Early, Accurate and Fast Yield Estimation through Monte Carlo-Alternative Probabilistic Behavioral Analog System Simulations

Rasit Onur Topaloglu
University of California at San Diego, Computer Science and Engineering Department
La Jolla, CA, 92093-0404, rtopalog@cse.ucsd.edu

Abstract

Monte Carlo analysis has so far been the corner stone for analog statistical simulations. Fast and accurate simulations are necessary for stringent time-to-market, design for manufacturability and yield concerns in the analog domain. Although Monte Carlo attains accuracy, it does so with a sacrifice in run-time for analog simulations. In this paper, we propose a fast and accurate probabilistic simulation method alternative to Monte Carlo using deterministic sampling and weight propagation. We furthermore propose accuracy improvement algorithms and a fast yield calculation method. The proposed method shows accuracy improvement combined with a 100-fold reduction in run-time with respect to a 1000-sample Monte Carlo analysis.

1 Introduction

Increased process variations and mismatch in new technologies make statistical simulation for analog systems a necessity. Today's requirements necessitate faster and more accurate statistical simulations. One major target is yield estimation under process variations. If yield estimation can be done accurately and fast in the early stages of an analog system design, costly design iterations can be avoided.

Output parameters are assumed to have Gaussian densities most of the time [25]. Although Gaussian assumption might be sufficient for most input parameters, it is far from being accurate as an assumption for most analog output parameters. Traditional techniques usually provide the mean and the variance for an output parameter. Yet, capturing the exact shape of a continuous density at the output is important when yield estimation is needed, as yield estimation requires an accurate integral under the density with limits determined by circuit parameter specification window.

Speed and accuracy of today's yield estimation techniques are lagging behind technology. Statistical simulation of large blocks brings significant time burdens on system designers. This has triggered us to come up with a technique to bring a solution to single input parameter statistical simulations. Since individual blocks are highly important in analog systems, single parameter simulations will give quite a deal of information. The whole system can then either be evaluated using single parameter as input and multiple parameters as output by selecting the dominant input parameter, or using the proposed system for more input parameters separately. The proposed estimation methods can be used for optimization as well.[1]

We first introduce the basis of the proposed technique, where probability discretization and sample propagation are implemented using a forward operation. We have gained significant run-time improvement over Monte Carlo through weight propagation and systematic sampling. Then, we provide algorithms to convert the propagated samples to a continuous density. These algorithms are rebinning, sample skipping and zero padding, followed by a traditional spline interpolation. Then yield-estimation algorithm is provided. We have used extensive behavioral simulations on the jitter analysis of a phase-locked loop to validate the proposed computational techniques.

2 Previous Work

A number of approaches for probabilistic simulation of analog blocks has been proposed. [13] and [10] have used principal component analysis on the correlation matrix of process variations to reduce the number of variables that are responsible for mismatch. In [23], sensitivity analysis followed by response surface polynomial fitting and Monte Carlo sampling is implemented. [28] has used regression analysis including second order terms. [9] has used variance propagation.

Particular interest in probabilistic simulation has been in the area of mismatch and process variation simulation. [21] has used Tailor series expansion. [6] has used sensitivity analysis and assumed that there are at least the same number of output parameters as there are input parameters, all of which are Gaussian. [18] has applied principal component analysis to account for correlations between input parameters. [30] has used hierarchical sensitivities for probabilistic simulation. These methods are good for calculating mean and variance. They will work accurate only if outputs are

[1] Most multi-parameter optimization techniques also work on single parameters at a time for improved convergence [7].

Proceedings of the 24th IEEE VLSI Test Symposium (VTS'06)
0-7695-2514-8/06 $20.00 © 2006 IEEE

Gaussian, although it is known that this assumption fails for analog circuits. Almost all designers count on Monte Carlo methods for probabilistic simulations.

Analog fault simulation is another field where probabilistic simulations have been important, since analog faults are related to parameter probability densities. [20] has approximated mean and variances at the output analytically. [15] has used a hierarchical variance analysis. [24] and [32] have used a sensitivity-based Monte Carlo technique.

