This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



# Accelerated and Reliable Analog Circuits Yield Analysis using SMT Solving Techniques

Ons Lahiouel, *Student Member, IEEE,* Mohamed H. Zaki, *Member, IEEE,* and Sofiène Tahar, *Senior Member, IEEE*

***Abstract*—Existing yield analysis methods are computationally expensive and generally encounter challenges with high-dimensional process parameters space. In this paper, we propose a new method for accelerated and reliable computation of parametric yield that combines the advantages of sparse regression and Satisfiability Modulo Theory (SMT) solving techniques, and avoids issues in both. The key idea is to characterize the failure regions as a collection of hyperrectangles in the parameters space. Towards this goal, the method constructs a sparse polynomial models based on adaptive LASSO (Least Absolute Shrinkage and Selection Operator) to find a low degree approximations of the circuit performances. A procedure inspired by statistical model checking is then introduced to assess the model accuracy. Given the constructed models, an SMT-based solving algorithm is employed to locate the failure hyperrectangles in the parameters space. The yield estimation is based on a geometric calculation of probabilistic volumes subtended by the located hyperrectangles. We demonstrate the effectiveness of our method using circuits that require expensive run-time simulation during yield evaluation. They include: an integrated ring oscillator, a 6T static RAM cell and a multi-stage fully-differential amplifier. Experimental results show that the proposed method is suitable for handling problems with tens of process parameters. Meanwhile, it can provide 5X∼2000X speed-up over Monte Carlo methods, when a high prediction accuracy is required.**

***Index Terms*—Analog circuit, Process variations, Yield, Surrogate model, Satisfiability Modulo Theory, Interval arithmetic.**

## I. Introduction

With aggressive technology scaling, process variation has become a major concern for today's analog integrated circuits (ICs), due to significantly increased circuit failures and parametric yield loss [1]. Indeed, analog IC components must be designed with sufficiently high yield in light of large-scale process variations (e.g., local mismatches caused by random doping fluctuations) [2]. For this reason, it becomes important to estimate the parametric yield both efficiently and accurately within the IC design flow [3].

The *standard* approach is the brute force Monte Carlo (MC) [4], which repeatedly draws samples from a predefined distribution of the process parameters and evaluates circuit performances with transistor-level SPICE simulation. MC has the advantages of simplicity and extremely general applicability. However, it can require very large numbers of expensive simulations for accurate yield estimation.

MC is inefficient especially for circuits with rare failure events (e.g., static random access memories (SRAMs)), because most of the samples fall into the feasible region, and only an extremely small fraction of samples are in the failure region [5]. It is then desirable that the simulation cost can be reduced. This is especially important if the yield estimation needs to be plugged into yield optimization flow since yield estimation needs to be done for many times.

To mitigate the inefficiency issue of MC method, various methodologies have been proposed in the past decade including advanced sampling techniques [6] [7] and boundary searching methods [8]. However, most of the existing approaches are either not general enough [7] or can be successfully applied to problems with a small number of process parameters, but, perform poorly with high-dimensional problems [8]. Given such limitations, a yield analysis method that tries to address the shortcoming of the above approaches is highly demanded.

Response surface-based surrogate modeling is a common approach to analyze the effects of process variations [9]. Accurate and not complex surrogate models can replace transistor-level simulation and significantly fasten the performances assessment and consequently the yield estimation. Though, the high-dimensional variational space and the strong nonlinearity of the performances models posed by advanced IC technologies lead to a large scale modeling problem that is hard to solve [9]. Furthermore, since the outcome of existing approaches is an approximation of the circuit response, weak guarantees on its accuracy can be provided.

This work is largely motivated by the powerful and new solving techniques in modern Satisfiability (SAT) Modulo Theory (SMT) [10] solvers. These solvers check the satisfiability of first-order formulas containing operations from various theories such as real numbers and integers. They are built upon a tight integration of modern Conflict-Driven Clause Learning (CDCL)-style SAT solving techniques with interval-based arithmetic constraint solving within an SMT framework. They are capable of handling constraints containing nonlinear functions over a very large number of variables [11], one inherent characteristics of analog circuits operation/performances models. Most importantly, they can be leveraged to exhaustively explore the search space of a constraint-satisfaction system, making them a potentially appealing choice for parameters space exploration strategies of analog circuits. Though, they should be properly employed.

In order to optimize the convergence of the yield estimation, our proposed work is based on two main directions:

- Focusing on the localization of only the failure regions in the parameters space: subsequently, the yield rate can be estimated by analytic computation of the probabilistic

The authors are with the Department of Electrical and Computer Engineering, Concordia University, Montréal, QC H4B 1R6, Canada (email:{lahiouel, mzaki, tahar}@ece.concordia.ca)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



hypervolume [12] of all failure regions. The challenge here is how to efficiently ensure a reliable characterization of the failure regions in a high dimensional space.
- Time-consuming MC simulations should be imperatively avoided so that the efficiency is further enhanced: this goal can be achieved through performance modeling. In this case, the fundamental challenge is not only the large-scale modeling problem but also verifying the model accuracy in any point of the parameters space with a minimum number of circuit simulations.

Towards these goals, we rethink the yield analysis from a completely different perspective. Principally, the key innovation of the proposed methodology is the formulation of the failure regions localization problem as a set of nonlinear constraints, that we solve using modern SMT solving techniques. To the best of our knowledge, this is the first work for yield estimation that is able to provide a guarantee on an exhaustive coverage of the circuit failure regions and hence tries to achieve reliable yield results.

The rest of the paper is organized as follows: Section II reviews existing techniques for analog circuit yield analysis. Section III details our yield estimation methodology. In Section IV, we provide experimental results for three analog circuits: an integrated ring oscillator, a 6T static RAM cell and a multi-stage fully-differential amplifier. In Section V, we present our conclusions and future work.

## II. Related Work

*State of the art* advanced MC methods for circuit yield analysis methods can be roughly divided into two categories: variance reduction techniques (e.g. Latin Hypercube Sampling (LHS) [13], Importance Sampling (IS) [7]) and low-discrepancy sequence-based methods (e.g. Quasi Monte Carlo (QMC) [14]). LHS partitions the range of each variable into non-overlapping intervals of equal probability and selects random values within each grid for every coordinate. By randomly combining the coordinate values, a set of latin hypercube are constructed. Because of this stratification technique, the LHS method is capable of providing variance reduction of the yield estimation. However, it does not work much better than the conventional MC, especially for some problems that are difficult to be decomposed into a sum of univariate functions [14].

The key idea of IS based-methods is to shift the original probability density function (PDF) of the process parameters towards the most likely failure region. They have achieved remarkable speed-up when applied for the yield analysis of circuits characterized by rare failure event. However, IS lacks generality as it is designed for circuits with very high/low yield rate. Furthermore, generating the shifted/distorted PDF is often challenging and circuit specific, since this depends on the actual distribution of the circuit performance which is unknown beforehand.

Another critical issue of IS is that the proposed (i.e., shifted) sampling distribution may not cover effectively all failed samples when the circuit presents multiple disjoint failure regions induced by conflicting or multiple specification requirements [5]. Besides the multiple specication requirements, high-dimensional process variables also induce the multiple failure regions since the process parameters may have opposite influence on the performance metrics [7]. Only few attempts have tackled the multiple failure regions case [15] [7]. In spite of that, while the method in [15] is applicable only to rare failure rate estimation in a very high-dimensional variation space (i.e., few hundreds), the authors in [7] cited that reduction techniques are required before applying their method for problems with more than 24 process parameters.

QMC is a popular approach that generates quasi-random numbers rather than purely-random samplings. It utilizes sample sets called Low Discrepancy Sequences (LDSs), in which deterministically generated samples are uniformly distributed on the parameter space [14]. QMC methods are able to provide improved integration error compared to LHS [14]. Yet, its convergence rate is found to be only asymptotically superior to MC only for circuit with a moderate number of process parameters [13].

Other existing methods try to construct a surface boundary which separates the success and failure regions [8]. Once the boundary is constructed, the yield can be obtained by computing the volume of the failure region without circuit simulation. For low dimensional problems, this method can be efficient. However, such methods cannot handle high-dimensional problems with no more than three process variables. Even when considering only three process parameters, searching the whole failure boundaries in the parameters space is extremely complicated. The high-dimensional analysis (18$\sim$24 process variables) is common and necessary in practical applications. Though, it makes the discrimination between failure and success regions by hypersurfaces very hard to achieve.

While above cited approaches present a variety of techniques to speed up and enhance the convergence of the traditional MC method, they fall short in addressing critical issues that can be summarized as follows:
- Optimally exploring the variational space that guarantees an acceptable accuracy and minimum computational time (i.e., a small number of transistor-level simulations).
- Scalability with respect to the process parameters size.
- Generality of application (i.e., handling different levels of yield rate, multiple performances metrics and multiple failure regions).

SMT solvers have been employed for formally verifying properties of analog circuits [16]. Recently, SMT solvers have been used for a primarily attempt to integrate formal techniques in the circuit sizing process. The goal was to avoid the instability of constrained optimization techniques in terms of convergence and local minima [17]. Different from this work, our solving strategy operates in the process parameters space for yield estimation purpose. Similar to our approach, the authors in [18] subdivide their SMT-based reachability analysis problem into subproblems that are solved in parallel in order to decrease the computational complexity. To do so, a analog circuit is decomposed into a set of smaller sub-circuits which decreases the number of variables involved in the SMT problems. Second, each SMT problem associated with a sub-circuit is further decomposed into a set of subproblems with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



less constraints to further improve the efficiency. However, the splitting strategy in [18] gives rise to significant complications. Indeed, the authors attempt to deal with the correlations between the partitioned sub-circuits. In contrast, in the proposed work, the partitioned subproblems are uncorrelated and the output of the SMT-based parameters space exploration is the union of all failure hyperrectangles located by all subproblems.

Sparse regression based on LASSO (Least Absolute Shrinkage and Selection Operator) [19] explores the fact that even though a large number of unknown model coefficients must be used to capture the high dimensional variation space, many of these model coefficients are close to zero, thereby rendering a unique sparse pattern. However, as discussed in [19], the LASSO shrinkage may not select the true coefficients values, as it causes the estimates of the non-zero coefficients to be biased towards zero, and in general they are not consistent [19]. One approach that overcomes this issue is to run the LASSO to identify the set of non-zero coefficients, and then fit an unrestricted linear model using least square regression as it has been proposed in [20]. Still, this solution is not always feasible, if the selected set is large.

