Statistical Run-Time Verification of Analog Circuits in Presence of Noise and Process Variation

Rajeev Narayanan, *Member IEEE,* Ibtissem Seghair, *Student Member IEEE,* Mohamed H. Zaki, *Member IEEE,* and Sofiène Tahar, *Senior Member, IEEE*

***Abstract*—Noise and Process variation present a practical limit on the performance of analog circuits. This paper proposes a methodology for modeling and verification of analog designs in the presence of shot noise, thermal noise, and process variations. The idea is to use stochastic differential equations (SDE) to model noise in additive and multiplicative form and then combine process variation due to 0.18μm technology in a statistical runtime verification environment. The efficiency of MonteCarlo and Bootstrap statistical techniques are compared for a Colpitts oscillator and a phase locked loop (PLL) based frequency synthesizer circuit.**

***Index Terms*—Analog Designs, Noise, Process Variation, Run-Time Verification, Statistical Techniques, Stochastic Differential Equations**

I. Introduction

Over the last decade, high performance System-on-Chip (SoC) [14] has played a pivotal role in the growth of consumer electronics, embedded systems, and computing servers. As the complexity of a SoC continues to escalate, analog designs have started to exhibit more of their stochastic behavior, where ensuring the correctness of such designs under all circumstances usually becomes impractically expensive. Additional effects due to noise and process variations have also influenced the quality and yield of those circuits [19].

Computer aided design (CAD) tools for analog circuits have seen a tremendous growth in recent times, yet it still lags its digital counterpart in many aspects such as, abstraction, automation, and IP reusability. The analog design flow has remained essentially the same for the past twenty years with the schematic capture of the individual blocks, followed by verification through multiple simulation in order to estimate various noise and process variation metrics. Unfortunately, with only little automation under its belt, analog circuit verification may lead to weeks/months of labor intensive circuit simulation to validate the design for optimal performance. Therefore, given such a widening gap between the complexity of analog designs and the maturity of CAD tools [16], a combination of traditional and new modeling/verification strategies is needed to mitigate several effects such as noise [19] and process variation [3]. One of the ways to accomplish this is by looking at modeling techniques at higher level of abstraction using hardware description languages (HDLs), so that verification for the whole design with noise and process variation can be automated and performed much faster. This speed-up, however, does not come without a price. The first cost is the accuracy of the behavioral model against the actual transistor-level designs. Secondly, the model has to account for physical device (threshold voltage, leakage current, etc.), functional (noise, jitter) and environment (temperature) constraints.

Noise is a random phenomena which origin has been studied by many researchers for decades. The sources of noise could be due to unwanted interaction between various design blocks (e.g., cross-talk) or it could be inherited from the circuit elements (e.g., thermal, shot and flicker) [9]. Also, noise can be either in additive or multiplicative form and since, for a given circuit, it is difficult to predict their exact form, it is necessary to account for both types of noise during analysis. *"Can we eliminate noise?"* is the question that has to be answered. With proper layout and shielding techniques in a design, interference noise can be nullified [9]. On the other hand, the inheritance noise can be reduced and cannot be eliminated completely. The quantification of such noise relies on the kind of measurement used between the noisy and the ideal signal. Important noise metrics is to derive the power spectral density (PSD) in terms of a measurable quantity such as the signal-to-noise ratio (SNR) and the noise figure (NF). SNR is a measure used to determine the quality of a signal that is corrupted by noise, and NF is a quality measure of SNR degradation.

From a verification perspective, the qualitative estimation in terms of SNR, NF has to be complemented with quantitative analysis by monitoring the circuit current and voltages. Using quantitative measurement, the circuit is evaluated either for one simulation trace or multiple simulation traces. To do so, designers need noise models in time domain to check the functional behavior (current, voltage) at run-time, and then compare it to the expected result. As the PSD for a thermal noise is Gaussian in nature it is necessary to have models in time domain that have the same distribution. One such time domain model is the Wiener process [6]. As the PSD of a shot noise take a Poisson distribution, time domain models based on Poisson white shot noise (PWSN) [7] allow Poisson distribution for the random pulses and a Gaussian distribution for its amplitude.

In conjunction with the effect of noise, designs become more challenging when the fabrication steps for a circuit are considered [22]. For process variation, designers use a combination of *Worst-Case*, *Monte-Carlo* or *Mismatch* [17] analysis for analog circuits. The worst-case analysis method

Rajeev Narayanan is with the Department of Electrical and Computer Engineering, State University of New York, New Paltz, New York, USA. Email: rajeev@newpaltz.edu

Ibtissem Seghair, Mohamed H. Zaki and Sofiène Tahar are with the Department of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, Canada. Email:{saghar, mzaki, tahar}@ece.concordia.ca

for analog circuits incorporates design models with pessimistic process corner. This worst-case variation is determined in the foundry design document, and the values are derived from certain parameter distribution. For instance, the process corners are constructed to maximize/minimize one specific performance of the device (e.g., speed, power, area, etc.) and can provide faster results [17]. However, the worst-case analysis may increase the overall design effort and cost.

The MonteCarlo method takes into account a predefined distribution (usually normal distribution) of the device parameters due to process variation. Unlike worst-case that targets for single device performance, MonteCarlo methods use a repeated simulation technique for multiple device performance [17]. In the end, it provides a statistical estimate of the analysis with a certain confidence level, but at the cost of simulation run-time.

In summary, right from the circuit level through behavioral level [16], current industrial designs rely heavily on simulation techniques to verify analog circuits. For noise and process variation, designers use a combination of statistical modeling and MonteCarlo simulation in order to achieve a good fit between the measured and the extracted values. But, such analysis can be time consuming and can have memory problems [16]. Alternatively, the behavior of the design can be captured as a purely mathematical model and then device variation could be integrated during analysis. By doing so, it leads to time domain statistical monitoring of the behavior rather than qualitative estimation of the circuit noise metrics. For such a method it is necessary that models retain both the functional and stochastic behavior. We therefore, adopt stochastic differential equations (SDE) [6] as an analog noise model in this paper.

This paper tries to answer some of the shortcoming of the above approaches by modeling an analog circuit with noise using SDE in additive and multiplicative form. Then, process variation due to 0.18μm technology [1] are integrated with the run-time verification environment for monitoring the statistical properties of the circuit. Statistical run-time verification combines hypothesis testing [23] and Monte-Carlo/Bootstrap simulation for monitoring the statistical behavior in an analog circuit. The efficiency of MonteCarlo and Bootstrap statistical techniques are compared for Colpitts oscillator and a phase locked loop (PLL) based frequency synthesizer circuit.

II. Related Work

Lately, Synopsys has introduced a tool, HSPICE RF [12] implementing SDE techniques to make a direct prediction on the statistical behavior of analog circuits. The results include the usual deterministic transient analysis waveforms, and also its time-varying root-mean square (RMS) behavior. A similar commercial tool [4] enables circuit designers to efficiently perform SPICE-accurate device noise analysis on complex non-periodic analog blocks. However, noise analysis at circuit level often rely on manual (visual) inspection of the simulation results, thereby slowing down the design and verification process.

In contrast, in [24] the authors have numerically evaluated an electronic oscillator based on a new physical description of thermal noise. The method involves combining the non-equilibrium statistical mechanics with the SDE based Langevin approach. But, the method fails to neglect non-linearity and also ignores the process variation. In a similar numerical approach [21], the authors have used SDEs to model the design with thermal noise in additive form and then combine device variation due to the *0.18μm* fabrication process in an assertion based run-time verification environment. Runtime verification is a technique for monitoring whether an execution of the design model violates the design specifications (properties). Unfortunately, such single simulation trace monitoring cannot provide us the confidence due to stochastic nature of noise and process variation. A similar work [26], demonstrates a runtime verification methodology for statistical properties of AMS designs. The approach combines system of recurrence (SRE) equations with MonteCarlo simulation and hypothesis testing to verify statistical property. The above approach fails to address issues related to noise and process variation, also SRE models are not accurate.

Other less known methods developed in the context of formal verification (of AMS designs) apply mathematical expressions and formal reasoning to prove correctness of a design [15]. For instance, statistical based model checking [10] has been successfully used to verify the saturation property in a simple analog circuit, such as a third-order $\Delta\Sigma$ modulator. Such model checking techniques are still in their infancy and can easily run into state-space explosion for complex circuits. By contrast, theorem proving can deliver the highest level of assurance for verification. In [20], the authors presented a closed-form solution based formal verification method for proving properties in analog circuits with noise and process variation. Unfortunately, not all analog circuits have closed-form solutions and hence such approach has limited application and is not accurate enough to gain confidence on the results. Also, due to the complexity of generating formal models and the computational overhead of the algorithms used, success has been mainly limited by scalability issues.

