Journal of Mathematics in Industry
a SpringerOpen Journal

RESEARCH Open Access

Hybrid importance sampling Monte Carlo approach for yield estimation in circuit design

Anuj K. Tyagi[1*], Xavier Jonsson[2], Theo G.J. Beelen[1] and Wil H.A. Schilders[1]

*Correspondence: a.k.tyagi@tue.nl
[1]Department of Mathematics and Computer Science, University of Technology, Eindhoven, The Netherlands
Full list of author information is available at the end of the article

Abstract

The dimension of transistors shrinks with each new technology developed in the semiconductor industry. The extreme scaling of transistors introduces important statistical variations in their process parameters. A large digital integrated circuit consists of a very large number (in millions or billions) of transistors, and therefore the number of statistical parameters may become very large if mismatch variations are modeled. The parametric variations often cause to the circuit performance degradation. Such degradation can lead to a circuit failure that directly affects the yield of the producing company and its fame for reliable products. As a consequence, the failure probability of a circuit must be estimated accurately enough. In this paper, we consider the Importance Sampling Monte Carlo method as a reference probability estimator for estimating tail probabilities. We propose a Hybrid ISMC approach for dealing with circuits having a large number of input parameters and provide a fast estimation of the probability. In the Hybrid approach, we replace the expensive to use circuit model by its cheap surrogate for most of the simulations. The expensive circuit model is used only for getting the training sets (to fit the surrogates) and near to the failure threshold for reducing the bias introduced by the replacement.

Keywords: Yield; Failure probability; Monte Carlo; Hybrid importance sampling Monte Carlo; Dimension reduction; Exploration phase; Estimation phase; Kriging model; Probability estimator

1 Introduction

Due to the continuously increase of the number of individual components on an Integrated Circuit (IC) the probability of a bad working IC will increase dramatically, see [1, 2]. This can simply be illustrated by the example in [2], where an IC with S "identical" components (each having a failure probability p_{fail}) has a rather large probability of $P_{fail} = 1 - (1 - p_{fail})^S$ on break-down, even p_{fail} is considerable small (i.e., being a rare event). For example, consider a 256 Mbit SRAM circuit, having 256 million "identical" bit cells. Then to guarantee a failure probability of 1% for this circuit (i.e. $P_{fail} = 0.01$ with $S = 256 \times 10^6$), it is required that $p_{fail} < 3.9 \times 10^{-11}$, being a rare event indeed. Notice that the yield Y of an IC is closely related to the failure probability P_{fail} and can be expressed as $Y = 1 - P_{fail} = (1 - p_{fail})^S$. Thus, the yield Y of an IC is estimated by using the failure probability p_{fail} of its component.

We consider Monte Carlo (MC) techniques [3] for estimating the failure probability p_{fail}. The standard Monte Carlo produces an estimator $\hat{p}_{fail} = k/n$ for the true probability p_{fail} by running the simulator n times with independent random inputs and counting the k

occurrences of the 'fail' event. Notice that $n\hat{p}_{fail} \sim \text{Bin}(n, p_{fail})$ follows a binomial law with probability p_{fail} of getting success out of n trials. The useful properties of the estimator $\hat{p}_{fail}$ are its unbiasedness i.e., $\mathbb{E}(\hat{p}_{fail}) = p_{fail}$ and its independency on the dimension d of the random vector $\mathbf{X}$. However, the variance of the estimator $\hat{p}_{fail}$ is given by $\text{Var}(\hat{p}_{fail}) = p_{fail}(1 - p_{fail})/n$, which can be (relatively) large for small p_{fail} and limited number n of MC runs. Using the 'normal approximation' of the binomial distribution, the 95% confidence interval for (small) $\hat{p}_{fail}$ is estimated to be $\pm 1.96/\sqrt{n\hat{p}_{fail}}$. So, to determine $\hat{p}_{fail}$ in the range 10^{-11} with an accuracy of $\pm 10\%$ with 95% confidence level, one needs about 4×10^{13} MC runs, which is intractable in industry even with the fastest computer simulations.

To overcome the drawback of the standard MC method, a variance reducing Importance Sampling Monte Carlo (ISMC) technique is proposed in [2]. There it was shown that a reduction of several orders can be achieved, from 4×10^{13} to at most few thousands runs. However, when we estimate the probability of failure, it is done at some fixed environmental parameters (such as temperature, supply voltage, and process corners). These parameters add multiple levels of complexity. For instance, the failure probability must be computed for a complete range of working temperatures. The complexity grows exponentially when the other dimensions are combined. For complex systems one usually can only afford a very limited number (say, in hundreds) of simulations, and therefore the ISMC technique remains unattractive. In [1], a model based ISMC approach has been proposed for estimating rare circuit events. In the model based approach the circuit model is replaced by a surrogate which is much faster to evaluate, the circuit model is only used for drawing training samples which are used to build a surrogate. Usually, the number of training samples is much smaller than the total number of MC simulations for estimating the probability. Hence, the overall computational cost is reduced. Nevertheless, it is often difficult or even impossible to quantify the error made by such a substitution. There is another model based approach proposed in [4] which introduces a statistical blockade approach. In this approach one draws a large number of samples from a surrogate model initially, find the samples which belong to the tail region, and replace them by the true responses. The authors use a linear classifier saying that such classifier is enough for SRAM bitcells. However, our goal is to address the large circuits (such as analog IPs (Intellectual Properties)). Our experiments show that a linear model does not work for such large circuits and it is difficult to fit a surrogate (nonlinear) model that is accurate in the tail so that one can classify the samples that really belong to the tail region. The other limitation of both of the above model based approaches is that they do not address the dimensionality issue of the problem.

In this paper, we propose a Hybrid Importance Sampling Monte Carlo (HISMC) approach for estimating small failure probability. This approach is a modification of the model based approach proposed in [1] and can be used for large dimensional circuit problems. The idea is to only use the expensive circuit model[a] (for a small portion of the overall samples used to estimate the probability) close to the failure threshold and the surrogate is used for the remaining samples that are reasonably away from the failure threshold. The use of these small number of samples of the circuit model can prevent loss of accuracy. The Kriging model [5, 6] is used as a surrogate of the circuit model because it inherits a solid mathematical foundation with several useful properties including interpolation of the training data and a closed formula for approximating the prediction error known as a Kriging variance. The latter is useful for improving the Kriging model near the failure

threshold as well as for selecting the samples near to the failure threshold for which the circuit model is to be used. Our experience with the circuits shows the Kriging model works well up to 35 input variables.

This paper is organised as follows. We start with the reference method mean-shift ISMC approach in Sect. 2. Then we introduce a surrogate modelling technique in Sect. 3 that combines a feature selection method and the Kriging model. Using this surrogate technique we present our HISMC approach in Sect. 4. Finally, the results are shown in Sect. 5 and a conclusion is made in Sect. 6.

2 The importance sampling Monte Carlo method

2.1 General framework

Let $\mathbf{x} \in \mathbb{R}^d$ be a vector of d process parameters, which is a realization of the random vector (r.v.) $\mathbf{X}$ with probability density function (pdf) $g(\mathbf{x})$, and let $H(\mathbf{x})$ be a corresponding response[b] of the circuit under examination. The mathematical equation of the failure probability $\mathrm{p}_{\mathrm{fail}} = \mathbb{P}(H(\mathbf{x}) \geq \gamma)$ is given by

$$\mathrm{p}^{+}_{\mathrm{fail}}(\gamma) = \mathbb{E}_g[\mathbb{1}_{\{H(\mathbf{X})\geq\gamma\}}] = \int \mathbb{1}_{\{H(\mathbf{x})\geq\gamma\}} g(\mathbf{x})\, d\mathbf{x}, \tag{2.1}$$

where subscript g means that the expectation is taken with respect to the pdf $g(\mathbf{x})$, γ is a given failure threshold and $\mathbb{1}_{\{H(\mathbf{x})\geq\gamma\}}$ is an indicator function that gives the value 1 if $H(\mathbf{x} \geq \gamma)$, 0 otherwise.

We assume that the (failure) region of interest lies on the upper tail of the output distribution. This is without loss of generality, because any lower tail can be converted to the upper tail by replacing $H(\mathbf{X}) = -H(\mathbf{X})$. Therefore, the probability $\mathbb{P}[H(\mathbf{X}) \leq \gamma']$ can be converted to $\mathrm{p}^{-}_{\mathrm{fail}}(\gamma') = \mathbb{P}[-H(\mathbf{X}) \geq -\gamma']$ for some give failure threshold γ' on the lower tail of the distribution. Hence, it is sufficient to estimate the probability for the upper tail and hereafter we will simply write $\mathrm{p}_{\mathrm{fail}}$ instead of $\mathrm{p}^{+}_{\mathrm{fail}}(\gamma)$.

