Advanced Statistical Methodologies for Tolerance Analysis in Analog Circuit Design

Bruno Apolloni[1], Simone Bassis[2], Angelo Ciccazzo[3], Angelo Marotta[4],
Salvatore Rinaudo[5] and Orazio Muscato[6]
[1,2]*Department of Computer Science, University of Milan,*
Via Comelico 39/41, 20135 Milano
[3,4,5]*STMicroelectronics, Stradale Primo Sole 50, 95121 Catania*
[6]*Department of Mathematics and Informatics, University of Catania,*
Viale Andrea Doria 6, 95125 Catania
Italy

1. Introduction

The influence of process variations is becoming extremely critical for nano technology nodes (90nm and below), due to geometric tolerances and manufacturing non-idealities (such as edge or surface roughness, or the fluctuation in the number of doping atoms). The most worrying of all is the statistical variability introduced by discreteness of charge and granularity matter in the transistors approaching molecular and atomic scale dimensions. The main sources of statistical variability are the random distributions of discrete dopants and charged defects, the line edge roughness of the photo resist and the granularity of the materials (Bernstein et al., 2006; Boning & Nassif, 1999). As a result, production yields and circuit figures of merit (such as performance, power, and reliability) have became extremely sensitive to incontrollable statistical process variations (PV). The main sources of variations are: environmental factors, whose transient arises during the operation of a circuit (e.g. power supply or temperature variations), and physical factors due to the manufacturing process, which result in a (permanent or aging) variation of the device structure and interconnections. The latter reflect into random (possibly spatial) drifts of the design parameter.

Although already considered in the past, the increasing impact of these drawbacks constitutes a completely new challenge. While process engineers have traditionally coped with die-to-die fluctuations, the today within-die variations are more subtle since they imply that different areas of the same die exhibit different values of the various parameters. With a further shrinking of process technology, the on-chip variation is getting worse for each technology node, thus having a direct impact on the design flows. By contrast, the latter conventionally rely on deterministic models.

At a front end, parameter variability has a significant impact both on the power dissipation and performance of a circuit, with a consequent yield decrease and remarkable cost implications. Indeed, to maintain production efficiency we must raise up control costs and cycle time, a drawback which dramatically increases with the process complexity. To contrast it, the following two joint tasks become essential:

- to characterize statistically integrated circuits (IC) manufacturing process fluctuations;
- to predict reliably circuit performance spreads at the design stage.

Failure in the former can result in a low parametric yield, since ICs do not meet design specifications. On the one hand, a successful statistical characterization promotes a robust manufacturability reflecting in a high fabrication yield (i.e. a high proportion of produced circuits which function properly). On the other hand, it requires managing complex design flows in the design-verification-production life-cycle of ICs.

Summing up, random and systematic defects as well as parametric process variations have a big influence on the design/production cycle, causing frequent re-spinning of the whole development and manufacturing chain. This leads to high costs of multiple manufacturing runs and entails extremely high risks of missing a given market window. One way to overcome these drawbacks is to implement the DFM/DFY paradigm (Bühler et al., 2006) where Design for Manufacturability (DFM) mates Design for Yield (DFY) to form a synergistic manufacturing chain to be dealt with in terms of: i) relationships between the statistical circuit parameters matching the production constraints, and ii) performance indicators ensuring correctly functioning dies. This chapter introduces a pair of procedures aimed at identifying these parameters exactly with the goal of maximizing performance indicators defined as a function of the parameters' confidence region.

The material is organized as follows. In Section 2 we discuss the statistical aspect of IC design and introduce the lead formalism. In Section 3 we focus on the statistical modeling task with special regard to two advanced solution methods. Hence we introduce benchmarks in Section 4 to both provide a comparison between the performances of the above methods and show their behaviors w.r.t. state-of-the-art procedures introduced by researchers in the last years. Concluding remarks are drawn in the final section.

2. Statistics in IC design

Electronic devices are replicated multiple times on a wafer and different wafers are produced, but each device *cannot* be produced in the same way in terms of electrical performance. Main factors that make the fabrication result uncertain are: the imperfections characterizing the masks and tolerances in their positionings, various changing effects of ion plant temperature during production, tolerances in size, etc. Generally *fluctuations' processes* produce *fluctuations in electrical performance*. Consequently, an essential tool for electronic circuit designing is represented by the statistical model which formally relates the former to the latter.

A circuit is classified as *acceptable* in performances if *all* specifications on its electrical behavior are met. In the context of the microelectronics industry, the term *yield* phrases the ratio between the number of acceptable chips and total number of produced chips:

$$\text{yield} = \frac{\#\text{ accetable chips}}{\#\text{ manufactured chips}} \tag{1}$$

The acceptability of each chip is decreed by checking that the questioned electrical parameters individually fall into tolerance intervals. In addition, each wafer contains several sites with special test structures that enable further performance measurements in order to verify the manufacturing process. All the measurements are collected in a database which statistically characterizes the electrical behavior of the devices.

As for the final product we may classify the integrated circuits into:

- *acceptable chips*, which satisfy *all* performance requests,

- *functional failures*, when malfunctions affect chips,
- *parametric failures*, when chips fail to reach performances.

Coming to their manufacturing, we are used to distinguish three categories of failures that we synthesize through:

2.1. *random yield (sometimes called statistical yield)*, concerning the random effects occurring during the manufacturing process, such as catastrophic faults in the form of open or short circuits. These faults may be a consequence of small particles in the atmosphere landing on the chip surface, no matter how clean is the wafer manufacturing environment. An example of a random component is that of threshold voltage variability due to random dopant fluctuations (Stolk et al., 1988);

2.2. *systematic yield (including printability issues)*, related to systematic manufacturability issues deriving from combinations and interactions of events that can be identified and addressed in a systematic way. An example of these events is the variation in wire thickness with layout density due to Chemical Mechanical Polishing/Planarization (CMP) (Chang et al., 1995). The distinction from the previous yield is important because the impact of systematic variability can be removed by adapting the design appropriately, while random variability will inevitably impact design margins in a negative manner;

2.3. *parametric yield (including variability issues)*, dealing with the performance drifts induced by changes in the parameter setting – for instance, lower drive capabilities, increased leakage current and greater power consumption, increased resistance and capacitance (RC) time constants, and slower chips deriving from corruptions of the transistor channels.

From a complementary perspective, the unacceptable performance causes for a circuit may be split into two categories of disturbances:

- *local*, caused by disruption of the crystalline structure of silicon, which typically determines the malfunctioning of a single chip in a silicon wafer;
- *global*, caused by inaccuracies during the production processes such as misalignment of masks, changes in temperature, changes in doses of implant. Unlike the local disturbance, the global one involves all chips in a wafer at different degrees and in different regions. The effect of this disturbance is usually the failure in the achievement of requested performances, in terms of working frequency decrease, increased power consumption, etc.

Both induce troubles on physical phenomena, such as electromagnetic coupling between elements, dissipation, dispersion, and the like.

The obvious goal of the microelectronics factory is to maximize the yield as defined in (1). This translates, from an operational perspective, into a *design target* of properly sizing the circuit parameters, and a *production target* of controlling their realization. Actually both targets are very demanding since the involved parameters $\boldsymbol{\pi}$ are of two kinds:

- *controllable*, when they allow changes in the manufacturing phase, such as the oxidation times,
- *non controllable*, in case they depend on physical parameters which cannot be changed during the design procedure, like the oxide growth coefficient.

