Temperature-Centric Reliability Analysis and Optimization of Electronic Systems under Process Variation

Ivan Ukhov, Petru Eles, *Member, IEEE*, and Zebo Peng, *Senior Member, IEEE*

***Abstract*—Electronic system designs that ignore process variation are unreliable and inefficient. In this work, we propose a system-level framework for the analysis of temperature-induced failures that takes into account the uncertainty due to process variation. As an intermediate step, we also develop a probabilistic technique for dynamic steady-state temperature analysis. Given an electronic system under a certain workload, our framework delivers the corresponding survival function, founded on the basis of well-established reliability models, with a closed-form stochastic parameterization in terms of the quantities that are uncertain at the design stage. The proposed solution is exemplified considering systems with periodic workloads that suffer from the thermal-cycling fatigue. The analysis of this fatigue is a challenging problem as it requires the availability of detailed temperature profiles, which are uncertain due to the variability of process parameters. In order to demonstrate the computational efficiency of our framework, we undertake a design-space exploration procedure to minimize the expected energy consumption under a set of timing, thermal, and reliability constraints.**

***Index Terms*—Process variation, reliability analysis, reliability optimization, temperature analysis, uncertainty quantification.**

I. Introduction

PROCESS VARIATION constitutes one of the major concerns of electronic system designs [1]. A crucial implication of process variation is that it renders the key parameters of a technological process, e.g., the effective channel length, gate oxide thickness, and threshold voltage, as random quantities at the design stage. Therefore, the same workload applied to two "identical" dies can lead to two different power and, thus, temperature profiles since the dissipation of power and heat essentially depends on the aforementioned stochastic parameters. This concern is especially urgent due to the interdependence between the leakage power and temperature [2]. Consequently, process variation leads to performance degradation in the best case and to severe faults or burnt silicon in the worst scenario. Under these circumstances, uncertainty quantification [3] has evolved into an indispensable asset of temperature-aware design workflows in order to provide them with guaranties on the efficiency and robustness of products.

Temperature analysis can be broadly classified into two categories: transient and steady-state. The latter can be further subdivided into static and dynamic. Transient temperature analysis is concerned with studying the thermal behavior of a system as a function of time. Intuitively speaking, the analysis takes a power curve and delivers the corresponding temperature curve. Static steady-state temperature analysis addresses the hypothetical scenario in which the power dissipation is constant, and one is interested in the temperature that the system will attain when it reaches a static steady state. In this case, the analysis takes a single value for power (or a power curve which is immediately averaged out) and outputs the corresponding single value for temperature. Dynamic steady-state (DSS) temperature analysis is a combination of the previous two: it is also targeted at a steady state of the system, but this steady state, referred to as a dynamic steady state, is now a temperature curve rather than a single value. The considered scenario is that the system is exposed to a periodic workload or to such a workload that can be approximated as periodic, and one is interested in the repetitive evolution of temperature over time when the thermal behavior of the system stabilizes and starts exhibiting the same pattern over and over again. Prominent examples here are various multimedia applications. The input to the analysis is a power curve, and the output is the corresponding periodic temperature curve. In the absence of uncertainty, this type of analysis can be efficiently undertaken using the technique developed in [4].

A typical design task, for which temperature analysis is of central importance, is temperature-aware reliability analysis and optimization. The crucial impact of temperature on the lifetime of electronic circuits is well known [5]. Examples of the commonly considered failure mechanisms include electromigration, time-dependent dielectric breakdown, and thermal cycling, which are directly driven by temperature. Among all failure mechanisms, thermal cycling has arguably the most prominent dependence on temperature: not only the average and maximum temperature but also the amplitude and frequency of temperature oscillations have a huge impact on the lifetime of the circuit. In this context, the availability of detailed temperature profiles is essential, which can be delivered by means of either transient or DSS temperature analysis.

Due to the urgent concern originating from process variation, deterministic temperature analysis and, thus, all procedures based on it are no longer a viable option for the designer. The presence of uncertainty has to be addressed in order to pursue efficiency and fail-safeness. In this context, probabilistic techniques are the way to go, which, however, implies a higher level of complexity. This paper builds upon the state-of-the-art techniques for deterministic DSS temperature analysis proposed in [4] and probabilistic transient temperature analysis proposed in [6] and presents a computationally efficient framework for probabilistic DSS temperature analysis and the subsequent reliability analysis and optimization of electronic systems.

The remainder of the paper is organized as follows. Sec. II provides an overview of the prior work. In Sec. III, we summarize the contribution of the present paper. The preliminaries are given in Sec. IV. The objective of our study is formulated in Sec. V. The proposed frameworks for uncertainty, temperature, and reliability analyses are presented in Sec. VI, Sec. VII, and Sec. VIII, respectively. An application of the proposed techniques in the context of reliability optimization is given in Sec. IX. In Sec. X, the experimental results are reported and discussed. Sec. XI concludes the paper. The appendix (Appendix A–E) contains a set of supplementary materials

with discussions on certain aspects of our solutions.

II. Prior Work

In this section, an overview of the literature related to our work is given. First, we discuss those studies that focus on probabilistic temperature analysis, and then we turn to those that focus on temperature-aware reliability analysis.

The most straightforward approach to analyze a stochastic system is Monte Carlo (MC) sampling [3]. The technique is general and has had a tremendous impact since the middle of the twentieth century when it was introduced. The success of MC sampling is due to the ease of implementation, independence of the stochastic dimensionality, and asymptotic behavior of the quantities estimated using this approach. The crucial problem with MC sampling, however, is the low rate of convergence: in order to obtain an additional decimal point of accuracy, one has to draw usually hundred times more samples. Each sample implies a complete realization of the whole system, which renders MC-based methods slow and often infeasible as the needed number of simulations can be extremely large [7].

In order to overcome the limitations of deterministic temperature analysis and, at the same time, to completely eliminate or, at least, to mitigate the costs associated with MC sampling, a number of alternative probabilistic techniques have been introduced. The overwhelming majority of the literature concerned with temperature relies on static steady-state temperature analysis. Examples include the work in [8], which employs stochastic collocation [3] as a means of uncertainty quantification, and the work in [9], which makes use of the linearity property of Gaussian distributions and time-invariant systems. The omnipresent assumption about static temperatures, however, can rarely be justified since power profiles are not invariant in reality. Nevertheless, the other two types of temperature analysis, i.e., transient and DSS, are deprived of attention. Only recently a probabilistic framework for the characterization of transient temperature profiles was introduced in [6]; the framework is based on polynomial chaos expansions [3]. Regarding the DSS case, to the best of our knowledge, it has not been studied yet in the literature from the stochastic perspective. However, as mentioned earlier, the knowledge of DSS variations is of practical importance when designing systems whose workloads tend to be periodic. In particular, the DSS analysis allows the designer to address the thermal-cycling fatigue, which we illustrate in this paper.

Let us now discuss temperature-aware reliability-driven studies. Reliability analysis is probabilistic by nature. Certain components of a reliability model, however, can be treated as either stochastic or deterministic, depending on the phenomena that the model is designed to account for. Temperature is an example of such a component: it can be considered as deterministic if the effect of process variation on temperature is neglected and as stochastic otherwise. The former scenario is the one that is typically addressed in the literature related to reliability. For instance, the reliability modeling framework proposed in [10] has a treatment of process variation, but temperature is included in the model as a deterministic quantity. Likewise, the aging-minimization procedure in [4] assumes temperature to be unaffected by process variation. In [11], a design methodology minimizing the energy consumption and temperature-related wear-outs of multiprocessor systems is introduced; yet neither energy nor temperature is aware of the uncertainty due to process variation. A similar observation can be made with respect to the work in [12] wherein a reinforcement learning algorithm is used to improve the lifetime of multiprocessor systems. An extensive and up-to-date survey on reliability-aware system-level design techniques given in [13] confirms the trend outlined above: the widespread device-level models of failure mechanisms generally ignore the impact of process variation on temperature. However, as motivated in the introduction, deterministic temperature is a strong assumption that can lead to substantial yield losses.

An example of a different kind is the work in [8]: it provides a statistical simulator for reliability analysis under process variations and does consider temperature as a stochastic parameter. However, as discussed previously, this study is bound to static steady-state temperatures, and the presented reliability analysis is essentially an analysis of maximal temperatures without any relation to the typical failure mechanisms [5].

To conclude, the designer's toolbox in our field does not yet include a tool for DSS temperature analysis under process variation, which is of high importance for certain classes of applications mentioned previously. Furthermore, the state-of-the-art reliability models lack a flexible approach for taking the effect of process variation on temperature into consideration. This work eliminates the aforementioned concerns.

III. Our Contributions

Our work brings the following major contributions:

- **Contribution 1.** Based on the stochastic approach to transient temperature analysis presented in [6], we extend the deterministic DSS temperature analysis presented in [4] to account for the uncertainty due to process variation.
- **Contribution 2.** We develop a framework for the reliability analysis of electronic systems that enriches the state-of-the-art reliability models by taking into consideration the effect of process variation on temperature.
- **Contribution 3.** We construct a computationally efficient design-space exploration procedure targeted at the minimization of the energy consumption, which is *a priori* random, under probabilistic constraints on the thermal behavior and lifetime of the system.

IV. Preliminaries

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, where Ω is a set of outcomes, $\mathcal{F} \subseteq 2^\Omega$ is a σ-algebra on Ω, and $\mathbb{P} : \mathcal{F} \to [0,1]$ is a probability measure [14]. A random variable on $(\Omega, \mathcal{F}, \mathbb{P})$ is an $\mathcal{F}$-measurable function $\zeta : \Omega \to \mathbb{R}$. A random variable ζ is uniquely characterized by its (cumulative) distribution function defined by

$$F_\zeta(x) := \mathbb{P}(\{\omega \in \Omega : \zeta(\omega) \leq x\}).$$

The expectation and variance of ζ are given by

$$\mathbb{E}[\zeta] := \int_\Omega \zeta(\omega)\, d\mathbb{P}(\omega) = \int_{\mathbb{R}} x\, dF_\zeta(x) \quad \text{and}$$
$$\mathbb{V}\text{ar}[\zeta] := \mathbb{E}\left[(\zeta - \mathbb{E}[\zeta])^2\right],$$

respectively. A random vector $\boldsymbol{\zeta} = (\zeta_i)$ and matrix $\mathbf{Z} = (\zeta_{ij})$ are a vector and matrix whose elements are random variables.

Denote by $L^2(\Omega, \mathcal{F}, \mathbb{P})$ the Hilbert space of square-integrable random variables [15] defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with the inner product and norm defined, respectively, by

$$\langle \zeta_1, \zeta_2 \rangle := \mathbb{E}[\zeta_1 \zeta_2] \quad \text{and} \quad \|\zeta\| := \langle \zeta, \zeta \rangle^{1/2}.$$

In what follows, all the random variables will tacitly have $(\Omega, \mathcal{F}, \mathbb{P})$ as the underlying probability space.

V. Problem Formulation

Consider a heterogeneous electronic system that consists of n_{p} processing elements and is equipped with a thermal package. The processing elements are the active components of the system that are identified at the system level with a desired level of granularity (e.g., SoCs, CPUs, and ALUs); the components can be subdivided whenever a finer level of modeling is required for the problem at hand.

We shall denote by $\mathcal{S}$ an abstract set containing all information about the system that is relevant to our analysis, and we shall refer to it as the system specification. The content of $\mathcal{S}$ is problem specific, and it will be gradually detailed when it is needed. For now, $\mathcal{S}$ is assumed to include: (a) the floorplan of the chip; (b) the geometry of the thermal package; and (c) the thermal parameters of the materials that the chip and package are made of (e.g., silicon thermal conductivity).

A power profile is defined as a matrix $\mathbf{P} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\mathrm{t}}}$ containing n_{t} samples of the power dissipation of the processing elements. The samples correspond to certain moments of time $(t_k)_{k=1}^{n_{\mathrm{t}}}$ that partition a time interval $[0, t_{\mathrm{period}}]$ as

$$0 = t_0 < t_1 < \ldots < t_{n_{\mathrm{t}}} = t_{\mathrm{period}}.$$

Analogously, a temperature profile $\mathbf{Q} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\mathrm{t}}}$ is a matrix containing samples of temperature. For clarity, power and temperature profiles are assumed to have a one-to-one correspondence and a constant sampling interval Δt, that is, $t_k - t_{k-1} = \Delta t$, for $k = 1, 2, \ldots, n_{\mathrm{t}}$. In what follows, a power profile that contains only the dynamic component of the (total) power dissipation will be denoted by $\mathbf{P}_{\mathrm{dyn}}$.

