

# Correlated Bayesian Model Fusion: Efficient Performance Modeling of Large-Scale Tunable Analog/RF Integrated Circuits

Fa Wang and Xin Li

ECE Department, Carnegie Mellon University, Pittsburgh, PA 15213  
[{fwang1,xinli}@ece.cmu.edu](mailto:{fwang1,xinli}@ece.cmu.edu)

## ABSTRACT

Tunable circuit has emerged as a promising methodology to address the grand challenge posed by process variations. Efficient high-dimensional performance modeling of tunable analog/RF circuits is an important yet challenging task. In this paper, we propose a novel performance modeling approach for tunable circuits, referred to as Correlated Bayesian Model Fusion (C-BMF). The key idea is to encode the correlation information for both model template and coefficient magnitude among different knob configurations by using a unified prior distribution. The prior distribution is then combined with a few simulation samples via Bayesian inference to efficiently determine the unknown model coefficients. Two circuit examples designed in a commercial 32nm SOI CMOS process demonstrate that C-BMF achieves more than 2 $\times$  cost reduction over the traditional state-of-the-art modeling technique without surrendering any accuracy.

## 1. INTRODUCTION

The advent of nanoscale technology makes analog/RF integrated circuit (IC) increasingly susceptible to process variations. Process variations, including inter-die variations and local mismatches, significantly impact the parametric yield of analog/RF circuit, and must be properly handled at all levels of design hierarchy [1]-[2]. Traditional approaches based on over-design [3]-[4] are not sufficient to maintain high parametric yield [5]-[6], due to the large-scale process variations and aggressive design specifications at advanced technology nodes.

In this context, tunable circuit has emerged as a promising methodology to address the variability issue [6]-[11]. It employs a set of tuning knobs (e.g., bias current) to facilitate post-silicon tuning. In particular, a tunable circuit possesses multiple ‘states’ associated with tuning knobs that can be adaptively selected after manufacturing. For each particular process corner and/or environmental condition, the tunable circuit is able to adaptively select the optimal state and, hence, achieve high parametric yield.

Since tunable circuit closely interacts with process variations, the variability associated with tunable circuit must be appropriately modeled. Toward this goal, performance modeling is an important task where circuit-level performance (e.g. phase noise) is approximated as an analytical function (e.g. polynomial) of device-level variations (e.g.  $\Delta V_{TH}$ ). The performance model, once built, can be applied to various important applications, such

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

DAC '16, June 05-09, 2016, Austin, TX, USA

© 2016 ACM. ISBN 978-1-4503-4236-0/16/06...\$15.00

DOI: <http://dx.doi.org/10.1145/2897937.2897999>

as yield estimation [12]-[13], corner extraction [14], design optimization [15], etc. Historically, the performance modeling problem of non-tunable circuit has been thoroughly studied [16]-[18]. For tunable circuit, however, this fundamental modeling problem can be extremely difficult with the following characteristics:

- *High-dimensional variation space*: A large number of device-level random variables must be used to model the process variations associated with a tunable analog/RF system at advanced technology nodes.
- *Large number of knob configurations*: To maximize tuning range and resolution, a large number of knob configurations (i.e. ‘states’) should be designed. This, however, significantly increases the complexity of performance modeling, since a unique performance model is required to accurately capture the variability at each state.
- *Correlated knob configurations*: Since different states are associated with the same tunable circuit, the performance models of the different states are correlated. In particular, the model template and model coefficients are expected to be similar among these states.

It is important to note that most traditional approaches [16]-[21] are not fully equipped for performance modeling of tunable circuit. Taking sparse regression [16]-[17] as an example, the sparsity of model coefficients is explored to improve efficiency. However, no correlation is considered between different states. In other words, performance modeling is done for each state independently and the performance modeling cost would quickly become intractable as the number of states increases. Several other performance modeling methods (e.g., simultaneous orthogonal matching pursuit [19], multi-mode multi-corner sparse regression [20], group lasso [21], etc.) have been proposed to exploit the aforementioned correlation information. However, these methods only take into account the shared model template among different states and they ignore the correlation of coefficient magnitude. In Bayesian model fusion (BMF) [18], a prior distribution is defined to encode the prior knowledge. However, the prior distribution in [18] is not able to appropriately encode the correlation information for tunable circuit.

In this paper, we propose a new Correlated Bayesian model fusion (C-BMF) technique to facilitate large-scale performance modeling of tunable analog/RF circuit. The proposed C-BMF method is motived by the fact that performance models associated with different states are correlated. Our key idea is to encode the correlation information for both model template and coefficient magnitude among different states by using a unified prior distribution. The prior distribution is then combined with a few simulation samples via Bayesian inference [22]-[23] to efficiently determine the unknown model coefficients. In addition, efficient and robust hyper-parameter inference is proposed to solve a large number of hyper-parameters associated with C-BMF. It consists of two major steps: (i) estimating the hyper-parameters to obtain an initial guess using a heuristic approach, and (ii) optimizing the

hyper-parameter values using an expectation-maximization algorithm [22]. As will be demonstrated by our numerical examples in Section 4, the proposed C-BMF approach is able to achieve 2× runtime speedup over the state-of-the-art modeling technique without surrendering any accuracy.

The remainder of this paper is organized as follows. We briefly review the background of performance modeling for tunable circuit in Section 2, and then describe the proposed C-BMF approach in Section 3. Two circuit examples are presented to demonstrate the efficacy of C-BMF in Section 4. Finally, we conclude in Section 5.