Moreover, in any case the relationships between $\boldsymbol{\pi}$ and the parameters ϕ characterizing the circuit performances are very complex and difficult to invert. This induces researchers to model both classes of parameters as vectors of random variables, respectively $\boldsymbol{\Pi}$ and

$\mathbf{\Phi}$ [1]. The corresponding problem of *yield maximization* reverts into a functional dependency among the problem variables. Namely, let $\mathbf{\Phi} = (\Phi_1, \Phi_2, \ldots, \Phi_t)$ be the vector of the performances determined by the parameter vector $\mathbf{\Pi} = (\Pi_1, \Pi_2, \ldots, \Pi_n)$, and denote with $D_{\mathbf{\Phi}}$ the acceptability region of a given chip. For instance, in the common case where each performance is checked singularly in a given range, i.e.:

$$\phi_k^l \leq \Phi_k \leq \phi_k^u \quad k = 1, \ldots, t \tag{2}$$

$D_{\mathbf{\Phi}}$ reads:

$$D_{\mathbf{\Phi}} = \left\{ \mathbf{\Phi} | \phi_k^l \leq \Phi_k \leq \phi_k^u \quad k = 1, \ldots, t \right\} \tag{3}$$

The yield goal is the maximization of the probability $\mathscr{P}$ that a manufactured circuit has an acceptable performance, i.e.

$$\mathscr{P} = \mathrm{P}\left[\mathbf{\Phi} \in D_{\mathbf{\Phi}}\right] = \int_{D_{\mathbf{\Phi}}} f_{\mathbf{\Phi}}(\phi) d\phi \tag{4}$$

where $f_{\mathbf{\Phi}}$ is the joint probability density of the performance $\mathbf{\Phi}$.

To solve this problem we need to know $f_{\mathbf{\Phi}}$ and manage its dependence on $\mathbf{\Pi}$. Namely, methodologies for maximizing the yield must incorporate tools that determine the region of acceptability, manipulate joint probabilities, evaluate multidimensional integrals, solve optimization problems. Those instruments that use *explicit* information about the joint probability and calculate the yield multidimensional integral (4) during the maximization process are called *direct methods*. The term *indirect* is therefore reserved for those methods that do not use this information directly. In the next section we will introduce two of these methods which look to be very promising when applied to real world benchmarks.

3. Statistical modeling

As mentioned in the introduction, a main way for maximizing yield passes through mating Design for Manufacturability with Design for Yield (DFM/DFY paradigm) along the entire manufacturing chain. Here we focus on model parameters at an intermediate location in this chain, representing a target of the production process and the root of the circuit performance. Their identification in correspondence to a performances' sample measured on produced circuits allows the designer to get a clear picture of how the latter react to the model parameters in the actual production process and, consequently, to grasp a guess on their variation impact. Typical model and performance parameters are described in Table 1 in Section 4.

In a greater detail, the first requirement for planning circuits is the availability of a model relating input/output vectors of the function implemented by the circuit. As aforementioned, its achievement is usually split into two phases directed towards the search of a couple of analytic relations: the former between model parameters and circuit performances, and the latter, tied to the process engigneers' experience, linking both design and phisical circuit parameters as they could be obtained during production. Given a wafer, different repeated measurements are effected on dies in a same circuit family. As usual, the final aim is the model

[1] By default, capital letters (such as X, Y) will denote random variables and small letters (x, y) their corresponding realizations; bold versions $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{y})$ of the above symbols apply to vectors of the objects represented by them. The sets the realizations belong to will be denoted by capital gothic symbols $(\mathfrak{X}, \mathfrak{Y})$.

identification, in terms of designating the input (respectively output) parameter values of the aforementioned analytical relation. In some way, their identification hints at synthesizing the overall aspects of the manufacturing process not only to use them satisfactory during development yet to improve oncoming planning and design phases, rather than directly weigh on the production.

For this purpose there are three different perspectives: synthesize simulated data, optimize a simulator, and statistically identify its optimal parameters. All three perspectives share the following common goals: ensure adequate manufacturing yield, reduce the production cost, predict design fails and product defects, and meet *zero defects* specification. We formalize the modeling problem in terms of a mapping $\boldsymbol{g}$ from a random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$, describing what is commonly denoted as model parameters [2], to a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_t)$, representing a meaningful subset of the performances $\boldsymbol{\Phi}$. The statistical features of $\boldsymbol{X}$, such as mean, variance, correlation, etc., constitute its parameter vector $\boldsymbol{\theta}_{\boldsymbol{X}}$, henceforth considered to be the statistical parameter of the input variable $\boldsymbol{X}$. Namely, $\boldsymbol{Y} = \boldsymbol{g}(\boldsymbol{X}) = (g_1(\boldsymbol{X}), \ldots, g_t(\boldsymbol{X}))$, and we look for a vector $\boldsymbol{\theta}_{\boldsymbol{Y}}$ that characterizes a performance population where $\mathrm{P}(\boldsymbol{Y} \in \widehat{D}_{\boldsymbol{Y}}) = \alpha$, having denoted with $\widehat{D}_{\boldsymbol{Y}}$ the *α-tolerance region*, i.e. the domain spanned by the measured performances, and with α a satisfactory probability value. In turn, $\widehat{D}_{\boldsymbol{Y}}$ is the statistic we draw from a sample $s_{\boldsymbol{y}}$ of the performances we actually measured on correctly working dies. Its simplest computation leads to a rectangular shape, as in (3), where we independently fix ranges on the singular performances. A more sophisticated instance is represented by the convex hull of the jointly observed performances in the overall $\mathfrak{Y}$ space (Liu et al., 1999). At a preliminary stage, we often appreciate the suitability of $\boldsymbol{\theta}_{\boldsymbol{Y}}$ by comparing first and second order moments of a performances' population generated through the currently identified parameters with those computed on $s_{\boldsymbol{y}}$.

As a first requisite, we need a comfortable function relating the $\boldsymbol{Y}$ distribution to $\boldsymbol{\theta}_{\boldsymbol{X}}$. The most common tool for modeling an analog circuit is represented by the Spice simulator (Kundert, 1998). It consists of a program which, having in input a textual description of the circuit elements (transistors, resistors, capacitors, etc.) and their connections, translates this description into nonlinear differential equations to be solved using implicit integration methods, Newton's method and sparse matrix techniques. A general drawback of Spice – and circuit simulators in general – is the complexity of the transfer function it implements to relate physical parameters to performances which hampers intensive exploration of the performance landscape in search of optimal parameters. The methods we propose in this section are mainly aimed at overtaking the difficulty of inverting this kind of functions, hence achieving a feasible solution to the problem: find a $\boldsymbol{\theta}_{\boldsymbol{X}}$ corresponding to the wanted $\boldsymbol{\theta}_{\boldsymbol{Y}}$.

3.1 Monte Carlo based statistical modeling

The lead idea of the former method we present is that the model parameters are the output of an optimization process aimed at satisfying some performance requirements. The optimization is carried out by wisely exploring the research space through a Monte Carlo (MC) method (Rubinstein & Kroese, 2007). As stated before, the proposed method uses the experimental statistics both as a target to be satisfied and, above all, as a selectivity factor for device model. In particular, a device model will be accepted only if it is characterized by parameters' values that allow to obtain, through electrical simulations, some performances which are included in the tolerance region.

[2] We speak of $\boldsymbol{X}$ as *controllable* model parameters to be defined as a suitable subset of $\boldsymbol{\Pi}$.

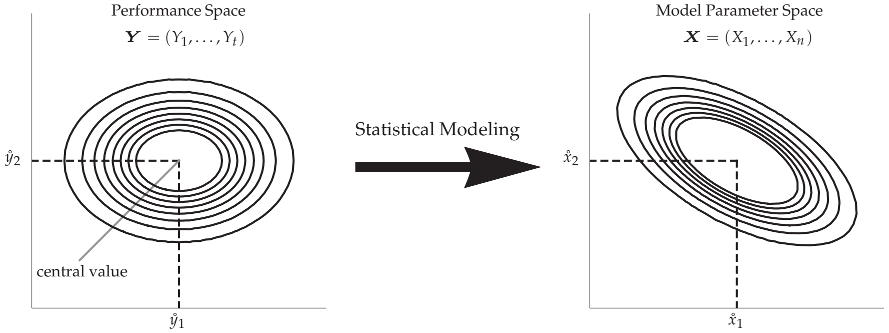


Fig. 1. Proposed flow: from the experimental statistics we determine a statistical Spice model for the device.