III. Preliminaries

A. Stochastic Differential Equation (SDE)

An SDE is an ordinary differential equation (ODE) with a stochastic process that can model unpredictable real-life behavior of any continuous system [6]. The random term in an SDE can be purely additive or it may multiply by some deterministic term. For example, let us consider a tunnel diode oscillator circuit shown in Figure 1.

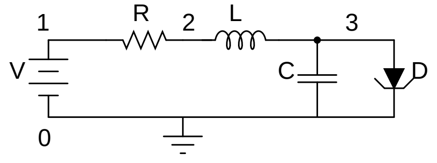


Fig. 1. Tunnel Diode Oscillator

The current through the resistor and inductor I and the voltage across the capacitor V_C can be described by

$$\begin{aligned} \dot{V_C} &= \frac{1}{C}(-I_d(V_C)+I) \\ \dot{I} &= \frac{1}{L}(-V_C - \frac{1}{G}I + V) \end{aligned} \tag{1}$$

where $I_d(V_C)$ describes the non-linear tunnel diode behavior. If we consider thermal noise in the passive elements (*R, L*) and shot noise in the diode *D*, a reasonable mathematical interpretation of the randomness for Equation 1 can be described as

$$\begin{aligned} \dot{V_C} &= \overbrace{\frac{1}{C}(-I_d(V_C)+I_L)}^{A^1} + \alpha\xi_1(t) \\ \dot{I} &= \overbrace{\frac{1}{L}(-V_C - RI + V)}^{A^2} + \alpha\xi_2(t) + \zeta(t) \end{aligned} \tag{2}$$

where $\sum_{k=1}^{2} \alpha\xi_k$ represents the thermal noise model for the passive elements with certain amplitude α and $\zeta(t)$ represents Poisson white shot noise (PWSN) that has random pulses, which occurrence is based on Poisson distribution. The strengths of the pulses take a white noise (Gaussian) distributed independent values as shown in Figure 2. In general,

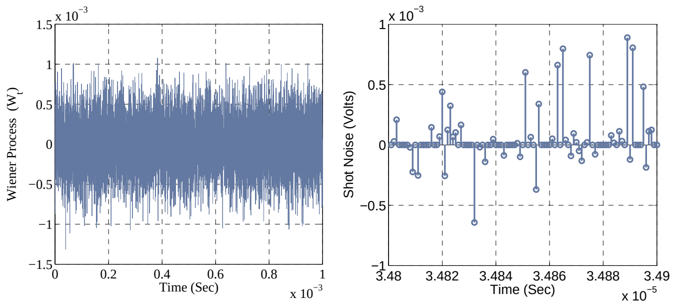


Fig. 2. Poisson White Shot Noise (PWSN)

the probability that a random sequence of *k* pulses occurs in the interval $(0,t)$ is given by

$$Prob\{n(t)=k\} = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \tag{3}$$

In SDE terminology, Equation (2) can take two forms [6]: *Itô* or *Stratonovich* representing differential and integral forms respectively. Since, the amplitude of shot noise is based on Gaussian distribution, the $\zeta(t)$ and ξ_t in Equation 2 can be considered to be the path-wise derivative of Brownian motion (or Wiener Process). If we set dB_t and dB_{st} to represent such process for thermal and shot noise respectively, then Equation (2) can be rewritten as

$$\begin{aligned} dV_C &= \frac{1}{C}(-I_d(V_C)+I)dt + \alpha dW_{1_t} \\ dI &= \frac{1}{L}(-V_C - RI + V)dt + \alpha dW_{2_t} + dW_{st} \end{aligned} \tag{4}$$

To solve Equation 4 classical calculus cannot handle stochastic process, and hence we need special mathematical interpretation in the form of stochastic calculus to solve the equations involving Brownian motion [6]. Stochastic calculus uses the concept of expectation and *Itô* isometry to solve SDEs. Expectation determines the behavior of any system in the absence of randomness and hence it is easy to conclude that the expectation of any random process (Brownian or Wiener) is zero. As Brownian motion cannot be solved using definite integral, the goal of *Itô* isometry is to replace the Brownian motion dB_s by a deterministic term *ds* for solving SDEs. In contrast, there is not always a closed form solution for SDEs, hence researchers have looked for solving them numerically. The methods based on numerical analysis are reported in [18], which involve discrete time approximation in a finite time interval over the sample paths.

Based on the simplest *Euler-Maruyama* time discretization approach [18], Equation (4) can be rewritten as

$$\begin{aligned} V_{C_{n+1}} &= V_{C_n} + \frac{\Delta_n}{C}(-I_d(V_{C_n}) + I_n) + \alpha\Delta W_{1n} \\ I_{n+1} &= I_n + \frac{\Delta_n}{L}(-V_{C_n} - RI_n + V) + \alpha\Delta W_{2_n} + \Delta W_{sn} \end{aligned} \tag{5}$$

where for time step τ,

$$\Delta_n = \tau_{n+1} - \tau_n; \;\; \Delta W_n = \Delta W_{sn} = W_{\tau n+1} - W_{\tau n} \tag{6}$$

for *n=0,1,2$\cdots$N-1*; and for maximum *N* simulation steps.

In general, any SDE that takes a form as in Equation (5), is suited to represent the additive noise behavior in an analog circuit. Higher order numerical approximation such as the *Milstein* method [18] uses multiple stochastic integrals in terms of several Wiener processes and can be used to model the multiplicative noise behavior. To better understand the *Milstein* method of noise model, let us consider the tunnel diode oscillator shown in Figure 1. If we consider noise to exists in multiplicative form, then, rewriting Equation (2) in matrix form, we get

$$dY = \begin{pmatrix} dV_C \\ dI \end{pmatrix} = \begin{pmatrix} A^1 \\ A^2 \end{pmatrix} dt + \begin{pmatrix} I \\ V_C \end{pmatrix} dW_t^1 + \begin{pmatrix} V_C \\ I \end{pmatrix} dW_t^2 \tag{7}$$

with

$$b^1 = \begin{pmatrix} b^{(1,1)} \\ b^{(2,1)} \end{pmatrix} = \begin{pmatrix} I \\ V_C \end{pmatrix}; b^2 = \begin{pmatrix} b^{(1,2)} \\ b^{(2,2)} \end{pmatrix} = \begin{pmatrix} V_C \\ I \end{pmatrix};$$

A general Milstein approximation for the SDE can be written as

$$Y_{n+1}^k = Y_n^k + a^k\Delta_n + \sum_{j=1}^{M} b^{j,k}\Delta W^j + \sum_{j_1,j_2=1}^{M} L^{j_1} b^{k,j_2} I(j_1,j_2) \tag{8}$$

Applying Equation (8) to Equation 7, we get

$$\begin{aligned} \begin{pmatrix} V_{C_{n+1}} \\ I_{n+1} \end{pmatrix} &= \begin{pmatrix} V_{C_n} \\ I_n \end{pmatrix} + \begin{pmatrix} A^1 \\ A^2 \end{pmatrix}\Delta_n + \begin{pmatrix} I_n \\ V_{C_n} \end{pmatrix}\Delta W_n^1 \\ &+ \begin{pmatrix} V_{Cn} \\ I_n \end{pmatrix}\Delta W_n^2 + \sum_{j_1,j_2=1}^{2} L^{j_1} b^{k,j_2} I(j_1,j_2) \end{aligned} \tag{9}$$

where L^j is the partial differential operator as defined by $L^j = \sum_{k=1}^{N} b^{k,j}\frac{\partial}{\partial x^k}$ and $I(j_1,j_2)$ is the *Ito* integral [18]. Expanding

L^j for $j = 1, 2$ we get

$$\begin{aligned} L^1 &= \sum_{k=1}^{2} b^{k,1}\frac{\partial}{\partial x^k} = b^{1,1}\frac{\partial}{\partial x^1} + b^{2,1}\frac{\partial}{\partial x^2} \\ L^2 &= \sum_{k=1}^{2} b^{k,2}\frac{\partial}{\partial x^k} = b^{1,2}\frac{\partial}{\partial x^1} + b^{2,2}\frac{\partial}{\partial x^2} \end{aligned} \tag{10}$$