The system depends on a set of parameters that are uncertain at the design stage due to process variation. We model such parameters using random variables and denote them by a random vector $\mathbf{u} = (u_i)_{i=1}^{n_{\mathrm{u}}} : \Omega \to \mathbb{R}^{n_{\mathrm{u}}}$. In this work, we are only concerned with those parameters that manifest themselves in the deviation of the actual power dissipation from nominal values and, consequently, in the deviation of temperature from the one corresponding to the nominal power consumption.

Given $\mathcal{S}$, we pursue the following major objectives:

- **Objective 1.** Extend probabilistic temperature analysis to include the DSS scenario under the uncertainty due to process variation specified by $\mathbf{u}$.
- **Objective 2.** Taking into consideration the effect of process variation on temperature, find the survival function of the system at hand under an arbitrary workload given as a dynamic power profile $\mathbf{P}_{\mathrm{dyn}}$.
- **Objective 3.** Develop a computationally tractable design-space exploration scheme exploiting the proposed framework for temperature/reliability analysis.

In order to give a better intuition about our solutions, we shall accompany the development of our framework with the development of a concrete example/application, which will eventually be utilized for the quantitative evaluation of the framework given in Sec. X. To this end, we have decided to focus on two process parameters, which are arguably the most crucial ones, namely, the effective channel length u_1 and the gate-oxide thickness u_2. In this example,

$$\mathbf{u} = (u_1, u_2) : \Omega \to \mathbb{R}^2, \tag{1}$$

which is to be discussed in detail shortly. Regarding reliability, we shall address the thermal-cycling fatigue as it is naturally connected with DSS temperature analysis that we develop.

VI. Uncertainty Analysis

The key building block of our solutions developed in Sec. VII–IX is the uncertainty quantification technique presented in this section. The main task of this technique is the propagation of uncertainty through the system, that is, from a set of inputs to a set of outputs. Specifically, the inputs are the uncertain parameters $\mathbf{u}$, and the outputs are the quantities that we are interested in studying. The latter can be, for instance, the energy consumption, maximal temperature, or temperature profile of the system over a certain period of time.

Due to the inherent complexity, uncertainty quantification problems are typically viewed as approximation problems: one first constructs a computationally efficient surrogate for the stochastic model under consideration and then studies this computationally efficient representation instead of the original model. In order to construct such an approximation, we appeal to spectral methods [3], [15], [16].

A. Uncertainty Model

Before we proceed to the construction of light surrogate models, let us first refine our definition of $\mathbf{u} = (u_i)_{i=1}^{n_{\mathrm{u}}}$. Each u_i is a characteristic of a single transistor (consider, e.g., the effective channel length), and, therefore, each device in the electrical circuits at hand can potentially have a different value of this parameter as, in general, the variability due to process variation is not uniform. Consequently, each u_i can be viewed as a random process $u_i : \Omega \times D \to \mathbb{R}$ defined on an appropriate spatial domain $D \subset \mathbb{R}^2$. Since this work is system-level oriented, we model each processing element with one variable for each such random process. More specifically, we let $u_{ij} = u_i(\cdot, \mathbf{r}_j)$ be the random variable representing the ith uncertain parameter at the jth processing element where $\mathbf{r}_j$ stands for the spatial location of the center of the processing element. Therefore, we redefine the parameterization $\mathbf{u}$ of the problem at hand as

$$\mathbf{u} = (u_i)_{i=1}^{n_{\mathrm{u}} n_{\mathrm{p}}} \tag{2}$$

such that there is a one-to-one correspondence between u_i, $i = 1, 2, \ldots, n_{\mathrm{u}} n_{\mathrm{p}}$, and u_{ij}, $i = 1, 2, \ldots, n_{\mathrm{u}}$, $j = 1, 2, \ldots, n_{\mathrm{p}}$. For instance, in our illustrative application with two process parameters, the total number of stochastic dimensions is $2n_{\mathrm{p}}$.

Remark 1. *Some authors prefer to split the variability of a process parameter at a spatial location into several parts such as wafer-to-wafer, die-to-die, and within-die; see, e.g., [9]. However, from the mathematical point of view, it is sufficient to consider just one random variable per location which is adequately correlated with the other locations of interest.*

A description of $\mathbf{u}$ is an input to our analysis given by the user, and we consider it to be a part of the system specification $\mathcal{S}$. A proper (complete, unambiguous) way to describe a set of random variables is to specify their joint probability distribution function. In practice, however, such exhaustive information is often unavailable, in particular, due to the high dimensionality in the presence of prominent dependencies inherent to the considered problem. A more realistic assumption is the knowledge of the marginal distributions and correlation matrix of $\mathbf{u}$. Denote by $\{F_{u_i}\}_{i=1}^{n_\mathrm{u}n_\mathrm{p}}$ and $\mathbf{K}_u \in \mathbb{R}^{n_\mathrm{u}n_\mathrm{p}\times n_\mathrm{u}n_\mathrm{p}}$ the marginal distribution functions and correlation matrix of the uncertain parameters $\mathbf{u}$ in (2), respectively. Note that the number of distinct marginals is only n_u since n_p components of $\mathbf{u}$ correspond to the same uncertain parameter.

B. Parameter Preprocessing

Our foremost task now is to transform $\mathbf{u}$ into mutually independent random variables as independence is essential for the forthcoming mathematical treatment and practical computations. To this end, an adequate probability transformation should be undertaken depending on the available information; see [16] for an overview. One transformation for which the assumed knowledge about $\mathbf{u}$ is sufficient is the Nataf transformation [17]. Denote this transformation by

$$\mathbf{u} = \mathbb{T}\left[\boldsymbol{\xi}\right], \tag{3}$$

which relates $n_\mathrm{u}n_\mathrm{p}$ dependent random variables, i.e., $\mathbf{u}$, with $n_\xi = n_\mathrm{u}n_\mathrm{p}$ independent random variables

$$\boldsymbol{\xi} = (\xi_i)_{i=1}^{n_\xi}. \tag{4}$$

Regardless of the marginals, $\xi_i \sim \mathcal{N}(0,1)$, $i = 1, 2, \ldots, n_\xi$, that is, each ξ_i has the standard Gaussian distribution. Refer to Appendix B for further details about the Nataf transformation.

As we shall discuss later on, the stochastic dimensionality n_ξ has a considerable impact on the computational complexity of our framework. Therefore, an important part of the preprocessing stage is model order reduction. To this end, we preserve only those stochastic dimensions whose contribution to the total variance of $\mathbf{u}$ is the most significant, which is identified by the eigenvalues of the correlation matrix $\mathbf{K}_u$:

$$\boldsymbol{\lambda} = (\lambda_i)_{i=1}^{n_\mathrm{u}n_\mathrm{p}}, \quad \|\boldsymbol{\lambda}\|_1 = 1, \tag{5}$$

as it is further discussed in Appendix C. Without introducing additional transformations, we let $\mathbb{T}$ in (3) be augmented with such a reduction procedure and redefine $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi}$ as the reduced independent random variables where $n_\xi \leq n_\mathrm{u}n_\mathrm{p}$. We would like to note that this procedure is highly preferable as it helps to keep n_ξ moderate, and it is especially advantages when refining the granularity of the analysis (see Sec. V).

Let us turn to the illustrative application. Recall that we exemplify our framework considering the effective channel length and gate-oxide thickness with the notation given in (1). Both parameters correspond to Euclidean distances; they take values on bounded intervals of the positive part of the real line. With this in mind, we model the two process parameters using the four-parametric family of beta distributions:

$$u_i \sim F_{u_i} = \mathrm{Beta}\left(a_i, b_i, c_i, d_i\right)$$

where $i = 1, 2, \ldots, 2n_\mathrm{p}$, a_i and b_i control the shape of the distributions, and $[c_i, d_i]$ correspond to their supports. Without loss of generality, we let the two considered process parameters be independent of each other, and the correlations among those elements of $\mathbf{u}$ that correspond to the same process parameter be given by the following correlation function:

$$k(\mathbf{r}_i, \mathbf{r}_j) = \varpi\, k_\mathrm{SE}(\mathbf{r}_i, \mathbf{r}_j) + (1-\varpi) k_\mathrm{OU}(\mathbf{r}_i, \mathbf{r}_j) \tag{6}$$

where $\mathbf{r}_i \in \mathbb{R}^2$ is the center of the ith processing element relative to the center of the die. The correlation function is a composition of two kernels:

$$k_\mathrm{SE}(\mathbf{r}_i, \mathbf{r}_j) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|^2}{\ell_\mathrm{SE}^2}\right) \text{ and}$$
$$k_\mathrm{OU}(\mathbf{r}_i, \mathbf{r}_j) = \exp\left(-\frac{|\,\|\mathbf{r}_i\| - \|\mathbf{r}_j\|\,|}{\ell_\mathrm{OU}}\right),$$

which are known as the squared-exponential and Ornstein–Uhlenbeck kernels, respectively. In the above formulae, $\varpi \in [0,1]$ is a weight coefficient balancing the kernels; ℓ_SE and $\ell_\mathrm{OU} > 0$ are so-called length-scale parameters; and $\|\cdot\|$ stands for the Euclidean norm in $\mathbb{R}^2$. The choice of these two kernels is guided by the observations of the correlation patterns induced by the fabrication process: k_SE imposes similarities between those spatial locations that are close to each other, and k_OU imposes similarities between those locations that are at the same distance from the center of the die; see, e.g., [18] for additional details. The length-scale parameters ℓ_SE and ℓ_OU control the extend of these similarities, i.e., the range wherein the influence of one point on another is significant.

C. Surrogate Construction

Let $\vartheta : \Omega \to \mathbb{R}$ be a quantity of interest dependent on $\mathbf{u}$. For convenience, ϑ is assumed to be one-dimensional, which will be generalized later on. In order to give a computationally efficient probabilistic characterization of ϑ, we utilize nonintrusive spectral decompositions based on orthogonal polynomials. The corresponding mathematical foundation is outlined in Appendix D and Appendix E, and here we go directly to the main results obtained in those sections.

1) Classical Decomposition: Assume $\vartheta \in L^2(\Omega, \mathcal{F}, \mathbb{P})$ (see Sec. IV). Then ϑ can be expanded into the following series:

$$\vartheta \approx \mathcal{C}_{l_\mathcal{C}}^{n_\xi}[\vartheta] := \sum_{\boldsymbol{\alpha}\in\mathcal{A}(l_\mathcal{C})} \hat{\vartheta}_{\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \tag{7}$$

where $l_\mathcal{C}$ is the expansion level; $\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}_0^{n_\xi}$ is a multi-index; $\mathcal{A}(l_\mathcal{C})$ is an index set to be discussed shortly; and $\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ is an n_ξ-variate Hermite polynomial constructed as a product of normalized one-dimensional Hermite polynomials of orders specified by the corresponding elements of $\boldsymbol{\alpha}$.

As discussed in Appendix D, each coefficient $\hat{\vartheta}_{\boldsymbol{\alpha}}$ in (7) is an n_ξ-dimensional integral of the product of ϑ with $\psi_{\boldsymbol{\alpha}}$, and this integral should be computed numerically. To this end, we construct a quadrature rule and calculate $\hat{\vartheta}_{\boldsymbol{\alpha}}$ as

$$\hat{\vartheta}_{\boldsymbol{\alpha}} \approx \mathcal{Q}_{l_\mathcal{Q}}^{n_\xi}[\vartheta\, \psi_{\boldsymbol{\alpha}}] := \sum_{i=1}^{n_\mathcal{Q}} \vartheta(\mathbb{T}[\mathbf{x}_i])\, \psi_{\boldsymbol{\alpha}}(\mathbf{x}_i)\, w_i \tag{8}$$

where $l_\mathcal{Q}$ is the quadrature level, and $\{(\mathbf{x}_i \in \mathbb{R}^{n_\xi}, w_i \in \mathbb{R})\}_{i=1}^{n_\mathcal{Q}}$ are the points and weights of the quadrature. The multivariate quadrature operator $\mathcal{Q}_{l_\mathcal{Q}}^{n_\xi}$ is based on a set of univariate

operators and is constructed as follows:

$$\mathcal{Q}_{l_{\mathcal{Q}}}^{n_\xi} = \bigoplus_{\boldsymbol{\alpha}\in\mathcal{A}(l_{\mathcal{Q}})} \Delta_{\alpha_1} \otimes \cdots \otimes \Delta_{\alpha_{n_\xi}}. \tag{9}$$

The notation used in the above equation is not essential for the present discussion and is explained in Appendix E. The important aspect to note is the structure of this operator, namely, the index set $\mathcal{A}(l_{\mathcal{Q}})$, which we shall come back to shortly.

