Relevance Vector and Feature Machine for Statistical Analog Circuit Characterization and Built-In Self-Test Optimization

Honghuang Lin

Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843
linhh@tamu.edu

Peng Li

Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843
pli@tamu.edu

ABSTRACT

Aiding design and test optimization of analog circuits requires accurate models that can reliably capture complex dependencies of circuit performances on essential circuit and device parameters, and test signatures. We present a novel Bayesian learning technique, namely *relevance vector and feature machine* (RVFM), for characterizing analog circuits with sparse statistical regression models. RVFM not only produces accurate models learned from a moderate amount of simulation or measurement data, but also computes a probabilistically inferred weighting factor quantifying the criticality of each parameter as part of the overall learning framework, hence offering a powerful enabler for variability modeling, failure diagnosis, and test development. Compared to other popular learning-based techniques, the proposed RVFM produces more accurate models, requires less amount of training data, and extracts more reliable parametric ranking. The effectiveness of RVFM is demonstrated in terms of the statistical variability modeling of a *low-dropout regulator* (LDO) and the *built-in self-test* (BIST) development of a charge-pump *phase-locked loop* (PLL).

1. INTRODUCTION

As the complexity of *analog/mixed-signal* (AMS) circuits keeps increasing at a rapid pace, the tasks of design, verification and test have become significant challenges. Nevertheless, it is essential to characterize the dependencies of circuit performances/specifications on various circuit and device parameters or test signatures for purposes such as design, verification and test optimization. However, doing so is not trivial since the targeted dependencies are usually complex and nonlinear with deep-rooted correlations, making it arduous to reliably quantify the importance of numerous parameters.

For characterizing sophisticated circuit systems, machine learning techniques based on circuit simulations or measurements have been proven to be effective and produced promising outcomes. For example, *support vector machines* (SVMs) [1] are used as nonlinear classifiers in [2] to capture

the mapping from input parameters to circuit performance. A regression extension to SVM is employed in [3] to rank circuit parameters based on their correlations with unexpected timing deviations. Additionally, Bayesian inference is often used to build statistical circuit models. For instance, a co-learning Bayesian model is proposed in [4] to efficiently model the performance of AMS circuits.

Input parameters of vastly different amplitudes are often normalized to the same value range before being fed to a machine learning algorithm. This and other issues may amplify the impact of redundant or noisy parameters in the model and aggravate its vulnerability to noisy training data and/or inclusion of noisy or redundant input features (parameters). Since the sensitivities of circuit performances to various parameters may vary vastly, it is instrumental to extract accurate statistical models and reliable parameter criticality simultaneously. While traditional feature selection or importance ranking techniques may help to identify and select some important parameters out of a large parameter set, building models only with the selected parameters usually degrades the model performance and few of those techniques can guide the model to achieve higher accuracy [5, 6]. These difficulties present important roadblocks to analog/mixed-signal circuit characterization with machine learning techniques.

To achieve both objectives, this paper proposes a novel Bayesian learning framework for characterizing analog circuits with sparse statistical regression models. The proposed framework is named *relevance vector and feature machine* (RVFM) and can be considered as a significant extension to the SVM [1] and *Relevance Vector Machine* (RVM) [7]. The RVFM simultaneously seeks relevant training samples (i.e. vectors) and parameters (i.e. features) to derive a sparse model in both the vector and parameter spaces based upon a novel decomposed feature-weighted kernel function. As a result, the RVFM not only produces accurate models learned from a moderate amount of simulation or measurement data, but also computes a probabilistically inferred weighting factor quantifying the criticality of each parameter as part of the overall learning framework, hence offering a powerful enabler for variability modeling, failure diagnosis, and test development. In addition, an iterative algorithm is developed for efficient training of the proposed RVFM.

Compared to other popular learning-based techniques, the RVFM produces more accurate models, requires less amount of training data, and extracts more reliable parametric ranking. The effectiveness of RVFM is demonstrated in terms of the statistical variability modeling of a *low-dropout regulator* (LDO) and the *built-in self-test* (BIST) development and optimization of a charge-pump *phase-locked loop* (PLL).

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.