The aim of the proposed flow is the following: on the basis of the information which constitutes the experimental statistics, we want to map the space $\mathfrak{Y}$ of the performances (such as gain and bandwidth) to the space $\mathfrak{X}$ of circuit parameters (such as Spice parameters or circuit components values), as outlined in Fig. 1. Variations in the fabrication process cause random fluctuations in $\mathfrak{Y}$ space, which in turn cause $\boldsymbol{X}$ to fluctuate (Koskinen & Cheung, 1993). In other words, we want to extract a Spice model whose parameters are random variables, each one characterized by a given probability distribution function. For instance, in agreement with the Central Limit Theorem (Rohatgi, 1976), we may work under usual Gaussianity assumptions. In this case, for the model parameters which have to be statistically described, it is necessary and sufficient to identify the mean values, standard deviations and correlation coefficients. In general, the flow of statistical modeling is based on several MC simulation steps (strictly related to bootstrap analysis (Efron & Tibshirani, 1993)), in order to estimate unknown features for each statistical model parameter. The method will proceed by executing iteratively the following steps, in the same way as in a multiobjective optimization algorithm, where the targets to be identified are the optimal parameters $\boldsymbol{\theta}_{\boldsymbol{X}}$ of the model.
In the following procedure, general steps (described in roman font) will be specialized to the specific scenario (in italics) used to perform simulations in Section 4.

Step 1. Assume a *typical* (*nominal*) device model m_0 is available, whose model parameters' means are described by the vector $\mathring{\boldsymbol{\nu}}_{\boldsymbol{X}}$ (central values). Let $\widehat{D}_{\boldsymbol{Y}}$ be the corresponding *typical* tolerance region estimated on $\boldsymbol{Y}$ observations $s_{\boldsymbol{y}}$. Choose an initial guess of $\boldsymbol{X}$ joint distribution function on the basis of moments estimated on given $\boldsymbol{X}$ observations $s_{\boldsymbol{x}}$. Let $\mathscr{M}$ denote the companion device statistical model, and set $k = 0$.

In the specific case of hyper-rectangle tolerance regions defined as in (3), let $\mathring{\nu}_{Y_j} \pm 3\mathring{\sigma}_{Y_j}, j = 1, \ldots, t$ denote the two extremes delimiting each admissable performance interval. Moreover, since model parameters $\boldsymbol{X}$ of $\mathscr{M}$ follows a multivariate Gaussian distribution, assume (in the first iteration) a null cross-correlation between $\{X_1, \ldots, X_n\}$, hence $\theta_{X_i} = \{\nu_{X_i}, \sigma_{X_i}\}, i = 1, \ldots, n$, where by default $\nu_{X_i} = \mathring{\nu}_{X_i}$, i.e. the same mean as the nominal model is chosen as initial value, and σ_{X_i} is assigned a relatively high value, for instance set equal to the double of the mean value.

Step 2. At the generic iteration k, an m-sized [3] sample $s_{\mathscr{M}_k} = \{\boldsymbol{x}_r\}, r = 1 \ldots, m$ will be generated according to the actual $\boldsymbol{X}$ distribution.

[3] A generally accepted rule to assign m is: for an expected probability level $10^{-\xi}$, the sample size m should be set in the range $[10^{\xi+2}, 10^{\xi+3}]$ (Johnson, 1994).

In particular, when X_i are nomore independent, the discrete Karhunen-Loeve expansion (Johnson, 1994) is adopted for sampling, starting from the actual covariance matrix.

Step 3. For each model parameter $\boldsymbol{x}_r$ in $s_{\mathscr{M}_k}$, the target performances $\boldsymbol{y}_r$ will be calculated through Spice circuit simulations.

Step 4. Only those model parameters in $s_{\mathscr{M}_k}$ reproducing performances lying within the chosen tolerance region $\widehat{D}_{\boldsymbol{Y}}$ will be accepted. On the basis of this criterion a subsample $s_{\widetilde{\mathscr{M}}_k}$ of $s_{\mathscr{M}_k}$ having size $m' \leq m$ will be selected.

In particular, by keeping a fraction $1 - \delta$, say 0.99, of those models having all performance values included in $\widehat{D}_{\boldsymbol{Y}}$, we are guaranteeing a confidence region of level δ under i.i.d. Gaussianity assumptions.

Step 5. On the basis of the subsample $s_{\widetilde{\mathscr{M}}_k'}$, a new model $\mathscr{M}_k'$ will be computed through standard statistical techniques.

For each model parameter $X_i, i = 1, \ldots, n$, the n standard deviations could be computed on the sample $s_{\widetilde{\mathscr{M}}}$ through Maximum Likelihood Estimators (MLE) (Mood et al., 1974), Spearman Rank-Order correlation coefficient (Lehmann, 2006; Press et al., 1993) may be used to estimate cross-correlation, while, according to circuit designers' report, the n means will be kept equal to the nominal $\mathring{v}_{X_i}, i = 1, \ldots, n$.

Step 6. If the number $\widetilde{m}$ of selected model parameters which have generated $\mathscr{M}'$ is sufficiently high (for instance they constitute a fraction $1 - \delta$, let's say 0.99, of the m instances, then the algorithm stops returning the statistical model $\mathscr{M}'$. Otherwise, set $k = k + 1$ and goto Step 2.

The iterative procedure described above is based on *Attractive Fixed Point* method (Allgower & Georg, 1990), where the optimal value of those features to be estimated represents the *fixed point* of the algorithm. When the number of the components significantly increases, the convergence of the algorithm may become weak. To manage this issue, a two-step procedure is introduced where the former phase is aimed at computing moments involving single features X_i while maintaining constant their cross-correlation; the latter is directed toward the estimation of the cross-correlation between them. The overall procedure is analogous to the previous one, with the exception that cross-correlation terms will be kept fixed until Step 5 has been executed. Subsequently, a further optimization process will be performed to determine the cross-correlation coefficients, for instance using the *Direct* method as described in Jones et al. (1993). The stop criterion in Step 6 is further strengthen, prolonging the running of the procedure until the difference between cross-correlation vectors obtained at two subsequent iterations will drop below a given threshold.

3.2 Reverse spice based statistical modeling

A second way we propose to bypass the complexity handicap of Spice functions passes through a principled philosophy of considering the region $D_{\boldsymbol{X}}$ where we expect to set the model parameters as an aggregate of fuzzy sets in various respects (Apolloni et al., 2008). First of all we locally interpolate the Spice function $\boldsymbol{g}$ through a polynomial, hence a mixture of monomials that we associate to the single fuzzy sets. Many studies show this interpolation to be feasible, even in the restricted form of using posynomials, i.e. linear combination of monomials through only positive coefficients (Eeckelaert et al., 2004). The granular construct we formalize is the following.

> Given a Spice function g mapping from $\boldsymbol{x}$ to y (the generic component of the performance vector $\boldsymbol{y}$), we assume the domain $D_{\boldsymbol{X}} \subseteq \mathbb{R}^n$ into which $\boldsymbol{x}$ ranges to be the support of c fuzzy sets $\{A_1, \ldots, A_c\}$, each pivoting around a monomial m_k. We consider this monomial to be a local interpolator that fits g well in a surrounding of the A_k centroid. In synthesis, we have $g(\boldsymbol{x}) \simeq \sum_{k=1}^{c} \mu_k(\boldsymbol{x})\mathsf{m}_k(\boldsymbol{x})$, where $\mu_k(\boldsymbol{x})$ is the membership degree of $\boldsymbol{x}$ to A_k, whose value is in turn computed as a function of the quadratic shift $(g(\boldsymbol{x}) - \mathsf{m}_k(\boldsymbol{x}))^2$.

On the one hand we have one fuzzy partition of $D_{\boldsymbol{X}}$ for each component of $\boldsymbol{y}$. On the other hand, we implement the construct with many simplifications, in order to meet specific goals. Namely:

- since we look for a polynomial interpolation of g, we move from point membership functions to sets, to a monomial membership function to g, so that $g(\boldsymbol{x}) \simeq \sum_{k=1}^{c} \mu_k \mathsf{m}_k(\boldsymbol{x})$. In turn, μ_k is a *sui generis* membership degree, since it may assume also negative values;
- since for interpolation purposes we do not need $\mu_k(\boldsymbol{x})$, we identify the centroids directly with a hard clustering method based on the same quadratic shift.