The standard choice of $\mathcal{A}(l_{\mathcal{C}})$ in (7) is $\{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\|_1 \leq l_{\mathcal{C}}\}$, which is called an isotropic total-order index set. *Isotropic* refers to the fact that all dimensions are trimmed identically, and *total-order* refers to the structure of the corresponding polynomial space. In (8), $\psi_{\boldsymbol{\alpha}}$ is a polynomial of total order at most $l_{\mathcal{C}}$, and ϑ is modeled as such a polynomial. Hence, the integrand in (8) is a polynomial of total order at most $2l_{\mathcal{C}}$. Having this aspect in mind, one usually constructs a quadrature rule such that it is exact for polynomials of total order $2l_{\mathcal{C}}$ [16]. In this work, we employ Gaussian quadratures for integration, in which case a quadrature of level $l_{\mathcal{Q}}$ is exact for integrating polynomials of total order $2l_{\mathcal{Q}}+1$ [19] (see also Appendix E). Therefore, it is sufficient to keep $l_{\mathcal{C}}$ and $l_{\mathcal{Q}}$ equal. More generally, the index sets $\mathcal{A}(l_{\mathcal{C}})$ and $\mathcal{A}(l_{\mathcal{Q}})$ should be synchronized; in what follows, we shall denote both by $\mathcal{A}(l)$.

2) Anisotropic Decomposition: In the context of sparse grids, an important generalization of the construction in (9) is the so-called anisotropic Smolyak algorithm [20]. The main difference between the isotropic and anisotropic versions lies in the constraints imposed on $\mathcal{A}(l)$. An anisotropic total-order index set is defined as follows:

$$\mathcal{A}(l) = \left\{\boldsymbol{\alpha} : \langle \mathbf{c}, \boldsymbol{\alpha}\rangle \leq l \, \min_i c_i\right\} \tag{10}$$

where $\mathbf{c} = (c_i) \in \mathbb{R}^{n_\xi}$, $c_i \geq 0$, is a vector assigning importance coefficients to each dimension, and $\langle\cdot,\cdot\rangle$ is the standard inner product on $\mathbb{R}^{n_\xi}$. Equation (10) plugged into (9) results in a sparse grid which is exact for the polynomial space that is tailored using the same index set.

The above approach allows one to leverage the highly anisotropic behaviors inherent for many practical problems [20]. It provides a great control over the computational time associated with the construction of spectral decompositions: a carefully chosen importance vector $\mathbf{c}$ in (10) can significantly reduce the number of polynomial terms in (7) and the number of quadrature points needed in (8) to compute the coefficients of those terms. The question to discuss now is the choice of $\mathbf{c}$. In this regard, we rely on the variance contributions of the dimensions given by $\boldsymbol{\lambda}$ in (5). Specifically, we let

$$\mathbf{c} = \boldsymbol{\lambda}^\gamma := (\lambda_i^\gamma)_{i=1}^{n_\xi} \tag{11}$$

where $\gamma \in [0,1]$ is a tuning parameter. The isotropic scenario can be recovered by setting $\gamma = 0$; the other values of γ correspond to various levels of anisotropy with the maximum attained by setting $\gamma = 1$.

Let us sum up what we have achieved at this point. In order to give a probabilistic characterization of a quantity of interest, we perform polynomial expansions as shown in (7). The coefficients of such expansions are evaluated by means of Gaussian quadratures as shown in (8). The quadratures are constructed using the Smolyak formula given in (9). The index sets used in both (7) and (9) are the one given in (10) wherein the anisotropic weights are set according to (5) and (11).

Algorithm 1 Surrogate construction

Input: Algorithm X % the subroutine evaluating ϑ
Output: $\hat{\mathbf{v}} \in \mathbb{R}^{n_\mathcal{C}}$ % the expansion coefficients

1: **for** $i \leftarrow 1$ to $n_{\mathcal{Q}}$ **do** % for each quadrature point $\mathbf{x}_i$
2: $\quad\mathbf{u} \leftarrow \mathbb{T}[\mathbf{x}_i]$
3: $\quad\mathbf{v}(i) \leftarrow$ call **Algorithm X** for $\mathbf{u}$
4: **end for**
5: $\hat{\mathbf{v}} \leftarrow \boldsymbol{\Pi}\,\mathbf{v}$

3) Efficient Implementation: The pair of $\boldsymbol{\xi}$ and $\mathbf{c}$ uniquely characterizes the uncertainty quantification problem at hand. Once they have been identified, and the desired approximation level $l = l_{\mathcal{C}} = l_{\mathcal{Q}}$ has been specified, the corresponding polynomial basis and quadrature stay the same for all quantities that one might be interested in studying. This observation is of high importance as a lot of preparatory work can and should be done only once and then stored for future uses. In particular, the construction in (7) can be reduced to one matrix multiplication with a precomputed matrix, which we shall demonstrate next.

Let $n_{\mathcal{C}} = \#\mathcal{A}(l)$ be the cardinality of $\mathcal{A}(l)$, which is also the number of polynomial terms and, hence, coefficients in (7). Assume the multi-indices contained in $\mathcal{A}(l)$ are arranged in a vector $(\boldsymbol{\alpha}_i)_{i=1}^{n_\mathcal{C}}$, which gives a certain ordering. Now, let

$$\boldsymbol{\Pi} = \left(\pi_{ij} = \psi_{\boldsymbol{\alpha}_i}(\mathbf{x}_j)\, w_j\right)_{i=1,\,j=1}^{i=n_\mathcal{C},\,j=n_\mathcal{Q}}, \tag{12}$$

that is, π_{ij} is the polynomial corresponding to the ith multi-index evaluated at the jth quadrature point and multiplied by the jth quadrature weight. We refer to $\boldsymbol{\Pi}$ as the projection matrix. The coefficients in (7) can now be computed as

$$\hat{\mathbf{v}} = \boldsymbol{\Pi}\,\mathbf{v} \tag{13}$$

where

$$\hat{\mathbf{v}} = (\hat{\vartheta}_i)_{i=1}^{n_\mathcal{C}} \quad \text{and} \quad \mathbf{v} = \left(\vartheta(\mathbb{T}[\mathbf{x}_i])\right)_{i=1}^{n_\mathcal{Q}}. \tag{14}$$

It can be seen that (13) is a matrix version of (8). $\boldsymbol{\Pi}$ is the one that should be precomputed. The pseudocode of the procedure is given in Algorithm 1 wherein Algorithm X stands for the routine that calculates ϑ for a given $\mathbf{u}$. Needless to say, Algorithm X is problem specific and has a crucial impact on the performance of the whole procedure presented in this section: any modeling errors inherent to this algorithm can propagate to the output of the uncertainty analysis. Algorithm X will be further discussed in Sec. VII–IX.

D. Postprocessing

The function given by (7) is nothing more than a polynomial; hence, it is easy to interpret and easy to evaluate. Consequently, having constructed such an expansion, various statistics about ϑ can be estimated with little effort. Moreover, (7) yields analytical formulae for the expected value and variance of ϑ solely based on the coefficients of (7):

$$\mathbb{E}[\vartheta] = \hat{\vartheta}_{\mathbf{0}} \quad \text{and} \quad \mathbb{V}\text{ar}[\vartheta] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}(l)\setminus\{\mathbf{0}\}} \hat{\vartheta}_{\boldsymbol{\alpha}}^2 \tag{15}$$

where $\mathbf{0} = (0)$ is a multi-index with all entries equal to zero. Such quantities as the cumulative distribution and probability

density functions can be estimated by sampling (7); each sample is a trivial evaluation of a polynomial.

Remark 2. *When ϑ is multidimensional, we shall consider it as a row vector with an appropriate number of elements. Then all the operations with respect to ϑ, such as those in (7), (8), and (15), should be undertaken elementwise. In (13), (14), and Algorithm 1, $\mathbf{v}$ and $\hat{\mathbf{v}}$ are to be treated as matrices with $n_{\mathcal{C}}$ rows, and ϑ_i as a row vector. The output of Algorithm X is assumed to be automatically reshaped into a row vector.*

In what follows, we shall apply the probabilistic analysis developed in this section to a number of concrete problems: temperature analysis (Sec. VII), reliability analysis (Sec. VIII), and reliability optimization (Sec. IX).

VII. Temperature Analysis

In this section, we detail our temperature analysis, which is suitable for system-level studies. We shall cover both the transient and dynamic steady-state scenarios as the former is a prerequisite for the latter. Since temperature is a direct consequence of power, we begin with the power model utilized in the proposed framework.

A. Power Model

Recall that the system is composed of n_{p} processing elements and depends on the outcome of the probability space $\omega \in \Omega$ via $\mathbf{u}$. The total dissipation of power is modeled as the following system of n_{p} temporal stochastic process:

$$\mathbf{p}(t, \mathbf{u}, \mathbf{q}(t, \mathbf{u})) = \mathbf{p}_{\text{dyn}}(t) + \mathbf{p}_{\text{stat}}(\mathbf{u}, \mathbf{q}(t, \mathbf{u})) \tag{16}$$

where, for time $t \geq 0$, $\mathbf{p}_{\text{dyn}} \in \mathbb{R}^{n_{\text{p}}}$ and $\mathbf{p}_{\text{stat}} \in \mathbb{R}^{n_{\text{p}}}$ are vectors representing the dynamic and static components of the total power, respectively, and $\mathbf{q} \in \mathbb{R}^{n_{\text{p}}}$ is the corresponding vector of temperature. $\mathbf{p}_{\text{dyn}}$ is deterministic, and the rest are random.

Remark 3. *In (16), $\mathbf{p}_{\text{dyn}}$ has no dependency on $\mathbf{u}$ as the influence of process variation on the dynamic power is known to be negligibly small [1]. On the other hand, the variability of $\mathbf{p}_{\text{stat}}$ is substantial and is further magnified by the well-known interdependency between leakage and temperature.*

B. Thermal Model

Based on the information gathered in $\mathcal{S}$ (see Sec. V), an equivalent thermal RC circuit of the system is constructed [21]. The circuit comprises n_{n} thermal nodes, and its structure depends on the intended level of granularity that impacts the resulting accuracy. For clarity, we assume that each processing element is mapped onto one corresponding node, and the thermal package is represented as a set of additional nodes.

The thermal dynamics of the system are modeled using the following system of differential-algebraic equations [4], [6]:

$$\begin{cases} \dfrac{d\,\mathbf{s}(t, \mathbf{u})}{dt} = \mathbf{A}\,\mathbf{s}(t, \mathbf{u}) + \mathbf{B}\,\mathbf{p}(t, \mathbf{u}, \mathbf{q}(t, \mathbf{u})) & \text{(17a)} \\ \mathbf{q}(t, \mathbf{u}) = \mathbf{B}^T \mathbf{s}(t, \mathbf{u}) + \mathbf{q}_{\text{amb}} & \text{(17b)} \end{cases}$$

where

$$\mathbf{A} = -\mathbf{C}^{-\frac{1}{2}} \mathbf{G} \mathbf{C}^{-\frac{1}{2}} \quad \text{and} \quad \mathbf{B} = \mathbf{C}^{-\frac{1}{2}} \mathbf{M}.$$

For time $t \geq 0$, $\mathbf{p} \in \mathbb{R}^{n_{\text{p}}}$, $\mathbf{q} \in \mathbb{R}^{n_{\text{p}}}$, and $\mathbf{s} \in \mathbb{R}^{n_{\text{n}}}$ are the power, temperature, and state vectors, respectively. $\mathbf{q}_{\text{amb}} \in \mathbb{R}^{n_{\text{p}}}$ is a vector of the ambient temperature. $\mathbf{M} \in \mathbb{R}^{n_{\text{n}} \times n_{\text{p}}}$ is a matrix that distributes the power dissipation of the processing elements across the thermal nodes; without loss of generality, $\mathbf{M}$ is a rectangular diagonal matrix wherein each diagonal element is equal to unity. $\mathbf{C} \in \mathbb{R}^{n_{\text{n}} \times n_{\text{n}}}$ and $\mathbf{G} \in \mathbb{R}^{n_{\text{n}} \times n_{\text{n}}}$ are a diagonal matrix of the thermal capacitance and a symmetric, positive-definite matrix of the thermal conductance, respectively.