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/10.1145/2897937.2898081>

2. BAYESIAN LEARNING

Throughout this paper bold capital letters denote matrices and bold letters in lower case denote vectors. For a matrix $\mathbf{X}$, we use $\mathbf{X}(i, j)$ to denote its entry at the i -th row and j -th column. We denote the i -th entry of a vector $\mathbf{x}$ by $\mathbf{x}(i)$.

2.1 Kernel Machines

Assuming that there are F circuit parameters of interest with which the circuit is described by a parameter (feature) vector $\mathbf{x}$. A sample of the circuit is defined by a pair $\{\mathbf{x}_i, t_i\}$ where t_i is the circuit performance under the configuration $\mathbf{x}_i$. By collecting a number of N samples, the objective of regression is to capture the mapping $\Psi : \mathbf{x} \rightarrow t$ with a function $y(\mathbf{x})$ whose output can be used as a prediction of the performance t .

For analog/mixed-signal circuits, the mapping from the parameters to the performance Ψ is usually nonlinear. For this, a kernel method in the following form is often used:

$$y(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^N w_i K(\mathbf{x}, \mathbf{x}_i), \quad (1)$$

where $\mathbf{x}$ is the sample being predicted, $\mathbf{x}_i$ is the training samples, and $K(\mathbf{x}, \mathbf{x}_i)$ is the kernel function. By applying (1) to all the training samples, the regression model be rewritten as:

$$\mathbf{t} = \Phi \cdot \mathbf{w} + \mathbf{e}, \quad (2)$$

where Φ is an $N \times N$ matrix defined by $\Phi(i, j) = K(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{t}$ is simply the target vector with $\mathbf{t}(i) = t_i$, and $\mathbf{e}$ is the error vector for the N training samples. Models in this form are called kernel machines and have been widely used in the SVMs [1] and other kernel methods.

2.2 Existing Bayesian Models

The *relevance vector machines* (RVM) [7] is a sparse Bayesian model providing a viable probabilistic framework for regression. The Bayesian network model of the RVM is shown in Fig. 1.

Figure 1: The network model of the RVM where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. Circles denote random variables and squares denote deterministic model parameters.

Given the input vectors $\{\mathbf{x}_n\}_{n=1}^N$ and their corresponding targets $\{t_n\}_{n=1}^N$, the RVM is used to probabilistically determine the model (2). The given target values are modeled as random variables by assuming every entry of the additive noise $\mathbf{e}$ is a zero-mean Gaussian random variable with variance σ^2 . Further assuming the independence of t_n gives the following probability distribution

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = \prod_{n=1}^N \mathcal{N}(y(\mathbf{x}_n; \mathbf{w}), \sigma^2). \quad (3)$$

Different from deterministic learning models (e.g. the SVM) that compute $\mathbf{w}$ directly, the RVM defines the prior

distribution of $\mathbf{w}$ as independent zero-mean Gaussian random variables with variance α , and compute α instead of $\mathbf{w}$ in the training process:

$$p(\mathbf{w}|\alpha) = \prod_{n=1}^N \mathcal{N}(0, \alpha_n^{-1}). \quad (4)$$

If $\alpha_i < \infty$, w_i is called a *relevance vector* since it has a variance greater than zero, allowing x_i making contributions to the decision function. Note that the RVM performs prediction with the posterior probability of the internal variables (i.e. the weights). Via convolution of Gaussian distributions, the covariance and mean of the posterior $p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2)$ can be shown to be respectively:

$$\Sigma = (\sigma^{-2} \Phi^T \Phi + \mathbf{A})^{-1} \quad (5)$$

$$\mu = \sigma^{-2} \Sigma \Phi^T \mathbf{t}, \quad (6)$$

where $\mathbf{A} = \text{diag}(\alpha)$.