## 2. BACKGROUND

Given a tunable analog/RF circuit (e.g. a down-conversion mixer) with totally  $K$  states (i.e. knob configurations), the performance model of the  $k$ -th state can be represented by an analytical function (e.g., polynomial) of device-level variations (e.g.,  $\Delta V_{TH}$ ):

$$y_k \approx f_k(\mathbf{x}) = \sum_{m=1}^M \alpha_{k,m} \cdot b_m(\mathbf{x}) \quad (k=1,2,\dots,K), \quad (1)$$

where  $\{y_k; k = 1, 2, \dots, K\}$  contains the performance of interest (PoIs) associated with the  $K$  states,  $\mathbf{x}$  is a vector representing the device-level variations,  $f_k(\mathbf{x})$  denotes the performance model of interest which establishes a mapping from  $\mathbf{x}$  to  $y_k$ ,  $\{\alpha_{k,m}; m = 1, 2, \dots, M\}$  contains the model coefficients of  $f_k(\mathbf{x})$ ,  $\{b_m(\mathbf{x}); m = 1, 2, \dots, M\}$  contains the basis functions (e.g., linear or quadratic polynomials), and  $M$  denotes the total number of basis functions. Without loss of generality, we assume that the same set of basis functions  $\{b_m(\mathbf{x}); m = 1, 2, \dots, M\}$  are shared among all the states.

In order to determine the performance models in (1), we need to find the model coefficients  $\{\alpha_{k,m}; k = 1, 2, \dots, K; m = 1, 2, \dots, M\}$ . Toward this goal, the traditional least-squares fitting method first collects a number of sampling points of  $\mathbf{x}$  and  $\{y_k; k = 1, 2, \dots, K\}$  for all  $K$  states, and then solves the model coefficients from the following optimization problems:

$$\min_{\mathbf{a}_k} \|\mathbf{y}_k - \mathbf{B}_k \cdot \mathbf{a}_k\|^2 \quad (k=1,2,\dots,K), \quad (2)$$

where

$$\mathbf{B}_k = \begin{bmatrix} b_1(\mathbf{x}_k^{(1)}) & b_2(\mathbf{x}_k^{(1)}) & \cdots & b_M(\mathbf{x}_k^{(1)}) \\ b_1(\mathbf{x}_k^{(2)}) & b_2(\mathbf{x}_k^{(2)}) & \cdots & b_M(\mathbf{x}_k^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ b_1(\mathbf{x}_k^{(N)}) & b_2(\mathbf{x}_k^{(N)}) & \cdots & b_M(\mathbf{x}_k^{(N)}) \end{bmatrix} \quad (k=1,2,\dots,K) \quad (3)$$

$$\mathbf{a}_k = [\alpha_{k,1} \ \alpha_{k,2} \ \cdots \ \alpha_{k,M}]^T \quad (k=1,2,\dots,K) \quad (4)$$

$$\mathbf{y}_k = [y_k^{(1)} \ y_k^{(2)} \ \cdots \ y_k^{(N)}]^T \quad (k=1,2,\dots,K). \quad (5)$$

In (3)-(5),  $\mathbf{x}_k^{(n)}$  is the value of  $\mathbf{x}$  at the  $n$ -th sampling point of the  $k$ -th state,  $y_k^{(n)}$  is the value of  $y_k$  at the  $n$ -th sampling point,  $\|\cdot\|_2$  stands for the L<sub>2</sub>-norm of a vector, and  $N$  represents the total number of sampling points collected for each state. To avoid overfitting, the number of sampling points must be substantially greater than the number of unknown coefficients.

To reduce the number of samples required for performance modeling, the simultaneous orthogonal matching pursuit (S-OMP) algorithm [19] has been proposed in the literature. The key idea of S-OMP is to explore the sparse property and the correlation of model template between different states. As an extension of the orthogonal matching pursuit (OMP) algorithm [16], S-OMP

naturally inherits the assumption of sparse regression. In other words, S-OMP exploits the fact that only a small number of basis functions in  $\{b_m(\mathbf{x}); m = 1, 2, \dots, M\}$  are important and, hence, should be used to approximate  $\{y_k; k = 1, 2, \dots, K\}$ . More importantly, S-OMP also explores the correlation of model template among  $\{y_k; k = 1, 2, \dots, K\}$ . In particular, S-OMP assumes that different states possess the same set of important basis functions. This is usually a valid assumption in practice, because different states are associated with the same tunable circuit.

While S-OMP has been successfully applied to many practical applications (e.g. wafer spatial pattern analysis [19]), it is possible to explore other correlation information between states to further improve the efficiency of performance modeling for tunable circuits. Motivated by this observation, we will propose a novel C-BMF approach that takes advantage of the correlation of coefficient magnitude in Section 3.

## 3. PROPOSED APPROACH

Similar to S-OMP, the proposed C-BMF method relies on two assumptions: (i) the sparsity of model template, and (ii) the correlation of model template. However, unlike S-OMP which ignores the correlation of coefficient magnitude among different states, C-BMF also encodes such magnitude correlation to improve modeling accuracy. In what follows, we will discuss the mathematical formulation of C-BMF in detail.

