Bayesian Model Fusion: A Statistical Framework for Efficient Pre-Silicon Validation and Post-Silicon Tuning of Complex Analog and Mixed-Signal Circuits

(Invited Special Session Paper)

Xin Li[1], Fa Wang[1], Shupeng Sun[1] and Chenjie Gu[2]
[1]ECE Department, Carnegie Mellon University, Pittsburgh, PA 15213, {xinli, fwang1, shupengs}@ece.cmu.edu
[2]Strategic CAD Labs, Intel Corporation, Hillsboro, OR 97124, chenjie.gu@intel.com

ABSTRACT

In this paper, we describe a novel statistical framework, referred to as *Bayesian Model Fusion* (BMF), that allows us to minimize the simulation and/or measurement cost for both pre-silicon validation and post-silicon tuning of analog and mixed-signal (AMS) circuits with consideration of large-scale process variations. The BMF technique is motivated by the fact that today's AMS design cycle typically spans multiple stages (e.g., schematic design, layout design, first tape-out, second tape-out, etc.). Hence, we can *reuse* the simulation and/or measurement data collected at an early stage to facilitate efficient validation and tuning of AMS circuits with *a minimal amount of data* at the late stage. The efficacy of BMF is demonstrated by using several industrial circuit examples.

1. INTRODUCTION

The aggressive technology scaling has made it more difficult than ever to design and manufacture high-performance analog and mixed-signal (AMS) circuits (e.g., RF front-end, high-speed I/O link, etc.) that are robust to large-scale process variations [1]-[3]. Most AMS performance metrics (e.g., random offset) are extremely sensitive to the inter-die and/or intra-die variations associated with today's nanoscale manufacturing technology. For this reason, AMS circuits are not as scalable as digital circuits and they have been considered as the major bottleneck for future scaling of integrated circuit (IC) technology [4].

To address these challenges associated with AMS circuits, adaptive post-silicon tuning has been recently proposed [5]-[7] to facilitate the continuous scaling of AMS circuits. The adoption of post-silicon tuning, however, introduces tremendous new design challenges. Each AMS circuit now becomes a large-scale, complex system that can adaptively vary over time. It, in turn, brings up enormous technical problems related to computer-aided design of tunable AMS circuits.

- **Challenges for pre-silicon validation**: A tunable AMS circuit must contain a number of control knobs (e.g., switches, tunable bias voltage/current, etc.) for post-silicon reconfiguration. In some cases, these circuits may even include on-chip sensors and control blocks for "self-healing". Hence, the complexity of these AMS circuits substantially increases and the corresponding simulation cost for pre-silicon validation becomes increasingly large. This cost issue is especially critical for highly complex AMS circuits, such as phase-locked loop and high-speed I/O link, where a single transistor-level simulation may take a few days or even a few weeks to finish. In this case, it is extremely expensive to repeatedly run a large number of simulations over all process variations and environmental corners to validate a given design.
- **Challenges for post-silicon tuning**: Post-silicon tuning requires us to first measure the circuit performance by either on-chip sensors and/or off-chip equipments and then determine the optimal setup for all control knobs based on the measurement results. Towards this goal, a *tuning policy* must be developed to map the measured performance to the knob setup. In order to learn such a tuning policy over all possible process and environmental conditions, a lot of measurement data must be collected [8]-[11]. Furthermore, the tuning policy must be periodically re-calibrated to accommodate the process shift over time, thereby requiring additional measurement data. Collecting all these silicon data can be expensive, or even infeasible, as silicon measurement has become extremely time-consuming today.

Most CAD methodologies today need to collect a large amount of simulation and measurement data for pre-silicon validation and post-silicon tuning, resulting in prohibitively high cost. The challenging issue here is how to make AMS validation and tuning affordable by *reducing the amount of required data*. This fundamental issue has not been appropriately addressed by the state-of-the-art CAD tools. It, in turn, poses an immediate need to develop a radically new CAD framework for efficient validation and tuning of today's AMS circuits in order to facilitate the continuous scaling of ICs.

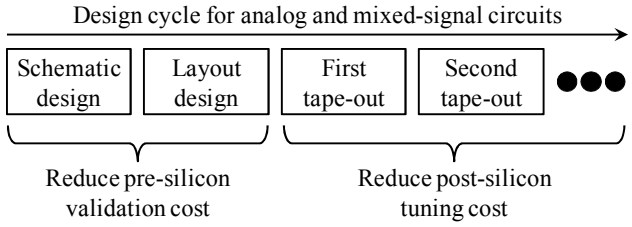

Figure 1. Bayesian Model Fusion (BMF) reuses the early-stage data to facilitate efficient validation and tuning of AMS circuits at a late stage.

In this paper, we describe a novel idea of *Bayesian Model Fusion* (BMF) to address the aforementioned challenges [12]-[15]. BMF is motivated by the fact that today's AMS design cycle typically spans multiple stages, as shown in Figure 1. At each stage, simulation and/or measurement data are collected to validate and/or tune the AMS circuit, before moving to the next stage. Most conventional AMS validation and tuning methods rely on the data collected at a single stage only and they completely ignore the data that are *already* generated at the previous stages. The key idea of BMF, however, is to *reuse* the early-stage data to facilitate efficient validation and tuning of AMS circuits at a late stage. As such, the amount of required data can be substantially reduced at the late stage.

Mathematically, BMF is derived from the theory of *Bayesian*

978-1-4799-1071-7/13/$31.00 ©2013 IEEE

inference [16]-[17]. Starting from a set of early-stage data, it first "learns" the prior knowledge from these existing data and statistically "encodes" them as a *prior distribution*. Next, the prior distribution is combined with very few late-stage data to efficiently solve the pre-silicon validation and/or post-silicon tuning problem via Bayesian inference. By *fusing* the early-stage and late-stage data together through Bayesian inference, the simulation and/or measurement cost can be significantly reduced.

The BMF framework is generally applicable to a broad range of practical problems related to pre-silicon validation and post-silicon tuning of AMS circuits. The following are a few representative examples.

- **Example problem for pre-silicon validation**: How do we estimate the pre-silicon performance distribution and parametric yield of an AMS circuit based on *10* post-layout Monte Carlo simulation samples only?
- **Example problem for post-silicon tuning**: How do we learn the post-silicon tuning policy for a tunable AMS circuit based on the measurement data collected from *2* silicon chips only?