The objective of the Bayesian network is to find the most probable model parameters with the given training samples, i.e. to maximize the posterior probability $p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t})$. Based on the Bayes' rule, such objective is equivalent to:

$$\arg \max_{\alpha, \sigma^2} p(\mathbf{t}|\alpha, \sigma). \quad (7)$$

A fast training algorithm is proposed in [8] to compute the optimal α and σ^2 . After the training, for any $\hat{\mathbf{x}}$, the expectation of its predicted target is:

$$\hat{y} = \mu^T \phi(\hat{\mathbf{x}}) \quad (8)$$

and the variance is:

$$\hat{\sigma} = \sigma^2 + \phi(\hat{\mathbf{x}})^T \Sigma \phi(\mathbf{x}_u) \quad (9)$$

where $\phi(\hat{\mathbf{x}})$ is a vector of size N whose i -th entry is defined by $\phi(\hat{\mathbf{x}})(i) = K(\hat{\mathbf{x}}, \mathbf{x}_i)$.

Under the same framework, a feature ranking technique is proposed in [6]. By defining the feature vector $\mathbf{f}_i = (\mathbf{x}_1(i), \mathbf{x}_2(i), \dots, \mathbf{x}_F(i))^T$ and a new design matrix $\Phi'(i, j) = K(\mathbf{f}_i, \mathbf{f}_j)$, [6] proposes a feature weighting model is:

$$\mathbf{t}' = \Phi' \cdot \mathbf{v} + \mathbf{e} \quad (10)$$

where $\mathbf{t}'(i) = K(\mathbf{t}, \mathbf{f}_i)$ and $\mathbf{v}$ is the weights of all features. Since (10) adapts the RVM to identify relevance features, we refer to this technique as RFM.

3. RELEVANCE VECTOR AND FEATURE MACHINE

To achieve high model accuracy and feature weighting with high quality, we propose the RVFM whose conceptual structure is shown in Fig. 2.

Here the key idea is to explore both vector weighting w in the sample space and feature weighting v in the parameter space simultaneously.

3.1 The RVFM Bayesian Network

As the first step towards developing the RVFM, we realize parameter weighting or feature ranking by directly assigning weights $\mathbf{v}$ to the parameters. For this, the Bayesian network described in Fig. 1 is extended to derive the new model on the left of Fig. 3. It is further assumed that $\mathbf{v}$ are also Gaussian random variables with $p(\mathbf{v}|\beta) = \prod_{n=1}^F \mathcal{N}(0, \beta_n^{-1})$. This implies the use of a new nonlinear kernel function (e.g.

Figure 2: The conceptual RVFM model. Small black dots denote training data, with each row representing a vector and each column representing a feature.

Figure 3: Development of RVFM Bayesian network where $K_{ijk} = K(\mathbf{x}_i(k), \mathbf{x}_j(k))$.

$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\text{diag}(\mathbf{v}) \cdot (\mathbf{x}_i - \mathbf{x}_j)\|^2}$ which is Gaussian kernel with feature weighting).

However, this nonlinear kernel makes the training of the network extremely challenging since the deterministic relationship from $\mathbf{w}$ and $\mathbf{v}$ to t is nonlinear. More specifically, the following optimization objective of the Bayesian training process is not analytically computable and hence hinders the optimization based training process:

$$p(\mathbf{t}|\alpha, \beta, \sigma^2) = \iint p(\mathbf{t}|\mathbf{w}, \mathbf{v}, \sigma^2) \cdot p(\mathbf{w}|\alpha) \cdot p(\mathbf{v}|\beta) d\mathbf{w} d\mathbf{v} \quad (11)$$

To address the above difficulty, instead of defining $\mathbf{v}$ as the weights of each parameter, we define a new kernel with tractable feature weights by assigning weights to the decomposed kernel functions:

$$K_{new}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^F \mathbf{v}(k) \cdot K(\mathbf{x}_i(k), \mathbf{x}_j(k)). \quad (12)$$

Substituting the new kernel function into the design matrix of (2) leads to the observation of an interesting and useful property of the new general liner model, i.e. $\mathbf{w}$ and $\mathbf{v}$ are now interchangeable. One way to express (2) is:

$$\mathbf{t} = \Phi' \cdot \mathbf{v} + \mathbf{e}, \quad (13)$$

where the new design matrix Φ' has a new size of $N \times F$ and is now defined by $\Phi'(i, j) = \sum_{k=1}^N \mathbf{w}(k) \cdot K(\mathbf{x}_i(j), \mathbf{x}_k(j))$.