### 3.1 Prior Knowledge Definition

To start with, we consider the performance models of  $K$  states in a tunable circuit as defined in (1). We rewrite their model coefficients in a concatenated form as:

$$\mathbf{a} = [\mathbf{\beta}_1^T \ \mathbf{\beta}_2^T \ \cdots \ \mathbf{\beta}_M^T]^T, \quad (6)$$

where

$$\mathbf{\beta}_m = [\alpha_{1,m} \ \alpha_{2,m} \ \cdots \ \alpha_{K,m}]^T \quad (m=1,2,\dots,M), \quad (7)$$

$\{\alpha_{k,m}; k = 1, 2, \dots, K; m = 1, 2, \dots, M\}$  are the model coefficients defined in (1), and  $\mathbf{a}$  is the combined column vector with size of  $M \cdot K$ . In (6), the model coefficients are organized such that the coefficients associated with the  $m$ -th basis function are grouped together in  $\mathbf{\beta}_m$  ( $m = 1, 2, \dots, M$ ). Since the model coefficients in  $\mathbf{\beta}_m$  are related to the same basis function, we assume that they are statistically correlated and define their prior distribution as:

$$pdf(\mathbf{\beta}_m) \sim N(\mathbf{0}, \lambda_m \cdot \mathbf{R}_m) \quad (m=1,2,\dots,M). \quad (8)$$

In (8),  $\{\lambda_m; m = 1, 2, \dots, M\}$  are the hyper-parameters that control the sparsity of the basis functions, and  $\{\mathbf{R}_m; m = 1, 2, \dots, M\}$  are the positive definite matrices that quantify the correlation among model coefficients.

Studying (8) yields several important observations. First, the sparse property is naturally encoded in  $\{\lambda_m; m = 1, 2, \dots, M\}$ . To understand this, we consider the  $m$ -th basis function where  $\lambda_m$  is zero. In this case, the variance of the zero-mean Gaussian distribution in (8) is also zero. As such, the coefficients  $\mathbf{\beta}_m$  can only take zero values based on their prior distribution, thereby implying the sparsity of  $\mathbf{\beta}_m$ . Second, the correlation of model template is also encoded by the mathematical representation in (8). In particular, all the model coefficients in  $\mathbf{\beta}_m$  share the same sparse pattern, because their prior distribution is controlled by the same hyper-parameter  $\lambda_m$ . Third, but most importantly, the correlation of coefficient magnitude is also encoded by this prior definition, as long as the covariance metrics  $\{\mathbf{R}_m; m = 1, 2, \dots, M\}$

are not diagonal. In this paper, we further assume that:

$$\mathbf{R}_1 = \mathbf{R}_2 = \dots = \mathbf{R}_M = \mathbf{R}, \quad (9)$$

in order to reduce the number of hyper-parameters and avoid overfitting.

As shown in Figure 1, the prior knowledge of sparsity and correlation is encoded by the hyper-parameters. To complete the definition of prior distribution for all model coefficients in  $\boldsymbol{\alpha}$ , we further assume that  $\beta_i$  and  $\beta_j$  are statistically independent for any  $i \neq j$ . Therefore, the joint distribution can be represented as:

$$pdf(\boldsymbol{\alpha}) \sim N(\mathbf{0}, \mathbf{A}), \quad (10)$$

where

$$\mathbf{A} = \begin{bmatrix} \lambda_1 \cdot \mathbf{R} & & \\ & \ddots & \\ & & \lambda_M \cdot \mathbf{R} \end{bmatrix}. \quad (11)$$

The independence assumption in (10)-(11) simply indicates that we do not know any correlation information between different basis functions as our prior knowledge. Such correlation will be learned from sampling points, when the posterior distribution is calculated from Bayesian inference in Section 3.2.

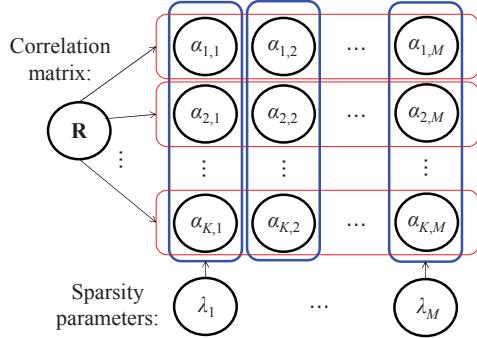


Figure 1. The prior knowledge for model coefficients is illustrated. Each row (shown by the red box) represents the model coefficients of a particular state, while each column (shown by the blue box) represents the model coefficients associated with the same basis function. The hyper-parameters  $\{\lambda_m; m = 1, 2, \dots, M\}$  and  $\mathbf{R}$  are defined to encode the sparsity and correlation respectively.

### 3.2 Maximum-A-Posteriori Estimation

Once the prior distribution  $pdf(\boldsymbol{\alpha})$  is defined in (10)-(11), we will combine  $pdf(\boldsymbol{\alpha})$  with a number of samples collected from  $K$  states  $\{(\mathbf{x}_k^{(n)}, y_k^{(n)}); n = 1, 2, \dots, N; k = 1, 2, \dots, K\}$  to solve the model coefficients  $\boldsymbol{\alpha}$  by maximum-a-posteriori (MAP) estimation. According to Bayes' theorem, the posterior distribution is proportional to the prior distribution  $pdf(\boldsymbol{\alpha})$  multiplied by the likelihood function, which can be represented as [22]:

$$pdf(\boldsymbol{\alpha} | \mathbf{y}) \propto pdf(\boldsymbol{\alpha}) \cdot pdf(\mathbf{y} | \boldsymbol{\alpha}). \quad (12)$$

In (12),  $pdf(\mathbf{y} | \boldsymbol{\alpha})$  is the likelihood function, where

$$\mathbf{y} = \begin{bmatrix} y_1^{(1)} & \dots & y_K^{(1)} & y_1^{(2)} & \dots & y_K^{(2)} & y_1^{(N)} & \dots & y_K^{(N)} \end{bmatrix}^T \quad (13)$$

is a vector containing  $N \cdot K$  sampling points collected from  $K$  states. The likelihood function  $pdf(\mathbf{y} | \boldsymbol{\alpha})$  represents the probability of observing these samples given the model coefficients  $\boldsymbol{\alpha}$ .

