Adjustment-Based Modeling for Timing Analysis Under Variability

Lin Xie, *Student Member, IEEE*, Azadeh Davoodi, *Member, IEEE*, Jun Zhang, and Tai-Hsuan Wu, *Student Member, IEEE*

***Abstract*—This paper presents an adjustment-based modeling framework for timing analysis under variability. Instead of building a complex model (such as polynomial one) directly between the circuit timing and parameter variability, we propose to build a model that adjusts an approximate variation-aware timing into an accurate one. The idea is that it is easier to build a model that adjusts an approximate estimate into an accurate one. In addition, it is more efficient to obtain an approximate circuit timing model. The combination of these two observations makes the use of an adjustment-based model a better choice for statistical static timing analysis with high dimension of parameter variability (e.g., at sign-off stage). It can also be used at the postsilicon stage to predict the circuit timing from a smaller subcircuit. To build the adjustment model, we use a simulation-driven approach based on Gaussian Process. Combined with the intelligent sampling, we show that an adjustment-based model can more effectively capture the nonlinearity of the circuit timing with respect to parameter variability compared to polynomial models. Simulation results show that with 42 independent device and interconnect parameter variations, our proposed adjustment-based model obtained using 200 circuit timing samples can achieve much higher accuracy than quadratic model obtained using 2000 samples.**

***Index Terms*—Adjustment-based model, Gaussian Process (GP), Latin Hypercube Design (LHD), postsilicon timing prediction, statistical static timing analysis (SSTA).**

I. Introduction

IN NOWADAYS sub-45-nm era, we expect an increase in the number of parameter variations as well as in their ranges [11], [13]. Consequently, we need efficient and accurate statistical static timing analysis (SSTA) techniques that can handle high dimension of variability. This type of modeling might be necessary nowadays during the sign-off stage when several accurate interconnect and device parameter variability should be incorporated during SSTA. It can also be used to more effectively make optimization decisions or to perform sensitivity analysis. The majority of the existing SSTA techniques are based on polynomial modeling of the circuit timing such as the following recent ones [3], [13], [16], [19]–[22]. Many of the existing works have experimented with few number of parameter variations (for example one global $V_{\rm t}$, $L_{\rm eff}$, $t_{\rm ox}$ for all devices) and reported high modeling accuracy. However, with increase in dimension of parameter variability, we require more complex polynomials to achieve the same accuracy.

In this paper, we first discuss the shortcomings of the low-order polynomial models and then introduce a new SSTA framework, which is particularly useful for high dimension of variability. We study Gaussian Process (GP) as an alternative to model the nonlinear dependence between the circuit timing and the parameter variability. Specifically, the GP model is built using a simulation-driven approach, and is based on the input/output behaviors of a limited number of intelligently chosen observation samples. We show that GP model has a simpler form than high-order polynomial models. However, the complexity of this model depends on the number of collected samples.

To overcome the aforementioned problems, we then introduce a novel adjustment-based modeling framework for SSTA. The adjustment-based modeling framework has recently been proposed in the statistics community [17], [24]. For SSTA, instead of building a complex model between the circuit timing and the parameter variability, we propose to build a simple model which adjusts an approximate circuit timing into an accurate one. The adjustment model is built using GP model, so it is also simulation driven. However, we show that the number of required samples can be greatly reduced for an adjustment-based model. This is because the approximate circuit timing models have already captured some nonlinear dependence in terms of the input parameter variability.

Specifically, we introduce two approaches to obtain approximate circuit timing. The first one predicts the circuit timing based on an extracted timing-critical subcircuit. The second one is to use a cheap linear SSTA. We then adjust these approximate timings into accurate ones using GP models. Here, by accurate timing, we mean the timing obtained using Monte Carlo (MC) simulation. Note that MC has always been used as the reference for accuracy [13], [16], [19], [20]. Simulation results show that our proposed adjustment-based timing model achieves higher accuracy compared to a quadratic models of the circuit timing that is directly a function of input parameter variables. We show that for 42 independent parameter variations, we only need 200 samples to build the adjustment-based model, while a fitting-based quadratic form requires at least 10 000 samples.

The remaining of this paper is organized as follows. We discuss the shortcomings of the existing polynomial-based SSTA

Manuscript received July 9, 2008; revised December 7, 2008. Current version published June 17, 2009. This work was supported by the National Science Foundation under Award 0811082. An extended abstract of this paper with few details was published by the 2008 International Conference on Computer-Aided Design (ICCAD'08) [9]. This paper was recommended by Associate Editor C. V. Kashyap.

L. Xie, A. Davoodi, and T.-H Wu are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: lxie2@wisc.edu; adavoodi@wisc.edu; twu3@wisc.edu).

J. Zhang is with the Department of Statistics, University of Wisconsin, Madison, WI 53706 USA (e-mail: jzhang@stat.wisc.edu).

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/TCAD.2009.2018874

0278-0070/$25.00 © 2009 IEEE

approaches in Section II. We review GP models in Section III. We introduce the adjustment-based modeling framework and its sampling design in Sections IV and V, respectively. We apply this framework to SSTA, and propose two approaches to generate approximate circuit timing models in Section VI. We discuss the application of this framework in postsilicon timing prediction in Section VII. Simulation results and conclusions are given in Sections VIII and IX, respectively.

II. Previous Works on Polynomial-Based SSTA and Their Shortcomings

The majority of existing SSTA approaches are based on polynomial models [3], [13], [16], [19]–[22], in which the delay of each logic gate is expressed as a linear or quadratic function of random variables (RVs) representing different types of parameter variability. With these gate delay models, SSTA applies "MAX" operations on signal arrival times at various stages of the circuit, and expresses the final circuit timing as a polynomial to capture the inherent nonlinear relationship with the parameter variations. A generic polynomial expression of d number of RVs is given by

$$f(x_1,\ldots,x_d) = f_0 + \sum_{i=1}^{d} A_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} B_{ij} x_i x_j + \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d} C_{ijk} x_i x_j x_k + \cdots \quad (1)$$

where each RV x_i represents variations in transistor or interconnect parameters.

Specifically, [16] and [18] use quadratic forms of (1) to express the circuit timing. Reference [19] further simplifies the quadratic form by eliminating the cross-terms. Using such expressions, the existing approaches all report high modeling accuracy. However, the number of parameter variations used for verification of accuracy is always low in current literature (e.g., only four types of parameter variations are reported in [20] and [22]). It is unclear whether the accuracy of the model will be maintained with the increase in the dimension of the parameter variability. In addition, with the increase in the range of each parameter variability, which is an expected trend for sub-45-nm IC, it is unclear whether the quadratic expressions can still provide a desired accuracy.

Recently, Mitev *et al.* [11] focus on reducing the dimension of parameter variability and report 25 sources of uncorrelated variations after applying parameter order reduction to 100 variables. Another recent work for variable dimension reduction is [13], which reports 30 sources of uncorrelated variations after reduction. With the scaling into sub-45-nm era, we expect larger number of uncorrelated variations and need to deal with larger range of variations for each parameter. Existing polynomial-based SSTA approaches need to be revisited in such a case, and the generated polynomial expression will likely look very complex. Even for quadratic forms, it should contain cross-terms. As an example, a quadratic expression with 30 uncorrelated variables has over 400 terms, which is obtained when the cross-terms are not ignored. We also show in our simulation results that the absence of cross-terms results in high modeling error in quadratic forms for 42 number of parameter variations.

To address the aforementioned problems, we propose to use another completely different class of models, which is referred to as GP [25]. Unlike low-order polynomial models, GP can flexibly capture a large class of complex nonlinear relationships [2], [25]. Its generic form is simpler than polynomial models and can be built using a simulation-driven approach, in which a limited number of intelligent samples are collected to capture the input/output behaviors.