Assume that we have another density f such that $\mathbb{1}_{\{H(\mathbf{x})\geq\gamma\}} g(\mathbf{x}) > 0 \implies f(\mathbf{x}) > 0$, we say g is absolutely continuous with respect to f. Then we can write (2.1) as

$$\begin{aligned}
\mathbb{E}_g[\mathbb{1}_{(H(\mathbf{X})\geq\gamma)}] &= \int \mathbb{1}_{(H(\mathbf{x})\geq\gamma)} g(\mathbf{x})\, d\mathbf{x} \\
&= \int \mathbb{1}_{(H(\mathbf{x})\geq\gamma)} \mathcal{L}(\mathbf{x}) f(\mathbf{x})\, d\mathbf{x} \\
&= \mathbb{E}_f\big[\mathbb{1}_{(H(\mathbf{Y})\geq\gamma)} \mathcal{L}(\mathbf{Y})\big],
\end{aligned} \tag{2.2}$$

where $\mathbf{Y}$ is a r.v. generated from the new pdf $f(\mathbf{x})$ and $\mathcal{L}(\mathbf{x}) = g(\mathbf{x})/f(\mathbf{x})$ if $f > 0$ and $\mathcal{L}(\mathbf{x}) = 0$ otherwise, is a likelihood ratio between two densities. The ISMC estimator is then given by

$$\hat{\mathrm{p}}^{\mathrm{IS}}_{\mathrm{fail}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{(H(\mathbf{Y}_i)\geq\gamma)} \mathcal{L}(\mathbf{Y}_i), \tag{2.3}$$

where $\mathbf{Y}_i$'s are N independent and identically distributed (iid) random samples generated from $f(\mathbf{x})$.

The $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{IS}}$ is unbiased [7] with the variance

$$\sigma_{\mathrm{IS}}^2 = \frac{1}{N} \mathrm{Var}_f\left(\mathbb{1}_{(H(\mathbf{Y})\geq\gamma)}\mathcal{L}(\mathbf{Y})\right), \tag{2.4}$$

where

$$\mathrm{Var}_f\left(\mathbb{1}_{(H(\mathbf{Y})\geq\gamma)}\mathcal{L}(\mathbf{Y})\right) = \mathbb{E}_g\left[\left(\mathbb{1}_{\{H(\mathbf{X})\geq\gamma\}}\right)^2\frac{g(\mathbf{X})}{f(\mathbf{X})}\right] - \mathrm{p}_{\mathrm{fail}}^2. \tag{2.5}$$

From a practical point of view, one has to find the 'best' pdf $f(\mathbf{x})$ in order to maximize the accuracy of the estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{IS}}$. One of the ways to find such $f(\mathbf{x})$ is by minimizing the variance σ_{IS}^2.

The work in this paper is an additional contribution to the developments at Mentor Graphics where a mean-shift ISMC technique (see Sect. 2.2) is being used, assuming that the original input distributions can be transformed into a Gaussian distribution. In this context, the importance density is found by shifting the mean of the original density to the area of interest. We use the same technique as a reference approach. The study of other ISMC techniques is out of scope of this paper.

2.2 The mean-shift approach

In this paper, we consider a particular case where the original pdf $g(\mathbf{x})$ is Gaussian with mean $\mathbf{0}$ and variance $\mathbf{I}$, i.e., $g(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We define the importance density $f(\mathbf{x}) = g^{\boldsymbol{\theta}}(\mathbf{x})$ with $g^{\boldsymbol{\theta}}(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\theta}, \mathbf{I})$ parameterized by its mean $\boldsymbol{\theta} \in \mathbb{R}^d$ (see, [2]), in other words $g^{\boldsymbol{\theta}}(\mathbf{x}) = g(\mathbf{x} - \boldsymbol{\theta})$. Then the likelihood ratio $\mathcal{L}(\mathbf{x})$ becomes

$$\mathcal{L}(\mathbf{x}) = \frac{g(\mathbf{x})}{g^{\boldsymbol{\theta}}(\mathbf{x})} = e^{-\boldsymbol{\theta}\mathbf{x}+\frac{1}{2}|\boldsymbol{\theta}|^2} \tag{2.6}$$

and the relation between the random vectors $\mathbf{X}$ and $\mathbf{Y}$ is

$$\mathbf{Y} = \mathbf{X} + \boldsymbol{\theta}. \tag{2.7}$$

Using (2.6) and (2.7), the ISMC probability estimator (2.3) can then be written as

$$\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{IS}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{(H(\mathbf{X}_i+\boldsymbol{\theta})\geq\gamma)}\, e^{-\boldsymbol{\theta}\mathbf{X}_i-\frac{1}{2}|\boldsymbol{\theta}|^2}, \tag{2.8}$$

where $\mathbf{X}_i$'s are N iid random vectors with density $g(\mathbf{x})$.

Furthermore, the second moment (first term in the right hand side of (2.5)) can also be written in a simplified way

$$\mathbb{E}_g\left[\mathbb{1}_{\{H(\mathbf{X})\geq\gamma\}}\,\frac{g(\mathbf{X})}{g^{\boldsymbol{\theta}}(\mathbf{X})}\right] = \mathbb{E}_g\left[\mathbb{1}_{\{H(\mathbf{X})\geq\gamma\}}\, e^{-\boldsymbol{\theta}\mathbf{X}+\frac{1}{2}|\boldsymbol{\theta}|^2}\right] =: \nu(\boldsymbol{\theta}). \tag{2.9}$$

Recalling that for maximizing the accuracy of the probability estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{IS}}$ one has to find the pdf $g^{\boldsymbol{\theta}}(\mathbf{x})$ or equivalently to find the mean-shift $\boldsymbol{\theta}$, such that the variance σ_{IS}^2 is minimum. Moreover, the variance σ_{IS}^2 would be minimum if the second moment $\nu(\boldsymbol{\theta})$ is

minimum. By minimizing $\nu(\boldsymbol{\theta})$ we obtain a probability estimator with a smaller number of simulations [2, Sect. 3.1] than with the standard MC simulation.

Under some assumptions it has been proved in [7] that the function $\nu(\boldsymbol{\theta})$ has a unique minimizer $\boldsymbol{\theta}^*$ such that $\nabla\nu(\boldsymbol{\theta}^*) = 0$. The optimal $\boldsymbol{\theta}^*$ can be approximated with the Newton algorithm by solving the following optimization problem

$$\boldsymbol{\theta}^* = \min_{\boldsymbol{\theta}} \nu_m(\boldsymbol{\theta}) \tag{2.10}$$

with

$$\nu_m(\boldsymbol{\theta}) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}_{(H(\mathbf{X}_j)\geq\gamma)}\, e^{-\boldsymbol{\theta}\mathbf{X}_j + \frac{|\boldsymbol{\theta}|^2}{2}}, \tag{2.11}$$

the MC approximation of the second moment $\nu(\boldsymbol{\theta})$. For details we refer to [2].

Following [2], there must be at least one $\mathbf{X}_j$ such that $\mathbb{1}_{(H(\mathbf{X}_j)\geq\gamma)} \neq 0$ to solve the optimization problem (2.10). However, this condition may fail in a rare event context. To overcome this problem, a multilevel approach is suggested in [8] for solving such problems in the context of cross-entropy approaches.

2.3 Multi-level approach for rare events simulations

In the multi-level approach, we solve the optimization problem (2.10) iteratively. Starting with the mean $\boldsymbol{\theta}^{(0)} = \mathbf{0}$ of the given density function $g(\mathbf{x})$, we construct a sequence of mean-shifts $\{\boldsymbol{\theta}^{(k)}, k \geq 1\}$ and a sequence of levels $\{\gamma_k, k \geq 1\}$, and iterate in both γ_k and $\boldsymbol{\theta}^{(k)}$ until convergence, see the steps 1 to 6 of the Algorithm 2.1 below. Following [8], each iteration k of the multi-level approach consists of two phases; in the first phase we fix $\boldsymbol{\theta}^{(k-1)}$ and obtain the level γ_k, and in the second phase we compute $\boldsymbol{\theta}^{(k)}$ using $\boldsymbol{\theta}^{(k-1)}$ and γ_k. The computation of γ_k and $\boldsymbol{\theta}^{(k)}$ at iteration k is as follows:

1. *Computation of* γ_k: For fixed $\boldsymbol{\theta}^{(k-1)}$, we let γ_k to be a $(1-\rho)$-quantile of $H(\mathbf{X}^{(k-1)})$, i.e.,

$$\mathbb{P}\big(H\big(\mathbf{X}^{(k-1)}\big) \geq \gamma_k\big) \geq \rho, \tag{2.12}$$

$$\mathbb{P}\big(H\big(\mathbf{X}^{(k-1)}\big) \leq \gamma_k\big) \geq 1-\rho, \tag{2.13}$$

where $\mathbf{X}^{(k-1)} \sim g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})$ and ρ is a probability which is to be chosen such that $\rho \gg \mathrm{p}_{\mathrm{fail}}$, the probability to be estimated.