To derive the likelihood function, we assume that the error associated with the performance model  $f_k(\mathbf{x})$  follows a zero-mean Gaussian distribution. We therefore rewrite (1) as:

$$y_k = \sum_{m=1}^M \alpha_{k,m} \cdot b_m(\mathbf{x}) + \varepsilon \quad (k = 1, 2, \dots, K), \quad (14)$$

where  $\varepsilon$  represents the modeling error with the distribution:

$$pdf(\varepsilon) = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left(-\frac{\varepsilon^2}{2 \cdot \sigma_0^2}\right) \sim N(0, \sigma_0^2). \quad (15)$$

In (15) the standard deviation  $\sigma_0$  controls the magnitude of the modeling error. Combining (14)-(15) yields the probability of observing a particular sampling point  $(\mathbf{x}_k^{(n)}, y_k^{(n)})$ :

$$pdf(y_k^{(n)} | \boldsymbol{\alpha}) \sim \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left(-\frac{1}{2 \cdot \sigma_0^2} \left[ y_k^{(n)} - \sum_{m=1}^M \alpha_{k,m} \cdot b_m(\mathbf{x}_k^{(n)}) \right]^2\right). \quad (16)$$

Given that all sampling points are independently generated, the joint likelihood function  $pdf(\mathbf{y} | \boldsymbol{\alpha})$  can be written as:

$$pdf(\mathbf{y} | \boldsymbol{\alpha}, \sigma_0) \sim N(\mathbf{D}\boldsymbol{\alpha}, \sigma_0^2 \mathbf{I}), \quad (17)$$

where  $\mathbf{D}$  is a matrix with  $N \cdot K$  rows and  $M \cdot K$  columns. The matrix  $\mathbf{D}$  can be obtained by permuting the rows and columns of the following block diagonal matrix:

$$\begin{bmatrix} \mathbf{B}_1 & & \\ & \ddots & \\ & & \mathbf{B}_K \end{bmatrix}, \quad (18)$$

where  $\{\mathbf{B}_k; k = 1, 2, \dots, K\}$  are defined in (3). Combining (10), (12) and (17), the posterior distribution can be calculated as [22]:

$$pdf(\boldsymbol{\alpha} | \mathbf{y}) \sim N(\boldsymbol{\mu}_p, \Sigma_p), \quad (19)$$

where

$$\Sigma_p = \mathbf{A} - \mathbf{AD}^T (\sigma_0^2 \mathbf{I} + \mathbf{DAD}^T)^{-1} \mathbf{DA} \quad (20)$$

$$\boldsymbol{\mu}_p = \sigma_0^{-2} \Sigma_p \mathbf{D}^T \mathbf{y}. \quad (21)$$

Since the Gaussian distribution reaches its maximum at the mean value, the MAP solution of the model coefficients  $\boldsymbol{\alpha}$  is:

$$\boldsymbol{\alpha} = \boldsymbol{\mu}_p = \sigma_0^{-2} \Sigma_p \mathbf{D}^T \mathbf{y}. \quad (22)$$

While the proposed Bayesian inference has been illustrated, the hyper-parameters in the prior distribution (10) and the likelihood function (17) must be appropriately determined in order to solve the model coefficients  $\boldsymbol{\alpha}$ . In the next section, we will further discuss a statistical inference to find the optimal hyper-parameter values.

### 3.3 Hyper-Parameter Inference

The hyper-parameters of our proposed Bayesian inference are defined in Section 3.1 and Section 3.2:  $\Omega = \{\lambda_1, \lambda_2, \dots, \lambda_M, \mathbf{R}, \sigma_0\}$ . To determine these hyper-parameters, one possible approach is to apply cross-validation [22] which has been successfully used for various applications [16]-[18]. However, the cross-validation approach is not applicable to our Bayesian inference in this paper. It is well-known that the computational complexity of cross-validation exponentially grows with the number of hyper-parameters, and thus is only suitable to handle small-size problems. In this paper, our Bayesian inference involves a large number of hyper-parameters, since the number of basis functions (i.e.,  $M$ ) can easily reach several thousand due to the large number of device-level process parameters at an advanced technology node. Hence, an alternative approach must be developed to efficiently determine these hyper-parameters in  $\Omega$ .

Statistically, the hyper-parameters can be determined by maximizing the conditional probability of observing the data set, i.e., the marginal likelihood:

$$l(\Omega) = \text{pdf}(\mathbf{y} | \Omega) = \int \text{pdf}(\mathbf{y} | \mathbf{a}, \Omega) \text{pdf}(\mathbf{a} | \Omega) \cdot d\mathbf{a}. \quad (23)$$

The integration in (23) indicates that the effect of model coefficients  $\mathbf{a}$  is averaged out in the marginal likelihood. Given that the two terms in the integration depend on different sets of hyper-parameters in  $\Omega$ , we can rewrite (23) as:

$$l(\Omega) = \int \text{pdf}(\mathbf{y} | \mathbf{a}, \sigma_0) \text{pdf}(\mathbf{a} | \lambda_1, \dots, \lambda_M, \mathbf{R}) \cdot d\mathbf{a}. \quad (24)$$

By combining (10) and (17) and taking the negative logarithm for (24), we have:

$$L(\Omega) = \mathbf{y}^T (\sigma_0^2 \mathbf{I} + \mathbf{D}\mathbf{A}\mathbf{D}^T)^{-1} \mathbf{y} + \log |\sigma_0^2 \mathbf{I} + \mathbf{D}\mathbf{A}\mathbf{D}^T|. \quad (25)$$

The hyper-parameters can be found by minimizing the cost function in (25). However, minimizing (25) is not trivial, because the cost function is not convex. In this paper, we adopt an expectation-maximization (EM) algorithm to address this challenge. In particular, instead of directly minimizing (25), the EM algorithm approaches a local optimum of (25) by iteratively performing two operations [22]-[23], known as the expectation step and the maximization step respectively.