It is important to note that, now due to the terms like $\mathbf{w}_i \mathbf{v}_j$ in (13), the target t is bilinear in terms of each vector or feature weight. Furthermore, we replace the product of two random variables $w_i v_j$ by a single random variable $u_{i,j}$, which leads to our proposed computable linear Bayesian network shown on the right of Fig. 3. If we define a new vector $\mathbf{u}$ of size $N \cdot F$ whose entry $\mathbf{u}((i-1)N+j) = u_{ij} = \mathbf{w}(i) \cdot \mathbf{v}(j)$,

the model becomes linear in $\mathbf{u}$:

$$\mathbf{t} = \Phi_{\mathbf{u}} \cdot \mathbf{u} + \mathbf{e}, \quad (14)$$

where $\Phi_{\mathbf{u}}$ is an $N \times (N \cdot F)$ matrix with the entries $\Phi_{\mathbf{u}}(i, (j-1)N+k) = K(\mathbf{x}_i(k), \mathbf{x}_j(k))$.

In the RVM, the zero-mean Gaussian prior distribution of $\mathbf{w}$ tends to help the model converge to a sparse solution since it results in the marginal prior distribution over $\mathbf{w}$ being *Student-t* distribution. A majority of the entries of $\mathbf{w}$ are likely to be zero or very close to zero, and thus can be pruned. Similarly, in our proposed Bayesian model, we use zero-mean Gaussian as the prior of $\mathbf{u}$ to achieve sparse solutions. Considering the nature of $\mathbf{u}$, if $u_{i,j}$ is pruned, either the i -th sample (the whole i -th row) or the j -th parameter (the whole j -th column) should be pruned as well. To reflect this, we define a proper prior for $\mathbf{u}$ as:

$$p(\mathbf{u}|\alpha, \beta) = \prod_{i=1}^N \prod_{j=1}^F \mathcal{N}(0, \alpha_i^{-1} \beta_j^{-1}). \quad (15)$$

To see how this new prior works, let us assume that $\alpha_i \rightarrow \infty$, then all the $\{u_{i,j}\}_{j=1}^F$ are zero, i.e. the i -th sample is discarded from the set of relevance vectors. Similarly, if $\beta_j < \infty$, the j -th parameter is relevant and there should be at least one non-zero $u_{i,j}$ for $i \in [1, N]$.

The posterior covariance and mean of $p(\mathbf{u}|\mathbf{t}, \alpha, \beta, \sigma^2)$ under the proposed Bayesian network are found to be:

$$\Sigma_{\mathbf{u}} = (\sigma^{-2} \Phi_{\mathbf{u}}^T \Phi_{\mathbf{u}} + \mathbf{A}_{\mathbf{u}})^{-1} \quad (16)$$

$$\mu_{\mathbf{u}} = \sigma^{-2} \Sigma_{\mathbf{u}} \Phi_{\mathbf{u}}^T \mathbf{t} \quad (17)$$

where $\mathbf{A}_{\mathbf{u}} = \text{diag}(\alpha_1 \beta_1, \alpha_1 \beta_2, \dots, \alpha_N \beta_F)$ and $\Sigma_{\mathbf{u}}$ is the new design matrix defined in (14).

3.2 Efficient Training Algorithm

The marginal likelihood maximization [8] required in training the RVM model is solved in an iterative process similar to the well-known *expectation maximization* (EM) algorithm. Due to the required matrix operations, the computational complexity in each iteration is $O(M^3)$ if there are M relevance vectors.

Figure 4: Efficient RVFM model with network reduction.

For the proposed RVFM whose architecture is shown in Fig. 2, if there are M relevance vectors and E relevance features, the number of the internal model variables is $M \cdot E$. Applying the RVM training algorithm will lead to a very high computational complexity of $O(M^3 E^3)$.