An estimator $\widehat{\gamma}_k$ of γ_k is obtained by drawing m random samples $\mathbf{X}_i^{(k-1)} \sim g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})$, calculating the responses $H(\mathbf{X}_i^{(k-1)})$ for all i, ordering them from smallest to largest $H_{(1)}^{(k-1)} \leq \cdots \leq H_{(m)}^{(k-1)}$ where $H_{(l)}^{(k-1)} := H(\mathbf{X}_l^{(k-1)})$ and finally evaluating the $(1-\rho)$ sample quantile as

$$\widehat{\gamma}_k = H_{(\lceil(1-\rho)m\rceil)}^{(k-1)}, \tag{2.14}$$

where $\lceil x\rceil$ is the smallest integer greater than or equal to x.

Note that the estimation $\widehat{\gamma}_k$ of γ_k depends on two parameters, the probability ρ and the number of samples m. Our empirical results show that if we fix $m = 1000$ then a good choice of ρ is 0.20 for getting an accurate estimation of γ_k. However, one may choose a smaller ρ but that may require larger m for estimating γ_k accurately. For more details we refer to [8, 9].

2. *Computation of* $\boldsymbol{\theta}^{(k)}$: Let $g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})$ be the density function known at iteration k and $g^{\boldsymbol{\theta}^{(k)}}(\mathbf{x})$ be the new density we want to obtain. The likelihood ratio of densities $g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})$ and $g^{\boldsymbol{\theta}^{(k)}}(\mathbf{x})$ at iteration k is given by

$$\mathcal{L}^{(k)}(\mathbf{x}) = \frac{g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})}{g^{\boldsymbol{\theta}^{(k)}}(\mathbf{x})} = e^{-(\boldsymbol{\theta}^{(k)}-\boldsymbol{\theta}^{(k-1)})\mathbf{x}+\frac{1}{2}(|\boldsymbol{\theta}^{(k)}|^2-|\boldsymbol{\theta}^{(k-1)}|^2)}. \tag{2.15}$$

Therefore, the second moment (2.9) at iteration k can be extended as

$$\begin{aligned} v^{(k)}\big(\boldsymbol{\theta}^{(k)}\big) &= \mathbb{E}_{g^{\boldsymbol{\theta}^{(k)}}}\Big[\big(\mathbb{1}_{(H(\mathbf{X}^{(k)})\geq\gamma_k)}\,\mathcal{L}^{(k)}\big(\mathbf{X}^{(k)}\big)\big)^2\Big] \\ &= \mathbb{E}_{g^{\boldsymbol{\theta}^{(k-1)}}}\Big[\mathbb{1}_{(H(\mathbf{X}^{(k-1)})\geq\gamma_k)}\,\mathcal{L}^{(k)}\big(\mathbf{X}^{(k-1)}\big)\Big], \end{aligned} \tag{2.16}$$

where $\mathbf{X}^{(k-1)} \sim g^{\boldsymbol{\theta}^{(k-1)}}$ and $\mathbf{X}^{(k)} \sim g^{\boldsymbol{\theta}^{(k)}}$.

Using the above information the optimal mean-shift $\boldsymbol{\theta}^{(k)}$ can be approximated with the Newton algorithm by solving the following optimization problem

$$\boldsymbol{\theta}^{(k)} = \min_{\boldsymbol{\theta}} v_m^{(k)}(\boldsymbol{\theta}) \tag{2.17}$$

with

$$v_m^{(k)}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}_{(H(\mathbf{X}_j^{(k-1)})\geq\gamma_k)}\, e^{-(\boldsymbol{\theta}-\boldsymbol{\theta}^{(k-1)})\mathbf{X}_j^{(k-1)}+\frac{1}{2}(|\boldsymbol{\theta}|^2-|\boldsymbol{\theta}^{(k-1)}|^2)} \tag{2.18}$$

the MC approximation of the second moment $v^{(k)}(\boldsymbol{\theta})$.

Below we present the multi-level ISMC algorithm.

Algorithm 2.1 (Multi-level ISMC)

1. Set $k = 1$, $\boldsymbol{\theta}^{(0)} = 0 \in \mathbb{R}^d$ (then $g(\mathbf{x}) \equiv g^{\boldsymbol{\theta}^{(0)}}(\mathbf{x})$), $\rho = 0.2$ and $m = 1000$.
2. Draw m samples $\mathbf{X}_1^{(k-1)}, \ldots, \mathbf{X}_m^{(k-1)}$ according to the distribution $g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})$.
3. Compute the $(1-\rho)$-quantile γ_k of $H(\mathbf{X}^{(k-1)})$. If $\gamma_k > \gamma$ reset γ_k to γ.
4. Introduce

$$v_m^{(k)}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}_{(H(\mathbf{X}_j^{(k-1)})\geq\gamma_k)}\, e^{-(\boldsymbol{\theta}-\boldsymbol{\theta}^{(k-1)})\cdot\mathbf{X}_j^{(k-1)}+\frac{|\boldsymbol{\theta}-\boldsymbol{\theta}^{(k-1)}|^2}{2}}.$$

5. Solve $\boldsymbol{\theta}^{(k)} = \min_{\boldsymbol{\theta}} v_m^{(k)}(\boldsymbol{\theta})$.
6. If $\gamma_k < \gamma$, go to step 2 with $k \leftarrow k+1$, otherwise, go to step 7.
7. Compute (2.8) with $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(k)}$.
8. End of the algorithm.

Figure 1 illustrates the multi-level ISMC approach (Algorithm 2.1) in the situation $\mathbf{x} \in \mathbb{R}^2$. Here, we assume that the optimal mean-shift $\boldsymbol{\theta}^*$ is estimated in four iterations. The sample distribution per mean-shift $\boldsymbol{\theta}^{(k)}$ is shown by the ellipses. Moreover, the blue curves (3-dotted are intermediate and 1-solid is the target) represent the contours at levels γ_k, $k = 1, \ldots, 4$. Notice that $\gamma_4 = \gamma$. We start by sampling from the original pdf $g^{\boldsymbol{\theta}^{(0)}}(\mathbf{x}) = g(\mathbf{x})$, see steps 1 and 2 of the algorithm for iteration $k = 1$. Then we find the level γ_1 (step 3).

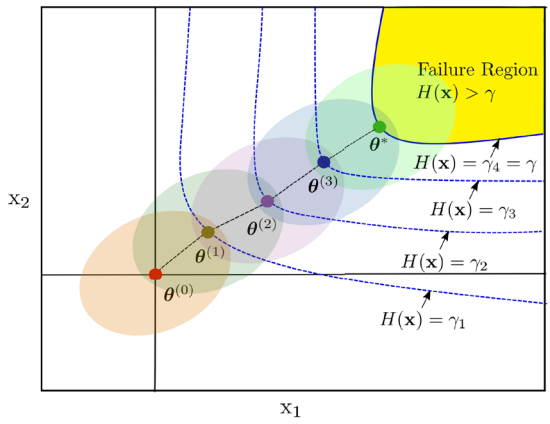

Figure 1 Multi-level (mean-shift) ISMC approach

Afterwards, we estimate $\boldsymbol{\theta}^{(1)}$ using the variance criterion in steps 4 and 5. Now, we go to the second iteration and start by sampling from the new distribution $g^{\boldsymbol{\theta}^{(1)}}(\mathbf{x})$. We repeat the same procedure until convergence to the optimal mean-shift $\boldsymbol{\theta}^*$, see step 6. Here we have $\gamma_4 = \gamma$. Finally, we sample from the optimal pdf $g^{\boldsymbol{\theta}^*}(\mathbf{x})$ (distribution is shown by green ellipse centered at $\boldsymbol{\theta}^*$) and compute $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{IS}}$, see step 7.