The GP model is first applied in computer-aided design (CAD) area as a technique for optimization of analog circuits [2]. Here, we elicit its use in modeling circuit timing with high dimension of parameter variability and postsilicon timing prediction. We also use GP as a component in our proposed adjustment-based modeling framework for SSTA.

III. Overview of GP Model

Given an input vector representing the set of parameter variations $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d\times 1}$, we model the deterministic circuit timing as [25]

$$y(\mathbf{x}) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{f}(\mathbf{x}) + \epsilon(\mathbf{x}) = \sum_{i=0}^{n} \beta_i f_i(\mathbf{x}) + \epsilon(\mathbf{x}) \quad (2)$$

where $\mathbf{f}(\mathbf{x}) \triangleq (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x}))$ represents prespecified functions, typically a constant or a linear expression. The $\epsilon(\mathbf{x})$ is a stationary GP with the correlation structure: $\mathrm{corr}(\epsilon(\mathbf{x}^{(1)}), \epsilon(\mathbf{x}^{(2)})) = R(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$, where $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ denote any two input sample vectors. We assume that $\epsilon(\mathbf{x})$ has mean 0 and variance σ^2.

In (2), the first term $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{f}(\mathbf{x})$ represents the circuit timing obtained using a simple model while the second term $\epsilon(\mathbf{x})$ is to correct the deviations of this simple model using GP. The intuition of GP is that the error of similar variation samples [i.e., $\epsilon(\mathbf{x})$] are highly correlated. Since the similarity of variation samples is defined using criteria $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\| \approx 0$, similar variation samples, in turn, generate highly correlated circuit timings. The correlation function $R(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ is typically modeled in the following form [25]:

$$R\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \prod_{i=1}^{d} \exp\left(-\theta_i \left|x_i^{(1)} - x_i^{(2)}\right|^2\right) \quad (3)$$

where $x_i^{(j)}$ denotes the ith element of $\mathbf{x}^{(j)}$ for $i = 1, 2, \ldots, d$ and $j = 1, 2$. There also exist some other forms of correlation functions for GP models. We refer the readers to [25] for more details. *Please note that the correlation function here is a standard term used in GP modeling, and should not be confused with what is typically used to describe spatial correlations between process variations.*

In this paper, we aim to build a GP model, which is used for the prediction of the circuit timing given new input variation vector. To build this GP model, we need to estimate the unknown parameters $\boldsymbol{\beta}$ in (2) as well as $\boldsymbol{\theta} \triangleq (\theta_1, \ldots, \theta_d)$ in (3)

and σ^2 which defines $\epsilon(\mathbf{x})$. The standard estimation procedure is illustrated as follows.

We first collect N input/output samples. Each sample refers to one input parameters variation vector and its corresponding output observation. We denote the ith sample by the input parameters $\mathbf{x}^{(i)}$ and output observation $y^{(i)}$. All these N output observations $y^{(i)}$ are assumed to be accurate (error-free). We then fit these N samples to the GP model in (2) to find the estimations for $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and σ^2. For convenience of notations, we denote these estimations as $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\theta}}$, and $\widehat{\sigma^2}$, respectively.

To obtain the estimation $\widehat{\boldsymbol{\theta}}$, we maximize the log-likelihood function [25]

$$\max_{\boldsymbol{\theta}} -\frac{1}{2}\left[N\log\widehat{\sigma}^2(\boldsymbol{\theta}) + \log\left(\det\left(\mathbf{R}(\boldsymbol{\theta})\right)\right) + N\right] \tag{4}$$

where

$$\widehat{\sigma}^2(\boldsymbol{\theta}) \triangleq \frac{(\mathbf{y}-\mathbf{F}\widehat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{R}^{-1}(\boldsymbol{\theta})(\mathbf{y}-\mathbf{F}\widehat{\boldsymbol{\beta}})}{N} \tag{5}$$

$\mathbf{R}(\boldsymbol{\theta})$ has its ijth entry $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ as defined in (3), and $\widehat{\boldsymbol{\beta}}$ is defined as

$$\widehat{\boldsymbol{\beta}} \triangleq (\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}\mathbf{R}^{-1}\mathbf{y}. \tag{6}$$

In (5), $\mathbf{y}$ is a column vector with its ith entry as $y^{(i)}$, $\mathbf{F}$ is a matrix with its ijth entry as $f_j(\mathbf{x}^{(i)})$. In other words, $\mathbf{y}$ and $\mathbf{F}$ are representing the actual observation values, and the estimates of f_j for the initially collected samples.

We can efficiently solve the optimization problem in (4) by using the recently released packages, such as maximum likelihood estimation of GP (mlegp) [14]. After obtaining the estimated coefficients (i.e., $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\theta}}$, and $\widehat{\sigma}^2$), we can compute the corresponding circuit timing for any given input variation vector $\mathbf{x}^\star$ using

$$y(\mathbf{x}^\star) = \widehat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{f}(\mathbf{x}^\star) + \widehat{\mathbf{r}}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{F}\widehat{\boldsymbol{\beta}}) \tag{7}$$

where $\widehat{\mathbf{r}}$ is a vector with its ith entry as $R(\mathbf{x}^\star, \mathbf{x}^{(i)})$.

Equation (7) provides an analytical expression for the deterministic circuit timing. Specifically, to evaluate the deterministic circuit timing for an input variation vector $\mathbf{x}^\star$, we need to compute the deviation (second term) using the correlation function $R(\cdot)$ and the initially collected N samples. This deviation term is modeled as a GP here, and the $\widehat{\mathbf{r}}$ as well as $\mathbf{R}$ and $\mathbf{y}$ are functions of the initially collected samples. However, in polynomial models, the deviation term is modeled as an independent process. Therefore, this type of model is inherently different from polynomial models.

A. Complexity of GP Modeling and GP Evaluation

As illustrated in (4), to build a GP model, we need to estimate the coefficients σ^2, $\boldsymbol{\theta}$, and $\boldsymbol{\beta}$. The number of the unknowns is equal to $d+n+1$, where d is the dimension of the variability and n is the number of $f_i(\mathbf{x})$ (n is always very small). On the other hand, if we fit the circuit timing to a quadratic model, the number of unknowns required for estimation will grow very high particularly when d is large. Therefore, the required number of samples for building GP models is intuitively smaller than that for building quadratic model when the dimension of variability is high. In addition, we would like to note that the main difficulty in building GP models lies in solving the optimization problem in (4). Particularly, we can use limited-memory quasi-Newton methods to improve the efficiency in solving this problem. Note that the final estimates will not change with different distributions of $\mathbf{x}$, and can thus be preprocessed.

As illustrated in (7), to predict the circuit timing with different $\mathbf{x}^\star$ only requires the computation of $f(\mathbf{x}^\star)$ and $\widehat{\mathbf{r}}$. All other components such as $\mathbf{R}$, $\mathbf{F}$, and $\widehat{\boldsymbol{\beta}}$ do not change for different $\mathbf{x}^\star$, and thus can be precomputed after GP modeling is complete. Note that the dimension of $\widehat{\mathbf{r}}$ is equal to the number of samples (i.e., N) used in GP modeling. Therefore, decreasing the value of N becomes necessary, and we will discuss one efficient sampling method in the following sections.

B. Handling Special Variation Case in GP Modeling

The main procedures of GP modeling are as follows. Using effective sampling techniques, we first collect initial variation samples which can uniformly cover the variation space. We also collect the corresponding accurate circuit timing for each variation sample. We estimate the coefficients θ_i in (3) to fully define the correlation function $R(\cdot)$ in (3). These coefficients are obtained from solving an optimization problem which maximizes the log-likelihood of the initial samples. Note that the generic intuition in GP modeling is the high correlation in ϵ for "similar" parameter variation samples.

