Efficient Representation, Stratification, and Compression of Variational CSM Library Waveforms Using Robust Principle Component Analysis

Safar Hatami and Massoud Pedram
University of Southern California, Department of Electrical Engineering, Los Angeles, California USA
{shatami,pedram}@usc.edu

ABSTRACT - **In deep sub-micron technology, accurate modeling of output waveforms of library cells under different input slew and load capacitance values is crucial for precise timing and noise analysis of VLSI circuits. Construction of a compact and efficient model of such waveforms becomes even more challenging when manufacturing process and environmental variations are considered. This paper introduces a rigorous and robust foundation to mathematically model output waveforms under sources of variability and to compress the library data. The proposed approach is suitable for today's current source model (CSM) based ASIC libraries. It employs an orthonormal transformation to represent the output waveforms as a linear combination of some appropriately-derived basis waveforms. More significantly Robust Principle Component Analysis (RPCA) is used to stratify the library waveforms into a small number of groups for which different sets of principle components are calculated. This stratification results in a very high compression ratio for the variational CSM library while meeting a maximum error tolerance. Interpolation and further compression is obtained by representing the coefficients as signomial functions of various parameters, e.g., input slew, load capacitance, supply voltage, and temperature. We propose a procedure to calculate the coefficients and power of the signomial functions. Experimental results demonstrate the effectiveness of the proposed variational CSM modeling framework and the stratification-based compression approach.**

Keywords- Current Source Model; Robust Principle Component Analysis; Stratification; signomial;

I. INTRODUCTION

As we move towards the 32nm and lower device feature sizes, process variations are becoming an ever increasing concern for the design of high performance integrated circuits [1]. The process variations can cause excessive uncertainty in timing calculation, which in turn calls for sophisticated analysis techniques to reduce the uncertainty. As the number of variation sources increases, corner-based static timing analysis (STA) techniques become computationally very expensive. Moreover, with the decreasing size of transistors and interconnect width, the variation of electrical characteristics of logic cells and on-chip wires is getting proportionally higher.

Delay and slew based library cell modeling methodology is not adequate for the new nanometer-era CMOS technologies any more [1][2]. This is due in part to the highly nonlinear response behavior of devices and interconnections in deep submicron regimes. Current Source Modeling (CSM) has been introduced as an alternative analysis approach for accurate delay modeling in such regimes [3]-[5]. To maintain some compatibility with standard flows and tools, ECSM [6] (proposed by Cadence Design Systems), and CCS [7] (developed by Synopsys) are extensions of the Liberty library format.

CSM modeling needs the storage of tables of current or voltage waveforms. Moreover, timing analysis requires a library characterization with more points in the process, voltage, and temperature (PVT) space so as to handle static and dynamic variations in device and interconnect behavior. A complete set of waveform-based library characterization data for different PVT variables and for timing, noise and power analyses would result in an explosion of the library modeling data.

To address this data explosion problem, a compact variational model waveform was presented in [8], which stores only the nominal waveforms, yet it allows the analysis tools to produce any perturbed waveform by using appropriate time/voltage shift or scale operations. This is a step in the right direction; but unfortunately it does not solve the problem i.e., the proposed method still requires large memory footprint to store the nominal waveforms. The Singular Value Decomposition (SVD) algorithm proposed in [9] solves the latter problem by modeling the voltage waveforms of the logic cells as a linear combination of a fixed set of basis waveforms. Reference [10] addresses adaptive compaction of current-source model libraries by representing each waveform using a variable number of basis waveforms, i.e. using 2-3 basis waveforms to represent a large subset of waveforms and using up to 14 basis waveforms for a small subset of waveforms. The method results in one to two orders of magnitude reduction in the size of the stored characterization data.

The present paper extends [10] to handle *variational* waveforms for statistical static timing analysis (SSTA). Additionally it presents a mathematical foundation to relate the accuracy of compressed data in the reduced space to that in the uncompressed space. In particular it utilizes *robust principle component analysis* (RPCA) to do stratification, robust compression, and outlying data detection. Our other key ideas are: (1) pre-align and then represent the variational output waveforms in a CSM library in terms of the alignment values and the coefficients of the first few principle directions, and (2) use input variable transformation and a combination of signomial hypersurface modeling to construct analytical models of the individual coefficients in terms of input elements which are random variables, input slews and load capacitances. The signomial modeling provides the necessary interpolated CSM waveforms at points not stored in the model tables as needed by timing algorithms. As another contribution, (3), this paper proposes a *library waveform stratification algorithm* which yields high compression ratios. The proposed algorithm is based on a robust (projection pursuit based) PCA. Finally, (4), we use RPCA to detect outlying and intricate waveforms.

II. BACKGROUND

This section gives an overview on the CSM based libraries and then reviews classical PCA.

A. CSM Libraries

Library cell pre-characterization is typically performed as follows. Circuit-level simulations are performed on the CMOS logic cells loaded by a range of capacitance values and excited by voltage ramps with a range of input slew values. ECSM-based characterization uses tables of cell voltage response tables while CCS-based libraries store the characterized data as tables of current as a function of time. In this paper, we focus on the ECSM type data which is more familiar and easier to explain. The concepts presented for variational ECSM waveform modeling can be generalized to handle variational CCS waveforms. The storage of entire waveforms in the CSM methodology as opposed to only the propagation delay and output slew in non-CSM methodologies represents an order of magnitude or more increase in the characterization data volume, and thus, calls for data compression which comes at the expense of some accuracy.

