes are introduced in the circuit we must repeat these simulations another million or more times.

In order to improve the efficiency of yield analysis techniques, classifier based approaches such as Importance Sampling [7, 8, 9] and Statistical Blockade [10] were proposed to obtain high accuracy with a minimal number of samples. Importance Sampling biases the input sample distribution to draw more “important” samples which are then reweighed and used to calculate a final probability. However, Importance Sampling methods have two drawbacks. First, they are unable to handle circuits with a large number of variables due to the “curse of dimensionality” which causes the reweighing process to become degenerate and unbounded [11, 12]. Second, they do not estimate the overall PDF of circuit behavior and thus require repetitive sampling to estimate different critical points, leading to potential inefficiency. Statistical Blockade [10] attempts to build a linear classifier to screen out/block samples that are likely to cause failure and evaluate these “likely to fail” samples to calculate a failure probability. However, the classifier does not account for the non-linearity between process variables and circuit outputs, or the multiplicity of input failure regions, leading to large errors.

In order to combat the dimensionality issue of the above methods, a moment matching technique based on Maximum Entropy [13], referred to as MAXENT, was proposed. The method is novel because it uses circuit output behavior (e.g. delay) as its only input and therefore performs moment matching solely in the output domain. Consequently, the method is constant in dimensionality and thus does not fall to the dimensionality issues in Importance Sampling and classifier methods outlined above. However, MAXENT uses only one set of moments that are accurate in the low sigma region. Obtaining moments that are accurate in the high sigma region requires both a large number of samples to obtain accurate moments and knowledge of which moments reflect behavior in the tail of the distribution, which is often unknown [14]. Consequently, the distribution that it uses is formulated on a global optimization framework that attempts to minimize overall error and is therefore unable to capture the high sigma behavior in non-Gaussian distributions.

To address both the issue of high-dimensionality *and* non-Gaussian distributions while maintaining high accuracy and efficiency, we propose a piecewise distribution fitting algorithm, known as PDM (Piecewise Distribution Model), that uses moment matching via maximum entropy to build two separate, region based distributions of circuit behavior. The first distribution, Segment1, matches moments that are accurate only in the body/bulk of the distribution. The second distribution, Segment2, matches moments that are accurate only in the high sigma/tail region of the distribution and models the tail of circuit behavior. Both distributions are constructed using the maximum entropy moment matching technique but differ by using two *different sets* of moments. The moments in Segment1 are obtained by using circuit behavior sample moments calculated from the original input (process variation) distributions. The moments in Segment2 are obtained using circuit behavior sample moments calculated from input distributions that are *shifted* towards regions that are more likely to fail.

The optimal Segment1 distribution is selected using Spearman's rank correlation coefficient to analyze the monotonic behavior of the CDF. The Segment2 distribution is assumed to be an exponential distribution. Because this distribution is constructed from shifted moments, its probability must be re-weighed and is done so using conditional probability and a scaling factor that corrects for continuity between the Segment2 distribution and the true model of the tail distribution. PDM has a constant complexity in terms of input dimensions as it works solely in the output (circuit behavior) domain. Experiments on both a mathematically known distribution and circuits demonstrate the method is accurate up to 4.8 sigma for non-Gaussian distributions with more than 2 orders of speedup relative to Monte Carlo.

The rest of the paper is organized as follows. Section 2 presents background on general statistical modeling and a detailed derivation of the maximum entropy moment matching technique, along with results on the MAXENT method [13]. Section 3 presents the proposed piecewise fitting algorithm and highlights the difference between it and the general maximum entropy moment matching technique. Section 4 evaluates the performance of PDM on the mathematically known distribution, one digital circuit, and one analog circuit, where all distributions are non-Gaussian. Section 5 concludes this thesis and presents some topics for future work.

CHAPTER 2

Background

2.1 General Statistical Modeling

Figure 2.1 shows the typical flow for performing statistical simulations on a circuit. We begin by sampling the process variations which are known as the input variables and are part of the "input domain". These variables, such as the effective channel length L_{eff} or oxide thickness t_{ox}, belong to the input domain because they are part of the transistor values that are fed into a circuit simulator. The input variables are typically assumed to be normally distributed with a known mean μ and variance σ^2. These parameters of the input variables are obtained via the transistor model which is typically provided by the designer or fabrication plant.

Once the input variables are sampled, we feed the values to a circuit simulator such as HSPICE. The simulator then produces a single circuit response such as 50% delay or a voltage at a node. These circuit responses are commonly known as the circuit output and are part of the "output domain". These circuit outputs are often one dimensional and often need an extended period of time to obtain even a single measurement. Due to the variations in the input variables, we will often get two different circuit outputs for a combination of different variable values. Therefore, if this work flow of sampling and simulation is repeated, the majority of the input domain space will be covered and an overall estimate of all circuit responses with their estimated probabilities can be extracted.

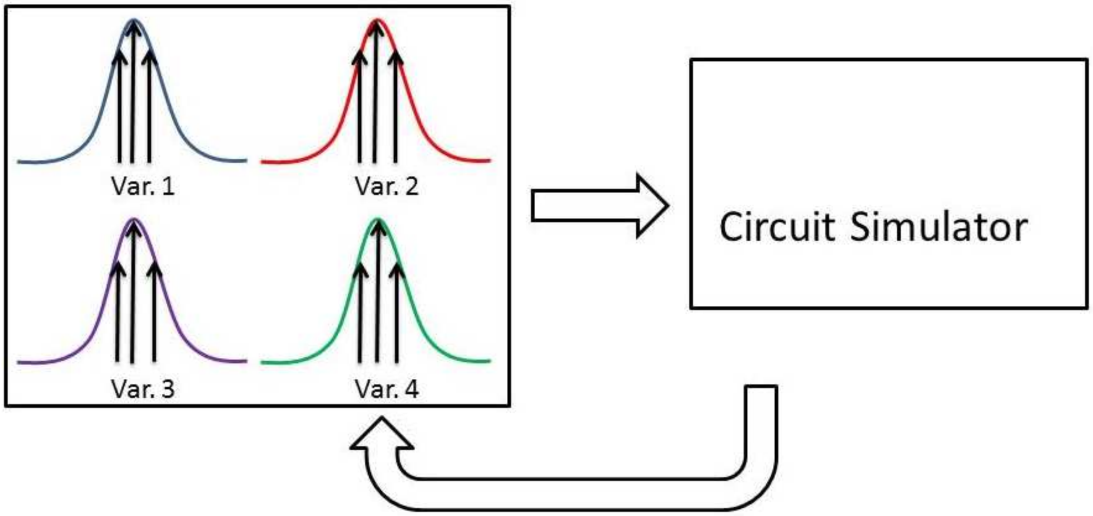

Figure 2.1: Flowchart for Circuit Simulation

The above flow of repeated sampling and simulation is straightforward Monte Carlo and is used to obtain an estimate of the distribution/probability density function (PDF) of circuit response [6]. Depending on the required accuracy of the response, the required number of samples will change. In some cases, a few hundred samples are required to estimate the probability of a circuit response around the mean (nominal) value of the PDF. However, we are often interested in the circuit responses that have very small probability because these "rare events" are typically the most harmful and destructive [9, 15]. Because we are interested in the distribution of circuit behavior, the random variable of interest is the circuit response.

One such method of quantifying the required number of samples for a target probability is simply taking the inverse. For example, consider a designer that is interested in the circuit response that will result in a failure rate of 16%. This means that we are interested in a circuit response that has approximately 16% probability in the tail of the PDF. This failure rate corresponds to approximately 1 failure every 6.25 samples, so a starting point would be drawing 7 samples, simulating each and selecting the largest value. However, the preceding case assumes that we will determinately see 1 failed sample every 6.25 which may not be true due to the large variations and unpredictability in the circuit. Consequently, in order to have a more confident estimate, we may require that we draw enough samples such that we have 5 failures, i.e. we would draw 32 samples. By using basic probability, we do not make any assumptions about the shape of the circuit PDF allowing for an unbiased estimate. Furthermore, to simplify the relationship between estimated probability (failure rate) and required number of samples, we utilize the Z score of a standard normal distribution which is typically referred to as the "sigma" value [14]. For example, instead of asking for the circuit response that gives a 16% failure rate, we would ask for the "1 sigma point" of the PDF. Note that the aforementioned probability and Z score methods work for both ends of the tail of a PDF.

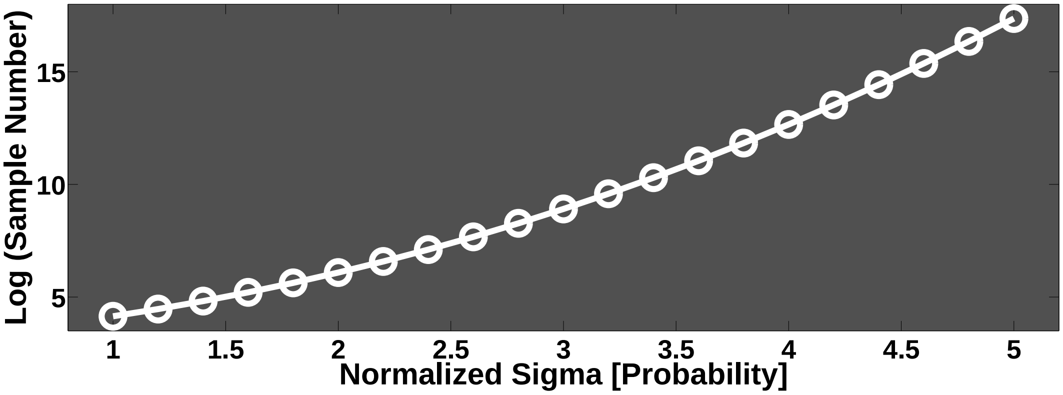

Figure 2.2: Required samples for various sigma (probability) points

In typical yield analysis, we require a failure rate of at least 0.003167% or approximately 4 sigma. Obtaining this probability would require one failure every $31,575$ samples, and using the estimate of 5 failure samples increases that required number to approximately $155,000$ samples. In many cases, this is the "largest" probability that is required and we typically need up to the 4.8 or 5 sigma point in the PDF, which corresponds to roughly $3.4E6$ samples for 1 failure. Furthermore, simulating this many samples is overly time consuming, making straightforward

Monte Carlo runtime prohibitive. Figure 2.2 shows the required number of samples (log scale) for a given sigma point (linear). We see that even in log scale, the required number of samples is non-linear. Alternative methods such as Quasi Monte Carlo may be utilized, however they still require a large number of samples and it can be shown that as the dimensionality of the sampling space increases (in this case, the input domain), the convergence rate of QMC and MC are similar [16]. Consequently, it is necessary to develop efficient algorithms that minimize the number of samples to accurately estimate a very small failure probability.