However, in practice, a slight change in parameter variations (e.g., change from $\mathbf{x_1}$ to $\mathbf{x_2}$) can result in a large change in circuit timing. Such behavior can be captured by setting appropriate coefficients θ_i. To well capture such occasions with minimum number of samples, we need good sampling strategy. We discuss Latin Hypercube Design (LHD) sampling as an effective sampling technique for GP modeling in the next section. Furthermore, the concept of adjustment-based model we propose later can help to significantly reduce the number of required samples. We also discuss a dedicated Nested Space-Filling Design sampling technique for adjustment-based modeling later in this paper.

C. Effective Data Collection Using LHD

LHD is one of the most effective sampling methods, which allows significant reduction in the number of samples but can still capture the behavior in (2) with high accuracy [15]. Here, we briefly explain LHD using a simple example for two parameter variations x_1 and x_2. We refer the readers for the extension of LHD to higher dimension and its sample generation procedure to [15].

For two parameter variations x_1 and x_2, we uniformly define a square grid based on the number of target samples. We refer to this square grid as a Latin square. Different samples of x_1 and x_2 can thus be mapped to different grids. The LHD aims to generate samples such that there is only one sample in each

row and each column. In this case, the collected samples can therefore represent all the rows and columns. Since each row and a corresponding column is covered by only one sample, the total number of samples are much fewer than the total number of grids.

Recently, LHD has been used in MC-based SSTA [8]. Other sampling techniques such as Quasi-MC have also been explored in the context of MC-based SSTA in an effort to reduce the total number of samples [7], [8]. We will show in this paper that our concept of an adjustment-based model can significantly reduce the number of required samples and can potentially be combined with all other sampling techniques.

IV. Adjustment-Based Modeling Framework

A. Motivation and Overview

Although we can utilize LHD for effective data collection to build GP models, the required number of samples to obtain a desired modeling accuracy might still be high, particularly with the increase in the dimension of the process variations [i.e., d in (2)]. Moreover, as explained in Section III-A, the larger number of samples indicates higher complexity for evaluation of the model given in (7).

In this paper, we propose a novel adjustment-based model to characterize the circuit timing under high-dimension of variability, which can significantly reduce the number of collected samples [17], [24]. The adjustment-based models have recently been proposed in the statistics community [17], [24]. To the best of our knowledge, we are the first to introduce its usage and discuss its benefits in the CAD area for SSTA. Instead of directly modeling the circuit timing with respect to the process variations using GP models, *we aim to build a model that adjusts an approximate circuit timing (which is subject to error) into an accurate one.* By accurate circuit timing, we do not mean the actual timing at postfabrication stage. Instead, it means an acceptable degree of accuracy depending on the design stage and the degree of available information at that stage. By approximate circuit timing, we mean a circuit timing that is subject to error relative to the accurate circuit timing. Particularly in this paper, we denote the accurate timing as that obtained by MC simulation.

In our proposed adjustment-based modeling framework, apart from generating accurate circuit timing samples, we also need to generate samples for approximate circuit timing which are subject to error compared to the accurate reference samples. These approximate samples can be obtained, for example, based on relaxing assumptions/models and can be generated much faster than the accurate ones, as we further illustrate. Moreover, even though the approximate samples are generated based on relaxed assumptions, they can still capture some nonlinear relationship between the input parameter variations and the actual output circuit timing.

Given the aforementioned observations, building a model that adjusts an approximate circuit timing into an accurate one becomes a simpler task, requiring much fewer number of samples compared to the direct GP modeling. Therefore, it can be more suitable for modeling in presence of high dimension of variations and as we illustrate, it can perform much better than direct polynomial modeling.

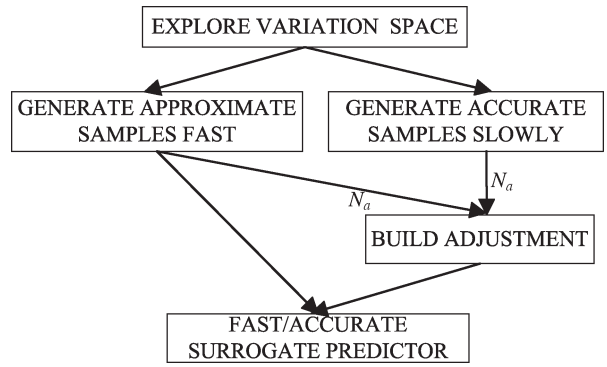


Fig. 1. Diagram of the adjustment-based modeling framework.

In this section, we elaborate the aforementioned statements in detail and discuss the proposed adjustment-based modeling framework. Fig. 1 shows the flow of our adjustment-based modeling framework which is outlined below.

1) We first conduct fast/approximate simulations together with slow/accurate simulation for only a few samples (denoted by N_{a}). We require the same set of input variation samples to find the mapping between the approximate circuit timing and its corresponding accurate value.
2) Using these samples, we then determine an adjustment function which can do the translation between the accurate and approximate simulations.
3) Once the adjustment function is built, we use it in the following way to obtain accurate circuit timing: 1) generate input samples and their approximate circuit timings which are subject to error efficiently and 2) apply adjustment to obtain accurate circuit timings.

The following principle summarizes the main intuition behind our adjustment-based modeling framework.

Principle of Economy: We can use a small number of slow yet accurate samples (i.e., N_{a}) to find the adjustment function, which can quickly and economically translate the results of an approximate simulation into accurate ones.

In order to economically build the adjustment-based model between the accurate and approximate simulations with a small number of samples, the following principles should be satisfied [17], [24].

1) **Principle of Accuracy**: The approximate and accurate simulations should generate different output values for the same input sample. However, the approximate simulation should still reflect the relationship between the input samples and their corresponding accurate outputs. This reflection is subject to error but nevertheless is sufficient to capture the complexity between the input and outputs to some extent.
2) **Principle of Uniformity**: The input samples for the approximate and accurate simulations should uniformly cover the entire input space.

Remark: In the aforementioned statement, by accuracy, we do not mean zero error compared to the actual values. Instead,

we mean an acceptable error, which we can tolerate or is determined based on the stage in the hierarchical design flow.

Next, we review the details of our proposed adjustment-based modeling framework, which is based on GP.

B. Adjustment-Based Modeling Process

In order to build the adjustment-based model, we first generate a small number (i.e., N_{a}) of input samples, and obtain their corresponding approximate and accurate observations. For the convenience of notations, we denote the approximate and accurate observations by $y_I(\mathbf{x})$ and $y_A(\mathbf{x})$ for the input sample $\mathbf{x}$, respectively. Then, we utilize the GP modeling procedures introduced in Section III to build the **adjustment function** between y_I and y_A

$$y_A(\mathbf{x}) = \rho(\mathbf{x}) y_I(\mathbf{x}) + \delta_0 + \delta(\mathbf{x}) \tag{8}$$

where

$$\rho(\mathbf{x}) = \rho_0 + \sum_{i=1}^{d} \rho_i x_i \tag{9}$$

x_i is the ith element of $\mathbf{x} \in \mathbb{R}^{d\times 1}$ for $i = 1, 2, \ldots, d$, and $\delta(\cdot)$ is assumed to be a GP with variance σ_δ^2 and $\boldsymbol{\theta}_\delta$, as defined in Section III. Note that the parameters σ^2, $\boldsymbol{\theta}_\delta$, ρ_i, and δ_0 in (8) and (9) all need to be estimated.