## III. Yield Rate Estimation Methodology

Before presenting the proposed methodology, we briefly explain our main objective and define terms that will be used in the rest of the paper. Suppose that $p = [p_1, p_2, \ldots, p_l]$ is a $l$-dimensional continuous random variable modeling process variations. Such random variables include the variations of gate length $\Delta L$, oxide thickness $\Delta t_{ox}$ and threshold voltage $\Delta V_{th}$, etc., associated with each circuit device. Without loss of generality, we further assume that the random variables in the vector $p$ are mutually independent and follow a truncated normal distribution with $\pm 3\sigma$ and zero mean. We define the parameters (i.e., variation) space $P$ as the set of all possible combinations of the random variables. In general, the yield rate can be mathematically represented as:

$$\mathbf{Y}^* \quad = \quad 1 - P_f = 1 - \int_{\Omega} pdf(p)dp \qquad (1)$$

where $pdf(p)$ is the joint probability density function of $p$, $\Omega$ denotes the failure region, i.e., the region of the parameters space where the performances are not satisfied (can be a single region or multiple disjoint regions). We denote the integral in Equation 1 to be the probabilistic hypervolume of $\Omega$ [12]. Figure 1 is a geometrical illustration in two dimensions.

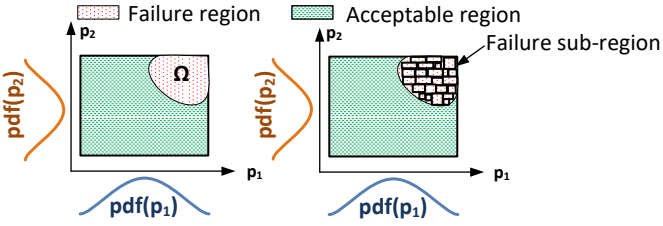


Figure 1: 2-D parameters space

In general, the multidimensional integral in Equation 1 cannot be directly computed since the failure region $\Omega$ usually establishes a complex nonlinear integration boundary. In our method, we propose to characterize $\Omega$ as a collection of high dimensional sub-regions (i.e., hyperrectangles). The probabilistic hypervolume of each sub-region is then evaluated and employed to estimate the total yield. Obviously, the accuracy of the yield estimation depends strongly on how well the sub-regions are approximated. In this paper, we will mainly focus on this characterization problem and develop novel algorithms to make it tractable and computationally efficient. The methodology in Figure 2 details our proposed approach.

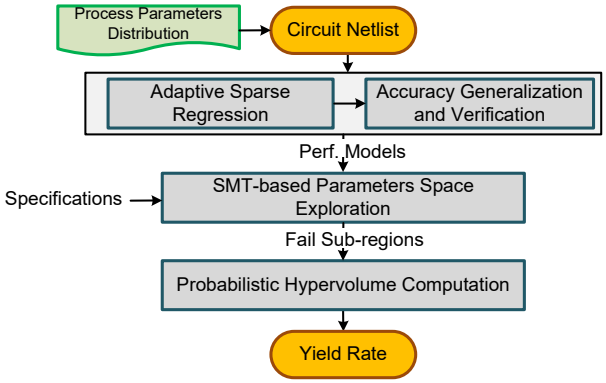


Figure 2: Yield estimation method overview

First, an adaptive sparse regression technique is applied to extract surrogate models of the circuit performances. In order to optimize the modeling step, a dimension reduction technique keeps the most significant process parameters. The proposed algorithm sorts the process parameters by weight assignment and prunes the unimportant parameters. Then, a low-degree and sparse polynomial model of each circuit performance is constructed in a stepwise fashion. The LASSO method assigns adaptive weights for penalizing the coefficients of the polynomial terms and yields a consistent estimate of the model coefficients. The model is iteratively built until the requirement in terms of accuracy is met. A procedure inspired by statistical model checking is then introduced to verify the model accuracy for a chosen confidence level. The resulting model can be viewed as a statistically guaranteed approximation of the circuit behavior. The subset of the circuit response space where each performance of interest does not meet the specification is conservatively characterized as a set of intervals. Based on the extracted models, SMT solving is not employed to compute the exact failure sub-regions in the parameters space. Instead, it is used to find only an over-approximation of them. The integration of interval arithmetics to remove the undesirable over-approximation, intelligently trades off between the computational cost and the conservativeness of SMT. A parallel exploration of the failure performance space allows the simultaneous finding of multiple satisfiable solutions and significantly speeds up the search process. Finally, the yield is estimated based on the probabilistic hypervolumes of the failure sub-regions.

### A. Adaptive Sparse Regression

In this section, we seek to produce an accurate surrogate model using polynomials with structured sparsity. The modeling technique should be performed with a minimum number of circuit simulations. Besides, the model must be computa-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



tionally efficient (i.e., not complex) and hence tractable for the subsequent SMT solving stage.

*1) Pre-sampling and dimension reduction:* The goal of pre-sampling is to approximately sketch the circuit behavior. We use the LHS method in the parameters space to generate a set of training samples. Given $n$ training samples, we denote $X = [x_1, x_2, \ldots, x_n]$ an $l \times n$ matrix, where each sample $x_i = [p_{i1}, p_{i2}, \ldots, p_{il}]$ is an $l$-dimensional vector. Next, transistor level SPICE simulation is performed to evaluate the performance metric using these samples. We denote $Y = [y_1, y_2, \ldots, y_n]$ the $n$ observations of the property, i.e., the value of the circuit response we seek to fit.

The parameters reduction maps the high dimensional process parameters space to a lower-dimensional space. In this work, we leverage the Regressional ReliefF (RReliefF) [21] algorithm to prune the process parameters and to select a smaller number of features. The algorithm uses samples based learning to assign a relevance weight to each parameter. Each feature weight reflects its ability to perturb the circuit response. The quality estimate ranges in $[-1, 1]$. Equation 2 [21] shows the weight updating formula for each feature of the process parameter vector $p$.

$$\begin{aligned} \mathbf{V(p)} &= \mathbf{W(p)} + \frac{\mathbf{N_{dCdp}}}{\mathbf{N_{dC}}} - \frac{(\mathbf{N_{dp}} - \mathbf{N_{dCdp}})}{\mathbf{n} - \mathbf{N_{dC}}} \qquad (2) \\ \mathbf{N_{dp}} &= \frac{|\mathbf{value(p, x_i)} - \mathbf{value(p, x_j)}|}{\mathbf{max(p)} - \mathbf{min(p)}} \mathbf{d(i, j)} \\ \mathbf{N_{dC}} &= |\mathbf{y_i} - \mathbf{y_j}| \mathbf{d(i, j)} \\ \mathbf{N_{dCdp}} &= |\mathbf{y_i} - \mathbf{y_j}| |\mathbf{value(p, x_i)} - \mathbf{value(p, x_j)}| \mathbf{d(i, j)} \end{aligned}$$

RReliefF starts with a $l$-long weight vector, $V$, of zeros, and iteratively updates $V$ for all features in $p$. This process is repeated for the total number of instances $n$. In each iteration, the algorithm randomly selects a sample $x_i$ and finds all $k$ nearest samples $x_j$ around $x_i$, in terms of Euclidean distance. The relevance level of each feature is then assigned by approximating the terms in Equation 2, where $N_{dp}$ is a normalized difference between the values of parameters in the vector $p$ for the two instances $x_i$ and $x_j$, the quantity $d(i, j)$ [21] takes into account the distance between samples by assigning greater weight to closer samples, and $N_{dC}$ corresponds to the difference between the performances of the two samples. The term $N_{dCdp}$ quantifies the probability that two nearest samples have different performances and different values of parameter. The weight increases if the circuit responses of nearest samples differs and decreases in the reverse case. In practice, a feature is relevant when the weight is strictly positive and irrelevant in the opposite case [22]. The algorithm only requires $O(lnlog(n))$ time, and is noise-tolerant and robust to feature interactions.

*2) Adaptive least-squares regression using LASSO:* Once the most relevant process parameters are captured, we seek to construct a surrogate model of each performance metric involved in the circuit specification. The performance function is a local perturbation around its nominal value. We use polynomial basis which are very often used to approximate such a local variation [9]:

$$\mathbf{f(p)} \simeq \sum_{\mathbf{m=1}}^{\mathbf{M}} \mathbf{c_m g_m(p)} \qquad (3)$$

where $f$ is a smooth circuit performance approximated as a linear combination of $M$ basis functions, $c_m$ are the model coefficients, $g_m(p)$ is a basis functions (linear, quadratic or cubic polynomials). The unknown model coefficients $c_m$ are determined by solving a set of linear equations at a number of sampling points (training data), which is usually solved as a least squares problem:

$$\min_{\mathbf{c_m, m\in[1,M]}} \|\mathbf{f(p)} - \mathbf{q(p)}\|_\mathbf{2}^\mathbf{2}, \quad \mathbf{q(p)} = \sum_{\mathbf{m=1}}^{\mathbf{M}} \mathbf{c_m g_m(p)} \qquad (4)$$

In fact, the number of process parameters is often large, while the number of training samples is greatly limited by the computational cost. Given the limited computational budget, the underlying system is rank deficient. Therefore, the solution $c_m$ (i.e., the vector containing unknown model coefficients) is not unique and impossible to identify without additional constraints. To solve this problem, we propose to employ adaptive LASSO as a weighted regularization technique for simultaneous consistent estimation and variable selection [19]:

$$\min_{\mathbf{c_m, m\in[1,M]}} \|\mathbf{f(p)} - \mathbf{q(p)}\|_\mathbf{2}^\mathbf{2} + \alpha \sum_{\mathbf{m=1}}^{\mathbf{M}} \|\frac{\mathbf{c_m}}{\mathbf{w_m}}\|_\mathbf{1} \qquad (5)$$

where $\alpha$ is a nonnegative regularization parameter. $\|\|_1$ stands for the $l_1$-norm of a vector which denotes the sum of the absolute values of all elements in the vector. The weighted penalty function $\alpha \sum_{m=1}^{M} \|\frac{c_m}{w_m}\|_1$ is an additional constraint that forces the coefficients $c_m$ to behave regularly by shrinking the coefficients towards 0 as $\alpha$ increases. Data-dependent weights $w$ are employed for penalizing different coefficients in the $l_1$ penalty. By allowing relatively higher penalty function (higher weight) for small coefficients and lower penalty function (lower weight) for larger coefficients, the adaptive LASSO neutralizes the influence of the coefficient magnitude on the $l_1$ penalty function. Thus, it reduces the coefficient estimation bias compared with the standard LASSO. Furthermore, the adaptive LASSO shrinkage retains the attractive convexity property of the standard LASSO [19]. Most importantly, it is proved to be near-minimax optimal [23]. The weight $w$ can be any consistent estimate of $c_m$. Here, we select $w = (X^T X)^{-1} X^T Y$ to be the ordinary least square estimate of $c_m$ [23], where $X^T$ denotes the vector transpose of $X$.