Denoting $\mathsf{m}_k(\boldsymbol{x}) = \beta_k \prod_{j=1}^{n} x_j^{\alpha_{kj}}$, if we work in logarithmic scales, the shifts we consider for the single (say the i-th) component of $\boldsymbol{y}$ are the distances between $\boldsymbol{z}_r = (\log \boldsymbol{x}_r, \log y_r)$ and the hyperplane $\mathsf{h}_k(\boldsymbol{z}) = \boldsymbol{w}_k \cdot \boldsymbol{z} + b_k = 0$, with $\boldsymbol{w}_k = \{\alpha_{k1}, \ldots, \alpha_{kn}\}$ and $b_k = \log \beta_k$, constituting the centroid of A_k in an adaptive metric. Indeed, both $\boldsymbol{w}_k$ and b_k are learnt by the clustering algorithm aimed at minimizing the sum of the distances of the $\boldsymbol{z}_r$s from the hyperplanes associated to the clusters they are assigned to.

With the clustering procedure we essentially learn the exponents α_{kj} through which the $\boldsymbol{x}$ components intervene in the various monomials, whereas the β_ks remain ancillary parameters. Indeed, to get the polynomial approximation of $g(\boldsymbol{x})$ we compute the mentioned *sui generis* memberships through a simple quadratic fitting, i.e. by solving w.r.t. the vector $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_c\}$ the quadratic optimization problem: $\boldsymbol{\mu} = \arg\min_{\widetilde{\boldsymbol{\mu}}} \sum_{r=1}^{m} (g(\boldsymbol{x}_r) - \boldsymbol{y}_r))^2$, where x_{rj} denotes the j-th component of the r-th element of the training set $s_{\boldsymbol{x}}$, y_{rj} its approximation, with

$$y_j = \sum_{k=1}^{c} m_{jk}(\boldsymbol{x}) = \sum_{k=1}^{c} \mu_{jk} \prod_{i=1}^{n} x_i^{\alpha_{jki}} \tag{5}$$

where the index r has been hidden for notational simplicity, and μ_ks override β_ks.

3.2.1 A suited interpretation of the moment method

An early solution of the inverse problem:

> Which statistical features of $\boldsymbol{X}$ ensure a good coverage (in terms of *α-tolerance regions*) of the $\boldsymbol{Y}$ domain spanned by the performances measured on a sample of produced dies?

relies on the first and second moments of the target distribution, which are estimated on the basis of a sample $s_{\boldsymbol{y}}$ of sole $\boldsymbol{Y}$ collected from the production lines as representatives of properly functioning circuits. Our goal is to identify the statistical parameters $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$ of $\boldsymbol{X}$ that produce through (5) a $\boldsymbol{Y}$ population best approximating the above first and second order moments. $\boldsymbol{X}$ is assumed to be a multidimensional Gaussian variable, so that we identify it completely through the mean vector $\boldsymbol{\nu}_{\boldsymbol{X}}$ and the covariance matrix $\Sigma_{\boldsymbol{X}}$ which we do not constrain in principle to be diagonal (Eshbaugh, 1992). The analogous $\boldsymbol{\nu}_{\boldsymbol{Y}}$ and $\Sigma_{\boldsymbol{Y}}$ are a function of the former through (5). Although they could not identify the $\boldsymbol{Y}$ distribution in full, we are conventionally satisfied when these functions get numerically close to the estimates of the parameters they compute (directly obtained from the observed performance sample). Denoting with $\nu_{X_j}, \sigma_{X_j}, \sigma_{X_{j,k}}$ and $\rho_{X_{j,k}}$, respectively, the mean and standard deviation of X_j and the covariance/correlation between X_j and X_k, the master equations of our method are the following:

1.
$$\nu_{Y_i} = \sum_{k=1}^{c} \alpha_{ikj} \nu_{M_{ik}} \tag{6}$$
where M_{ik} on the right is a short notation of $m_{ik}(\boldsymbol{X})$, and $\nu_{M_{ik}}$ denotes its mean.

2. Thanks to the approximations
$$\nu_{\Xi} \simeq \log \nu_X, \quad \sigma_{\Xi} \simeq \sigma_X / \nu_X, \quad \rho_{\Xi_{i,j}} \simeq \rho_{X_{i,j}} \tag{7}$$
with $\Xi = \log X$, coming from the Taylor expansion of respectively Ξ, $(\Xi - \nu_{\Xi})^2$ and $(\Xi_i - \nu_{\Xi_i})(\Xi_j - \nu_{\Xi_j})$ around (ν_{X_i}, ν_{X_j}) disregarding others than the second terms, the rewriting of $\Sigma_{\boldsymbol{Y}}$ reads
$$\sigma^2_{Y_i} = \sum_{k=1}^{c} \sigma^2_{M_{ik}} + 2 \sum_{\substack{k,r=1 \\ k<r}}^{c} \sigma_{M_{ik,ir}} \tag{8}$$
$$\sigma_{Y_{i,j}} = \sum_{k,r=1}^{c} \sigma_{M_{ik,jr}} \tag{9}$$
with
$$\sigma^2_{M_{ik}} \simeq \nu^2_{M_{ik}} \left(\sum_{j=1}^{n} a^2_{ikj} \frac{\sigma^2_{X_j}}{\nu^2_{X_j}} + 2 \sum_{\substack{j,r=1 \\ j<r}}^{n} \rho_{X_{j,r}} a_{ikj} a_{ikr} \frac{\sigma_{X_j}}{\nu_{X_j}} \frac{\sigma_{X_r}}{\mu_{X_r}} \right) \tag{10}$$
$$\sigma_{M_{ik,ir}} \simeq \nu_{M_{ik}} \nu_{M_{ir}} \sum_{j,w=1}^{n} a_{ikj} a_{irw} \rho_{X_{j,w}} \frac{\sigma_{X_j}}{\nu_{X_j}} \frac{\sigma_{X_w}}{\nu_{X_w}} \tag{11}$$

We numerically solve (6) and (8-9) in $\nu_{\boldsymbol{X}}$ and $\Sigma_{\boldsymbol{X}}$ when the left members coincide with the target values of $\nu_{\boldsymbol{Y}}$ and $\Sigma_{\boldsymbol{Y}}$, respectively, and $\nu_{M_{ik}}$ is approximated with its sample estimate computed on samples artificially generated with the current values of the parameters. Solving equations means minimizing the differences between left and right members, so that the crucial point is the optimization method employed.The building blocks are the following.

The steepest descent strategy. Using the Taylor series expansion limited to second order (Mood et al., 1974), we obtain an approximate expression of the gradient components of $\boldsymbol{\nu_Y}$ w.r.t. $\boldsymbol{\nu_X}$ through
$$\frac{\partial \nu_{Y_i}}{\partial \nu_{X_j}} \simeq \sum_{k=1}^{c} \alpha_{ikj} \left(\frac{1}{\nu_{X_j}} + \frac{\sigma^2_{X_j}}{\nu^3_{X_j}} \right) \nu_{M_{ik}} \tag{12}$$

Thus we may easily look for the incremental descent on the quadratic error surface accounting for the difference between computed and observed means. Expression (12) confirms the scarce sensitivity of the unbiased mean $\boldsymbol{\nu_X}$, and its gradient as well, to the second moments, so that we may expect to obtain an early approximation of the mean vector to be subsequently refined. While analogous to the previous task, the identification of $\boldsymbol{X}$ variances and correlations owns one additional benefit and one additional drawback. The former derives from the fact that we may start with a, possibly well accurate, estimate of the means. The latter descends from the high interrelations among the target parameters which render the exploration of the quadratic error landscape troublesome and very lengthy.

Identification of second order moments. An alternative strategy for $\boldsymbol{X}$ second moment identification is represented by the evolutionary computation. Given the mentioned computational length of the gradient descent procedures, algorithms of this family become competitive on our target. Namely, we used Differential Evolution (Price et al., 2005), with specific bounds on the correlation values to avoid degenerate solutions.

A brute force numerical variant. We may move to a still more rudimentary strategy to get rid of the loose approximations introduced in (6) to (12). Thus we: i) avoid computing approximate analytical derivatives, by substituting them with direct numerical computations (Duch & Kordos, 2003), and ii) adopt the strategy of exploring one component at a time of the questioned parameter vector, rather than a combination of them all, until the error descent stops. Spanning numerically one direction at a time allows us to ask the software to directly identify the minimum along this direction. The further benefit of this task is that the function we want to minimize is analytic, so that the search for the minimum along one single direction is a very easy task for typical optimizers, such as the naive Nelder-Mead simplex method (Nelder & Mean, 1965) implemented in Mathematica (Wolfram Research Inc., 2008). We structured the method in a cyclic way, plus stopping criterion based on the amount of parameter variation. Each cycle is composed of: i) an iterative algorithm which circularly visits each component direction minimizing the error in the means' identification, until no improvement may be achieved over a given threshold, and ii) a fitting polynomial refresh on the basis of a Spice sample in the neighborhood of the current mean vector. We conclude the routine with a last assessment of the parameters that we pursue by running jointly on all them a local descent method such as Quasi-Newton procedure in one of its many variants (Nocedal & Wright, 1999).