These problems seem extremely challenging, if not impossible, to solve. However, we will show in this paper that BMF can successfully address these fundamental-yet-challenging problems and makes them tractable, as will be demonstrated by the simulation and measurement data of several industrial circuit examples.

The remainder of this paper is organized as follows. In Section 2, we first describe the BMF formulation for two different applications of pre-silicon validation: (i) moment estimation, and (ii) distribution estimation. Next, we further extend BMF to post-silicon tuning in Section 3. Finally, we conclude in Section 4.

2. PRE-SILICON VALIDATION

In this section, we describe the theoretical framework of BMF for pre-silicon validation. While pre-silicon validation is a broad area involving many different research topics, our focus is to accurately estimate the statistics of a given AMS circuit performance (e.g., gain of an amplifier, phase noise of an oscillator, etc.). These statistical metrics include moments (e.g., mean and variance) and distributions (e.g., probability density function and cumulative distribution function) that are particularly useful for estimating the parametric yield of the AMS circuit.

2.1 Moment Estimation

Moment estimation is an important task to predict the performance distribution and, consequently, the parametric yield. In particular, if the performance distribution is Gaussian, it is fully characterized by the first two moments (i.e., mean and variance) [18]. Even if the performance distribution is non-Gaussian, it can be accurately estimated by finding the high-order moments [19].

Without loss of generality, we consider a given AMS performance of interest x. Most conventional statistical algorithms require a lot of data to estimate the moments of x [18]-[20]. However, in practice, pre-silicon simulation (e.g., post-layout simulation) is time-consuming. Especially for large-scale, complex AMS circuits such as phase-locked loop and high-speed I/O link, only a limited amount of (e.g., less than 5) simulation runs can be afforded. The key idea of BMF is to reuse the early-stage (e.g., schematic-level) data to accurately estimate the late-stage (e.g., post-layout) moments with very few late-stage data [13].

In this sub-section, we consider mean estimation as an example to illustrate the mathematical formulation of BMF. It should be noted that the BMF methodology can be generally extended to estimate other high-order moments as well [13]. Assume that the circuit performance x follows a Gaussian distribution; however, the early-stage performance distribution $pdf_E(x)$ and the late-stage performance distribution $pdf_L(x)$ have different mean and variance values:

$$pdf_E(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma_E}\cdot\exp\left[-\frac{(x-\mu_E)^2}{2\cdot\sigma_E^2}\right] \sim Gauss\left(\mu_E,\sigma_E^2\right) \quad (1)$$

$$pdf_L(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma_L}\cdot\exp\left[-\frac{(x-\mu_L)^2}{2\cdot\sigma_L^2}\right] \sim Gauss\left(\mu_L,\sigma_L^2\right), \quad (2)$$

where μ_E and σ_E are the mean and standard deviation of the early-stage distribution respectively, and μ_L and σ_L are the mean and standard deviation of the late-stage distribution respectively.

The early-stage mean μ_E can be estimated from the early-stage data. In most practical applications, the early-stage data are collected to validate the early-stage design, before we move to the late stage. For this reason, we should already know the early-stage mean μ_E, before estimating the late-stage mean μ_L. Namely, we assume that the early-stage mean μ_E is provided as the input to the BMF framework. Our goal is to find an accurate estimation of the late-stage mean μ_L based on very few late-stage data.

Unlike the conventional moment estimation techniques that completely ignore the correlation between the early-stage mean μ_E and the late-stage mean μ_L, BMF fully exploits such correlation information to reduce the amount of required data at the late stage. It consists of two major steps: (i) statistically encoding the prior knowledge of the early-stage data as a prior distribution, and (ii) optimally determining the late-stage mean μ_L by combining the prior distribution and very few late-stage data. In what follows, we will describe the mathematical formulations of these two steps respectively.

Given the early-stage mean μ_E, we need to extract the prior knowledge that can be used to efficiently estimate the late-stage mean μ_L. On one hand, we expect that the early-stage mean μ_E and the late-stage mean μ_L are similar. In other words, the difference between μ_E and μ_L is small. On the other hand, the early-stage mean μ_E is not exactly identical to the late-stage mean μ_L due to multiple reasons. For example, comparing the schematic-level (i.e., early-stage) performance distribution with the post-layout (i.e., late-stage) performance distribution, the mean values of these two distributions can be different because the post-layout simulation of an AMS circuit includes the device and interconnect parasitics that are not available during the schematic-level simulation of the same circuit.

To statistically encode the "common" information between the early-stage mean μ_E and the late-stage mean μ_L, we conceptually assume that the "uncertainty" of the late-stage mean μ_L follows a prior distribution:

$$pdf(\mu_L) = \frac{1}{\sqrt{2\pi}\cdot\sigma_\mu}\cdot\exp\left[-\frac{(\mu_L-\mu_E)^2}{2\cdot\sigma_\mu^2}\right] \sim Gauss\left(\mu_E,\sigma_\mu^2\right), \quad (3)$$

where $pdf(\mu_L)$ represents a Gaussian distribution with the mean μ_E and the standard deviation σ_μ. In practice, the value of the standard deviation σ_μ is unknown; however, it can be estimated by the cross-validation method developed by the machine learning community [17], as will be further discussed later in this sub-section.

Figure 2 shows a simple example of the prior distribution for the late-stage mean μ_L. The prior distribution defined in (3) has a

two-fold meaning. First, the Gaussian distribution $pdf(\mu_L)$ is peaked at its mean value $\mu_L = \mu_E$, implying that the early-stage mean μ_E and the late-stage mean μ_L are likely to be similar. In other words, since the Gaussian distribution $pdf(\mu_L)$ exponentially decays with $(\mu_L - \mu_E)^2$, it is unlikely to observe a late-stage mean μ_L that is completely different from the early-stage mean μ_E. Second, the standard deviation σ_μ in (3) encodes our "confidence" of the prior knowledge. If the standard deviation σ_μ is small, the prior distribution is narrowly peaked around its mean value μ_E, implying that the late-stage mean is possibly close to the early-stage mean μ_E. Otherwise, if the standard deviation σ_μ is large, the prior distribution $pdf(\mu_L)$ widely spreads over a large range and the late-stage mean μ_L can possibly take a value that is far away from the early-stage mean μ_E.

