Indirect Performance Sensing for On-Chip Self-Healing of Analog and RF Circuits

Shupeng Sun, *Student Member, IEEE*, Fa Wang, *Student Member, IEEE*, Soner Yaldiz, *Member, IEEE*, Xin Li, *Senior Member, IEEE*, Lawrence Pileggi, *Fellow, IEEE*, Arun Natarajan, Mark Ferriss, Jean-Olivier Plouchart, *Senior Member, IEEE*, Bodhisatwa Sadhu, *Member, IEEE*, Ben Parker, Alberto Valdes-Garcia, *Senior Member, IEEE*, Mihai A. T. Sanduleanu, *Member, IEEE*, Jose Tierno, and Daniel Friedman, *Member, IEEE*

***Abstract*—The advent of the nanoscale integrated circuit (IC) technology makes high performance analog and RF circuits increasingly susceptible to large-scale process variations. On-chip self-healing has been proposed as a promising remedy to address the variability issue. The key idea of on-chip self-healing is to adaptively adjust a set of on-chip tuning knobs (e.g., bias voltage) in order to satisfy all performance specifications. One major challenge with on-chip self-healing is to efficiently implement on-chip sensors to accurately measure various analog and RF performance metrics. In this paper, we propose a novel indirect performance sensing technique to facilitate inexpensive-yet-accurate on-chip performance measurement. Towards this goal, several advanced statistical algorithms (i.e., sparse regression and Bayesian inference) are adopted from the statistics community. A 25 GHz differential Colpitts voltage-controlled oscillator (VCO) designed in a 32 nm CMOS SOI process is used to validate the proposed indirect performance sensing and self-healing methodology. Our silicon measurement results demonstrate that the parametric yield of the VCO is significantly improved for a wafer after the proposed self-healing is applied.**

***Index Terms*—Indirect performance sensing, integrated circuit, parametric yield, process variation, self-healing.**

Manuscript received December 23, 2013; revised March 14, 2014; accepted April 01, 2014. Date of publication July 16, 2014; date of current version July 24, 2014. This work was supported in part by the DARPA HEALICS (Self-Healing Mixed-Signal Integrated Circuits) program under Air Force Research Laboratory (AFRL) contract FA8650-09-C-7924. 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. This paper was presented in part at the Custom Integrated Circuits Conference in 2011 [1] and 2013 [2]. This paper was recommended by Associate Editor J. W. M. Rogers.

S. Sun, F. Wang, X. Li, and L. Pileggi are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: shupengs@ece.cmu.edu; fwang1@andrew.cmu.edu; xinli@ece.cmu.edu; pileggi@cmu.edu).

S. Yaldiz was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Intel Corporation, Hillsboro, OR 97124 USA (e-mail: syaldiz@gmail.com).

A. Natarajan was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA. He is now with Oregon State University, Corvallis, OR 97331 USA (e-mail: nataraja@eecs.oregonstate.edu).

M. Ferriss, J.-O. Plouchart, B. Sadhu, B. Parker, A. Valdes-Garcia, and D. Friedman are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: mferriss@us.ibm.com; plouchar@us.ibm.com; sadhu@us.ibm.com; nj@us.ibm.com; avaldes@us.ibm.com; dfriedmn@us.ibm.com).

M. A. T. Sanduleanu was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA. He is now with Masdar Institute of Science and Technology, Masdar City, Abu Dhabi, UAE (e-mail: msanduleanu@masdar.ac.ae).

J. Tierno was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA. He is now with Apple Inc., CA 95015 USA (e-mail: jose.tierno@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2014.2333311

I. Introduction

With the aggressive scaling of nanoscale integrated circuit (IC) technology, large-scale process variation becomes a critical issue for today's analog and RF ICs [3]–[7]. As the traditional overdesign technique becomes impractical, on-chip self-healing has emerged as a promising methodology to address the variability issue [9]–[15]. The key idea of self-healing is to actively monitor the post-manufacturing circuit performance metrics and then adaptively adjust a number of tuning knobs (e.g., bias voltage) in order to meet the given performance specifications.

To practically implement on-chip self-healing, a large number of performance metrics must be measured accurately and inexpensively by on-chip sensors. Such a measurement task, however, is not trivial, because many analog and RF performance metrics (e.g., phase noise) cannot be easily measured by on-chip sensors. For this reason, alternate test methodology, also called indirect performance sensing, has recently attracted great attention [8], [9], [11]–[13], [15], where the *performance of interest* (PoI) is not directly measured by an on-chip sensor. Instead, it is accurately predicted from a set of other performance metrics, referred to as the *performances of measurement* (PoMs) that are highly correlated with PoI and are easy to measure. Towards this goal, indirect sensor modeling is a critical task where the objective is to build a mathematical model to capture the correlation between PoI and PoMs so that PoI can be accurately predicted from PoMs. To achieve this goal, PoMs and PoI are first measured from several training chips, and then indirect sensor models are constructed off-line based on these measurement data. Such indirect sensor models are eventually stored in an on-chip microcontroller for self-healing.

To describe an indirect senor, its model coefficients are stored in an on-chip microcontroller as fixed-point values. A complex model that is composed of many model terms would consume massive hardware resources, since a large number of model coefficients must be stored. Furthermore, during on-chip self-healing, an indirect sensor model is repeatedly evaluated to predict the corresponding PoI based on different PoMs and, therefore, a compact model could dramatically reduce the computational cost. Here, the computational cost accounts for on-chip multiplication and addition, and multiplication dominates the overall computational cost. For these reasons, an indirect sensor model should be compact in order to minimize the cost of on-chip self-healing.

Such a modeling task, however, is nontrivial since there is a tradeoff between the model complexity and the model accuracy. In general, it is likely that an oversimplified model will induce

a large modeling error. Here, how to construct a compact indirect sensor model without sacrificing its modeling accuracy remains an open question. In addition, these indirect sensor models must be repeatedly calibrated to accommodate the process shift associated with manufacturing lines. Such a model calibration issue has not been extensively studied yet. Hence, there is a strong need to develop a new methodology to facilitate efficient model calibration with low cost (i.e., requiring few additional measurement data). As such, the overhead of indirect performance sensing and, eventually, the overhead of analog and RF self-healing can be minimized.

To address the aforementioned issues, a novel indirect performance sensing approach is proposed in this paper. The proposed method consists of two major steps: i) pre-silicon indirect sensor modeling, and ii) post-silicon indirect sensor calibration. In the first step, a compact indirect sensor model between PoMs and PoI is constructed based on pre-silicon simulation data by using sparse regression (SR) [16], [17]. SR starts with a complicated model template (e.g., a high-order polynomial) that can accurately capture the correlation between PoMs and PoI. L_1-norm regularization is then applied, resulting in a convex optimization problem which can be efficiently solved to determine the most important model terms in the template without sacrificing any modeling accuracy. Other model coefficients corresponding to the unimportant terms are simply set to zero, and are ignored in the final indirect sensor model. Intuitively, the unimportant model terms have negligible contribution for accurately predicting the value of PoI and, hence, can be discarded to minimize the self-healing cost.

Furthermore, in the second step, an indirect sensor model is repeatedly calibrated based on post-silicon measurement data. To perform efficient model calibration with low cost, a novel Bayesian model fusion (BMF) technique is proposed. The key idea of BMF is to combine the old (i.e., before process shift) indirect sensor model with very few new (i.e., after process shift) measurement data to generate a new model that is aligned with the new process condition. Mathematically, the old model is encoded as prior knowledge, and a Bayesian inference is derived to optimally fit the new model by maximum-a-posteriori (MAP) estimation.