In the expectation step, we first calculate the posterior distribution  $\text{pdf}(\mathbf{a} | \mathbf{y}, \Omega^{(\text{old})})$  according to (19)-(21) based on  $\Omega^{(\text{old})}$ , where  $\Omega^{(\text{old})}$  denotes the hyper-parameters calculated in the last iteration. Next, we define the following quantity [22]:

$$\mathcal{Q}(\Omega | \Omega^{(\text{old})}) = \mathbb{E}_{\mathbf{a} | \mathbf{y}, \Omega^{(\text{old})}} [\log p(\mathbf{y}, \mathbf{a} | \Omega^{(\text{old})})], \quad (26)$$

where the operator  $\mathbb{E}$  represents the expectation of the log probability function with respect to the posterior distribution  $\text{pdf}(\mathbf{a} | \mathbf{y}, \Omega^{(\text{old})})$ . Note that the expected value in (26) depends on the mean vector  $\mathbf{\mu}_p$  and the covariance matrix  $\Sigma_p$  of the posterior distribution in (19).

In the maximization step, we try to find  $\Omega$  such that:

$$\Omega = \arg \max_{\Omega} \mathcal{Q}(\Omega | \Omega^{(\text{old})}). \quad (27)$$

The optimal solution of  $\Omega$  corresponds to:

$$\frac{\partial}{\partial \Omega} \mathcal{Q}(\Omega | \Omega^{(\text{old})}) = 0. \quad (28)$$

Combining (26) and (28), we can find the optimal solution  $\Omega$  of (27) as a function of  $\Omega^{(\text{old})}$ , resulting in the following rule for updating  $\Omega$  [23]:

$$\lambda_m \leftarrow \frac{\text{Tr} \left[ \mathbf{R}^{-1} \left( \Sigma_p^m + \mathbf{\mu}_p^m (\mathbf{\mu}_p^m)^T \right) \right]}{K} \quad (m=1,2,\dots,M) \quad (29)$$

$$\mathbf{R} \leftarrow \frac{1}{M} \sum_{m=1}^M \frac{\Sigma_p^m + \mathbf{\mu}_p^m (\mathbf{\mu}_p^m)^T}{\lambda_m} \quad (m=1,2,\dots,M) \quad (30)$$

$$\sigma_0 \leftarrow \sqrt{\frac{\|\mathbf{y} - \mathbf{D}\mathbf{\mu}_p\|_2^2 + \text{Tr}(\Sigma_p \mathbf{D}^T \mathbf{D})}{N}} \quad (m=1,2,\dots,M), \quad (31)$$

where  $\mathbf{\mu}_p^m$  corresponds to the  $m$ -th sub-vector in  $\mathbf{\mu}_p$  with size of  $K$ ,  $\Sigma_p^m$  corresponds to the  $m$ -th diagonal block in  $\Sigma_p$  with size of  $K \times K$ . The expectation step (19)-(21) and the maximization step (29)-(31) are performed iteratively until convergence is reached.

Given that the EM algorithm only converges to a local optimum, the initial guess of  $\Omega$  must be carefully chosen. Here we propose a modified S-OMP algorithm to achieve this goal. The key idea of the proposed algorithm is to (i) reduce the hyper-parameter space by posing additional constraints on  $\mathbf{R}$ , (ii) apply a greedy algorithm for basis function selection, and (iii) apply cross-validation to solve this simplified problem within the reduced hyper-parameter space.

Following this idea, we first consider the parameterized

covariance matrix  $\mathbf{R}$  defined as:

$$\mathbf{R} = \begin{bmatrix} 1 & r_0 & \cdots & r_0^{K-1} \\ r_0 & 1 & & \\ & \ddots & \ddots & r_0 \\ r_0^{K-1} & r_0 & 1 & \end{bmatrix}, \quad (32)$$

where  $r_0 < 1$  is a non-negative hyper-parameter. The matrix in (32) has several implications. First, the correlation becomes small when two states (i.e., their indexes) are substantially different. Second, the same decaying rate  $r_0$  is used for all the states. This assumption often provides a good approximation, even though it is not highly accurate. However, since our goal here is to find a good initial guess of  $\Omega$ , a reasonably accurate approximation should suffice. As the covariance matrix  $\mathbf{R}$  depends on a single parameter  $r_0$  in (32), we can now apply cross-validation to estimate its value with low computational cost. On the other hand, a greedy basis selection algorithm based on S-OMP [19] is used to infer the sparsity information that is encoded by the hyper-parameters  $\{\lambda_1, \lambda_2, \dots, \lambda_M\}$ . In this way, we do not have to directly estimate  $\{\lambda_1, \lambda_2, \dots, \lambda_M\}$ , which is computationally expensive.