3 The surrogate modeling technique

As stated in Sect. 1, we deal with circuits having a large number of statistical input variables. Usually, only a few of them are important i.e., having a statistical effect on the circuit response. The remaining variables have a negligible or no statistical effect on the circuit response. It should be noticed that almost all surrogate models (including Kriging) lose their accuracy with increasing dimension and it is not possible to build an accurate surrogate model with a complete set of variables. Therefore, dimension reduction is required before fitting a surrogate model. Dimension reduction is the process of reducing the number of random variables under consideration and is often divided into two categories: feature extraction and feature selection [10, Chap. 6]. In *feature extraction*, one performs the projection of the original input space to a reduced space. The dimension of the reduced space is much smaller than the original space. There exists a variety of different projection methods. Among others Partial Least Squares Regression (PLSR) [11, 12] belongs to this category. On the other hand, *feature selection* is the process of selecting a subset of important variables while other variables are set to their nominal values and removed from the set of input variables used for building a surrogate model. Our experience with feature extraction, especially with PLSR, shows that it requires a large number (which grows with the dimension of the problem) of training samples to perform accurately, and therefore it cannot be used when only a limited number of simulations are allowed. Hence, we prefer to use a feature selection method that performs accurately even with a limited number of simulations.

3.1 Feature selection

In circuit simulations, *feature selection* is a process of selecting a subset of important variables, having statistical effect on the circuit response of the circuit under study, while other variables having no or negligible effect are set to their nominal values without incurring

much loss of information. Several different approaches for feature selection exist, see for examples [13]. Here we will employ the well known Least Absolute Shrinkage and Selection Operator (LASSO) [14] which is stable and exhibits good properties for feature selection.

3.1.1 LASSO

Let $\mathbf{x} = [x_1, \ldots, x_d]$ be a vector of input variables and $H(\mathbf{x})$ be a circuit response of interest. We can approximate $H(\mathbf{x})$ by a linear model

$$H(\mathbf{x}) \approx \mathbf{x}\mathbf{b} = b_1 x_1 + \cdots + b_d x_d, \tag{3.1}$$

where $\mathbf{b} = [b_1, \ldots, b_d]^T$ is a column vector of regression coefficients. It is clear from (3.1) that each component b_j of $\mathbf{b}$ gives a certain amount of information of the relative importance of x_j. In the context of circuit simulation the dimension of the vector $\mathbf{x}$ of input variables is very large. However, only few of x_j's are important. This means we are willing to determine a small subset of variables that gives the main effects and neglect the small effects. LASSO is a well known method that performs this task. Given a training set $(\mathbf{X}_i, H_i)_{i=1,\ldots,n}$ where $\mathbf{X}_i$ are n random input vectors generated with some design of experiment (see Sect. 3.2.1) and $H_i := H(\mathbf{X}_i)$ are the corresponding responses, and a tuning parameter $\lambda \geq 0$, LASSO estimates the components of $\mathbf{b}$ by minimizing the following Lagrangian expression

$$\widehat{\mathbf{b}} = \min_{\mathbf{b}} \left\{ \sum_{i=1}^{n} (H_i - \mathbf{X}_i \mathbf{b})^2 + \lambda \sum_{j=1}^{d} |b_j| \right\}. \tag{3.2}$$

Depending on the value of the tuning parameter λ, LASSO sets many coefficients exactly to zero. The variables x_j corresponding to the nonzero coefficients b_j are selected as important variables. The computation of the LASSO solutions is a quadratic programming problem and can be tackled by standard numerical analysis algorithms. However, the least angle regression (LARS) [15] procedure is a better approach in terms of the computational efficiency. The LARS algorithm exploits the special structure of the LASSO problem, and provides an efficient way to compute the solutions simultaneously for all possible values of λ. For detail we refer to [16, Sect. 3]. Among all solutions we choose the one that fits the model (3.2) best by cross-validation. Another aspect of LASSO is the choice of the number n of training samples. Typically, n should be much smaller than the dimension d (when d is large). Our experiments show that n depends on the number of important parameters rather than the dimension. The rule of thumb suggests to have 10 samples to each important parameter. Therefore, if we assume there are maximum 50 important variables then at most $50 \times 10 = 500$ simulations are required to performed LASSO accurately.

Once we have selected the important variables we are ready to build a surrogate (Kriging) model. The surrogate model will be built with the important variables only.

3.2 The Kriging model

Remark 3.1 To avoid the use of additional notations, we will introduce the Kriging model for an input vector $\mathbf{x}$ having full dimension d. However, in our algorithms, $\mathbf{x}$ will be used in its reduced form. See Algorithm 4.1.

The notion of a Kriging model, also known as Gaussian process regression in literature [6], was initially developed in a geostatic framework [17]. The Kriging model also plays a central role in the design and analysis of "computer experiments" [5, 18]. The main idea of the Kriging model is to assume that the response function $H(\mathbf{x})$ is a realization of a Gaussian process $\mathcal{G}(\mathbf{x})$

$$\mathcal{G}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + Z(\mathbf{x}), \tag{3.3}$$

where $\mathbf{f}(\mathbf{x})^T\boldsymbol{\beta}$ is a linear regression model on a given functional basis $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_q(\mathbf{x})]^T$ and a vector of regression parameters $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_q]^T$.

For the numerical experiments in Sect. 5, we use the DACE Kriging toolbox [19] for Matlab® to model the response function $H(\mathbf{x})$. DACE provides the functional basis $\mathbf{f}(\mathbf{x})$ as a set of polynomials of order 0, 1 and 2. In this paper, we use the linear functional basis, i.e.,

$$f_1(\mathbf{x}) = 1, \qquad f_2(\mathbf{x}) = \mathrm{x}_1, \qquad \ldots, \qquad f_{d+1}(\mathbf{x}) = \mathrm{x}_d. \tag{3.4}$$

For details we refer to [19].

The second term $Z(\mathbf{x})$ on the right hand side of (3.3) is a Gaussian process with zero mean and covariance

$$\mathbb{E}\big[Z(\mathbf{x})Z(\mathbf{x}')\big] = \sigma^2 \mathcal{R}\big(\mathbf{x}, \mathbf{x}', \boldsymbol{\ell}\big), \quad \forall \big(\mathbf{x}, \mathbf{x}'\big) \in \mathbb{X} \times \mathbb{X}, \tag{3.5}$$

where $\mathbb{X} \in \mathbb{R}^d$ is the domain of input variable $\mathbf{x}$, σ^2 is the process variance of $\mathcal{G}(\mathbf{x})$ and $\mathcal{R}(\mathbf{x}, \mathbf{x}', \boldsymbol{\ell})$ is a correlation function characterized by a vector of parameters $\boldsymbol{\ell} = [\ell_1, \ldots, \ell_d]^T$.

The choice of the correlation function $\mathcal{R}$ depends on the smoothness of the response function $H(\mathbf{x})$ [20]. Assuming $H(\mathbf{x})$ is smooth, we use the following Gaussian correlation function

$$\mathcal{R}\big(\mathbf{x}, \mathbf{x}', \boldsymbol{\ell}\big) = \prod_{j=1}^{d} \exp\left(-\left|\frac{(\mathrm{x}_j - \mathrm{x}'_j)^2}{\ell_j}\right|\right). \tag{3.6}$$

For other correlation functions we refer to [19, 20].

3.2.1 The Kriging predictor

We have a circuit model to be used for simulating the behaviour of a circuit. We find a ordered set $\mathcal{D} = (\mathbf{S}, \mathbf{H})$, called the training set, where $\mathbf{H} = [H(\mathbf{s}_1), \ldots, H(\mathbf{s}_{N_{\mathrm{tr}}})]^T$ is a vector of observations that results from the circuit model on a set of experiments, $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_{N_{\mathrm{tr}}}]^T$. Notice that the notation $\mathbf{s}$ is used here for the input $\mathbf{x}$ to distinguish the training samples from the one that will be used for the other simulations. The set of experiments is usually referred to as a *Design Of Experiments* (DOE), see [5]. The construction of a Kriging predictor depends on $\mathcal{D}$, and the DOE should be selected carefully in order to get the largest amount of the statistical information about the response function over the input space. In this report, we use the *Latin Hypercube Sampling* (LHS) space filling technique for our DOE. We will not discuss the LHS in this paper, but we refer to [21].