Finally, an on-chip self-healing flow is developed where the indirect sensor models are extracted by the proposed technique. Measurement data of a 25 GHz differential Colpitts VCO designed in a 32 nm CMOS SOI process are used to perform an off-chip data analysis to validate the aforementioned on-chip self-healing flow. Our silicon measurement results demonstrate that the parametric yield of the VCO is significantly improved for a wafer after self-healing is applied.

The remainder of this paper is organized as follows. In Section II, we will present an overview of our proposed indirect sensing methodology. The mathematical details for pre-silicon indirect sensor modeling and post-silicon indirect sensor calibration will be discussed in Section III and Section IV respectively. In Section V, an on-chip self-healing flow based on our proposed indirect sensing approach is described. A 25 GHz differential Colpitts VCO example designed in a 32 nm CMOS SOI process is used to validate our proposed on-chip self-healing flow in Section VI. Finally, we conclude in Section VII.

II. Indirect Performance Sensing

Without loss of generality, we denote PoI as f and PoMs as:

$$\mathbf{x} = [x_1 \; x_2 \; \cdots \; x_M]^T, \tag{1}$$

where M stands for the number of performance metrics belonging to PoMs. The objective of indirect performance sensing is to accurately predict the PoI f from the PoMs $\mathbf{x}$ that are highly correlated with f and can be easily measured by on-chip sensors.

Generating an indirect sensor model $f(\mathbf{x})$ consists of two major steps:

- **Pre-silicon indirect sensor modeling** aims to construct a compact model $f(\mathbf{x})$ that can accurately capture the correlation between the PoI f and the PoMs $\mathbf{x}$ based on pre-silicon simulation data.
- **Post-silicon indirect sensor calibration** aims to calibrate the indirect sensor model $f(\mathbf{x})$ based on post-silicon measurement data. Such model calibration must be repeatedly performed in order to accommodate the process shift associated with manufacturing lines.

We start with a generic and complicated model template (e.g., a high-order polynomial) to accurately capture the mapping between PoI and PoMs. The reason we choose a generic model is simply because we do not know the relation between PoI and PoMs in advance. Mathematically, we can write the model $f(\mathbf{x})$ as the linear combination of several basis functions:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \cdot b_k(\mathbf{x}), \tag{2}$$

where $\{b_k(\mathbf{x}); k = 1, 2, \ldots, K\}$ are the basis functions (e.g., linear and quadratic polynomials), $\{\alpha_k; k = 1, 2, \ldots, K\}$ are the model coefficients, and K is the total number of basis functions.

Such a complicated model, though accurate, consumes considerable hardware resources to implement, as all model coefficients must be stored in an on-chip microcontroller to perform on-chip self-healing. To reduce the overhead of on-chip self-healing, we aim to select a small set of basis functions during pre-silicon modeling without surrendering any accuracy. Such a basis function selection task, however, is extremely challenging due to the tradeoff between the model complexity and the modeling error. In general, an oversimplified model is likely to have a large modeling error. SR [16], [17] is applied to efficiently address the aforementioned basis function selection problem. More details about pre-silicon indirect sensor modeling via SR will be discussed in Section III.

Furthermore, at the post-silicon stage, the indirect sensor must be repeatedly calibrated to accommodate the process shift associated with manufacturing lines. Since post-silicon measurement is extremely expensive, sensor calibration must be accomplished with very few post-silicon measurement data to facilitate efficient generation of accurate indirect sensor models and, eventually, minimize the overhead of on-chip self-healing. To this end, a novel Bayesian model fusion (BMF) technique is proposed to keep the calibration cost affordable. The details about post-silicon indirect sensor calibration via BMF will be presented in Section IV.

III. Pre-Silicon Indirect Sensor Modeling via Sparse Regression

In this section, we aim to construct a compact indirect sensor model to accurately capture the relation between PoI and PoMs. Since the mapping from PoMs to PoI is not known in advance, we start with a generic and complicated model template consisting of a large number of basis functions (e.g., a high-order polynomial), as shown in (2). Our objective here is to automatically identify a small set of most important basis functions, and

then determine their corresponding model coefficients based on pre-silicon simulation data.

To start with, we first collect a number of pre-silicon simulation samples $\{(\mathbf{x}^{(n)}, f^{(n)}); n = 1, \ldots, N\}$, where $\mathbf{x}^{(n)}$ and $f^{(n)}$ denote the values of $\mathbf{x}$ and f for the n-th sampling point respectively, and N denotes the total number of sampling points. Based on these sampling points, a set of linear equations can be expressed as:

$$\mathbf{B}^T \cdot \boldsymbol{\alpha} = \mathbf{f}, \tag{3}$$

where

$$\mathbf{B} = \begin{bmatrix} b_1\left(\mathbf{x}^{(1)}\right) & b_1\left(\mathbf{x}^{(2)}\right) & \cdots & b_1\left(\mathbf{x}^{(N)}\right) \\ b_2\left(\mathbf{x}^{(1)}\right) & b_2\left(\mathbf{x}^{(2)}\right) & \cdots & b_2\left(\mathbf{x}^{(N)}\right) \\ \vdots & \vdots & \vdots & \vdots \\ b_K\left(\mathbf{x}^{(1)}\right) & b_K\left(\mathbf{x}^{(2)}\right) & \cdots & b_K\left(\mathbf{x}^{(N)}\right) \end{bmatrix} \tag{4}$$

$$\boldsymbol{\alpha} = [\,\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_K\,]^T \tag{5}$$

$$\mathbf{f} = [\,f^{(1)} \quad f^{(2)} \quad \cdots \quad f^{(N)}\,]^T. \tag{6}$$

One simple approach to solve the model coefficients $\boldsymbol{\alpha}$ is to apply the traditional ordinary least squares (OLS) fitting method [18]. OLS determines the model coefficients $\boldsymbol{\alpha}$ by solving the following optimization problem:

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2, \tag{7}$$

where $\| \bullet \|_2$ denotes the L_2-norm of a vector. Intuitively, OLS intends to find a solution $\boldsymbol{\alpha}$ that can minimize the mean squared modeling error.

As mentioned at the beginning of this section, we aim to identify a small set of important basis functions from a large number of possible candidates. All other unimportant basis functions will be discarded due to their negligible contribution for accurately predicting the value of PoI. From this point of view, all model coefficients associated with these unimportant basis functions should be set to zero. Hence, identifying the most important basis functions is equivalent to finding a sparse solution $\boldsymbol{\alpha}$ for the linear equation in (3). The OLS formulation in (7) poses no constraint on the sparsity of $\boldsymbol{\alpha}$. In other words, the unconstrained optimization in (7) used by OLS cannot fit our need of basis function selection. Realizing this limitation of OLS, SR, instead, solves the following L_1-norm regularization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\mathbf{B}^T \cdot \boldsymbol{\alpha} - \mathbf{f}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\alpha}\|_1 \le \lambda, \end{aligned} \tag{8}$$

where $\| \bullet \|_1$ denotes the L_1-norm of a vector, and $\lambda > 0$ is a user-defined parameter. The formulation in (8) is a convex optimization problem and can be solved both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global optimum) [20].