Algorithm 1 provides a simplified description of the adaptive sparse regression algorithm. This algorithm is applied to construct a surrogate model $q(\tilde{p})$ of each performance metric intervening in the circuit specification. It requires as inputs a set of training $X$ and test samples $X^t$ and their corresponding circuit responses $Y$ and $Y^t$, respectively. Typically, the number of training samples can be selected from 200 to 500 while the test samples from 100 to 300. In Line 1, we use the RReliefF algorithm to select a smaller number of features $\tilde{p}$ and filter out features that hardly have contributions to the circuit response.

The parameter $k$ is the number of nearest instance con-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



**Alg. 1** Response surface-based surrogate model training

**Require:** $X$, $X^t$: Data samples, $Y$, $Y^t$ : Circuit response,
$D = 3$: Maximum degree, $d = 0$, $k = 15$, $R_{th}$: Accuracy threshold
1: $\tilde{p} \leftarrow$RReliefF$(X, Y, k)$,
2: $X_{\tilde{p}} \leftarrow$ select$(X, \tilde{p})$, $X^t_{\tilde{p}} \leftarrow$ select$(X^t, \tilde{p})$
3: **while** $d < D$ and $\varepsilon > R_{th}$ **do**
4: $d \leftarrow d + 1$
5: $\tilde{X}_f \leftarrow$ expand_polynomial_basis ($X_{\tilde{p}}$, $\tilde{p}$, $d$)
6: $w \leftarrow$ compute_weight$(\tilde{X}_f, Y)$
7: $q(\tilde{p}) \leftarrow$ adaptive_lasso$(w, \tilde{X}_f, Y)$
8: $\varepsilon \leftarrow$ verify$(q, X^t_{\tilde{p}}, Y^t)$
9: **end while**
10: **if** $\varepsilon \leq R_{th}$ **then**
11: Return (Accuracy model met!)
12: **else**
13: Generate fresh samples and go to **5**
14: **end if**

sidered by RReliefF [24]. In all experiments conducted in this work (cf. Section IV), we find that $k = 15$ provides stable and reliable reduction results. In Line 2, the function $select$ extracts the observation $X_{\tilde{p}}$ and $X^t_{\tilde{p}}$ corresponding to the reduced process parameters space $\tilde{p}$ from the original set $X$ and $X^t$, respectively. Then, the algorithm operates in an iterative fashion. At each iteration, the polynomial degree is incremented (Line 4). The idea is that higher degree terms are included only when necessary to avoid high order model. In Line 5, we construct a set of polynomial basis $g_m(\tilde{p})$ of degree $d$. The polynomial terms of $g_m(\tilde{p})$ are obtained by expanding all the terms in the d-degree polynomial $(1+p_1+\cdots)^d$. Then, $\tilde{X}_f$ maps the reduced data matrix $X_{\tilde{p}}$ to each expansion terms of $g_m(\tilde{p})$. In Lines 6 and 7, the weights $w$ are computed and the adaptive LASSO problem in Equation 6 is solved using the coordinate descent algorithm [24].

$$\min_{\mathbf{c_m}, \mathbf{m}\in[\mathbf{1},\mathbf{M}]} \|\mathbf{Y} - \tilde{\mathbf{X}}_\mathbf{f}\mathbf{c_m}\|_\mathbf{2}^\mathbf{2} + \alpha\|\frac{\mathbf{c_m}}{\mathbf{w_m}}\|_\mathbf{1} \tag{6}$$

The coordinate descent iterations terminate when the relative change in the size of the estimated coefficients drops below $1e^{-9}$. It is important to note that $c_m$ are computed each time the degree $d$ is incremented. This re-calculation is required because the new basis function constructed at the current iteration step may change the model coefficient values calculated at previous iteration steps. The regularization parameter $\alpha$ is chosen during the training process. It is selected such that it minimizes an estimate of expected prediction error based on 10 fold cross-validation applied to the training samples. In Line 8, the test samples $X^t_{\tilde{p}}$ are used to verify the accuracy of the current trained model. The prediction ability of the model is tested by calculating the normalized mean square error (NMSE=$\frac{\|q(X^t_{\tilde{p}})-Y^t\|_2^2}{\|Y^t\|_2^2}$). When the error of the performance model $\varepsilon$ is less than a given threshold, named $R_{th}$, or the degree $d$ reaches the limit $D$, the iteration stops.

If the desired accuracy is not met and $d$ reached the maximum degree $D$, then fresh samples are generated and added incrementally to the training sample set as long as the model accuracy does not satisfy the convergence condition (Line 13). The generation of the fresh samples uses a triangulation approach as explained in [7]. How to select the parameter $R_{th}$ will be discussed in Section IV-D.

In practice, the number of samples required to compute $\varepsilon$ cannot be fixed in advance. If a very large evaluation set is employed to evaluate the error $\varepsilon$, then the resulting model accuracy can be trusted. However, this would prohibitively increase the computational cost. Next, we propose to employ statistics to provide a certain confidence level on the model accuracy with a probability of error which can be pre-specified.

### *B. Accuracy Generalization and Verification*

On one hand, the surrogate model error $\varepsilon$ can never be totally eliminated. On the other hand, its accuracy verification is primordial to prove the reliability of the yield estimation methodology. The surrogate model accuracy $(1 - \varepsilon)$ can be considered $\varphi$-guaranteed if :

$$\forall \mathbf{p}, \tilde{\mathbf{p}} \in \mathbf{P}, \quad \mathbf{Pr}((\mathbf{err}(\mathbf{f}(\mathbf{p}), \mathbf{q}(\tilde{\mathbf{p}})) \leq \varepsilon) \quad \geq \varphi \tag{7}$$

where $Pr$ and $err$ stand for probability and model error, respectively. In other words, the model error is at most $\varepsilon$ for at least $\varphi$ portion of the parameter space. Clearly, at this stage there is no guarantee on the model accuracy $(1 - \varepsilon)$. The purpose of this step is to determine a generalized accuracy under the process parameter space, given a probability/level of confidence $\varphi$.

To do so, we employ and extend the statistical procedure proposed by Younes [25] that regards the model checking of a system as a hypothesis testing problem and solves it using Walds sequential probability ratio test (SPRT) [26]. The idea is to check the accuracy property in Equation 7 on a samples set of simulations and to decide whether the model $q(\tilde{p})$ satisfies the property based on the number of executions for which the property holds compared to the total number of executions. With such an approach, we do not need to explore and test all possible values of process parameters. We rather answer the question of whether the model satisfies the property with a probability greater than or equal to a value $\varphi \in [0, 1]$. Furthermore, we propose a simple, yet elegant modification to the SPRT test which allows the computation of a generalized model accuracy $\varepsilon$. The problem is treated based on two exclusive hypothesis testing given as follows:

$$\begin{aligned} \mathbf{H_0} &= \mathbf{Pr}(\mathbf{err}(\mathbf{f}(\mathbf{p}), \mathbf{q}(\tilde{\mathbf{p}})) \leq \varepsilon) \geq \varphi + \delta = \boldsymbol{\varphi_2} \\ \mathbf{H_1} &= \mathbf{Pr}(\mathbf{err}(\mathbf{f}(\mathbf{p}), \mathbf{q}(\tilde{\mathbf{p}})) \leq \varepsilon) < \varphi - \delta = \boldsymbol{\varphi_1} \end{aligned} \tag{8}$$

where $H_0$ and $H_1$ are known as the null and the alternative hypothesis and $2\delta$ forms a small region called the *indifference region* [25], on both sides of the cutting point $\varphi$. If the probability is between $\varphi_1$ and $\varphi_1$ (*the indifference region)*, then we say that the probability is sufficiently close to $\varphi$ so that we are indifferent with respect to which of the two hypotheses is accepted. The method determines on the fly the number of simulations needed to achieve a desired accuracy and provides a convenient way to control the trade-off between precision and computational cost. To decide between the two hypothesis, the test proceeds by computing at the $n^{th}$ stage of the test, i.e., after making $n$ observations, a log likelihood ratio given as:

$$\mathbf{\Lambda_n} = log\frac{\prod_{i=1}^n z_{\varphi_1}(b_i)}{\prod_{i=1}^n z_{\varphi_2}(b_i)} = log\frac{\int_0^{\varphi_1}\prod_{i=1}^n z^{b_i}(1-z)^{1-b_i}dz}{\int_{\varphi_2}^1\prod_{i=1}^n z^{b_i}(1-z)^{1-b_i}dz} \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



where $n$ represents the total number of samples or the test length, $b_1, b_2, \cdots, b_n$ is a collection of Bernouilli random variables denoting the outcome of the accuracy property (Equation 7) with random samples $x_1, x_2, \cdots, x_n$ drawn from the parameters space. $z_{\varphi_1}(b_i)$ and $z_{\varphi_2}(b_i)$ are the probability mass function of the Bernouilli distribution parameterized by $\varphi_1$ and $\varphi_2$, respectively. The quantity $\mathbf{\Lambda_n}$ is finally given as:

$$\mathbf{\Lambda_n} = log\frac{B_{\varphi_1}(k+1, n-k+1)}{A - B_{\varphi_2}(k+1, n-k+1)} \tag{10}$$

where $0 \leq k \leq n$ is the number of successful inequality test, $A = \frac{1}{(n+1)C_k^n}$ and $B_{\varphi 1}$ and $B_{\varphi 2}$ are the incomplete Beta functions. $H_0$ is accepted if $\Lambda_n \leq a$ and $H_1$ is accepted if $\Lambda_n \geq b$, where $a$ and $b$ are given in Line 1 of Algorithm 2. $\alpha$ and $\beta$ are two decision error rates that determine the strength of the test, where $\alpha$ is the type I error rate or false positive and $\beta$ is the type II error rate or false negative.