Equation (8) resembles the extended GP model in (2) as follow:

$$\begin{cases} f_i(\mathbf{x}) = y_I(\mathbf{x}), & \beta_i = \rho_0, & \text{when } i = 0 \\ f_i(\mathbf{x}) = y_I(\mathbf{x}) \cdot x_i, & \beta_i = \rho_i, & \text{when } i = 1, 2, \ldots, d \\ f_i(\mathbf{x}) = 1, & \beta_i = \delta_0, & \text{when } i = d+1. \end{cases}$$

Consequently, we can similarly obtain the unknowns (i.e., σ^2, $\boldsymbol{\theta}_\delta$, ρ_i, δ_0) by following the maximum likelihood estimation procedures for GP using the mlegp package [14]. We then build the final adjustment model and accurately compute the output performance for any new input vector using (7).

Remark: We use GP to model the relationship between the approximate and accurate circuit timing as illustrated in (8). However, the idea of adjustment-based modeling is not restricted to GP modeling. In some cases, we can use polynomial model to capture their relationship. The expression in (8) might become simpler, but we may need more samples to build this model.

C. Justification

When all ρ_is in (8) are equal to zero, (8) is equivalent to the original GP form in (2), where $f_i(\cdot)$ is a constant. Therefore, we can expect not to loose accuracy by applying our adjustment-based model compared to building GP model based on accurate simulation. We intuitively justify the expected decrease in the number of samples as follows.

1) The approximate circuit timing has already captured many information about the input parameters. For example, complex and nonlinear dependencies of the output in terms of the inputs can still be captured using an approximate circuit timing—with less accuracy.
2) Both the approximate and accurate models are required to capture the output trends in a similar fashion.

For further theoretical details, we refer the reader to [17] and [24].

As illustrated in Section III-A, the evaluation of GP model depends on the number of samples during the modeling process. Due to the reasons discussed above, we expect a decrease in the number of samples in adjusting the approximate circuit timings to accurate ones. As shown in our simulation results, we can reduce the number of samples by 33% using our proposed adjustment-based model compared to directly GP modeling. We also show that our adjustment-based model with much smaller number of samples can achieve higher accuracy than quadratic model (which requires 50 times more samples).

V. Efficient Nested Space-Filling Design

In Section IV, we have explained how we can efficiently adjust approximate circuit timing into accurate ones by building an adjustment model. Once this model is built, we can efficiently obtain a large number of accurate circuit timing using this adjustment-based framework. We designate this large number with N_{i}. We expect $N_{\mathrm{i}} \gg N_{\mathrm{a}}$, where N_{a} is a small number of samples used for adjustment-based modeling.

In practice, if we know the required value of N_{i}, we can use a customized sampling scheme to draw the small N_{a} subset from the large N_{i} set so that we can further improve the accuracy of the adjustment function. As illustrated in Principle of Uniformity in Section IV-A, we require that the input samples for both approximate and accurate simulations to uniformly cover the entire input space. If we further impose that the input samples for accurate simulations are a subset of those for approximate simulations, we can use nested space-filling design to improve the sampling performance.

The idea of nested space-filling design is to draw the N_{a} subset such that they are the best representatives of the N_{i} set. In this paper, we would like to introduce a simple form of nested space-filling design method, which is quantified through minimizing the maximum distance between the drawn N_{a} subset from the N_{i} set. This methodology is used in [24] and is briefly discussed below.

First, we use orthogonal array-based LHD to generate N_{i} samples for approximate simulations. Then, we use a simulated annealing algorithm and a min–max distance criterion to identify the N_{a} samples from the generated N_{i} samples. Specifically, we aim to minimize the maximum distance between the sample in approximate simulation and that in accurate simulation. We can use either L_1 or L_2 norm to denote the distance between the input samples. Note that this space-filling design can be considered as a preprocessing step.

In summary, by obtaining the N_{i} inaccurate samples *a priori* and building the adjustment function for the selected N_{a} samples, we can improve the prediction efficiency. However, in this paper, we do not use the nested space-filling design to generate samples for both accurate and approximate simulations. This is

because we are able to afford the approximate simulations by using LHD.

Remark: Qian provides alternatives to efficiently generate samples for approximate and accurate simulations in [24]. We refer the readers who are interested in space-filling designs for more details to [24].

VI. Adjustment-Based Modeling for SSTA

In this section, we propose two adjustment-based models for SSTA. Specifically, in both of these two models, we obtain the accurate circuit timings using MC-based SSTA over the Entire Circuit as our reference, which we refer to as MC-SSTA-EC. We propose two different approaches to generate approximate variation-aware circuit timings, which can capture the nonlinear dependence of the process variations in an efficient manner (see Principles of Accuracy and Economy discussed in Section IV). They are discussed as follows.

A. Approximate Circuit Timing Using MC-Based SSTA on Critical Subcircuit

The first approach to obtain approximate circuit timing is to conduct MC simulation, not over the entire circuit, but just a Critical Subcircuit, which we refer to as MC-SSTA-CS. In presence of the process variations, the critical path will not be a constant path. The identification of the critical paths under device and interconnect parameter variations is still an open problem, which is getting investigated (such as recent paper [4], [23]) and is beyond the scope of this paper.

Here, we extract the top-most critical paths of the circuit in the nominal variation case. We traverse the circuit topologically from the primary outputs (POs) toward the primary inputs (PIs). At each logic gate, we consider the arrival times of all its fan-ins. Their arrival times are computed under the nominal case. The fan-in with maximum arrival time are chosen to be part of the critical subtree. If the arrival times of other fan-ins are close to this maximum arrival time by a given threshold of $t\%$, we also put these fan-ins into this critical subtree. The steps of MC-SSTA-CS is listed below.

1) Select a set of critical PO nodes (i.e., $\eta\%$ of total PO nodes) under the average variation case (or alternatively when ignoring the standardized variations).
2) For each selected PO, select a corresponding critical subtree.
3) Apply MC-based SSTA for the extracted critical subcircuit and record these as the observations of the approximate simulation.

According to Principle of Accuracy, the circuit delays obtained from MC-SSTA-CS should reflect the relationship between the parameter variations and the accurate circuit timing. Moreover, according to Principle of Economy, we need the delay values obtained from the extracted critical subcircuit to be evaluated much faster than the entire circuit.

To illustrate Principle of Accuracy, we select S35932 as an example. Fig. 2 shows the similarity between the circuit timings obtained by using MC-SSTA-CS (i.e., Y-axis) and MC-SSTA-EC (i.e., X-axis). These samples are generated for 42 parameter variation cases as described in Section VIII. We also plot the line $y = x$ to indicate the similarity between these two sets of circuit timing. As can be seen, in some variation sample cases, the approximate circuit timing is the same as the accurate one (i.e., the circuit timing is indeed determined by the extracted critical subcircuit). However, in some variation sample cases, the critical subcircuit does not determine the circuit timing, but they still exhibit a correct pattern of increase or decrease in the circuit timing. This is likely due to the common (global) source of parameter variations which result in identical increase or decrease in the parameters of all the gates and interconnects for each sample of variations. Furthermore, different gate types still have similar range of sensitivities to different parameter variations (e.g., sensitivity to L_{eff} in general is higher than to t_{ox}). For further illustration, we also plot the best linear fit to these points (in red) which as can be seen is also subject to error. We aim to use our adjustment-based model to adjust each approximate circuit delay to actual ones.

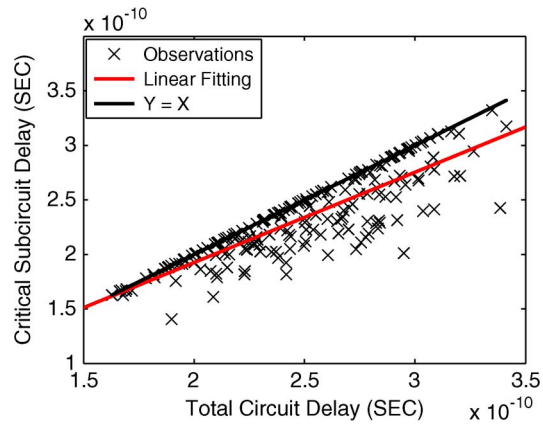


Fig. 2. Similarity between the approximate and accurate simulations obtained using critical subcircuit extraction.