There are a number of Monte Carlo methods such as Latin Hypercube sampling [16] or rejection sampling [17] [3] [8]. Importance sampling and rejection sampling require a sampling density. As long as the sampling density is similar to the density to be estimated, these methods work fine. If such a sampling density is not available, a set of densities can be provided as in the Metropolis method. Yet, this requires more samples eventually. Gibbs sampling works for dimensions higher than 2, hence it is not applicable to the problem in this paper [17] [2].

Propagation of the densities among parameters is also a related topic. This can be handled through Monte Carlo by propagating individual data points. Traditionally, variance propagation has been used. Parametric approaches, such as belief propagation [27], propagate certain moments of the density.

For behavioral simulations, particular attention has been on phase-locked loop simulations. [12] has introduced RF front-end models both for time and frequency domains. [19] has proposed a model calibration method for PLLs. [4] has compared numerical methods for phase/delay-locked systems. Behavioral simulations for phase-locked loops (PLLs) are important as most analyses require transient simulations over many cycles, indicating high run-times. The probabilistic nature makes the problem harder.

Although probabilistic simulation has been traditionally used for analog circuits, digital circuits within the last couple of years took its portion [33] [14] [26] [34]. Recently, major contributions have come in the digital domain due to the necessity to have probabilistic static timing and leakage analysis. These methods are usually specific to the particular problem but not suitable for black-box-type simulations. For example, only a limited set of all possible non-linear operators, e.g. addition and maximum operations, are targeted in the probabilistic timing simulations in the literature and the formulae that tie inputs to outputs are assumed to be analytically provided. RF domain yield estimation [1] and communication systems, on the other hand, are very similar to the analog system studied herein and hence can take advantage of the proposed techniques in this paper.

An accurate yet computationally expensive hierarchical version of the proposed method has been introduced for probabilistic device simulation [31] and preliminary algorithms for analog systems were presented as a poster at [29]. In this paper, we introduce new algorithms for black-box type of system simulations suitable for very fast analog behavioral simulations specialized in accurate density and yield estimation. In particular, we introduce algorithms for accuracy improvement and yield estimation over [29].

3 Preliminaries

In order to introduce forward discrete probability propagation (FDPP), a number of definitions will be useful. Let X be a random variable. We will denote the probability density of X as $pdf(X)$. $pdf(X)$ is assumed to be continuous. We propose to attain an approximation of this pdf by sampling the pdf at equidistant points of the random variable X.

3.1 Deterministic Sampling

The sampling can be done by dividing the $pdf(X)$ to bins and approximating the values that fall in any bin by the value at the mid-point of the bin. The bins are generated such that the highest and lowest samples are accounted for. Let b_i be an enumeration over the bins where $1 \leq i \leq N$ and N is the total number of bins. b_i will be defined to be bounded by $[m + (i-1)\Delta, m + i\Delta)$, where Δ is the step-size defined by $\frac{n-m}{N}$. The N'th bin, however, is bounded by $[m + (N-1)\Delta, m + N\Delta]$, which is equal to $[n - \Delta, n]$. We denote the sampled $pdf(X)$ as $\phi(X)$ or $spdf(X)$.

The procedure of converting a pdf to an $spdf$ will be represented with the $\mathcal{Q}_{\mathcal{N}}$ operator:

$$\phi(X) = \mathcal{Q}_N(pdf(X)) \tag{1}$$

The domain of this operator is a pdf, and the range of this operator is an $\phi(X)$. The result of this operator on the pdf of random variable X, $\phi(X)$, is essentially a Riemann sum of samples and is given by:

$$\phi(X) = \sum_{i \epsilon 1..N} p_i \delta(x - w_i) \tag{2}$$

where

$$p_i = \int_{m+(i-1)\Delta}^{m+i\Delta} pdf(X)dx \tag{3}$$

$$w_i = m + (i-1)\frac{\Delta}{2} \tag{4}$$

In these equations, p_i is the sample height, which corresponds to the integral in the corresponding bin b_i. w_i is the mid-point in the particular bin b_i.