There are several important properties associated with the optimization problem in (8). First, unlike the conventional OLS that minimizes the mean squared error only, the formulation in (8) minimizes the mean squared error subject to an L_1-norm constraint posed on the model coefficients $\boldsymbol{\alpha}$. It, in turn, promotes a sparse solution of $\boldsymbol{\alpha}$ [16], [17] that is desired by our application of basis function selection for on-chip self-healing.

Second, the parameter λ in (8) provides a tradeoff between the sparsity of the solution $\boldsymbol{\alpha}$ and the modeling error. For instance, a large λ is likely to result in a small modeling error, but meanwhile it will increase the number of non-zeros in $\boldsymbol{\alpha}$. It is important to note that if the vector $\boldsymbol{\alpha}$ contains many non-zeros, a large number of model coefficients have to be stored in the on-chip microcontroller to predict the PoI and, hence, the cost of indirect performance sensing can be overly expensive. In practice, the value of λ must be appropriately set to accurately predict the PoI with a small set of basis functions. To find the optimal value of λ, we must accurately estimate the modeling error for different λ values. To avoid overfitting, we cannot simply measure the modeling error from the set of sampling data that is used to calculate the model coefficients. Instead, modeling error must be measured from an independent data set.

To determine the modeling error for a given λ value, we adopt the idea of Q-fold cross-validation from the statistics community [18]. Namely, we partition the entire data set $\{(\mathbf{x}^{(n)}, f^{(n)}); n = 1, 2, \ldots, N\}$ into Q groups. Modeling error is estimated from Q independent runs. In each run, one of the Q groups is used to estimate the modeling error and all other groups are used to calculate the model coefficients by solving (8). Note that the training data for coefficient estimation and the testing data for error estimation are not overlapped. Hence, overfitting can be easily detected. In addition, different groups should be selected for error estimation in different runs. As such, each run results in an error value $e_q (q = 1, 2, \ldots, Q)$ that is measured from a unique group of the data set. The final modeling error is computed as the average of $\{e_q; q = 1, 2, \ldots, Q\}$, i.e., $e = (e_1 + e_2 + \ldots + e_Q)/Q$. More details about cross-validation can be found in [18].

So far, we only consider how to reduce the number of basis functions (i.e., the number of non-zeros in $\boldsymbol{\alpha}$) in order to save on-chip self-healing cost. Actually, different basis functions may involve different number of multiplications, and the computational cost to calculate each basis function when evaluating the indirect sensor can be quite different. For instance, $\alpha_1 \cdot x$ requires only one multiplication, while $\alpha_2 \cdot x^3$ needs three multiplications. To further reduce the computational cost, we can assign different weights for different coefficients (e.g., a small weight for α_1 while a large weight for α_2) in the constraint of (8). Intuitively, a coefficient with a larger weight is more likely to be set to zero in a weighted L_1-norm regularization [20]. Because of the space limitation, the extended version of (8) to handle weighted $\boldsymbol{\alpha}$ is not mentioned here.

The aforementioned SR method can be efficiently applied to pre-silicon basis function selection and model coefficient estimation. However, the device models used for pre-silicon simulation are not perfectly accurate and may differ from the post-silicon measurement results. For this reason, there is a strong need to further calibrate the proposed indirect sensor models based on post-silicon measurement data, as will be discussed in the next section.

IV. Post-Silicon Indirect Sensor Calibration via Bayesian Model Fusion

The objective of post-silicon indirect sensor calibration is to further correct the modeling error posed by pre-silicon simulation and also accommodate the process shift associated with manufacturing lines. One straightforward approach for sensor calibration is to collect a large amount of post-silicon measurement data and then completely re-fit the indirect sensor model. Such a simple approach, however, can be practically unaffordable, since post-silicon testing is time-consuming and, hence, it is overly expensive to collect a large set of post-silicon measurement data.

To address this cost issue, we propose a novel statistical framework, referred to as *Bayesian model fusion* (BMF) [19], for efficient post-silicon sensor calibration. BMF relies on an important observation that even though the simulation and/or measurement

data collected at multiple stages (e.g., pre-silicon vs. post-silicon) are not exactly identical, they are expected to be strongly correlated. Hence, it is possible to *borrow* the data from an early stage (e.g., pre-silicon) for sensor calibration at a late stage (e.g., post-silicon). As such, only few post-silicon data should be measured at the late stage and, hence, the cost of sensor calibration is substantially reduced.

More specifically, our indirect sensor models are initially fitted by using the early-stage (e.g., pre-silicon) data. Next, the early-stage sensor model is encoded as our *prior* knowledge. Finally, the indirect sensor model is further calibrated by applying Bayesian inference with very few late-stage (e.g., post-silicon) measurement data. Here, by "fusing" the early-stage and late-stage sensor models through Bayesian inference, the amount of required measurement data (hence, the measurement cost) can be substantially reduced at the late stage.

To fully understand the proposed BMF method, let us consider two different models: the early-stage model $f_E(\mathbf{x})$ and the late-stage model $f_L(\mathbf{x})$:

$$f_E(\mathbf{x}) = \sum_{k=1}^{K} \alpha_{E,k} \cdot b_k(\mathbf{x}) + \varepsilon_E \tag{9}$$

$$f_L(\mathbf{x}) = \sum_{k=1}^{K} \alpha_{L,k} \cdot b_k(\mathbf{x}) + \varepsilon_L, \tag{10}$$

where $\{b_k(\mathbf{x}); k = 1, 2, \ldots, K\}$ are the basis functions selected by SR at the early stage, $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ and $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ contain the early-stage and late-stage model coefficients respectively, and ε_E and ε_L denote the modeling error associated with the early-stage and late-stage models respectively.

The early-stage model $f_E(\mathbf{x})$ in (9) is fitted by using the early-stage (e.g., pre-silicon) data. Hence, we assume that the early-stage model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ are already known, before fitting the late-stage model $f_L(\mathbf{x})$ in (10) based on the late-stage (e.g., post-silicon) measurement data. The objective of BMF is to accurately determine the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ by combining the early-stage model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ with very few late-stage measurement data.

Our proposed BMF method consists of two major steps: i) statistically extracting the prior knowledge from the early-stage model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ and encoding it as a prior distribution, and ii) optimally determining the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ by MAP estimation. In what follows, we will describe these two steps in detail.

A. Prior Knowledge Definition

Since the two models $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$ in (9), (10) both approximate the mathematical mapping from PoMs to PoI, we expect that the model coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$ and $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ are similar. On the other hand, $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$ cannot be exactly identical, since they represent the indirect sensor models at two different stages. To statistically encode the "common" information between $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$, we define a Gaussian distribution as our prior distribution for each late-stage model coefficient $\alpha_{L,k}$:

$$pdf(\alpha_{L,k}) = \frac{1}{\sqrt{2\pi} \cdot \rho \cdot |\alpha_{E,k}|} \cdot \exp\left[-\frac{(\alpha_{L,k} - \alpha_{E,k})^2}{2 \cdot \rho^2 \cdot \alpha_{E,k}^2}\right] \quad k = 1, 2, \cdots, K, \tag{11}$$

where $\alpha_{E,k}$ and $\rho^2 \cdot \alpha_{E,k}^2$ are the mean and variance of the Gaussian distribution respectively, and $\rho > 0$ is a parameter that can be determined by cross-validation [18].