Hence the final Milstein numerical approximation for Equation (7) is given by

$$\begin{aligned} &\begin{pmatrix} V_{C_{n+1}} \\ I_{n+1} \end{pmatrix} = \begin{pmatrix} V_{Cn} \\ I_n \end{pmatrix} + \begin{pmatrix} \frac{1}{C}(-I_d(V_{Cn}) + I_n) \\ \frac{1}{L}(-V_{Cn} - \frac{1}{G}I_n + V) \end{pmatrix}\Delta_n \\ &+ \begin{pmatrix} I_n \\ V_{Cn} \end{pmatrix}\Delta W_n^1 + \begin{pmatrix} V_{Cn} \\ I_n \end{pmatrix}\Delta W_n^2 + \begin{pmatrix} V_{Cn} \\ I_n \end{pmatrix} I(1,1) \\ &+ \begin{pmatrix} I_n \\ V_{Cn} \end{pmatrix} I(1,2) + \begin{pmatrix} I_n \\ V_{Cn} \end{pmatrix} I(2,1) + \begin{pmatrix} V_{Cn} \\ I_n \end{pmatrix} I(2,2) \end{aligned} \tag{11}$$

where the *Ito* integral $I(j_1, j_2)$ can be expressed as [18]

$$I(1,1) = I(2,2) = \int_{t_n}^{t_{n+1}}\int_{t_n}^{t} dW_s^{j_1} dW_t^{j_2} = \frac{1}{2}\left((\Delta W_n^{j_1})^2 - \Delta_n\right)$$

$$I(1,2) = I(2,1) = \int_{t_n}^{t_{n+1}}\int_{t_n}^{t} dW_s^{j_1} dW_t^{j_2} = \frac{1}{2}\left(\Delta W_n^{j_1}\Delta W_n^{j_2}\right)$$

B. Statistical Hypothesis Testing

Hypothesis testing [23] is the use of statistics to make decision about acceptance or rejection of some statements based on the data from a random sample, meaning, to determine the probability that a given hypothesis is true. Hypothesis testing in general, has two parts: *Null hypothesis*, denoted by H_0, which is what we want to test (e.g., $jitterperiod \leq$ *3.2 ns*) and *Alternative hypothesis*, denoted by H_1, which is what we want to test against the null hypothesis (e.g., $jitterperiod$ $>$ *3.2 ns*). If we reject H_0, then the decision to accept H_1 is made. The conclusion is drawn with certain probability of error (α and β) along with specific confidence level.

The quantification of error can be made by measuring the probability of accepting/rejecting H_0 when it is actually true/false, respectively. Usually, α, also called the *significance level*, denotes the probability of rejecting H_0 when it is actually true (Type I error) and β denotes the probability of accepting H_0 when it is actually false (Type II error). For instance, $\alpha = 0.05$ and $\alpha = 0.01$ refer to the confidence levels of 95% and 99%, respectively.

The choice to accept or reject is determined by the direction with which the null hypothesis is proved to be true or false. This direction is decided based on a one-tailed test (*upper* or *lower*) or a *two-tailed* test as shown in Figure 3.

The *upper tail* represents the rejection region for the case where a large value of the test statistic provides evidence for rejecting H_0. On the other hand, a *lower* tail distribution is used if only a small value of the test statistic shows proof of H_0 rejection [13].

The *bounded hypothesis testing* [13] also called the *two-tailed* test is determined by a bounded region $[x_1, x_2]$, such

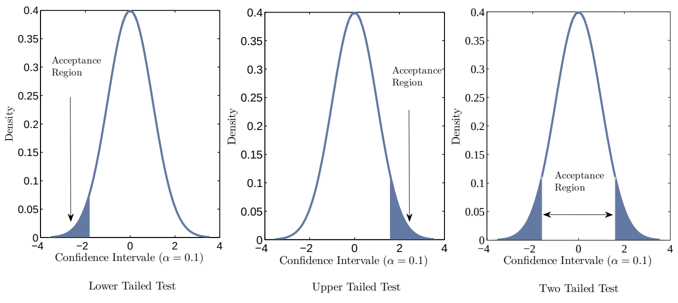


Fig. 3. Accept/Reject Regions for Hypothesis Testing

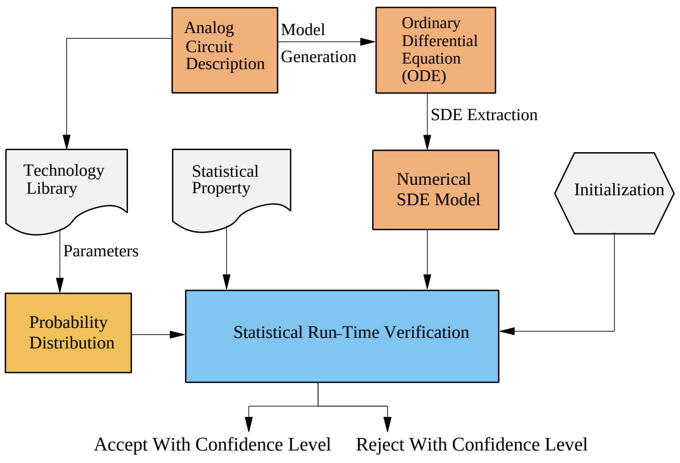
that such that H_0 satisfies the following:

$$H_0 : P(x_1 < X < x_2) = P(X < x_2) - P(X < x_1) = 1 - \alpha \tag{12}$$

In any of the above hypothesis testing measures, if the observed sample data T_{obs} over a given interval is within some critical region, then we reject the null hypothesis H_0, else we accept H_0 as shown by the shaded region in Figure 3.

IV. Proposed Methodology

Figure 4 shows the overall statistical run-time verification methodology. Thereafter, given an analog design described



Analog Circuit Description
Model Generation
Ordinary Differential Equation (ODE)
SDE Extraction
Technology Library
Statistical Property
Numerical SDE Model
Initialization
Parameters
Probability Distribution
Statistical Run-Time Verification
Accept With Confidence Level
Reject With Confidence Level

Fig. 4. Statistical Run-Time Verification Methodology

as a system of ODEs, the idea is to generate SDEs that expresses the noise behavior. For the case of the circuits that do not have closed form solution, the approach is to numerically approximate the SDE's based on Euler-Maruyama technique as described in Section III. For process variation, the technology vendors create a library of devices with different corners [22] that characterize the device in terms of power, speed, etc. This allows the designers to choose from a range of devices based on the application and requirements. For a 0.18μm process, different circuit parameters are derived using Gaussian distribution with a known $\pm 3\sigma$ deviation.

For environment constraints, this may include the amplitude of the noise, initial conditions of the circuit current and voltages. The SDE model, process variation, and the environment constraints are evaluated using MonteCarlo/Bootstrap statistical technique in a MATLAB simulation environment.

Statistical run-time verification combines hypothesis testing and resampling methods for monitoring the statistical behavior in an analog circuit. The basic idea behind the resampling methods is to simulate the SDE model and sample them in order to calculate the desired statistics for a given confidence level δ.

Figure 5 shows the methodology for the statistical simulation procedure based on hypothesis testing. The statistical property, is expressed as a null hypothesis H_0, while the alternative hypothesis H_1 becomes the counterexample naturally. For the given numerical SDE model and the specified tail test, MonteCarlo or Bootstrap monitoring is carried out based on the given confidence level δ and the calculated significance level α. The statistical property is verified if the null hypothesis H_0 is accepted, else, the monitor reports the violation of the property. In all cases, an error margin ϵ is generated as shown in Figure 5.

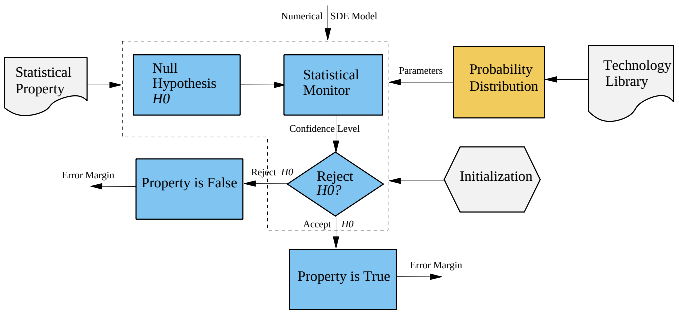


Fig. 5. Statistical Hypothesis Testing

A. MonteCarlo Algorithm

The MonteCarlo method [23] refers to a technique of solving problems using random variables. It is widely used to investigate statistical problems such as inference statistics of a given population of interest. The basic idea behind the MonteCarlo method is to sample the given population model for M trials and then calculate the desired statistics (such as mean, median, variance, skewness, etc.). To apply MonteCarlo based hypothesis testing, it is necessary that the distribution of the sampling population is known in advance [11].