Following [19], given a training set $\mathcal{D} = (\mathbf{S}, \mathbf{H})$ the Kriging predictor of an untried point $\mathbf{x}$, i.e., $\mathbf{x} \notin \mathbf{S}$ is given by:

$$\widehat{H}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \widehat{\boldsymbol{\beta}} + r(\mathbf{x})^T R^{-1}(\mathbf{H} - F\widehat{\boldsymbol{\beta}}), \tag{3.7}$$

where

$$\mathbf{H} = \left[H(\mathbf{s}_i)\right]_{i=1,\ldots,N_{\mathrm{tr}}}, \tag{3.8}$$

$$F = \left[f_i(\mathbf{s}_j)\right]_{i=1,\ldots,q, j=1,\ldots,N_{\mathrm{tr}}}, \tag{3.9}$$

$$\widehat{\boldsymbol{\beta}} = \left(F^T R^{-1} F\right)^{-1} F^T R^{-1} \mathbf{H}, \tag{3.10}$$

$$R = \left[\mathcal{R}(\mathbf{s}_i, \mathbf{s}_j, \boldsymbol{\ell})\right]_{i=1,\ldots,N_{\mathrm{tr}}, j=1,\ldots,N_{\mathrm{tr}}}, \tag{3.11}$$

$$r(\mathbf{x}) = \left[\mathcal{R}(\mathbf{x}, \mathbf{s}_i, \boldsymbol{\ell})\right]_{i=1,\ldots,N_{\mathrm{tr}}}. \tag{3.12}$$

The prediction error also known as the Kriging variance at a point $\mathbf{x}$ is given by:

$$\widehat{\sigma}_K^2(\mathbf{x}) = \sigma^2\left(1 + u(\mathbf{x})^T \left(F^T R^{-1} F\right)^{-1} u(\mathbf{x}) - r(\mathbf{x})^T R^{-1} r(\mathbf{x})\right), \tag{3.13}$$

where $u(\mathbf{x}) = F^T R^{-1} r(\mathbf{x}) - \mathbf{f}(\mathbf{x})$.

3.2.2 Estimation of parameters

Given a choice of regression and correlation models, the optimal values of the parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\ell}$ can be inferred using the *Maximum Likelihood Estimation* (MLE). The MLE of $\boldsymbol{\beta}$ is the generalized least-square estimate $\widehat{\boldsymbol{\beta}}$ given in (3.9) and the MLE of σ^2 (see, [19, Sect. 3]) is

$$\widehat{\sigma}^2 = \frac{1}{N_{\mathrm{tr}}}(\mathbf{H} - F\widehat{\boldsymbol{\beta}})^T R^{-1}(\mathbf{H} - F\widehat{\boldsymbol{\beta}}). \tag{3.14}$$

Using these $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma}^2$ the optimal correlation coefficients $\widehat{\boldsymbol{\ell}}$ of the correlation function solve the following optimization problem

$$\widehat{\boldsymbol{\ell}} = \min_{\boldsymbol{\ell}} |R|^{1/N_{\mathrm{tr}}} \widehat{\sigma}^2, \tag{3.15}$$

where $|R|$ is the determinant of R. See [19].

4 A hybrid importance sampling Monte Carlo approach

In this section we propose a HISMC approach. This approach is a modification of the hybrid approach proposed in [1, Sect. 4] and can be used for large circuits. Similar to [1], we split the ISMC Algorithm 2.1 into two phases; The first is the *Exploration Phase* that includes the steps 1 to 6 for estimating the optimal mean-shift $\boldsymbol{\theta}^*$, and the second is the *Estimation Phase* that consists of step 7 for estimating the probability (2.8) using the optimal value $\boldsymbol{\theta}^*$ of the mean-shift $\boldsymbol{\theta}$. In the HISMC approach, we will use a hybrid combination of LASSO, the Kriging model and the circuit model. We will treat the two phases (the exploration phase and the estimation phase) separately.

4.1 The exploration phase

The goal of the exploration phase is to find the optimal mean-shift $\boldsymbol{\theta}^*$. Here we replace the indicator function $\mathbb{1}_{(H(\mathbf{x})\geq\gamma_k)}$ in step 4 of Algorithm 2.1 by an approximation $\mathbb{1}_{(\widehat{H}(\mathbf{x})\geq\gamma_k)}$, where $\widehat{H}(\mathbf{x})$ is the Kriging prediction of the response $H(\mathbf{x})$ at some input $\mathbf{x}$. Notice that the accuracy of the model $\mathbb{1}_{(\widehat{H}(\mathbf{x})\geq\gamma_k)}$ is important and must be checked before its use. The *Misclassification Error* (MCR) is used as a measure of accuracy for the metamodel $\mathbb{1}_{(\widehat{H}(\mathbf{x})\geq\gamma_k)}$.

$$\text{MCR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{(\mathbb{1}_{H(\mathbf{x}_i)\geq\gamma_k} \neq \mathbb{1}_{\widehat{H}(\mathbf{x}_i)\geq\gamma_k})}. \tag{4.1}$$

A leave-one-out cross validation technique for the Kriging model is proposed by [22] and it is used to estimate the MCR in this paper.

Before presenting our algorithm for the exploration phase, we want to mention some preliminaries for extra understanding of the algorithm.

Remark 4.1 (Preliminaries)

(i) A new Kriging model is built at each level of the exploration phase, i.e., in general, the Kriging model at iteration k is different from $k-1$.

(ii) At each iteration k of the exploration phase, the feature selection (using LASSO) is performed before fitting a Kriging model. Note that for iteration k, the feature selection is performed on the reduced set of variables selected at iteration $k-1$.

(iii) The notation $\text{LHS}(\boldsymbol{\theta}^{(k-1)} \pm \underline{a})$ indicates the LHS in the interval $[\boldsymbol{\theta}^{(k-1)} - \underline{a}, \boldsymbol{\theta}^{(k-1)} + \underline{a}]$ where $\boldsymbol{\theta}^{(k-1)} \in \mathbb{R}^d$ is the mean of the known pdf $g^{\boldsymbol{\theta}^{(k-1)}}(\mathbf{x})$ at iteration k and $\underline{a} = [a, a, \ldots, a] \in \mathbb{R}^d$ is a vector with a user defined positive integer a.

(iv) N_{tr}, d and d_r denote the size of a training set, the full and reduced dimensions of the input vector $\mathbf{X}$ respectively.

(v) Define a design matrix $\mathbf{S}^{(k)} = [\mathbf{s}_i^{(k)}]_{N_{\text{tr}}\times d}$ where $\mathbf{s}_i^{(k)} \sim \text{LHS}(\boldsymbol{\theta}^{(k-1)} \pm \underline{a})$ are N_{tr} training samples. In this paper, we use $a = 3$ so that a surrogate model can be fitted to predict the response for the inputs lie in the range of 3-sigma. For a Gaussian distribution 99.7% (almost all) samples lies in this range.

(vi) Given the design matrix $\mathbf{S}^{(k)}$, the corresponding response vector is evaluated from the circuit model and is denoted by $\mathbf{H}^{(k)}$. Note that for iteration k, the columns of $\mathbf{S}^{(k)}$ corresponding to irrelevant variables (outcome of LASSO at iteration $k-1$) are set to zero (nominal values) before evaluating the outputs $\mathbf{H}^{(k)}$.

(vii) Introduce the training set $\mathcal{D}_r^{(k)} = (\mathbf{S}_r^{(k)}, \mathbf{H}^{(k)})$ where $\mathbf{S}_r^{(k)} \subseteq \mathbf{S}^{(k)}$ is the reduced design matrix containing the columns of $\mathbf{S}^{(k)}$ corresponding to the important variables (outcome of the feature selection process).

(viii) In ISMC Algorithm 2.1, we have $\rho = 0.2$ and $m = 1000$ for estimating an intermediate level γ_k at iteration k. Notice (from Fig. 1) that choosing larger value of γ_k will give faster convergence to failure threshold γ. However, for doing that we use a smaller ρ which needs a large n. In a surrogate based approach we use a cheap model instead of the full circuit model and thus a large number m (say, 10,000) of simulations can be used, and therefore a small ρ (say, 0.05) is acceptable.

Algorithm 4.1 (HISMC/Exploration phase)

1. Set $k = 1$, $\boldsymbol{\theta}^{(k-1)} = \boldsymbol{\theta}^{(0)} = 0 \in \mathbb{R}^d$, $\rho = 0.05$, $m = 10{,}000$, $d_r^{(k-1)} = d$ and $a = 3$.