3.2.2 Fine tuning via reverse mapping

Once a good fitting has been realized in the questioned part of the Spice mapping, we may solve the identification problem in a more direct way by first inverting the polynomial mapping to obtain the $\boldsymbol{X}$ sample at the root of the observed $\boldsymbol{Y}$ sample, and then estimating $\boldsymbol{\theta_X}$ directly from the sample defined in the $D_{\boldsymbol{X}}$ domain. The inversion is almost immediate if it is univocal, i.e., apart from controllable pathologies, when $\boldsymbol{X}$ and $\boldsymbol{Y}$ have the same number of components. Otherwise the problem is either overconstrained (number n of $\boldsymbol{X}$ components less than t, dimensionality of $\boldsymbol{Y}$ components) or underconstrained (opposite relation between component numbers). The first case is avoided by simply discarding exceeding $\boldsymbol{Y}$ components, possibly retaining the ones that improve the final accuracy and avoid numeric instability. The latter calls for a reduction in the number of questioned $\boldsymbol{X}$ components. Since $\boldsymbol{X}$ follows a multivariate Gaussian distribution law, by assumption, we may substitute some components with their conditional values, given the others.

4. Numerical experiments

The procedures we propose derive from a wise implementation of the Monte Carlo methods, as for the former, and a skillful implementation of granular computing ideas (Apolloni et al.,

device	model parameter		performance parameter	
	label	description	label	description
pMOS	U_0	Mobility at nominal temperature		
	A_0	Bulk charge effect coefficient	GM	conductance
	VTH_0	Threshold voltage at $V_{BS} = 0$ for large L	ID_{SAT}	source drain current
	K_1	First order body effect coefficient	VTH_{25-25}	saturation voltage
	B_{01}	Bulk charge effect coefficient for channel lenght	VTH_{25-08}	saturation voltage
	B_{11}	Bulk charge effect coefficient for channel width		
nMOS	U_0	Mobility at nominal temperature	GM	conductance
	V_{SAT}	Saturation voltage	ID_{SAT}	source drain current
	VTH_0	Threshold voltage at $V_{BS} = 0$ for large L	VTH_{25-25}	saturation voltage
	K_1	First order body effect coefficient	VTH_{25-08}	saturation voltage
NPN-DIB12	Bf	Ideal maximum foward Beta	HFE	Current Gain
	Re	Emitter Resistance	VA	Early Voltage
	Is	Transport Saturation Current	I_c	Collector Current
	Vaf	Forward Early Voltage		

Table 1. Model parameters and performances of the identification problems.

2008), as for the latter, however without theoretical proof of efficiency. While no worse from this perspective than the general literature in the field *per se* (McConaghy & Gielen, 2005), it needs numerical proof of suitability. To this aim we basically work with three real world benchmarks collected by manufacturers to stress the peculiarities of the methods. Namely, the benchmarks refer to:

1. A unipolar pMOS device realized in Hcmos4TZ technology.
2. A unipolar nMOS device differentiating from the former for the sign (negative here, positive there) of the charge of the majority mobile charge carriers. Spice model and technology are the same, and performance parameters as well. However, the domain spanned by the model parameters is quite different, as will be discussed shortly.
3. A bipolar NPN circuit realized in DIB12 technology. DIB technology achieves the full dielectric isolation of devices using SOI substrates by the integration of the dielectric trench that comes into contact with the buried oxide layer.

The related model parameter took into consideration and measured performances are reported in Table 1.

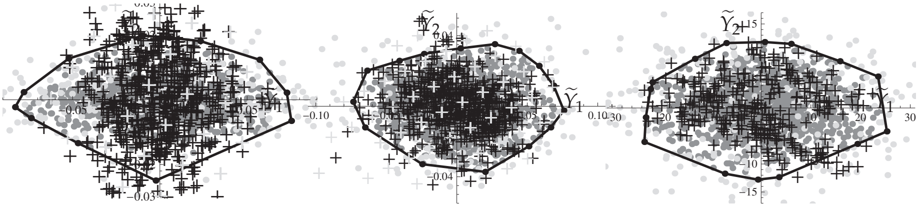
We have different kinds of samples for the various benchmarks as for both the sample size which ranges from $14,000$ (pMOS and nMOS) to 300 (NPN-DIB12) and the measures they report: joint measures of 4 performance parameters in the former two cases, partially independent measures of 3 performance parameters in the latter, where only HFE and VA are jointly measured. Taking into account the model parameters, and recalling the meaning of t and n in terms of number of performance and model parameters, respectively, the sensitivity of the former parameters to the latter and the different difficulties of the identification tasks lead us to face in principle one balanced problem with $n = t = 4$ (nMOS), and two unbalanced ones with $n = 6$ and $t = 4$ (pMOS) and $n = 4$ and $t = 3$ (NPN-DIB12). In addition, only 4 of the 6 second order moments are observed with the third benchmark.

4.1 Reverting the Spice model on the three benchmarks

With reference to Table 2, in column $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$ we report the parameters of the input multivariate Gaussian distribution we identify in the aim of reproducing the $\boldsymbol{\theta}_{\boldsymbol{Y}}$ of the $\boldsymbol{Y}$ population observed through $s_{\boldsymbol{y}}$. Of the latter parameter, in the subsequent column $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{Y}}/\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}}$ we compare