**Alg. 2** Verification and generalization of the model accuracy

**Require:** $q$: Surrogate model, $\varepsilon$: model error, $\tilde{p}$,$p$: Process parameters, $\varphi_1, \varphi_2$: Probabilities, $\alpha, \beta$: Error rates.
1: $a = \log(\frac{\alpha}{1-\beta})$; $b = \log(\frac{1-\alpha}{\beta})$, $X_{\tilde{p}}^t, Y^t$
2: $n = 0$; $k = 0$;
3: **while** $a < \Lambda_n < b$ **do**
4: $n \leftarrow n + 1$
5: $x_n \leftarrow$ Sample the parameters space $P$
6: $f \leftarrow$ Simulate the circuit at the parameters $x_n$ and measure $f$
7: $X_{\tilde{p}}^t, Y^t \leftarrow$Update($X_{\tilde{p}}^t, Y^t, x_n, f$)
8: **if** err($q(X_{\tilde{p}}^t), Y^t$)$> \varepsilon$ **then**
9: $\varepsilon \leftarrow$ err($q(X_{\tilde{p}}^t), Y^t$)
10: **else**
11: $k \leftarrow k + 1$
12: **end if**
13: Evaluate $\Lambda_n(n, k, \varphi_1, \varphi_2)$
14: **end while**
15: **if** $\Lambda_n \leq a$ **then**
16: Accept $H_0$
17: **else**
18: Accept $H_1$
19: **end if**

The procedure is summarized in Algorithm 2. It repeatedly checks the accuracy property with fresh samples $x_n$ drawn from the parameters space $p$ (Line 5). After measuring the sample response $f$ (Line 6), we add the fresh observation $(x_n, f)$ to the testing samples $(X_{\tilde{p}}^t, Y^t)$ (Line 7) and we compute the normalized mean square error (Line 8). We say that the inequality test is a success if the property holds, and a failure otherwise. Upon each success, we increment the counter $k$ (Line 11) and continue with fresh samples until a failure occurs. In this case, we update and generalize the error $\varepsilon$ (Line 9). We can therefore characterize the required number of observations as $inf\{n, \Lambda_n \notin ]a, b[\}$. Clearly, this number increases if $\alpha$ and $\beta$ are smaller but also if $\varphi$ is very close to one. We provide in Section IV-D a discussion concerning these parameters.

### C. SMT-based Parameters Space Exploration

The objective of this stage of the methodology is to exhaustively probe the parameters space and to determine failure hyperrectangles, i.e., regions where the circuit fails to satisfy the design specification. Our approach is summarized in Algorithm 3. In order to conservatively find the reachable parameters values, we formulate the SMT problem $constr$ as a conjunction of the space of the process parameters, the constructed surrogate models and the specification violation constraints. In general, the problem can be formulated as:

$$\begin{array}{rcl} \mathbf{p^{min}} & \leq \mathbf{p} \leq & \mathbf{p^{max}} \\ \mathbf{f_k}(\mathbf{\tilde{p}_k}) & = & \mathbf{q_k}(\mathbf{\tilde{p}_k}) \\ \mathbf{f_K^{min}} & \leq \mathbf{f_K} \leq & \mathbf{f_K^{max}}, \mathbf{K = 1} \\ \bigvee_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{f_k^{min}} & \leq \mathbf{f_k} \leq & \mathbf{f_k^{max}}, \mathbf{K > 1} \end{array} \tag{11}$$

where $f_k(\tilde{p}_k), k = 1 \ldots K$, are the performance equations, $K$ is the total number of performance metrics involved in the design specification, $\tilde{p}_k$ is the reduced process parameters set associated to the $k^{th}$ performance metric. $[p^{min}, p^{max}]$ are the ranges of the process parameters determined from their probabilities distributions, where $p = [p_1, p_2, \ldots, p_r]$ and $r = dim(\cup_1^k \tilde{p}_k)$ is the dimension of the reduced parameters space. As mentioned before, a truncated normal shape is used in this paper to model the process parameters. If $\pm 3\sigma$ variation is considered then, the upper and lower bounds of the process parameters $p^{min}$ and $p^{max}$, respectively, are defined as:

$$\mathbf{p^{min}} = \mathbf{p^{nom}} - \mathbf{3}\sigma; \mathbf{p^{max}} = \mathbf{p^{nom}} + \mathbf{3}\sigma \tag{12}$$

where $p^{nom}$ is a vector of nominal values. $[f_k^{min}, f_k^{max}]$ are the bounds that approximate the failure region of the circuit operation in the performance space. For example, if we are given an oscillator circuit designed at a nominal frequency $f_{nom}$ and the maximum allowed frequency deviation is $\triangle f$, then the $failure$ frequency region is defined as: $[f^l, f_{nom} - \triangle f[\cup] f_{nom} + \triangle f, f^u]$, where $f^l$ and $f^u$ are the minimum and maximum performances values reached by the circuit. It is important to set a conservative approximation of $f^l$ and $f^u$ in order to let the solver discover any possible failure of the circuit response under the defined parameters variation. The over-conservativeness is especially necessary for circuits with rare failure event where the circuit simulation in the initial pre-sampling cannot be sufficient to sketch the performance bound. We provide in Section IV-D a discussion concerning the setting of the failure performance bound.

In case of multiple performance metrics, the specification violation is mathematically formulated as a disjunction of failure performance bounds, as given in Line 4 of Equation 11, where $\bigvee$ denotes the logical OR operator. In fact, a high dimensional region in the parameters space is considered as a failure region if any performance metric involved in the specification is not satisfied.

The SMT solver iSAT3 [11] we are using in this work is known to attempt to solve NP-complete problems. Solving these problems, in their worst case, would take time which is exponential in the number of variables to solve. It would be then infeasible to run the search over a large initial space of $failure$ performance bounds $[f_k^{min}, f_k^{max}]$. For these reasons, we propose first to split the SMT problem $constr$ into $N_S = S^K$ subproblems that we solve simultaneously (Line 1). For example, if the circuit requires two performance metrics $(K = 2)$ with $S = 5$ uniform descritazation steps,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



**Alg. 3** SMT-based parameters space exploration

**Require:** $S, K, constr$, $N_S = S^K$
1: **for all** $ind = 1 \rightarrow N_S$ **do in parallel**
2: $f_k \subseteq [f_k^{min}, f_k^{max}]_{ind}$
3: **repeat**
4: Invoke iSAT3($constr$)
5: **if** a $candidate$ is found **then**
6: Invoke INTLAB($constr, candidate$)
7: **if** Locate $p^{box}$ **then**
8: Return($Perf^{box}, p^{box}$)
9: Update($Perf^{box}, f_k$)
10: **end if**
11: **end if**
12: **until** *Unsatisfiable*
13: **end for**

then the overall combinations of performance space to be explored is $N_S = S^K = 5^2$. Each subproblem is limited to a possible combination of performance boundaries. More precisely, for each subproblem, a possible combination of the failure regions in the performance space is traversed and the specification violation constraint is formulated as: $\bigvee_{k=1}^{K} f_k \subseteq [f_k^{min}, f_k^{max}]_{ind}, k = 1 \ldots K$. Also, it is important to note that all subproblems have the same SMT constraints and the same process parameters variables. Based on this, solving all subproblems is completely equivalent to solving the original SMT problem.

Obviously, we can observe that the complexity increases with more performance metrics and greater precision in sampling. For this reason, the SMT subproblems are solved in parallel to reduce the timing complexity. The solver returns a set of continuous ranges of each variable (i.e., a hyperectangle) in the SMT constraints (Line 5). However, the set of interval solutions is only an over-approximation ($candidate$) that can be devoid of any real solution to the constraints. The uncertainty can be alleviated by setting a high resolution of the returned $candidate$. Still, this will dramatically increase the computation time. Owing to this, the size of the interval solution (resolution) is adjusted for a tradeoff between computational cost and over-approximation.

In fact, we only use the SMT solver to refine the initial search space towards a $candidate$ solution and to discard the infeasible solution. Afterwards, for each set of intervals proposed by iSAT3, we exploit the Matlab toolbox for interval arithmetic INTLAB [27] to further refine the $candidate$ solution (Line 6). Given the $candidate$ solution as interval initial condition and the performance equations, INTLAB either refutes the existence of any solution in the candidate solution returned by the SMT solver or produces an hyperrectangle $p^{box}$ that is contained in the $candidate$ region and guaranteed to contain the solution (Line 7). The widths of the interval solution $p^{box}$ returned by INTLAB are smaller than the $candidate$ region proposed by the SMT solver.

The result of the refinement process is a set of interval process parameters $p^{box}$ and its corresponding reachable performances $Perf^{box}$ (Line 8). The function $Update$ in Line 9 removes $Perf^{box}$ from the search space by adding the constraint $Perf^{box} \nsubseteq f_k$. This will force the solver to search for new solutions. Finally, when all reachable hyperrectangles are found, the solver will return *Unsatisfiable*, providing a guarantee on a complete coverage of the search space (i.e., the failure region). In fact, Algorithm 3 exploits the strength of the SMT solver (i.e., its search space coverage capabilities) while avoiding its disadvantages.

### D. Yield Estimation

In the previous stage of the methodology, we have characterized $\Omega$ as a set of high dimensional sub-regions in the parameters space: $\Omega \simeq \{p^{box}\}_{1 \longrightarrow n_f}$, where $n_f$ is the total number of located sub-regions. A failure sub-region is a hyperrectangle that is modeled as a cartesian product of orthogonal intervals $p^{box} = ([p_1^l, p_1^u] \times \ldots \times [p_r^l, p_r^u])$. We recall that the parameters $p$ are assumed independent and continuous random variables. The probability that the process parameters fall into a single sub-region $p^{box}$ is estimated in two dimensions (for illustrative purposes) as:

$$\begin{aligned} P(p_1, p_2 \in p^{box}) &= \int_{p^{box}} pdf(p)dp = \prod_{i=1}^{2} P(p_i^l \leq p_i \leq p_i^u) \\ &= \prod_{i=1}^{2} CDF(p_i^u) - CDF(p_i^l) \end{aligned} \quad (13)$$

where $P$ stands for probability, $p_1^u, p_1^l, p_2^u, p_2^l$ are the coordinates of the sub-region in two dimension (as shown in Figure 3), and $CDF(p_i)$ [24] represents the cumulative distribution function of $p_i$. For the total $n_f$ failure sub-regions in $r$-dimensional parameters space, the probability that the design constraints are satisfied in the presence of parameters variation is generalized as:

$$\begin{aligned} \mathbf{Y}^* &= 1 - P_f = 1 - \sum_{j=1}^{n_f} \int_{\{p^{box}\}_j} pdf(p)dp \\ &= 1 - \sum_{j=1}^{n_f} [\prod_{i=1}^{r} CDF(p_i^u) - CDF(p_i^l)]_j \end{aligned} \quad (14)$$

The multidimensional integral in Equation 14 is the probabilistic hypervolume of a single sub-region. Obviously, the contribution of a located sub-region to the failure probability $P_f$ is higher when the the coordinates of the hyperectangles are closer to the center of the process parameters space. The circuit yield is computed according to Algorithm 4. In Line 2, the hyperrectangle is refined for more precision and accuracy. The term $\bigcap(p_j^{box}, p_{1 \rightarrow j-1}^{box})$ is the region resulting from the overlapping between the located boxes. The overlay may occur if some hyperrectangle share the same values of process parameters or due to the conservativeness of interval arithmetic computation.