One of the most important components of MonteCarlo simulation is the use of deterministic algorithm to generate a normally distributed unbiased pseudo random number. These random numbers are then used to sample the true population of interest, in this case the analog circuit output. In general, there is no theory that governs the number of trials in MonteCarlo simulation. However, a trade off exists between those numbers and the simulation run-times. The higher confidence can be gained by choosing a larger number of trials, but at the cost of run-times [23].

The detailed procedure for Monte-Carlo hypothesis testing for an analog circuit is illustrated in Algorithm 1, where *output_vector* denotes the observed output of an analog circuit with noise and process variation. M represents the number of MonteCarlo trials, α a chosen significant level and *type_test* represents the type of test to be performed (*upper*, *lower*, or *two-tailed*).

Algorithm 1 MonteCarlo Based Hypothesis Testing:

```
Require: $output\_vector$, $M$, $\alpha$, $type\_test$
1:  $V \leftarrow output\_vector$
2:  $N \leftarrow length(V)$
3:  $mu \leftarrow mean(V)$
4:  $sig \leftarrow standard\_error(V)$
5:  for $i \leftarrow 1$ to $M$ do
6:      $r \leftarrow random\_number\_generator(N)$
7:      $MC_{sample} \leftarrow sig * r + mu$
8:      $T_{obs}(i) \leftarrow \frac{mean(MC\_sample) - mu}{sig}$
9:  end for
10: while $type\_test =$ "$upper\ tail\ test$" do
11:     $critical\_value = quantile(1 - \alpha)$
12:     if $critical\_value \geq T_{obs}$ then
13:         $Accept\ H_0$
14:     else
15:         $Reject\ H_0$
16:     end if
17: end while
18: while $type\_test =$ "$lower\ tail\ test$" do
19:     $critical\_value = quantile(\alpha)$
20:     if $critical\_value \leq T_{obs}$ then
21:         $Accept\ H_0$
22:     else
23:         $Reject\ H_0$
24:     end if
25: end while
26: while $type\_test =$ "$two\ tail\ test$" do
27:     $critical\_value\_low = quantile(\frac{\alpha}{2})$
28:     $critical\_value\_up = quantile(\frac{1-\alpha}{2})$
29:     if $critical\_value\_up \leq T_{obs} \,\|\, critical\_value\_low \geq T_{obs}$
        then
30:         $Reject\ H_0$
31:     else
32:         $Accept\ H_0$
33:     end if
34: end while
```

The initialization steps (lines 1-4) are followed by the computation of the standard score to determine the observed analog output T_{obs} (loop between lines 5 and 9). This calculation is done with certain standard error margin as defined by

$$E = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{N(N-1)} \tag{13}$$

where $x = (x_1, x_2, ..., x_N)$ represents the MonteCarlo sample, $\bar{x}$ the pseudo random sample mean, and E defines the standard error of the population under the hypothesis that H_0 is true. The next step is to compute the critical value in order to specify the rejection region (alternative hypothesis). Depending on the type of the test, the *quantile* procedure [23] (lines 11,19, 27 and 28) can be used to determine the critical value. With the estimated critical value, the rejection region under the assumption of H_0 being true can be determined for each of the tail test as defined in Table I.

TABLE I
REJECTION REGION FOR DIFFERENT TAIL TEST

Tail Test	Rejection Region
Upper	$[(100 \times (1 - \alpha))\%, +\infty]$
Lower	$[-\infty, (100 \times \alpha)\%]$
Two-Tailed	$[-\infty, (100 \times \frac{\alpha}{2})\%] \cup [(100 \times (1 - \frac{\alpha}{2}))\%, +\infty]$

If the observed value T_{obs} is greater than the critical value, the null hypothesis H_0 is rejected as described in Algorithm 1 (lines 10-32). The MonteCarlo algorithm is efficient only if the probability distribution of the sample is known in advance. In most cases, a normal distribution is assumed for the samples, which may not be true always for analog circuits influenced by noise and process variation. To overcome the above drawbacks, Bootstrap algorithm have been developed in the context of finance to reason about the statistical inference of the population when the assumption of the underlying distribution is violated or unknown.

B. Bootstrap Algorithm

The Bootstrap [5] is a general purpose method for estimating a statistical property without making any assumptions about the underlying distribution of the population [8]. In this sense, it is considered as a non parametric technique or distribution free. The basic idea behind the Bootstrap technique can be described as follows [8]: *"Given a random sample of* N *data* $X = (x_1, x_2, ..., x_N)$ *from an unspecified distribution F, the maximum likelihood estimator of F is the distribution that puts an equal point probability of* $\frac{1}{N}$ *to each data of X".*

The detailed procedure for Bootstrap based hypothesis testing for an analog circuit is illustrated in Algorithm 2, where $output\_vector$ denotes the observed output of an analog circuit with noise and process variation. B represents the number of Bootstrap samples, α a chosen significant level and $type\_test$ represents the type of test to be performed (*upper*, *lower*, or *two-tailed*).

The first step is to draw randomly B samples with replacement from the simulated circuit output of size N (line 4). This is followed by test statistic estimation for each bootstrap replication in order to measure discrepancy between the data and H_0. The results are then in T_{boot} as a vector (line 5). The *quantile* procedure is then used to compute the critical value by type of test:

- The $1 - \alpha$ quantiles of the empirical distribution for an upper tail test as shown in line 9.
- The α quantile of the empirical distribution for a lower tail test as mentioned in line 17.
- The $\frac{\alpha}{2}$ and $(1 - \frac{\alpha}{2})$ quantile of the empirical distribution for a two sided test as given in lines 25 and 26, respectively.

Once the critical value is determined, a decision regarding the violation of a statistical property is done using hypothesis testing (lines 16-31). For instance, in the case of a lower tail test (lines 16-23), if the observed value T_{boot} is lower than the computed critical value, then we reject H_0, meaning the statistical property has failed.

Algorithm 2 Bootstrap Based Hypothesis Testing:

```
Require: $output\_vector$, $B$, $\alpha$, $type\_test$
1: $V \leftarrow output\_vector$
2: $N \leftarrow length(V)$
3: for $i \leftarrow 1$ to $B$ do
4:     $rep \leftarrow Resample\_Bootstrap(V, N)$
5:     $T_{boot}(i) \leftarrow Compute\_test\_statistic(rep)$
6: end for
7: $T_{sorted} \leftarrow sort\_ascending\_order(T_{boot})$
8: while $type\_test =$ "$upper\ tail\ test$" do
9:     $critical\_value = T_{sorted}(B * (1 - \alpha))$
10:    if $critical\_value \geq T_{obs}$ then
11:        $Accept\ H_0$
12:    else
13:        $Reject\ H_0$
14:    end if
15: end while
16: while $type\_test =$ "$lower\ tail\ test$" do
17:    $critical\_value = T_{sorted}(B * \alpha)$
18:    if $critical\_value \leq T_{obs}$ then
19:        $Accept\ H_0$
20:    else
21:        $Reject\ H_0$
22:    end if
23: end while
24: while $type\_test =$ "$two\ tail\ test$" do
25:    $critical\_value\_low = T_{sorted}(B * \frac{\alpha}{2})$
26:    $critical\_value\_up = T_{sorted}(B * \frac{1-\alpha}{2})$
27:    if $critical\_value\_up \leq T_{obs} \parallel critical\_value\_low \geq T_{obs}$
       then
28:        $Reject\ H_0$
29:    else
30:        $Accept\ H_0$
31:    end if
32: end while
```

V. Experimental Results

To illustrate the efficiency of the proposed methodology, the approach is illustrated on a tunnel diode, Colpitts oscillator and PLL circuits. The effect of thermal noise on passive components and shot noise on the transistors has been analyzed in additive and multiplicative form. The first step in noise analysis is to identify and incorporate the sources of noise as a stochastic process in the form of SDE. Thermal and Shot noise are defined based on the method described in Section III. The experiment results are derived separately for additive and multiplicative noise in a statistical based MATLAB simulation environment on a Windows Vista OS (AMD Dual-Core, 4GB RAM) machine.