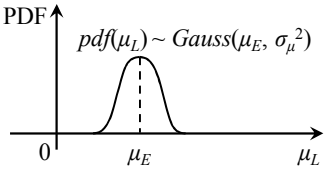

Figure 2. A simple example is shown to illustrate the approach of extracting the prior knowledge from the early-stage data where the late-stage mean μ_L is close to the early-stage mean μ_E as encoded by the prior distribution $pdf(\mu_L)$.

Given the prior distribution $pdf(\mu_L)$ in (3), we further combine $pdf(\mu_L)$ with a few late-stage random samples $\{x_{L,n}; n = 1, 2, \ldots, N\}$ to accurately estimate the late-stage mean μ_L. These late-stage samples can be generated by running Monte Carlo simulation for the AMS circuit. Once the late-stage data are available, they can tell us additional information about the late-stage mean μ_L and, hence, help us to accurately estimate μ_L.

Based on Bayes' theorem [16]-[17], the uncertainties of the late-stage mean μ_L after knowing the data $\{x_{L,n}; n = 1, 2, \ldots, N\}$ can be mathematically described by the following *posterior distribution*:

$$pdf(\mu_L|\mathbf{x}_L) \propto pdf(\mu_L) \cdot pdf(\mathbf{x}_L|\mu_L), \tag{4}$$

where $\mathbf{x}_L$ is a vector containing the late-stage random samples $\{x_{L,n}; n = 1, 2, \ldots, N\}$. In (4), the prior distribution $pdf(\mu_L)$ is defined by (3). The conditional distribution $pdf(\mathbf{x}_L \mid \mu_L)$ is referred to as the *likelihood function*. It measures the probability of observing the late-stage data $\{x_{L,n}; n = 1, 2, \ldots, N\}$ associated with the performance distribution $pdf_L(x)$:

$$pdf(\mathbf{x}_L|\mu_L) = \prod_{n=1}^{N} pdf_L(x_{L,n}|\mu_L). \tag{5}$$

Substituting (2) into (5) yields:

$$pdf(\mathbf{x}_L|\mu_L) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\cdot\sigma_L} \cdot \exp\left[-\frac{(x_{L,n}-\mu_L)^2}{2\cdot\sigma_L^2}\right]. \tag{6}$$

As shown in (6), the likelihood function $pdf(\mathbf{x}_L \mid \mu_L)$ depends on the late-stage mean μ_L that we aim to solve. The mean value μ_L controls the location of the probability density function $pdf_L(x)$ and, therefore, directly influences the likelihood function in (6).

Even after the late-stage data $\{x_{L,n}; n = 1, 2, \ldots, N\}$ are available, the late-stage mean μ_L is not deterministic. They must be modeled by the probability density function $pdf(\mu_L \mid \mathbf{x}_L)$ (i.e., the posterior distribution) in (4). Depending on the shape of the posterior distribution $pdf(\mu_L \mid \mathbf{x}_L)$, the late-stage mean μ_L does not take all possible values with equal probability. If the posterior distribution $pdf(\mu_L \mid \mathbf{x}_L)$ reaches its maximum value at μ_L*, the value μ_L* is considered as the optimal estimation of the late-stage mean, since it is the mean value that is most likely to occur. Such a method is referred to as the *maximum-a-posteriori* (MAP) estimation in the literature [16]-[17].

The aforementioned MAP estimation can be mathematically formulated as an optimization problem:

$$\underset{\mu_L}{maximize} \quad pdf(\mu_L) \cdot pdf(\mathbf{x}_L|\mu_L). \tag{7}$$

Substituting (3) and (6) into (7) and taking the logarithm for the merit function, we get:

$$\underset{\mu_L}{maximize} \quad -\frac{(\mu_L-\mu_E)^2}{2\cdot\sigma_\mu^2} - \sum_{n=1}^{N}\frac{(x_{L,n}-\mu_L)^2}{2\cdot\sigma_L^2} - \log\left(\sqrt{2\pi}\cdot\sigma_\mu\right) - N\cdot\log\left(\sqrt{2\pi}\cdot\sigma_L\right). \tag{8}$$

Based on the first-order optimality condition [21], we have:

$$-\frac{\mu_L-\mu_E}{\sigma_\mu^2} - \sum_{n=1}^{N}\frac{\mu_L-x_{L,n}}{\sigma_L^2} = 0. \tag{9}$$

Solving (9) results in the optimal estimation of the late-stage mean μ_L:

$$\mu_L = \frac{\sigma_L^2}{N\cdot\sigma_\mu^2+\sigma_L^2}\cdot\mu_E + \frac{\sigma_\mu^2}{N\cdot\sigma_\mu^2+\sigma_L^2}\cdot\sum_{n=1}^{N}x_{L,n}. \tag{10}$$

Defining the symbol:

$$\rho = \frac{\sigma_\mu}{\sigma_L}, \tag{11}$$

Eq. (10) can be re-written as:

$$\mu_L = \frac{1}{N\cdot\rho^2+1}\cdot\mu_E + \frac{\rho^2}{N\cdot\rho^2+1}\cdot\sum_{n=1}^{N}x_{L,n}. \tag{12}$$

Studying (12) reveals two important observations. First, the estimated late-stage mean μ_L is a function of the early-stage mean μ_E. If the early-stage mean μ_E appropriately carries the prior information of the late-stage mean μ_L, it can help us to accurately estimate the late-stage mean μ_L via Bayesian inference. Second, the optimal estimation of the late-stage mean μ_L depends on both the standard deviation σ_μ of the prior distribution $pdf(\mu_L)$ in (3) and the standard deviation σ_L of the late-stage performance distribution $pdf_L(x)$ in (2). In practice, both σ_μ and σ_L are unknown. However, they can be estimated by cross-validation [17]. Importantly, since μ_L is a function of the ratio ρ in (11), we only need to estimate the ratio ρ, instead of the individual values of σ_μ and σ_L, by cross-validation.

To demonstrate the efficacy of BMF for mean estimation, we consider an industrial example of high-speed I/O link. One of its critical performance metrics is time margin. It measures the eye width at the receiver and is directly related to the bit error rate (BER) of the link. In this example, our objective is to estimate the mean of time margin for 8 different configurations where one configuration referrers to a specific environmental corner (i.e., supply voltage and temperature) and a specific testing setup (i.e., channel for data transmission). Because accurately estimating the time margin requires us to pass a large number of random bits through the link and, hence, is extremely time-consuming, we cannot afford to simulate or measure a large number of dies (i.e., generate a large number of random samples) for mean estimation. For test and comparison purposes, two different estimators are implemented: (i) the conventional direct estimation based on sample mean [18], and (ii) the BMF technique described in this sub-section. When applying BMF, we borrow the statistical information from one configuration and consider it as our prior

knowledge to estimate the mean values of other configurations.