C. Our Solution

Let us fix $\omega \in \Omega$, meaning that $\mathbf{u}$ is assumed to be known, and consider the system in (17) as deterministic. In general, (17a) is a system of ordinary differential equations which is nonlinear due to the power term, given in (16) as an arbitrary function. Hence, the system in (17) does not have a general closed-form solution. A robust and computationally efficient solution to (17) for a given $\mathbf{u}$ is an essential part of our probabilistic framework. In order to attain such a solution, we utilize a numerical method from the family of exponential integrators [22]. The procedure is described in Appendix A, and here we use the final result.

Recall that we are to analyze a dynamic power profile $\mathbf{P}_{\text{dyn}}$ covering a time interval $[0, t_{\text{period}}]$ with n_{t} samples that are evenly spaced in time. The transient solution of (17a) is reduced to the following recurrence for $k = 1, 2, \ldots, n_{\text{t}}$:

$$\mathbf{s}_k = \mathbf{E}\,\mathbf{s}_{k-1} + \mathbf{F}\,\mathbf{p}_k \tag{18}$$

where the subscript k stands for time $k\Delta t$, $\mathbf{s}_0 = \mathbf{0}$,

$$\mathbf{E} = e^{\mathbf{A}\Delta t}, \quad \text{and} \quad \mathbf{F} = \mathbf{A}^{-1}(e^{\mathbf{A}\Delta t} - \mathbf{I})\,\mathbf{B}.$$

For computational efficiency, we perform the eigendecomposition of the state matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{19}$$

where $\mathbf{V}$ and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_i)$ are an orthogonal matrix of the eigenvectors and a diagonal matrix of the eigenvectors of $\mathbf{A}$, respectively. The matrices $\mathbf{E}$ and $\mathbf{F}$ are then

$$\mathbf{E} = \mathbf{V} \operatorname{diag}\left(e^{\lambda_i \Delta t}\right) \mathbf{V}^T \text{ and}$$
$$\mathbf{F} = \mathbf{V} \operatorname{diag}\left(\frac{e^{\lambda_i \Delta t} - 1}{\lambda_i}\right) \mathbf{V}^T \mathbf{B}.$$

To sum up, the derivation up to this point is sufficient for transient temperature analysis via (18) followed by (17b).

Remark 4. *Although the focus of this paper is on temperature, each temperature analysis developed is accompanied by the corresponding power analysis as the two are inseparable due to the leakage-temperature interplay. Consequently, when it is appropriate, one can easily extract only the (temperature-aware) power part of the presented solutions.*

Let us move on to the dynamic steady-state (DSS) case. Assume for now that $\mathbf{p}_{\text{stat}}$ in (16) does not depend on $\mathbf{q}$, i.e., no interdependency between leakage and temperature. The DSS boundary condition is $\mathbf{s}_1 = \mathbf{s}_{n_{\text{t}}+1}$. In other words, the system is required to come back to its original state by the end of the analyzed time frame. This constraint and (18) yield a block-circulant system of $n_{\text{n}} n_{\text{t}}$ linear equations with $n_{\text{n}} n_{\text{t}}$ unknowns. As describe in detail in [4], the system can be efficiently solved

Algorithm 2 Deterministic dynamic steady-state temperature analysis with no static dissipation of power

Input: $\mathbf{P} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{t}}$
Output: $\mathbf{Q} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{t}}$

$\widehat{\mathbf{A}} \leftarrow \mathbf{F}\,\mathbf{P}$
$\widehat{\mathbf{a}} \leftarrow \widehat{\mathbf{A}}(:,1)$
for $k \leftarrow 2$ to n_t **do**
  $\widehat{\mathbf{a}} \leftarrow \mathbf{E}\,\widehat{\mathbf{a}} + \widehat{\mathbf{A}}(:,k)$
end for
$\mathbf{S}(:,1) \leftarrow \mathbf{V}\ \mathrm{diag}\left(\left(1 - e^{n_\mathrm{t}\Delta t \lambda_i}\right)^{-1}\right)\ \mathbf{V}^T\,\widehat{\mathbf{a}}$
for $k \leftarrow 2$ to n_t **do**
  $\mathbf{S}(:,k) \leftarrow \mathbf{E}\,\mathbf{S}(:,k-1) + \widehat{\mathbf{A}}(:,k-1)$
end for
$\mathbf{Q} \leftarrow \mathbf{B}^T\mathbf{S} + \mathbf{Q}_\mathrm{amb}$

Algorithm 3 Deterministic dynamic steady-state temperature analysis considering the static power

Input: $\mathbf{P}_\mathrm{dyn} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{t}}$ and $\mathbf{u} \in \mathbb{R}^{n_\mathrm{p}}$
Output: $\mathbf{Q} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{t}}$ and $\mathbf{P} \in \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{t}}$

$\mathbf{Q} \leftarrow \mathbf{Q}_\mathrm{amb}$
repeat
  $\mathbf{P} \leftarrow \mathbf{P}_\mathrm{dyn} + \mathbf{P}_\mathrm{stat}(\mathbf{u}, \mathbf{Q})$
  $\mathbf{Q} \leftarrow$ call **Algorithm 2** for $\mathbf{P}$
until a stopping condition is satisfied

by exploiting its particular structure and the decomposition in (19). The pseudocode of this algorithm, which delivers the exact solution under the above assumptions, is given in Algorithm 2. Here we adopt MATLAB's [23] notations $\mathbf{A}(k,:)$ and $\mathbf{A}(:,k)$ to refer to the kth row and the kth column of a matrix $\mathbf{A}$, respectively. In the pseudocode, auxiliary variables are written with hats, and $\mathbf{Q}_\mathrm{amb}$ is a matrix of the ambient temperature.

Remark 5. *The time complexity of direct dense and sparse solvers (e.g., the LU decomposition) of the system of linear equations is $\mathcal{O}\left({n_t}^3 {n_n}^3\right)$ while the one of Algorithm 2 is only $\mathcal{O}\left(n_t {n_n}^2 + {n_n}^3\right)$, which the algorithm is able to achieve by exploiting the specific structure of the system; see [4].*

Let us now bring the leakage-temperature interdependence into the picture. To this end, we repeat Algorithm 2 for a sequence of total power profiles $\{\mathbf{P}_k = \mathbf{P}_\mathrm{dyn} + \mathbf{P}_{\mathrm{stat},k}\}$ wherein the static part $\mathbf{P}_{\mathrm{stat},k}$ is being updated using (16) given the temperature profile $\mathbf{Q}_{k-1}$ computed at the previous iteration starting from the ambient temperature. The procedure stops when the sequence of temperature profiles $\{\mathbf{Q}_k\}$ converges in an appropriate norm, or some other stopping condition is satisfied (e.g., a maximal temperature constraint is violated). This procedure is illustrated in Algorithm 3.

In Algorithm 3, $\mathbf{P}_\mathrm{stat}(\mathbf{u}, \mathbf{Q})$ should be understood as a call to a subroutine that returns an $n_\mathrm{p} \times n_\mathrm{t}$ matrix wherein the (i,k)th element is the static component of the power dissipation of the ith processing element at the kth moment of time with respect to $\mathbf{u}$ and the temperature given by the (i,k)th entry of $\mathbf{Q}$.

Remark 6. *A widespread approach to account for leakage is to linearize it with respect to temperature. As shown in [2], already one linear segment can deliver sufficiently accurate results. One notable feature of such a linearization is that no iterating-until-convergence is needed in this case; see [4]. However, this technique assumes that the only varying parameter of leakage is temperature, and all other parameters have nominal values. In that case, it is relatively easy to decide on a representative temperature range and undertake a one-dimensional curve-fitting procedure with respect to it. In our case, the power model has multiple parameters stepping far from their nominal values, which makes it difficult to construct a good linear fit with respect to temperature. Thus, in order to be accurate, we use a nonlinear model of leakage.*

So far in this subsection, $\mathbf{u}$ has been assumed to be deterministic. Now we turn to the stochastic scenario and let $\mathbf{u}$ be random. Then we apply Algorithm 1 to one particular quantity of interest ϑ. Specifically, ϑ is now the temperature profile $\mathbf{Q}$ corresponding to a given $\mathbf{P}_\mathrm{dyn}$. Since $\mathbf{Q}$ is an $n_\mathrm{p} \times n_\mathrm{t}$ matrix, following Remark 2, ϑ is viewed as an $n_\mathrm{p} n_\mathrm{t}$-element row vector, in which case each coefficient $\hat{\vartheta}_{\boldsymbol{\alpha}}$ in (7) is also such a vector. The projection in (13) and, consequently, Algorithm 1 should be interpreted as follows: $\mathbf{v}$ is an $n_\mathcal{Q} \times n_\mathrm{p} n_\mathrm{t}$ matrix, and the ith row of this matrix is the temperature profile computed at the ith quadrature point and reshaped into a row vector. Similarly, $\hat{\mathbf{v}}$ is an $n_\mathcal{C} \times n_\mathrm{p} n_\mathrm{t}$ matrix, and the ith row of this matrix is the ith coefficient $\hat{\vartheta}_{\boldsymbol{\alpha}_i}$ of the spectral decomposition in (7) (recall that a fixed ordering is assumed to be imposed on the multi-indices). Keeping the above in mind, a call to Algorithm 1 should be made such that Algorithm X points at an auxiliary routine which receives $\mathbf{u}$, forwards it to Algorithm 3 along with $\mathbf{P}_\mathrm{dyn}$, and returns the resulting temperature profile to Algorithm 1. The constructed expansion can now be postprocessed as needed; see Sec. VI-D.

To give a concrete example, for a dual-core system ($n_\mathrm{p} = 2$) with one independent random variable ($n_\xi = 1$), a second-level expansion ($l_\mathcal{C} = 2$) of the temperature profile with 100 time steps ($n_\mathrm{t} = 100$) can be written as follows (see (7)):

$$\vartheta \approx \hat{\vartheta}_0 + \hat{\vartheta}_1 \xi + \hat{\vartheta}_2 (\xi^2 - 1),$$

which is a polynomial in ξ, and each coefficient is a vector with $n_\mathrm{p} n_\mathrm{t} = 200$ elements. Then, for any outcome $\xi \equiv \xi(\omega)$, the corresponding temperature profile $\mathbf{Q}$ can be evaluated by plugging in ξ into the above equation and reshaping the result into an $n_\mathrm{p} \times n_\mathrm{t} = 2 \times 100$ matrix. In this case, the three rows of $\hat{\mathbf{v}}$ are $\hat{\vartheta}_0$, $\hat{\vartheta}_1$, and $\hat{\vartheta}_2$; the first one is also a flattened version of the expected value of $\mathbf{Q}$ as shown in (15).

VIII. Reliability Analysis

In this section, our primary objective is to build a flexible and computationally efficient framework for the reliability analysis of electronic systems affected by process variation. Let us begin with a description of a generic reliability model and make several observations with respect to it.