**Alg. 4** Yield rate computation

**Require:** $\{p_{box}\}_{1 \longrightarrow n_f}$
$P_f = [\prod_{i=1}^{r} CDF(p_i^u) - CDF(p_i^l)]_1$
1: **for all** $j = 2 \rightarrow n_f$ **do**
2: $p_j^{box} \leftarrow p_j^{box} - \bigcap(p_j^{box}, p_{1 \rightarrow j-1}^{box})$
3: $P_f \leftarrow P_f + [\prod_{i=1}^{r} CDF(p_i^u) - CDF(p_i^l)]_j$
4: **end for**
5: $\mathbf{Y}^* \leftarrow 1 - P_f$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



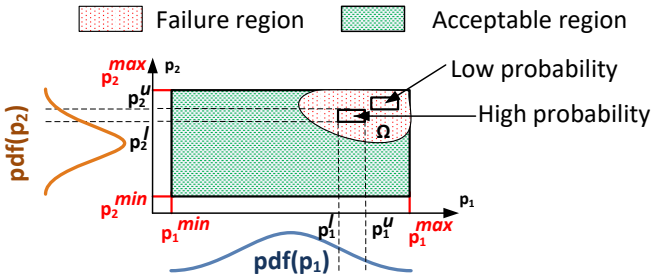


Figure 3: Illustration of the coordinates of a failure sub-region in 2-D parameters space

## IV. Applications

In this section, we present the application of the yield rate estimation methodology on the examples of a three-stage ring oscillator, a six transistor SRAM cell and a three-stage operational amplifier (op-amp). In the experiments, the circuits are designed in a commercial TSMC 65 nm process and simulated in HSPICE with BSIM4 transistor models. The local mismatch variables are considered as the process parameters including the oxide thickness $\triangle t_{ox}$, threshold voltage under zero bias $\triangle V_{th}$, channel width $\triangle w$ and channel length $\triangle L$. We use the TSMC 65 nm transistor mismatch model with $Vdd = 1V$ and standard threshold voltage. The mismatch model uses principal component analysis (PCA) [19] to model the process parameters as a set of independent random variables.

The algorithms parameters are selected as follows. The value of the convergence condition $R_{th}$ in Algorithm 1 is selected as $2.10^{-2}$. We also choose a degree limit $D$ of 3 for all performances models. For the model verification step, we used $\varphi = 0.95$, a symmetric test strength $\alpha = \beta = 0.01$ and an indifference region of size $10^{-3}$, indicating that the statistical test covers at least $95\%$ of the parameter space with a high statistical condence. All simulations were performed using an 8-core Intel CPU i7- 860 processor running at 2.8 GHz with 32 GB memory and Linux operating system.

### A. Three-stage Ring Oscillator

We consider a three-stage ring oscillator as shown in Figure 4. The oscillation frequency is chosen to be the performance metric of interest. The nominal frequency $f_{nom}$ is $3.207\ GHz$ calculated via periodical steady state (PSS) simulation. The design specification requires that the variation of the frequency should be within $2.5\%$ of $f_{nom}$.

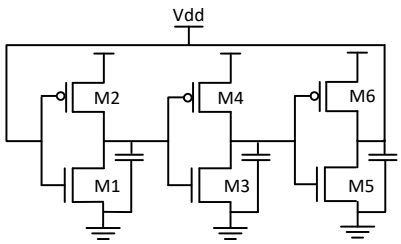


Figure 4: A Three-stage Ring Oscillator

The oscillation frequency is affected by various process parameters in the transistors. The local mismatch variables of each transistor are considered as the process parameters, which results in a 24-dimensional problem.

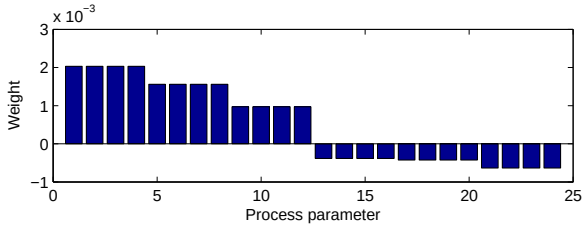


Figure 5: Weight of all 24 process variations for the frequency oscillation performance

Firstly, we consider 400 LHS data samples with 300 of them for training and 100 for testing. On this 24-dim problem, RReliefF is performed to reduce the dimension before constructing the frequency model. For each process parameter, the weight is evaluated and ranked as illustrated in Figure 5. The process parameters with negative weight are discarded and 12 parameters are kept.

We measure the oscillation frequency under the effect of the reduced set of process parameters, in order to check the accuracy of the reduction process. Figure 6 shows the frequency performance of 300 LHS data samples when considering the total number of process parameters (Original dim-24) and the reduced one (Reduced dim-12). The frequency responses are evaluated using the circuit simulator HSPICE. As it can be observed in Figure 6, the frequency response with the reduced set exhibits some deviation as expected. The reduction error is checked by calculating the normalized mean square error (NMSE), which is given as: $(\frac{\|freq(12-dim)-freq(24-dim)\|_2^2}{\|freq(24-dim)\|_2^2}$=0.0245%). The actual error is less than $0.1\%$ which is considered excellent in practice [28].

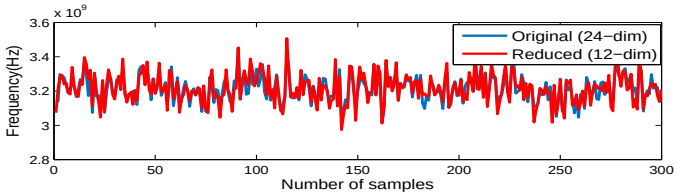


Figure 6: Ring Oscillator frequency responses under the original and the reduced process parameters variational space

After applying the proposed adaptive LASSO scheme for surrogate modeling, we extract a frequency model of degree 3. The ability of the proposed modeling technique is compared to the generic sparse regression (SR) using the standard LASSO method, applied without the parameters pruning stage. The frequency of the test samples are calculated by both the constructed frequency model and HSPICE simulation. The modeling results are summarized in Table I.

Table I: Frequency modeling result

| | ASR | SR |
|---|---|---|
| Fitting time (s) | 85 | 160 |
| Model accuracy (1-NMSE)(%) | 98.65 | 65.61 |
| ♯ of training samples | 300 | |
| ♯ of testing samples | 100 | |

First, the proposed ASR algorithm appropriately selects a small subset of important monomial polynomial basis when

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



Table II: Yield results for the ring oscillator with 24 process parameters.

| Method | Total (♯) of HB sim. runs | Time Cost (h) | Yield (%) | Speed-up | Relative Error(%) |
|---|---|---|---|---|---|
| Brute-force MC | 10000 | 4.79 | 73.57 | 1X | – |
| Brute-force MC | 5000 | 2.38 | 71.80 | 2X | 2.41 |
| Quasi MC | 4619 | 2.2125 | 73.57 | 2.16X | 0.001 |
| MC+LHS | 6475 | 3.1015 | 73.88 | 1.54X | 0.42 |
| Our method | 560 | 0.45 | 73.51 | 11X | 0.081 |

Table III: Yield results for the ring oscillator with 3 process parameters.

| Method | Total (♯) of HB sim. runs | Time Cost (h) | Yield (%) | Speed-up | Relative Error(%) |
|---|---|---|---|---|---|
| Brute-force MC | 10000 | 4.79 | 89.95 | 1X | – |
| Our method | 380 | 0.1813 | 90.02 | 26.42X | 0.077 |

compared to SR. Second, ASR achieves 33% better fitting accuracy than the standard LASSO. This in turn demonstrates the advantage of the weighted regression approach to consistently approximate the frequency model coefficients so that the results are not over-fitted due to the limited training set. Third, the fitting time (i.e., the cost of solving all model coefficients from the sampling points) is almost two time less than the generic SR. The fitting time reduction has been achieved thanks to the process parameters pruning.

Algorithm 2 computes 160 circuit simulations required to generalize and verify the frequency model accuracy. Figure 7 shows a graphical representation of the statistical test. The line $a$ is the acceptance line. Similarly, the line $b$ is the rejection line for the test. The curve intersects the line $a$ at the observation number 160. The test is achieved at this point with a high generalized accuracy of 98.1%. At the $80^{th}$ and $82^{th}$ circuit simulation, the accuracy test has failed and the model error has been updated (i.e., generalized).

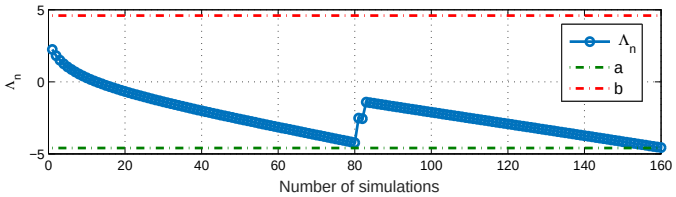


Figure 7: Generalization and verification of the frequency model accuracy

In Table II, we compare our results with different sampling methods including the brute-force Monte Carlo (MC), Quasi Monte Carlo (QMC) and Latin Hypercube Sampling (MC+LHS), implemented in HSPICE. Column 2 of Table II shows the number of harmonic balance (HB) circuit simulations and "Time Cost" is the time spent on simulations. The brute force Monte Carlo with 10000 is able to compute a highly accurate result of the yield rate with an estimated error $< 1\%$ at a 99% level of confidence. It is used as the golden result to assess the accuracy and efficiency of all others methods in this experiment.

For our method, the number of HB simulations includes the number of simulations performed in the model fitting and accuracy verification phases. The 560 HB simulation runs correspond to 300 training samples, 100 testing samples and 160 samples for accuracy verification. The column "Time Cost" includes the time for all stages in the proposed methodology (i.e., the surrogate model fitting and verification, the parameter space exploration and the yield calculation). During the SMT-based parameters exploration stage, we define the full fail performance intervals as $[2.5Ghz, f_{nom} - f_{nom}2.5\%[\cup]f_{nom} + f_{nom}2.5\%, 4Ghz]$. In this experiment, $S^P = 5^2 = 25$ combinations of performance boundaries have been explored in parallel. The SMT solver [11] has reported 2820 candidate regions. 2643 regions were confirmed by INTLAB during the solution refinement step. The regions found by the SMT solver and not confirmed during the refinement step are spurious. In this case, INTLAB refuted the existence of any solutions within the candidate regions.