3.2 Forward Propagation

Assume that we have a random variable given as X whose $spdf$ is given by $\phi(X)$. Let Y be another random variable that is given by a deterministic function f of the given random variable: $Y = f(X)$. Then $\phi(Y)$ is given by the $\mathcal{F}$ operator as:

$$\phi(Y) = \mathcal{F}(\phi(X)) \tag{5}$$

which is defined as being equivalent to:

$$\phi(Y) = \sum_{i \epsilon \phi(X)} p_i \delta(y - f(w_i)) \tag{6}$$

Algorithm Implementing the $\mathcal{F}$ Operator:

```
While all samples of the target pdf computed
    For each sample in X
        Place an sample with height $p_i$ at x=f($w_i$)
```

Here, Y is the output parameter to be observed and X is the input parameter. The domain and the range of the $\mathcal{F}$ operator is an ϕ. This operator presents a one-to-one relationship. The $\mathcal{F}$ operator essentially produces a new $spdf$, where the multiplication term p_i denotes the probability at the point $f(w_i)$. $i \epsilon X$ indicates all samples which belong to the $spdf$ of X.

4 Yield Estimation

Yield estimation requires an accurate shape for the density function so that the density can be integrated. But since f is a non-linear function, the samples may no longer be situated at fixed distances. Hence the sample and weight propagation step should be followed by another binning process.

4.1 Sample Grouping

Yield estimation requires an integral under the probability density. Hence, the shape of the density needs to be known. To convert the FDPP samples into a density, spline interpolation is used on the samples. Interpolation requires equidistant samples. Hence, a re-bin operator needs to be implemented to separate out the samples evenly. Samples that fall in any particular bin will be approximated by a single sample at the center of the corresponding bin, with the height of this sample being the sum of the samples that fall into this bin. Binning of samples makes interpolation of the samples possible; without the binning process, neighboring samples with largely differing sample probabilities would cause a great amount of distortion over the density. The re-bin operator is defined as:

$$\phi'(X) = \mathcal{R}_M(\phi(X)) \tag{7}$$

where the prime indicates a new $spdf$. The re-bin operator can be implemented using the following algorithm:

Algorithm Implementing the $\mathcal{R}$ Operator:

```
Find maximum and minimum values $w_i$ within propagated samples
Divide this range into M bins
For each bin
    Place a quantizing sample at the center of the bin with a height $p_i$ equal to the sum of all samples within the bin
```

The complexity of this algorithm is $O(N.M)$ where N is the number of initial samples and M is the number of final samples. Since these numbers are rather small, the run-time is highly efficient. The domain and range of the re-bin function is an $spdf$. The result of $\mathcal{R}_M$ can be written as:

$$\phi'(X) = \sum_i p_i \delta(x - w_i) \tag{8}$$

where $p_i = \sum_{j:w_j \epsilon b_i} p_j$ and w_i's can be written according to the new bin locations. M corresponds to the new number of bins, as the re-bin operator can bin the samples into a different number of bins than the one used in the $spdf$ that this operation is applied to. The re-bin operation is illustrated in Figures 1 to 3.

4.2 Sample Skipping and Zero Padding

After re-binning, some of the new bins might be empty as samples initially falling into this bin might have been pulled in with the samples on the neighboring bins. If spline interpolation was applied to get the final density, the density would end up having dips that are not supposed to be there in reality. We provide an algorithm for sample skipping below. On line 3, the much greater sign $>>$ is introduced for the case when one of the neighboring samples is also close to 0 and the other neighbor is large. Spline interpolation would end up acquiring a dip with a value lower than 0 if such a sample is removed. This would reveal a contradiction to the definition of a density function, each point of which must be greater than or equal to zero.

As the density needs to die out to 0 on both ends, zero-padding is employed to the samples whenever the last samples is not 0. This is because we know that the density should die out to 0 so that the underlying area is restricted by 1.

Sample Skipping Algorithm:

```
[1] For i=2 to number of FDPP samples-1,
[2]     If (Probability of i'th sample is 0
[3]         & Probability of at least one of neighboring samples >>0)
[4]             Remove sample
```

The complexity of this algorithm is $O(N)$, where N is the number of input samples.