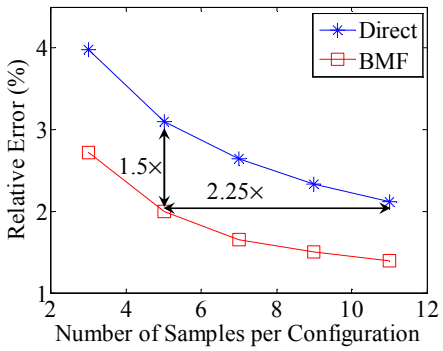

Figure 3. Relative error of mean estimation over 8 different configurations is shown for the conventional direct estimation and the BMF method.

Figure 3 shows the relative error of mean estimation over 8 different configurations. Studying Figure 3 reveals an important fact that given the same number of random samples, BMF consistently achieves 1.5× error reduction over the conventional direct estimation based on sample mean. On the other hand, to achieve the same accuracy, BMF can reduce the number of required random samples and, hence, the simulation and/or measurement cost by 2.25× over the conventional approach.

2.2 Distribution Estimation

In addition to moment estimation, BMF can also be applied to estimate the probability distribution of a given AMS performance metric x. Unlike most conventional algorithms that require a lot of data to determine the distribution $pdf(x)$ [18]-[20], BMF attempts to reuse the early-stage data so that the late-stage performance distribution can be accurately estimated with very few late-stage data [12].

Towards this goal, instead of assuming that the early-stage and late-stage probability density functions $pdf_E(x)$ and $pdf_L(x)$ follow the Gaussian distributions in (1)-(2), we approximate $pdf_E(x)$ and $pdf_L(x)$ as the linear combinations of a set of basis functions:

$$pdf_E(x) \approx \sum_{m=1}^{M} \alpha_{E,m} \cdot b_m(x) \tag{13}$$

$$pdf_L(x) \approx \sum_{m=1}^{M} \alpha_{L,m} \cdot b_m(x), \tag{14}$$

where $\{b_m(x);\ m = 1, 2, \ldots, M\}$ contains the basis functions (e.g., orthogonal polynomials [22], wavelet basis functions [23], etc.), $\{\alpha_{E,m};\ m = 1, 2, \ldots, M\}$ and $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ contain the early-stage and late-stage coefficients respectively, and M represents the total number of basis functions.

For a given set of basis functions $\{b_m(x);\ m = 1, 2, \ldots, M\}$, the performance distributions $pdf_E(x)$ and $pdf_L(x)$ are uniquely determined by the coefficients $\{\alpha_{E,m};\ m = 1, 2, \ldots, M\}$ and $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$. Here, we assume that the early-stage coefficients $\{\alpha_{E,m};\ m = 1, 2, \ldots, M\}$ are already known and the late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ should be estimated to determine the late-stage performance distribution $pdf_L(x)$. To statistically define the prior knowledge for the late-stage performance distribution $pdf_L(x)$, we consider the following two possible approaches.

- **Encoding the prior information based on coefficient magnitude**: Several previous works in the literature have demonstrated that when approximating a nonlinear function such as $pdf_E(x)$ or $pdf_L(x)$ in (13)-(14) by a set of appropriately selected basis functions (e.g., wavelet basis), many of the resulting coefficients are close to zero [23]. In other words, these coefficients carry a unique *sparse* pattern. To statistically encode such a sparse pattern as our prior knowledge, we exploit the fact that if the early-stage coefficient $\alpha_{E,m}$ has a large (or small) magnitude, it is likely that the late-stage coefficient $\alpha_{L,m}$ also has a large (or small) magnitude. In particular, we model each late-stage coefficient as a zero-mean Gaussian distribution:

$$pdf(\alpha_{L,m}) = \frac{1}{\sqrt{2\pi} \cdot \lambda \cdot |\alpha_{E,m}|} \cdot \exp\left(-\frac{\alpha_{L,m}^2}{2 \cdot \lambda^2 \cdot \alpha_{E,m}^2} \right), \tag{15}$$
$$\sim Gauss\left(0, \lambda^2 \cdot \alpha_{E,m}^2\right) \quad (m = 1,2,\cdots,M)$$

where the standard deviation of the Gaussian distribution is equal to $\lambda \cdot |\alpha_{E,m}|$. In (15), the parameter λ is positive and it controls the variance of the distribution. The appropriate value of λ can be determined by cross-validation [17].

The standard deviation $\lambda \cdot |\alpha_{E,m}|$ in (15) encodes the magnitude information of the late-stage coefficient $\alpha_{L,m}$. If the magnitude of the early-stage coefficient $|\alpha_{E,m}|$ is small and, hence, the standard deviation $\lambda \cdot |\alpha_{E,m}|$ is small, the prior distribution $pdf(\alpha_{L,m})$ is narrowly peaked around zero, implying that the late-stage coefficient $\alpha_{L,m}$ is possibly close to zero. Otherwise, if the magnitude of the early-stage coefficient $|\alpha_{E,m}|$ is large and, hence, the standard deviation $\lambda \cdot |\alpha_{E,m}|$ is large, the prior distribution $pdf(\alpha_{L,m})$ widely spreads over a large range and the late-stage coefficient $\alpha_{L,m}$ can possibly take a value that is far away from zero. Figure 4(a) shows a simple example of the prior distribution for two late-stage coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$ where $|\alpha_{E,1}|$ is small and $|\alpha_{E,2}|$ is large.