On the basis of Table II, it can be observed that the performance of the MC variants do not achieve significant improvement when compared to the brute-force Monte Carlo analysis engine. QMC is able to reach the MC golden result with around 2.16X speedup, while the MC+LHS method is 1.54X times faster than MC with approximately the same yield rate. Collecting extra random samples for MC+LHS does not help to converge exactly to the MC golden estimation. This observation can be explained by a bad exploration of the parameters space and a moderate uniformity properties of MC+LHS in this 24-dimensional problem.

Since the proposed method attempts to ensure an exhaustive coverage of the failure regions in the parameters space, it tends to under-estimate the yield. It explains why the predicted yield from our procedure is slightly lower than the sampling yield from MC simulations. However, the computed yield rate is almost identical to that estimated by the brute-force Monte Carlo engine with 10000 samples. Algorithm 3 completed the search for the failure sub-regions in $0.16h$, which is affordable and clearly demonstrates the scalability of the proposed method. In fact, the SMT problem subdivision allowed the reduction of the search space (i.e., failure performance space), and when coupled with the parallel implementation, it highly relieves the computational cost of the SMT solver.

Our method can achieve 11X speedup over the MC method while it adopts a more exhaustive approach for the yield estimation. The achieved speedup can be explained by: (1) the process parameters reduction step; (2) the employment of a surrogate model of the frequency model to replace time consuming transistor-level HB simulation; and (3) the tracking of a complete hyperrectangle in the parameters space rather one sample point which allows a faster coverage of the failure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



regions.

In order to illustrate the capability of our method in handling multiple and distinct failure regions, we use a simplified process variational space, which only considers the threshold voltages of the n-mos transistors $M1$, $M3$ and $M5$ as the sources of process variations. In this experiment, we applied the proposed scheme without the parameters pruning stage and we formulate the SMT constraints to locate the failure regions in this 3-dimensional problem. The results are summarized in Table III. As less process variables are taken into account, the time cost has significantly decreased comparing with the 24-dimensional problem and the yield rate has also increased. The failure sub-regions located by our method and the fail samples of the brute-force MC engine can be clearly visualized on a 3-dimensional parameters space as shown in Figure 8(a) and (b), respectively. The data is projected on the three directions $(V_{thM1}, V_{thM3}, V_{thM5})$ of the 3-dimensional space, where $V_{thMi} = V_{th0Mi} + \Delta V_{thMi}, i = 1, 3, 5$.

Figure 8(b) shows that, similarly to the MC method, the proposed method locates two failure regions. The two regions result from the interval specification of the frequency performance metric which can be equivalently expressed as two conflicting specifications. For both methods, the frequency specification is violated for high and low threshold voltage variations of the n-mos transistors of $M_1$, $M_3$ and $M_5$. However, while the MC method randomly samples the process parameters probability distribution $pdf(p)$ towards locating the failure operation, our method directly locates three dimensional failure sub-regions in the parameters space. Also, during the SMT-based parameters space exploration, the process parameters are modeled as a set of intervals in the SMT constraints. It explains why the located failure-sub regions covers the complete parameters space in Figure 8(b) and differs from the failure characterization of the brute-force MC method in Figure 8(a). It is only at the yield calculation step that the $pdf(p)$ of the process parameters are taken into consideration to estimate the probabilistic hypervolume of each single sub-region as given in Equation 14.

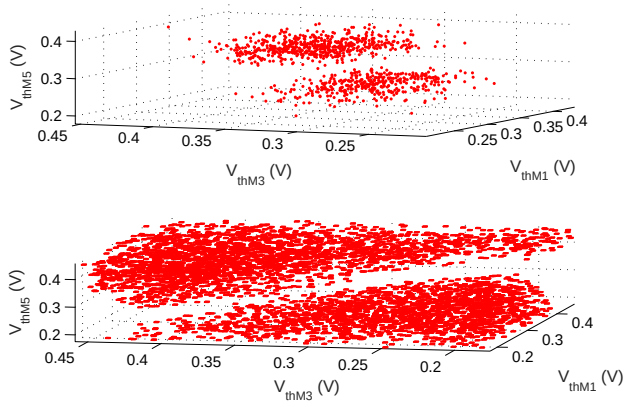


Figure 8: (a) Fail samples of the brute-force MC method (b) 3-dimensional failure sub-regions probed by the proposed method.

Furthermore, although the proposed approach may miss some failure sub-regions due to the modeling error, the probabilistic hypervolume of the located sub-regions still can be employed to estimate the yield with $0.077\%$ relative error when compared with the MC method. Based on this example, the ability of the proposed method in solving problems with multiple failure regions is verified.

### B. 6-Transistor SRAM Cell

In this section, a standard 6-T SRAM cell, shown in Figure 9, is used to validate the proposed method on a circuit with extremely high yield probability (i.e., very low failure rate $(P_f)$). In this example, a larger number of sigma variation $(6\sigma)$ is considered. We also suppose that the brute-force MC methods converges when the relative standard deviation of the failure probability $(std(P_f)/P_f)$ is equal to 0.1, (i.e., $90\%$ accuracy with $90\%$ confidence) [29]. The SRAM cell is used to store one memory bit: the four transistors M1, M2, M3 and M4 have two stable states, i.e., either a logic 0 or 1, and the two additional access transistors M5 and M6 serve to control the access to the cell during read and write operations.

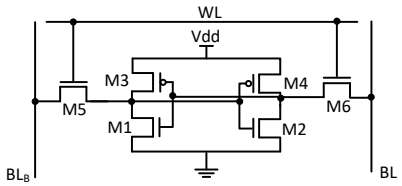


Figure 9: Schematic of a 6-T SRAM cell

The circuit performance is chosen as the read static noise margin (SNM) to evaluate the stability of the SRAM cell during read operation. The SNM is defined as the maximum value of DC noise voltage that can be tolerated by the SRAM cell without changing the stored bit [30]. A positive value of SNM represents a stable read operation while a zero or negative value of SNM signifies that the read operation will cause the cell to lose its state, resulting in the read stability failure. In this work, the SNM is measured by the length of the maximum embedded square in the butterfly curves (i.e., the voltage transfer curves (VTC) of the two inverters). When SNM is smaller than zero, the butterfly curves collapse and a data retention failure happens [30].

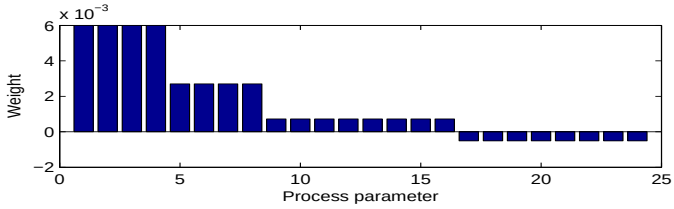


Figure 10: Weight of all 24 process parameters for the SNM performance

The local $\Delta t_{ox}$, $\Delta V_{th}$, $\Delta w$ and $\Delta L$ mismatch of each transistor are considered as the process variables, which result in 24 process parameters. On this 24 dimensional problem, RReliefF is applied to reduce the dimension before constructing the SNM performance model. For each process variation parameter, the weight is evaluated based on 300 training samples. The reduction process discarded $8$ process parameters as it can be observed in Figure 10. Figure 11 plots the SNM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



Table IV: Yield results for the SRAM with 24 process parameters.

| Method | Total (♯) of DC sim. runs | Time Cost | $P_f$ | Speed-up | Relative Error |
|---|---|---|---|---|---|
| Brute-force MC | $4.146090e^{+6}$ | 19.1951 Days | $7.2357e^{-5}$ | 1X | - |
| Quasi MC | $2.093045e^{+6}$ | 9.6903 Days | $7.2144e^{-5}$ | 1.9809X | 0.29% |
| Our method | 528 | 0.2484 hours | $7.2468e^{-5}$ | 1855X | 0.15% |

responses simulated by HSPICE, under the effect of the full and reduced process parameters set. We evaluate the NMSE to estimate the responses deviation. The computed error is $0.5\%$ which is low and can be considered as negligible.

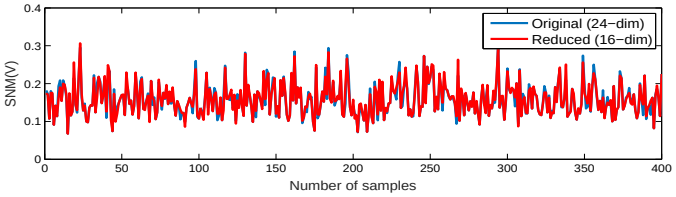


Figure 11: SNM responses under the original and reduced process parameters space

We apply the adaptive LASSO scheme for modeling the SNM surrogate model. We extract a polynomial model of degree 2 and we use 100 test samples to evaluate its accuracy. We verify and generalize the SNM model accuracy. The accuracy verification result is shown in Figure 12. Algorithm 2 computes a generalized model accuracy equal to $98.7\%$ based on 128 simulation runs.

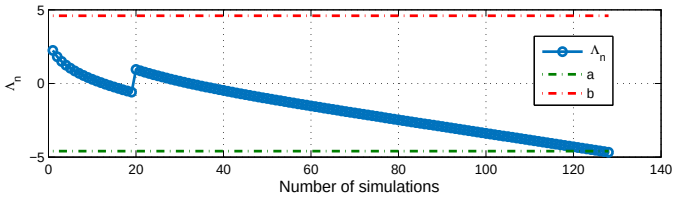


Figure 12: Generalization of the SNM model accuracy

The experimental results are summarized in Table IV. For our method, we define the full SNM fail interval as: $[-0.3V, 0V[$. We subdivide the SMT problem into $S^P = 5^1 = 5$ that we solve in parallel according to Algorithm 1. Column 2 of Table IV reports the number of simulations performed in the SNM model fitting and accuracy verification phases. The column "Time Cost" shows the time for the total stages in the proposed methodology.