4.3 Yield Estimation Algorithm

Yield is given by:

$$yield = 1 - \int_a^b pdf(X)dx \tag{9}$$

where a and b are lower and upper limits respectively for the pass/fail criteria. After a P-point interpolation, the estimate density is given by the set of point pairs :

$$\{(w_i, p_i) \quad s.t. \quad i \epsilon 1..P\} \tag{10}$$

and hence the percentage yield can be calculated by:

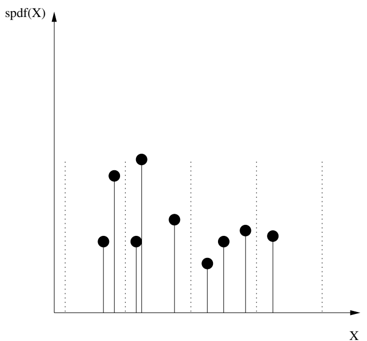

Figure 1. An illustrative example of samples generated through FDPP before the re-binning operation

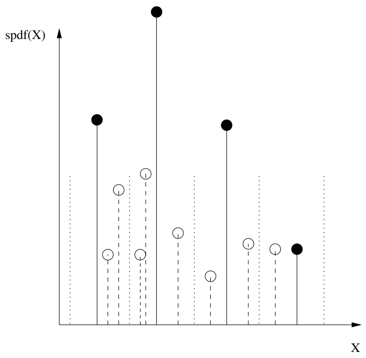

Figure 2. Samples added to bin centers during the re-binning process

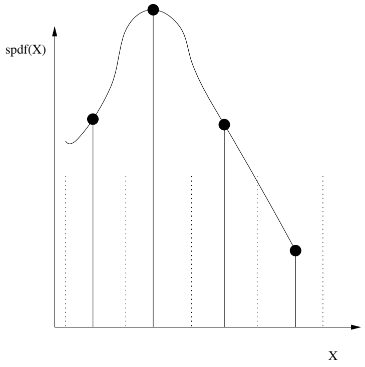

Figure 3. Interpolation over samples

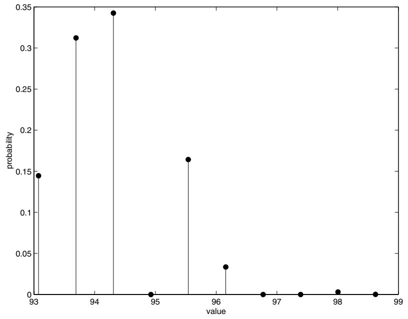

Figure 4. Samples before sample skipping and zero padding

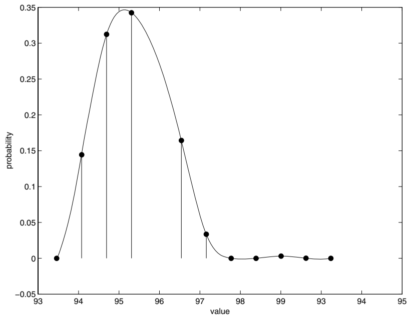

Figure 5. Spline interpolation after sample skipping and zero padding

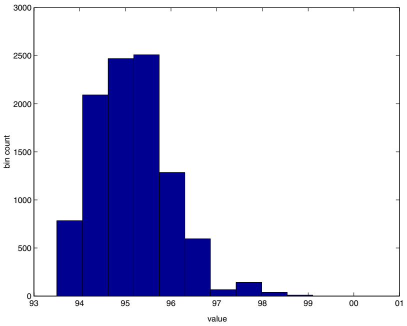

Figure 6. Reference 10k Monte Carlo run

$$yield(X) = (1 - \sum_{i \epsilon 1..P \& w_i < a \& w_i > b} p_i) * 100 \qquad (11)$$

The complexity of this algorithm is $O(N)$, where N is the number of input samples.[2]

5 Experimental Results

Estimation of parameter densities at the behavioral block level can be useful when process variations need to be considered. Most high-level analog blocks exhibit probability densities for their output parameters instead of a constant nominal value. For example, the loop filter in Figure 8 acquires a probability density for its output frequency as a result of process variations, as does the charge pump for the charged current. The deviation from a Gaussian density for both of these examples has been confirmed in the experiments we have conducted.