To address this computational challenge, our proposed efficient algorithm leverages the elegant property that $\mathbf{w}$ and $\mathbf{v}$ are interchangeable in the bilinear Bayesian network, reducing our model to a standard RVM. As Fig. 4 shows, fixing α and moving the resulting expectation of $\mathbf{w}$ into the design

Figure 5: Low-dropout regulator (LDO).

matrix (i.e. converting Φ_u in (14) to Φ' in (13)) will reduce every column in Fig. 4 to a single weight v_j with its prior β_j . Similarly, row-wise reduction by fixing β converts the proposed network to another RVM network with w and α .

The above discussion suggests an efficient iterative training process. In each iteration, we reduce the model row-wise and column-wise, respectively, and update α and β subsequently. The complexity in each iteration is now either $O(M^3)$ or $O(E^3)$.

Note that $u_{i,j} = w_i v_j$ is defined only for the conceptual model development and illustration purposes. In the actual training process, only α and β rather than w , v or u are updated in each iteration.

4. EXPERIMENTS

To demonstrate the superiority of the proposed RVFM, we compare its performance with popular learning-based techniques including the SVM [1] and the RVM [7]. We also compare the RVFM with the RFM outlined in Section 2.2 in terms of parameter (feature) ranking.

4.1 Variability Analysis of an LDO

Building an accurate regression model for a given analog performance and performing feature ranking among all sorts of process parameters are key to the understanding of the impacts of process variabilities on analog circuits. Since simulations or measurements are usually expensive, it is of great significance to build an accurate regression model and obtain reliable parameter weighting with a moderate amount of samples, which turns out to be a task well handled by the proposed method.

We investigate the process variations in a realistic *low-dropout regulator* (LDO) design (Fig. 5) proposed in [9]. We build RVFMs to analyze the impact of process variations on LDO specifications including its quiescent current, undershoot of the output voltage V_{out} and load regulation. Channel length variations of all transistors in the LDO are modeled at the SPICE level using a commercial 90nm CMOS technology design kit. We use various numbers of simulation samples to build a regression model relating the model process parameters with each targeted specification and test the accuracies of these models using a testing set of 1,000 simulation samples. The results are shown in Fig. 6.

In this experiment, *normalized mean square error* (NMSE) is used as the metric to evaluate the performance of the predictors trained with different techniques. As Fig. 6 shows, the RVFM out-performs the popular SVM and RVM in all cases by achieving one-order of magnitude lower NMSEs.

We compare the ranking produced by the RVFM and the RFM on feature ranking in Fig. 8. To evaluate the quality of the ranking, for each design specification, we train two

Figure 6: Regression performance comparison.

RVMs only in the process parameters selected by RVFM and RFM, respectively. A parameter is selected by RVFM or RFM if its expected weight is greater than 0.01. Such procedure is firstly applied to the regression model with 20 channel length variations, i.e. the three columns on the left of Fig. 7. Then, the same procedure is applied to an expanded full parameter set of 60 parameters involving variations of each transistor's channel length, oxide thickness and threshold voltage (on the right of Fig. 7). The resulting NMSEs and the numbers of parameters selected indicate that the RVFM produces more reliable parameter weighting and reaches similar sparsity compared to the RFM.

Figure 7: Ranking quality comparison with the number of selected parameters (QC: quiescent current, US: undershoot, LR: load regulation).

We use design knowledge to provide further insights and validation for the parameter rankings of the 20 channel length variations computed by the RVFM. For example, based on the analysis in [10] a majority portion of the multi-loop LDO's quiescent current is consumed by the fastest two loops in the output stage and hence the variation on $M2$ has

Figure 8: Weights of transistor’s channel length variation in the model of: (a) quiescent current; (b) undershoot; (c) load regulation. Red lines represent the 95% confidence intervals estimated by RVFM.

significant impact on the quiescent current. Moreover, variations on $M3$, $M7$, $M8$ and $M9$ may lead to mismatches and considerable changes at the two output nodes of the error amplifier, one of which is V_g of $M2$. This analysis matches the ranking shown in Fig. 8(a).

The undershoot of the LDO is mainly determined by the load capacitor and the loop bandwidth, which is further determined by the error amplifier (involving $M3 \sim M10$), the fast loop in the output stage ($M12$), and the in-band zeros locations defined as:

$$\omega_{LCZ} \approx \sqrt{\frac{g_{m1}g_{m11}g_a}{g_{m2}C_{C2}(C_{C1} + C_{C3})}}, \quad (18)$$

where g_a is the output admittance of the error amplifier defined by the g_m of $M7 \sim M10$. The ranking of the RVFM in Fig. 8(b) is reliable since it captures all these relevant variations.