The prior distribution in (11) has a two-fold meaning. First, the Gaussian distribution $pdf(\alpha_{L,k})$ is peaked at its mean value $\alpha_{E,k}$, implying that the early-stage model coefficient $\alpha_{E,k}$ and the late-stage model coefficient $\alpha_{L,k}$ are likely to be similar. In other words, since the Gaussian distribution $pdf(\alpha_{L,k})$ exponentially decays with $(\alpha_{L,k} - \alpha_{E,k})^2$, it is unlikely to observe a late-stage coefficient $\alpha_{L,k}$ that is extremely different from the early-stage coefficient $\alpha_{E,k}$. Second, the standard deviation of the prior distribution $pdf(\alpha_{L,k})$ is proportional to $|\alpha_{E,k}|$. It means that the absolute difference between the late-stage coefficient $\alpha_{L,k}$ and the early-stage coefficient $\alpha_{E,k}$ can be large (or small), if the magnitude of the early-stage coefficient $|\alpha_{E,k}|$ is large (or small). Restating in words, each late-stage coefficient $\alpha_{L,k}$ has been provided with a relatively equal opportunity to deviate from the corresponding early-stage coefficient $\alpha_{E,k}$.

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

$$pdf(\boldsymbol{\alpha}_L) = \prod_{k=1}^{K} pdf(\alpha_{L,k}), \tag{12}$$

where

$$\boldsymbol{\alpha}_L = [\alpha_{L,1}\ \alpha_{L,2}\ \cdots\ \alpha_{L,K}]^T \tag{13}$$

is a vector containing all late-stage coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$. Combining (11) and (12) yields:

$$pdf(\boldsymbol{\alpha}_L) = \frac{1}{(\sqrt{2\pi} \cdot \rho)^K \cdot \prod_{k=1}^{K} |\alpha_{E,k}|} \cdot \exp\left[-\frac{(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)}{2 \cdot \rho^2}\right], \tag{14}$$

where

$$\boldsymbol{\alpha}_E = [\alpha_{E,1}\ \alpha_{E,2}\ \cdots\ \alpha_{E,K}]^T \tag{15}$$

is a vector containing all early-stage coefficients $\{\alpha_{E,k}; k = 1, 2, \ldots, K\}$, and

$$\mathbf{A} = \mathrm{diag}\left(\frac{1}{\alpha_{E,1}^2}\ \frac{1}{\alpha_{E,2}^2}\ \cdots\ \frac{1}{\alpha_{E,K}^2}\right) \tag{16}$$

denotes a diagonal matrix. The independence assumption in (12) 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 measurement data, when the posterior distribution is calculated by MAP estimation in the next sub-section.

B. Maximum-A-Posteriori Estimation

Once the prior distribution is defined, we collect a few (i.e., N) late-stage measurement 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 $f_L(\mathbf{x})$ for the n-th data point respectively. These new measurement data can tell us additional information about the difference between early and late stages and, hence, help us to determine the late-stage coefficients $\boldsymbol{\alpha}_L$.

Based on Bayes' theorem [18], the uncertainties of the late-stage coefficients $\boldsymbol{\alpha}_L$ after knowing the data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, 2, \ldots, N\}$ can be mathematically described by the following posterior distribution:

$$pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L) \propto pdf(\boldsymbol{\alpha}_L) \cdot pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L), \tag{17}$$

where

$$\mathbf{f}_L = \left[f_L^{(1)}\ f_L^{(2)}\ \cdots\ f_L^{(N)}\right]^T \tag{18}$$

$$\mathbf{X} = \left[\mathbf{x}^{(1)}\ \mathbf{x}^{(2)}\ \cdots\ \mathbf{x}^{(N)}\right]^T. \tag{19}$$

In (17), the prior distribution $pdf(\boldsymbol{\alpha}_L)$ is defined by (14). The conditional distribution $pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L)$ is referred to as the likelihood function. It measures the probability of observing the new data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, 2, \ldots, N\}$.

To derive the likelihood function $pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L)$, we assume that the modeling error ε_L in (10) can be represented as a random variable with zero-mean Gaussian distribution:

$$pdf(\varepsilon_L) = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left[-\frac{\varepsilon_L^2}{2 \cdot \sigma_0^2}\right], \tag{20}$$

where the standard deviation σ_0 indicates the magnitude of the modeling error. Similar to the parameter ρ in (11), the value of σ_0 can be determined by cross-validation [18]. Since the modeling error associated with the n-th data point $(\mathbf{x}^{(n)}, f_L^{(n)})$ is simply one sampling point of the random variable ε_L that follows the Gaussian distribution in (20), the probability of observing the n-th data point $(\mathbf{x}^{(n)}, f_L^{(n)})$ is:

$$pdf\left[\mathbf{x}^{(n)}, f_L^{(n)}|\boldsymbol{\alpha}_L\right] = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left\{-\frac{1}{2 \cdot \sigma_0^2} \cdot \left[f_L^{(n)} - \sum_{k=1}^{K} \alpha_{L,k} \cdot b_k\left(\mathbf{x}^{(n)}\right)\right]^2\right\}. \tag{21}$$

Note that the likelihood function $pdf[\mathbf{x}^{(n)}, f_L^{(n)}|\boldsymbol{\alpha}_L]$ in (21) depends on the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$. Assuming that all data points $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, 2, \ldots, N\}$ are independently generated, we can write the likelihood function $pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L)$ as:

$$pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L) = \prod_{n=1}^{N} pdf\left[\mathbf{x}^{(n)}, f_L^{(n)}|\boldsymbol{\alpha}_L\right] = \frac{1}{(\sqrt{2\pi} \cdot \sigma_0)^N} \cdot \exp\left\{-\frac{1}{2 \cdot \sigma_0^2} \cdot \sum_{n=1}^{N}\left[f_L^{(n)} - \sum_{k=1}^{K} \alpha_{L,k} \cdot b_k\left(\mathbf{x}^{(n)}\right)\right]^2\right\}. \tag{22}$$

Eq. (22) can be re-written as:

$$pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L) = \frac{1}{(\sqrt{2\pi} \cdot \sigma_0)^N} \cdot \exp\left\{-\frac{(\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)^T \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)}{2 \cdot \sigma_0^2}\right\}, \tag{23}$$

where $\mathbf{B}$, $\boldsymbol{\alpha}_L$ and $\mathbf{f}_L$ are defined in (4), (13) and (18), respectively.

After the new data $\{(\mathbf{x}^{(n)}, f_L^{(n)}); n = 1, 2, \ldots, N\}$ are available, the late-stage coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ can be described by the probability density function $pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L)$ (i.e., the posterior distribution) in (17). Depending on the shape of the posterior distribution $pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L)$, the late-stage coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ do not take all possible values with equal probability. If the posterior distribution $pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L)$ reaches its maximum value at $\{\alpha_{L,k}^*; k = 1, 2, \ldots, K\}$, these values $\{\alpha_{L,k}^*; k = 1, 2, \ldots, K\}$ are the optimal estimation of the late-stage coefficients, since these coefficient values are most likely to occur. Such a method is referred to as the MAP estimation in the literature [18].