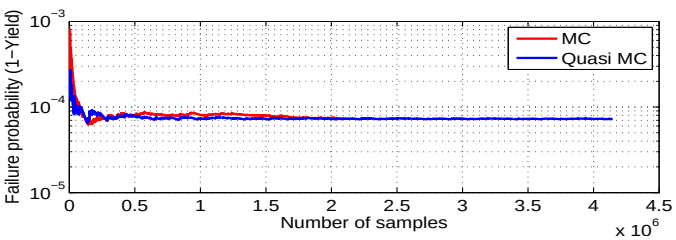


Figure 13: Evolution of the failure rate estimation as function of samples for the brute-force MC and the Quasi MC method

The MC method tries to randomly select samples to cover the entire parameters space, so it needs a huge number of samplings to achieve the target $90\%$ level of accuracy as shown in Figure 13. QMC is able to reduce the number of samplings by covering the entire space with deterministic sequences. It can be observed that the QMC method achieves around 2X speedup over the MC method with very close failure rate estimation. The method we propose in this paper achieves a speedup of approximately 2000X comparing with the MC method.

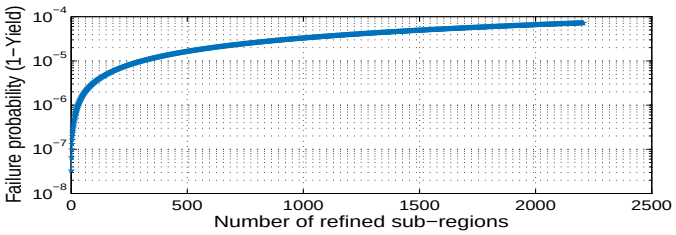


Figure 14: Evolution of failure rate estimation as function of tracked failure sub-regions for the proposed method

As shown in Figure 14, the proposed algorithm covers the failure region in the parameters space within 2205 located regions, which explains the relief in terms of computational cost. The first 400 refined regions had more contribution to the failure rate estimation in terms of probabilistic hypervolumes. The method also reaches a higher failure probability ($P_f$) compared to the MC method. This can be explained by the approach adopted in the proposed methodology that concentrates on the localization of only the failure regions in the parameters space. Meanwhile, the sampling methods waste a large number of samples that are far from the failure region.

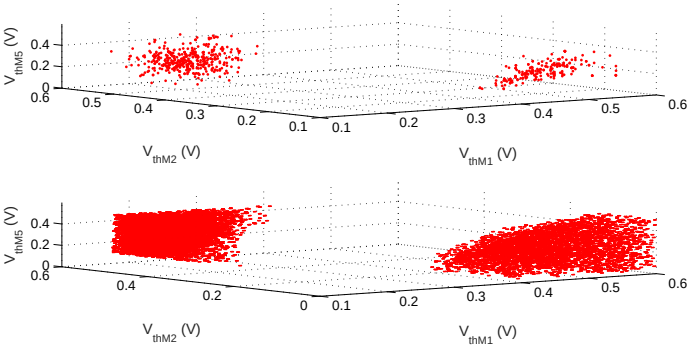


Figure 15: (a) Fail samples drawn from the simulation of the brute-force MC (b) Failure sub-regions located by the proposed method

Fail samples of the MC simulation result are drawn in Figure 15(a) which clearly shows two regions with rare failure samples. The failure occurs for asymmetrical local $V_{th}$ variation affecting the adjacent pulling-down transistors $M1$ and $M2$. A similar localization of the failure region is reached by the proposed yield analysis scheme as it can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



Table V: Yield results for the Op-amp with 56 process parameters

| Perf metrics | Brute-force MC | MC+LHS | Quasi MC | Our method | | | |
|---|---|---|---|---|---|---|---|
| | Sim(♯) | Sim(♯) | Sim(♯) | Sim(♯) | Sim Time | Fitting/Verif Time | Time |
| $Av$ | 8740 | 6650 | 6420 | 300/135 | 0.28h | 96s/3s | 0.26h |
| $GBW$ | 8740 | 6650 | 6420 | 300/119 | | 5s/4s | |
| $DCOffset$ | 8740 | 6650 | 6420 | 300/69 | | 6s/3s | |
| $PM$ | 8740 | 6650 | 6420 | 300/201 | | 91s/6s | |
| Time Cost | 2.97h | 2.26h | 2.18h | 0.60h | | | |
| Speedup | 1X | 1.32X | 1.37X | 5X | | | |
| Yield (%) | 81.61 | 79.60 | 81.6 | 80.53 | | | |
| Relative Error | - | 2.46% | 1.24% | 1.32% | | | |

observed in Figure 15(b). In both figures, the simulation data is projected on the three directions ($V_{thM1}, V_{thM2}, V_{thM5}$) for visualization purpose. The proposed failure regions localization technique neutralizes the rare failure event issue of the SRAM circuit. Based on this example, the advantage of the proposed method in locating very rare failure regions has been demonstrated.

### C. Three-stage Operational Amplifier

In this section, we will verify that the proposed method is suitable for solving problems with multiple performances specifications as well as high dimensional parameters space. We consider a three-stage amplifier (op-amp) as shown in Figure 16.

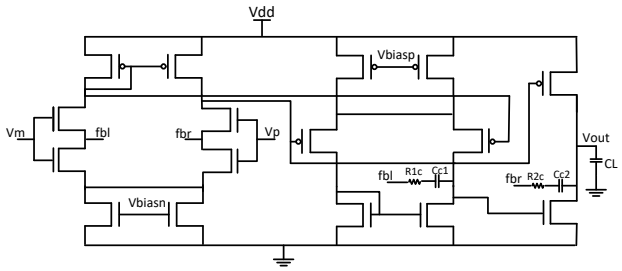


Figure 16: A Three-stage operational amplifier

We select $\Delta t_{ox}$, $\Delta V_{th}$, $\Delta w$ and $\Delta L$ as the process variables. The local mismatch in each transistor pairs is considered. It leads to a total of 56 process parameters. The performance of the circuit is characterized by many properties, such as voltage gain ($Av$) and phase margin ($PM$). The op-amp is designed to satisfy the list of specifications shown in Table VI.

Table VI: The set of specifications for the three-stage op-amp

| Perf metrics | Simulation | Specification |
|---|---|---|
| $Av(dB)$ | AC | $\geq 40$ |
| $GBW(MHz)$ | AC | $\geq 80$ |
| $DCOffset(mV)$ | DC | $\leq 50$ |
| $PM(^\circ)$ | AC | $\geq 60$ |

Firstly, 300 initial LHS simulations are used to build a surrogate model of for all properties. 200 of them are employed for model training and 100 for subsequent model testing. Each property is measured using a specific type of simulation. Note that even though we analyze and model each performance metric individually, these performance metrics are not necessarily independent as they are sharing the transistor-level simulations of the pre-sampling stage. In fact, by evaluating all performance metrics for each individual sample drawn from the process parameters space, we substantially reduce the total number of simulation runs and, hence, the computational cost.

Table VII: Result of the process parameters reduction stage

| Perf metrics | Reduced Set (♯) | Reduction Error |
|---|---|---|
| $Av(dB)$ | 32 | 0.79% |
| $GBW(MHz)$ | 24 | 0.95% |
| $DCOffset(mV)$ | 40 | 0.85% |
| $PM(^\circ)$ | 44 | 0.95% |

On this 56-dim problem, RReliefF is performed to reduce the dimension the process parameters. The experimental results of the reduction process are summarized in Table VII. It can be observed that in this example the dimension of the original set of process parameters for each performance metric did not largely decrease. This can be explained by the consideration of multiple performance metrics that depend on most of the process variables. Furthermore, the accuracy of the circuit response under the reduced set of process parameters is maintained.

Table VIII: Surrogate models degree and accuracy (1-NMSE)%

| Perf metrics | Degree | Model Accuracy | Gen-Accuracy |
|---|---|---|---|
| $Av$ | 3 | 98.0% | 97.8% |
| $GBW$ | 1 | 98.1% | 98.05% |
| $DCOffset$ | 1 | 98.8% | 98.3% |
| $PM$ | 3 | 98.7% | 98.2% |

We evaluate the accuracy of the models trained using the adaptive LASSO scheme. We report the final degree of the approximations and the models accuracies in Table VIII. In the "Degree" column, we see that for some properties, we are able to construct polynomial models with a degree lower than the limit $D = 3$. The accuracy generalization step statistically verify the op-amp properties model with respect to the reduced set of process parameters. In the column "Gen-Accuracy", we report the result of the accuracy generalization stage. We can find that the accuracy is more than $97\%$ for all models.

We apply the brute-force MC, Quasi MC and MC+LHS to estimate the yield of the op-amp. The brute-force MC method is run with a target accuracy of $99\%$ and a confidence level of $95\%$. For the sampling methods, "Time Cost" is the circuit simulation time and "Sim(♯)" refers to the number of samples. The column "Sim(♯)" in our method includes the number of circuit simulations performed in the surrogate model fitting and accuracy verification phases. "Sim Time" shows the total circuit simulation time and "Fitting/Verif Time" indicates the time spent in the model fitting and verification stages excluding the circuit simulation time. "Time" is the time spent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



in the parameter space exploration and the yield calculation. Finally, "Time Cost" is the total computational time.

As in the previous experiments, we observe that the predicted yield from our approach closely matches the yield estimation of the MC method. Our method requires fewer simulations and finishes faster with a speedup of almost 5X. This application shows again the benefits of a model building approach rather than direct yield estimation from a circuit simulator. Also, the column "Fitting/Verif Time" in our method shows that even though the reduction result was not very significant, the proposed adaptive sparse regression algorithm still renders the fitting time quite affordable. This result further demonstrates the scalability of the proposed technique to handle larger problems. The regression time of the model performance with a degree lower than the degree limit (i.e., $GBW$ and $DCOffset$) is significantly smaller. In fact, the major cost in regression lies in the computation of the LASSO coefficients. The former can be easily parallelized, leading to further performance improvements.

### D. Parameters Discussion

In this section, the key parameters in the proposed method will be discussed.

*1) Parameter $R_{th}$ in Algorithm 1:* We construct the frequency model of the Ring Oscillator example in Section IV-A with different $R_{th}$ from $9.10^{-2}$ to $1.10^{-2}$. Figure 17 shows the error of the yield rate with respect to the model accuracy defined as $(1 - R_{th})\%$. The error of the yield is computed relatively to the yield result of 10000 MC simulations run. We can find that when the accuracy is smaller than $97\%$, the relative error resulting primarily from the fitting error of the frequency model increases signicantly. To ensure the viability of the proposed method, we must ensure that the accuracy is high enough at the modeling stage. So, in practice, the value of $R_{th}$ should be selected from $3.10^{-2}$ to $1.10^{-2}$.