At this point, the hyper-parameter space has been reduced to three variables:  $r_0$ ,  $\sigma_0$  and  $\eta$ , where  $\eta$  denotes the number of selected basis functions. These three hyper-parameters can be efficiently determined using cross-validation [22]. Namely, we start from a candidate set  $\{(r_0^{(q)}, \sigma_0^{(q)}, \eta^{(q)}); q = 1, 2, \dots, \omega\}$ , where  $\omega$  denotes the size of the set. At the  $q$ -th iteration step, we choose the hyper-parameter values  $(r_0^{(q)}, \sigma_0^{(q)}, \eta^{(q)})$ , estimate the model coefficients, and evaluate the modeling error. To solve the model coefficients with given  $(r_0^{(q)}, \sigma_0^{(q)}, \eta^{(q)})$ , we adopt a modified S-OMP algorithm to iteratively identify the important basis functions. Similar to S-OMP [19], a single most important basis function is greedily chosen at each iteration step by maximizing the correlation between the selected basis function and the modeling residual over all states. The model coefficients are then calculated from the Bayesian inference in (10), (20)-(22) and (32) based on  $r_0^{(q)}$ ,  $\sigma_0^{(q)}$  and the currently selected basis functions. Different from the traditional S-OMP method, we take into account the correlation of coefficient magnitude when solving the unknown model coefficients. The aforementioned iteration continues until the number of chosen basis functions reaches  $\eta^{(q)}$ . The modeling error can now be evaluated by cross-validation for the hyper-parameters  $(r_0^{(q)}, \sigma_0^{(q)}, \eta^{(q)})$ . Eventually, the optimal values of  $(r_0, \sigma_0, \eta)$  are chosen to minimize the modeling error, and they are used to initialize the hyper-parameters  $\Omega$  for the iterative EM algorithm.

### 3.4 Summary

Algorithm 1 summarizes the major steps of the correlated Bayesian model fusion (C-BMF) algorithm. It consists of two main components: (i) a modified S-OMP algorithm from Step 1 to Step 17 to find the initial guess of the hyper-parameters  $\Omega$ , and (ii) the EM algorithm from Step 18 to Step 20 to iteratively solve the optimal values of  $\Omega$  where the model coefficients are updated at each iteration step.

#### Algorithm 1: Correlated Bayesian Model Fusion Algorithm

1. Start from a candidate set of hyper-parameters  $\{(r_0^{(q)}, \sigma_0^{(q)}, \eta^{(q)}); q = 1, 2, \dots, \omega\}$ . Partition the sampling points into  $C$  groups with the same size for cross-validation.
2. Set the candidate index  $q = 1$ .
3. Set the cross-validation index  $c = 1$ .
4. Assign the  $c$ -th group as the testing set and all other groups as

the training set.

5. Initialize the residual  $Res_k = \mathbf{y}_k$ , the basis vector set  $\Theta = \{\}$ , and the iteration index  $p = 1$ .
6. Calculate the inner product values  $\{\zeta_{k,m}; k = 1, 2, \dots, K; m = 1, 2, \dots, M\}$  between  $Res_k$  and all basis vectors  $\mathbf{b}_{k,m}$  for each state, where  $\mathbf{b}_{k,m}$  represents the  $m$ -th column of  $\mathbf{B}_k$  in (3).
7. Select the index  $s$  based on the following criterion:

$$s = \arg \max_m \sum_{k=1}^K |\zeta_{k,m}|. \quad (33)$$

8. Update  $\Theta = \Theta \cup s$ .
  9. By considering the basis functions in  $\Theta$  and  $\lambda_m = 1$  where  $m \in \Theta$ , solve the model coefficients  $\{\alpha_{k,m}; k = 1, 2, \dots, K; m \in \Theta\}$  using (10), (20)-(22) with  $r_0^{(q)}$ ,  $\sigma_0^{(q)}$  and  $\mathbf{R}$  defined in (32).
  10. Calculate the residual:
- $$Res_k = \mathbf{y}_k - \sum_{m \in \Theta} \alpha_{k,m} \cdot \mathbf{b}_{k,m} \quad (k=1,2,\dots,K). \quad (34)$$
11. If  $p < \eta^{(q)}$ ,  $p = p + 1$  and go to Step 6.
  12. Calculate the modeling error  $e_{q,c}$  using the testing set.
  13. If  $c < C$ ,  $c = c + 1$  and go to Step 4.
  14. Calculate  $e_q = (e_{q,1} + e_{q,2} + \dots + e_{q,C})/C$ .
  15. If  $q < \omega$ ,  $q = q + 1$  and go to Step 3.
  16. Find the optimal  $r_0$ ,  $\sigma_0$ , and  $\eta$  among the set  $\{(r_0^{(q)}, \sigma_0^{(q)}, \eta^{(q)}); q = 1, 2, \dots, \omega\}$  such that the modeling error  $e_q$  is minimized.
  17. Initialize the matrix  $\mathbf{R}$  using (32),  $\lambda_m = 1$  where  $m \in \Theta$ , and  $\lambda_m = 10^{-5}$  where  $m \notin \Theta$ .
  18. Calculate  $\mu_p$  and  $\Sigma_p$  based on the expectation step in (19)-(21).
  19. Update  $\{\lambda_m; m = 1, 2, \dots, M\}$ ,  $\mathbf{R}$  and  $\sigma_0$  based on the maximization step using (29)-(31).
  20. If convergence is not reached, go to Step 18. Otherwise, stop iteration and calculate the model coefficients  $\alpha$  by using (22).

## 4. NUMERICAL EXAMPLES

In this section, two circuit examples designed in a commercial 32nm SOI CMOS process are used to demonstrate the efficacy of the proposed C-BMF algorithm. Our objective is to build the performance models for tunable circuits. For testing and comparison purposes, two different modeling algorithms are implemented: (i) S-OMP [19], and (ii) C-BMF. Here, S-OMP is chosen for comparison, as it is one of the state-of-the-art techniques in the literature.