2. Choose $N_{\text{tr}} = 200$ if $d_r^{(k-1)} < 20$, $N_{\text{tr}} = 10d_r^{(k-1)}$ if $20 \leq d_r^{(k-1)} \leq 50$ otherwise $N_{\text{tr}} = 500$.
3. Find the training set $\mathcal{D}^{(k)} = (\mathbf{S}^{(k)}, \mathbf{H}^{(k)})$ at iteration k.
4. Perform feature selection using LASSO on the training set $\mathcal{D}^{(k)}$.
5. Find $d_r^{(k)}$ and $\mathcal{D}_r^k = (\mathbf{S}_r^k, \mathbf{H}^k)$ the number of important variables and the reduced training set, respectively.
6. Fit a Kriging model using the training set $\mathcal{D}_r^k$.
7. Draw m iid random samples $\mathbf{X}_1^{(k)}, \ldots, \mathbf{X}_m^{(k)} \sim f(\mathbf{x}, \boldsymbol{\theta}^{(k-1)})$ and estimate $\widehat{H}(\mathbf{X}_1^{(k)}), \ldots, \widehat{H}(\mathbf{X}_m^{(k)})$ using the Kriging model.
8. Compute the $(1-\rho)$ sample quantile γ_k. If $\gamma_k > \gamma$ reset γ_k to γ.
9. Introduce the Kriging based variance criterion

$$v_m^{(k)}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{j=1}^{m}(\mathbb{1}_{\{\widehat{H}(\mathbf{X}_j^{(k)})\geq\gamma_k\}})^2\, e^{(-\mathbf{X}_j^{(k)}\cdot(\boldsymbol{\theta}-\boldsymbol{\theta}^{(k-1)})+\frac{1}{2}(|\boldsymbol{\theta}|-|\boldsymbol{\theta}^{(k-1)}|^2))}$$

   and compute $\theta^{(k)} = \arg\min_\theta v_m^{(k)}(\theta)$.
10. If $\gamma_k < \gamma$, return to step 2 and proceed with $k \leftarrow k+1$, otherwise save $\theta^* = \theta^{(k)}$, $d_r = d_r^{(k)}$ and the reduced set (say $\mathbf{x}_r$) of input variables.
11. Go to the estimation phase.

4.2 The estimation phase

Assuming that the optimal value $\boldsymbol{\theta}^*$ of the mean-shift $\boldsymbol{\theta}$ is computed in the exploration phase, our next goal is to find an estimation of the probability p_{fail}. To this end, we build a surrogate based accurate probability estimator. One simple approach [1] is to replace the indicator function $\mathbb{1}_{(H(\mathbf{Y})\geq\gamma)}$ in (2.8) by its surrogate model $\mathbb{1}_{(\widehat{H}(\mathbf{Y})\geq\gamma)}$, where $\widehat{H}(\mathbf{Y})$ is the Kriging prediction of the response $H(\mathbf{Y})$ at some input $\mathbf{Y} = \mathbf{X} + \boldsymbol{\theta}^*$. Here the Kriging predictor is built on the training set with input vectors centered at $\boldsymbol{\theta}^*$. To get an impression of accuracy of the probability estimator we would like to have a confidence interval. A candidate is Pseudo Confidence Interval (PCI), that depends on the Kriging variance, is provided in [1]. However, there is no proof that the true probability would lie in the PCI. Moreover, the PCI can be very wide if the Kriging prediction has a large variance. To prevent loss of accuracy of the probability estimator, a hybrid approach is proposed by [23, 24] that combines the simulations of the Kriging model and the original system (circuit model in our context). The original system is used only for the responses that are close to the failure threshold. In this section we will use a similar approach based on the Kriging model. Unlike the hybrid approach in [23] where the first surrogate model is used to simulate, we check the accuracy of the model online and improve (re-build) it, if required. The authors in [25] demonstrate the benefits of using an improved surrogate model which might be useful to our application. In this paper, we use an adaptive sampling technique to improve the Kriging model. We add some samples (adaptively) to the initial training set from the region of interest, see step 14 in Algorithm 4.2. Due to the interpolation nature, the Kriging model gives a better fit in that region after the improvement.

We start with drawing a training set $\mathcal{D}^* = (\mathbf{S}^*, \mathbf{H}^*)$ where $\mathbf{S}^*$ is the $N_{\text{tr}} \times d$ design matrix with rows representing the N_{tr} random vectors generated with the LHS($\boldsymbol{\theta}^* \pm \underline{a}$) and $\mathbf{H}^*$ is an $N_{\text{tr}} \times 1$ vector of corresponding responses. Then we perform the feature selection (using LASSO) and find the reduced training set $\mathcal{D}_r^* = (\mathbf{S}_r^*, \mathbf{H}^*)$ where $\mathbf{S}_r^* \subseteq \mathbf{S}^*$ is the

reduced design matrix containing the columns of $\mathbf{S}^*$ corresponding to the important variables. Afterward, we build a Kriging model on the updated training set $\mathcal{D}_r^*$. Finally, we build a hybrid indicator function $\mathcal{I}_\gamma(\mathbf{Y})$ called an emulator that combines the true indicator function $\mathbb{1}_{(H(\mathbf{Y})\geq\gamma)}$ and its surrogate $\mathbb{1}_{(\widehat{H}(\mathbf{Y})\geq\gamma)}$ based on an accept/reject criterion. See next section.

4.2.1 The emulator

We define an interval known as the "margin of uncertainty" [20] around the threshold γ.

$$\mathbb{M} = \left\{\mathbf{Y} : \left|\widehat{H}(\mathbf{Y}) - \gamma\right| \leq z_{\alpha/2}\widehat{\sigma}_K(\mathbf{Y})\right\}, \tag{4.2}$$

where $\widehat{\sigma}_K(\mathbf{Y})$ is the Kriging variance at point $\mathbf{Y}$, $z_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$ with $\Phi^{-1}(x)$ is the inverse cumulative distribution function of the standard normal distribution. If $\alpha = 0.01$ for which $z_{\alpha/2} = 2.58$, we assume that there is 99% chance that a true value lies in the interval $\gamma - z_{\alpha/2}\widehat{\sigma}_K(\mathbf{Y}) \leq \widehat{H}(\mathbf{Y}) \leq \gamma + z_{\alpha/2}\widehat{\sigma}_K(\mathbf{Y})$.

The accept/reject regions are indicated in Fig. 2 which says that we accept the simulation of the Kriging predictor $\widehat{H}(\mathbf{Y})$ if $\widehat{H}(\mathbf{Y})$ is reasonably away ($\mathbf{Y} \notin \mathbb{M}$) from the failure threshold γ and we reject $\widehat{H}(\mathbf{Y})$ if it is close ($\mathbf{Y} \in \mathbb{M}$) to γ. In the latter case the circuit model must be used.

The mathematical formulation of the Emulator is given as follows

$$\mathcal{I}_\gamma(\mathbf{Y}) = \begin{cases} \mathbb{1}_{(H(\mathbf{Y})\geq\gamma)} & \text{if } \mathbf{Y} \in \mathbb{M}, \\ \mathbb{1}_{(\widehat{H}(\mathbf{Y})\geq\gamma)}, & \text{otherwise.} \end{cases} \tag{4.3}$$

4.2.2 Probability estimator

The emulator-based probability estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}$ of the failure probability $\mathrm{p}_{\mathrm{fail}}$ is given by

$$\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{I}_\gamma(\mathbf{Y}_i)\, e^{-\boldsymbol{\theta}^*\mathbf{x}_i - \frac{1}{2}|\boldsymbol{\theta}^*|^2} \tag{4.4}$$

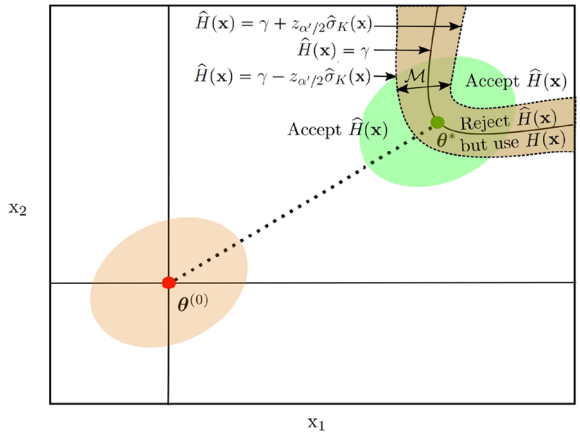

Figure 2 The visualization of the accept/reject criterion

and its variance can be estimated by

$$\begin{aligned}\hat{\sigma}_E^2 &= \frac{1}{N}\operatorname{Var}\left(\mathcal{I}_\gamma(\mathbf{Y})\, e^{-\boldsymbol{\theta}^*\mathbf{X}-\frac{1}{2}|\boldsymbol{\theta}^*|^2}\right)\\ &= \frac{1}{N}\mathbb{E}\left[\left(\mathcal{I}_\gamma(\mathbf{Y})\, e^{-\boldsymbol{\theta}^*\mathbf{X}-\frac{1}{2}|\boldsymbol{\theta}^*|^2} - \hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}\right)^2\right]\\ &\approx \frac{1}{N}\left[\frac{1}{N-1}\left(\sum_{i=1}^{N}\left(\mathcal{I}_\gamma(\mathbf{Y}_i)\, e^{-\boldsymbol{\theta}^*\mathbf{X}_i-\frac{1}{2}|\boldsymbol{\theta}^*|^2}\right)^2 - N\left(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}\right)^2\right)\right].\end{aligned} \tag{4.5}$$