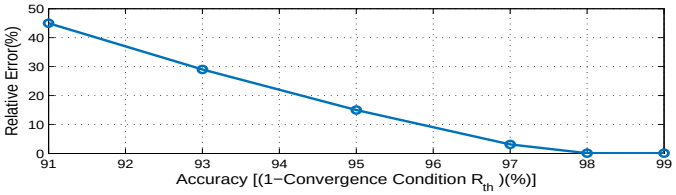


Figure 17: Relative error with respect to $R_{th}$

*2) Parameters $(\alpha, \beta, \varphi)$ in Algorithm 2:* We applied Algorithm 2 to verify and generalize the frequency model accuracy $freq(\tilde{p})$ of the Ring Oscillator example in Section IV-A. We checked the property:

$$\forall p, \tilde{p} \in P \quad Pr((err(f(p), freq(\tilde{p})) \leq \varepsilon) \geq \quad \varphi \quad (15)$$

where $\varepsilon = 0.0135$ (i.e., $98.65\%$ accuracy). We applied the algorithm for different values of $\varphi$ and equal error rates $(\alpha, \beta)$. We used an indifference region $[\varphi - \delta, \varphi + \delta]$ where $\delta = 0.001$. The results are summarized in Table IX. Increasing $\varphi$ and decreasing $(\alpha, \beta)$ requires a larger number of simulations, leading to a model verification with better statistical guarantee. The model accuracy has been verified and generalized to $0.019$ (i.e., $98.1\%$ accuracy). In practice, we find that $\varphi = 0.95$ and $\alpha = \beta = 0.01$ often provide a good trade-off between statistical guarantee and computational cost.

Table IX: Run length for common values of $\varphi$ and $(\alpha, \beta)$

| $\alpha(=\beta)$ | 0.02 | 0.01 | $10^{-3}$ |
|---|---|---|---|
| $\varphi$=0.9 | 37 | 44 | 65 |
| 0.95 | 75 | 160 | 214 |
| 0.99 | 683 | 762 | 1015 |

*3) Tolerance margin in failure performance bounds:* If the circuit specification includes a performance metric $f$ that should be greater than a limit $f_{limit}$ (i.e., $f > f_{limit}$), then the failure performance region is defined as $f \in [f^l, f_{limit}]$. If the value $f^l$ is over-approximated (i.e., it is below the value that can be reached in reality), it will not affect the result of the yield estimation and it will slightly affect the computation time. In fact, the SMT solver rapidly discards parts from the search space that contains no solutions. However, if it is under-approximated (i.e., it is greater than the value that can be reached in reality), it will prevent the SMT solver from locating failure regions in the parameters space and affect the final yield estimation. In practice, we firstly set $f_l = f_{min} - \Delta f$, where $\Delta f = |f^{min} - f_{nom}|$, $f_{nom}$ is the nominal value of $f$ and $f^{min}$ is the minimum value of $f$ discovered during the initial pre-sampling and circuit simulation step. If the SMT solver discovers failure regions in the parameters space with performance values $f$ in the neighborhood of $f_l$, that is $f \in [f_l, f_l + 3\varepsilon]$, where $\varepsilon$ is the model error, then $f_l$ should be further decreased by $\Delta f$. Otherwise, the user can be highly assured that the failure performance bounds have been conservatively characterized.

## V. Conclusion

This paper presented a methodology for analog circuits yield analysis. Different techniques such as parameters pruning, adaptive sparse regression and sequential probability ratio test were used to build performances models and verify their accuracy. We then employed an SMT solving technique and interval arithmetic to exhaustively probe the parameters space and locate the failure regions of the circuit operation. The yield is calculated based on the probabilistic volume of the located failure regions. Compared with existing methods, the proposed method tried to handle yield problems with: (1) many process parameters; (2) multiple and distinct failure regions; (3) multiple performances specification; and (4) extremely high yield rate. The experimental results on several analog circuits show that the presented method is reliable while leading to a simulation speedup when compared to the brute-force MC.

The proposed method enhanced the run-time and scalability of SMT solving techniques by adopting multiple strategies including: (1) reduction of the SMT problem variables; (2) building low complex performances models; (3) reduction of the SMT problem search space; and (4) adjustment of the SMT solver resolution and solution refinement. Furthermore, the computational cost of the proposed surrogate modeling algorithm has been enhanced by reducing the number of process parameters and avoiding high polynomial degree.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2017.2651807, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



Also, note that the computational cost of the modeling algorithm may increase if the number of process parameters and the number of performance metrics largely increases. For example, in the case where the dimensionality is extremely high, the adaptive regression must choose a set of important polynomial terms from numerous (e.g., millions of) possible candidates and, hence, the surrogate model training algorithm described in this paper may become computationally unaffordable. In our future research, we will further study more efficient heuristics and parallelization techniques that may address this issue. We also plan to integrate the yield estimation method with the nominal sizing method proposed in [17] to further prove its usefulness in analog design.

## REFERENCES


[1] M. Wirnshofer. *Variation-Aware Adaptive Voltage Scaling for Digital CMOS Circuits*. Springer, 2013.

[2] B. Liu, F. V. Fernandez, and G. G. E. Gielen. Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(6):793–805, June 2011.

[3] T. Mukherjee, L. R. Carley, and R. A. Rutenbar. Efficient handling of operating range and manufacturing line variations in analog cell synthesis. *IEEE Transactions on Computeer-Aided Design of Integrated Circuits and Systems*, 19(8):825–839, Aug 2000.

[4] C. Jacoboni and P. Lugli. *The Monte Carlo Method for Semiconductor Device Simulation*. Springer Vienna, 19989.

[5] S. Sun, Y. Feng, C. Dong, and X. Li. Efficient sram failure rate prediction via gibbs sampling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(12):1831–1844, 2012.

[6] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992.

[7] J. Yao, Z. Ye, and Y. Wang. Importance boundary sampling for sram yield analysis with multiple failure regions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(3):384–396, March 2014.

[8] F. Gong, H. Yu, Y. Shi, D. Kim, J. Ren, and L. He. Quickyield: An efficient global-search based parametric yield estimation with performance constraints. *Design Automation Conference*, pages 392–397, 2010.

[9] X. Li. Finding deterministic solution from underdetermined equation: Large-scale performance variability modeling of analog/rf circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11):1661–1668, Nov 2010.

[10] M. Fronzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.

[11] iSAT3. iSAT3: Tight integration of satisfiability and constraint solving, December 2014.

[12] C. Gu and J. Roychowdhury. *Yield Estimation by Computing Probabilistic Hypervolumes. In: Extreme Statistics in Nanoscale Memory Design*, pages 137–177. Springer, 2010.

[13] J. Jaffari and M. Anis. On efficient lhs-based yield analysis of analog circuits. *Computer-Aided Design of Integrated Circuits and Systems*, 30(1):159–163, 2011.

[14] A. Singhee and R. A. Rutenbar. Why quasi-monte carlo is better than monte carlo or latin hypercube sampling for statistical circuit analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11):1763–1776, Nov 2010.

[15] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu. Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(7):1096–1109, July 2015.

[16] L. Yin, Y. Deng, and P. Li. Simulation-assisted formal verification of nonlinear mixed-signal circuits with bayesian inference guidance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(7):977–990, July 2013.

[17] O. Lahiouel, M. H. Zaki, and S. Tahar. Towards enhancing analog circuits sizing using SMT-based techniques. *Design Automation Conference*, pages 1–6, 2015.

[18] H. Lin and P. Li. Parallel hierarchical reachability analysis for analog verification. *Design Automation Conference*, pages 1–6, 2014.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[20] Y. Zhang, S. Sankaranarayanan, and F. Somenzi. Sparse statistical model inference for analog circuits under process variations. *Asia and South Pacific Design Automation Conference*, pages 449–454, 2014.

[21] M. Robnik-Sikonja and I. Kononenko. An adaptation of Relief for attribute estimation in regression. *International Conference on Machine Learning*, pages 296–304, 1997.

[22] K. Kira and L. A. Rendell. A Practical Approach to Feature Selection. *International Conference on Machine Learning*, pages 249–256, 1992.

[23] H. Zou. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[24] MATLAB. Documentation center, July 2016.

[25] H. L. S. Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2005.

[26] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 06 1945.

[27] S. M. Rump. *Developments in Reliable Computing*. Springer, 1999.

[28] P.R. Bevington and D.K. Robinson. *Data reduction and error analysis for the physical sciences*. McGraw-Hill, 2003.

[29] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan. Breaking the simulation barrier: Sram evaluation through norm minimization. *International Conference on Computer-Aided Design*, pages 322–329, 2008.

[30] E. Seevinck, F. J. List, and J. Lohstroh. Static-noise margin analysis of mos sram cells. *IEEE Journal of Solid-State Circuits*, 22(5):748–754, 1987.


**Ons Lahiouel** (S'17) received the Bachelors and Masters degrees in Telecommunication Systems from the National Engineering School of Tunis, Tunisia, in 2011 and 2012, respectively, and the Ph.D. degree in electrical engineering from Concordia University, Montréal, QC, Canada, in 2017. She pursued her graduate studies in the field of analog and mixed signal systems in the Hardware Verification Group, Concordia University, under the supervision of Prof. S. Tahar. Her current research interests include design and verification of analog and mixed-signal integrated circuits and hardware verification.

**Mohamed H. Zaki** (M'05) received the bachelors degree in electrical engineering from Ain Shams University, Cairo, Egypt, in 2000, and the masters and Ph.D. degrees from Concordia University, Montréal, QC, Canada, in 2002 and 2008, respectively. He continued his graduate studies with the Hardware Verification Group, Concordia University, under the supervision of Prof. S. Tahar. He then conducted a post-doctoral training with the Department of Computer Science, University of British Columbia, Vancouver, BC, Canada. He has been a Consultant at the start-up company Innovia Technologies, Vancouver, BC, Canada, since 2010. He is currently doing research in intelligent transportation systems at the University of British Columbia.

**Sofiène Tahar** (M'96-SM'07) received the Diploma degree in computer engineering from the University of Darmstadt, Darmstadt, Germany, in 1990, and the Ph.D. (Hons.) degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 1994. He is currently a Professor and Research Chair in formal verification of Systems-on-Chip with the Department of Electrical and Computer Engineering, Concordia University, Montréal, QC, Canada. He has made contributions and published papers in the areas of formal hardware verification, system-on-chip verification, AMS circuit verification, and probabilistic, statistical, and reliability analysis of systems. He is also the Founder and Director of the Hardware Verification Group with Concordia University. Dr. Tahar is a Senior Member of the Association for Computing Machinery and a Professional Engineer in the Province of Quebec. He was named University Research Fellow upon receiving Concordia Universitys Senior Research Award in 2007.