B. Principal Component Analysis (PCA)

The key idea of PCA is to use a linear representation of the given data in a new coordinate system in which the subspace of high-information-content data is easily distinguished from the subspace of low-information-content or redundant data. PCA can be used for dimensionality reduction in a data set by retaining those characteristics of the data that contribute most to its variance. Compression is achieved by keeping lower-order principal components and ignoring higher-order ones. PCA is theoretically the optimum transform for fitting a given data set in the least squares sense.

The input data set for a typical CSM (e.g., ECSM) includes the set of *crossing time vectors* representing the monotonic CSM time-versus-voltage waveforms $W(t)$ for various cells in the library:

$$T = [t_1,...,t_d] \quad (1)$$

Here t_r, $r = 1,\ldots,d$, represents the time instance at which an output waveform crosses the voltage threshold $V_r = W(t_r) = (r - 1) \times \mathrm{V_{DD}} / (d - 1)$. The crossing time vector T is represented by d orthonormal bases P_r and associated coefficients α_r as follows:

$$T = \sum_{r=1}^{d} \alpha_r P_r \quad (2)$$

We denote the vector of coefficients as $A = [\alpha_1,\ldots,\alpha_d]$. The dimensionality reduction is accomplished by keeping only the first m coefficients out of d coefficients. Therefore, in the reduced space $\Re^m$, any original crossing time vector T is approximated with a smaller vector T_a. The approximation error e is calculated as follows:

$$e = \left\| T - T_a \right\| = \left\| \sum_{r=1}^{d} \alpha_r P_r - \sum_{r=1}^{m} \alpha_r P_r \right\| = \sqrt{\sum_{r=m+1}^{d} (\alpha_r)^2} \quad (3)$$

The upper bound on the approximation error, e_m, is given by:

$$e_m = \underset{\text{all}\, T}{Max}\left(\left\| T - T_a \right\|\right) = \underset{\text{all}\, T}{Max} \sqrt{\sum_{r=m+1}^{d} (\alpha_r)^2} \quad (4)$$

Note that although all the presented theorems and algorithms assume monotonic voltage waveforms as the uncompressed input data, the proposed algorithm can be extended to handle the non-monotonic input waveforms. For example reference [10] showed how to develop a PCA-based compression technique for current waveforms which are not monotonic. We do not however discuss non-monotonic waveform handling in the rest of this paper.

III. Variational Waveform Modeling in an Orthonormal Space

An accurate and memory-efficient statistical model of the CSM waveforms is desirable for statistical timing analysis. It is shown below that any PVT induced variation of CSM waveform can be fully accounted for by a combination of *pre-alignment operations* and *orthonormal transformation*. The pre-align operations and orthonormal transformation capture variations of CSM waveform in vertical (along the voltage axis) and horizontal (along the time axis) directions.

We refer to sources of variations (mathematically modeled as random variables) as *input parameters*, $B = [b_1, \ldots, b_N]$, which perturb a CSM waveform from its nominal (typical case) form. For example supply voltage, temperature, L_{eff} and V_{th} are four input parameters. Assume the nominal waveform W is changed to waveform W^P due to a variation of random variable b_i from its nominal value b_i^1 to a perturbed value b_i^* as shown in Fig. 1 (a). We refer the shift and scale operations in the direction of the voltage axis as *V-operators*.

Theorem 1: Let W^P denote a variational waveform of the nominal waveform W. If W^P is monotone function, then W^P can be constructed exactly by using the V-operators and orthonormal transform with d bases, where d is the number of sample points for each waveform W.

All proofs in this paper are omitted due to space limitation.

The shifting operator (the first V-operator) aligns the perturbed waveform W^P to the origin. The scaling operator (the second V-operator) confines the shifted waveform between 0 and $\mathrm{V_{DD}}$. The shifted and scaled waveform W^N can be modeled by orthonormal bases or principal components. Therefore the variational waveform W^P can be modeled by using basis coefficients plus shift and scale values. Note that the basis coefficients for W and W^N are not the same, i.e., any coefficient α_r may undergo a variation because of variation of b_i. Section 5 presents an approach to parameterize the variations of coefficients. Hence the impact of process variation can be captured by utilizing orthonormal transform and V-operators.

The basis coefficients and V-operator values are collectively referred to as the *output features* and denoted by *output vector, G,* as follows:

$$G = [g_1, \ldots, g_{d+2}] = [k, h, \alpha_1, \ldots, \alpha_d] \quad (5)$$

where k and h are V-operator values.