Using (4.4) and (4.5), the $100(1-\alpha')\%$ confidence interval for the true probability $\mathrm{p}_{\mathrm{fail}}$ is defined as

$$\mathrm{CI} = \left[\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}} - z_{\alpha'/2}\hat{\sigma}_E,\ \hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}} + z_{\alpha'/2}\hat{\sigma}_E\right] \tag{4.6}$$

and the accuracy of the probability estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}$ can be measured by the coefficient of variation

$$\mathrm{CV} = z_{\alpha'/2}\,\frac{\hat{\sigma}_E}{\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}}. \tag{4.7}$$

The 95% confidence interval is the most commonly used interval that corresponds to $z_{\alpha'/2} = 1.96$ for $\alpha' = 0.05$. The accepted probability $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}$ is the one with a coefficient of variation $\mathrm{CV} \leq 10\%$.

Combining the information from the exploration and estimation phases, the full HISMC algorithm can be formulated as follows:

Algorithm 4.2 (HISMC/Estimation phase)

1. Use Algorithm 4.1 for finding the optimal mean-shift $\boldsymbol{\theta}^*$, the number d_r of important variables and the reduced set of (important) input variables $\mathbf{x}_r$.
2. *Initialize the Estimation phase*:
   (a) Set $a = 3$ and $N_{\mathrm{tr}} = 200, 10d_r, 500$ if $d_r < 20$, $20 \leq d_r \leq 50$, otherwise.
   (b) Set the iteration parameter $l = 0$, maximum number of iterations $l_{\max} = 10$, the number $n = 1000$ of simulations used at iteration l, initialize the total number $N = 0$ of simulations before iteration l, $z_{\alpha/2} = 2.58$, $z_{\alpha'/2} = 1.96$, tolerance of CV tol_cv $= 0.1$, $\tau = 0$, $\eta = 0$.
3. Get the training set $\mathcal{D}^* = (\mathbf{S}^*, \mathbf{H}^*)$ where $\mathbf{S}^* = [\mathbf{s}_i^*]_{N_{\mathrm{tr}}\times d}$ with $\mathbf{s}_i^* \sim \mathrm{LHS}(\boldsymbol{\theta}^* \pm \underline{a})$ and $\mathbf{H}^*$ is a vector of corresponding responses.[c]
4. Perform feature selection using LASSO on the training set $\mathcal{D}^*$.
5. Update the number d_r of important parameters and find the reduced training set $\mathcal{D}_r^* = (\mathbf{S}_r^*, \mathbf{H}^*)$. See preliminaries for Algorithm 4.1.
6. Fit a Kriging model.
7. Draw n iid random samples $\mathbf{X}_i^{(l)} \sim g(\mathbf{x})$ and shift them to $\mathbf{Y}_i^{(l)} = \mathbf{X}_i^{(l)} + \boldsymbol{\theta}^*$.
8. Find the Kriging predictions $\widehat{H}(\mathbf{Y}_i^{(l)})$ and Kriging variances $\widehat{\sigma}_K^2(\mathbf{Y}_i^{(l)})$ using (3.7) and (3.13), respectively.
9. Compute the likelihood weights $w_i^{(l)} = e^{-\boldsymbol{\theta}^*\mathbf{X}_i^{(l)}-\frac{1}{2}|\boldsymbol{\theta}^*|^2}$ and $v_i^{(l)} = (w_i^{(l)})^2$ for all $i = 1, \ldots, n$.

10. Determine $\tau \leftarrow \tau + \sum_{i=1}^{n} w_i^{(l)} \mathcal{I}_\gamma(\mathbf{Y}_i^{(l)})$, $\eta \leftarrow \eta + \sum_{i=1}^{n} v_i^{(l)} \mathcal{I}_\gamma(\mathbf{Y}_i^{(l)})$ and $N \leftarrow N + n$.
11. Compute $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}} = \frac{\tau}{N}$ and $\hat{\sigma}_E^2 = \frac{1}{N(N-1)}(\eta - N(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}})^2)$.
12. Calculate $\mathrm{CV} = z_{\alpha'/2} \frac{\hat{\sigma}_E}{\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}}$.
13. If CV > tol_cv and $l < l_{\max}$ go to step 14 otherwise go to step 15.
14. Use the full simulations N_{full} drawn to compute $\mathcal{I}_\gamma(\mathbf{Y}_i^{(l)})$, see definition (4.3). Select some points[d] (say, $\min\{10, N_{\mathrm{full}}\}$) uniformly among all N_{full} simulations. Add these samples into the training set and rebuild the Kriging model with the updated training set. Go to step 6 with $l \leftarrow l + 1$.
15. Determine the 95% confidence interval

$$\mathrm{CI} = \left[\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}} - z_{\alpha'/2}\hat{\sigma}_E,\ \hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}} + z_{\alpha'/2}\hat{\sigma}_E\right].$$

16. Save the probability $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{E}}$, the variance $\hat{\sigma}_E^2$, CI and the CV.
17. End of the algorithm.

5 Results and discussion

In this paper two realistic circuits are used for validation purpose. The first one is a VCO with 1500 statistical input parameters and scalar response 'oscillation frequency' and the second one is a memory cell with 2096 statistical input parameters and scalar response 'read delay'. The ISMC and HISMC algorithms are repeated 100 times, and the empirical results for both the exploration and estimation phase are compared. In the exploration phase we compare the optimal mean-shift computed by both the algorithms and the required number of simulations. On the other hand, we compute the probability in the estimation phase, and therefore we need to measure the efficiency of the probability. When we replace the circuit model with a surrogate model, the simulation time becomes negligible, and one can take only true simulation runs into account for estimating the efficiency of the algorithm. However, the surrogate model introduces a bias that may be very large. Therefore, we use the following procedure to measure the efficiency of the emulator based probability estimator.

1. *Get a reference probability*: Let $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$ be a probability estimator of a method M for estimating the probability $\mathrm{p}_{\mathrm{fail}}$. To estimate empirically the bias in $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$ we need a reference probability $\mathrm{p}_{\mathrm{fail}}^{\mathrm{ref}}$. A simple estimation $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$ of $\mathrm{p}_{\mathrm{fail}}^{\mathrm{ref}}$ can be obtained by running ISMC Algorithm 2.1 with a small coefficient of variation (say less than 1%).
2. *Perform N_{rep} experiments of method* M: Notice that a probability generated from the MC estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$ is a random number and thus comparing a single outcome of $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$ with the reference probability $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$ is not valid. Hence, we perform N_{rep} independent experiments of the method M and store N_{rep} outcomes $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}(i)$ and their confidence intervals $\mathrm{CI}^{\mathrm{M}}(i)$ for $i = 1, \ldots, N_{\mathrm{rep}}$.
3. Then we compute
   (a) *Relative bias*: The relative bias ϵ_{rel} of the estimator $\hat{\mathrm{p}}_{\mathrm{fail}}$ with respect to the reference probability $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$

$$\epsilon_{\mathrm{rel}}\left(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\right) = \frac{\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}} - \mathrm{mean}(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}})}{|\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}|} \times 100\%, \tag{5.1}$$

where

$$\operatorname{mean}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr) = \frac{1}{N_{\mathrm{rep}}} \sum_{i=1}^{N_{\mathrm{rep}}} \hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}(i). \tag{5.2}$$

Note that we do not use absolute value for the numerator in (5.1), since it gives an indication whether or not $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$ underestimates or overestimates the reference value.