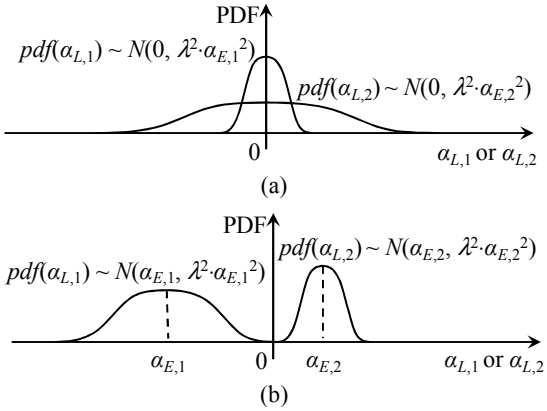

Figure 4. Two simple examples are shown to illustrate the approaches of extracting the prior knowledge from the early-stage data. (a) Encode the prior information based on coefficient magnitude where the late-stage coefficient $\alpha_{L,1}$ is possibly close to zero and the late-stage coefficient $\alpha_{L,2}$ can possibly be far away from zero. (b) Encode the prior information based on coefficient value where the late-stage coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$ are close to the early-stage coefficients $\alpha_{E,1}$ and $\alpha_{E,2}$ respectively, and both $\alpha_{L,1}$ and $\alpha_{L,2}$ are provided with a relatively equal opportunity to deviate from the corresponding early-stage coefficients $\alpha_{E,1}$ and $\alpha_{E,2}$.

- **Encoding the prior information based on coefficient value**:

If the values of the early-stage and late-stage coefficients, in addition to their magnitudes, are close, we can define a prior distribution that is different from (15):

$$pdf\left(\alpha_{L,m}\right)=\frac{1}{\sqrt{2\pi}\cdot\lambda\cdot\left|\alpha_{E,m}\right|}\cdot\exp\left[-\frac{\left(\alpha_{L,m}-\alpha_{E,m}\right)^2}{2\cdot\lambda^2\cdot\alpha_{E,m}^2}\right], \quad (16)$$
$$\sim Gauss\left(\alpha_{E,m},\lambda^2\cdot\alpha_{E,m}^2\right)\ \ (m=1,2,\cdots,M)$$

where $pdf(\alpha_{L,m})$ represents a Gaussian distribution with the mean value $\alpha_{E,m}$ and the standard deviation $\lambda\cdot|\alpha_{E,m}|$. Similar to (15), the parameter λ in (16) is positive and its value can be determined by cross-validation [17].

The prior distribution in (16) has a two-fold meaning. First, the Gaussian distribution $pdf(\alpha_{L,m})$ is peaked at its mean value $\alpha_{L,m} = \alpha_{E,m}$, implying that the early-stage coefficient $\alpha_{E,m}$ and the late-stage coefficient $\alpha_{L,m}$ are likely to be similar. Second, the standard deviation of the prior distribution $pdf(\alpha_{L,m})$ is proportional to $|\alpha_{E,m}|$. It means that the absolute difference between the late-stage coefficient $\alpha_{L,m}$ and the early-stage coefficient $\alpha_{E,m}$ can be large (or small), if the magnitude of the early-stage coefficient $|\alpha_{E,m}|$ is large (or small). Restating in words, each late-stage coefficient $\alpha_{L,m}$ has been provided with a relatively equal opportunity to deviate from the corresponding early-stage coefficient $\alpha_{E,m}$. Figure 4(b) shows a simple example of the aforementioned prior distribution for two late-stage coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$ where $\alpha_{E,1}$ is negative and $\alpha_{E,2}$ is positive.

Comparing the two prior distributions in (15) and (16), we notice that $pdf(\alpha_{L,m})$ in (16) carries the sign information (i.e., positive or negative) of the late-stage coefficient, while $pdf(\alpha_{L,m})$ in (15) does not. In practice, the efficacy of different prior definitions is case-dependent. How to choose the optimal prior distribution for a given application case remains an open question and should be considered as an important topic for future research.

To complete the definition of the prior distribution for all late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, ..., M\}$, we further assume that these coefficients are statistically independent and their joint distribution is represented as:

$$pdf\left(\boldsymbol{\alpha}_L\right)=\prod_{m=1}^{M}pdf\left(\alpha_{L,m}\right), \quad (17)$$

where $\boldsymbol{\alpha}_L$ is a vector containing all late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ and the prior distribution $pdf(\alpha_{L,m})$ for the mth late-stage coefficient $\alpha_{L,m}$ can be taken from either (15) or (16), depending on the specific application case. The independence assumption in (17) simply implies that we do not know the correlation information among these coefficients as our prior knowledge. The correlation information will be learned from the late-stage data, when the posterior distribution is calculated by Bayesian inference.

Given the prior distribution $pdf(\boldsymbol{\alpha}_L)$ defined in (17), we further combine $pdf(\boldsymbol{\alpha}_L)$ with a few late-stage random samples $\{x_{L,n};\ n = 1, 2, \ldots, N\}$ to calculate the posterior distribution:

$$pdf\left(\boldsymbol{\alpha}_L\middle|\mathbf{x}_L\right)\propto pdf\left(\boldsymbol{\alpha}_L\right)\cdot pdf\left(\mathbf{x}_L\middle|\boldsymbol{\alpha}_L\right), \quad (18)$$

where $\mathbf{x}_L$ is a vector containing the late-stage random samples $\{x_{L,n};\ n = 1, 2, \ldots, N\}$. In (18), the likelihood function $pdf(\mathbf{x}_L \mid \boldsymbol{\alpha}_L)$ measures the probability of observing the late-stage data $\{x_{L,n};\ n = 1, 2, \ldots, N\}$ associated with the performance distribution $pdf_L(x)$ in (14):

$$pdf\left(\mathbf{x}_L\middle|\boldsymbol{\alpha}_L\right)=\prod_{n=1}^{N}\sum_{m=1}^{M}\alpha_{L,m}\cdot b_m\left(x_{L,n}\right). \quad (19)$$

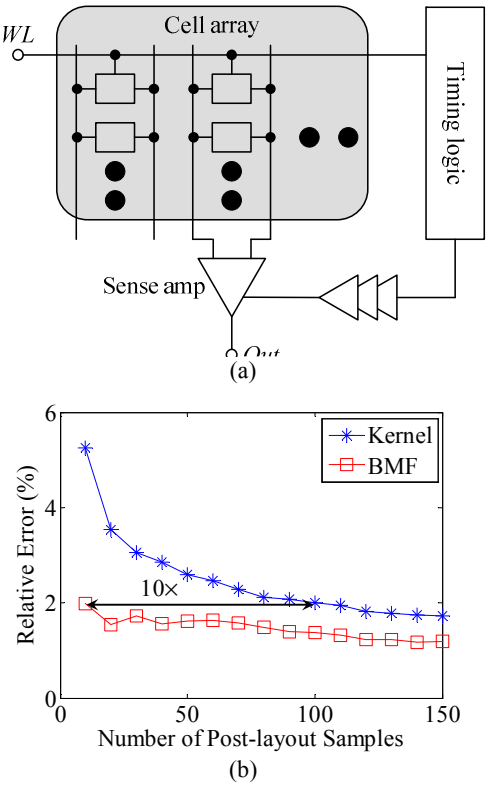

Figure 5. (a) A simplified circuit schematic is shown for an SRAM read path designed in a commercial 32 nm CMOS process. (b) Estimation error of the cumulative distribution function for the post-layout read path delay is shown for the conventional kernel estimation and the BMF technique.