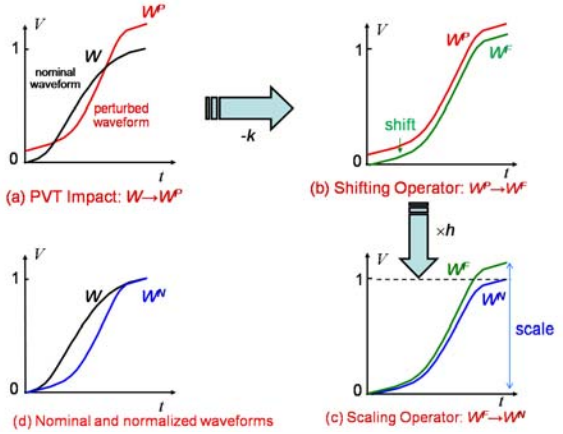


Fig. 1. (a) Impact of PVT variations on the nominal CSM waveform, (b) and (c) An illustration of the V-operators (d) Nominal and normalized waveforms

For example, Fig. 1 (b) shows that the first V-operator aligns W^P to time axis by performing k units vertical shift toward the origin. The scaling operator (the second V-operator) confines the shifted waveform between 0 and $\mathrm{V_{DD}}$. Applying two V-operators to the perturbed waveform W^P one after the other, results in the normalized waveform W^N as shown Fig. 1(d).

It is worth mentioning that the proposed statistical modeling is applicable even when the CSM waveform is characterized based on uniform time steps and constructing basis data set based on the voltage vector.

IV. Robust PCA

Classical principal components analysis (PCA) is very sensitive to outlying data, since it is computed from eigenvectors and eigenvalues of the non robust sample covariance or correlation matrix. A practitioner interpreting multivariate data solely on a classical PCA may therefore end up with wrong conclusions. This fact has been pointed out by many authors and has led to several robustifications of PCA [11].

Outlying data may arise when the library characterizer fails to properly characterize the output waveforms and thereby produces some erroneous data which is hard to detect them among tens of thousands waveforms. The characterization of complex cells such as multiplexers or simple cells under extreme input slew or output capacitance conditions may also give rise to strange waveforms which are another potential source of outlying data.

A robust Projection-Pursuit (PP) based PCA method has been developed by Li and Chen [12]. Like the classical PCA, this method searches for directions with maximal dispersion of the data projected on it. But instead of using the variance as a measure of dispersion, it uses a robust scale estimator S as the projection-pursuit index. In the following section, we introduce two robust scale estimators which are used in robust PP-based PCAs. Robust scale estimators are not unduly affected by small departures from model assumptions. In probability theory and statistics, a scale parameter is a special kind of numerical parameter of a parametric family of probability distributions. The larger the scale parameter, the more spread out the distribution. An estimator of a scale parameter is called a scale estimator. In order to quantify the robustness of a scale estimator, it is necessary to define some measures of robustness. Perhaps the most common of these are the *breakdown point* and the *influence function*, which are explained in the following paragraphs.

Although many robust estimators of *location* exist, the sample median is still the most widely known. If $Y = \{y_1,\ldots, y_b\}$ is a batch of numbers, its sample median is given by:

$$\text{med}_i\, x_i \tag{6}$$

which is simply the middle order statistic when b is odd. The median has a breakdown point of 50% (which is the highest possible), because the estimate remains bounded when fewer than 50% of the data points are replaced by large numbers.

Perhaps the most well-known robust dispersion measure is the Median Absolute Deviation (MAD). For the sample set Y, MAD is defined as follows:

$$\text{MAD}_b\left(y_1,\ldots,y_b\right) = \text{med}_i\left|y_i - \text{med}_j\, y_j\right| \tag{7}$$

Example: Consider the data {1, 1, 2, 2, 4, 5, 9}. It has a median of 2. The absolute deviations about 2 are {1, 1, 0, 0, 2, 3, 7} which in turn have a median value of 1. So the MAD for this data is 1.

The median absolute deviation is a measure of statistical dispersion. It is a more robust estimator of scale than the sample variance or standard deviation. MAD has a 50% breakdown point, but a non-smooth influence function with efficiency 37% at Gaussian distribution. The influence function measures sensitivity of scale estimator when we slightly change the distribution of the data (e.g., one data point in a data sample is changed).

In spite of its high breakdown value, MAD also has some drawbacks. First, its efficiency is very low. Second, the MAD takes a symmetric view on dispersion, because one first estimates a central value (the median) and then attaches equal importance to positive and negative deviation from it.

Another alternative to the MAD is the estimator Q_b of Rousseeuw and Croux [13], which is highly robust, fairly efficient and has an explicit definition since it is the first quartile of the pairwise differences between the data. Computational details about the Q_b estimator may be found in the provided reference.

The idea of RPCA [12] is to search for the direction in which the projected observations have the largest robust scale. In subsequent steps, each new direction is constrained to be orthogonal to all previous directions. Reference [14] presented a computationally attractive method for RPCA.

A. *Outlying Waveform Detection*

As mentioned before, RPCA is robust to the outlying data. Therefore RPCA can be exploited to extract the bizarre (potentially erroneous) and/or more complex waveforms. The bizarre waveforms may be generated because of an imprecise or buggy library characterizer. One of the applications of RPCA transformers is to check the output waveforms generated by library characterization tools for erroneous waveforms.

The RPCA may also be used to identify the situations for which complex waveforms are produced by the library characterizers. For example complex cells such as multiplexers might have more complicated output waveform compared to common cells such as AND and OR cells. More complicated waveforms also can be generated when cells are characterized on extreme situations such as small load capacitances, large input slews or low/high power supply. By categorizing the more complicated waveforms during the compaction process, high rate compressions can be achieved.