In each example, a set of random samples are generated by transistor-level Monte Carlo simulations. The data set is partitioned into two groups, referred to as the training and testing sets respectively. The training set is used for coefficient fitting, while the testing set is used for model validation. All numerical experiments are run on a 2.53GHz Linux server with 64GB memory.

### 4.1 Low Noise Amplifier

Figure 2(a) shows the simplified circuit schematic of a tunable 2.4GHz low noise amplifier (LNA) designed in a commercial 32nm SOI CMOS process. In this example, there are totally 1264 independent random variables to model the device-level process variations, including both the inter-die variations and the random mismatches. The LNA is designed with 32 different knob configurations (i.e., states) controlled by a tunable current source. Our goal is to model three performance metrics, noise figure (NF), voltage gain (VG) and third-order intercept point (IIP3), as linear functions of all random variables for 32 states. The modeling error is estimated by using a testing set with 50 samples per state.

Figure 2(b)-(d) show the performance modeling error for NF,

VG and IIP3 respectively. Two important observations can be made here. First, for both S-OMP and C-BMF, the modeling error decreases when the number of samples increases. Second, with the same number of samples, C-BMF achieves significantly higher accuracy than S-OMP, because C-BMF takes into account the additional correlation information of coefficient magnitude.

Table 1 further compares the performance modeling error and cost for S-OMP and C-BMF. The overall modeling cost consists of two major components: (i) the circuit simulation cost for collecting training samples, and (ii) the model fitting cost for solving all model coefficients. As shown in Table 1, the overall modeling cost is dominated by the simulation cost in this example. C-BMF achieves more than 2× cost reduction over S-OMP without surrendering any accuracy.

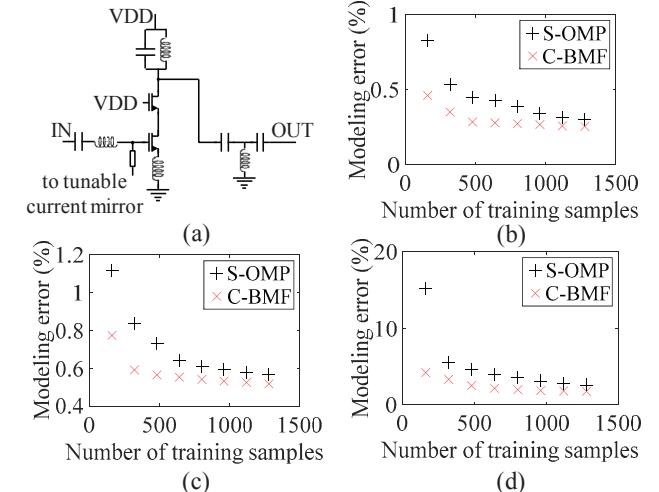


Figure 2. A tunable 2.4GHz LNA is used as an example for performance modeling: (a) the simplified circuit schematic, and (b)-(d) the performance modeling error for NF, VG and IIP3 respectively.

Table 1. Performance modeling error and cost for LNA

	S-OMP	C-BMF
Number of training samples	1120	480
Modeling error for NF	0.316%	0.285%
Modeling error for VG	0.577%	0.566%
Modeling error for IIP3	2.738%	2.497%
Simulation cost (Hours)	2.72	1.16
Fitting cost (Sec.)	1.32	316.13
Overall modeling cost (Hours)	2.72	1.25

### 4.2 Down-conversion Mixer

Shown in Figure 3(a) is the simplified circuit schematic of a tunable 2.4GHz down-conversion mixer designed in a commercial 32nm SOI CMOS process. In this example, there are totally 1303 independent random variables to model the device-level process variations, including both the inter-die variations and the random mismatches. The mixer is designed with 32 different states controlled by two tunable load resistors. The performances of interest include NF, VG, and input referred 1dB compression point (I1dBCP). The modeling error is estimated by using a test set with 50 samples per state.

Figure 3(b)-(d) show the performance modeling error for NF, VG and I1dBCP respectively. Note that C-BMF requires substantially less training samples than S-OMP to achieve the

same modeling accuracy. In this example, C-BMF achieves more than 2x cost reduction over S-OMP, as shown in Table 2.

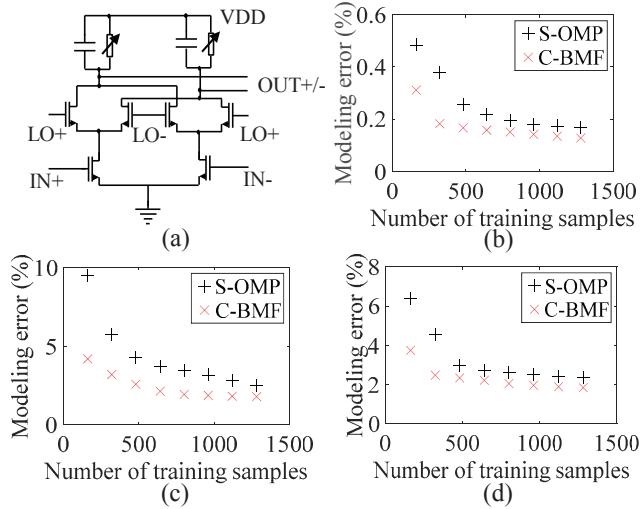


Figure 3. A tunable 2.4GHz down-conversion mixer is used as an example for performance modeling: (a) the simplified circuit schematic, (b)-(d) the modeling error for NF, VG and I1dB CP respectively.