benchmark			solution					
dataset	(n,t)	m	$\tilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$			$\tilde{\boldsymbol{\theta}}_{\boldsymbol{Y}} / \hat{\ }_{Y}$		
benchmark			$\boldsymbol{\mu}_{\boldsymbol{X}}$	$\boldsymbol{\sigma}_{\boldsymbol{X}}$	$\boldsymbol{\rho}_{\boldsymbol{X}}$	$\boldsymbol{\mu}_{\boldsymbol{Y}}$	$\boldsymbol{\sigma}_{\boldsymbol{Y}}$	$\boldsymbol{\rho}_{\boldsymbol{Y}}$
pMOS	$(6,4)$	$14,000$	$\begin{pmatrix} 233.424 \\ 0.28798 \\ 0.99185 \\ 0.45255 \\ 4.06626 \times 10^{-5} \\ 4.67824 \times 10^{-5} \end{pmatrix}$	$\begin{pmatrix} 3.63673 \\ 0.01806 \\ 0.01083 \\ 0.03275 \\ 4.48106 \times 10^{-6} \\ 9.90006 \times 10^{-6} \end{pmatrix}$	$\begin{pmatrix} -0.16582 \\ -0.46312 \\ -0.41451 \\ -0.49665 \\ -0.35008 \\ -0.12573 \\ -0.47067 \\ -0.07056 \\ -0.39330 \\ 0.09484 \\ -0.16367 \\ 0.21068 \\ 0.49711 \\ 0.22781 \\ 0.48312 \end{pmatrix}$	$\begin{pmatrix} \mathit{-0.835824} \\ \mathbf{-0.838496} \\ \mathit{-0.971835} \\ \mathbf{-0.969196} \\ \mathit{0.000973318} \\ \mathbf{0.00097472} \\ \mathit{0.00448103} \\ \mathbf{0.00447346} \end{pmatrix}$	$\begin{pmatrix} \mathit{0.0118109} \\ \mathbf{0.0187507} \\ \mathit{0.0121665} \\ \mathbf{0.0164674} \\ \mathit{0.000029378} \\ \mathbf{0.000029348} \\ \mathit{0.000146626} \\ \mathbf{0.000130486} \end{pmatrix}$	$\begin{pmatrix} \mathit{0.933746} \\ \mathbf{0.451486} \\ \mathit{-0.28765} \\ \mathbf{-0.28251} \\ \mathit{-0.38997} \\ \mathbf{-0.38744} \\ \mathit{-0.25444} \\ \mathbf{-0.07276} \\ \mathit{-0.36747} \\ \mathbf{-0.17454} \\ \mathit{0.900391} \\ \mathbf{0.983658} \end{pmatrix}$
nMOS	$(4,4)$	$14,000$	$\begin{pmatrix} 752.395 \\ 152858.0 \\ 0.68184 \\ 0.521661 \end{pmatrix}$	$\begin{pmatrix} 134.099 \\ 9667.22 \\ 0.0186854 \\ 0.131933 \end{pmatrix}$	$\begin{pmatrix} -0.765278 \\ -0.467972 \\ 0.756786 \\ 0.306389 \\ -0.786377 \\ -0.468842 \end{pmatrix}$	$\begin{pmatrix} \mathit{0.552391} \\ \mathbf{0.550715} \\ \mathit{0.66383} \\ \mathbf{0.664162} \\ \mathit{0.00221691} \\ \mathbf{0.00222077} \\ \mathit{0.0100527} \\ \mathbf{0.0100711} \end{pmatrix}$	$\begin{pmatrix} \mathit{0.028568} \\ \mathbf{0.0276768} \\ \mathit{0.0176982} \\ \mathbf{0.0173677} \\ \mathit{0.0000830626} \\ \mathbf{0.0000619134} \\ \mathit{0.000355129} \\ \mathbf{0.000280373} \end{pmatrix}$	$\begin{pmatrix} \mathit{0.445093} \\ \mathbf{0.395429} \\ \mathit{-0.49927} \\ \mathbf{-0.43243} \\ \mathit{-0.63796} \\ \mathbf{-0.64032} \\ \mathit{-0.29840} \\ \mathbf{-0.27195} \\ \mathit{-0.37584} \\ \mathbf{-0.35488} \\ \mathit{0.92015} \\ \mathbf{0.950419} \end{pmatrix}$
NPN-DIB12	$(4,3)$	322	$\begin{pmatrix} 138.302 \\ 0.67258 \\ 5.28102 \times 10^{-18} \\ 136.319 \end{pmatrix}$	$\begin{pmatrix} 8.3859 \\ 0.263238 \\ 4.14306 \times 10^{-19} \\ 13.6538 \end{pmatrix}$	$\begin{pmatrix} -0.192107 \\ 0.00139749 \\ -0.477207 \\ -0.980327 \\ 0.167527 \\ -0.0444712 \end{pmatrix}$	$\begin{pmatrix} \mathit{113.244} \\ \mathbf{113.242} \\ \mathit{0.0000654246} \\ \mathbf{0.0000653275} \\ \mathit{110.164} \\ \mathbf{110.238} \end{pmatrix}$	$\begin{pmatrix} \mathit{6.82099} \\ \mathbf{6.95918} \\ \mathit{4.96031 \times 10^{-6}} \\ \mathbf{4.81021 \times 10^{-6}} \\ \mathit{11.1459} \\ \mathbf{11.2166} \end{pmatrix}$	$\begin{pmatrix} \mathit{-0.49079} \\ \mathbf{-0.56667} \end{pmatrix}$

Table 2. Benchmarks used for testing the proposed procedure and analysis of the identification solution. Columns: inferred model distribution parameters (indexed by $\boldsymbol{X}$) and reconstructed performance paran comparative levels of the tolerance regions (as a function of δ).



(a) pMOS (b) nMOS (c) npn-DIB

Fig. 2. Comparison between output data and reconstruction provided by Reverse Spice based procedure for the devices listed in Table 2 when projected on the two principal components of the target. Points: reconstructed population lying within (dark gray) and outside (light gray) 0.90 tolerance region (black curves) identified by black points. Gray crosses: original target output; black crosses: target output uniformly spread with noise terms.

the values computed on the basis of $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$ (referring to a reconstructed distribution – in italics) with those computed through the maximum likelihood estimate from $s_{\boldsymbol{y}}$ (referring to the original distribution – in bold). As a further accuracy indicator, we will consider tolerance regions obtained through convex hull peeling depth (Barnett, 1976) containing a given percentage $1 - \delta$ of the performance population. In the last column of Table 2, headed by $(1 - \widetilde{\delta})/(\mathbf{1} - \boldsymbol{\delta})$, we appreciate the difference between planned tolerance rate (in bold), as a function of the identified $\boldsymbol{Y}$ distribution, and ratio of sampled measures found in these regions (in italics). We consider single values in the table cells since the results are substantially insensitive to the random components affecting the procedure, such as algorithm initialization. Rather, especially with *difficult* benchmarks, they may depend on the user options during the run of the algorithm. Thus, what we report are the best results we obtain, reckoning the overall trial time in the computational complexity consideration we will do later on in this section.

For a graphical counterpart, in Fig. 2 we report the scatterplot of the original $\boldsymbol{Y}$ sample and an analogous one generated through the reconstructed distribution, both projected on the plane identified by the two principal components (Jolliffe, 1986) of the original distribution. We also draw the intercept of this plane with a tolerance region containing 90% of the reconstructed points (hence $\delta = 0.1$).

An overview of these data looks very satisfactory, registering a relative shift between sample and identified parameters that is always less than 0.17% as for the mean values, 45% for the standard deviations and 25% for the correlation. The analogous shift between planned and actual percentages of points inside the tolerance region is always less than 2%. We distinguish between *difficult* and *easy* benchmarks, where the pMOS sample falls in the first category. Indeed the same percentages referring to the remaining benchmarks decreases to 0.13%, 10% and 9%.

Given the high computational costs of the Spice models, their approximation through cheaper functions is the first step in many numerical procedures on microelectronic circuits. Within the vast set of methods proposed by researchers on the matter (Ampazis & Perantonis, 2002a;b; Daems et al., 2003; Friedman, 1991; Hatami et al., 2004; Hershenson et al., 2001; McConaghy et al., 2009; Taher et al., 2005; Vancorenland et al., 2001) in Table 3 we report a numerical comparison between two well reputed fitting methods and our proposed Reverse Spice based algorithm (for short RS). The methods are Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991), i.e. piecewise polynomials, and Polynomial Neural Networks

	$\widetilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$		$\widetilde{\boldsymbol{\theta}}'_{\boldsymbol{X}}$	
	train	test	train	test
RS	*0.0000125623* $\begin{pmatrix} 0.0000350975 \\ 0.0000151476 \\ 3.06034 \times 10^{-10} \\ 3.59774 \times 10^{-9} \end{pmatrix}$	*0.0000242739* $\begin{pmatrix} 0.0000759397 \\ 0.0000211444 \\ 6.62265 \times 10^{-10} \\ 1.10138 \times 10^{-8} \end{pmatrix}$	*0.000228931* $\begin{pmatrix} 0.000751481 \\ 0.000164105 \\ 1.54286 \times 10^{-8} \\ 1.24052 \times 10^{-7} \end{pmatrix}$	*0.000369871* $\begin{pmatrix} 0.00131925 \\ 0.000159924 \\ 2.33858 \times 10^{-8} \\ 2.92353 \times 10^{-7} \end{pmatrix}$
MARS	$8.68173 * 10^{-}6$ $\begin{pmatrix} 0.0000246876 \\ 0.0000100344 \\ 2.80773 \times 10^{-10} \\ 4.66935 \times 10^{-9} \end{pmatrix}$	*0.0000168024* $\begin{pmatrix} 0.0000528055 \\ 0.0000143915 \\ 5.92204 \times 10^{-10} \\ 1.19291 \times 10^{-8} \end{pmatrix}$	*0.000124012* $\begin{pmatrix} 0.000401349 \\ 0.0000946271 \\ 5.3722 \times 10^{-9} \\ 6.47147 \times 10^{-8} \end{pmatrix}$	*0.0002805* $\begin{pmatrix} 0.00100927 \\ 0.000112503 \\ 6.07291 \times 10^{-9} \\ 2.22601 \times 10^{-7} \end{pmatrix}$
PNN	*0.0000602061* $\begin{pmatrix} 0.000230822 \\ 0.0000100003 \\ 2.7761 \times 10^{-10} \\ 2.38434 \times 10^{-9} \end{pmatrix}$	*0.0000769737* $\begin{pmatrix} 0.000293665 \\ 0.0000142199 \\ 5.70282 \times 10^{-10} \\ 9.12621 \times 10^{-9} \end{pmatrix}$	*0.000125976* $\begin{pmatrix} 0.000409046 \\ 0.0000948249 \\ 4.14671 \times 10^{-9} \\ 2.84136 \times 10^{-8} \end{pmatrix}$	*0.000280898* $\begin{pmatrix} 0.00101197 \\ 0.000111354 \\ 7.14833 \times 10^{-9} \\ 2.62591 \times 10^{-7} \end{pmatrix}$

Table 3. Performance comparison between fitting algorithms. Rows: algorithms; main columns: benchmark parameterization; subcolumns: experimental environments (training set, test set).