2.2 Derivation of Maximum Entropy Method

Entropy is a measure of uncertainty. When choosing a distribution, one should choose a distribution that maximizes the entropy [17]. By doing this, we can ensure that the distribution is uniquely determined to be maximally unbiased with regard to missing information, while still agreeing with what is known [17]. The entropy W of a distribution $p(x)$ is defined in (2.1). To choose the distribution with maximum entropy, we simply maximize this function with respect to some constraints. These constraints are typically moment constraints as shown in (2.2), as a probability distribution can be completely defined by its set of moments [18]. When applying the maximum entropy method to circuit simulation algorithms, we consider probabilistic sample moments of circuit response with moment order $i = 0, 1, ..., k$ where k assumes multiple values to generate multiple distributions, of which one is optimally selected.

$$W = \int -p(x) \log p(x) dx \tag{2.1}$$

$$\int x^i p(x) dx = \mu_i, \qquad i = 0, 1, ..., k. \tag{2.2}$$

To maximize (2.1) we first introduce Lagrange multipliers, resulting in the Lagrangian function

$$L = -\int (p(x) \log p(x))dx + \sum_{i=0}^{k} \lambda_i (\int x^i p(x)dx - \mu_i) \tag{2.3}$$

Next, we take partial derivatives of L with respect to $p(x)$ and λ to find the points where it reaches a maximum, as shown in (2.5) and (2.4).

$$\frac{\delta L}{\delta \lambda_i} = 0 \tag{2.4}$$

$$\frac{\delta L}{\delta p(x)} = 0 \tag{2.5}$$

Taking the derivative with respect to λ results in the original moment constraints from (2.2) and is redundant information. The derivative with respect to $p(x)$ yields (2.6)

$$\frac{\delta L}{\delta p(x)} = \int (\log p(x)dx) + 1 - \{\sum_{i=0}^{k} \lambda_i (\int (x^i dx))\} = 0 \tag{2.6}$$

We can further simplify this by absorbing the constant 1 into the λ_0 term and combining the finite sum with the integrand resulting in (2.7)

$$\int (\log p(x)dx) - \int (\sum_{i=0}^{k} \lambda_i x^i dx) = 0 \tag{2.7}$$

Note that the limits on both integrals are identical and are typically from ∞ to $-\infty$ for standard probability distributions because the distribution is assumed to be 0 outside of the support of random variable x. In the case of circuit simulation algorithms, this is also true, i.e. the circuit has maximum and minimum operating values and is zero outside these points. Consequently, because the above equation must hold in the general case of arbitrary limits, the integrand must be 0 and we can rearrange terms to solve for the unknown variable $p(x)$ as shown in (2.8).

$$p(x) = \exp\left(-\sum_{i=0}^{k} \lambda_i x^i\right) \tag{2.8}$$

However, the solution in (2.8) does not exist for values of $k \geq 2$ [19]. Consequently, [20] propose that we transform the constrained problem into an unconstrained problem by utilizing its dual. Utilizing duality allows us to recast the original problem of maximizing (2.3) into its dual form that we can minimize. This dual function can be obtained by plugging the results of (2.8) into the Lagrange function resulting in (2.9) and (2.10)

$$\Gamma = \ln Z + \sum_{i=1}^{k} \lambda_i \mu_i \tag{2.9}$$

$$Z = \exp(\lambda_0) = \int \exp\left(-\sum_{i=1}^{k} \lambda_i x^i\right) dx \tag{2.10}$$

Now this dual problem can be solved for any value of k. One approach is using an iterative method such as traditional Newton's method as shown in [19, 21, 13]. Here, Newton's method is used to solve for the Lagrangian multipliers $\lambda = [\lambda_0, \lambda_1, ..., \lambda_k]'$ for a corresponding set of moments $i = 0, 1, ..., k$. The standard Newton update equation for iteration m is shown in (2.11)

$$\lambda_{(m)} = \lambda_{(m)} - H^{-1}\frac{\delta\Gamma}{\delta\lambda} \tag{2.11}$$

Where the gradient (2.12) and Hessian (2.13) are defined as

$$\frac{\delta\Gamma}{\delta\lambda_i} = \mu_i - \frac{\int x^i \exp\left(-\sum\limits_{i=1}^{k} \lambda_i \mu_i\right) dx}{\int \exp\left(-\sum\limits_{i=1}^{k} \lambda_i \mu_i\right) dx} = \mu_i - \mu_i(\lambda) \tag{2.12}$$

$$H_{ij} = \frac{\delta^2 \Gamma}{\delta \lambda_i \delta \lambda_j} = \mu_{i+j}(\lambda) - \mu_i(\lambda)\mu_j(\lambda) \tag{2.13}$$

$$\mu_{i+j}(\lambda) = \frac{\int x^{i+j} \exp\left(-\sum_{i=1}^{k} \lambda_i \mu_i\right) dx}{\int \exp\left(-\sum_{i=1}^{k} \lambda_i \mu_i\right) dx} \tag{2.14}$$

Equation (2.13) indicates that the dual function Γ has a second derivative and that it is positive definite [22]. Consequently, the function (2.9) is everywhere convex which guarantees that if a stationary point exists it must be the *unique absolute minimum.* However, convexity does not ensure that a minimum does exist. Consequently, a necessary and sufficient condition that (2.9) has a unique absolute minimum at a finite value of λ is that the moment sequence $\{\mu_i, i = 0, 1, ..., k\}$ be completely monotonic [22]. We note that the derivation of such an existence condition is outside of the circuit simulation topic and we therefore refer to [22] for its derivation.

2.3 Application to Statistical Circuit Modeling

We begin by drawing a small number of samples from the input variables and feeding them to a circuit simulator to produce a set of outputs. These small number of outputs are realizations of the random variable x which are used to construct the moments μ_i that are to be matched in the optimization. Note that probabilistic moments are typically calculated as $\mu_i = \int x^i p(x) dx$. However, because we have no apriori information about the shape or form of the distribution $p(x)$, we cannot use this method. Consequently, we utilize sample moments [14] $\mu_i = \sum_{j=1}^{N} x_j^i / N$ (N is the number of samples that we draw) to construct the moments for this generic case. By using sample moments, we ensure that the requirement for monotonic moments is satisfied because the random variable x is assumed to be always positive (we can always transform the circuit response to be positive). Consequently, we are guaranteed that the estimated probability distribution $p(x)$ will be stable.

After obtaining the sample moments μ_i for a set $i = 0, 1, ...k$ we perform the maximum entropy moment matching method using traditional Newton's method. We initialize the Lagrange multipliers to 0, $\lambda = [0; 0; ...; 0]$, resulting in the initial guess of the distribution as a uniform distribution. This result is sensical as the uniform distribution inherently has the maximum entropy of all distributions. Next, we let the algorithm continue until the successive changes in multipliers λ_i are within a user specified tolerance. As such, we obtain a probability distribution $p(x)_k$ where k denotes the number of moments that are used.

2.4 Experimental Results using MAXENT

Examples of this work are implemented as the MAXENT algorithm and are shown in [13]. We implemented the proposed algorithm in MATLAB. The first circuit is a 6-T SRAM bit-cell with 54 variables, while the second circuit is a Operational Amplifier with 70 variables. HSPICE is used to simulate these 2 circuits for circuit performance. Also, MC [6] and PEM [1] are used for comparison. PEM is another circuit modeling algorithm that converts probabilistic moments of circuit performance into corresponding time moments of an LTI system then uses Asymptotic Waveform Evaluatiation (AWE) to match these time moments to the transfer function of the system. AWE uses the Pade approximation which generates poles (eigenvalues) that correspond to the dominant poles of the original system, and also poles that do not correspond to the poles of the original system but account for the effects of the remaining poles [23].

Figure 2.3 depicts the 6T SRAM bit cell circuit overview. The reading operation of this cell is viewed as the circuit performance. The reading operation of the cell is determined by the voltage ΔV between $BL$and $\overline{BL}$. If this voltage is large enough to be sensed, it is deemed to be a successful read. The discharge behavior at $\overline{BL}$ plays a crucial role in the value of ΔV. Due to process variations in all transistors, the discharge behavior of $\overline{BL}$ may not be as predicted and therefore the voltage ΔV may not be large enough. Figure 2.4 depicts the Operational Amplifier circuit overview. The bandwidth of this circuit is viewed as the circuit performance.