B. *Library Waveform Stratification*

As stated above RPCA is a powerful transformer which can result in highly compressed libraries. It is obvious that PCA needs more bases, e.g. 4 to 14, in order to attain compressed libraries with bounded peak errors. In fact some output waveforms need more many bases in order to be accurately represented while others need fewer bases. If a single set of principle components (bases) is used, then we end up using a variable number of bases per output gate, or must simply use a large number of bases for all output waveforms in order to bound the worst-case error of reconstructed waveforms.

We propose a transformation based algorithm for stratification of the output waveforms in a CSM library. The algorithm has a tolerable error E_t and a compression ratio R_d as inputs. The algorithm recursively stratifies the waveforms based on the approximation error. The input parameter R_d gives the maximum number of the bases, N_b, which can be used for the compression. In each epoch i of the stratification, the waveforms are categorized into two classes, the ones which have error less than E_t, L_{i+1}, and the ones which have error more than E_t, H_{i+1}. In each epoch i, H_i is used to generate the bases vectors. For example, H_1 denotes the entire set of output waveforms in the cell library and is used to produce N_b bases vectors in the first epoch.

The stratification algorithm is explained in the following:

1. Get a maximum tolerable error, E_t, for bounding from above the Euclidian relative error of each waveform.
2. Get the desirable compression ratio R_d.
3. Set the whole library as H_1.
4. At epoch i, extract N_b bases vectors from H_i.
5. At each epoch i, stratify H_i into two classes, the ones which have error less than E_t, L_{i+1}, and the ones which have error more than E_t, H_{i+1}.
6. If H_{i+1} is empty, stratification converged else go to step 4.

It can be proved that the PCA stratifications diverge for some E_t and R_d. In contrast, RPCA stratification is able to classify the entire library after a limited number of iterations (epochs).

V. Hypersurface Modeling

To analytically model variational waveforms, we parameterize the entries of output vector, G, as functions of the input parameters. The parameterization is performed over the set of all variables including random PVT parameters, input slew, and output load capacitance. We refer to this complete collection of variables as *input elements*. Parameterization of each entry of G is a hypersurface modeling in a multidimensional space of input elements. We represent the hypersurfaces associated with G by using *signomial models*. A polynomial model is a special case of signomial models.

Any signomial parameterization over a set of characterized or measured data is actually an interpolation over the available data to estimate the uncharacterized or previously unmeasured points. Therefore, before we present the hypersurface modeling of G in a subsequent section, we show that there is a close relationship between the interpolation (surface modeling) in the original space over the crossing time vectors (output waveforms) and interpolation in the new orthonormal space over the output features. In fact it is proved that given some conditions, the interpolation (modeling) in the transformed space is no less accurate than that in the original space. The proofs are given for polynomial models which are special cases of signomial models.

As mentioned earlier, we refer to all random PVT variables, input slew, and output capacitance as the input elements, $Q = [q_1,\ldots,q_M]$. We denote a variational waveform W for a given vector of input elements by the vector pair (Q, W). The vector pair (Q, T) refers to the correspondence between input elements and a crossing time vector T. When only one input element q_i is subject to variation, the vector pair (Q, W) is simply shown with (q_i, T), which means that input elements other than q_i are fixed.

Assume T can be linearly represented by the *orthonormal bases* $P_1,\ldots,P_d$ and the corresponding *coefficient vector* is $A = [\alpha_1,\ldots,\alpha_d]$. The vector pairs ($Q$, A) and (q_i, A) relate A to Q and q_i, respectively. If T is approximated only with the first m bases, then the coefficient vector will be $A_m=[\alpha_1,\ldots,\alpha_m]$ and the pairs are written as (Q, A_m) and (q_i, A_m).

In the remainder of this paper, we shall use subscripts on vectors to denote elements of the vector and superscripts to denote the specific vector corresponding to a particular sample data in the library. For example, $q_i^{\,j}$ shall denote the value of input element q_i of vector Q for the j^{th} data value in the sample set.

A. Theoretical Results

Consider a nonlinear function $u(x)$ for which we are given n+1 sample data points $(x_j, y_j) = (x_j, u(x_j))$, j=1,…,n+1. Let S denote the set of sample pairs. A *one-dimensional n^{th}-order polynomial interpolation function* $y = I^n(x \mid S)$ over n+1 sample points may be defined as follows:

$$y = I^n(x \mid S) = \sum_{j=1}^{n+1} y_j \prod_{k=1,k\neq j}^{n+1} \frac{x - x_k}{x_j - x_k} \tag{8}$$

For simplicity of formulation, suppose the sample points are sorted i.e., $x_j < x_{j+1}$. Since we use (8) only for interpolation and not extrapolation, x is bounded, i.e., $x_1 \le x \le x_n$. The notation $I^n(x^* \mid S)$ refers to the estimated value of $u(x^*)$.

The interpolation error $E(x^*)$ between $I^n(x^* \mid S)$ and the actual function $y^*=u(x^*)$ is given by:

$$E(x^*) = \mid I^n(x^* \mid S) - u(x^*)| \tag{9}$$

In general, the actual value $u(x^*)$ is evaluated by doing circuit simulation of the gate under characterization.