(PNN) (Elder IV & Brown, 2000). Namely, we consider the $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$ reported in Table 2 as the result of the nMOS circuit identification. On the basis of these parameters and through Spice functions, we draw a sample of 250 pairs $(\boldsymbol{x}_r, \boldsymbol{y}_r)$ that we used to feed both competitor algorithms and our own. In detail we used VariReg software (Jekabsons, 2010a;b) to implement both MARS and PNN. To ensure a fair comparison among the differente methods, we: i) set equal to 6 the number of monomials in our algorithm and the maximum number of basis functions in MARS, where we used a cubic interpolation, and ii) employ the default configuration in PNN by setting the degree of single neurons polynomial equal to 2. Moreover, in order to understand how the various algorithms scale with the fitting domain, we repeat the procedure with a second set $\widetilde{\boldsymbol{\theta}}'_{\boldsymbol{X}}$ of parameters, where the original standard deviations have been uniformly doubled. In the table we report the mean squared errors measured on a test set of size 1000, whose values are both split on the four components of the performance vector and resumed by their average. The comparison denotes similar accuracies with the most concentrated sample – the actual operational domain of our polynomials – and a small deterioration of our accuracy in the most dispersed sample, as a necessary price we have to pay for the simplicity of our fitting function.

As for the whole procedure, we reckon overall running times of around half an hour. Though not easily contrastable with computational costs of analogous tasks, this order of magnitude results adequate for an intensive use of the procedure in a circuit design framework.

4.2 Stochastically optimizing the third benchmark model

The same NPN-DIB12 benchmark discussed in Section 4.1 was also used to run the two-step MC procedure depicted in Section 3.1. In particular, estimation of the sole standard deviations σ_{X_i}s in the former phase alternates with cross-correlation coefficients' in the latter, while the means remain fixed to their nominal values $\nu_{X_i} = \hat{\nu}_{X_i}$ Namely, at each iteration a sample $s_{\mathscr{M}} = \{\boldsymbol{x}_r\}, r = 1\ldots, m = 5000$ was generated, and the whole procedure was repeated 7 times, until over 99% of sample instances were included in the tolerance region. Fig. 3 shows the number $\widetilde{m}$ of selected instances for each iteration of the algorithm.

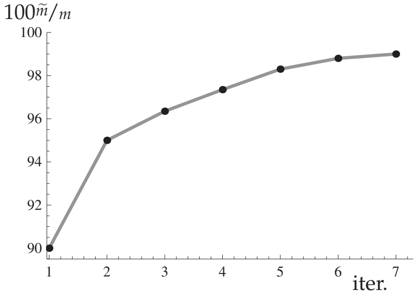


Fig. 3. Percentage of selected instances at each iteration of the two-step MC algorithm.

4.3 Comparing the proposed methods

In order to grasp insights on the comparative performances of the proposed methods, we list their main features on the common NPN-DIB12 benchmark. Namely, in the first row of Table 4 we report the reference value of the means and standard deviations of both $\boldsymbol{X}$ and $\boldsymbol{Y}$ distributions. As for the first variable, we rely on the nominal values of the parameters for the means, leaving empty the cell concerning the standard deviations. As for the performances, we just use the moment MLE estimate computed on the sample $s_{\boldsymbol{y}}$. In the remaining rows we report the analogous values computed from a huge sample of the above variables artificially generated through the statistical models we identify.

	$\widetilde{\boldsymbol{\theta}}_{\boldsymbol{X}}$		$\widetilde{\boldsymbol{\theta}}_{\boldsymbol{Y}}$	
	$\boldsymbol{\mu_X}$	$\boldsymbol{\sigma_X}$	$\boldsymbol{\mu_Y}$	$\boldsymbol{\sigma_Y}$
Reference	$\begin{pmatrix} 135 \\ 0.8 \\ 5.12 \times 10^{-18} \\ 138 \end{pmatrix}$		$\begin{pmatrix} 113.242 \\ 6.5328 \times 10^{-5} \\ 110.238 \end{pmatrix}$	$\begin{pmatrix} 6.9592 \\ 4.8102 \times 10^{-6} \\ 11.2166 \end{pmatrix}$
MC	$\begin{pmatrix} 135 \\ 0.8 \\ 5.12 \times 10^{-18} \\ 138 \end{pmatrix}$	$\begin{pmatrix} 8.2375 \\ 7.9064 \times 10^{-2} \\ 3.9744 \times 10^{-19} \\ 9.4 \end{pmatrix}$	$\begin{pmatrix} 110.5854 \\ 6.346 \times 10^{-5} \\ 110.039 \end{pmatrix}$	$\begin{pmatrix} 6.6418 \\ 4.691 \times 10^{-6} \\ 7.507 \end{pmatrix}$
RS	$\begin{pmatrix} 138.302 \\ 0.6726 \\ 5.281 \times 10^{-18} \\ 136.319 \end{pmatrix}$	$\begin{pmatrix} 8.3859 \\ 0.2632 \\ 4.1431 \times 10^{-19} \\ 13.6538 \end{pmatrix}$	$\begin{pmatrix} 113.244 \\ 6.5425 \times 10^{-5} \\ 110.164 \end{pmatrix}$	$\begin{pmatrix} 6.821 \\ 4.9603 \times 10^{-6} \\ 11.1459 \end{pmatrix}$

Table 4. Comparison between both model and performance moments re reference and reconstructed frameworks.

Both tables denote a slight comparative benefit of using the reverse modeling (row RS), in terms of both a greater variance of the model parameters and a better similarity of the reconstructed performance parameters with the estimated ones w.r.t. the analogous parameters obtained with Monte Carlo method (row MC). The former feature reflects into less severe constraints in the production process. The latter denotes some improvement in the reconstruction of the performances' distribution law, possibly deriving from both freeing the $\boldsymbol{\nu_X}$ from their nominal values and a massive use of the Spice function analytical forms.

5. Conclusions

A major challenge posed by new deep-submicron technologies is to design and verify integrated circuits to obtain a high fabrication yield, i.e. a high proportion of produced circuits that function properly. The classical approach implemented in commercial tools for parameter extraction (IC-Cap by Agilent Technology (2010), and UTMOST by Silvaco Engineered (2010)) requires a dedicated electrical characterization for a large number of devices, in turn demanding for a very long time in terms both of experimental characterization and parameter extraction.

Thus, a relevant goal with these procedures is to reduce the computational time to have a statistical description of the device model. We fill it by using two non conventional methods so as to get a speed-up factor greater than 10 w.r.t. standard procedures in literature. The first method we propose is based on a Monte Carlo technique to estimate the (second order) moments for several statistical model parameters, on the basis of characterizated data, collected during the manufacturing process.

The second method exploits a granular construct. In spite of the methodology broadness the attribute *granular* may evoke, we obtain a very accurate solution taking advantage from strict exploitation of state-of-the-art theoretical results. Starting from the basic idea of considering the Spice function as a mixture of fuzzy sets, we enriched its implementation with a series of sophisticated methodologies for: i) identifying clusters based on proper metrics on functional spaces, ii) descending, direction by direction, along the *ravines* of the cost functions of the related optimization problems, iii) inverting the $(\boldsymbol{X}, \boldsymbol{Y})$ mapping in case of unbalanced problems through the bootstrapping of conditional Gaussian distributions, and iv) computing tolerance regions through convex hull based peeling techniques. In this way we supply a very accurate and fast algorithm to identify statistically the circuit model.