To illustrate Principle of Economy, note that the size of the selected subcircuit is controlled using the threshold parameter t and selected PO nodes η. These parameters can be tuned to make sure the selected subcircuit is sufficiently small to gain acceptable speedup compared to accurate simulation.

B. Approximate Circuit Timing Using Linear-Gaussian SSTA

The second approach to obtain approximate circuit timing that follows Principles of Accuracy and Economy is to use a fast and erroneous SSTA such as the linear Gaussian approximation based on Clark's method [18]. We denote this technique by LG-SSTA. In LG-SSTA, the arrival time at each gate is always approximated to have a Gaussian distribution. These arrival times, along with the individual gate delays are expressed as a linear combination of Gaussian RVs representing different device and interconnect parameter variations. Particularly, in computing the arrival times in the circuit, two atomic operations, namely, summation and maximization should be computed. The summation operation can be exactly computed by adding two linear expressions. However, the maximum of the Gaussian arrival times is approximated as a Gaussian arrival time and then can be represented in the same linear form. This approximation is

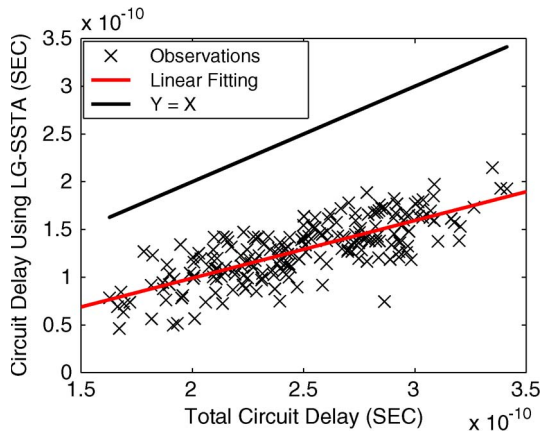


Fig. 3. Similarity between the approximate and accurate simulations obtained using LG-SSTA.

based on Clark's approach [10]. Briefly explaining, given two Gaussian arrival times X and Y, we define Z as the maximum of X and Y, and approximate it as

$$Z \triangleq \max(X, Y) \approx tX + (1-t)Y + \Delta \tag{10}$$

where t is the tightness probability defined as

$$t \triangleq \int_{-\infty}^{\lambda} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \tag{11}$$

and Δ is an additional RV to ensure that this approximation has the same mean and covariance as those of Z. In addition, in (11), λ is defined as (μ_{X-Y}/σ_{X-Y}), where μ_{X-Y} and σ_{X-Y} denote the mean and standard deviation of $X-Y$, respectively. The Gaussian approximation in (10) makes MAX operation easy to implement, but suffers from high error. The authors in [21] pointed that the Clark's approximation in LG-SSTA introduces high error when X and Y have very similar mean but very different variances, and when X and Y have very similar mean but highly negative correlation. Moreover, LG-SSTA is based on moment matching and can only guarantee correctness of statistical information such as mean and covariance at each maximization, but it cannot capture the actual value of circuit timing for any given variation sample.

Despite the intrinsic problems in LG-SSTA illustrated above, we can still use LG-SSTA as one approach to obtain approximate circuit timings. First, LG-SSTA is the fastest known SSTA with linear complexity in terms of the number of gates/edges in the circuit; for each node with multiple fan-ins, it only needs to approximate the mean and variance of the maximum of their arrival times. Therefore, by using LG-SSTA, we can satisfy Principle of Economy. Second, the LG-SSTA is subject to error; for example, the mean of the circuit timing obtained from LG-SSTA is proven to be an underestimation of the exact mean. However, LG-SSTA can still capture the output/input relationship.

Take Fig. 3 as an example. We plot the circuit timings obtained by using LG-SSTA in the Y-axis and those by using MC-SSTA-EC in the X-axis. For LG-SSTA, we obtain the samples by implementing MC simulation on the linear Gaussian expression of circuit timing obtained from LG-SSTA. As shown in this figure, even though that the circuit timings using LG-SSTA suffer from high error, the samples for the first-order approximation can still reflect the output/input relationship.

C. Comparison Between the Two Approximate Simulations

In MC-SSTA-CS, the source of approximation is from the removal of remaining paths in the circuit. These paths might dominate timing for certain variation scenarios which we cannot capture. In other words, we underestimate the influence of other parameter variations which might cause new paths in the circuit to become critical. Nevertheless, we can capture some of the variation space, using our selected paths. In terms of global and local variations, we can capture all the global variations but only some of local variations. *The modeling is supposed to take care of capturing the impact of the uncovered part of the variation space (i.e., other parts of local variations and random variations).* This is achieved because our initially collected samples include approximate and accurate circuit timing observations. The accurate observations capture all local, global, and random variability. Therefore, we have data for all parameter variations during the process of building an adjustment model.

In LG-SSTA, we match the mean and covariance after every MAX operation. However, we do not consider any nonlinearity which will be added after each MAX operation. Unlike the first approach, here, we capture dependence on all type of parameter variations but the source of inaccuracy in lack of nonlinearity in the approximate model. To capture the lack of nonlinearity in the model, instead of adding complicated higher order terms, we used GP modeling which is a more generic modeling approach to capture nonlinearity. The number of samples is much less compared to the higher order polynomial models, which will be particularly effective when adjusting the modeling of LG-SSTA.

VII. Application in Postsilicon Timing Prediction

Apart from SSTA with high dimension of variability, we can also apply adjustment-based model in postsilicon timing prediction.

Current research works focus on developing test structures to measure the exact path delays at postsilicon stage. Wang *et al.* [5] add test structures to the desired paths. However, these test structures can only be used to measure the delay of selected paths at the postsilicon stage, instead of the entire circuit. On the other hand, Liu and Sapatnekar [6] use the on-chip test structures to predict the timing distribution at postsilicon stage. Bastani *et al.* [1] propose an approach to measure the delay of each chip by gradually increasing the frequency until the failure is observed. Such approaches might turn expensive to apply per chip basis.

Our proposed adjustment-based model can overcome the aforementioned problems and predict the postsilicon timing with low test cost. Specifically, we first use the approach in [5] to quickly and accurately measure the delays of selected critical paths at postsilicon stage for each chip. For a few selected chips, we apply the approach in [1] to measure the entire

circuit delay. We then build an adjustment function between the delay estimated by the paths and the circuit delay. Afterward, for each test chip, we can quickly predict the circuit delays by measuring the selected path delays (provided by the test structures) and applying our adjustment function to predict the actual circuit timing of this test chip. Our proposed adjustment-based modeling framework is particularly helpful for efficient postsilicon timing prediction as well as speed binning.

VIII. Simulation Results

A. Simulation Configurations and Tools

We synthesize large ISCAS89 and ITC99 benchmarks using 90-nm TSMC library and Synopsys Design Compiler. The synthesis is fairly conducted for minimum area under a timing constraint to ensure that the design is optimized and many paths are critical after optimization.