Table 2. Performance modeling error and cost for mixer

	S-OMP	C-BMF
Number of samples	1120	480
Modeling error for NF	0.173%	0.166%
Modeling error for VG	2.758%	2.569%
Modeling error for I1dB CP	2.401%	2.340%
Simulation cost (Hours)	17.20	7.37
Fitting cost (Sec.)	1.39	407.10
Overall modeling cost (Hours)	17.20	7.48

## 5. CONCLUSIONS

In this paper, a novel C-BMF algorithm is proposed for efficient performance modeling of tunable analog/RF circuits. C-BMF encodes the correlation information for both model template and coefficient magnitude among different states by a prior distribution. Next, the prior distribution is combined with very few simulation samples to accurately solve the model coefficients. An efficient and robust hyper-parameter inference is proposed to determine the large number of hyper-parameters associated with our proposed Bayesian inference. As is demonstrated by two circuit examples designed in a commercial 32nm SOI CMOS process, the proposed C-BMF method achieves more than 2x cost reduction compared to the state-of-the-art modeling technique. Finally, it is worth mentioning that C-BMF assumes a unified correlation model across all states. If the states are mutually different, such an assumption will no longer hold. In this case, a clustering algorithm is needed to group similar states into clusters before applying the proposed C-BMF algorithm.

## 6. ACKNOWLEDGMENT

This work is sponsored in part by the DARPA HEALICS program under Air Force Research Laboratory (AFRL) contract FA8650-09-C-7924 and the National Science Foundation under contract CCF-1316363. The views expressed are those of the author and do not reflect the official policy or position of the

Department of Defense or the U.S. Government. The authors would thank B. Sadhu, J.-O. Plouchart and A. Valdes-Garcia for valuable discussions and D. Friedman for management support.

## 7. REFERENCES

- [1] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.
- [2] X. Li *et al.*, *Statistical Performance Modeling and Optimization*, Now Publishers, 2007.
- [3] X. Li *et al.*, “Robust analog/RF circuit design with projection-based performance modeling,” *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 2-15, Jan. 2007.
- [4] G. Debryser *et al.*, “Efficient analog circuit synthesis with simultaneous yield and robustness optimization,” *IEEE ICCAD*, pp. 308-311, 1998.
- [5] M. Erturk *et al.*, “Statistical variations in VCO phase noise due to upconverted MOSFET 1/f noise,” *IEEE RFIC*, pp. 255-258, 2008.
- [6] S. Sun *et al.*, “Indirect performance sensing for on-chip self-healing of analog and RF circuits,” *IEEE Trans. on CAS-I*, vol. 61, no. 8, pp. 2243-2252, Jul. 2014.
- [7] C. Maxey *et al.*, “Mixed-signal SoCs with in situ self-healing circuitry,” *IEEE D&T*, vol. 29, no. 6, pp. 27-39, Dec. 2012.
- [8] S. Sen, “Channel-adaptive zero-margin & process-adaptive self-healing communication circuits/systems,” *IEEE ICCAD*, pp. 80-85, 2014.
- [9] D. Banerjee *et al.*, “Real-time use-aware adaptive MIMO RF receiver systems for energy efficiency under BER constraints,” *IEEE DAC*, pp. 1-7, 2013.
- [10] M. Andraud *et al.*, “One-shot calibration of RF circuits based on non-intrusive sensors,” *IEEE DAC*, pp. 1-6, 2014.
- [11] J.-O. Plouchart *et al.*, “Adaptive circuit design methodology and test applied to millimeter-wave circuits,” *IEEE D&T*, vol. 31, no. 6, pp. 8-18, Jul. 2014.
- [12] X. Li *et al.*, “Asymptotic probability extraction for nonnormal performance distributions,” *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 16-37, Jan. 2007.
- [13] F. Gong *et al.*, “Variability-aware parametric yield estimation for analog/mixed-signal circuits: concepts, algorithms, and challenges,” *IEEE Design & Test of Comp.*, vol. 31, no. 4, pp. 6-15, Jan. 2014.
- [14] H. Zhang *et al.*, “Efficient design-specific worst-case corner extraction for integrated circuits,” *IEEE DAC*, pp. 386-389, 2009.
- [15] Y. Wang *et al.*, “Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization,” *IEEE DAC*, pp. 1-6, 2014.
- [16] X. Li, “Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/RF circuits,” *IEEE Trans. on CAD*, vol. 29, no. 11, pp. 1661-1668, Nov. 2010.
- [17] T. McConaghy, “High-dimensional statistical modeling and analysis of custom integrated circuits,” *IEEE CICC*, pp. 1-8, 2011.
- [18] F. Wang *et al.*, “Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data,” *IEEE DAC*, pp. 1-6, 2013.
- [19] W. Zhang *et al.*, “Efficient spatial pattern analysis for variation decomposition via robust sparse regression,” *IEEE Trans. on CAD*, vol. 32, no. 7, pp. 1072-1085, Jul. 2013.
- [20] W. Zhang *et al.*, “Toward efficient large-scale performance modeling of integrated circuits via multi-mode/multi-corner sparse regression,” *IEEE DAC*, pp. 897-902, 2010.
- [21] X. Liu *et al.*, “A statistical methodology for noise sensor placement and full-chip voltage map generation,” *IEEE DAC*, pp. 1-6, 2015.
- [22] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [23] Z. Zhang *et al.*, “Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning,” *IEEE JSTSP*, vol. 5, no. 5, pp. 912-926, Sep. 2011.