Definition 1: Consider a sample set S_i comprised of pairs (q_i^j,T^j) for j=1,…,n+1. The one dimensional n^{th}-order polynomial vector interpolation function $J^n(q_i \mid S_i)$ over n+1 sample pairs is defined:

$$T = J^n(q_i \mid S_i) = [t_1=I^n(q_i \mid S_{i,1}), ..., t_r=I^n(q_i \mid S_{i,r}), ..., t_d=I^n(q_i \mid S_{i,n})] \tag{10}$$

where $S_{i,r}$ denotes $\{(q_i^j,t_r^j)\}$.

Let $J^n(q_i^* \mid S_i)$ denote the estimated waveform T^* for a given q_i^*. The interpolation error between the estimated value $J^n(q_i^* \mid S_i)$ and *actual* crossing time vector T_{act} is given by:

$$E(T^*) = \| J^n(q_i^* \mid S_i) - T_{act}\|=\|T^* - T_{act}\| \tag{11}$$

Definition 2: Consider a sample set R_i comprised of pairs (q_i^j, A^j) for j=1,…, n+1 where A^j refers to the corresponding coefficient vector for the j^{th} data sample. We use the vector interpolation function $J^n(q_i \mid R_i)$ over n+1 sample pairs (q_i^j, A^j) in the orthonormal space to estimate coefficient vector *A:*

$$A=J^n(q_i \mid R_i) = [\alpha_1=I^n(q_i \mid R_{i,1}), ..., \alpha_r=I^n(q_i \mid R_{i,r}), ..., \alpha_d=I^n(q_i \mid R_{i,n})] \tag{12}$$

where $R_{i,r}$ denotes $\{(q_i^j,\alpha_r^j)\}$.

Let $J^n(q_i^* \mid R_i)$ denote the estimated A^* for a given q_i^*. The corresponding crossing time vector T_{ons}^* for a value q_i^* can be reconstructed by using the estimated coefficient vector A^* from (12) as follows (subscript <u>*ons*</u> stands for *orthonormal space*):

$$T_{ons}^* = \sum_{r=1}^{d}\left(I^n(q_i^* \mid R_{i,r}) \cdot P_r \right) \tag{13}$$

Let T_{act}^* denote the actual crossing time vector corresponding to q_i^*. Interpolation error for orthonormal-space-based estimation is:

$$E(T_{ons}^*) = \|T_{ons}^* - T_{act}^*\| \tag{14}$$

If only the first m bases are used to represent a crossing time vector T in the reduced space $\Re^m$, the estimated crossing time vector $T_{ros}{}^*$ with n^{th} order interpolation is given by:

$$T_{ros}^* = \sum_{r=1}^{m}\left(I^n(q_i^* \mid R_{i,r}) \cdot P_r \right) \tag{15}$$

The corresponding interpolation error for reduced-order-space-based estimation in space $\Re^m$ is:

$$\mathrm{E}(\mathrm{T}_{ros}{}^*) = \|\mathrm{T}_{ros}{}^* - \mathrm{T}_{act}{}^*\| \tag{16}$$

Theorem 2: Consider two sample sets S_i and R_i comprised of pairs (q_i^j, T^j) and their associated pair (q_i^j, A^j), respectively, where j = 1, …, n+1 and $q_i^j < q_i^{j+1}$. Let T^* and T_{ons}^* denote the estimated (interpolated) crossing time vectors for q_i^*, obtained from (10) and (13) in that order. The interpolation accuracy for estimated crossing time vectors T^* and T_{ons}^* are the same, i.e.,

$$\mathrm{E}(\mathrm{T}_{ons}{}^*) = \mathrm{E}(\mathrm{T}^*) \tag{17}$$

This means that the interpolation in the time domain and the orthonormal space are equivalent.

***Theorem* 3**: Assume only the first m out of d coefficients are used to approximate any crossing time vector T_u with T_{ros}. Assume the upper bound on the approximation error in the reduced-order space $\Re^m$ is e_m, which is given by (4). Consider two sample sets S_i and R_i comprised of pairs (q_i^j, T^j) and their associated pair (q_i^j, A^j), respectively, where j = 1, …, n+1 and $q_i^j < q_i^{j+1}$. Let T^* and $T_{ros}{}^*$ denote the estimated (interpolated) crossing time vectors for input element q_i^*, obtained from (10) and (15). An upper bound on the interpolation error $E\,(T_{ros}{}^*)$ is:

$$E(T_{ros}^*) \le E(T^*) + \sqrt{(n+1)H^2 e_m{}^2 + 2(d-m)\binom{n+1}{2} H^2 e_m{}^2} \tag{18}$$

where H is:

$$H = \operatorname*{Max}_{\substack{i=m+1,...,d \\ \& \, j=1,...,n+1 \\ \& \, q_i^1 \le q_i^* \le q_i^{n+1}}} \left(\left| \prod_{k=1,k\neq j}^{n+1} \frac{q_i{}^* - q_i{}^k}{q_i{}^j - q_i{}^k} \right| \right) \tag{19}$$

B. Input Variable Transform and Signomial Model

A signomial function hypersurface modeling technique is presented in this section. Finding an analytical hypersurface model is a challenging task especially when the output features are complex, nonlinear functions of the input elements. For example consider an output feature is inversely proportional to or is a logarithmic function of an input element. In this case, transforming the input element through an inverse or a logarithmic function enables us to model the output feature by a signomial function. We utilize variable transformation [15] which is a useful method, to lessen the complexity of hypersurface modeling.