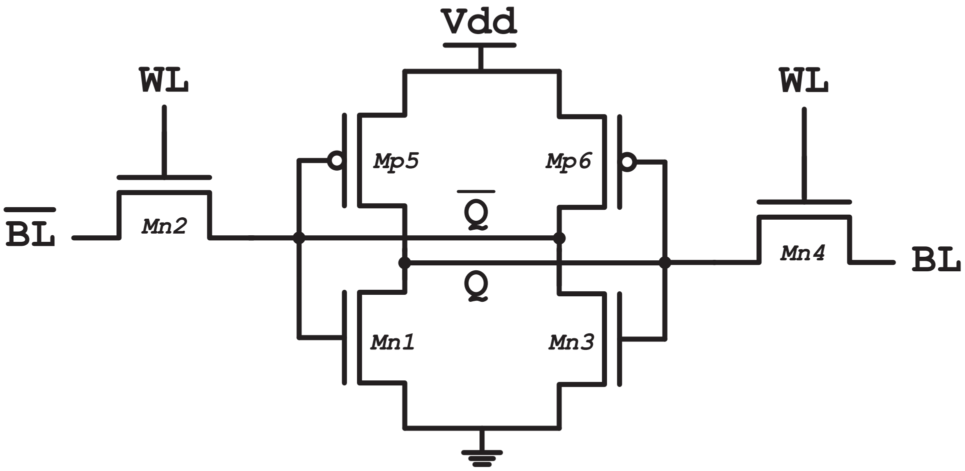

Figure 2.3: 6T SRAM Circuit Layout

2.5 Stability

Figure 2.5 shows the performance distributions generated by MAXENT, PEM, and MC for the first 16 moments and first 18 moments using the 6T SRAM circuit using 200 samples. As we can see, MAXENT is stable under both conditions. The curves representing MAXENT for the first 16 moments and first 18 moments show very good overlap with the ground truth (MC) distribution. On the other hand, only the PEM curve corresponding to 16 moments is stable and overlaps with the ground truth distribution. The only value that changed between these curves are the order of moments that were used. The sample number, circuit topology,

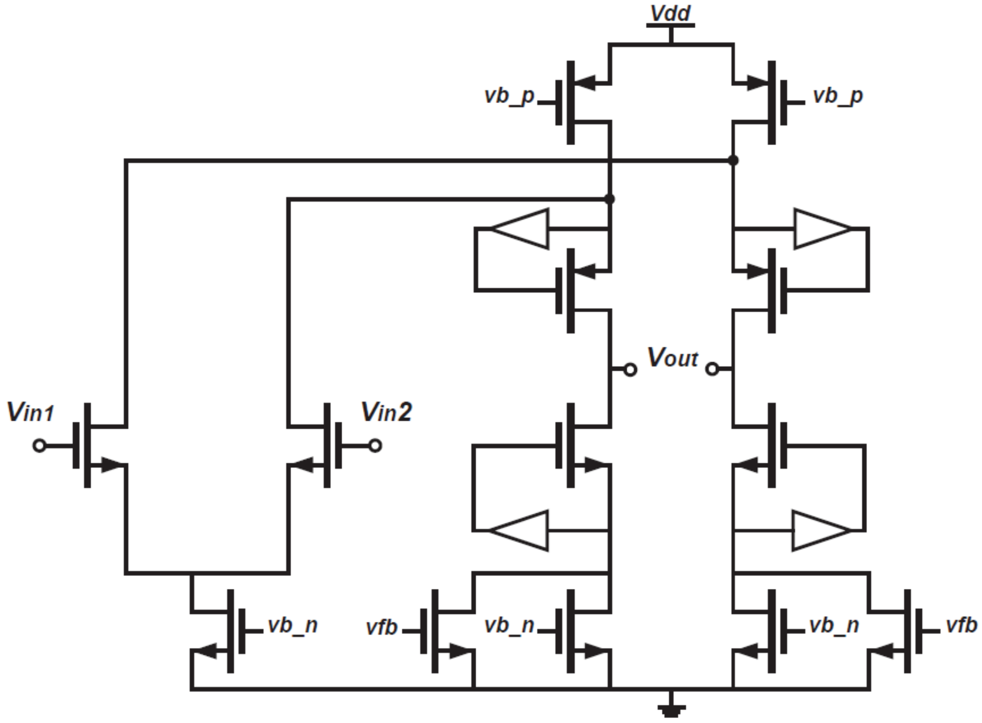

Figure 2.4: Operational Amplifier Circuit Layout

process variations, and all other inputs were held constant. These results imply that PEM is very sensitive to the moments that are used.

Figure 2.6 shows the performance distributions generated by MAXENT, PEM, and MC for the first 16 moments and first 18 moments using the 6T SRAM circuit using 250 samples. As we can see, MAXENT is stable under both 16 moments and 18 moments and overlap well with the ground truth distribution. Moreover, we see that PEM is now stable under both 16 moments and 18 moments and also overlap well with the ground truth distribution. Previously, PEM was unstable for the SRAM circuit using 200 samples and 18 moments, whereas now it is stable for the SRAM circuit using 250 samples and 18 moments.

Figure 2.7 shows the performance distributions generated by MAXENT, PEM, and MC for the first 16 moments and first 18 moments using the 6T SRAM circuit using 300 samples. In this case, we have returned to the instability of PEM. We

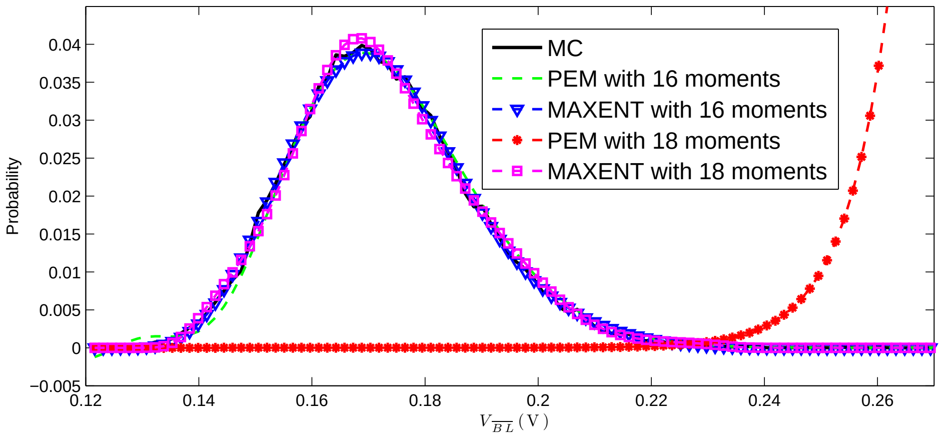

Figure 2.5: PEM lack of stability on SRAM circuit (200 samples)

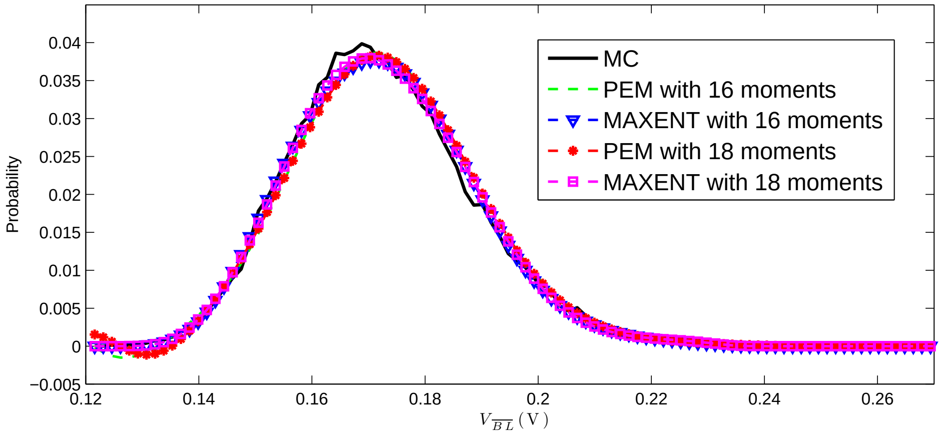

Figure 2.6: PEM stability on SRAM circuit (250 samples)

see that MAXENT is still stable as always, but PEM is now unstable with 18 moments.

Based on Figures 2.6, 2.5, 2.7 we can see that other moment matching methods such as PEM are very sensitive to both the number of moments and the number of samples used. It is sensitive to the number of moments because the Pade approximation (used to estimate the transfer function model of the moment matching algorithm) can produce positive poles leading to instability. Furthermore, because a different number of samples generates a different set of moments

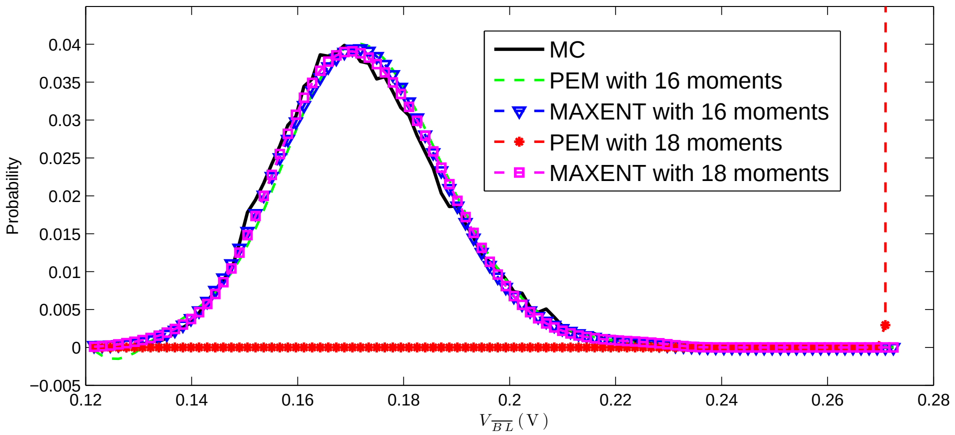

Figure 2.7: PEM lack of stability on SRAM circuit (300 samples)

(recall that moments in both algorithms are sample moments) we may also incur instability because of the Pade approximation. Additionally, this moment matching method uses a transfer function which may allow for negative probabilities - a non-sensical result in the case of probability. On the other hand, MAXENT is guaranteed to be stable if it uses a monotonic set of moments. Its stability is not sensitive to the number of moments or samples used. Moreover, because it is estimated as a product of exponential functions it will never have a negative probability. Consequently, we can see that MAXENT is quite robust compared to other moment matching methods.