We assume process variations in effective channel length (i.e., L_{eff}) and zero-bias threshold voltage (i.e., V_{t}). We use the model in [13] to capture the spatial correlation between these parameter variations. Specifically, we divide the chip into four regions. This is level 1 and introduces four independent L_{eff} and four V_{t} variables. We further divided each region into four subregions. This is level 2 in which 32 additional L_{eff} and V_{t} variables are introduced. We also have a global L_{eff} and V_{t} variable at level 0. Overall, we have 21 RVs to represent V_{t} and 21 to represent L_{eff}. We assume that these 42 RVs are Gaussian distributed and are independent from each other. We also assume a standard deviation of 7% of their mean for each RV. Based on the placement of each gate, we first model L_{eff} of each gate as a linear function of three L_{eff}s in levels 0, 1, and 2. We similarly capture the dependence of gate delay with respect to V_{t} variables. Then, we build variation-aware delay models for each logic gate based on the aforementioned assumptions.

To build our proposed adjustment model with respect to the 42 RVs, we implement MC-SSTA-EC for accurate circuit timing simulation, MC-SSTA-CS (critical subcircuit) and LG-SSTA (linear Gaussian) for approximate simulations. These two sets of simulations are conducted using C++ which is run on a Linux machine. We use Matlab to generate intelligent samples for these RVs, which were utilized as the inputs of the aforementioned simulations. We will mention the functions used within Matlab for sample generation in the future sections. We use "R," an academic environment for statistical computing [12], to build the circuit timing model with respect to process variations.

To show the improvement of our proposed adjustment-based modeling framework, we implement three types of SSTA approaches, which are based on quadratic (QR) model, GP model, and adjustment-based (ADJ) model, respectively. Specifically, in all the aforementioned SSTA approaches, we first generate a small number of observation samples for the 42 RVs and their circuit timings. We then fit these observation samples to obtain the corresponding expression of the circuit timing in terms of these 42 RVs. Finally, we generate a large number of samples for the RVs, plug the values into the circuit timing expression, and then finally build the probability density function (pdf) for the circuit timing.

In total, we consider six variations of these three types of SSTA, which differ in the number of samples and the sampling scheme. They are listed as follows.

1) Approach 1): QR modelWe generate samples for the 42 RVs and obtain their accurate circuit delays using MC-SSTA-EC. We consider a quadratic form without cross-terms, which is an assumption similar to [19]. We fit these observation samples to the quadratic form using "R."[1] Based on the sampling scheme, we consider two variations of QR-based SSTA. They are given as below.
   a) QR with uniform sampling: We generate 10 000 uniform samples for the process variations using "rand" in Matlab and obtain their accurate circuit delays using MC-SSTA-EC. We build quadratic model for the circuit delay and denote the corresponding SSTA approach by *QR-UNI-10000*.
   b) QR with LHD sampling: We generate 2000 LHD samples for the process variations using "lhsdesign" in Matlab and obtain their accurate circuit delays using MC-SSTA-EC. We build the quadratic model for the circuit delay and denote the corresponding SSTA approach by *QR-LHD-2000*.
2) Approach 2): GP modelWe generate 200 and 300 LHD samples for the process variations using "lhsdesign" in Matlab and build GP models as in Section III, to characterize the circuit delay in terms of the process variations. In particular, we use the mlegp package in "R" to build the GP model. We denote the corresponding SSTA approaches by *GP-LHD-200* and *GP-LHD-300*, respectively.
3) Approach 3): ADJ modelWe use two approaches to generate approximate circuit timings, which are MC-SSTA-CS and LG-SSTA. We use MC-SSTA-EC to generate accurate circuit timings. Correspondingly, we can have two adjustment-based models for SSTA. They are *ADJ-CS-LHD-200* and *ADJ-LG-LHD-200*, which will be illustrated in the following sections.

B. Adjustment Model Using Critical Subcircuit Extraction

In this section, we build adjustment model based on critical subcircuit extraction. We first generate 200 LHD samples for process variations and implement MC-SSTA-EC and MC-SSTA-CS for their circuit delays. We consider these obtained circuit delays as the accurate and approximate simulations, respectively. Following the procedures in Section IV, we build an adjustment model between the approximate and accurate simulations using the mlegp package of "R" [14]. Since this adjustment model is based on critical subcircuit extraction, we denote this approach by ADJ-CS-LHD-200.

To measure and compare the accuracy among different SSTA approaches, we take MC-SSTA-EC as the reference for accuracy. We consider 10 000 test samples assuming that the

[1]For 42 RVs, we need to determine 85 coefficients. Note that the standard form of the quadratic fitting for 42 RVs has several more coefficients, which requires a large number of samples for modeling.

TABLE I
ERROR IN PERCENTILE OF TIMING CDF AND THE RELATIVE ERROR IN TIMING SAMPLES COMPARED TO MC-EC-SSTA

	Number of Gates		Error in Percentiles of Timing CDF (%)						Relative Error ε (%)	Speedup
BENCH	Circuit	Subcircuit	0%	5%	25%	50%	75%	95%	$\frac{\sum\|\varepsilon\|}{10{,}000}$	$\frac{MC-SSTA-EC}{ADJ-CS-LHD-200}$
S35932	11373	484	-2.281	-1.554	-0.541	-1.159	-2.182	-3.795	4.42	23.50X
S38417	10037	70	0.166	0.335	-0.567	-0.609	-0.551	-0.394	0.33	143.39X
S38584	8738	29	-0.047	-0.101	-0.207	-0.234	-0.283	-0.311	0.53	301.31X
b18	81122	6922	0.145	-1.504	-1.300	-1.167	-1.154	-1.136	1.69	11.72X
b20	24787	647	0.556	2.000	2.221	2.573	2.611	2.273	2.68	38.31X
b21	21475	344	-0.108	1.049	1.149	1.186	1.202	1.093	1.71	62.43X
b22	34755	1099	-0.410	-0.346	0.232	0.258	0.621	-2.289	2.39	31.62X
Ave $\|Err\|$			0.530	0.984	0.888	1.026	1.229	1.678	1.96	87.47X

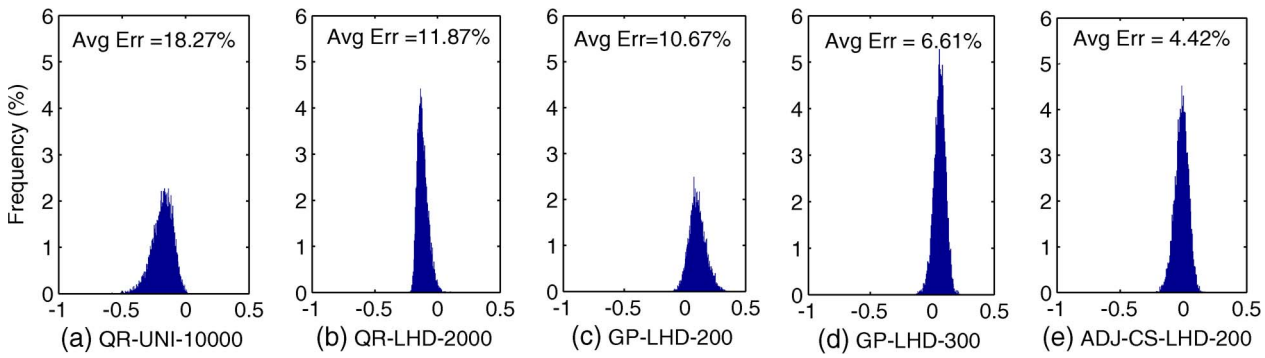


Fig. 4. Histograms of relative error for S35932. (a) and (b) Quadratic models. (c) and (d) GP models. (e) Adjustment-based model.

parameter variations have Gaussian distribution[2] with mean 0 and variance 1. These 10 000 samples are sufficient to build the pdf of the circuit timing, and we observe that the increase in the number of samples will not provide much higher accuracy. For each test sample, we obtain its circuit delays using the five SSTA approaches previously mentioned (i.e., QR models, GP models, and ADJ-CS-LHD-200 model) as well as MC-SSTA-EC, and build their pdfs of the circuit timing.