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

$$\underset{\boldsymbol{\alpha}_L}{\text{maximize}} \quad pdf(\boldsymbol{\alpha}_L|\mathbf{X}, \mathbf{f}_L). \tag{24}$$

Substituting (17) into (24) yields:

$$\underset{\boldsymbol{\alpha}_L}{\text{maximize}} \quad pdf(\boldsymbol{\alpha}_L) \cdot pdf(\mathbf{X}, \mathbf{f}_L|\boldsymbol{\alpha}_L). \tag{25}$$

Combining (14), (23) and (25), we have:

$$\underset{\boldsymbol{\alpha}_L}{\text{maximize}} \quad \exp\left[-\frac{(\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)}{2 \cdot \rho^2} - \frac{(\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)^T \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)}{2 \cdot \sigma_0^2}\right]. \tag{26}$$

Since the exponential function is monotonically increasing, Eq. (26) can be re-written as:

$$\underset{\boldsymbol{\alpha}_L}{\text{minimize}} \quad \eta \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E) + (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)^T \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L), \tag{27}$$

where

$$\eta = \frac{\sigma_0^2}{\rho^2}. \tag{28}$$

It is straightforward to prove that the cost function in (27) is convex [20]. Hence, its global optimum can be directly solved by applying the first-order optimality condition [20]:

$$\frac{\partial}{\partial \boldsymbol{\alpha}_L}\left[\eta \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E)^T \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E) + (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)^T \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L)\right] = 2 \cdot \eta \cdot \mathbf{A} \cdot (\boldsymbol{\alpha}_L - \boldsymbol{\alpha}_E) + 2 \cdot \mathbf{B} \cdot (\mathbf{B}^T \cdot \boldsymbol{\alpha}_L - \mathbf{f}_L) = \mathbf{0}. \tag{29}$$

Solving the linear equation in (29) results in the optimal value of $\boldsymbol{\alpha}_L$:

$$\boldsymbol{\alpha}_L = (\eta \cdot \mathbf{A} + \mathbf{B} \cdot \mathbf{B}^T)^{-1} \cdot (\eta \cdot \mathbf{A} \cdot \boldsymbol{\alpha}_E + \mathbf{B} \cdot \mathbf{f}_L). \tag{30}$$

Studying (30), we observe that only the value of η is required to find the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ and, hence, we only need to determine η, instead of the individual parameters σ_0 and ρ. In our work, the optimal value of η is determined by cross-validation [18].

Once the optimal η value is found, the late-stage model coefficients $\{\alpha_{L,k}; k = 1, 2, \ldots, K\}$ are calculated from (30), and then an updated indirect sensor model $f_L(\mathbf{x})$ in (10) is generated to match the late-stage measurement data. Such a calibrated indirect sensor model is eventually stored in an on-chip microcontroller to facilitate efficient on-chip self-healing, as will be discussed in detail in the next section.

Finally, it is important to mention that the post-silicon indirect sensor calibration is performed off-chip and, hence, no hardware overhead is introduced. To further reduce the indirect sensor calibration cost (i.e., with very few number of post-silicon measurement data), we can calibrate the indirect sensor if and only if the new measurement data are not consistent with the old indirect sensor model. In practice, such inconsistency can be detected by measuring a small number of dies from each wafer or lot to estimate the indirect sensing error. If the error is not sufficiently small, the indirect sensor model must be calibrated.

V. On-Chip Self-Healing Flow

In this section, we will further develop a practical on-chip self-healing flow based on our proposed indirect sensing approach. As mentioned earlier, the key idea of on-chip self-healing is to actively monitor the post-manufacturing circuit performance metrics and then adaptively adjust a number of tuning knobs (e.g., bias voltage) in order to meet the given performance specifications. In this work, we mathematically formulate the self-healing problem as a constrained optimization where one particular performance metric is minimized subject to a set of given performance constraints:

$$\begin{aligned} \underset{\mathbf{t}}{\text{minimize}} \quad & f(\mathbf{t}) \\ \text{subject to} \quad & g_p(\mathbf{t}) \geq s_p \quad (p = 1, 2, \cdots, P), \end{aligned} \tag{31}$$

where $\mathbf{t}$ denotes the set of tuning knobs, $f(\mathbf{t})$ denotes the performance metric that we aim to minimize, and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ denote the other P performance metrics with the given specifications $\{s_p; p = 1, 2, \ldots, P\}$. Take mixer as an example. We aim to minimize the mixer power while keeping its gain and 1 dB compression point larger than their specifications. In this case, $f(\mathbf{t})$ is the mixer power, $g_1(\mathbf{t})$ is the mixer gain, and $g_2(\mathbf{t})$ is the 1 dB compression point.

There are two important clarifications we need to make for the optimization formulation in (31). First, the formulation in (31) is set up for a circuit where one performance metric $f(\mathbf{t})$ should be minimized while constraining all other performance metrics $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ to their lower bounds $\{s_p; p = 1, 2, \ldots, P\}$. For a circuit where a performance metric $f(\mathbf{t})$ should be maximized, the objective function in (31) can be simply modified to $-f(\mathbf{t})$. Similarly, for a circuit where the performance metrics $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ should be constrained to their upper bounds $\{s_p; p = 1, 2, \ldots, P\}$, the constraints in (31) can be adjusted as $-g_p(\mathbf{t}) \geq -s_p (p = 1, 2, \ldots, P)$. Second, not all the performance metrics $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p \in 1, 2, \ldots, P\}$ in our self-healing circuit can be directly measured by on-chip sensors. For the performance metrics that cannot be easily measured by on-chip sensors, the proposed indirect performance sensing technique is applied to efficiently and accurately predict their values.

To find the optimal solution $\mathbf{t}^*$ in (31), we propose an on-chip self-healing flow shown in Fig. 1 where the indirect sensors are modeled and calibrated by our proposed SR and BMF techniques described in previous sections. The indirect sensor models are stored and evaluated by a microcontroller for on-chip self-healing. The search algorithm starts with an initial guess $\mathbf{t}_0$. We set $\mathbf{t} = \mathbf{t}_0$ and all performance metrics $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ are measured either directly or indirectly. Here, we use the symbol PMs to represent the performance metrics that are directly measured by on-chip sensors, and the symbol PoIs to represent the performance metrics that are estimated by the proposed indirect sensors. In particular, to estimate the PoIs, the corresponding PoMs are first measured by on-chip sensors. Next, the indirect sensor models stored in the on-chip microcontroller are evaluated to predict the PoIs, as shown in Fig. 1. Based on the performance values $\{f(\mathbf{t}), g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$, $\mathbf{t}$ is updated and the aforementioned process is repeated until the optimal solution $\mathbf{t}^*$ is found. Algorithm 1 summarizes the details of such an optimization flow with indirect performance sensing for on-chip self-healing. Once the optimal solution $\mathbf{t}^*$ is found, tuning knobs are adjusted to the values of $\mathbf{t}^*$ and the self-healing process is complete.

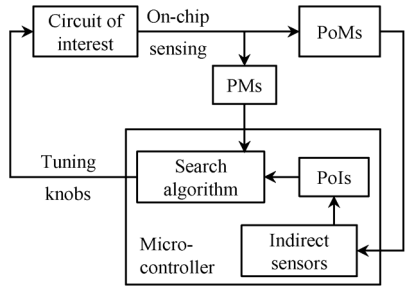


Fig. 1. A simplified block diagram describes the on-chip self-healing flow.

Algorithm 1: On-chip Self-healing Flow

1. Start with the constrained optimization problem in (31) and an initial guess $\mathbf{t}_0$.
2. Set $\mathbf{t} = \mathbf{t}_0$.
3. Measure $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ either directly or indirectly.
4. Based on the performance values $\{f(\mathbf{t}), g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$, update $\mathbf{t}$.
5. If $\mathbf{t}$ is the optimal solution, stop iteration. Otherwise, go to Step 3.