2.6 Accuracy

We also evaluated the accuracy of the MAXENT algorithm compared to PEM. Throughout our experiments, MAXENT consistently offers lower error relative to the ground truth than PEM does for any order of moments. We determine the error using the following equation:

$$error = \int (f_1(x) - f_2(x))\, dx \tag{2.15}$$

where $f_1(x)$ is our distribution from MAXENT or PEM and $f_2(x)$ is the ground truth distribution from MC. Figures 2.5, 2.6, 2.7 already illustrate the accuracy of MAXENT on the SRAM circuit for various samples and moment orders. Figure 2.8 illustrates the accuracy of MAXENT on the Operational Amplifier circuit. We see that at a moment order of 10, MAXENT already does a good job of mimicing the overall shape of the distribution, but it lacks some key details. Increasing the moment order to 12 gives an almost exact replica of the ground truth distribution. On the other hand, for a moment order of 10, PEM fails to give an accurate representation of the shape of the distribution. Moreover, increasing the moment order to 12 still yields a disappointing result. The overall shape and accuracy of the distribution from PEM is still very different from the ground truth distribution.

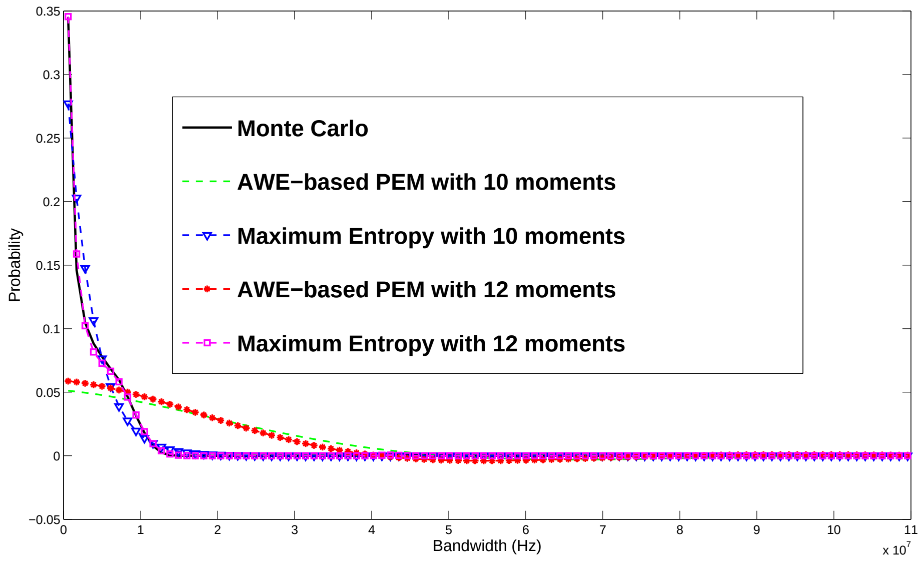

Figure 2.8: Operational Amplifier Accuracy (800 samples)

To quantify the results, Table 2.1 displays the relative error for both MAXENT and PEM in the SRAM and Operational Amplifier circuits. As we can see from both Table 2.1 and Table 2.2, once we reach a steady-state value, MAXENT offers up to 110% lower error for the OpAmp, and up to 27% lower error for the SRAM circuit. We note that although the values of the variance and kurtosis (moment orders 2 and 4) themselves are accurate, the distributions generated using such few orders of moments is very inaccurate. This seems to be an issue of all moment matching algorithms. Consequently, results using such low order of moments is excluded.

Table 2.1: Accuracy Comparision

Circuit	# Samples	Moment Order	PEM Error(%)	MAXENT Error(%)
SRAM	200	6	46.349	11.85
		8	30.656	3.988
		10	15.577	3.281
		12	9.4457	3.394
		14	6.6038	3.181
		18	198.97	5.470
Op. Amp.	200	10	125.54	30.943
		12	116.39	30.881
		14	108.43	5.374
		16	102.05	5.506
		18	93.793	5.567
		20	111.49	5.584

Table 2.2: Accuracy Comparison

Circuit	# Samples	Moment Order	PEM Error(%)	MAXENT Error(%)
SRAM	300	6	46.117	11.043
		8	30.251	5.331
		10	15.097	6.046
		12	11.341	5.818
		14	10.74	6.516
		18	200	6.222
Op. Amp.	800	10	126.51	28.271
		12	117.26	3.851
		14	108.40	4.232
		16	101.110	3.679
		18	94.682	3.465
		20	89.264	3.568

CHAPTER 3

Piecewise Distribution Model

In this section, we propose the PDM (Piecewise Distribution Model) algorithm to accurately and effective model the high sigma portion of non-linear distributions from circuits in high dimensionality. The motivation behind PDM is to accurately model the tail distribution of circuit behavior by using region specific moments. In general, moment matching techniques such as [1, 13] use moments that may accurately reflect the bulk or body of the distribution. However, these global approximation methods use general probabilistic moments which give very little information about the high sigma areas and thus fail to accurately model the tail distribution. To this end, PDM utilizes moment matching to fit the high sigma distribution by using *region specific moments* which capture highly accurate information in regions of interest. In general, an *arbitrary* number of pieces can be used to fit the overall distribution. To this end the proposed algorithm fits two pieces - the first distribution (Segment1) matches the low sigma region and is accurate in the body (typically $\leq 4\sigma$) while the second distribution (Segment2) matches the high sigma region and is accurate in the tail (typically $\geq 4\sigma$). The flow of the method is shown in Figure in 3.1 while details are given below.

3.1 Building the Segment1 Distribution

To build the Segment1 distribution, we first draw samples $q_i; i = \{1, ..., N\}$ from input parameter distributions $f(x_j); j = \{1, ..., p\}$ where p is the number of variables. Next, we simulate these samples using a circuit simulator to obtain circuit

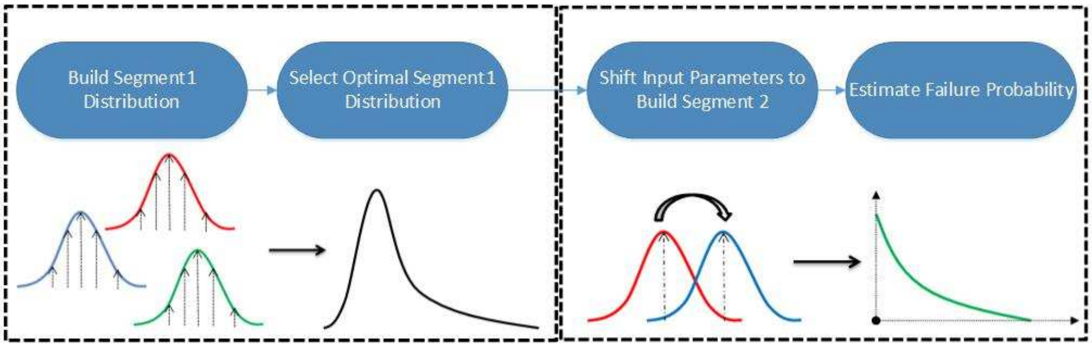
Build Segment1 Distribution
Select Optimal Segment1 Distribution
Shift Input Parameters to Build Segment 2
Estimate Failure Probability

Figure 3.1: PDM contains 4 steps: building the Segment1 distribution, selecting the optimal Segment1 distribution, shifting input parameters to build the Segment2 distribution, and estimating the final probability

behavior outputs $y_i; i = \{1, ..., N\}$. Finally, sample probabilistic moments μ_k are calculated and matched using MAXENT as outlined in [13, 22]. Depending on the number of moments that are matched, we will obtain different Segment1 distributions. However, the exact number of moments to be matched is unknown because we do not know which set of moments map to different areas of the distribution [14]. Consequently, we sweep across a range of values $k = 5, 7, 9, ...$, build multiple Segment1 distributions and select a single, "optimal" Segment1 distribution as explained below.

3.2 Selecting the Optimal Segment1 Distribution

One of the key characteristics of non-Gaussian distributions is that the gradient of their CDFs are monotonically increasing, i.e. the change in circuit behavior for a fixed change in probability continuously increases as the sigma value increases. Here, the sigma value is simply the standard Z-score of a Standard Normal distribution, $P(Z \geq \sigma)$. On the other hand, the gradient is constant for a Gaussian distribution. This is illustrated in Figure 3.2 which shows the gradient of the CDF for a LogNormal (non-Gaussian) distribution vs a Gaussian distribution. Here, although the LogNormal distribution is a mathematical distribution, we label the $y - axis$ of the figure as *Circuit Behavior* to emphasize that this type of circuit behavior is of interest. Consequently, we select the optimal Segment1 distribution by choosing the one with a monotonically increasing gradient.

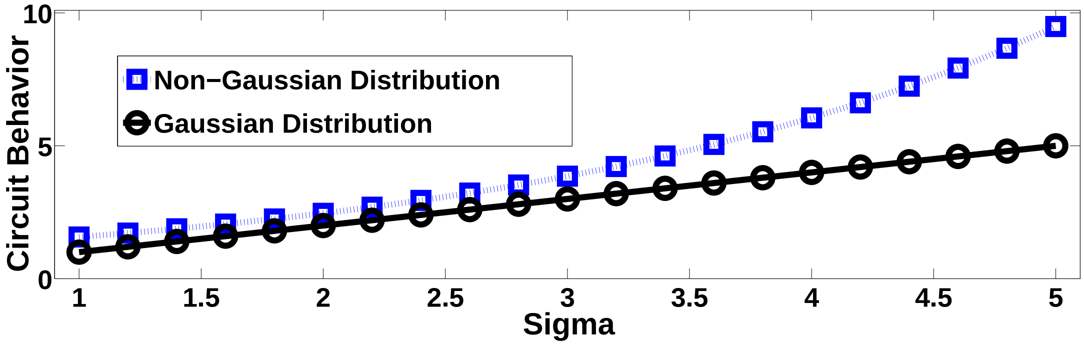

Figure 3.2: Slope of Gaussian vs Non-Gaussian Distribution

In order to gauge the monotonicity of the gradient, we turn to Spearman's Rank Correlation Coefficient [24]. The correlation coefficient ρ is a measure of how well a set of data can be described using a monotonic function. A coefficient of +1 indicates strong correlation to a monotonically increasing function while a coefficient of -1 indicates strong correlation to a monotonically decreasing function. To this end, we measure the gradient of the CDF for various body distributions and compare the data set to a monotonically increasing set using Spearman's Coefficient ρ and select the distribution with the largest, positive coefficient. Figure 3.3 compares various Segment1 distributions, each built with a different number of moments, that are used for approximating a non-Gaussian distribution. We see that the coefficient for 5 of 6 distributions indicates that the gradient data set is monotonically decreasing or uncorrelated. However, there is a single distribution using 14 moments with a coefficient of $\rho = 0.98$, indicating it is a monotonically increasing set and should be used as the optimal Segment 1 distribution. In general, the optimal Segment 1 distribution may not have 14 moments.