To determine the late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ by MAP estimation, we formulate the following optimization:

$$\underset{\boldsymbol{\alpha}_L}{maximize}\quad pdf\left(\boldsymbol{\alpha}_L\right)\cdot pdf\left(\mathbf{x}_L\middle|\boldsymbol{\alpha}_L\right). \quad (20)$$

Substituting (17) and (19) into (20) and taking the logarithm for the merit function, we have:

$$\underset{\boldsymbol{\alpha}_L}{maximize}\quad \sum_{m=1}^{M}\log\left[pdf\left(\alpha_{L,m}\right)\right]+\sum_{n=1}^{N}\log\left[\sum_{m=1}^{M}\alpha_{L,m}\cdot b_m\left(x_{L,n}\right)\right]. \quad (21)$$

In addition, since the integral of any probability density function must be equal to 1, we need to further consider the following constraint when solving the late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$:

$$\int_{-\infty}^{+\infty}pdf_L(x)\cdot dx=\sum_{m=1}^{M}\alpha_{L,m}\cdot\int_{-\infty}^{+\infty}b_m(x)\cdot dx=1. \quad (22)$$

Combining (21) and (22) results in the following constrained optimization problem:

$$\underset{\boldsymbol{\alpha}_L}{maximize}\quad \sum_{m=1}^{M}\log\left[pdf\left(\alpha_{L,m}\right)\right]+\sum_{n=1}^{N}\log\left[\sum_{m=1}^{M}\alpha_{L,m}\cdot b_m\left(x_{L,n}\right)\right]$$
$$subject\ to\quad \sum_{m=1}^{M}\alpha_{L,m}\cdot\int_{-\infty}^{+\infty}b_m(x)\cdot dx=1 \quad (23)$$

It can be proven that the merit function in (23) is concave [12], if the prior distribution $pdf(\alpha_{L,m})$ is taken from either (15) or (16). Furthermore, the constraint function is simply linear with respect to the problem unknowns $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$, implying that

the constraint set is convex. For these reasons, the optimization in (23) is a convex programming problem and can be solved both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global optimum) [21]. Once the late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ are found from (23), the late-stage performance distribution $pdf_L(x)$ in (14) is determined.

As an example to demonstrate the efficacy of BMF for distribution estimation, Figure 5(a) shows the simplified circuit schematic of an SRAM (static random-access memory) read path designed in a commercial 32 nm CMOS process. In this example, our objective is to reuse the schematic-level (i.e. early-stage) simulation data to efficiently estimate the post-layout (i.e., late-stage) probability distribution of the read path delay from the word line (*WL*) to the sense amplifier output (*Out*). For testing and comparison purposes, two different techniques are implemented: (i) the conventional kernel estimation based on Gaussian kernel with optimal bandwidth [24], and (ii) the BMF technique based on DCT (discrete cosine transform) basis functions [23]. Figure 5(b) shows the estimation error for the cumulative distribution function (i.e., the integral of the probability density function). Note that the conventional kernel estimation requires 10× more simulation samples than BMF to achieve the same accuracy. In other words, BMF achieves 10× runtime speedup over kernel estimation in this example.

3. POST-SILICON TUNING

In this section, we further extend BMF to the application of post-silicon tuning. In particular, we focus on the problem of tuning policy generation with consideration of large-scale process shift over time. Namely, we aim to solve a regression modeling problem to find the mapping between the measured AMS performance and the control knob setup:

$$f(\mathbf{x}) \approx \sum_{m=1}^{M} \alpha_m \cdot b_m(\mathbf{x}), \tag{24}$$

where $\mathbf{x}$ is a vector containing the measurement results of circuit performance, f represents the optimal value of a control knob, $\{\alpha_m;\ m = 1, 2, ..., M\}$ contains the model coefficients, $\{b_m(\mathbf{x});\ m = 1, 2, ..., M\}$ contains the basis functions (e.g., polynomials), and M is the total number of basis functions. Note that the regression modeling problem in (24) is substantially *different* from the distribution estimation problem in (13)-(14), even though both problems involve the linear combination of M basis functions.

There have been many regression modeling algorithms that were developed in the literature [25]-[34]. Most of these existing methods require a lot of data to fit the model, especially if the function $f(\mathbf{x})$ is strongly nonlinear and/or high-dimensional. BMF, however, reuses the early-stage data so that $f(\mathbf{x})$ can be accurately approximated with very few late-stage data [14]-[15]. In what follows, we will describe the BMF formulation for the application of post-silicon tuning.

Similar to (13)-(14), we consider two different regression models: the early-stage model $f_E(\mathbf{x})$ and the late-stage model $f_L(\mathbf{x})$:

$$f_E(\mathbf{x}) \approx \sum_{m=1}^{M} \alpha_{E,m} \cdot b_m(\mathbf{x}) \tag{25}$$

$$f_L(\mathbf{x}) \approx \sum_{m=1}^{M} \alpha_{L,m} \cdot b_m(\mathbf{x}), \tag{26}$$

where $\{\alpha_{E,m};\ m = 1, 2, ..., M\}$ and $\{\alpha_{L,m};\ m = 1, 2, ..., M\}$ represent the early-stage and late-stage coefficients, respectively. The early-stage model $f_E(\mathbf{x})$ can be fitted by either the pre-silicon data collected from numerical simulation or the post-silicon data measured from previous tape-out. We assume that the early-stage model $f_E(\mathbf{x})$ is already available. Given the early-stage coefficients $\{\alpha_{E,m};\ m = 1, 2, ..., M\}$, we can define the prior distribution $pdf(\boldsymbol{\alpha}_L)$ based on (15)-(17), similar to the case of pre-silicon validation described in Section 2.2.