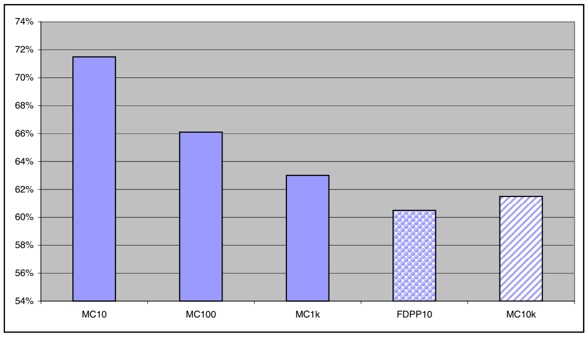

Figure 7. Yield estimate comparisons

Phase-locked loops, being analog components, are an important part of almost all large-scale digital and mixed-signal systems. Their gate-accurate simulation takes very long time as transient analysis has to be done for many cycles. That's why, in this paper, we have chosen this block as our benchmark. The behavioral simulator described in [22] is used for the analysis.

[2] Note that re-binning consumes the bulk of the runtime, whereas the implemented post processing algorithms consume only a fraction of this time for all the samples.

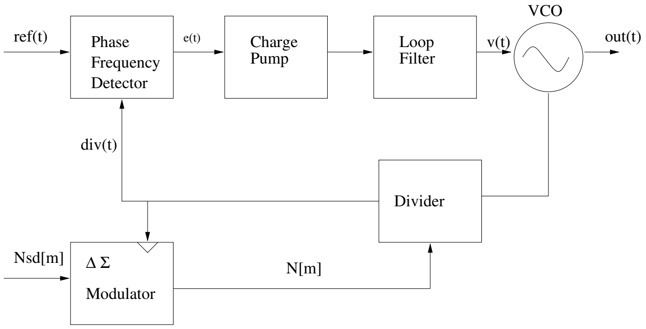

Figure 8. PLL architecture used in experiments

In support of our proposal, we have conducted a set of experiments centered largely on a PLL architecture as presented in [22]. This architecture is shown in Figure 8. Matlab R12 was employed for mathematical evaluations, by incorporating a Perl script to calculate the rms jitter at the voltage controlled oscillator output [11]. For each sample, 3E+5 simulations steps are run with a sample period of 1E-11s. Each sample took about 10 seconds to simulate on a 1.44GHz desktop. The output current of the charge pump is assumed to vary with a probability density function with a σ of 10%. These densities stem from mismatch or process variations in the charge pump and are to be extracted using transistor level accurate simulations.

We have started FDPP with a *forward* operator on the density samples of the charge pump output current. 10 samples have been used according to the error bounds given in [5][3]. The application of the forward operation is followed by a *re-bin* operation which re-bins the 10 samples. Sample skipping resulted in eliminating one of the samples and zero padding added two zeros on both ends.

Monte Carlo runs for the same example consume long run-times for the same accuracy. Since we do not know the true density for the output jitter, a Monte Carlo run with $10k$ samples is assumed to give the reference true density. In reality, a Monte Carlo run with a sample number approaching infinity in the limit would give the true density. But since this is practically unattainable, 10k runs are assumed to give the true density.

Sample skipping and zero padding steps are shown on the experimental data in Figures 4-5. Since the x axis values are many decimals accurate, they are simplified such that 1.221293E-11 seconds is represented as 93, 1.221294E-11 seconds is represented as 94 and so on. The fourth sample is skipped and two 0's are padded to each end of the density. The >> operation is used to avoid deletion of samples such as the seventh. Figure 5 also shows the spline interpolation after this step. y axis shows the values before normalization, which will satisfy the condition that the sum of all points add up to 1. A comparison with Monte Carlo with 10k samples is shown in Figure 6. The deviation from Gaussian due to nonlinear nature of the system is captured as a bulgy tail in both methods.