A. Colpitts Oscillator

The circuit diagram for an MOS transistor based Colpitts oscillator is shown in Figure 6. For the correct choice of component values, the circuit will oscillate due to the bias current and negative resistance of the passive tank. The frequency of oscillation is determined by L, C_1 and C_2.

From the small-signal representation, the simplified system of equations that describe the behavior of the Colpitts oscillator can be derived as

$$\begin{aligned}\dot{V}_{C1} &= \frac{1.2 - (V_{C1} + V_{C2})}{RC} + \frac{I_L}{C} - \frac{I_{ds}}{C} \\ \dot{V}_{C2} &= \frac{1.2 - (V_{C1} + V_{C2})}{RC} + \frac{I_L}{C} - \frac{I_{ss}}{C} \qquad (14) \\ \dot{I}_L &= \frac{1.2 - (V_{C1} + V_{C2})}{L}\end{aligned}$$

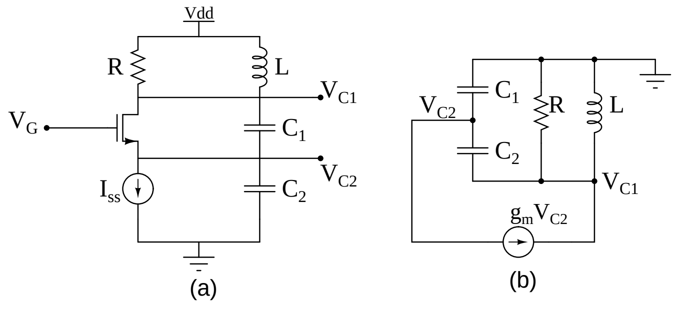


Fig. 6. (a) Colpitts Oscillator (b) Small-Signal Model

where for $V = V_{C1} + V_{C2}$,

$$I_{ds} = \begin{cases} 0 \text{ if } V_{C2} > 0.3 \\ K\frac{W}{L}((0.3 - V_{C2})(V_{C1}) - 0.5(V_{C1})^2) \text{ if } V < 0.3 \\ K\frac{W}{L}(0.3 - V_{C2})^2 \text{ if } V \geq 0.3 \end{cases}$$

If thermal noise is considered for the passive components and shot noise for the MOS transistor, then Equation 14 can be extended to SDE form as given below

$$\begin{aligned} \dot{V}_{C1} &= \frac{1.2 - (V_{C1} + V_{C2})}{RC} + \frac{I_L}{C} - \frac{I_{ds}}{C} + \sum_{k=1}^{2} \alpha\xi_k(t) \\ \dot{V}_{C2} &= \frac{1.2 - (V_{C1} + V_{C2})}{RC} + \frac{I_L}{C} - \frac{I_{ss}}{C} + \sum_{k=1}^{2} \alpha\xi_k(t) \\ \dot{I}_L &= \frac{1.2 - (V_{C1} + V_{C2})}{L} + \alpha\xi_3(t) + \zeta(t) \end{aligned} \quad (15)$$

where $\sum_{k=1}^{3} \alpha\xi_k$ represents the thermal noise model for the passive elements with certain amplitude α, and $\zeta(t)$ represents Poisson white shot noise (PWSN) due to random carrier motion (current) in the MOS transistor. The above SDE model is numerically approximated using Euler/Milstein technique and simulated with process variation in a MATLAB simulation environment.

The deterministic property that was verified in [21] is *"Whether for the given parameters and initial conditions, the inductor current is within a certain bound or not for oscillation?"* The analysis was done only for thermal noise in additive form, and this paper addresses the issue of shot noise and thermal noise in additive and multiplicative form.

Based on the simulation results, the authors in [21] were able to verify the bounded property using assertion based verification. However, this single simulation trace verification environment does not provide enough insight to gain higher confidence on the oscillator circuit.

For statistical run-time verification one would be interested to know *"Whether for the given confidence level α, process variation and MonteCarlo/Bootstrap trials, what is the probability of acceptance and rejection of oscillation for multiple trajectories $Trac$?"* For the oscillator, the current through the inductor I_L should be bounded within $[-0.004, 0.004]$. As a

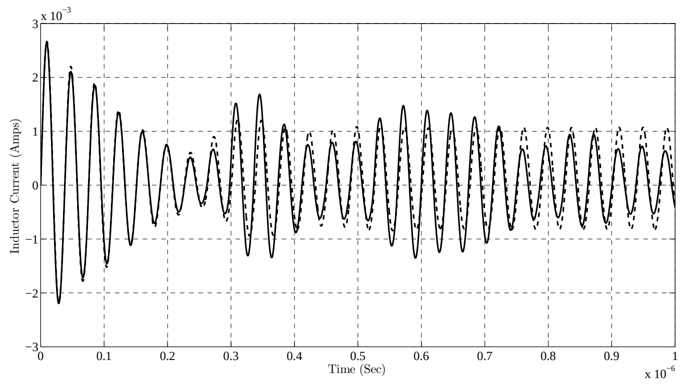


Fig. 7. Simulation Result of Colpitts Oscillator

result, the null hypothesis H_0 and the alternative hypothesis H_1 of this property can be expressed as

$$\begin{aligned} H_0 &: -0.004 \leq I_L \leq 0.004; \\ H_1 &: I_L > 0.004 \;\|\; I_L < -0.004; \end{aligned} \quad (16)$$

Both the MonteCarlo and Bootstrap experiments were conducted for the confidence level $\alpha = 0.05$ for different tail test, with shot/thermal noise in the circuit elements, and with the circuit parameter generation using a normally distributed process variation model. The results are summarized in Table II. $Trac$ in the table represents the total number of colpitts oscillator circuit that have been evaluated for various MonteCarlo/Bootstrap trials (M and B) and with independent noise and technology constraints.

From Table II, it can be noted that, irrespective of the tail test, that the MonteCarlo technique exhibits false violation (shaded column). In the MonteCarlo method, first the mean of the output is derived, followed by the creation of different sampling points based on normal distribution with a known standard deviation. Such a process may sometimes lead to a value that is out of bound with the observed value thereby, giving rise to false violation. In the absence of process variation the number of failures remains reasonably low for both additive and multiplicative noise (columns 7-10). It can also be seen that process variation (columns 11-14) in all the passive components has a greater effect on the acceptance/rejection of the circuit and with the combined effect of noise and process variation (columns 15-18) the hypothesis testing exhibited considerable failure of the statistical property.

The effect of process variation and noise on the statistical results can be visualized using shmoo plots as shown in Figure 8. Though the process variation is considered in all circuit elements, the figure is shown only for the process variation in capacitor with respect to the resistor. At each capacitance value, the resistor is swept based on the values generated using normal distribution and the hypothesis testing result is analyzed by writing '1' for acceptance or '0' for rejection.

In addition, the number of MonteCarlo trials has an adverse effect on the outcome of the acceptance, but, a large Bootstrap trial made little impact on the outcome. This is because, the data generation process for the Bootstrap does not assume any distribution of the output data. For this experiment, the

TABLE II
STATISTICAL RUNTIME VERIFICATION RESULTS FOR COLPITTS OSCILLATOR.

Additive Noise (TRAC = 200, M = Number of MonteCarlo Trials, B = Number of Bootstrap Trials, P.V = Process Variation, A = Accept, R = Reject)																	
M=	**Tail**	**No Shot/Thermal Noise & P.V**				**Shot/Thermal Noise Only**				**P.V Only**				**Shot/Thermal Noise & P.V**			
		MonteCarlo		**BootStrap**		**MonteCarlo**		**BootStrap**		**MonteCarlo**		**BootStrap**		**MonteCarlo**		**BootStrap**	
B=	**Test**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**
1000	Lower	192	8	200	0	179	21	191	9	180	20	187	13	147	53	184	16
	Upper	194	6	200	0	173	27	194	6	178	22	189	11	153	47	182	18
	Two	190	10	200	0	169	31	189	11	175	25	181	19	138	62	180	20
10000	Lower	188	12	200	0	164	36	193	7	172	28	185	15	151	49	179	21
	Upper	187	13	200	0	163	37	194	6	180	20	187	13	143	57	184	16
	Two	183	17	200	0	164	36	191	9	175	25	181	19	132	68	179	21
50000	Lower	188	12	200	0	159	41	189	11	177	23	182	18	122	78	174	26
	Upper	187	13	200	0	161	39	188	12	174	26	181	19	127	73	179	21
	Two	181	19	200	0	159	41	183	17	171	29	179	21	120	80	173	27
Multiplicative Noise																	
1000	Lower	194	6	200	0	188	12	196	4	180	20	187	13	189	11	194	6
	Upper	197	3	200	0	190	10	193	6	178	22	189	11	185	15	196	4
	Two	193	7	200	0	185	15	194	6	175	25	181	19	177	23	189	11
10000	Lower	199	1	200	0	181	19	191	9	172	28	185	15	180	29	184	16
	Upper	197	3	200	0	183	17	193	7	180	20	187	13	177	23	181	19
	Two	194	6	200	0	188	12	196	4	175	25	181	19	179	21	184	16
50000	Lower	197	3	200	0	187	13	191	9	177	23	182	18	181	19	188	12
	Upper	197	3	200	0	182	18	193	7	174	26	181	19	171	29	185	15
	Two	195	5	200	0	186	14	191	9	182	18	187	13	164	36	181	19