We introduce our proposed variable transformation procedure with the aid of a simple example. Suppose the statistical output g_i and input element q_1. Assume a well fitted surface model for g_i in terms of q_1 is given by:

$$g_i = \beta_0 + \beta_1 \sqrt{q_1} \tag{20}$$

So we look for the power of q_1, denoted by λ, which is ½, β_0 and β_1. A Taylor approximation of (20) around λ=λ$_0$ is given by:

$$g_i = \beta_0 + \beta_1 (q_1)^{\lambda} \simeq \beta_0 + \beta_1 (q_1)^{\lambda_0} + (\lambda - \lambda_0)\beta_1 (q_1)^{\lambda_0} \ln(q_1) \tag{21}$$

where $\simeq$ stands for Taylor approximation around λ=λ$_0$. In order to estimate λ, β_0 and β_1, first suppose $\lambda=\lambda_0=1$ and use the data points (g_i,q_1) to calculate β_0 and β_1. By using the estimated β_0 and β_1 and $\lambda_0=1$, λ is estimated and subsequently used to calculate new β_0 and β_1 values. This process is continued recursively until we converge on final values of λ, β_0 and β_1.

Variable Transformation Procedure – fixed point iteration:

The procedure is described for a second order signomial although it can be generalized for any other signomial order.

Assume the statistical output g_i is modeled by equation (22).

$$g_i = \beta_0 + \sum_j \beta_j q_j^{\lambda_j} + \sum_{i\ge j}\sum \beta_{ij} q_i^{\lambda_i} q_j^{\lambda_j} \tag{22}$$

First we construct the new variables:

$$Z_j = \frac{\partial g_i}{\partial q_j} q_j \ln(q_j) \tag{23}$$

Then we construct approximated g_i as:

$$g_i = \beta_0 + \sum_j \beta_j q_j^{\lambda_j} + \sum_{i\ge j}\sum \beta_{ij} q_i^{\lambda_i} q_j^{\lambda_j} + \sum_{j=1}^{M} (\lambda_j - 1) Z_j \tag{24}$$

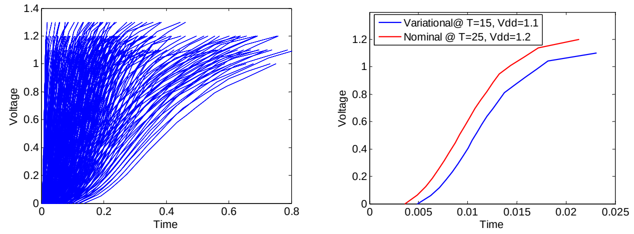
In the first iteration, we calculate β_j and β_{ij} by letting λ_j = 1 and using the characterized data which give $\mathrm{g_i}$ for different sample values of input elements (we construct equations from (24) for different sample points in order to calculate all unknowns β_j and β_{ij}). The calculated β_j and β_{ij} are used in the equations constructed by (24), to give a new guess for λ_j's. The calculated λ_j's are used in the equations to give new estimation of β_j and β_{ij}'s. The process is repeated until the all parameters converge to their final values.

VI. Simulation Results

This section presents different simulation results for ECSM libraries. Time, voltage and current units are nanosecond and volt, respectively. All reported errors are calculated with respect to actual waveforms generated by SPICE simulations. As for the impact of the proposed compression algorithm on the run time of a timing analysis tool that uses the compressed waveforms instead of the original waveform data, the answer depends on the way the tool utilizes the compressed waveforms. For example a timing analysis tool may directly propagate the output vector in the principle component space or it may reconstruct the original waveform and then propagate the reconstructed waveform forward in the circuit. Clearly the runtime savings will be significantly higher for a tool using the first approach.

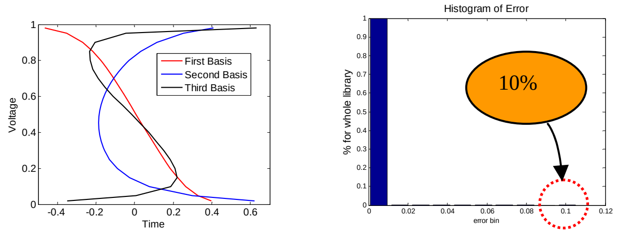
A. Modeling of Variational Waveforms

To verify the accuracy and efficiency of the proposed techniques, experiments are carried out on a sample 65nm industrial-strength ECSM library, characterized at nominal process corner, 1.2 *volt*, and 25^o*C* plus 25 other corners (different voltage and temperature combinations). Each gate is characterized for 7x7x5x5 (input slew, output capacitive load, supply voltage and temperature) combinations and for each of these combinations, a voltage waveform with 21 uniform voltage point increments ({0%, 0.05%, …, 0.95%, 100%} of V$_{DD}$) is stored in the library. Fig. 2(a) depicts the superposition of the 1225 waveforms for an inverter. The pre-alignment operation and orthonormalization steps have been performed on the ECSM library. Fig. 2(b) shows a nominal waveform and its variational waveform.