A fair comparison is particularly paid attention whereby Monte Carlo samples are first binned into 10 bins, the same number of samples used as in FDPP. Then, a 500 point spline interpolation is used for all methods. To calculate yield, the rms jitter is assumed to be limited by lower and upper bounds as the passing criteria.[4] The absolute values of yield are plot in Figure 7. MC and $FDPP$ in the figure correspond to Monte Carlo and the proposed method respectively, and the appended number is the number of samples used in the corresponding method. The reference simulation is indicated with a diagonal pattern and the proposed one with a pattern different than others. Here, FDPP gives the closest match to the reference 10k-sample Monte Carlo run with an error rate of 1.44%. Not only the error rate of 10-sample FDPP is better than a 1k-sample Monte Carlo, which has an error rate of 2.87%, but also the run-time for 10-sample FDPP is 100 times less than a 1k-sample Monte Carlo and FDPP has better accuracy for yield.[5] Monte Carlo runs with 10 and 100 samples do not even come close to the reference yield estimate, leaving the 1k sample Monte Carlo as the only rival. Furthermore, FDPP gives a lower yield than the 10k-sample Monte Carlo as shown in Figure 7. This and the decreasing nature of yield as Monte Carlo samples are increased suggests that 10k-sample Monte Carlo may not be giving the true yield and that FDPP may possibly be more accurate than the 10k-sample Monte Carlo (perhaps as accurate as a 100k run).

6 Summary

Forward discrete probability propagation (FDPP) has been suggested as an alternative to the Monte Carlo method. FDPP requires a far smaller number of samples than Monte Carlo using weight propagation and deterministic sampling. We introduce herein FDPP for behavioral level fast and accurate yield estimation. Behavioral simulations are increasingly popular as large circuits with possible feedback loops take a long time to simulate. Re-binning, sample skipping and zero padding have been used to implement the accurate yield estimation system that uses deterministic sampling and probability weight propagation as the basis. FDPP, using a recent PLL structure as an example, is shown to be highly efficient for behavioral level yield estimation and simulation by showing improved accuracy and 100 times run-time speed-up as compared to a standard 1000 sample Monte Carlo.

[3] The sample number selection in the re-bin operator is not automated but is given to the user as a choice and a theoretic limit because of similar issues in most mesh generation algorithms. Accordingly, one way to validate the correct selection is to repeat the calculations with finer bin widths and compare that the results are same.

[4] Although the lower the jitter the better, the lower limit is assumed to be provided as a sanity check as it could be the case that jitter values smaller than the lower limit could indicate faults at other blocks of the circuit.

[5] The run-time is approximately proportional to the number of samples.

References

[1] P. Arcioni, M. Bozzi, M. Bressan, G. Conciauro, and L. Perregrini. Fast optimization, tolerance analysis, and yield estimation of h-/e-plane waveguide components with irregular shapes. *IEEE Transactions on Microwave Theory and Techniques*, 52(1):319–328, 2004.

[2] B. Balzs, Szirmay-Kalos, and A. Gyrgy. Weighted importance sampling in shooting algorithms. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 177–184, New York, NY, USA, 2003. ACM Press.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[4] A. Demir, E. Liu, A. L. Sangiovanni-Vincentelli, and I. Vassiliou. Behavioral simulation techniques for phase/delay-locked systems. In *Custom Integrated Circuits Conference*, pages 453–456, 1994.

[5] L. Devroye and G. Lugosi. Bin width selection in multivariate histograms by the combinatorial method. *Test*, 13:129–145, 2004.

[6] P. G. Drennan and C. C. McAndrew. Understanding mosfet mismatch for analog design. *IEEE JSSC*, 38(3):450–456, 2003.

[7] P. M. P. et. al. *Handbook of Applied Optimization*. Oxfor University Press, 2002.

[8] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, 2003.

[9] A. Graupner, W. Schwarz, and R. Schuffny. Statistical analysis of analog structures through variance calculation. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 49:1071–1078, 2002.

[10] C. Guardiani, S. Saxena, P. McNamara, P. Schumaker, and D. Coder. An asymptotically constant, linearly bounded methodology for the statistical simulation of analog circuits including component mismatch effects. In *Design Automation Conference*, pages 15–18, 2000.