Of course, both procedures are susceptible of further improvements deriving from a more and more deep statistics' exploitation. In addition, nobody may guarantee that they will *resist* to a further reduction of the technology scales. However the underlying methods we propose could remain at the root of new solution algorithms of the yield maximization problem.

6. References

Agilent Technology (2010). *IC-CAP Device Modeling Software – Measurement Control and Parameter Extraction*, Santa Clara, CA.
URL: *http://www.home.agilent.com/agilent/home.jspx*

Allgower, E. L. & Georg, K. (1990). *Computational solution of nonlinear systems of equations*, American Mathematical Society, Providence, RI.

Ampazis, N. & Perantonis, S. J. (2002a). OLMAM Neural Network toolbox for Matlab.
URL: `http://iit.demokritos.gr/ abazis/toolbox/`

Ampazis, N. & Perantonis, S. J. (2002b). Two highly efficient second order algorithms for training feedforward networks, *IEEE Transactions on Neural Networks* 13(5): 1064–1074.

Apolloni, B., Bassis, S., Malchiodi, D. & Witold, P. (2008). *The Puzzle of Granular Computing*, Vol. 138 of *Studies in Computational Intelligence*, Springer Verlag.

Barnett, V. (1976). The ordering of multivariate data, *Journal of Royal Statistical Society Series A* 139: 319–354.

Bernstein, K., Frank, D. J., Gattiker, A. E., Haensch, W., Ji, B. L., Nassif, S. R., Nowak, E. J., Pearson, D. J. & Rohrer, N. J. (2006). High-performance CMOS variability in the 65-nm regime and beyond, *IBM Journal of Research Development* 50(4/5): 433–449.

Boning, D. S. & Nassif, S. (1999). Models of process variations in device and interconnect, *in* A. Chandrakasan (ed.), *Design of High Performance Microprocessor Circuits, chapter 6*, IEEE Press.

Bühler, M., Koehl, J., Bickford, J., Hibbeler, J., Schlichtmann, U., Sommer, R., Pronath, M. & Ripp, A. (2006). DFM/DFY design for manufacturability and yield - influence of process variations in digital, analog and mixed-signal circuit design, *DATE'06*, pp. 387–392.

Chang, E., Stine, B., Maung, T., Divecha, R., Boning, D., Chung, J., Chang, K., Ray, G., Bradbury, D., Nakagawa, O. S., Oh, S. & Bartelink, D. (1995). Using a statistical metrology framework to identify systematic and random sources of die- and wafer-level ILD thickness variation in CMP processes, *in CMP processes, IEDM Technology Digest*, pp. 499–502.

Daems, S., Gielen, G. & Sansen, W. (2003). Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22(5): 517–534.

Duch, W. & Kordos, M. (2003). Multilayer perceptron trained with numerical gradient, *Proceedings of the International Conference on Artificial Neural Networks (ICANN) and International Conference on Neural Information Processing (ICONIP)*, Istanbul, pp. 106–109.

Eeckelaert, T., Daems, W., Gielen, G. & Sansen, W. (2004). Generalized simulation-based posynomial model generation for analog integrated circuits, *Analog Integrated Circuits Signal Processing* 40(3): 193–203.

Efron, B. & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*, Chapman & Hall, New York.

Elder IV, J. F. & Brown, D. E. (2000). Induction and polynomial networks. network models for control and processing, *in* M. Fraser (ed.), *Intellect*, Portland, OR, pp. 143–198.

Eshbaugh, K. S. (1992). Generation of correlated parameters for statistical circuit simulation, *IEEE Transactions on CAD of Integrated Circuits and Systems* 11(10): 1198–1206.

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines, *Annals of Statistics* 19: 1–141.

Hatami, S., Azizi, M. Y., Bahrami, H. R., Motavalizadeh, D. & Afzali-Kusha, A. (2004). Accurate and efficient modeling of SOI MOSFET with technology independent neural networks, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23(11): 1580–1587.

Hershenson, M., Boyd, S. & Lee, T. (2001). Optimal design of a CMOS OP-AMP via geometric programming, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 20(1): 1–21.

Jekabsons, G. (2010a). Adaptive basis function construction: an approach for adaptive building of sparse polynomial regression models, *Machine Learning, In-Tech* p. 28. In Press.

Jekabsons, G. (2010b). VariReg software.
URL: *http://www.cs.rtu.lv/jekabsons/*

Johnson, G. E. (1994). Constructions of particular random processes, *Proceedings of the IEEE* 82(2): 270–285.

Jolliffe, I. T. (1986). *Principal Component Analysis*, Springer Verlag.

Jones, D. R., Perttunen, C. D. & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant, *Journal of Optimization Theory and Applications* 79(1): 157–181.

Koskinen, T. & Cheung, P. (1993). Statistical and behavioural modelling of analogue integrated circuits, *Circuits, Devices and Systems, IEE Proceedings G* 140(3): 171–176.

Kundert, K. S. (1998). *The DesignerâĂŹs Guide to SPICE and SPECTRE*, Kluwer Academic Publishers, Boston.

Lehmann, E. (2006). *Nonparametrics, Statistical Methods Based on Ranks*, Vol. XVI, Prentice-Hall. 1st edition in 1975, revised edition in 2006.

Liu, R. Y., Parelius, J. M. & Singh, K. (1999). Multivariate analysis by data depth: Descriptive statistics, graphics and inference, *The Annals of Statistics* 27: 783–858.

McConaghy, T. & Gielen, G. (2005). Analysis of simulation-driven numerical performance modeling techniques for application to analog circuit optimization, *Proceedings of IEEE International Symposium on Circuits and Systems*.

McConaghy, T., Palmers, P., Gao, P., Steyaert, M. & Gielen, G. G. E. (2009). *Variation-Aware Analog Structural Synthesis: A Computational Intelligence Approach*, Springer.

Mood, A. M., Graybill, F. A. & Boes, D. C. (1974). *Introduction to the Theory of Statistics*, McGraw-Hill, New York.

Nelder, J. A. & Mean, R. (1965). A simplex method for function minimization, *Computer Journal* 7: 308–313.

Nocedal, J. & Wright, S. J. (1999). *Numerical Optimization*, Series: Springer series in operations research, Springer, New York.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1993). *Numerical Recipes in Fortran; the Art of Scientific Computing*, Cambridge University Press, New York, NY, USA.

Price, K. V., Storn, R. M. & Lampinen, J. A. (2005). *Differential Evolution, A Practical Approach to Global Optimization*, Vol. 538 of *Natural Computing Series*, Springer.

Rohatgi, V. K. (1976). *An Introduction to Probablity Theory and Mathematical Statistics*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York.

Rubinstein, R. Y. & Kroese, D. P. (2007). *Simulation and the Monte Carlo Methods*, Probability and Statistics, 2nd edn, John Wiley and Sons Inc.

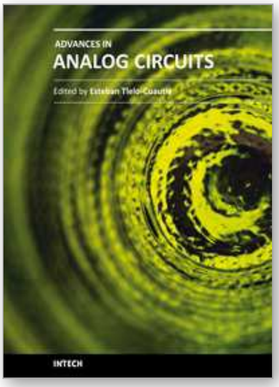
Silvaco Engineered (2010). *UTMOST III – SPICE Modeling Software*, Santa Clara, CA.

Stolk, P. A., Widdershoven, F. P. & Klaassen, D. B. M. (1988). Modeling statistical dopant fluctuations in MOS transistors, *IEEE Transactions on Electron Devices* 45(9): 1960 – 1971.

Taher, H., Schreurs, D. & Nauwelaers, B. (2005). Extraction of small signal equivalent circuit model parameters for statistical modeling of HBT using artificial neural networks, *Gallium Arsenide Applications Symposium (GAAS 2005) 3-7 ottobre 2005*.

Vancorenland, P., Van der Plas, G., Steyaert, M., Gielen, G. & Sansen, W. (2001). A layout-aware synthesis methodology for RF circuits, *ICCAD'01: Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design*, IEEE Press, Piscataway, NJ, USA, pp. 358–362.

Wolfram Research Inc. (2008). Mathematica 7.
URL: *http://www.wolfram.com/products/ mathematica/index.html*



ISBN 978-953-307-323-1
Hard cover, 368 pages
Publisher InTech
Published online 02, February, 2011
Published in print edition February, 2011

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821