A. Reliability Model

Let $\tau : \Omega \to \mathbb{R}$ be a random variable representing the lifetime of the considered system. The lifetime is the time span until the system experiences a fault after which the system no longer meets the imposed requirements. Let $F_\tau(\cdot|\boldsymbol{\theta})$ be the

distribution of τ where $\boldsymbol{\theta} = (\theta_i)$ is a vector of parameters. The survival function of the system is

$$R_\tau(t|\boldsymbol{\theta}) = 1 - F_\tau(t|\boldsymbol{\theta}).$$

The overall lifetime τ is a function of the lifetimes of the processing elements, which are denoted by a set of random variables $\{\tau_i\}_{i=1}^{n_\mathrm{p}}$. Each τ_i is characterized by a physical model of wear [5] describing the fatigues that the corresponding processing element is exposed to. Each τ_i is also assigned an individual survival function $R_{\tau_i}(\cdot|\boldsymbol{\theta})$ describing the failures due to those fatigues. The structure of $R_\tau(\cdot|\boldsymbol{\theta})$ with respect to $\{R_{\tau_i}(\cdot|\boldsymbol{\theta})\}_{i=1}^{n_\mathrm{p}}$ is problem specific, and it can be especially diverse in the context of fault-tolerant systems. $R_\tau(\cdot|\boldsymbol{\theta})$ is to be specified by the designer of the system, and it is assumed to be included in the specification $\mathcal{S}$ (see Sec. V). To give an example, suppose the failure of any of the n_p processing elements makes the system fail, and $\{\tau_i\}_{i=1}^{n_\mathrm{p}}$ are conditionally independent given the parameters gathered in $\boldsymbol{\theta}$. In this scenario,

$$\tau = \min_{i=1}^{n_\mathrm{p}} \tau_i \quad \text{and} \quad R_\tau(t|\boldsymbol{\theta}) = \prod_{i=1}^{n_\mathrm{p}} R_{\tau_i}(t|\boldsymbol{\theta}). \tag{20}$$

Our work in this context is motivated by the following two observations. First, temperature is the driving force of the dominant failure mechanisms. The most prominent examples include electromigration, time-dependent dielectric breakdown, stress migration, and thermal cycling [10]; see [5] for an exhaustive overview. All of the aforementioned mechanisms have strong dependencies on the operating temperature, which is taken into account by considering the parameters in $\boldsymbol{\theta}$ as adequate functions of temperature. At the same time, temperature is tightly related to process parameters, such as the effective channel length and gate-oxide thickness, and can vary dramatically when those parameters deviate from their nominal values [6], [9]. Meanwhile, the state-of-the-art techniques for reliability analysis of electronic systems lack a systematic treatment of process variation and, in particular, of the effect of process variation on temperature.

Second, having determined a probabilistic model $R_\tau(\cdot|\boldsymbol{\theta})$ of the considered system, the major portion of the associated computational time is ascribed to the evaluation of the parameterization $\boldsymbol{\theta}$ rather than to the model *per se*, that is, when $\boldsymbol{\theta}$ is known. For instance, $\boldsymbol{\theta}$ often contains estimates of the mean time to failure of each processing element given for a range of stress levels. Therefore, $\boldsymbol{\theta}$ typically involves (computationally intensive) full-system simulations including power analysis paired with temperature analysis [10].

Remark 7. *It is important to realize that there are two levels of probabilistic modeling here. First, the reliability model* per se *is a probabilistic model describing the lifetime of the system. Second, the parameterization $\boldsymbol{\theta}$ is another probabilistic model characterizing the impact of the uncertainty due to process variation on the reliability model. Consequently, the overall model can be thought of as a probability distribution over probability distributions. Given an outcome of the fabrication process, that is, $\boldsymbol{\theta}$, the lifetime remains random.*

B. Our Solution

Guided by the aforementioned observations, we propose to use the spectral decompositions developed in Sec. VI and Sec. VII in order to construct a light surrogate for $\boldsymbol{\theta}$. The proposed technique is founded on the basis of the state-of-the-art reliability models by enriching their modeling capabilities with respect to process variation and by speeding up the associated computational process. This approach allows one to seamlessly incorporate into reliability analysis the effect of process variation on process parameters. In particular, the framework allows for a straightforward propagation of the uncertainty from process parameters through power and temperature to the lifetime of the system. In contrast to the straightforward use of Monte Carlo (MC) sampling, the spectral representation that we construct makes the subsequent analysis highly efficient from the computational perspective.

It is worth noting that $R_\tau(\cdot|\boldsymbol{\theta})$ is left intact, meaning that our approach does not impose any restrictions on $R_\tau(\cdot|\boldsymbol{\theta})$. Thus, the user can take advantage of various reliability models in a straightforward manner. Naturally, this also implies that the modeling errors associated with the chosen $R_\tau(\cdot|\boldsymbol{\theta})$ can affect the quality of the results delivered by our technique. Therefore, choosing an adequate reliability model for the problem at hand is a responsibility of the user.

Let us now apply our general technique to address one of the major concerns of the designer of electronic systems: the thermal-cycling fatigue [5]. This fatigue has a sophisticated dependency on temperature: apart from average/maximal temperatures, the frequencies and amplitudes of temperature fluctuations matter in this case. Suppose that the system at hand is experiencing a periodic workload due to the execution of a periodic or nearly periodic application with period t_period. The power consumption is changing during the execution of the application, and, thus, the system is inevitably exposed to the damage from thermal oscillations. The corresponding temperature profile $\mathbf{Q}$ is then a DSS profile, which, for a given $\mathbf{u}$, can be computed using Algorithm 3.

Assume further that the structure of the reliability model is the one shown in (20). Regarding the individual survival functions, we shall rely on Weibull distributions. In this case,

$$\ln R_{\tau_i}(t|\boldsymbol{\theta}) = -\left(\frac{t}{\eta_i}\right)^{\beta_i} \tag{21}$$

and the mean time to failure is

$$\mu_i = \mathbb{E}[\tau_i] = \eta_i \, \Gamma\left(1 + \frac{1}{\beta_i}\right) \tag{22}$$

where η_i and β_i are the scale and shape parameters of the distribution, respectively, and Γ is the gamma function. At this point, $\boldsymbol{\theta} = (\eta_1, \ldots, \eta_{n_\mathrm{p}}, \beta_1, \ldots, \beta_{n_\mathrm{p}})$.

During one iteration of the application, the temperature of the ith processing element exhibits $n_{\mathrm{s}\,i}$ cycles. Each cycle generally has different characteristics and, therefore, causes different damage to the system. This aspect is taken into account by adjusting η_i as follows. Let $\mathbf{Q}$ be the DSS temperature profile of the system under analysis and denote by $\mathbf{Q}(i,:)$ the ith row of $\mathbf{Q}$, which corresponds to the temperature curve of the ith processing element. First, $\mathbf{Q}(i,:)$ is analyzed using a peak-detection procedure in order to extract the extrema of this curve. The found extrema are then fed to the rainflow counting algorithm [10] for an adequate identification of thermal cycles. Denote by $n_{\mathrm{c}\,ij}$ the expected number of cycles to failure corresponding to the ith processing element and its jth cycle

(as if it was the only cycle damaging the processing element). $n_{\mathrm{c}\,ij}$ is computed using the corresponding physical model of wear that can be found in [4], [5], [10]. Let η_{ij} and μ_{ij} be the scale parameter and expectation of the lifetime corresponding to the ith processing element under the stress of the jth cycle; the two are related as shown in (22). Then [4], [10]

$$\eta_i = \frac{t_{\text{period}}}{\Gamma\left(1+\frac{1}{\beta_i}\right)\sum_{j=1}^{n_{\mathrm{s}\,i}}\frac{1}{n_{\mathrm{c}\,ij}}}. \tag{23}$$

Note that η_i accounts for process variation via temperature (in the above equation, $n_{\mathrm{c}\,ij}$ is a function of temperature).

Remark 8. *A cycle need not be formed by adjacent extrema; cycles can overlap. In this regard, the rainflow counting method is known to be the best as it efficiently mitigates overestimation. A cycle can be a half cycle, meaning that only an upward or downward temperature swing is present in the time series, which is assumed to be taken into account in $n_{\mathrm{c}\,ij}$.*

The shape parameter β_i is known to be indifferent to temperature. For simplicity, we also assume that β_i does not depend on process parameters and $\beta_i = \beta$ for $i = 1, 2, \ldots, n_{\mathrm{p}}$. However, we would like to emphasize that these assumptions are not a limitation of the proposed techniques. Then it can be shown that the compositional survival function $R_\tau(\cdot|\boldsymbol{\theta})$ corresponds to a Weibull distribution, and the shape parameter of this distribution is β whereas the scale parameter is given by the following equation:

$$\eta = \left(\sum\left(\frac{1}{\eta_i}\right)^{\beta}\right)^{-\frac{1}{\beta}} \tag{24}$$

where η_i is as in (23). Consequently, the parameterization of the reliability model has boiled down to two parameters, η and β, among which only η is random.

Now we let the scale parameter η be our quantity of interest ϑ and apply the technique in Sec. VI to this quantity. In this case, Algorithm X in Algorithm 1 is an auxiliary function that makes a call to Algorithm 3, processes the resulting temperature profile as it was described earlier in this subsection, and returns η computed according to the formula in (24). Consequently, we obtain a light polynomial surrogate of the parameterization of the reliability model, which can be then studied from various perspectives. The example for a dual-core system given at the end of Sec. VII-C can be considered in this context as well with the only change that the dimensionality of the polynomial coefficients would be two here (since $\eta \in \mathbb{R}^{n_{\mathrm{p}}}$ and $n_{\mathrm{p}} = 2$).

IX. Reliability Optimization

In this section, the proposed analysis techniques are applied in the context of design-space exploration.

A. Problem Formulation

Consider a periodic application which is composed of a number of tasks and is given as a directed acyclic graph. The graph has n_{v} vertices representing the tasks and a number of edges specifying data dependencies between those tasks. Any processing element can execute any task, and each pair of a processing element and a task is characterized by an execution time and dynamic power. Since the proposed techniques are orientated towards the design stage, static scheduling is considered, which is typically done offline. More specifically, the application is scheduled using a static cyclic scheduler, and schedules are generated using the list scheduling policy [24]. A schedule is defined as a mapping of the tasks onto the processing elements and the corresponding starting times; we shall denote it by $\mathcal{S}$. The goal of our optimization is to find such a schedule $\mathcal{S}$ that minimizes the energy consumption while satisfying certain constraints.

Since energy is a function of power, and power depends on a set of uncertain parameters, the energy consumption is a random variable at the design stage, which we denote by $\mathcal{E}$. Our objective is to minimize the expected value of $\mathcal{E}$:

$$\min_{\mathcal{S}} \mathbb{E}\left[\mathcal{E}(\mathcal{S})\right] \tag{25}$$

where

$$\mathcal{E}(\mathcal{S}) = \Delta t \sum \mathbf{P}(\mathcal{S}),$$

Δt is the sampling interval of the power profile $\mathbf{P}$, and $\sum \mathbf{P}$ denotes the summation over all elements of $\mathbf{P}$. Hereafter, we also emphasize the dependency on $\mathcal{S}$. Our constraints are (i) time, (ii) temperature, and (iii) reliability as follows. (i) The period of the application is constrained by $t_{\max}$ (a deadline). (ii) The maximal temperature that the system can tolerate is constrained by $q_{\max}$, and ρ_{burn} is an acceptable probability of burning the chip. (iii) The minimal time that the system should survive is constrained by $\tau_{\min}$, and ρ_{wear} is an acceptable probability of having a premature fault due to wear. The three constraints are formalized as follows:

$$t_{\text{period}}(\mathcal{S}) \le t_{\max}, \tag{26}$$

$$\mathbb{P}\left(\mathcal{Q}(\mathcal{S}) \ge q_{\max}\right) \le \rho_{\text{burn}}, \text{ and} \tag{27}$$

$$\mathbb{P}\left(\mathcal{T}(\mathcal{S}) \le \tau_{\min}\right) \le \rho_{\text{wear}}. \tag{28}$$

In (26)–(28), t_{period} is the period of the application according to the schedule,

$$\mathcal{Q}(\mathcal{S}) = \|\mathbf{Q}(\mathcal{S})\|_\infty,$$

$$\mathcal{T}(\mathcal{S}) = \mathbb{E}\left[\tau(\mathcal{S}) \,|\, \eta\right] = \eta(\mathcal{S})\,\Gamma\left(1+\frac{1}{\beta}\right), \text{ and}$$

$\|\mathbf{Q}\|_\infty$ denotes the extraction of the maximal value from the temperature profile $\mathbf{Q}$. The last two constraints, i.e., (27) and (28), are probabilistic as the quantities under consideration are random. In (28), we set an upper bound on the probability of the expected value of τ, and it is important to realize that this expectation is a random variable itself due to the nested structure of the reliability model described in Remark 7.