Load regulation of the LDO is mainly determined by the DC loop gain, which is the product of the gains of all stages in the loop. The gain of the EA stage is inversely proportional to the g_m of $M7 \sim M10$ and the second stage is comprised of $M17$ and $M11$. Again, the ranking of the RVFM as shown in Fig. 8(c) successfully identifies all these important variations.

4.2 PLL BIST Scheme Optimization

Built-in-self-test (BIST) is very effective in detecting operational failures of deployed analog/mixed-signal circuits. Base on the concept of alternative test, efficient BIST solutions can be formed by collecting low-cost test signatures and relating the signatures to targeted performance specifications using statistical prediction models. The effectiveness of BIST heavily depends on the quality of the selected signatures and the tradeoffs between accuracy, overhead, and test time. We apply the RVFM to the BIST of a charge-pump PLL targeting three key specifications: *lock-time*(LT), *frequency overshoot*(OVS), and *jitter*(JT).

Figure 9: A PLL and three BIST schemes.

Fig. 9 shows the PLL along with three BIST schemes using various test signatures. Jitter, frequency overshoot and lock-time are important specifications but cannot be easily measured directly on the chip. To capture failures in those specifications, the first candidate BIST scheme [11] collects the readouts of the counter in the divider as its test signature, while the second scheme [12] collects the accumulated *up* and *dn* phase detector outputs via integrators and *time-to-digital converters* (TDCs). The third scheme is an example of IDDQ testing, measuring the quiescent currents of the *charge pump* (CP) and the *voltage control oscillator* (VCO) as test signatures similar to the approach of [13].

The first two schemes operate in a special test mode which instead of feeding back the divider output, it first feeds the one-buffer delayed reference clock to the phase detector for 8 reference cycles with a cycle time of $0.1 \mu s$. Then, the reference clock input is replaced by the double delayed reference clock for another 8 cycles. Each cycle generates one signature for Scheme 1 and two for Scheme 2, making a total of 16 and 32 signatures for Scheme 1 and 2, respectively. Scheme 3 reads out two signatures, i.e. the CP and VCO quiescent currents, in the quiescent mode.

Figure 10: Classifier performance comparison.

Recently, learning-based classifiers like the SVM have been trained to perform the failure detection in BIST [11, 14]. To make better usage of the collected test signatures, we apply the proposed RVFM in each scheme. We fit the target specification into a sigmoid function before we employ the RVFM as a classifier for failure detection. Three classification techniques, the SVM, RVM, and RVFM, are trained with 200 simulation samples and tested with 4,000 samples. The classifying errors are compared in Fig. 10 which shows the superior BIST classifier accuracy of the proposed RVFM.

In addition, the RVFM also produces reliable ranking among test signatures, which can be further leveraged to improve the efficiency of BIST schemes. For example, the RVFM ranks the 16 test signatures in Scheme 1 as shown in Fig. 11 when building the classifier for jitter failure detection. The tenth signature is the last one with a significant weight. After that, the remaining 6 signatures are of little importance and can be considered as redundant. Using the same procedure, we reduce the test time for each of the three

Spec.	Original Best Accuracy	Synthesized Signatures			Test Time (us)			Test Time Reduction	Optimized Accuracy
		Sch.1	Sch.2	Sch.3	Sch.1&2	Sch.3	Total		
JT	97.22%	1 - 3	1	VCO	0.3	0.6	0.9	43.75%	99.98%
OVS	98.00%	1 - 3	1 - 7	CP	0.4	0.6	1.0	37.50%	99.88%
LT	96.40%	1 - 4	1 - 2	VCO	0.4	0.6	1.0	37.50%	99.95%

Spec.	Original Accuracy	Selected Readouts	Resulting Accuracy	Test Time Reduction
JT	97.22%	1 - 10	96.20%	37.5%
OVS	95.78%	1 - 12	94.89%	25.0%
LT	96.20%	1 - 6	97.00%	62.5%