[11] A. Hajimiri and T. Lee. A general theory of phase noise in electrical oscillators. *IEEE JSSC*, 33(2):179–194, 1998.

[12] M. Hinz, I. Konenkamp, and E.-H. Horneber. Behavioral modeling and simulation of phase-locked loops for rf front ends. In *Midwest Symposium on Circuits and Systems*, pages 194–197, 2000.

[13] S. Inohira, T. Shinmi, M. Nagata, T. Toyabe, and K. Iida. A statistical model including parameter matching for analog integrated circuits simulation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 4:621–628, 1985.

[14] V. Khandelwal and A. Srivastava. A general framework for accurate statistical timing analysis considering correlations. In *Design Automation Conference*, pages 89–94, 2005.

[15] F. Liu, J. Flomenberg, D. Yasaratne, and S. Ozev. Hierarchical variance analysis for analog circuits based on graph modelling and correlation loop tracing. In *Design Automation and Test in Europe*, pages 126–131, 2005.

[16] W.-L. Loh. On latin-hypercube sampling. *Annals of Statistics*, 24:2058–2080, 5.

[17] D. J. C. Mackay. *Introduction to Monte Carlo Methods*. Kluwer Academic Press, 1997.

[18] C. Michael and M. Ismail. Statistical modeling of device mismatch for analog mos integrated circuits. *IEEE JSSC*, 27(2):154–166, 1992.

[19] A. Mounir, A. Mostafa, and M. Fikry. Automatic behavioural model calibration for efficient pll system verification. In *Design Automation and Test in Europe*, pages 280–285, 2003.

[20] S. Ozev and A. Orailoglu. Boosting the accuracy of analog test coverage computation through statistical tolerance analysis. In *VLSI Test Symposium*, pages 213–219, 2002.

[21] M. Pelgrom, A. Duinmaijer, and A. Welbers. Matching properties of mos transistors. *IEEE JSSC*, 24(5):1433–1439, 1989.

[22] M. Perrott. Fast and accurate behavioral simulation of fractional-n synthesizers and other pll/dll circuits. In *Design Automation Conference*, pages 498–503, 2002.

[23] M. Rencher. Analog statistical simulation. In *Custom Integrated Circuits Conference*, pages 29.2/1–29.2/4, 1991.

[24] K. Saab, N. Ben-Hamida, and B. Kaminska. Parametric fault simulation and test vector generation. In *Design Automation and Test in Europe*, pages 650–656, 2000.

[25] M. Seeger. Gaussian processes for machine learning.

[26] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director. Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance. In *Design Automation Conference*, pages 535–540, 2005.

[27] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition*, pages 605–612, 2003.

[28] J. Swidzinski, D. Alexander, M. Qu, and M. Styblinski. A systematic approach to statistical simulation of complex analog integrated circuits. In *Int. Workshop on Statistical Metrology*, pages 86–89, 1997.

[29] R. O. Topaloglu. Monte carlo-alternative probabilistic simulations for analog systems. In *Int. Symposium on Quality Electronic Design*, 2006.

[30] R. O. Topaloglu and A. Orailoglu. On mismatch in the deep sub-micron era-from physics to circuits. In *ASP-DAC*, pages 62–67, 2004.

[31] R. O. Topaloglu and A. Orailoglu. Forward discrete probability propagation method for device performance characterization under process variations. In *ASP-DAC*, pages 220–223, 2005.

[32] H. Yoon, P. Variyam, A. Chatterjee, and N. Nagi. Hierarchical statistical inference model for specification based testing of analog circuits. In *VLSI Test Symposium*, pages 145–150, 1999.

[33] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C.-P. Chen. Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model. In *Design Automation Conference*, pages 83–88, 2005.

[34] S. Zhang, V. Wason, and K. Banerjee. A probabilistic framework to estimate full-chip subthreshold leakage power distribution considering within-die and die-to-die p-t-v variations. In *Low Power Electronics and Design*, pages 156–161, 2004.