B. Our Solution

In order to evaluate (25)–(28), we utilize the uncertainty analysis technique presented in Sec. VI. In this case, the quantity of interest is a vector with three elements:

$$\boldsymbol{\vartheta} = (\mathcal{E}, \mathcal{Q}, \mathcal{T}). \tag{29}$$

Although it is not spelled out, each quantity depends on $\mathcal{S}$. The first element corresponds to the energy consumption used in (25), the second element is the maximal temperature used in (27), and the last one is the scale parameter of the reliability model (see Sec. VIII) used in (28). The uncertainty analysis in Sec. VI should be applied as explained in Remark 2. In

Algorithm 1, Algorithm X is an intermediate procedure that makes a call to Algorithm 3 and processes the resulting power and temperature profiles as required by (29).

We use a genetic algorithm for optimization. Each chromosome is a $2n_{\mathrm{v}}$-element vector (twice the number of tasks) concatenating a pair of two vectors. The first is a vector in $\{1, 2, \ldots, n_{\mathrm{p}}\}^{n_{\mathrm{v}}}$ that maps the tasks onto the processing elements (i.e., a mapping). The second is a vector in $\{1, 2, \ldots, n_{\mathrm{v}}\}^{n_{\mathrm{v}}}$ that orders the tasks according to their priorities (i.e., a ranking). Since we rely on a static cyclic scheduler and the list scheduling policy [24], such a pair of vectors uniquely encodes a schedule $\mathcal{S}$. The population contains $4n_{\mathrm{v}}$ individuals which are initialized using uniform distributions. The parents for the next generation are chosen by a tournament selection with the number of competitors equal to 20% of n_{v}. A one-point crossover is then applied to 80% of the parents. Each parent undergoes a uniform mutation wherein each gene is altered with probability 0.01. The top five-percent individuals always survive. The stopping condition is the absence of improvement within 10 successive generations.

Let us turn to the evaluation of a chromosome's fitness. We begin by checking the timing constraint given in (26) as it does not require any probabilistic analysis; the constraint is purely deterministic. If (26) is violated, we set the fitness to the amount of this violation relative to the constraint—that is, to the difference between the actual application period and the deadline $t_{\max}$ divided by $t_{\max}$—and add a large constant, say, C, on top. If (26) is satisfied, we perform our probabilistic analysis and proceed to checking the constraints in (27) and (28). If any of the two is violated, we set the fitness to the total relative amount of violation plus $C/2$. If all the constraints are satisfied, the fitness value of the chromosome is set to the expected consumption of energy, as in shown in (25).

In order to speed up the optimization, we make use of caching and parallel computing. Specifically, the fitness value of each evaluated chromosome is stored in memory and pulled out when a chromosome with the same set of genes is encountered, and unseen (not cached) individuals are assessed in parallel.

X. Experimental Results

In this section, we evaluate the performance of the proposed techniques. All the experiments are conducted on a GNU/Linux machine equipped with 16 processors Intel Xeon E5520 2.27 GHz and 24 GB of RAM. Parallel computing is utilized only in the experiments reported in Sec. X-C.

A. Configuration

We consider a 45-nm technological process and rely on the 45-nm standard-cell library published and maintained by NanGate [25]. The effective channel length and gate-oxide thickness are assumed to have nominal values equal to 22.5 nm and 1 nm, respectively. Following the information about process variation reported by ITRS [26], we assume that each process parameter can deviate up to 12% of its nominal value, and this percentage is treated as three standard deviations. The corresponding probabilistic model is the one described in Sec. VI-B. Regarding the correlation function in (6), the weight coefficient ϖ is set to 0.5, and the length-scale parameters ℓ_{SE} and ℓ_{OU} are set to half the size of the die (see the next paragraph). The model order reduction is set to preserve 95% of the variance of the problem (see also Appendix C). The tuning parameter γ in (11) is set to 0.25.

Heterogeneous platforms and periodic applications are generated randomly using TGFF [27] in such a way that the execution time of each task is uniformly distributed between 10 and 30 ms, and its dynamic power between 6 and 20 W. The floorplans of the platforms are regular grids wherein each processing element occupies $2 \times 2\,\mathrm{mm}^2$. Thermal RC circuits—which are essentially pairs of a thermal capacitance matrix $\mathbf{C}$ and a thermal conductance $\mathbf{G}$ matrix needed in the equations given in Sec. VII-B—are constructed using the HotSpot thermal model [21]. The granularity of power and temperature profiles, that is, Δt in Sec. VII-C and Appendix A, is set to 1 ms; in practice, Δt should be set to a value that is reasonable for the problem at hand. The stopping condition in Algorithm 3 is a decrease of the normalized root-mean-square error between two successive temperature profiles smaller than 1%, which typically requires 3–5 iterations.

The leakage model needed for the calculation of $\mathbf{P}_{\mathrm{stat}}(\mathbf{u}, \mathbf{Q})$ in Algorithm 3 is based on SPICE simulations of a series of CMOS invertors taken from the NanGate cell library and configured according to the high-performance 45-nm PTM [28]. The simulations are performed on a fine-grained and sufficiently broad three-dimensional grid comprising the effective channel length, gate-oxide thickness, and temperature; the results are tabulated. The interpolation facilities of MATLAB [23] are then utilized whenever we need to evaluate the leakage power for a particular point within the range of the grid. The output of the constructed leakage model is scaled up to account for about 40% of the total power dissipation [2].

B. Probabilistic Analysis

Our objective here is to study the accuracy and speed of the proposed solutions. Since the optimization procedure described in Sec. IX embraces all the techniques developed throughout the paper, we shall perform the assessment directly in the design-space-exploration context. In other words, we do not consider temperature analysis or reliability analysis as a separate uncertainty quantification problem in our experiments and shall focus on the quantity of interest given in (29). This quantity plays the key role as the objective function in (25) and the constraints in (27) and (28) are entirely based on it.

We shall compare our performance with the performance of Monte Carlo (MC) sampling. The operations performed by the MC-based approach for one sample are exactly the same as those performed by our technique for one quadrature point. The only difference is that no reduction of any kind is undertaken prior to MC sampling. In other words, the MC-based approach samples the *original* model and, hence, does not compromise any resulting accuracy. The number of MC samples is set to 10^4, which is a practical assumption that conforms to the experience from the literature [6], [8], [9], [10] and to the theoretical estimates given in [7]. Hence, we consider this setup of MC sampling to be a paragon of accuracy.

The results concerning accuracy are displayed in Table I where we consider a quad-core platform, i.e., $n_{\mathrm{p}} = 4$, with ten randomly generated applications and vary the level of

Table I
ASSESSMENT OF THE ACCURACY

$l_{\mathcal{C}}$	$n_{\mathcal{C}}$	$n_{\mathcal{Q}}$	$\epsilon_{\mathcal{E}}$, KLD	$\epsilon_{\mathcal{Q}}$, KLD	$\epsilon_{\mathcal{T}}$, KLD
1	3	5	0.0415	0.1935	0.3390
2	10	21	0.0085	0.0187	0.0320
3	22	69	0.0022	0.0025	0.0046
4	49	193	0.0017	0.0024	0.0033
5	111	589	0.0016	0.0027	0.0037

Table II
ASSESSMENT OF THE COMPUTATIONAL SPEED

n_{p}	n_{ξ}	$n_{\mathcal{C}}$	$n_{\mathcal{Q}}$	Time, s	Speedup, times
2	4	19	57	0.18	175.44
4	6	22	69	0.26	144.93
8	8	27	81	0.73	123.46
16	10	30	93	1.28	107.53
32	10	33	101	2.23	99.01

polynomial expansions $l_{\mathcal{C}}$ from one to five. The errors for the three components of $\boldsymbol{\vartheta} = (\mathcal{E}, \mathcal{Q}, \mathcal{T})$ are denoted by $\epsilon_{\mathcal{E}}$, $\epsilon_{\mathcal{Q}}$, and $\epsilon_{\mathcal{T}}$, respectively. Each error indicator shows the distance between the empirical probability distributions produced by our approach and the ones produced by MC sampling, and the measure of this distance is the popular Kullback–Leibler divergence (KLD) wherein the results of MC sampling are treaded as the "true" ones. The KLD takes nonnegative values and attains zero only when two distributions are equal almost everywhere [14]. In general, the errors decrease as $l_{\mathcal{C}}$ increases. This trend, however, is not monotonic for expansions of high levels (see $\epsilon_{\mathcal{Q}}$ and $\epsilon_{\mathcal{T}}$ for $l_{\mathcal{C}} = 5$). The observation can be ascribed to the random nature of sampling and the fact that the reduction procedures, which we undertake to gain speed, might impose limitations on the accuracy that can be attained by polynomial expansions. Table I also contains the numbers of polynomial terms $n_{\mathcal{C}}$ and quadrature points $n_{\mathcal{Q}}$ corresponding to each value of $l_{\mathcal{C}}$. We also performed the above experiment for platforms with fewer/more processing elements; the observations were similar to the ones in Table I.

Based on Table I, we consider the results delivered by third-level polynomial expansions, where the KLD drops to the third decimal place for all quantities, to be sufficiently accurate, and, therefore, we fix $l_{\mathcal{C}} = l_{\mathcal{Q}} = l = 3$ (recall the notation in the last paragraph of Sec. VI-C1) for the rest of the experiments.

Table II displays the time needed to perform one characterization of $\boldsymbol{\vartheta}$ for the number of processing elements n_{p} swept from 2 to 32. It can be seen that the computational time ranges from a fraction of a second to around two seconds. More importantly, Table II provides information about a number of complementary quantities that are of high interest for the user of the proposed techniques, which we discuss below.

The primary quantity to pay attention to is the number of random variables n_{ξ} preserved after the reduction procedure described in Sec. VI-B and Appendix C. Without this reduction, n_{ξ} would be $2n_{\mathrm{p}}$ as there are two process parameters per processing element. It can be seen that there is no reduction for the dual-core platform while around 80% of the stochastic dimensions have been eliminated for the platform with 32 cores. In addition, one can note that n_{ξ} is the same for the last two platforms. The magnitude of reduction is solely determined by the correlation patterns assumed (see Sec. VI-B) and the floorplans of the considered platforms.

Another important quantity displayed in Table II is the number of quadrature nodes $n_{\mathcal{Q}}$. This number is the main indicator of the computational complexity of our probabilistic analysis: it equals to the number of times Algorithm X in Algorithm 1 is executed to construct a polynomial expansion of (29) needed for the evaluation of the fitness function. It can be seen that $n_{\mathcal{Q}}$ is very low. To illustrate this, the last column of Table II shows the speedup of our approach with respect to 10^4 MC. Our solution is faster by approximately 100–200 times while delivering highly accurate results as discussed earlier. It should be noted that the comparison has been drawn based on the number of evaluation points rather than on the actual time since the relative cost of other computations is negligible.

To conclude, the proposed solutions to temperature and reliability analyses under process variation have been assessed using the compositional quantity of interest given in (29). The results shown in Table I and Table II allow us to conclude that our approach is both accurate and computationally efficient.

C. Probabilistic Optimization

In this subsection, we report the results of the optimization procedure formulated in Sec. IX. To reiterate, the objective is to minimize energy as shown in (25) while satisfying a set of constraints on the application period, maximal temperature, and minimal lifetime as shown in (26), (27), and (28), respectively. We employ a genetic algorithm for optimization. The population is evaluated in parallel using 16 processors; this job is delegated to the parallel computing toolbox of MATLAB [23].