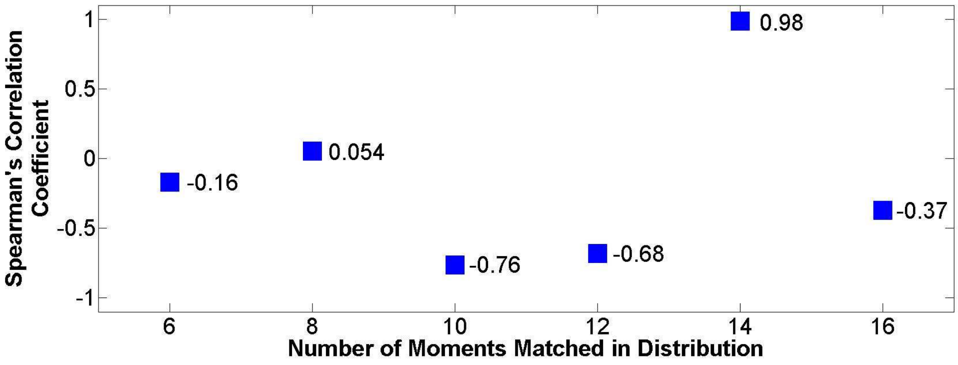

Figure 3.3: Spearman's Correlation of Distributions with Different Moments

To confirm that this is the optimal choice of the above example, we compare the estimated data from the selected Segment1 distribution (strong Spearman's correlation), one non-selected distribution (poor Spearman's correlation), and the ground truth values as shown in Figure 3.4. We see that the selected distribution matches very well with the ground truth because both distributions are non-Gaussian and exhibit monotonically increasing gradients. On the other hand, the distribution with poor correlation is very inaccurate. In short, we utilize this combination of gradient and Spearman's correlation to select the optimal Segment1 distribution used in PDM.

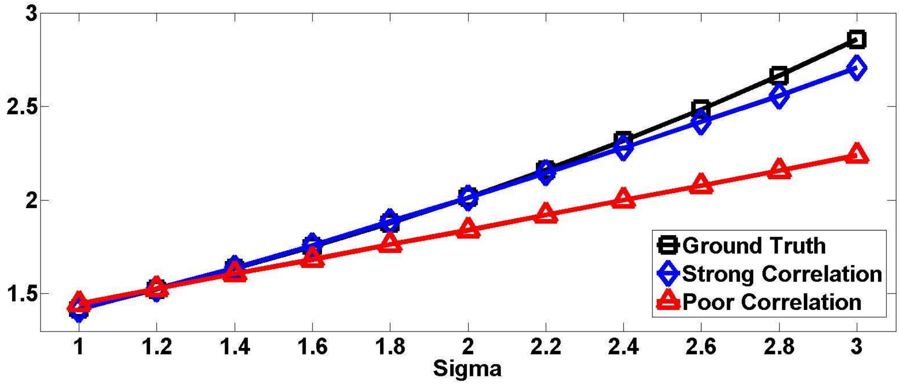

Figure 3.4: Segment1 Comparison using Spearman's Correlation Results

3.3 Shifting Input Distributions and Building the Segment2 Distribution

The motivation behind shifting the input distributions is to draw more samples that yield an output in the tail of the original circuit behavior distribution. By generating more samples in this region, we can generate region specific moments that are highly accurate in the tail. To obtain moments ν_l that are specific to the tail of the distribution, we must shift the mean of the input parameter distributions from m to $\hat{m}$ for each input parameter individually. To shift the mean, we first find the largest circuit behavior y_{max} from the set y_i used when building the Segment1 distribution. Each circuit behavior y_i has a corresponding set of input samples q_j for each input parameter $j = 1, ..., p$. The largest circuit behavior y_{max} will have a sample value q_j^* for each input parameter $j = 1, ..., p$. To obtain the shifted distributions, we simply shift the mean m_j of parameter j to the sample q_j^*.

Once the input parameters are shifted, an additional N_2 samples $\hat{q}_i; i = 1, ..., N_2$ are drawn and simulated yielding an output $\hat{y}_i; i = 1, ..., N_2$. To ensure that the moments ν_l are comprised of information *only* in the tail distribution, we must first screen the simulated data $\hat{y}_i$ such that only samples that lay in the tail are used. To do this, we simply pick a circuit behavior t^* that separates the Segment1 distribution and the next distribution, in this case Segment2. The value of t^* is obtained by selecting a sigma point s in the Segment1 distribution and extracting the corresponding circuit behavior. Typically, s is chosen to be a sigma value between 3 and 4 as this is where the long, flat region of the tail begins as shown in Figure 3.2. Next, the circuit behavior values are screened to obtain $w_k = \hat{y}_i \geq t^*; k = 1, ..., N3$ where $N3$ is the number of points beyond t^*. Because the output was screened, we ensure that the moments ν_l shall only be reflective of the tail distribution's domain and not be polluted by information outside of it.

Finally, to build the Segment2 distribution, we calculate $l = 4$ moments using

$\mu_i = \int x^i p(x)dx$ and match them using maximum entropy as in [13, 19, 21, 22]. The motivation behind using only 4 moments is that this forces the maximum entropy method to yield an exponential distribution as shown in [25]. The exponential distribution is a good approximation of the tail as it is monotonically decreasing and can easily be obtained using the maximum entropy method.

3.4 Reweighing Segment2 via Conditional Probability

Once the Segment2 distribution is obtained, the probability for a specified circuit behavior $t_{critical}$ can be obtained; however, it will be inherently biased because the input parameters were shifted to draw more important samples. To resolve this issue, we use conditional probability to "re-weigh" probabilities as follows

$$P(H \geq t_{critical}) = P(H \geq t_{critical}|B \geq t) * P(B \geq t) \tag{3.1}$$

Where H is the random variable associated with the Segment2 distribution, B is the random variable associated with the Segment1 distribution, $t_{critical}$ is the circuit behavior whose probability is of interest, and t^* is the circuit behavior for sigma point s. This conditional probability relationship works well when the two distributions are identical, i.e. if we are calculating conditional probability under one distribution. However, the proposed algorithm uses conditional probability to re-weigh probabilities of *two* different random variables. If we rearrange (3.1) we get

$$P(B \geq t) = \frac{P(H \geq t_{critical})}{P(H \geq t_{critical}|B \geq t)} \tag{3.2}$$

For a new point $t^{'}_{critical}$, the relationship is

$$P(H \geq t^{'}_{critical}) = P(H \geq t^{'}_{critical}|B \geq t)P(B \geq t) \tag{3.3}$$

$$P(B \geq t) = \frac{P(H \geq t'_{critical})}{P(H \geq t'_{critical} | B \geq t)} \tag{3.4}$$

Rearranging (3.2) and (3.4) and equating the common term yields

$$P(B \geq t) = \frac{P(H \geq t_{critical})}{P(H \geq t_{critical} | B \geq t)} = \frac{P(H \geq t'_{critical})}{P(H \geq t'_{critical} | B \geq t)} \tag{3.5}$$

If this relationship were true, then the denominator changing by a factor α would force the numerator to change by this same factor. However, the only time this will happen is if the distributions from the numerator and denominator (joint and conditional, respectively) are identical as in importance sampling algorithms such as [26]. Consequently, we propose a dynamic scaling technique that additionally reweighs the probability under the Segment2 distribution by a scaling factor β. The scaling factor is a heuristic that is calculated based on the number of outputs in w_k that lay beyond the circuit behavior of interest $t_{critical}$, and the total number of outputs N_3 as shown in (3.7). The scaling factor attempts to improve the accuracy in the conditional probability weight purely based on the samples rather than assumptions about the shape of the distribution.

$$I(w_k) = \begin{cases} 0 \text{ if } w_k < t_{critical} \\ 1 \text{ if } w_k \geq t_{critical} \end{cases} \tag{3.6}$$

$$\beta = \sum_{k=1}^{N_3} \frac{I(w_k)}{N_3} \tag{3.7}$$

Using this scaling factor yields the final probability of a specified circuit behavior $t_{critical}$ as (3.8)

$$P(H \geq t_{critical}) = P(H \geq t_{critical} | B \geq t) * P(B \geq t) * \beta \tag{3.8}$$

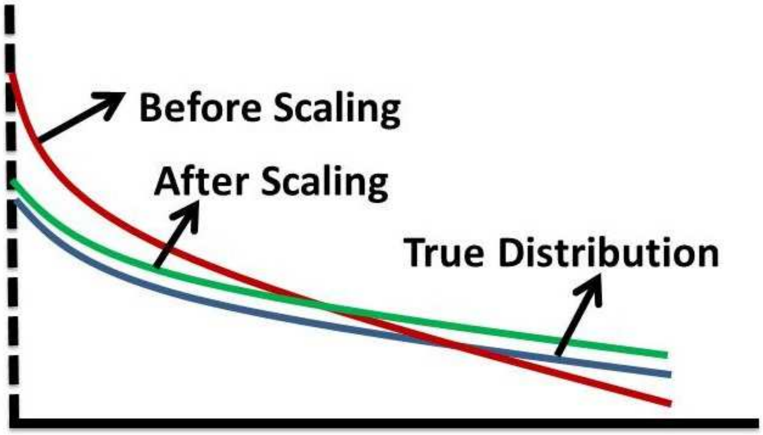

Figure 3.5: Shape Issue in Conditional Probability

Figure 3.5 shows an example of the difference in shape between the true tail distribution, the unscaled Segment2 distribution and the scaled Segment2 distribution. Additionally, we note that both Segment1 and Segment2 distributions are guaranteed to be stable, i.e. they will have a non-negative probability and therefore the CDF is guaranteed to be monotonic. This naturally arises because both distributions are calculated using the maximum entropy method and all moments in both segments are monotonically increasing.