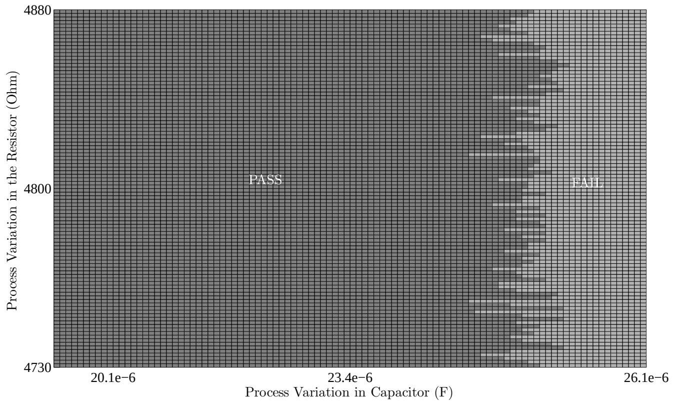


Fig. 8. Shmoo Plotting of Colpitts Oscillator Results.

worst case run-time for $M/B = 50000$ is around *5-6* hrs, which though is considerably less than a simulation done at the circuit level.

B. Band-Gap Reference Generator

For any biasing circuits, one of the most important performance issue is their dependence on temperature. The variation in temperature, noise and process variation attributes to the fractional change in the output voltage/current, thereby affecting the functionality of the design [16]. Figure 9 shows a BJT based reference generator biasing circuit, and the question is *"How does the variation of noise with respect to temperature affect the behavior of the circuit?"*

The output voltage is based on the summation of the voltage across the base-emitter (V_{BE}) and the reference voltage (V_T). The behavior of the above circuit can be described as

$$\frac{dV_O}{dT} = (\gamma - \beta)\frac{V_T}{T}\left(\frac{T_0 - T}{T}\right) \tag{17}$$

where V_T is the input voltage. If we consider a temperature varying shot noise process $\zeta(T)$ in the transistor, Equation 17, can be rewritten to incorporate randomness in additive and

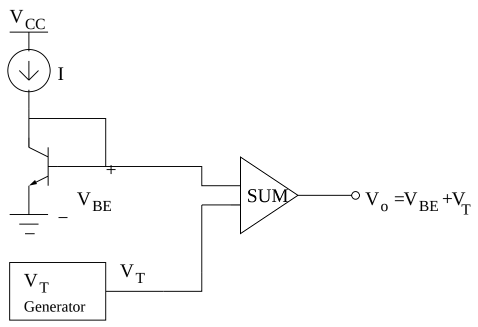


Fig. 9. Band Gap Reference Circuit [16].

multiplicative form as

$$\begin{aligned} \frac{dV_O}{dT} &= (\gamma - \beta)\frac{V_T}{T}\left(\frac{T_0 - T}{T}\right) + \zeta(T) \\ \frac{dV_O}{dT} &= (\gamma - \beta)\frac{V_T}{T}\left(\frac{T_0 - T}{T}\right) + \zeta(T)V_o(T) \end{aligned} \tag{18}$$

where γ and β are temperature independent constants [16] and T is the temperature. The shot noise process in Equation 18 is modeled with the technique described in Section III. For additive/multiplicative SDEs in the form of Equation 18, the Euler/Milstein scheme described in Section III is applied to generate the following numerical model:

$$\begin{aligned} V_{O_{n+1}} &= V_{O_n} + (\gamma - \beta)\frac{V_T}{T}\left(\frac{T_0 - T}{T}\right)\Delta_n + \Delta W_{sn} \\ V_{O_{n+1}} &= V_{O_n} + (\gamma - \beta)\frac{V_T}{T}\left(\frac{T_0 - T}{T}\right) + \Delta W_{sn} + \\ &\quad \frac{1}{2}\left((\Delta W_{sn})^2 - \Delta_n\right)V_{O_n} \end{aligned} \tag{19}$$

In the statistical analysis presented for the band-gap reference generator, since manufacturing techniques for BJT are different from those of CMOS, the effect of process variation for BJT's are not considered. The effect of the variation in the input voltage (V_T) is also studied.

The property of interest is: *"Whether for the given set of parameters and variation in temperature T, will the output voltage V_O be greater than a certain threshold voltage?"* The analysis was done only for thermal noise in additive form and does not provide a statistical estimate to gain confidence in the circuit verification.

For statistical run-time verification, it would be intriguing to extend the above property to *"Whether for the given confidence level α, M MonteCarlo trials and B Bootstrap trials, what is the probability of acceptance and rejection of the output voltage V_O for multiple trajectories $TRAC$ and varying input voltage V_T?"* Here, $TRAC$ is used to depict the band-gap reference circuit under different shot noise processes. For instance, if $TRAC = 100$, it represents "100" band-gap reference circuit models that have independent shot noise characteristics. For this case, the output voltage V_o should be bounded within $[V_o \geq 3.13mV]$ [16]. As a result, the null hypothesis H_0 and the alternative hypothesis H_1 of this property can be, respectively, expressed as

$$\begin{aligned} H_0 &: V_o \geq 3.13e{-3}; \\ H_1 &: V_o < 3.13e{-3}; \end{aligned} \tag{20}$$

Both the MonteCarlo and Bootstrap experiments were conducted for the confidence level $\delta = 0.95$ ($\alpha = 0.05$) for different tail tests, with shot noise only and with $TRAC = 200$. The results are summarized in Table III.

TABLE III
STATISTICAL RUNTIME VERIFICATION RESULTS FOR BAND-GAP REFERENCE GENERATOR.

Additive Noise (P.V = Process Variation, A = Accept, R = Reject)									
		No Noise				With Shot Noise and V_T			
M =	Tail	MonteCarlo		BootStrap		MonteCarlo		BootStrap	
B =	Test	A	R	A	R	A	R	A	R
1000	Lower	197	3	200	0	151	49	185	15
	Upper	193	7	200	0	159	41	187	13
	Two	191	9	200	0	147	53	176	24
10000	Lower	198	2	200	0	142	58	181	19
	Upper	199	1	200	0	151	49	182	18
	Two	193	7	200	0	152	48	178	22
50000	Lower	198	2	200	0	133	67	186	14
	Upper	198	2	200	0	127	83	190	10
	Two	197	3	200	0	121	89	179	21
Multiplicative Noise									
1000	Lower	200	0	200	0	181	19	199	1
	Upper	199	1	200	0	181	19	199	1
	Two	199	1	200	0	179	21	194	6
10000	Lower	199	1	200	0	188	12	199	1
	Upper	198	2	200	0	183	17	199	1
	Two	198	2	200	0	171	29	198	2
50000	Lower	198	2	200	0	193	7	197	3
	Upper	199	1	200	0	191	9	197	3
	Two	197	3	200	0	191	9	191	9

From Table III, it is interesting to see that even in the absence of noise (shaded column), irrespective of the tail test, the MonteCarlo technique produces some rejection. This is because of the MonteCarlo theoretical assumption of normal distribution of the output voltage V_o has resulted in this false rejection. As the Bootstrap technique does not take into account any assumption on the output distribution, it has 100% acceptance of the output in the absence of noise (column 5). In cases where the shot noise and V_T variation are considered (columns 7-10), it can be seen that Bootstrap has more acceptance than MonteCarlo because of their resampling method. Also, the effect of additive shot noise is greater than that of multiplicative noise. This is because, the amplitude of shot noise is in the order of millivolts and hence the effect is almost negligible. In addition, the higher the number of MonteCarlo/Bootstrap trials, the higher the rejection but at the cost of simulation run-time. The effect of shot noise and V_T on the statistical results can be pictured using shmoo plots as shown in Figure 10.