(a) (b)

Fig. 2. (a)Superposition of variational waveforms for an inverter gate (b) Two waveforms: a variational and the nominal



(a) (b)

Fig. 3. (a) First, second and third basis vectors (b) Histogram of relative error for 43141 voltage waveforms approximated by using five bases

In this experiment, we pre-align a 43141-waveform library and then extract the orthogonal bases. Each waveform is approximated by using five bases. Fig. 3(a) shows the first three bases. The average relative error is 0.04% and the relative maximum error is 10%. Fig. 3(a) shows the relative error histogram. The compression ratio is 76%.

B. Parameterization Using Signomial Model

This subsection shows the parametric compression for slew, capacitance, supply voltage and temperature. The 65nm ECSM library with 2,157 voltage waveforms is utilized for simulations. We apply the *Variable Transformation Procedure* to the output features of 2,157 voltage waveforms and calculate the power of each term of the hypersurface model. An example signomial equation for the third output feature, which is first principle component, in terms of the load capacitance and input slew is:

$$g_3 = \beta_0 + \sum_{j=1}^{2} \beta_j q_j^{\lambda_j} + \sum_{i \ge j} \sum \beta_{ij} q_i^{\lambda_i} q_j^{\lambda_j} \qquad (25)$$

where q_1 and q_2 stand for load capacitance and input slew. The calculated parameter values are reported in Table 1.

Table 1: Coefficients given by *Variable Transformation Procedure*

β_0	β_1	β_2	β_{12}	β_{21}	β_{11}	λ_1	λ_2
-0.02	-2.5	-1.8	-0.01	0.001	0.08	0.97	1.2

Fig. 4 shows the relative error of signomial for calculating the first components of waveforms for all cells, which is less than 4.5%.

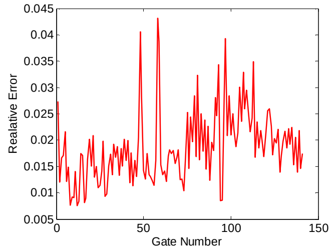


Fig. 4. Relative error of signomial model for calculating the first component of waveforms.

Storing the coefficients of signomial model instead of output features provides a 97% compression ratio with maximum relative error of 3% and average relative error of 1%.

C. Robust PP-PCA

1) Compression

First the compression results of conventional (non robust) PCA and two different RPCA methods are compared. To compare the performance of three different transformations, a library of 43,141 waveforms is used. We produce 20 sub-libraries having 2,157 waveforms from the original library and apply three different transformations. The error measured for each transformation is averaged over the 20 sub-libraries and reported in Fig. 5. Fig. 5(a) compares the maximum relative L^2 norm errors which show that the robust transformations, MAD and Q_b, result in superior accuracy for compression ratio greater than 80%. Fig. 5(b) also shows that the average L^2 norm errors of RPCAs are less than that of the PCA. For example, the RPCA gives a 5-fold reduction of the average error and a 2-fold reduction of maximum error compared to the PCA at a compression ratio 95%.

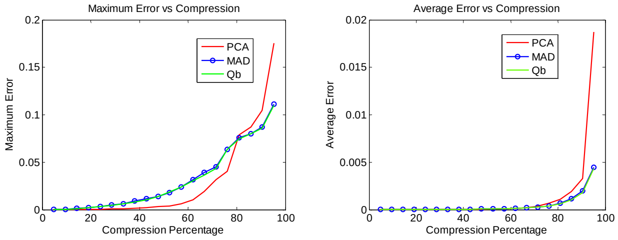


Fig. 5. (a) Maximum error (b) Average error versus compression for PCA and two RPCAs.

2) Outlying Waveform Detection

Another application of PCA is the detection of outlying waveform. After applying the transformation to a library, the relative approximation error can be used as a criterion to identify the outlying waveforms. Two bases are used to approximate the waveforms. We apply PCA and Q_b RPCA to a library of 2,157 waveforms which include the waveform shown in Fig. 6. It is shown that both PCA and RPCA are able to find this waveform as a waveform with maximum approximation error. However the RPCA approximation, blue waveform, has lower deviation from the original waveform which means basis extraction by RPCA has been affected less by the outlying waveforms. In addition, the RPCA transformation approximates all waveforms with an average error 0.5% which is one-third of the average error resulted by the PCA transformation. This means RPCA is more effective for approximating the majority of the waveforms with higher accuracy.

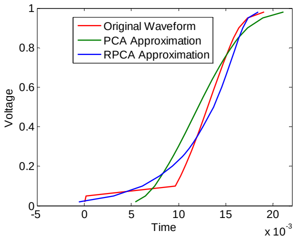
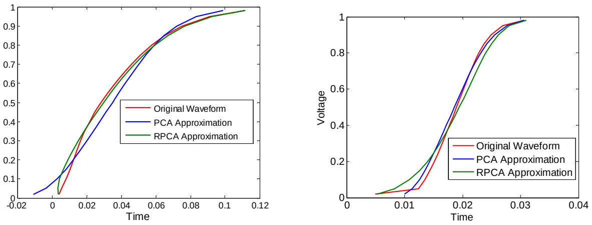


Fig. 6. Outlying waveform detection by using PCA and RPCA.

We applied RPCA and PCA to another library with 2,157 waveforms. As shown in Fig. 7(a) PCA incorrectly classifies a fairly good waveform (subjective judgment) as an outlying waveform whereas RPCA does the classification accurately. As another example, Fig. 7(b) shows that RPCA finds an outlying waveform and approximates/converts it to a normal looking waveform while PCA approximates it incorrectly.