CHAPTER 4

Experiment Results

4.1 Experiment Settings

We implemented PDM in MATLAB using simulation outputs from HSPICE. The algorithm was tested against the mathematically known LogNormal distribution, along with the high sigma delay of a six stage clock path circuit and gain of an Operational Amplifier. The results show the estimated sigma for multiple $t_{critical}$ values and are compared to Monte Carlo as ground truth. The Monte Carlo results were generated with roughly $8E6$ samples for the Time Critical Path and $2.5E6$ samples for the Operational Amplifier. Additionally, we compare the results to the moment matching algorithm MAXENT [13] to show the improvements using a piecewise fitting method rather than a global approach. We also compare the results to High Dimensional Importance Sampling (HDIS) [9] to show that the reweighing portion of PDM is robust as it is independent of dimensionality because it is performed in the output domain, leading to high accuracy in high dimensional circuits. Table 4.1 gives an overview of the variables used in each circuit.

The time critical path circuit has six stages and nine process parameters per transistor, while the circuit behavior of interest is the delay from input to output. Figure 4.1 displays a schematic of the two-stage differential cascode operational amplifier. The circuit has a total of thirteen transistors and four gain boosting amplifiers. In total, only ten transistors are considered to be independently varied. However, transistors in the gain boosting amplifiers are also varied, though due

Table 4.1: Parameters of MOSFETs

Variable Name	Time Critical Path	OpAmp
Flat-band Voltage	†	
Threshold Voltage		†
Gate Oxide Thickness	†	†
Mobility	†	†
Doping concentration at depletion	†	
Channel-length offset	†	†
Channel-width offset	†	
Source/drain sheet resistance	†	†
Source-gate overlap unit capacitance	†	†
Drain-gate overlap unit capacitance	†	†

to the mirrored properties of the circuit they are varied simultaneously and are counted as one variation. As such, although each transistor has seven process parameters resulting in a total of seventy variables, the true number of variables is much higher. The circuit behavior of interest is the gain $\frac{V_{out1}}{V_{in1}}$.

4.2 Experiment on Mathematical Distribution

To illustrate the capability of modeling strongly non-Gaussian distributions, we use PDM to model a LogNormal distribution. The LogNormal distribution with mean and sigma parameters $\mu = 0$, $\sigma = 0.35$ was selected because of its strongly non-Gaussian behavior. A plot of the PDF of this distribution is presented in Figure 4.2. The distribution appears to be Gaussian for a small portion due to the bell shaped curve, but it has a very long tail, giving it the non-Gaussian properties that are of interest.

Figure 4.3 shows the high sigma modeling results for Monte Carlo, MAXENT, and PDM at multiple $t_{critical}$ points. The figure is the CDF zoomed into the tail area with the x-axis as probability and y-axis as the value of the random variable, precisely circuit behavior. Here sigma is used to represent probability, i.e. $4\sigma \approx 0.000064$ in the tail. The motivation for this type of plot is to best

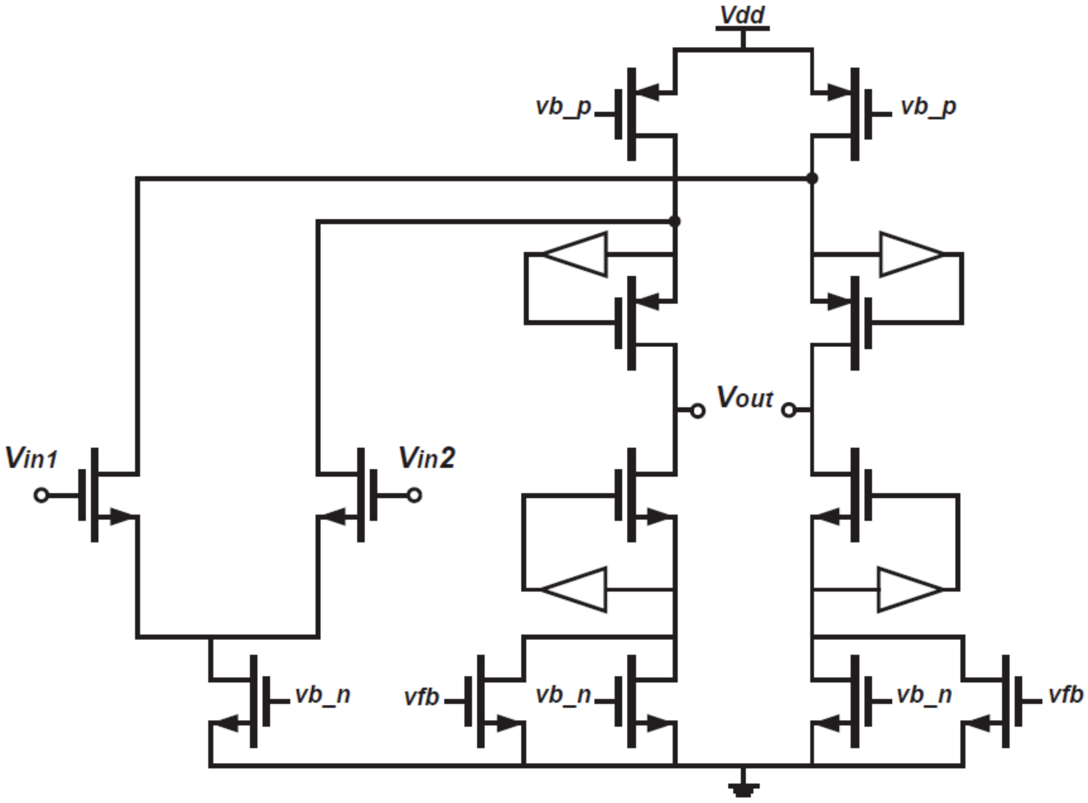

Figure 4.1: Op. Amp Schematic

represent the non-linear behavior of a non-Gaussian PDF. Additionally, it shows only the high sigma behavior rather than the overall distribution because that is the motivation and focus behind this algorithm.

HDIS, MAXENT and PDM each used a total of 4000 samples, with PDM using 3000 samples to calculate the Segment1 distribution and 1000 samples to calculate the Segment2 distribution. In this case, the point s that separates Segment1 and Segment2 is selected to be the 4 sigma point, i.e. whatever circuit behavior that corresponds to a tail probability of $6.4E-5$ in the Segment1 distribution. By introducing the Segment2 distribution at the point s, PDM is able to avoid any errors that MAXENT suffers from, allowing PDM to match almost identically with the Monte Carlo results up to 4.8 sigma. By utilizing region specific moments and doing a piecewise fitting of the distribution, PDM keeps consistently small errors.

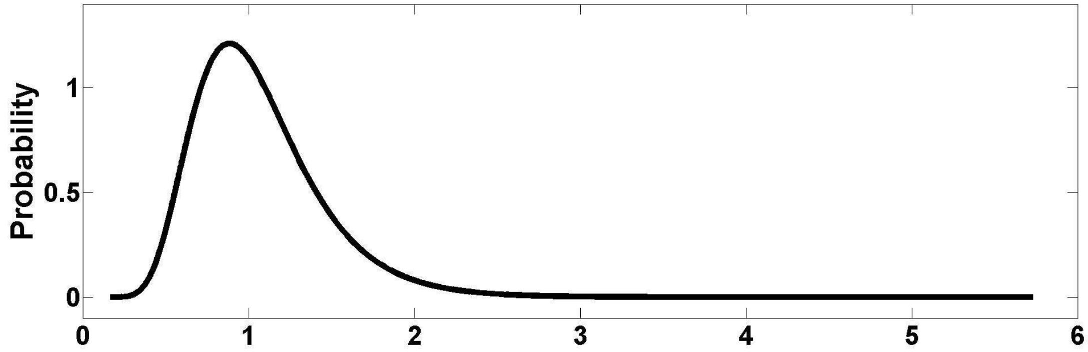

Figure 4.2: LogNormal PDF

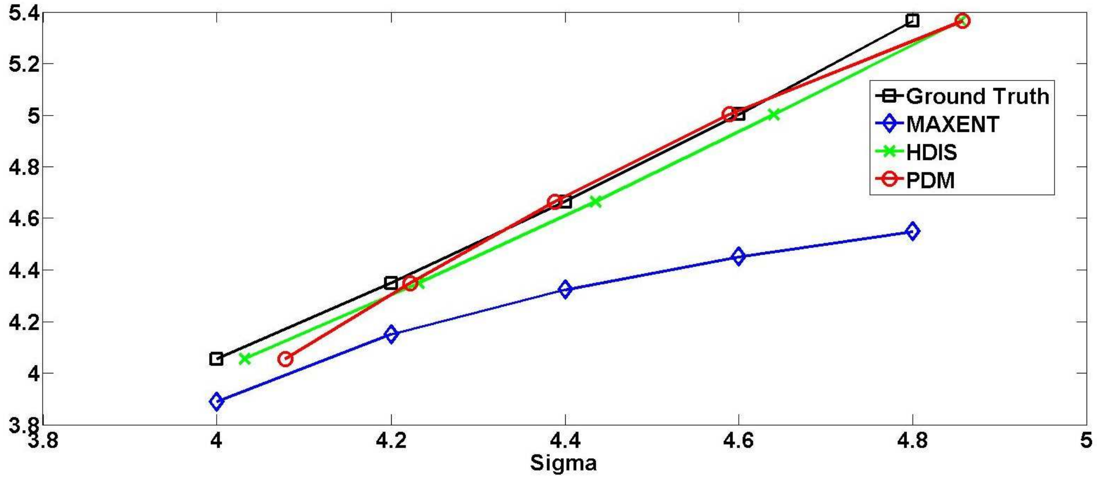

Figure 4.3: LogNormal Sigma Behavior

On the other hand, the MAXENT algorithm begins to lose accuracy and fails to capture the tail of the distribution because it only uses one distribution to model the overall behavior.