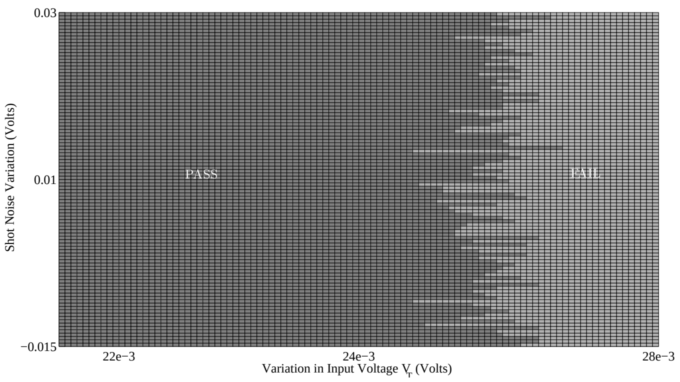


Fig. 10. Shmoo Plotting of Band-Gap Reference Generator Results.

For each V_T, the circuit is evaluated for different shot noise models by sweeping the amplitude appropriately. In the end, the total statistical result represents the number of passed/failed circuits that equal the total number of trajectories $TRAC$. The hypothesis testing result is analyzed by writing '1' for acceptance or '0' for rejection. The plot is shown for the MonteCarlo statistical results.

For this circuit, the worst case run-time for $M/B = 50000$ is around *3-4* hrs, which though is considerably less than the simulation done at the circuit level.

C. PLL Based Frequency Synthesizer

One of the major challenges for the verification of an AMS design, such as the PLL, is evaluating the uncertainties due to short-term frequency perturbation known as the *jitter* [2]. Jitter, a time-domain measure, is an unwanted contraction or expansion in the output oscillating signal from its ideal position. Such instability can result in wrong synchronization of the AMS design and eventually leads to the loss of data.

Recently, the authors in [26] have made use of the jitter models from [2] and have combined MonteCarlo and hypothesis testing to provide a statistical estimate of the jitter property specification. Unfortunately, they have failed to address the issue related to noise in the filter circuit and process variation associated with the circuit elements. Figure 11 shows a PLL based frequency synthesizer that is commonly seen in communication systems for clock generation and recovery. It is composed of two comparators, a phase/frequency detector, charge pump, analog filter, voltage controlled oscillator (VCO) and a divider.

The reference signal (*Ref_Signal*) at the input is a simple sinusoidal wave with frequency ω_0. The VCO output (*VCO_out*) is a cosine signal with frequency *N+1* times of the reference frequency, where *N* is determined by the frequency select signal (*Freq_Sel*). If the *Freq_Sel* is '0', the frequency of the reference input and VCO output will be the same or else the frequency will be divided accordingly based on the divider. The jitter model from [26] is used to address the issue of

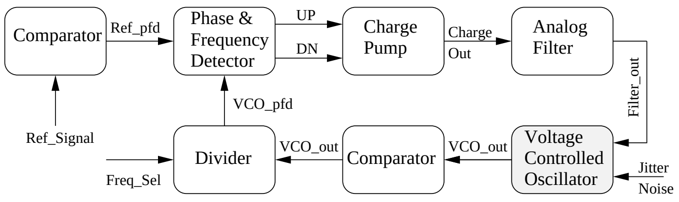


Fig. 11. PLL Based Frequency Synthesizer

period jitter associated with the VCO. The SDE representation of the filter behavior with additive and multiplicative thermal noise can be described as

$$\begin{aligned}\dot{F_o} &= \frac{1}{RC}(CP_o(t) - F_o(t)) + \sum_{k=1}^{2}\alpha\xi_k(t)\\ \dot{F_o} &= \frac{1}{RC}(CP_o(t) - F_o(t)) + \alpha\xi_k(t)F_o(t)\end{aligned} \tag{21}$$

where, F_o and CP_o represent the filter and charge-pump outputs, respectively, and R and C represent the resistor and capacitor components in the filter circuit. The next step is to apply the Euler/Milstein scheme described in Section III to generate the following numerical model:

$$\begin{aligned}F_{O_{n+1}} &= F_{O_n} + \left(\frac{\Delta_n}{RC}\right)(CP_o(n) - F_o(n)) + \alpha\Delta W_{sn}\\ F_{O_{n+1}} &= F_{O_n} + \left(\frac{\Delta_n}{RC}\right)(CP_o(n) - F_o(n)) + \alpha\Delta W_n +\\ &\quad \frac{1}{2}\left((\Delta W_n)^2 - \Delta_n\right)F_{O_n}\end{aligned} \tag{22}$$

The lock-time is an isolated property for all PLL based frequency synthesizers, i.e., once the PLL gets locked, the VCO will start oscillating until there is a change to the $Freq\_Sel$ signal. The method of verifying the “lock time” property is to check if the output of the low-pass filter has reached a new DC value within the lock time. In [25], the authors have verified the property without accounting for jitter in VCO and thermal noise in the filter as shown in Figure 12.

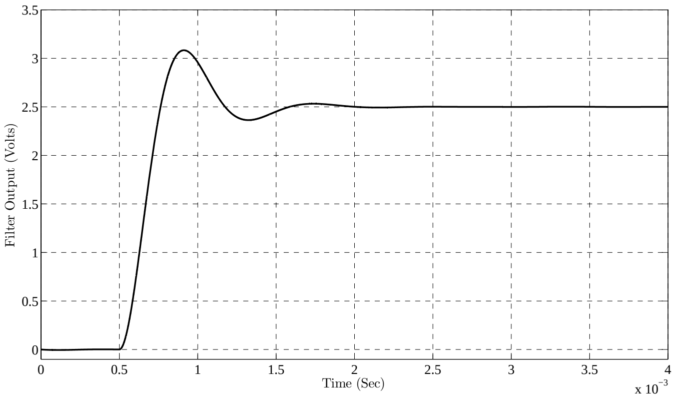


Fig. 12. PLL Lock-Time Verification

For statistical run-time verification, the lock-time property is *“For the given confidence level α, M Monte Carlo trials, B Bootstrap trials, and multiple trajectory $TRAC$, what is the probability of acceptance and rejection that the PLL meet the lock-time?”* In this case, the PLL has been design with a lock-time of 0.001sec [25]. Hence, the null hypothesis H_0 and the alternative hypothesis H_1 of this property can be, respectively, expressed as

$$\begin{aligned}H_0 &\quad : lock\_time \leq 0.001;\\ H_1 &\quad : lock\_time > 0.001;\end{aligned} \tag{23}$$

The simulation was carried out under the confidence interval $\alpha = 0.05$ and the jitter deviation as a normally distributed model. The results for “200” trajectories are summarized in Table IV. From Table IV, the combined jitter/thermal noise and process variation (columns 15-18) have substantially increased the PLL rejection, meaning the PLL has failed to lock. The presence of jitter/thermal noise alone has shown higher rejection. This is obvious that the effect of thermal noise is reflected through the filter output, and at the VCO input, which again adds up the jitter noise. As the VCO is considered to be very sensitive, even a slight change to the input may cause substantial changes to its output. In some cases, the failure to lock does not mean that the VCO is not oscillating but, the oscillation is either “ugly” or delayed.

It is also obvious that the case of process variation only (columns 11-14) for additive/multiplicative noise have resulted in the same number of rejection. This is because, both these cases have been simulated with the same process variation parameters. Also, both the additive/multiplicative noise have an equal influence on the overall rejections. This is because of the sensitive nature of the VCO, and even a millivolt drift in the input can cause substantial changes to the oscillation. A shmoo plot representing the pass/fail based on the lock-time is shown in Figure 13. For this circuit, the worst case run-time for $M/B = 50000$ is around *7-8* hrs, which is substantially high compared to previous circuits.

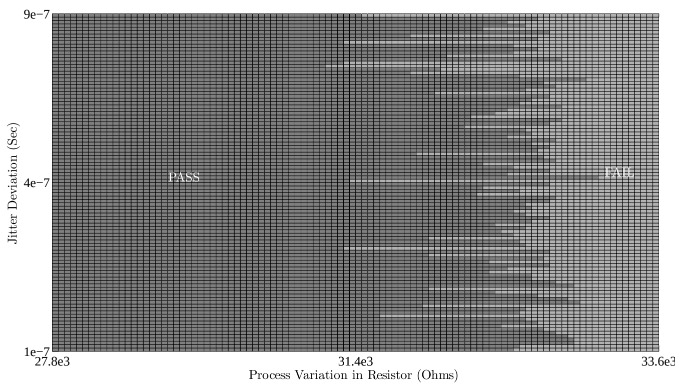


Fig. 13. Shmoo Plotting of PLL Results.