Assuming that realizing all three schemes on-chip does not lead to significant overhead, we seek to improve BIST accuracy by leveraging the signatures of all the schemes. While combining all the signatures can offer the best accuracy, it may not be completely efficient due to the existence of redundant signatures. For this, we train an RVFM on all the signatures across the three schemes to predict the jitter. Based on the signature ranking shown in Fig. 12, we collect the first three signatures in Scheme 1 and the first signature in Scheme 2. Although the third last signature in Scheme 2 also possesses a notable weight, collecting such signature is not cost-effective in terms of test time, and thus it is discarded. For Scheme 3, only the quiescent current of VCO is selected, which can be measured in $0.6us$ according to [13].

5. CONCLUSION

dependencies on assorted parameters or signatures via a statistical regression model. The advantages of the proposed framework are demonstrated by variability analysis of an LDO and BIST optimization for a charge-pump PLL.

This material is based upon work supported by the National Science Foundation under Grant No. 1117660 and by the Semiconductor Research Corporation (SRC) through Texas Analog Center of Excellence at the University of Texas at Dallas (Task ID:1836.128).

- [1] V. Vapnik, S.E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, pages 281–287, 1997.
- [2] Honghuang Lin and Peng Li. Circuit performance classification with active learning guided sampling for support vector machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(9):1467–1480, Sept 2015.
- [3] Pouria Bastani, Nicholas Callegari, Li-C Wang, and Magdy S Abadir. Statistical diagnosis of unmodeled systematic timing effects. In *Proceedings of the 45th annual Design Automation Conference*, pages 355–360. ACM, 2008.
- [4] Fa Wang, Manzil Zaheer, Xin Li, Jean-Olivier Plouchart, and Alberto Valdes-Garcia. Co-learning bayesian model fusion: Efficient performance modeling of analog and mixed-signal circuits using side information. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 575–582. IEEE Press, 2015.
- [5] Sepp Hochreiter and Klaus Obermayer. Nonlinear feature selection with the potential support vector machine. In *Feature Extraction*, pages 419–438. Springer, 2006.
- [6] Haibin Cheng, Haifeng Chen, Guofei Jiang, and Kenji Yoshihira. Nonlinear feature selection by relevance feature vector machine. In *Machine learning and data mining in pattern recognition*, pages 144–159. Springer, 2007.
- [7] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244, 2001.
- [8] Michael E Tipping, Anita C Faul, et al. Fast marginal likelihood maximisation for sparse bayesian models. In *Proceedings of the ninth international workshop on artificial intelligence and statistics*, volume 1, 2003.
- [9] Suming Lai and Peng Li. A fully on-chip area-efficient cmos low-dropout regulator with fast load regulation. *Analog Integrated Circuits and Signal Processing*, 72(2):433–450, 2012.
- [10] Suming Lai. *Modeling, Design and Optimization of IC Power Delivery with On-Chip Regulation*. Doctoral dissertation, Texas A&M University, 2014.
- [11] Guo Yu and Peng Li. A methodology for systematic built-in self-test of phase-locked loops targeting at parametric failures. In *Test Conference, 2007. ITC 2007. IEEE International*, pages 1–10. IEEE, 2007.
- [12] Sen-Wen Hsiao, Nicholas Tzou, and Avyishek Chatterjee. A programmable bist design for pll static phase offset estimation and clock duty cycle detection. In *VLSI Test Symposium (VTS), 2013 IEEE 31st*, pages 1–6. IEEE, 2013.
- [13] Samed Maltabas, Osman Kubilay Ekekon, Kemal Kulovic, Anne Meixner, and Martin Margala. An iddq bist approach to characterize phase-locked loop parameters. In *VLSI Test Symposium (VTS), 2013 IEEE 31st*, pages 1–6. IEEE, 2013.
- [14] Ahcene Bounceur, Belkacem Brahmi, Kamel Beznia, and Reinhardt Euler. Accurate analog/rf bist evaluation based on svm classification of the process parameters. In *Design & Test Symposium (IDT), 2014 9th International*, pages 55–60. IEEE, 2014.