For different circuits of interest with different performance metrics and tuning knobs, the search strategy of updating $\mathbf{t}$ in Step 4 of Algorithm 1 could be substantially different. For instance, if there is only a small number of (e.g., one or two) tuning knobs, we can apply a simple brute-force search algorithm to find the optimal solution of (31). Without loss of generality, we assume that the tuning knobs can take H possible values $\{\mathbf{t}_h; h = 1, 2, \ldots, H\}$. The initial value of $\mathbf{t}$ is set as $\mathbf{t}_1$ in the first iteration, and $\{f(\mathbf{t}_1), g_p(\mathbf{t}_1); p = 1, 2, \ldots, P\}$ are either directly or indirectly measured. Next, in the second iteration, $\mathbf{t}$ is updated to $\mathbf{t}_2$, and $\{f(\mathbf{t}_2), g_p(\mathbf{t}_2); p = 1, 2, \ldots, P\}$ are measured. Similarly, in the h-th iteration, $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P\}$ are measured. In the end, we have a large data set $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P, h = 1, 2, \ldots, H\}$. The optimal solution $\mathbf{t}^*$ for the optimization in (31) can be eventually determined based on the performance values $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P, h = 1, 2, \ldots, H\}$. Algorithm 2 summarizes the details of the aforementioned brute-force search algorithm.

Algorithm 2: Brute-force Search for On-chip Self-healing

1. Start with the constrained optimization problem in (31) and H possible values $\{\mathbf{t}_h; h = 1, 2, \ldots, H\}$ for the tuning knobs. Set $h = 1$.
2. Set the tuning knobs to $\mathbf{t} = \mathbf{t}_h$.
3. Measure $f(\mathbf{t})$ and $\{g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$ and set $f(\mathbf{t}_h) = f(\mathbf{t})$, and $\{g_p(\mathbf{t}_h) = g_p(\mathbf{t}); p = 1, 2, \ldots, P\}$.
4. If $h < H$, $h = h + 1$ and go to Step 2. Otherwise, go to Step 5.
5. Based on the performance values $\{f(\mathbf{t}_h), g_p(\mathbf{t}_h); p = 1, 2, \ldots, P, h = 1, 2, \ldots, H\}$, determine the optimal solution $\mathbf{t}^*$ for the optimization in (31).

The brute-force search algorithm (i.e., Algorithm 2), though simple to implement, has no practical utility if we have a large number of tuning knobs. To understand the reason, let us consider the general case of U tuning knobs where the u-th tuning knob can take V_u possible values. In this case, we have $H =$

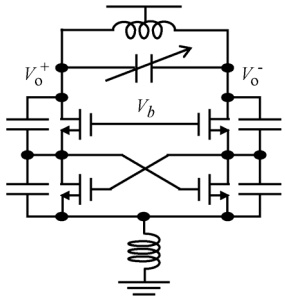


Fig. 2. Simplified circuit schematic is shown for a Colpitts VCO.

TABLE I
FREQUENCIES AND CORRESPONDING PHASE NOISE SPECIFICATIONS

Frequency (GHz)	26.2	24.6	23.3	22.2
PN Specification (dBc/Hz)	-123.5	-123.5	-123.5	-123.5

$V_1 \times V_2 \times \ldots \times V_U$ possible values for $\mathbf{t}$ in (31). With the increasing number of tuning knobs, the total number of possible values for these tuning knobs (i.e., H) will dramatically increase, thereby making the brute-force search algorithm quickly intractable. In these cases, other efficient search algorithms (e.g., interior point method [20]) must be applied to solve the optimization in (31) for on-chip self-healing.

Before ending this section, it is important to discuss the design overhead of on-chip self-healing that requires a number of additional circuitries (e.g., on-chip sensors, on-chip microcontroller, etc.), as shown in Fig. 1. There are several important clarifications we need to make here. First, many analog and RF circuit blocks on the same chip may require self-healing, and they can possibly share the same on-chip sensors and microcontroller. Second, for a typical system-on-chip (SoC) application, the microcontroller is needed for other computing tasks during the normal operation. In other words, the microcontroller is not added for on-chip self-healing only. For these reasons, the design overhead of on-chip self-healing is fairly small, or even negligible, in many application scenarios.

VI. CASE STUDY

In this section, a 25 GHz differential Colpitts VCO designed in a 32 nm CMOS SOI process is used to validate the proposed on-chip self-healing flow based on off-line data analysis. Fig. 2 shows the simplified schematic of the VCO. It consists of a cross-coupled differential pair connected to two common-gate Colpitts oscillators. The capacitor at the output is tunable so that the VCO frequency can be centered at different frequency bands. The bias voltage V_b is controlled by a DAC for self-healing. More details about the VCO design can be found in [21].

For the VCO shown in Fig. 2, since we only have one tuning knob (i.e., the bias voltage V_b), the simple brute-force search algorithm described in Algorithm 2 is applied for self-healing. In this example, phase noise is an important performance of interest, and its specifications derived from the system requirement for four different center frequencies are shown in Table I. If the phase noise value of a VCO is smaller than the given specification at all four frequencies shown in Table I, this VCO is considered as "PASS." Otherwise, we consider it as "FAIL." The objective of self-healing is to find the optimal bias voltage to minimize the phase noise.

Accurately measuring the phase noise at 25 GHz is not trivial. Hence, an indirect sensor is used for on-chip phase noise measurement (i.e., phase noise is considered as a PoI in Fig. 1). According to Leeson's model [22], oscillation frequency (x_1), oscillation amplitude (x_2), and bias current (x_3), all of which are easy to measure using fully integrated sub-circuits, have strong correlation with phase noise and, hence, are first chosen as PoMs. The sensors used to measure these three PoMs are listed in Table III. An on-chip current sensor to measure bias current will be integrated in our future work. More details about how to measure these three PoMs can be found in [15]. In addition, the tuning knob V_b (x_4) is considered as another PoM. Since V_b is directly controlled by a DAC, the digitized value of V_b is known, and no measurement is required. In total, four PoMs (i.e., $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]^T$) are chosen as the indirect sensor inputs, and are summarized in Table II. Next, a quadratic model template with four input variables (i.e., x_1, x_2, x_3 and x_4) and, hence, fifteen polynomial terms in total is used to build the indirect sensor for phase noise. With all fifteen polynomial terms, the average modeling error is 0.36 dBc/Hz. SR is then applied to simplify the quadratic model template. Nine polynomial terms are eventually selected by SR, as summarized in Table IV. The average modeling error of the simplified quadratic model is 0.41 dBc/Hz. The degradation of the modeling accuracy is negligible (0.05 dBc/Hz only). The accuracy of the simplified quadratic model with nine polynomial terms can be further demonstrated by the scatter plot between the actual phase noise and the predicted phase noise shown in Fig. 3.

Next, we collect four PoMs and phase noise at all possible bias voltages from a silicon wafer that contains 61 functional VCOs. The VCOs that are not functioning in this wafer are not considered here. These data are further used to calibrate the indirect

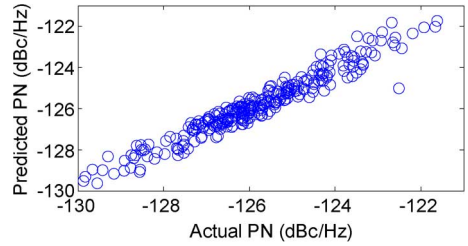


Fig. 3. Scatter plot is shown for the actual phase noise and the predicted phase noise based on the simplified quadratic model.