VI. Conclusion

This paper presented a methodology for the statistical verification of noise and process variation in analog circuits. The approach is based on thermal and shot noise modeling in additive and multiplicative form using stochastic differential equations, and then integrating the device variation due to the *0.18μm* fabrication process in an SDE based simulation framework. We have combined hypothesis testing with MonteCarlo/Bootstrap

TABLE IV
STATISTICAL RUNTIME VERIFICATION RESULTS FOR THE PLL LOCK-TIME PROPERTY.

Additive Noise (P.V = Process Variation, A = Accept, R = Reject)																	
		No Noise & P.V				**Noise Only**				**P.V Only**				**Noise & P.V Only**			
M=	**Tail**	**MonteCarlo**		**Bootstrap**		**MonteCarlo**		**Bootstrap**		**MonteCarlo**		**Bootstrap**		**MonteCarlo**		**Bootstrap**	
B=	**Test**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**	**A**	**R**
1000	Lower	152	48	200	0	129	71	181	19	180	20	187	13	137	63	174	26
	Upper	154	46	200	0	123	77	184	16	178	22	189	11	133	67	172	28
	Two	150	50	200	0	129	71	189	11	175	25	181	19	131	69	170	30
10000	Lower	168	32	200	0	124	76	173	27	172	28	185	15	121	79	169	31
	Upper	167	33	200	0	123	77	174	26	180	20	187	13	123	77	167	33
	Two	163	37	200	0	124	76	171	29	175	25	181	19	122	78	164	36
50000	Lower	148	52	200	0	119	81	177	23	177	23	182	18	112	88	164	36
	Upper	147	53	200	0	111	89	178	22	174	26	181	19	117	83	169	31
	Two	141	59	200	0	107	93	171	29	171	29	179	21	110	90	161	39
Multiplicative Noise																	
1000	Lower	154	46	200	0	128	72	189	11	180	20	187	13	139	61	177	26
	Upper	157	43	200	0	120	80	185	15	178	22	189	11	135	65	176	24
	Two	153	47	200	0	125	75	181	19	175	25	181	19	137	63	175	25
10000	Lower	159	41	200	0	121	79	175	25	172	28	185	15	130	70	174	26
	Upper	157	43	200	0	123	77	174	26	180	20	187	13	127	73	171	29
	Two	154	46	200	0	118	82	171	29	187	13	188	12	124	76	169	31
50000	Lower	147	53	200	0	107	93	171	29	175	25	181	19	111	89	168	32
	Upper	147	53	200	0	102	98	172	28	177	23	182	18	111	89	165	35
	Two	145	55	200	0	106	94	169	31	171	29	179	21	114	86	161	39

technique for verifying the statistical properties of the design. Our approach is illustrated on a Colpitts Oscillator and a PLL based frequency synthesizer circuit.

The statistical run-time verification method involves repeated simulation and can consume a lot of time and memory resources. The idea is to build a certain level of confidence in the circuit by analyzing the results from a large sample. The total number of samples depends on the values of MonteCarlo and Bootstrap trials (M and B), and it is obvious that the higher those values, the higher will be the confidence. As 100% confidence cannot be achieved using the run-time verification approach, it is necessary to complement them with other methods. Many enhancements can be made by combining run-time verification with formal methods to prove properties of a given circuit. The formalization and verification of AMS design is an interesting direction. The theories and infrastructure developed in the context of higher-order logic (HOL) have used random variables to verify the statistical properties of probabilistic systems. Due to the stochastic nature of noise, it would be intriguing to use HOL to develop an infrastructure for noise.

REFERENCES

[1] *0.18μm* CMOS Fabrication Process. http://www.tsmc.com, 2008.

[2] A. Demir, E. Liu, A. Vincentelli and I. Vassiliou. Behavioral Simulation Techniques for Phase/Delay Locked Systems. IEEE Custom Integrated Circuits Conference, pp. 453-456, 1994.

[3] B. Ankele, W. Hölzl and P. O'Leary. Enhanced MOS Parameter Extraction and SPICE Modeling for Mixed Signal Analogue and Digital Circuit Simulation. IEEE International Conference on Microelectronic Test Structures, pp. 133-137, 1989.

[4] Efficient Noise Analysis for Complex Non-Periodic Analog/RF Blocks, Berkeley Design Automation, Inc. 2009.

[5] B. Efron and R. Tibshirani. An Introduction to the Bootstrap, Chapman & Hall, 1993.

[6] B. Oksendal. Stochastic Differential Equations: An Introduction with Applications. Springer, 2000.

[7] C. Kim, E. K. Lee, P. Hänggi and P. Talkner. Numerical Method for Solving Stochastic Differential Equations with Poissonian White Shot Noise. Physical Review E. (76)1: 1-10, 2007.

[8] C. F. J. Wu. Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis. Annals of Statistics. (14)4: 1261-1295, 1986.

[9] D. A. Johns and K. Martin. Analog Integrated Circuit Design, Wiley, 1997.

[10] E. Clarke, A. Donze and A. Legay. Statistical Model Checking of Mixed-Analog Circuits with an Application to a Third-Order Delta-Sigma Modulator. Haifa Verification Conference, LNCS 5394, pp. 149-163, Springer, 2008.

[11] E. L. Lehmann and J. P. Romano. Testing Statistical Hypotheses, Springer, 2005.

[12] HSPICE User Guide: RF Analysis. http://www.synopsys.com, 2009.

[13] J. E. Freund. Modern Elementary Statistics. Prentice hall, 1984.

[14] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi and F. Sendig. Design of Mixed-signal Systems-on-a-chip, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, 19(12):1561-1571, 2000.

[15] M.H. Zaki, S. Tahar and G. Bois. Formal Verification of Analog and Mixed Signal Designs: A Survey. Microelectronics Journal, 39(12): 1395-1404, Elsevier, 2008.

[16] P. A. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer. Analysis and Design of Analog Integrator Circuits, Wiley, 2009.

[17] P. Bolcato and R. Poujois. A New Approach for Noise Simulation in Transient Analysis. IEEE International Symposium on Circuits and Systems, pp. 887-890, 1992.

[18] P. E. Kloden and E. Platen. Numerical Solution of Stochastic Differential Equations. Springer, 1995.

[19] P. Paper, M. Jamal Deen and O. Marinov. Noise in Advanced Electronic Devices and Circuits. AIP International Conference on Noise in Physical Systems and 1/f Fluctuations, 780: 3-12, 2005.

[20] R. Narayanan, B. Akbarpour, M.H. Zaki, S. Tahar and L. Paulson. Formal Verification of Analog/RF Circuits in the Presence of Noise and Process Variation. IEEE/ACM Design, Automation and Test in Europe, pp. 1309-1312, 2010.

[21] R. Narayanan, M.H. Zaki and S. Tahar. Using Stochastic Differential Equation for Verification of Noise in Analog/ RF Circuits. Journal of Electronic Testing: Theory and Applications, 26(1): 97-109, Springer, 2010.

[22] Y. Cheng. The Influence and Modeling of Process Variation and Device Mismatch on Analog/RF Circuit Design. IEEE International Conference on Devices, Circuits and Systems, pp. 1-8, 2002.

[23] W. L. Martinez and A. R. Martinez. Computational Statistics Handbook with MATLAB. Chapman & Hall/CRC, 2002.

[24] W. Mathis and T. Thiessen. On Noise Analysis of Oscillators Based on Statistical Mechanics, International Conference on Mixed Design of Integrated Circuits & Systems, pp. 472-477, 2009.

[25] Z. Wang, N. Abbasi, R. Narayanan, M.H. Zaki, G. Sammane and S. Tahar. Verification of Analog and Mixed Signal Designs using On-line Monitoring, IEEE Mixed-Signals, Sensors, System Test Workshop, pp. 1-8, 2009.

[26] Z. Wang, M.H. Zaki and S. Tahar. Statistical Runtime Verification of Analog and Mixed Signal Designs. IEEE International Conference on Signals, Circuits and Systems. pp. 1-6, 2009.