Once the prior distribution $pdf(\boldsymbol{\alpha}_L)$ is defined, we collect a few late-stage data $\{(\mathbf{x}^{(n)}, f_L^{(n)});\ n = 1, 2, \ldots, N\}$, where $\mathbf{x}^{(n)}$ and $f_L^{(n)}$ are the values of $\mathbf{x}$ and $f_L(\mathbf{x})$ at the nth data point respectively, by measuring several chips from the new tape-out. Next, we combine the prior distribution $pdf(\boldsymbol{\alpha}_L)$ with the late-stage data $\{(\mathbf{x}^{(n)}, f_L^{(n)});\ n = 1, 2, \ldots, N\}$ to solve the late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, ..., M\}$ by MAP estimation. As such, a new tuning policy is efficiently generated to accommodate the process shift. To this end, we formulate the following posterior distribution:

$$pdf(\boldsymbol{\alpha}_L | \mathbf{f}_L) \propto pdf(\boldsymbol{\alpha}_L) \cdot pdf(\mathbf{f}_L | \boldsymbol{\alpha}_L), \tag{27}$$

where $\boldsymbol{\alpha}_L$ is a vector containing all late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ and $\mathbf{f}_L$ is a vector containing the late-stage data $\{f_L^{(n)};\ n = 1, 2, \ldots, N\}$.

To derive the likelihood function $pdf(\mathbf{f}_L \mid \boldsymbol{\alpha}_L)$, we further assume that the error for the late-stage model $f_L(\mathbf{x})$ can be represented as a random variable that follows a zero-mean Gaussian distribution and, hence, the approximate equality in (26) can be re-written as:

$$f_L(\mathbf{x}) = \sum_{m=1}^{M} \alpha_{L,m} \cdot b_m(\mathbf{x}) + \varepsilon_L, \tag{28}$$

where ε_L denotes the modeling error:

$$pdf(\varepsilon_L) = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left(-\frac{\varepsilon_L^2}{2 \cdot \sigma_0^2}\right) \sim Gauss(0, \sigma_0^2). \tag{29}$$

In (29), the standard deviation σ_0 indicates the magnitude of the modeling error. Its value can be determined by cross-validation [17]. Given (28)-(29), since the modeling error associated with the nth data point $(\mathbf{x}^{(n)}, f_L^{(n)})$ is simply one sample of the random variable ε_L, it follows the Gaussian distribution:

$$f_L^{(n)} - \sum_{m=1}^{M} \alpha_{L,m} \cdot b_m(\mathbf{x}^{(n)}) \sim Gauss(0, \sigma_0^2). \tag{30}$$

Therefore, the probability of observing the nth data point is:

$$pdf(f_L^{(n)} | \boldsymbol{\alpha}_L) = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left\{-\frac{1}{2 \cdot \sigma_0^2} \cdot \left[f_L^{(n)} - \sum_{m=1}^{M} \alpha_{L,m} \cdot b_m(\mathbf{x}^{(n)})\right]^2\right\}. \tag{31}$$

Assuming that all sampling points are independently generated, we can write the likelihood function $pdf(\mathbf{f}_L \mid \boldsymbol{\alpha}_L)$ as:

$$pdf(\mathbf{f}_L | \boldsymbol{\alpha}_L) = \prod_{n=1}^{N} pdf(f_L^{(n)} | \boldsymbol{\alpha}_L). \tag{32}$$

Combining (17), (27) and (32) results in the following optimization formulation for MAP estimation:

$$\underset{\boldsymbol{\alpha}_L}{maximize} \quad \sum_{m=1}^{M} \log[pdf(\alpha_{L,m})] - \frac{1}{2 \cdot \sigma_0^2} \cdot \sum_{n=1}^{N} \left[f_L^{(n)} - \sum_{m=1}^{M} \alpha_{L,m} \cdot b_m(\mathbf{x}^{(n)})\right]^2. \tag{33}$$

Note that the MAP formulation in (33) is completely different from the formulation in (23) that was previously developed for

distribution estimation of pre-silicon validation. It can be proven that the unconstrained optimization in (33) is convex [14]-[15] and, hence, can be solved both efficiently and robustly [21]. Once the late-stage coefficients $\{\alpha_{L,m};\ m = 1, 2, \ldots, M\}$ are found from (33), the new tuning policy (i.e., the late-stage model) $f_L(\mathbf{x})$ in (26) is determined.

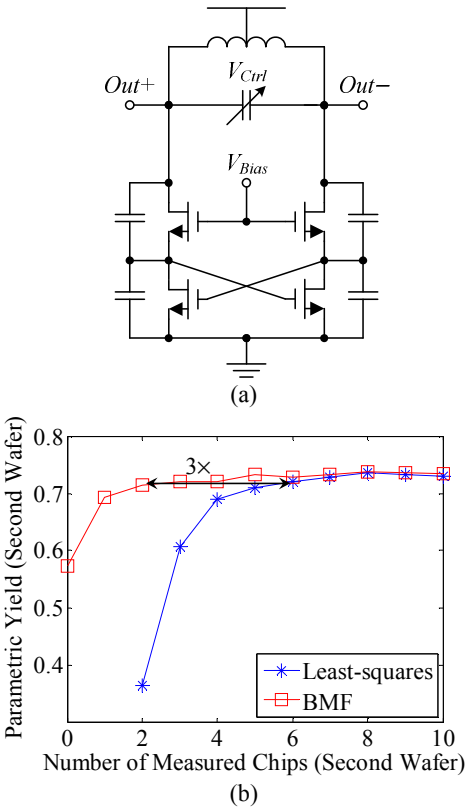

Figure 6. (a) A simplified circuit schematic is shown for a VCO designed in a commercial 32 nm CMOS process where 116 chips from two wafers are measured. (b) Parametric yield achieved by post-silicon tuning for the second wafer is shown for two different tuning policies generated by the conventional least-squares fitting and the BMF technique respectively.

As an example for demonstration, we apply BMF to generate the tuning policy for a VCO (voltage-controlled oscillator) designed in a commercial 32 nm CMOS process, as shown in Figure 6(a). In this example, the objective of post-silicon tuning is to appropriately set the bias voltage V_{Bias} so that the phase noise of the VCO is minimized. To this end, we measure a number of important performance metrics (i.e., oscillation frequency, oscillation amplitude, bias current, etc.) under the nominal bias voltage and then determine the optimal value of the control knob V_{Bias} based on these measurement results.