TABLE II
POI AND POMS OF INDIRECT PHASE NOISE SENSOR

	Performance Metric
PoI	Phase Noise
PoMs	Oscillation Frequency (x_1)
	Oscillation Amplitude (x_2)
	Bias Current (x_3)
	Bias Voltage (x_4)

TABLE III
POMS AND MEASUREMENT SENSORS

PoM	x_1	x_2	x_3
Measurement Sensor	On-chip Counter	On-chip Peak Detector + On-chip 6 Bit ADC	Off-chip Current Sensor

TABLE IV
BASIS FUNCTIONS SELECTED FOR INDIRECT PHASE NOISE SENSOR

Index	Term	Index	Term	Index	Term
1	x_2	4	x_1^2	7	$x_3 \cdot x_4$
2	x_3	5	$x_1 \cdot x_2$	8	x_4^2
3	x_4	6	$x_1 \cdot x_4$	9	const

TABLE V
PARAMETRIC YIELD OF THE WAFER BY USING A FIXED BIAS VOLTAGE

Bias Code	1	2	3	4	5	6
Parametric Yield	0	0	1.64%	11.48%	3.28%	0

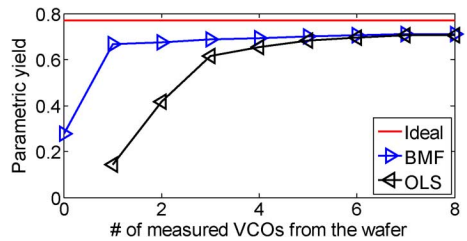


Fig. 4. Post-self-healing parametric yield of the wafer is shown as a function of the number of measured VCOs from the wafer.

phase noise sensor to improve its accuracy. Without self-healing, the parametric yield achieved by using a fixed bias voltage for all the VCOs on the wafer is summarized in Table V. Here, bias code denotes a digitized bias voltage, and parametric yield is defined as the ratio between the number of functional VCOs that can meet all four given phase noise specifications shown in Table I and the total number of functional VCOs. From Table V, we can see that the best parametric yield achieved by using a fixed bias voltage (i.e., bias code is 4) is only 11.48%. If other bias voltages are selected during the design, the parametric yield is even worse, which is almost zero for this wafer. It, in turn, serves as an excellent design case to demonstrate the importance of self-healing.

For testing and comparison purposes, three different self-healing methods are implemented:

- **Ideal**: The optimal bias voltage is determined by directly measuring the phase noise with an off-chip tester for all bias voltages. As a result, no indirect phase noise sensor is needed, and all the off-chip measurement data from the wafer will be used. This approach is not considered as on-chip self-healing; however, it provides the upper bound of the yield improvement that can be achieved by self-healing.
- **OLS**: The traditional OLS method is applied to fit the indirect phase noise sensor based on a number of measured VCOs from the wafer. Next, the indirect sensor is applied to self-heal all the VCOs on the wafer.
- **BMF**: The indirect phase noise sensor learned by SR is considered as the early-stage model. Next, the proposed BMF algorithm is applied to calibrate the early-stage model and generate a late-stage model based on a few measured VCOs from the wafer. The late-stage model is then applied to self-heal all the VCOs on the wafer.

Fig. 4 shows the parametric yield of the wafer achieved by three different self-healing methods given different number of measured VCOs from the wafer. Table VI further summarizes the measurement cost for self-healing. Studying Fig. 4 and Table VI reveals several important observations. First, BMF requires substantially less number of measured VCOs to build the indirect phase noise sensor than the traditional OLS method. In this example, BMF needs to measure one VCO only, while OLS requires measuring four VCOs (4×) to achieve a similar yield.

Second, studying the BMF results in Fig. 4, we notice that if no measurement data is collected from the wafer (i.e., the number of measured VCOs from the wafer is zero) and the self-healing is performed with the indirect sensor fitted from the

TABLE VI
MEASUREMENT COST AND PARAMETRIC YIELD BY SELF-HEALING

Self-Healing Method	# of Measured VCOs	Parametric Yield
Ideal	61	77.05%
OLS	4	65.30%
BMF	1	66.80%

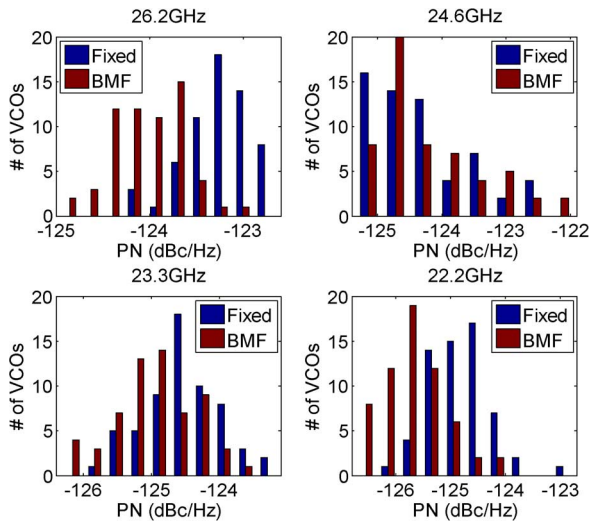


Fig. 5. Histogram of the measured phase noise values from all the VCOs on the wafer. Blue bars represent the results from Fixed where bias code is 4, and red bars represent the results from BMF where a single measured VCO is used from the wafer.

early-stage data by SR, the post-self-healing parametric yield is only 27.87%. Once a single VCO is measured from the wafer, the indirect phase noise sensor is calibrated by BMF and the post-self-healing parametric yield is increased to 66.80%. It, in turn, demonstrates that the aforementioned model calibration is a critical step for yield enhancement.

Finally, it is important to note that the post-self-healing parametric yield of BMF is close to that of the "ideal" case. It, in turn, implies that the modeling error of our proposed BMF method is fairly small, even if only a single VCO is measured from the wafer.

Before ending this section, we compare the phase noise values from the proposed self-healing flow to those from the fixed bias voltage method (Fixed) to study why the proposed flow can achieve a much better parametric yield than Fixed. Fig. 5 shows the histogram of the measured phase noise values from all the VCOs at different frequencies. The blue bars show the results from Fixed where bias code is 4, and the red bars show the results from our proposed BMF technique with a single measured VCO from the wafer. From Fig. 5, we have several observations. First, both BMF and Fixed get larger phase noise values at higher frequencies, which is consistent with our expectation. Hence, the phase noise specification at 26.2 GHz is the most difficult one to meet among all four phase noise specifications. Second, the proposed BMF technique can get much smaller phase noise values than Fixed at 26.2 GHz, which is the reason that BMF achieves a much better parametric yield than Fixed.

VII. CONCLUSIONS

In this paper, we propose a novel indirect performance sensing technique for on-chip self-healing of analog and RF circuits. In particular, a set of important basis functions are first identified

by SR so that the overhead of on-chip self-healing can be minimized. Next, the indirect sensors are repeatedly calibrated by BMF to accommodate the process shift associated with manufacturing lines. The indirect sensors are eventually stored in an on-chip microcontroller to facilitate efficient on-chip self-healing. The proposed indirect performance sensing and on-chip self-healing methodology is validated by a 25 GHz differential Colpitts VCO designed in a 32 nm CMOS SOI process. Our silicon measurement data show that the parametric yield of the VCO is significantly improved after applying self-healing. In our future work, we will further extend the proposed indirect performance sensing and on-chip self-healing methodology to other large-scale circuits such as phase-locked loop (PLL).