Furthermore, we see that HDIS has accuracy comparable to both PDM and Monte Carlo. At the 4 sigma point, we see that HDIS is slightly more accurate. However, between 4 and 4.8 sigma we see that PDM is more accurate, with both algorithms exhibiting the same, good accuracy at the 4.8 sigma point. These results intuitively make sense as the LogNormal distribution has only 1 variable so HDIS does not suffer from the curse of dimensionality. Moreover, because it has only 1 variable, it is able to find a good shift. Similarly, PDM is able to maintain very high accuracy because it matches region specific moments. Table 4.2 shows the error in estimated sigma for PDM. The error is between -0.25% and 2% all the way to the 4.8 sigma point.

We also note that MAXENT and PDM do *not* assume the distributions to be matched are Gaussian distributions because they do not match only 3 moments. [25] outlines that the maximum entropy moment matching method can be forced to assume a Gaussian distribution if we match exactly 3 moments. However, because we sweep through a wide range of moments for both MAXENT and PDM we, in general, will never pick a Gaussian distribution because it does not agree with the gradient criteria selected by Spearman's correlation coefficient. Consequently, the high error that MAXENT suffers from is due to its limitations of using one set of moments, not from any assumptions about its model.

Table 4.2: Sigma Error for LogNormal

True Sigma	Estimated Sigma	% Error
4.0	4.0786	1.9650%
4.2	4.2224	0.5333%
4.4	4.3886	-0.2591%
4.6	4.5888	-0.2435%
4.8	4.8569	1.1854%

4.3 Experiments on Circuits

The Monte Carlo distribution of the time critical path circuit delay is presented in Figure 4.4. Because the circuit operates at a very low VDD level, it behaves in a slightly non-linear way. The distribution, while not as long tailed as the LogNormal, has a more elongated tail than a Gaussian distribution.

Figure 4.5 shows the high sigma modeling results for Monte Carlo, MAXENT, HDIS, and PDM at multiple $t_{critical}$ points. HDIS, MAXENT and PDM each used

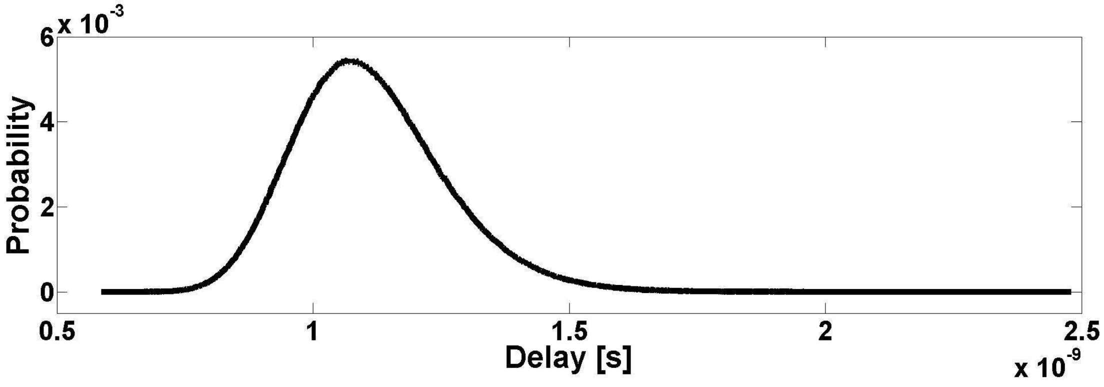

Figure 4.4: Clock Path PDF

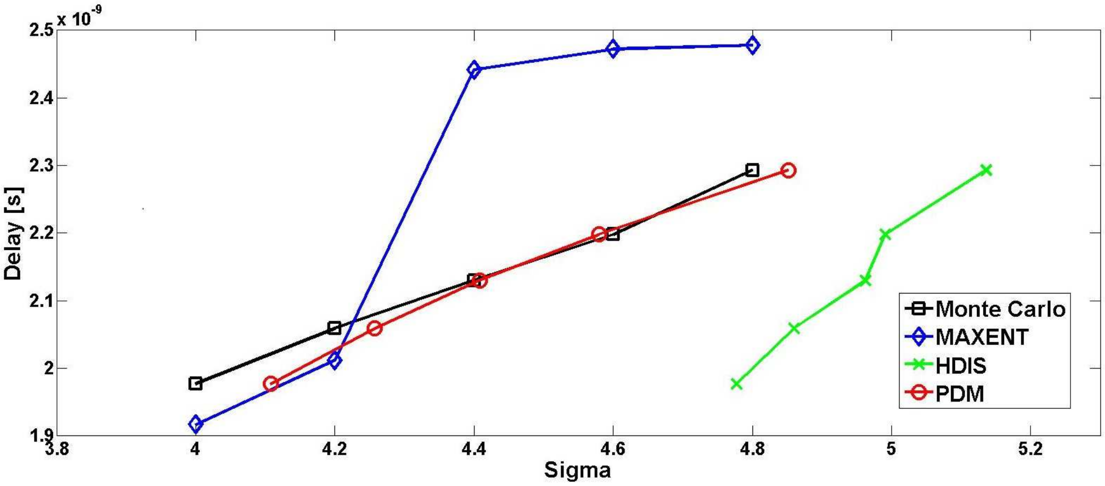

Figure 4.5: Clock Path Sigma Behavior

a total of 4000 samples, with PDM using 3000 samples to calculate the Segment1 distribution and 1000 samples to calculate the Segment2 distribution. Once again in PDM, the point s is the 4 sigma point from the Segment1 distribution. By introducing the Segment2 distribution at the point s, PDM is able to avoid any errors that MAXENT may suffer from. This is most apparent at the 4.4 sigma point and beyond. Additionally, PDM is able to capture the increase in slope as the circuit approaches higher sigma. On the other hand, MAXENT is able to perform somewhat well up to 4.2 sigma but then blows up and becomes completely inaccurate afterwards. The significant increase in accuracy with PDM is, again, due to matching region specific moments that allow piecewise fitting of the distribution. Because MAXENT uses a single distribution to make a global approximation it is unable to capture the tail of the distribution and instead models the high sigma points purely as noise. We again note that MAXENT does not assume the distribution is a Gaussian model, so its error is due to limitations of using one set of moments to model the total distribution which PDM does not suffer from.

Furthermore, we see that the results from HDIS are completely inaccurate compared to both Monte Carlo and PDM. HDIS is unable to come anywhere to the proper sigma value for any of the points that it estimates. This is likely inaccurate from a combination of high dimensionality and an inaccurate shift in the mean and sigma of the new sampling distribution that causes the re-weighing process to again become inaccurate. Simply put, if the shifting method is inaccurate the results from HDIS will be inaccurate. If a larger number of samples is used, then the shift and corresponding samples drawn from the new distribution will be more accurate; however, due to the run time prohibitive nature of high dimensional circuits, it is imperative to minimize the number of samples. On the other hand, the shifting method in PDM is more robust because the re-weighing process is performed in the output domain and is performed using conditional probability rather than as a ratio of two distributions. Table 4.3 shows the error in sigma between PDM and the ground truth from Monte Carlo. We see a worst case error of 2.7% at 4 sigma but significantly lower errors at higher sigma values.

The Monte Carlo distribution of the Operational Amplifier circuit gain is shown in Figure 4.6. The distribution is heavily skewed and has a very sharp peak near the beginning and proceeds to drop very quickly, However, it also has a slightly flatter portion that eventually decreases to a long, flat region of the tail. It clearly has a long tail and behaves in a strongly non-Gaussian way.