The goal of this experiment is to justify the following assertion: reliability analysis has to account for the effect of process variation on temperature. To this end, for each problem (a pair of a platform and an application), we shall run the optimization procedure twice: once using the setup that has been described so far and once making the objective in (25) and the constraints in (27) and (28) deterministic. To elaborate, the second run assumes that temperature is deterministic and can be computed using the nominal values of the process parameters. Consequently, only one simulation of the system is needed in the deterministic case to evaluate the fitness function, and (25), (27), and (28) become, respectively,

$$\min_{\mathcal{S}} \mathcal{E}(\mathcal{S}), \quad \mathcal{Q}(\mathcal{S}) \geq q_{\max}, \quad \text{and} \quad \mathcal{T}(\mathcal{S}) \leq \tau_{\min}.$$

We consider platforms with $n_{\mathrm{p}} = 2$, 4, 8, 16, and 32 cores. Ten applications with the number of tasks $n_{\mathrm{v}} = 20\, n_{\mathrm{p}}$ (that is, 40 tasks for 2 cores up to 640 tasks for 32 cores) are randomly generated for each platform; thus, 50 problems in total. The floorplans of the platforms and the task graphs of the applications, including the execution time and dynamic power consumption of each task on each core, are available online at [29]. ρ_{burn} and ρ_{wear} in (27) and (28), respectively, are set to 0.01. Due to the diversity of the problems, $t_{\max}$, $q_{\max}$, and $\tau_{\min}$ are found individually for each problem, ensuring that they make sense for the subsequent optimization. For instance, $q_{\max}$ was found within the range 90–120°C. Note, however, that these three parameters stay the same for both the probabilistic and deterministic variants of the optimization.

The obtained results are reported in Table III, and the most important message is in the last column. *Failure rate* refers

Table III
STOCHASTIC VS. DETERMINISTIC OPTIMIZATION

n_p	Stochastic Time, min	Deterministic Time, min	Deterministic Failure rate, %
2	1.07	0.67	40
4	5.38	1.99	60
8	16.65	3.89	70
16	56.23	7.54	100
32	341.08	9.26	100

to the ratio of the solutions produced by the deterministic optimization that, after being reevaluated using the probabilistic approach (i.e., after taking process variation into account), have been found to be violating the probabilistic constraints given in (27) and/or (28). To give an example, for the quad-core platform, six out of ten schedules proposed by the deterministic approach violate the constraints on the maximal temperature and/or minimal lifetime when evaluated considering process variation. The more complex the problem becomes, the higher values the failure rate attains: with 16 and 32 processing elements (320 and 640 tasks, respectively), all deterministic solutions violate the imposed constraints. Moreover, the difference between the acceptable one percent of burn/wear ($\rho_{\text{burn}} = \rho_{\text{wear}} = 0.01$) and the actual probability of burn/wear was found to be as high as 80% in some cases, which is unacceptable.

In addition, we inspected those few deterministic solutions that had passed the probabilistic reevaluation and observed that the reported reduction of the energy consumption and maximal temperature as well as the reported increase of the lifetime were overoptimistic. More precisely, the predictions produced by the deterministic optimization, which ignores variations, were compared with the expected values obtained when process variation was taken into account. The comparison showed that the expected energy and temperature were up to 5% higher while the expected lifetime was up to 20% shorter than the ones estimated by the deterministic approach. This aspect of the deterministic optimization can mislead the designer.

Consequently, when studying those aspects of electronic systems that are concerned with power, temperature, and reliability, the ignorance of the deteriorating effect of process variation can severely compromise the associated design decisions making them less profitable in the best case and dangerous, harmful in the worst scenario.

Let us now comment on the optimization time shown in Table III. It can be seen that the prototype of the proposed framework takes from about one minute to six hours (utilizing 16 CPUs) in order to perform optimization, and the deterministic optimization is approximately 2–40 times faster. However, the price to pay when relying on the deterministic approach is considerably high as we discussed in the previous paragraphs. It can be summarized as "blind guessing with highly unfavorable odds of succeeding." Consequently, we consider the computational time of our framework to be reasonable and affordable, especially in an industrial setting.

Lastly, we performed experiments also to investigate the impact of the lifetime constraint in (28) on the reduction of the expected energy consumption. To this end, we ran our probabilistic optimization (all 50 problems) without the constraint in (28) and compared the corresponding results with those obtained considering the lifetime constraint. We observed that the expected energy consumption was higher when (28) was taken into account, but the difference vanishes when the complexity of the problems increases. On average, the cost of (28) was below 5% of the expected energy consumption. Without (28), however, no (probabilistic) guarantees on the lifetime of the considered systems can be given.

XI. CONCLUSION

We have presented a number of techniques for uncertainty quantification of electronic systems subjected to process variation. First, we developed a process-variation-aware approach to DSS temperature analysis. Second, we proposed a framework for reliability analysis that seamlessly takes into account the variability of process parameters and, in particular, the effect of process variation on temperature. We drew a comparison with MC sampling, which confirmed the efficiency of our solutions in terms of both accuracy and speed. The low computational demand of our techniques implies that they are readily applicable for practical instantiations inside design-space exploration loops, which was also demonstrated in this work considering an energy-driven probabilistic optimization procedure under reliability-related constraints. We have shown that temperature is to be treaded as a stochastic quantity in order to pursue robustness of electronic system designs.

ACKNOWLEDGMENTS

We would like to thank Paul Constantine from Colorado School of Mines for the valuable discussions regarding uncertainty quantification with polynomial expansions.

APPENDIX

A. Temperature Solution

The technique used to solve the thermal system in (17) belongs to the family of exponential integrators [22], which are known to have good stability properties. More specifically, we rely on a one-step member from that family. In what follows, for compactness, we shall omit $\mathbf{u}$.

Multiplying both sides of (17a) by $e^{-\mathbf{A}t}$ and noting that

$$e^{-\mathbf{A}t}\frac{d\mathbf{s}(t)}{dt} = \frac{d\, e^{-\mathbf{A}t}\mathbf{s}(t)}{dt} + e^{-\mathbf{A}t}\mathbf{A}\mathbf{s}(t),$$

we obtain the exact solution of (17a) over a time interval $\Delta t = t_k - t_{k-1}$ given as follows:

$$\mathbf{s}(t_k) = e^{\mathbf{A}\Delta t}\mathbf{s}(t_{k-1}) + \int_0^{\Delta t} e^{\mathbf{A}(\Delta t - t)}\mathbf{B}\,\mathbf{p}(t_k + t, \mathbf{s}(t_k + t))\, dt.$$

The integral on the right-hand side is approximated by assuming that, within Δt, the power dissipation does not change and is equal to the power dissipation at t_k. Thus, we have

$$\mathbf{s}(t_k) = e^{\mathbf{A}\Delta t}\mathbf{s}(t_{k-1}) + \mathbf{A}^{-1}(e^{\mathbf{A}\Delta t} - \mathbf{I})\mathbf{B}\,\mathbf{p}(t_{k-1}, \mathbf{s}(t_{k-1})),$$

which leads to the recurrence in (18).

B. Probability Transformation

The uncertain parameters $\mathbf{u}$ should be preprocessed in order to extract a set of mutually independent random variables $\boldsymbol{\xi}$. This task is accomplished by virtue of the Nataf transformation. Here we describe the algorithm in brief and refer the interested reader to [17] for additional details.

The transformation has two steps. First, $\mathbf{u} \in \mathbb{R}^{n_\mathrm{u} n_\mathrm{p}}$ are morphed into correlated standard Gaussian variables $\boldsymbol{\zeta} \in \mathbb{R}^{n_\mathrm{u} n_\mathrm{p}}$ using the knowledge of the marginal distributions and correlation matrix of $\mathbf{u}$. Second, the obtained variables are mapped into independent standard Gaussian variables $\boldsymbol{\xi} \in \mathbb{R}^{n_\mathrm{u} n_\mathrm{p}}$ using the eigendecomposition as we show next. Let $\mathbf{K}_\zeta$ be the correlation matrix of $\boldsymbol{\zeta}$. Since any correlation matrix is real and symmetric, $\mathbf{K}_\zeta$ admits the eigendecomposition: $\mathbf{K}_\zeta = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$ (see Sec. VII-B for the notation). $\boldsymbol{\zeta}$ can then be represented as $\boldsymbol{\zeta} = \mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\xi}$ where the vector $\boldsymbol{\xi}$ is standardized and uncorrelated, which is also independent as $\boldsymbol{\xi}$ is Gaussian.

C. Model Order Reduction

In this section, we discuss the model order reduction performed by $\mathbb{T}$ in Sec. VI-B. This reduction is based on the eigendecomposition undertaken in Appendix B. The intuition is that, due to the correlations possessed by $\mathbf{u} \in \mathbb{R}^{n_\mathrm{u} n_\mathrm{p}}$, it can be recovered from a small subset with only n_ξ variables where $n_\xi \ll n_\mathrm{u} n_\mathrm{p}$. Such redundancies can be revealed by analyzing the eigenvalues $\boldsymbol{\lambda} = (\lambda_i)_{i=1}^{n_\mathrm{u} n_\mathrm{p}}$ located on the diagonal of $\boldsymbol{\Lambda}$, which are all nonnegative. Without loss of generality, we let $\lambda_i \geq \lambda_j$ whenever $i < j$ and assume $\|\boldsymbol{\lambda}\|_1 = 1$. Then we can identify the smallest n_ξ such that $\sum_{i=1}^{n_\xi} \lambda_i$ is greater than a certain threshold chosen from the interval $(0, 1]$. When this threshold is sufficiently high (close to one), the rest of the eigenvalues and the corresponding eigenvectors can be dropped as being insignificant, reducing the number of stochastic dimensions to n_ξ. With a slight abuse of notation, we let $\boldsymbol{\xi}$ be the result of the reduction.

D. Spectral Decomposition

Let $H \subset L^2(\Omega, \mathcal{F}, \mathbb{P})$ be the Gaussian Hilbert space [15] spanned by the random variables $\{\xi_i\}_{i=1}^{n_\xi}$ contained in $\boldsymbol{\xi}$ as defined in (4). Since these variables are independent and standard, they form an orthonormal basis in H, and the dimensionality of H is n_ξ. Let $\Psi_{l_\mathcal{C}}(H)$ be the space of n_ξ-variate polynomials over H such that the total order of each polynomial is less or equal to $l_\mathcal{C}$. $\Psi_{l_\mathcal{C}}(H)$ can be constructed as a span of n_ξ-variate Hermite polynomials [3], [16]:

$$\Psi_{l_\mathcal{C}}(H) = \operatorname{span}\left(\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}) :\ \boldsymbol{\alpha} \in \mathcal{A}\left(l_\mathcal{C}\right),\ \boldsymbol{\zeta} \in H^{n_\xi}\}\right)$$

where $\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}_0^{n_\xi}$ is a multi-index,

$$\mathcal{A}\left(l_\mathcal{C}\right) = \left\{\boldsymbol{\alpha} :\ \|\boldsymbol{\alpha}\|_1 := \sum_{i=1}^{n_\xi} |\alpha_i| \leq l_\mathcal{C}\right\}, \tag{30}$$

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}) = \prod_{i=1}^{n_\xi} \psi_{\alpha_i}(\zeta_i), \quad \text{and}$$

ψ_{α_i} is a one-dimensional Hermite polynomial of order α_i, which is assumed to be normalized for convenience. Define $\mathcal{H}_0 := \Psi_0(H)$ (the space of constants) and, for $i \geq 1$,

$$\mathcal{H}_i := \Psi_i(H) \cap \Psi_{i-1}(H)^\perp.$$

The spaces $\mathcal{H}_i$, $i \geq 0$, are mutually orthogonal, closed subspaces of $L^2(\Omega, \mathcal{F}, \mathbb{P})$. Since our scope of interest is restricted to functions of $\boldsymbol{\xi}$ (via (3)), $\mathcal{F}$ is assumed to be generated by $\{\xi_i\}_{i=1}^{n_\xi}$. Then, by the Cameron–Martin theorem,

$$L^2(\Omega, \mathcal{F}, \mathbb{P}) = \bigoplus_{i=0}^{\infty} \mathcal{H}_i,$$

which is known as the Wiener chaos decomposition. Thus, any $\vartheta \in L^2(\Omega, \mathcal{F}, \mathbb{P})$ admits an expansion with respect to the polynomial basis. Define the associated linear operator by

$$\mathcal{C}_{l_\mathcal{C}}^{n_\xi}[\vartheta] := \sum_{\boldsymbol{\alpha} \in \mathcal{A}(l_\mathcal{C})} \langle \vartheta, \psi_{\boldsymbol{\alpha}} \rangle\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{31}$$

Remark 2 is relevant for the present discussion. The spectral decomposition in (31) converges in mean square to ϑ as $l_\mathcal{C} \to \infty$. We shall refer to $l_\mathcal{C}$ as the level of the chaotic expansion. The cardinality of $\mathcal{A}\left(l_\mathcal{C}\right)$ in (30) is

$$\#\mathcal{A}\left(l_\mathcal{C}\right) := \binom{l_\mathcal{C} + n_\xi}{n_\xi}.$$

Denote by dF_ξ the probability measure induced on $\mathbb{R}^{n_\xi}$ by $\boldsymbol{\xi}$, which is standard Gaussian given by

$$dF_\xi(\mathbf{x}) = (2\pi)^{-n_\xi/2}\, e^{-\|\mathbf{x}\|_2^2/2}\, d\mathbf{x}. \tag{32}$$

The inner product in (31) projects ϑ onto the corresponding polynomial subspaces and is given by

$$\hat{\vartheta}_{\boldsymbol{\alpha}} = \langle \vartheta, \psi_{\boldsymbol{\alpha}} \rangle = \int_{\mathbb{R}^{n_\xi}} \vartheta(\mathbb{T}[\mathbf{x}])\, \psi_{\boldsymbol{\alpha}}(\mathbf{x})\, dF_\xi(\mathbf{x}). \tag{33}$$

Recall that ϑ is a function of $\mathbf{u}$, and $\mathbb{T}$, defined in (3), bridges $\mathbf{u}$ with $\boldsymbol{\xi}$. Lastly, we note that $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}_0^{n_\xi}\}$ are orthonormal with respect to dF_ξ: $\langle \psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}} \rangle = \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ where $\boldsymbol{\alpha} = (\alpha_i)$ and $\boldsymbol{\beta} = (\beta_i)$ are two arbitrary multi-indices, $\delta_{\boldsymbol{\alpha}\boldsymbol{\beta}} := \prod_i \delta_{\alpha_i \beta_i}$, and δ_{ij} is the Kronecker delta.