For this VCO example, two wafers with 116 VCO chips in total are produced by an industrial partner. These two wafers are not manufactured at the same time and a significant process shift is observed. Our goal is to reuse the measurement data from the first wafer (i.e., the early-stage data) to efficiently generate the tuning policy for the second wafer in order to accommodate the large-scale process shift. For testing and comparison purposes, two different techniques are implemented: (i) the conventional least-squares fitting [25], and (ii) the BMF technique described in this section. Figure 6(b) plots the parametric yield achieved by post-silicon tuning for the second wafer as a function of the number of measured chips. Note that BMF only needs to measure 2 chips from the second wafer in order to generate the new tuning policy. To achieve the same parametric yield, the conventional least-squares fitting must measure 6 chips (3× more) in this example.

4. CONCLUSIONS

In this paper, a Bayesian Model Fusion (BMF) framework is described to reduce the simulation and/or measurement cost for both pre-silicon validation and post-silicon tuning. The key idea of BMF is to reuse the data that was previously generated at an early stage. It statistically extracts the prior knowledge from the early-stage data to facilitate efficient late-stage validation and/or tuning with minimal simulation and/or measurement cost. BMF is mathematically derived from the theory of Bayesian theory and its efficacy is demonstrated for several industrial circuit examples in this paper. We envision that BMF can be generally applied to a broad range of circuit applications and a large number of follow-up research works are needed in this emerging area.

5. ACKNOWLEDGEMENTS

This work is supported in part by Intel Corporation and the National Science Foundation under contract CCF–1316363.

6. REFERENCES

[1] S. Nassif, "Modeling and analysis of manufacturing variations," *IEEE CICC*, pp. 223-228, 2001.

[2] X. Li, et al., *Statistical Performance Modeling and Optimization*, Now Publishers, 2007.

[3] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.

[4] R. Rutenbar, et al., "Hierarchical modeling, optimization and synthesis for system-level analog and RF designs," *Proceedings of The IEEE*, vol. 95, no. 3, pp. 640-669, Mar. 2007.

[5] A. Tang, et al., "A low-overhead self-healing embedded system for ensuring high yield and long-term sustainability of 60GHz 4Gb/s radio-on-a-chip," *IEEE ISSCC*, pp. 316-318, 2012.

[6] S. Bowers, et al., "A fully-integrated self-healing power amplifier," *IEEE RFIC*, 2012.

[7] J. Plouchart, et al., "A 23.5GHz PLL with an adaptively biased VCO in 32nm SOI-CMOS," *IEEE CICC*, 2012.

[8] N. Kupp, et al., "Post-production performance calibration in analog/RF devices," *IEEE ITC*, pp. 245-254, 2010.

[9] D. Han, et al., "DSP-driven self-tuning of RF circuits for process-induced performance variability," *IEEE Trans. on VLSI*, vol. 18, no. 2, pp. 305-314, Feb. 2010.

[10] E. Acar et al., "Low cost MIMO testing for RF integrated circuits," *IEEE Trans. on VLSI*, vol. 18, no. 9, pp. 1348-1356, Sep. 2010.

[11] S. Yaldiz, et al., "Indirect phase noise sensing for self-healing voltage controlled oscillators," *IEEE CICC*, 2011.

[12] X. Li, et al., "Efficient parametric yield estimation of analog/mixed-signal circuits via Bayesian model fusion," *IEEE ICCAD*, pp. 627-634, 2012.

[13] C. Gu, et al., "Efficient moment estimation with extremely small sample size via Bayesian inference for analog/mixed-signal validation," *IEEE DAC*, 2013.

[14] F. Wang, et al., "Bayesian model fusion: large-scale

performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE DAC*, 2013.
[15] S. Sun, et al., "Indirect performance sensing for on-chip analog self-healing via Bayesian model fusion," *IEEE CICC*, 2013.
[16] G. Casella, et al., *Statistical Inference*, Duxbury Press, 2001.
[17] C. Bishop, *Pattern Recognition and Machine Learning*, Prentice Hall, 2007.
[18] A. Papoulis, et al., *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 2001.
[19] X. Li, et al., "Asymptotic probability extraction for nonnormal performance distributions," *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 16-37, Jan. 2007.
[20] A. Graupner, et al., "Statistical analysis of analog structures through variance calculation," *IEEE Trans. on CAS－I*, vol. 49, no. 8, pp. 1071-1078, Aug. 2002.
[21] S. Boyd, et al., *Convex Optimization*. Cambridge University Press, 2004.
[22] P. Beckmann, *Orthogonal Polynomials for Engineers and Physicists*, Golem Press, 1973.
[23] R. Gonzalez, et al., *Digital Image Processing*, Prentice Hall, 2007.
[24] B. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC, 1986.
[25] G. Box, et al., *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, 1987.
[26] J. Swidzinski, et al., "Statistical behavioral modeling of integrated circuits," *IEEE ISCAS*, vol. 6, pp. 98-101, 1998.
[27] W. Daems, et al, "An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits," *IEEE DAC*, pp. 431-436, 2002.
[28] H. Liu, et al., "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," *IEEE DAC*, pp. 437-442, 2002.
[29] X. Li, et al., "Projection-based performance modeling for inter/intra-die variations," *IEEE ICCAD*, pp. 721-727, 2005.
[30] Z. Feng, et al., "Performance-oriented statistical parameter reduction of parameterized systems via reduced rank regression," *IEEE ICCAD*, pp. 868-875, 2006.
[31] A. Singhee, et al., "Beyond low-order statistical response surfaces: latent variable regression for efficient, highly nonlinear fitting," *IEEE DAC*, pp. 256-261, 2007.
[32] A. Mitev, et al., "Principle Hessian direction based parameter reduction for interconnect networks with process variation," *IEEE ICCAD*, pp. 632-637, 2007.
[33] T. McConaghy, et al., "Template-free symbolic performance modeling of analog circuits via canonical-form functions and genetic programming," *IEEE Trans. on CAD*, vol. 28, no. 8, pp. 1162-1175, Aug. 2009.
[34] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits," *IEEE Trans. on CAD*, vol. 29, no. 11, pp. 1661-1668, Nov. 2010.