REFERENCES

[1] S. Yaldiz, V. Calayir, X. Li, L. Pileggi, A. S. Natarajan, M. A. Ferriss, and J. A. Tierno, "Indirect phase noise sensing for self-healing voltage controlled oscillators," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2011.

[2] S. Sun, F. Wang, S. Yaldiz, X. Li, L. Pileggi, A. S. Natarajan, M. A. Ferriss, J. Plouchart, B. Sadhu, B. D. Parker, A. Valdes-Garcia, M. Sanduleanu, J. Tierno, and D. Friedman, "Indirect performance sensing for on-chip analog self-healing via Bayesian model fusion," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2013.

[3] S. Nassif, "Modeling and analysis of manufacturing variations," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2001, pp. 223–228.

[4] X. Li, J. Le, and L. Pileggi, *Statistical Performance Modeling and Optimization*. Delft, The Netherlands: Now Publ., 2007.

[5] X. Li, P. Gopalakrishnan, Y. Xu, and L. Pileggi, "Robust analog/RF circuit design with projection-based performance modeling," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 1, pp. 2–15, Jan. 2007.

[6] T. McConaghy, "High-dimensional statistical modeling and analysis of custom integrated circuits," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2011.

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

[8] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 3, pp. 349–361, Mar. 2002.

[9] D. Han, S. S. Akbay, S. Bhattacharya, and A. Chatterjee, "On-chip self-calibration of RF circuits using specification-driven built-in self test (S-BIST)," in *Proc. IEEE Int. On-Line Testing Symp.*, Jul. 2005.

[10] E. Acar and S. Ozev, "Low-cost characterization and calibration of RF integrated circuits through I-Q data analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 7, pp. 993–1005, Jul. 2009.

[11] D. Han, B. S. Kim, and A. Chatterjee, "DSP-driven self-tuning of RF circuits for process-induced performance variability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp. 305–314, Feb. 2010.

[12] V. Nataranjan, S. Sen, A. Banerjee, and A. Chatterjee, "Analog signature- driven postmanufacture multidimensional tuning of RF systems," *IEEE Design Test Comput.*, vol. 27, no. 6, pp. 6–17, Dec. 2010.

[13] N. Kupp, H. Huang, P. Drineas, and Y. Makris, "Post-production performance calibration in analog/RF devices," in *Proc. IEEE Int. Test Conf.*, Nov. 2010.

[14] C. Chien, A. Tang, F. Hsiao, and M. F. Chang, "Dual-control self-healing architecture for high performance radio SoC's," *IEEE Design Test Comput.*, vol. 29, no. 6, pp. 40–51, Nov.–Dec. 2012.

[15] B. Sadhu, M. A. Ferriss, A. S. Natarajan, S. Yaldiz, J. Plouchart, A. V. Rylyakov, A. Valdes-Garcia, B. D. Parker, A. Babakhani, S. Reynolds, X. Li, L. Pileggi, R. Harjani, J. A. Tierno, and D. Friedman, "A linearized low-phase-noise VCO-based 25-GHz PLL with autonomic biasing," *IEEE J. Solid-State Circuits*, vol. 48, no. 5, pp. 1138–1150, May 2013.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[17] X. Li, "Finding deterministic solution from underdetermined equation: Large-scale performance modeling of analog/RF circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 11, pp. 1661–1668, Nov. 2010.

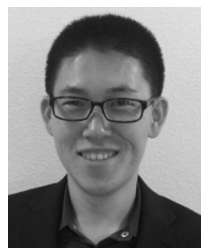
[18] C. Bishop, *Pattern Recognition and Machine Learning*. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

[19] F. Wang, W. Zhang, S. Sun, X. Li, and C. Gu, "Bayesian model fusion: Large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," in *Proc. Design Autom. Conf.*, 2013.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[21] J. Plouchart, M. A. Ferriss, A. S. Natarajan, A. Valdes-Garcia, B. Sadhu, A. V. Rylyakov, B. D. Parker, M. Beakes, A. Babakhani, S. Yaldiz, L. Pileggi, R. Harjani, S. Reynolds, J. A. Tierno, and D. Friedman, "A 23.5 GHz PLL with an adaptively biased VCO in 32 nm SOI-CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2009–2017, Aug. 2013.

[22] D. Leeson, "A simple model of feedback oscillator noise spectrum," *Proc. IEEE*, vol. 54, no. 2, pp. 329–330, Feb. 1966.



Shupeng Sun (S'11) received the B.S. degree in control theory and information technology from Tsinghua University, Beijing, China, in 2010. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

Fa Wang (S'13) received the B.S. degree in control theory and information technology from Tsinghua University, Beijing, China, in 2009. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

Soner Yaldiz (S'04–M'13) received the B.S. degree in microelectronics engineering from Sabanci University, Istanbul, Turkey, in 2004, the M.S. degree in electrical and computer engineering from Koc University, Istanbul, Turkey, in 2006, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2012. He has been working on analog circuit design methodologies at Intel Corporation, OR, USA, since January 2012.

Xin Li (S'01–M'06–SM'10) received the B.S. and M.S. degrees in electronics engineering from Fudan University, Shanghai, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2005. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

Lawrence Pileggi (F'02) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 1989. He is now the Tanoto professor in the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

Arun Natarajan received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, India, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2003 and 2007, respectively. He is currently an Assistant Professor in the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.

Mark Ferriss received the B.E. degree in electrical engineering from University College Cork, Cork, Ireland, in 1998, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2008. He has been working at the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, since 2009.

Jean-Olivier Plouchart (M'96–SM'06) received the Ph.D. degree in electronics from Paris VI University, Paris, France, in 1994. Currently, he leads the development of nanometer high-speed circuit design and high-yield nanometer design at the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.

Bodhisatwa Sadhu (S'08–M'12) received the B.E. (Hons.) degree in electrical and electronics engineering from Birla Institute of Technology and Science, Pilani, India, in 2007, and the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 2012. Currently, he works at the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.

Ben Parker received the B.S. degree from Bowdoin College, Brunswick, ME, USA, in 1979, and the M.S. degree from Brown University, Providence, RI, USA, in 1981, both in physics. Currently, he works at the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.

Alberto Valdes-Garcia (S'00–M'06–SM'13) received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 2006. He is currently a Research Staff Member and Manager of the RF Circuits and Systems Group at the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.

Mihai A. T. Sanduleanu received the M.Sc. degree from Technical University of Iasi, Romania, in 1990, the M.E.E. degree from Eindhoven University of Technology, The Netherlands, in 1994, and the Ph.D. in electrical engineering from the University of Twente, Enschede, The Netherlands, in 1999. He is currently an Associate Professor at Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates.

Jose Tierno received the engineering degree from the Universidad de la República, Montevideo, Uruguay, in 1988, the M.S. degree in electrical engineering and Ph.D. degree in computer science from the California Institute of Technology, Pasadena, CA, USA, in 1989 and 1995, respectively. Currently, he works at Apple Inc., Cupertino, CA, USA, as a Research Scientist.

Daniel Friedman (S'91–M'92) received the Ph.D. degree in engineering science from Harvard University, Cambridge, MA, USA, in 1992. Currently, he is the Manager of the communication circuits and systems group at the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.