To demonstrate the quality of the adjustment-based modeling framework based on critical subcircuit extraction, Table I reports the comparisons between ADJ-CS-LHD-200 and MC-SSTA-EC. Columns 4–9 show the error in the cumulative density function (cdf) of the circuit timing at different percentiles. The average of all percentile errors is smaller than 2%. Note that in only three table entries, the percentile error is larger than 2.5%.

Column 10 in Table I reports the mean of the absolute *relative error* (denoted by $|\varepsilon|$) for those 10 000 test samples. The ε is defined as the difference between the circuit delays obtained using ADJ-CS-LHD and MC-SSTA-EC for each test sample. As shown in this table, the average absolute relative error was 1.96% over all benchmarks. *Please note that the majority of the existing SSTA works do not directly compare this error. The error of the circuit timing cdf percentile is generally smaller than the error from direct comparison of the circuit timing samples or of the circuit timing pdf.*

Columns 2 and 3 in Table I give the size of the critical subcircuit for various benchmarks. The selected subcircuit is of very small size; hence, each round of timing analysis is done fast and approximately proportional to the number of selected nodes. Column 11 shows the speedup of ADJ-CS-LHD-200 compared to MC-SSTA-EC is 87.47X on average.

To show the quality and efficiency (in terms of number of samples) of ADJ-CS-LHD-200 over all other SSTA approaches, we focus on the interesting case of balanced benchmarks. For unbalanced circuits, MC-SSTA-EC and MC-SSTA-CS achieve nearly the same performance. This is because the critical subcircuit under nominal case is nearly the same as any of the variation cases. The paths which are not included in the critical subcircuit will rarely become critical under any variation case. Since we use linear delay models for all the logic gates, the delays of the dominated paths are also a linear function with respect to the circuit delays. Therefore, all SSTA approaches including quadratic model-based SSTA are highly accurate for those unbalanced circuits. Specifically, if there is only one critical path, the circuit timing will be a linear function of the process variations. However, for balanced circuits, all paths can be critical, which introduces error in MC-SSTA-CS. Therefore, we only focus on the balanced circuits S35932 and b18.

1) Discussion for Balanced S35932: We make comparisons among five SSTA approaches (i.e., QR-UNI-10000, QR-LHD-2000, GP-LHD-200, and GP-LHD-300, ADJ-CS-LHD-200). Fig. 4 shows the histograms of the relative errors of these five approaches for S35932. The average *absolute* relative errors of QR-UNI-10000, QR-LHD-2000, GP-LHD-200, GP-LHD-300, and ADJ-CS-LHD-200 are 18.27%, 11.87%, 10.67%, 6.61%, and 4.42%, respectively.

We can also make the following conclusions.

1) QR-LHD-2000 is more accurate than QR-UNI-10000 [see Fig. 4(a) and (b)]. Therefore, LHD is very efficient.

[2]For non-Gaussian distributions of variations such as Skew-Normal and triangular, we observed similar trends.

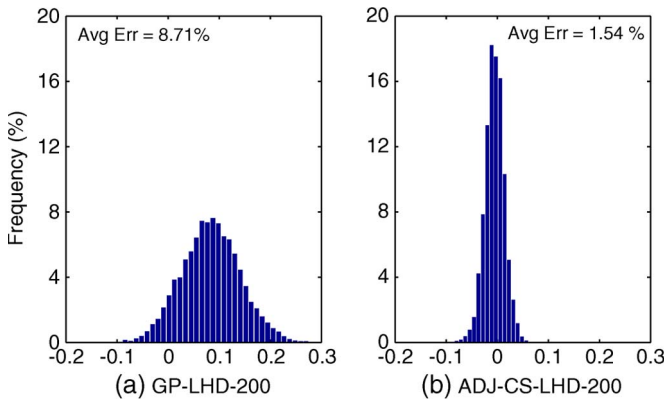


Fig. 5. Histograms of relative error for b18. (a) GP-LHD-200. (b) ADJ-CS-LHD-200.

2) Among all these five SSTA approaches, ADJ-CS-LHD-200 [Fig. 4(e)] has the minimum average absolute error 4.42% and number of samples for building the model is 50× lower than QR-UNI-10000.
3) The histogram of relative error of ADJ-CS-LHD-200 is symmetric around zero while others are skewed.
4) Both GP-LHD-200 and GP-LHD-300 have higher relative errors compared to ADJ-CS-LHD-200. We therefore illustrate better accuracy of ADJ-CS-LHD-200 with the same number of samples.

2) Discussion for Balanced B18: Fig. 5 shows the histogram of the relative error for b18 using GP-LHD-200 and ADJ-CS-200. The average *absolute* relative errors are 8.71% and 1.54%, respectively. Therefore, our proposed adjustment-based model can achieve much lower error compared to GP model. Since the errors achieved using quadratic models are very high, we skip their plots due to lack of space.

In addition, as shown in Fig. 5(b), for some variation samples, the relative error could reach to 9%. However, they are rare cases with very low probability. Finally, the errors of the circuit timing cdf percentiles can become much smaller compared to those relative errors.

C. Adjustment Model Using Linear-Gaussian SSTA

In this section, we implement adjustment-based model based on Linear-Guassian SSTA and 200 LHD samples, which we denote by ADJ-LG-200. To build this model, we first generate 200 LHD samples for the process variations. We obtain their approximate and accurate circuit delays using LG-SSTA and MC-SSTA-EC, respectively, as discussed in Section VI. Similar as ADJ-CS-LHD-200, we build the adjustment-based model between the approximate and accurate circuit timing using mlegp package of "R."

Once the adjustment-based model is built, we predict the circuit delays for the 10 000 test samples of process variations. We implement MC-SSTA-EC for the reference of accuracy. For example, for S35932, ADJ-LG-LHD-200 results in errors of 3.44%, 2.07%, 1.14%, 0.51% at the cdf percentile points of 25%, 50%, 75%, 95%, respectively. In addition, ADJ-LG-LHD-200 achieves average *absolute* relative error (over these 10 000 test samples) of 3.12%, compared to around 50% in LG-SSTA, which is shown in Fig. 3. For all other benchmarks, we also achieve very high accuracy and, thus, skip them due to the lack of space.

D. Discussion on Efficiency

In this section, we discuss the efficiency of our proposed adjustment-based models. *Here, we refer the efficiency mainly to the reduction of the required number of samples to build the circuit timing model.* This referral is reasonable since each MC run on timing analysis is very expensive particularly when the dimension of variability is high and the circuit is very complicated. From the previous simulations, we draw the following conclusions.

1) The simulation runtime of ADJ-CS-LHD-200 is nearly proportional to the size of the extracted subcircuit, which is much lower than the simulation runtime for the entire circuit. The runtime to collect the approximate circuit timings using MC-SSTA-CS is on average about 5% of that of MC-SSTA-EC. Therefore, we can conclude that our proposed ADJ-CS-LHD-200 has runtime a little more than one MC simulation on the entire circuit with only 200 samples. Similarly, the runtime of ADJ-LG-LHD-200 is very fast; this is because LG-SSTA is known to have linear runtime complexity in the number of nodes/edges in the timing graph and is the fastest SSTA approach [18].
2) By solving the nonlinear optimization problem in (4), we can build our adjustment-based model. This problem is well studied in [26] (using L-BFGS Algorithm), and the corresponding software has been developed by both "R" and "Matlab." In our simulation, we utilized "R" to build the adjustment-based model, which is much faster (and negligible) compared to accurate MC SSTA. Note that this model can also be repeatedly and incrementally utilized, for analysis over different pdfs of process variations.

In conclusion, for large circuits with high dimension of variability, the major gain of our approach is on *the significant reduction in the number of samples* required for adjustment-based model, compared to GP and Quadratic models.