(a) (b)

Fig. 7. Outlying waveform detection by using RPCA and PCA.

3) *Stratification*

We apply three different transformations to a library of 2157 waveforms. We set the compression ratio to 95% and the tolerable maximum error to 4%. It is clear from Table 2 that the PCA cannot stratify the library waveforms while meeting the peak error bound. However the MAD and Q_b RPCAs manage to stratify the library into 4 levels. The PCA stratification did not succeed when it was applied to 20 different libraries. The MAD RPCA is faster than the Q_b RPCA. It is notable that a level ID is assigned to each compressed waveform, which is used for the reconstruction process. This level ID may reduce the compression ratio down to 91%. Fig. 8 shows the stratification tree of the library by using the Q_b RPCA. Each node in this rooted denotes the total number of waveforms in a subset of the library waveforms with the maximum and average percentage errors obtained for that subset. Note that, without waveform stratification, five bases would be needed to attain this level of accuracy which gives a compression ratio of 76%.

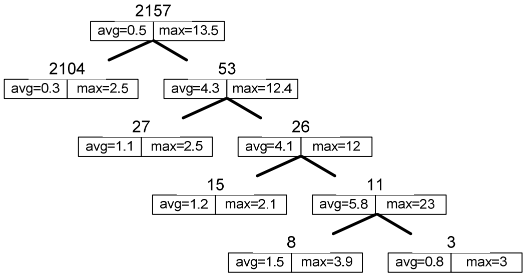


Fig. 8. A sample 4-level stratification tree

Table 2: Stratification using one basis

Transf.	Max Error	Avg. Error	Branch. Level	Compress. Ratio	Time (min)
PCA	–	–	∞	–	∞
RPCA-MAD	4%	0.5%	4	95%	1.5
RPCA- Q_b	3.9%	0.5%	4	95%	6.6

VII. Summary

We described a robust and extensible framework for modeling variational CSM waveforms in terms of a set of statistically varying input parameter set. The key ideas were: (1) represent the CSM data in terms of V-operator values and the coefficients of the first few principle directions in an orthonormal space, and (2) use input variable transformation and a combination of signomial hypersurface modeling to construct analytical models of the individual coefficients in terms of input elements. The proposed variational CSM framework results in significantly compressed CSM library data; at the same time it enables interpolation among given sample data to produce the required CSM data for a previously uncharacterized set of input element values. As another contribution, (3), this paper proposes a *library waveform stratification algorithm* which yields high compression ratios. The proposed algorithm is based on a robust (projection pursuit based) PCA.

References

[1] S. Nassif, "Design for variability in DSM technologies," *Proc. Int'l. Symp. on Quality of Electronic Designs*, 2000.

[2] R. Trihy, "Addressing library creation challenges from recent liberty extensions," *Proc. of Design Automation Conference*, pp 474-479, 2008.

[3] I. Keller, et al., "A robust cell-level crosstalk delay change analysis," *Proc. of the International conference on Computer-aided design*, Nov. 2004.

[4] C. Amin, et al., "A multi-port current source model for multiple-input switching effects in CMOS library cells," *Proc. of Design Automation Conference*, Jun. 2006.

[5] H. Fatemi, S. Nazarian, and M. Pedram, "Statistical logic cell delay analysis using a current-based model," *Proc. of Design Automation Conference*, pp. 253-256, Jun. 2006.

[6] http://www.cadence.com/Alliances/languages/Pages/ecsm.aspxRef3

[7] http://www.synopsys.com/products/solutions/galaxy/ccs/ccs_faq.html

[8] V. Zolotov and J. Xiong and S. Abbaspour, D.J. Hathaway and C. Visweswariah, "Compact modeling of variational waveforms," *Proc. of International Conference on Computer Aided Design*, pp 705-712, 2007.

[9] A. Ramalingam, et al., "Accurate Waveform Modeling using Singular Value Decomposition with Applications to Timing Analysis," *Proc. of Design Automation Conference*, pp 148-153, 2007.

[10] S. Hatami, P. Feldmann, S. Abbaspour, and M. Pedram, "Efficient Compression and Handling of Current Source Model Library Waveforms," *Proc. of Design Automation and Test in Europe*, April 2009.

[11] Ch. Croux, and A. Ruiz-Gazen, "High breakdown estimators for principal components: the projection-pursuit approach revisited," *Journal of Multivariate Analysis*, Vol. 95 , No. 1, pp. 206-226, 2005.

[12] G. Li, Z. Chen, "Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo," *Journal of American Statistical Association*, Vol. 80, No. 391, 759-766, 1985.

[13] P. J. Rousseeuw, et al., "Alternatives to the Median Absolute Deviation", *Journal of the American Statistical Association*, Vol. 88, No. 424, 1993.

[14] M. Hubert , P. J. Rousseeuw and S. Verboven, "A fast method for robust principal components with applications to chemometrics," *Chemometrics and Intelligent Laboratory Systems*, Vol. 60, No. 1-2, pp. 101-111, 2002.

[15] G. E. P. Box and P. W. Tidwell, "Transformation of the independent variables," *Technometrics*, Vol. 4, No. 4, pp 531-550, 1962.