E. Numerical Integration

As shown in (33), each $\hat{\vartheta}_{\boldsymbol{\alpha}}$ is an n_ξ-dimensional integral, which, in general, should be computed numerically. This task is accomplished by virtue of an adequate n_ξ-dimensional quadrature, which is essentially a set of n_ξ-dimensional points accompanied by scalar weights. Since we are interested in integration with respect to the standard Gaussian measure over $\mathbb{R}^{n_\xi}$ (see (32)), we shall rely on the Gauss–Hermite family of quadrature rules [3], which is a subset of a broader family known as Gaussian quadratures. The construction of high-dimensional rules should be undertaken with a great care as, without special treatments, the number of points grows exponentially. In what follows, we address this crucial aspect.

Let $f : \mathbb{R}^{n_\xi} \to \mathbb{R}$ and define the quadrature-based approximation of the integral of f by the linear functional

$$\mathcal{Q}_{l_\mathcal{Q}}^{n_\xi}[f] := \sum_{i=1}^{n_\mathcal{Q}} f(\mathbf{x}_i)\, w_i$$

where $\{(\mathbf{x}_i \in \mathbb{R}^{n_\xi}, w_i \in \mathbb{R})\}_{i=1}^{n_\mathcal{Q}}$ are the points and weights of the chosen quadrature. Remark 2 applies in this context as well. The subscript $l_\mathcal{Q} \in \mathbb{N}_0$ denotes the level of the rule, which is its index in the corresponding family of rules with increasing precision. The precision refers to the maximal total order of polynomials that the quadrature integrates exactly. The number of points $n_\mathcal{Q}$ can be deduced from the pair $(n_\xi, l_\mathcal{Q})$, which we shall occasionally emphasize by writing $n_\mathcal{Q}(n_\xi, l_\mathcal{Q})$. For the Gauss–Hermite quadrature rules in one dimension, we

have that $n_{\mathcal{Q}} = l_{\mathcal{Q}} + 1$ and the precision is $2n_{\mathcal{Q}} - 1$ [19] or, equivalently, $2l_{\mathcal{Q}} + 1$, which is a remarkable property of Gaussian quadratures.

The foundation of a multidimensional rule $\mathcal{Q}_{l_{\mathcal{Q}}}^{n_\xi}$ is a set of one-dimensional counterparts $\{\mathcal{Q}_i^1\}_{i=0}^{l_{\mathcal{Q}}}$. A straightforward construction is the tensor product of n_ξ copies of $\mathcal{Q}_{l_{\mathcal{Q}}}^1$:

$$\mathcal{Q}_{l_{\mathcal{Q}}}^{n_\xi} = \bigotimes_{i=1}^{n_\xi} \mathcal{Q}_{l_{\mathcal{Q}}}^1, \tag{34}$$

which is referred to as the full-tensor product. In this case, $n_{\mathcal{Q}}(n_\xi, l_{\mathcal{Q}}) = n_{\mathcal{Q}}(1, l_{\mathcal{Q}})^{n_\xi}$, i.e., the growth of the number of points is exponential. Moreover, it can be shown that most of the points obtained via this construction are an excess as the full-tensor product does not take into account the fact that the integrands under consideration are polynomials whose total order is constrained according to a certain strategy.

An alternative construction is the Smolyak algorithm [3], [16]. Intuitively, the algorithm combines $\{\mathcal{Q}_i^1\}_{i=0}^{l_{\mathcal{Q}}}$ such that $\mathcal{Q}_{l_{\mathcal{Q}}}^{n_\xi}$ is tailored to be exact only for a specific polynomial subspace. Define $\Delta_0 := \mathcal{Q}_0^1$ and $\Delta_i := \mathcal{Q}_i^1 - \mathcal{Q}_{i-1}^1$ for $i \geq 1$. Then Smolyak's approximating formula is

$$\mathcal{Q}_{l_{\mathcal{Q}}}^{n_\xi} = \bigoplus_{\boldsymbol{\alpha} \in \mathcal{A}(l_{\mathcal{Q}})} \Delta_{\alpha_1} \otimes \cdots \otimes \Delta_{\alpha_{n_\xi}}. \tag{35}$$

In the original (isotropic) formulation of the Smolyak algorithm, $\mathcal{A}(l_{\mathcal{Q}})$ is the same as the one defined in (30); the resulting sparse grid is exact for polynomials of total order $2l_{\mathcal{Q}} + 1$ (analogous to the integration in one dimension). Note that, although we use the same notation in (34) and (35), the two constructions are generally different. (The latter reduces to the former if $\mathcal{A}(l_{\mathcal{Q}})$ is set to $\{\boldsymbol{\alpha} : \max_i \alpha_i \leq l_{\mathcal{Q}}\}$.) It can be seen that the construction in (35) is a summation of cherry-picked tensor products of one-dimensional quadrature rules. Equation (35) is well suited for grasping the structure of the resulting sparse grids; more implementation-oriented versions of the Smolyak formula can be found in the cited literature.

REFERENCES

[1] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*. Springer, 2010.

[2] Y. Liu, R. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy," in *DATE*, 2007, pp. 1526–1531.

[3] O. Maître and O. Knio, *Spectral Methods for Uncertainty Quantification With Applications to Computational Fluid Dynamics*. Springer, 2010.

[4] I. Ukhov, M. Bao, P. Eles, and Z. Peng, "Steady-state dynamic temperature analysis and reliability optimization for embedded multiproc. sys." in *DAC*, 2012, pp. 197–204.

[5] (2014, January) JEDEC failure mechanisms and models for semiconductor devices. JEDEC. [Online]. Available: http://www.jedec.org/

[6] I. Ukhov, P. Eles, and Z. Peng, "Probabilistic analysis of power and temperature under process variation for electronic system design," *Trans. CAD Integr. Circuits Syst.*, vol. 33, pp. 931–944, June 2014.

[7] I. Díaz-Emparanza, "Is a small Monte Carlo analysis a good analysis?" *Statistical Papers*, vol. 43, pp. 567–577, October 2002.

[8] Y.-M. Lee and P.-Y. Huang, "An efficient method for analyzing on-chip thermal reliability considering process variations," *ACM Trans. on Design Automation of Electronic Systems*, vol. 18, pp. 41:1–41:32, July 2013.

[9] D.-C. Juan, Y.-L. Chuang, D. Marculescu, and Y.-W. Chang, "Statistical thermal modeling and optimization considering leakage power variations," in *DATE*, 2012, pp. 605–610.

[10] Y. Xiang, T. Chantem, R. Dick, X. Hu, and L. Shang, "System-level reliability modeling for MPSoCs," in *CODES+ISSS*, 2010, pp. 297–306.

[11] A. Das, A. Kumar, and B. Veeravalli, "Reliability-aware platform-based design methodology for energy-efficient multiprocessor systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2014, to be published.

[12] A. Das, R. Shafik, G. Merrett, B. Al-Hashimi, A. Kumar, and B. Veeravalli, "Reinforcement learning-based inter- and intra-application thermal optimization for lifetime improvement of multic. sys." in *DAC*, 2014.

[13] A. Das, A. Kumar, and B. Veeravalli, "A survey of lifetime reliability-aware system-level design techniques for embedded multip. sys." *IEEE Comput.*, 2014, to be published.

[14] R. Durrett, *Probability: Theory and Examples*. Cambridge University Press, 2010.

[15] S. Janson, *Gaussian Hilbert Spaces*. Cambridge University Press, 1997.

[16] M. Eldred, C. Webster, and P. Constantine, "Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos," in *AIAA Non-deterministic Approaches Conference*, 2008.

[17] H. Li, Z. Lü, and X. Yuan, "Nataf transformation-based point estimate method," *Chinese Science Bulletin*, pp. 2586–2592, September 2008.

[18] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization," in *Symp. Quality of Electronic Design*, 2005, pp. 516–521.

[19] F. Heiss and V. Winschel, "Likelihood approximation by numerical integration on sparse grids," *J. Econometrics*, vol. 144, pp. 62–80, 2008.

[20] F. Nobile, R. Tempone, and C. Webster, "An anisotropic sparse grid stochastic collocation method for elliptic PDEs with random input data," *SIAM J. Numerical Analysis*, vol. 46, pp. 2411–2442, May 2008.

[21] K. Skadron, M. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Architecture and Code Optimization*, vol. 1, pp. 94–125, March 2004.

[22] M. Hochbruck and A. Ostermann, "Exponential integrators," *Acta Numerica*, vol. 19, pp. 209–286, May 2010.

[23] (2014, June) MATLAB. MathWorks. [Online]. Available: http://www.mathworks.com/products/matlab/

[24] T. Adam, K. Chandy, and J. Dickson, "A comparison of list schedules for parallel processing systems," *Commun. ACM*, vol. 17, pp. 685–690, December 1974.

[25] (2014, January) The NanGate 45nm open cell library. NanGate. [Online]. Available: http://www.nangate.com/

[26] (2014, January) ITRS reports. ITRS. [Online]. Available: http://www.itrs.net/reports.html

[27] R. Dick, D. Rhodes, and W. Wolf, "TGFF: Task graphs for free," in *CODES/CASHE*, March 1998, pp. 97–101.

[28] (2014, January) PTM. Nanoscale Integration and Modeling Group at Arizona State University. [Online]. Available: http://ptm.asu.edu/

[29] (2014, June) Supplementary materials related to the experimental results. Embedded Systems Laboratory at Linköping University. [Online]. Available: http://www.ida.liu.se/~ivauk83/research/TRAO/

Ivan Ukhov received the B.Sc. and M.Sc. degrees (Hons) in computer science and computer engineering from St. Petersburg State Polytechnical University, St. Petersburg, Russia, in 2008 and 2010, respectively, and is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Linköping University, Linköping, Sweden, focusing on uncertainty quantification for electronic system design.

Petru Eles (M'99) is currently a Professor of embedded computer systems with the Department of Computer and Information Science, Linköping University, Linköping, Sweden. He has published a large number of technical papers and several books in the area of embedded systems.

He received two best paper awards at the European Design Automation Conferences, a best paper award at the Design Automation and Test in Europe Conference, and a best paper award at the International Conference on Hardware/Software Codesign and System Synthesis.

Zebo Peng (M'91–SM'02) is currently a Professor of computer systems, the Director of the Embedded Systems Laboratory, and the Chairman of the Division for Software and Systems with the Department of Computer Science, Linköping University, Linköping, Sweden. He has published over 300 technical papers and four books in various topics related to embedded systems.

He has received four best paper awards and one best presentation award in major international conferences.