IX. Conclusion

We introduce a novel adjustment-based model for SSTA, which translates an approximate variation-aware circuit timing into an accurate one. Particularly, we first propose two approaches to generate approximate variation-aware timing estimates to satisfy the Principles of Economy and Accuracy. We then build the adjustment-based model based on GP which we show is more accurate than low-order polynomial models.

In simulation results, for the more challenging case of balanced circuits, we show that our proposed adjustment-based model can be built using only 200 samples for 42 process variables. An alternative quadratic model requires 10 000 uniform samples and still achieves much larger average error. In terms of accuracy, our proposed adjustment-based model achieves an error of less than 2% on average in various percentiles of the

circuit timing distribution over all the benchmarks. We expect our proposed adjustment-based model to outperform all other models, particularly when the dimension of the variation is high and when we cannot afford the accurate simulations.

Finally, our proposed adjustment model is independent from the distribution of the parameter variations because of the use of sampling techniques which can uniformly cover the variation space. Therefore, it is a robust modeling framework.

ACKNOWLEDGMENT

The authors would like to thank Prof. Z. Qian in the Department of Statistics for guidance, as well as the anonymous reviewers for their insightful comments.

REFERENCES

[1] P. Bastani, K. Killpack, L.-C. Wang, and E. Chiprout, "Speedpath prediction based on learning from a small set of examples," in *Proc. Des. Autom. Conf.*, 2008, pp. 217–222.

[2] M. Bernardo, R. Buck, L. Liu, W. Nazaret, J. Sacks, and W. Welch, "Integrated circuit design optimization using a sequential strategy," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 11, no. 3, pp. 361–372, Mar. 1992.

[3] H. Chang and S. S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1467–1482, Sep. 2005.

[4] L.-C. Wang, J.-J. Liou, and K.-T. Cheng, "Critical path selection for delay fault testing based upon a statistical timing model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 11, pp. 1550–1565, Nov. 2004.

[5] X. Wang, M. Tehranipoor, and R. Datta, "Path-RO: A novel on-chip critical path delay measurement under process variations," in *Proc. Int. Conf. Comput.-Aided Des.*, 2008, pp. 640–646.

[6] Q. Liu and S. S. Sapatnekar, "Confidence scalable post-silicon statistical delay prediction under process variations," in *Proc. Des. Autom. Conf.*, 2007, pp. 497–502.

[7] J. Jafari and M. Anis, "On efficient Monte Carlo-based statistical static timing analysis of digital circuits," in *Proc. Int. Conf. Comput.-Aided Des.*, 2008, pp. 196–203.

[8] V. Veetil, D. Sylvester, and D. Blaauw, "Efficient Monte Carlo based incremental statistical timing analysis," in *Proc. Des. Autom. Conf.*, 2008, pp. 676–681.

[9] L. Xie, A. Davoodi, J. Zhang, and T.-H. Wu, "Adjustment-based modeling for statistical static timing analysis with high dimension of variability," in *Proc. Int. Conf. Comput.-Aided Des.*, 2008, pp. 181–184.

[10] C. E. Clark, "The greatest of a finite set of random variables," *Oper. Res.*, vol. 9, no. 2, pp. 145–162, Mar./Apr. 1961.

[11] A. Mitev, M. Marefat, D. Ma, and J. M. Wang, "Principle Hessian direction based parameter reduction with process variation," in *Proc. Int. Conf. Comput.-Aided Des.*, 2007, pp. 632–637.

[12] *The R Project for Statistical Computing*. [Online]. Available: http://www.r-project.org/

[13] Z. Feng, P. Li, and Y. Zhan, "Fast second-order statistical static timing analysis using parameter dimension reduction," in *Proc. Des. Autom. Conf.*, 2007, pp. 244–249.

[14] *MLEGP: An R Package for Maximum Likelihood Estimates for Gaussian Processes*. [Online]. Available: http://www.public.iastate.edu/~gdancik/mlegp/

[15] E. J. Pebesma and G. B. M. Heuvelink, "Latin hypercube sampling of Gaussian random fields," *Technometrics*, vol. 41, no. 4, pp. 303–312, Nov. 1999.

[16] V. Khandelwal and A. Srivastava, "A general framework for accurate statistical timing analysis considering correlations," in *Proc. Des. Autom. Conf.*, 2005, pp. 89–94.

[17] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. F. Wu, "Building surrogate models based on detailed and approximate simulations," *J. Mech. Des.*, vol. 128, no. 4, pp. 668–677, 2006.

[18] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. Des. Autom. Conf.*, 2004, pp. 331–336.

[19] L. Cheng, J. Xiong, and L. He, "Non-linear statistical static timing analysis for non-Gaussian variation sources," in *Proc. Des. Autom. Conf.*, 2007, pp. 250–255.

[20] J. Singh and S. S. Sapatnekar, "Statistical timing analysis with correlated non-Gaussian parameters using independent component analysis," in *Proc. Des. Autom. Conf.*, 2006, pp. 155–160.

[21] L. Zhang, W. Chen, Y.-H. Hu, J. Gubner, and C. Chen, "Correlation-preserved non-Gaussian statistical timing analysis with quadratic timing model," in *Proc. Des. Autom. Conf.*, 2005, pp. 83–88.

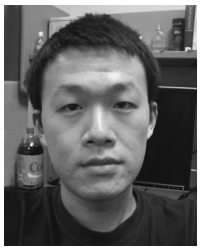
[22] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah, "Parameterized block-based statistical timing analysis with non-Gaussian parameters, nonlinear delay functions," in *Proc. Des. Autom. Conf.*, 2005, pp. 71–76.

[23] H. D. Mogal, H. Qian, S. S. Sapatnekar, and K. Bazargan, "Clustering based pruning for statistical criticality computation under process variations," in *Proc. Int. Conf. Comput.-Aided Des.*, 2007, pp. 340–343.

[24] Z. Qian, "Computer experiments: Design, modeling and integration," Ph.D. dissertation, School Ind. Syst. Eng., Georgia Inst. Technol., Atlanta, GA, 2006.

[25] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. New York: Springer-Verlag, 2003.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer-Verlag, 2006.



Lin Xie (S'07) received the B.S. and M.S. degrees (with honors) in electrical engineering from Zhejiang University, Hangzhou, China, in 2003 and 2006, respectively. He is currently working toward the M.S. degree in operations research and decision science and Ph.D. degree in electrical and computer engineering at the University of Wisconsin, Madison.

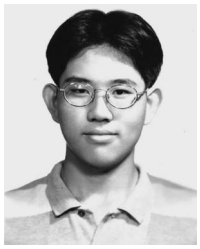
His research interests include statistical modeling and optimization, design automation for very large-scale integrated circuits and systems, test for variability, robust circuit optimization, data mining, and risk analysis.

Azadeh Davoodi (M'06) received the B.S. degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 1999 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 2002 and 2006, respectively.

Since 2006, she has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison. Her research interests are in Design automation in presence of manufacturing-induced variations, parallel optimization on different computing platforms, and power reduction computer-aided design techniques.

Jun Zhang received the B.S. degree in electrical engineering from Shanghai Jiaotong University, Shanghai, China, in 1993, the M.S. degree in electrical engineering from Southeast University, Nanjing, China, in 1996, the M.S. degree in electrical engineering from Iowa State University, Ames, in 1999, and the Ph.D. degree in statistics from the University of Wisconsin, Madison, in 2008.

His research interests include spatial statistics and statistical methods for complex high-dimensional data.



Tai-Hsuan Wu (S'08) received the B.S. degree from the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2002. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Wisconsin, Madison.

His research interests are in the area of computer-aided design and with an emphasis on power aware and high-performance design techniques, parallel optimization, and grid computing environments.