Figure 4.7 shows the high sigma modeling results for Monte Carlo, MAXENT,

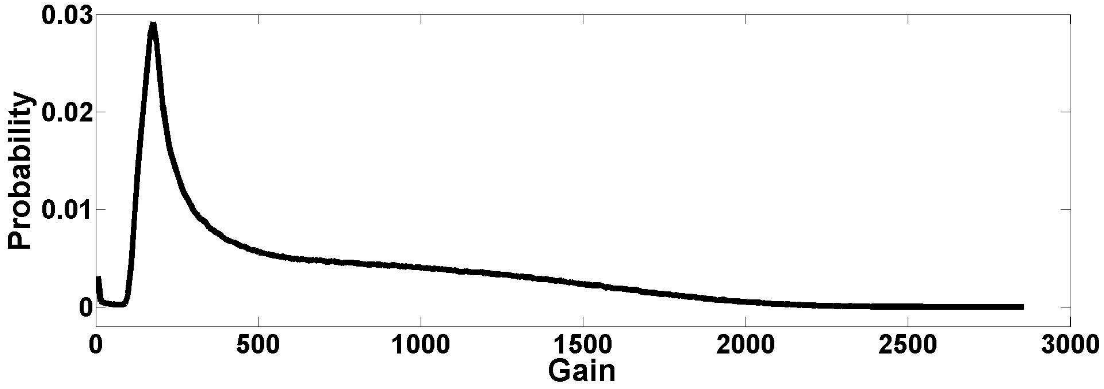

Figure 4.6: Op. Amp PDF

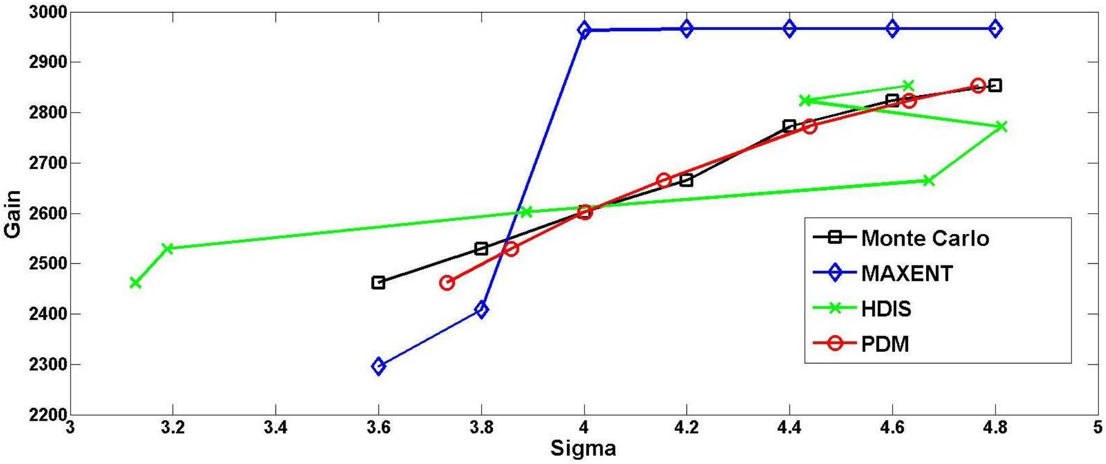

Figure 4.7: Op Amp Sigma Behavior

and PDM at multiple $t_{critical}$ points. The figure shows only the high sigma behavior rather than the overall distribution because that is the motivation and focus behind this algorithm. Both MAXENT and PDM used a total of 3000 samples, with PDM using 2000 samples to calculate the Segment1 distribution and 1000 samples to calculate the Segment2 distribution. In the case of the OpAmp, the point s was determined to be the 3.6 sigma point rather than the 4 sigma point as in the previous cases due to the extremely long-tailed nature of the distribution. Before the point s, it's clear that PDM has a larger error (roughly 5%) than in previous cases. However, when we introduce the Segment2 distribution, PDM is able to immediately recover and match the 3.8 sigma point closely and continues to match larger sigma points and the overall shape of the Monte Carlo curve very well. By introducing this second "piece" to fit the distribution, we are able to get a significant increase in accuracy. On the other hand, the MAXENT method has a large error, blows up and returns noise values because it is unable to capture the tail of the distribution as it does not use moments that are specific to that region. We again note that MAXENT does not assume the distribution is a Gaussian model because it matches more than 3 moments. Hence, its error is due to limitations of using one set of moments to model the total distribution.

Moreover, we see that the results from HDIS are inaccurate throughout the entire curve and at one point has a huge jump in its results and is simply noisy throughout. Although the Operational Amplifier circuit is not as high dimensional as the Clock Path, HDIS is still unable to properly model the high sigma region. Again, the inaccuracy is most likely from an inaccurate shift in the mean and sigma of the new sampling distribution that causes the re-weighing process to again become inaccurate. Table 4.3 shows the error in estimated sigma between PDM and the ground truth from Monte Carlo. We see very accurate results with a worst case error of about -1% at 4.2 sigma.

Table 4.3: Sigma Error for Circuits

Time Critical Path			Op Amp		
True Sigma	Estimated Sigma	% Error	True Sigma	Estimated Sigma	% Error
4.0	4.1077	2.693%	4.0	4.0015	0.0375%
4.2	4.2571	1.360%	4.2	4.1547	-1.0786%
4.4	4.4080	0.182%	4.4	4.4386	0.8773%
4.6	4.5793	-0.450%	4.6	4.6329	0.7152%
4.8	4.8517	1.077%	4.8	4.7662	-0.7042%

4.4 Speedup Comparison

To analyze the efficiency of the proposed method, we compare the number of samples required by PDM to the number of samples used for Monte Carlo. Since the LogNormal distribution is a mathematically known circuit and requires no Monte Carlo simulations, we exclude that speedup comparison. In the clock path circuit, PDM requires a total of 4000 samples - 3000 samples for the body distribution and 1000 for the hybrid distribution. In the Operational Amplifier, PDM requires a total of 3000 samples - 2000 samples for the Segment1 distribution and 1000 for the Segment2 distribution. Table 4.4 compares the Monte Carlo and PDM runtime requirements and the speedup for all circuit examples. We note that the speedup of the algorithm compared to Monte Carlo will vary based on the number of samples that are used; however, it is clear that PDM offers a significant speedup at very little loss in accuracy.

Table 4.4: Speedup Comparison

Circuit	Monte Carlo Runtime	PDM Runtime	Speedup
Clock Path	8,000,000	4000	2000x
Op. Amp.	2,500,000	4000	625x

CHAPTER 5

Conclusions and Future Work

In this thesis, we presented the motivation for statistical modeling of circuit performance and presented two novel algorithms that solve this problem. The first algorithm was based on the maximum entropy moment matching method in the communications and signal processing field. We showed that this algorithm is provably stable under general statistical circuit analysis methods. We also showed that it offers high accuracy and stability when compared to other moment matching methods [1, 3]. However, we showed that MAXENT is unable to accurately model the high sigma behavior of non-Gaussian circuits and is therefore unsuitable for yield analysis. To this end, we proposed PDM - a piecewise distribution fitting method that performs region based moment matching to extract the PDF of circuit performance. We showed that PDM is provably stable because it is based on the maximum entropy method, and also showed that is is able to model the high sigma regions of the circuit performance PDF. In particular, we showed that introducing a second distribution based on a set of moments that are accurate in the tail of the PDF leads to significantly improved accuracy over MAXENT [13] and shows little error with respect to Monte Carlo. Future work involves application of both MAXENT and PDM to multiple digital and analog circuits. Additionally, we propose utilizing a General Pareto Distribution (GPD) to fit the Segment2 distribution in PDM as the GPD is known to be an accurate model for the tail of probability distributions [10]. Finally, we plan to develop a weighted moment matching based approach that allows us to pick and choose the important moments of a distribution. The motivation behind this is not all moments are important to the distribution, e.g. in a Gaussian distribution only even order moments are non-zero, and therefore applying more weight to “important” moments may help improve accuracy and reduce noise.

References

[1] F. Gong, H. Yu, and L. He, “Stochastic analog circuit behavior modeling by point estimation method,” *Proceedings of the 2011 international symposium on Physical design*, pp. 175–182, 2011.

[2] S. Nassif, “Modeling and analysis of manufacturing variations,” *IEEE article on Custom Integrated Circuits*, pp. 223–228, 2001.

[3] X. Li, J. Le, P. Gopalakrishnan, and L. Pileggi, “Asymptotic probability extraction for non-normal distributions of circuit performance,” *Proceedings of the 2004 IEEE/ACM International article on Computer-aided design*, pp. 2–9, 2004.

[4] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar, “Near-threshold voltage (ntv) design: opportunities and challenges,” in *Proceedings of the 49th Annual Design Automation Conference.* ACM, 2012, pp. 1153–1158.

[5] S. Vrudhula, J. Wang, and P. Ghanta, “Hermite polynomial based interconnect analysis in the presence of process variations,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 10, pp. 2001–2011, 2006.

[6] C. Jacoboni and P. Lugli, *The Monte Carlo method for semiconductor device simulation.* Springer, 2002, vol. 3.

[7] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, “Breaking the simulation barrier: Sram evaluation through norm minimization,” in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design.* IEEE Press, 2008, pp. 322–329.

[8] F. Gong, S. Basir-Kazeruni, L. Dolecek, and L. He, “A fast estimation of sram failure rate using probability collectives,” *ACM International Symposium on Physical Design*, pp. 41–47, 2012.

[9] W. Wu, F. Gong, C. Gengsheng, and L. He, “A fast and provably bounded failure analysis of memory circuits in high dimensions,” in *Proceedings of the 2013 Asia and South Pacific Design Automation Conference.* IEEE, 2013.

[10] A. Singhee and R. A. Rutenbar, “Statistical blockade: very fast statistical simulation and modeling of rare circuit events and its application to memory design,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 8, pp. 1176–1189, 2009.

[11] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," *Probability and statistics: Essays in honor of David A. Freedman*, vol. 2, pp. 316–334, 2008.

[12] R. Y. Rubinstein and P. W. Glynn, "How to deal with the curse of dimensionality of likelihood ratios in monte carlo simulation," *Stochastic Models*, vol. 25, no. 4, pp. 547–568, 2009.

[13] R. Krishnan, W. Wu, F. Gong, and L. He, "Stochastic behavioral modeling of analog/mixed-signal circuits by maximizing entropy," in *Quality Electronic Design (ISQED), 2013 14th International Symposium on.* IEEE, 2013, pp. 572–579.

[14] R. Durrett, *Probability: theory and examples.* Cambridge university press, 2010, vol. 3.

[15] W. Wu, W. Xu, R. Krishnan, Y.-L. Chen, and L. He, "Rescope: High-dimensional statistical circuit simulation towards full failure region coverage," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference.* ACM, 2014, pp. 1–6.

[16] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods.* SIAM, 1992, vol. 63.

[17] E. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[18] P. McCullagh, *Tensor methods in statistics.* Chapman and Hall London, 1987, vol. 161.

[19] X. Wu, "Calculation of maximum entropy densities with application to income distribution," *Journal of Econometrics*, vol. 115, no. 2, pp. 347–354, 2003.

[20] G. Judge and D. Miller, *Maximum entropy econometrics: Robust estimation with limited data.* John Wiley & Sons, 1997.

[21] B. Chen, J. Hu, and Y. Zhu, "Computing maximum entropy densities: A hybrid approach," *Signal Processing: An International Journal (SPIJ)*, vol. 4, no. 2, p. 114, 2010.

[22] L. Mead and N. Papanicolaou, "Maximum entropy in the problem of moments," *Journal of Mathematical Physics*, vol. 25, p. 2404, 1984.

[23] P. Feldmann and R. Freund, "Efficient linear circuit analysis by padé approximation via the lanczos process," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 14, no. 5, pp. 639–649, 1995.

[24] M. G. Kendall, “Rank correlation methods.” 1948.

[25] K. Conrad, “Probability distributions and maximum entropy,” *Entropy*, vol. 6, no. 452, p. 10, 2004.

[26] F. Gong, S. Basir-Kazeruni, L. He, and Y. Hao, “Stochastic behavioral modeling and analysis for analog/mixed-signal circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 24–33, 2013.