(b) *Central Coverage Probability* (*CCP*): The CCP for the estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$, which is the probability that $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$ lies within CI^{M}, is given by

$$\mathrm{CCP}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr) = \frac{1}{N_{\mathrm{rep}}} \sum_{i=1}^{N_{\mathrm{rep}}} \mathbb{1}_{\{\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}} \in \mathrm{CI}^{\mathrm{M}}(i)\}}. \tag{5.3}$$

For a 95% confidence interval CI^{M}, an unbiased estimator $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}$ and a large N_{rep} the CCP must be 0.95 (approximately). However, for a biased estimator CCP might be smaller than 0.95. We assume that a (biased) estimator is good enough if it does not have CCP lower than 0.90, i.e., a 5% error in the confidence interval is acceptable.

(c) *Mean Squared Error (MSE)*: The MSE of $\hat{\mathrm{p}}_{\mathrm{fail}}$ is computed as

$$\mathrm{MSE}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr) = \frac{1}{N_{\mathrm{rep}}} \sum_{i=1}^{N_{\mathrm{rep}}} \bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}(i) - \hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}\bigr)^2. \tag{5.4}$$

Note that $\mathrm{MSE}(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}})$ can be written as

$$\begin{aligned} \mathrm{MSE}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr) &= \frac{1}{N_{\mathrm{rep}}} \sum_{i=1}^{N_{\mathrm{rep}}} \bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}(i) - \operatorname{mean}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr)\bigr)^2 + \bigl(\operatorname{mean}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr) - \hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}\bigr)^2 \\ &= \operatorname{Var}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr) + \bigl(\operatorname{bias}\bigl(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M}}\bigr)\bigr)^2. \end{aligned} \tag{5.5}$$

Then we can say that the MSE is the sum of variance and squared bias of the estimator which provide a useful way to estimate the efficiency of a biased estimator (see 4 below).

4. *Estimate the Efficiency Metric*: We now introduce an efficiency estimator denoted by $\widehat{\mathrm{Eff}}$ and given as

$$\widehat{\mathrm{Eff}}(\mathrm{M1}, \mathrm{M2}) = \frac{\mathrm{MSE}(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M1}})}{\mathrm{MSE}(\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{M2}})} \frac{\overline{T}_{\mathrm{M1}}}{\overline{T}_{\mathrm{M2}}}, \tag{5.6}$$

where $\overline{T}_{\mathrm{M1}}$ and $\overline{T}_{\mathrm{M2}}$ are the average computational costs (CPU time) required for method M1 and M2, respectively.

If $\widehat{\mathrm{Eff}}(\mathrm{M1}, \mathrm{M2}) = \kappa > 1$, it means that method M1 requires κ times more computational cost than M2 to obtain the same accuracy. If $\widehat{\mathrm{Eff}}(\mathrm{M1}, \mathrm{M2}) > 1$ then estimator M2 is preferred to M1.

5.1 The VCO

5.1.1 Results of the exploration phase

Figure 3 indicates the computation of the mean-shift $\boldsymbol{\theta}^{(k)}$ and the target level γ_k at each iteration of the exploration phase. Note that we plot the empirical means of $\|\boldsymbol{\theta}^{(k)}\|$ and γ_k from the ISMC and HISMC algorithms repeated 100 times. In both figures, the blue dotted curves with asterisks stand for the ISMC approach and the red dotted curves with squares are for the HISMC approach. We start with $\boldsymbol{\theta}^{(0)} = 0$ and compute the pair $(\gamma_k, \boldsymbol{\theta}^{(k)})$, iteratively. It can be seen from the figure (at right) that the ISMC algorithm takes 7 iterations to reach the target $\gamma = 1900$. However, the HISMC algorithm takes only 4 iterations. Indeed, in the case of ISMC we used $\rho = 0.2$ and $m = 1000$ for estimating the intermediate levels γ_k. On the other hand, in HISMC we replace the expensive circuit model by the Kriging model that allows us to use more samples ($m = 10{,}000$) and then a smaller value of ρ (0.05) is acceptable. Clearly, HISMC makes bigger steps that results into less iterations. In the left plot the estimated norm of the mean-shifts corresponding to γ_k is shown at level k. From Tables 1 and 2, the last $\|\boldsymbol{\theta}^{(k)}\|$ of both methods are 6.42 and 6.44. The HISMC has only 0.3% relative error with respect to ISMC and thus we can see in Fig. 3 that the last $\|\boldsymbol{\theta}^{(k)}\|$ of both methods lies on the same (black horizontal) line, approximately.

Tables 1 and 2 represent the numerical results for the exploration phase of the ISMC and the HISMC algorithm, respectively. The first, second, third and fourth columns represent the iteration number k, the number of full simulations, the intermediate level γ_k and the mean-shift $\boldsymbol{\theta}^{(k)}$ per iteration, respectively. The fifth and sixth column in Table 2 represent the reduced dimension d_r and the LOO-MCR (leave-one-out misclassification

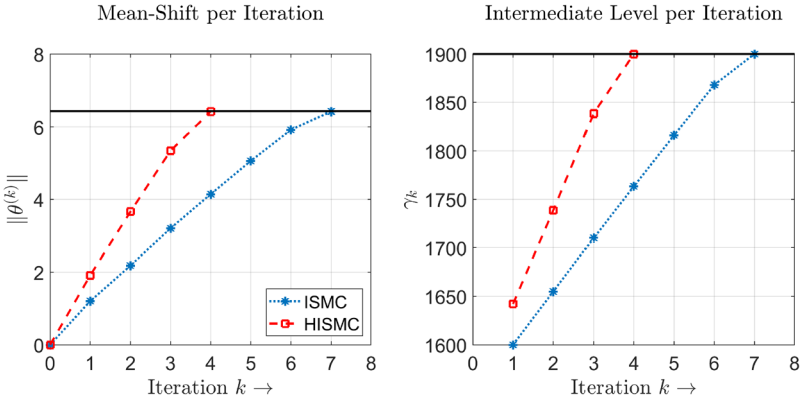

Figure 3 VCO: The mean-shifts $\boldsymbol{\theta}^{(k)}$ and intermediate levels γ_k at iteration k

Table 1 Memory cell: ISMC exploration phase

Iteration (k)	#Runs	γ_k	$\|\boldsymbol{\theta}_k\|$
1	1000	1600	1.20
2	1000	1655	2.18
3	1000	1710	3.20
4	1000	1764	4.14
5	1000	1816	5.06
6	1000	1868	5.92
7	1000	1900	6.42
Total	7000		

Table 2 VCO: HISMC exploration phase

Iteration (k)	#Runs	γ_k	$\|\boldsymbol{\theta}_k\|$	d_r	LOO-MCR (%)
1	500	1645	1.87	27	0.6
2	270	1738	3.64	25	0.3
3	250	1835	5.38	24	1.0
4	240	1900	6.44	24	1.0
Total	1260				

Table 3 VCO: reference probability estimation

Method	Probability ($\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$)	CV (%)	#Runs
Reference	1.10×10^{-10}	0.99	345,000

Table 4 VCO: ISMC versus HISMC probability estimation

Method	Mean probability	CV (%)	ϵ_{rel} (%)	CCP	MSE	#Runs
ISMC	1.10×10^{-10}	8.14	0	0.94	2.34×10^{-23}	5000
HISMC	1.07×10^{-10}	9.21	2.92	0.92	3.27×10^{-23}	457

error) at iteration k, respectively. Recalling the training sample rule (see Algorithm 4.1), we draw 500 training samples at iteration $k = 1$ (see the second column) since $d = 1500$ which is greater than 50. The training sample size at iteration $k + 1$ depends on the reduced dimension d_r (fifth column) at iteration k. Further, the LOO-MCR (4.1) stands for the "leave one out misclassification error" of the Kriging model at iteration k. We can see that the maximum error is 1% (for $k = 3$ and 4).

Moreover, it can be seen that the ISMC requires total 7000 simulations to estimate the optimal mean-shift. On the other hand, HISMC requires 1260 full simulations only.

5.1.2 Results of the estimation phase

Given the optimal mean-shift from the exploration phase, we estimate the failure probability $\mathrm{p}_{\mathrm{fail}}$. First we get a reference probability $\mathrm{p}_{\mathrm{fail}}^{\mathrm{ref}}$ by running the ISMC algorithm with a small (less than 1%) coefficient of variation. The reference results are shown in Table 3. The probability $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$ is being useful to measure the bias, MSE and thus the efficiency of the HISMC method.

Now we present the empirical results of the estimation phase of the ISMC and HISMC algorithms. Recall that these results are based on the 100 experiments, i.e., both the algorithms are repeated 100 times with different 'seed' of the random generator each time. It is worth mentioning here that we performed a feature selection before fitting the Kriging model in the estimation phase of the HISMC algorithm and we also rebuild the Kriging model many times during the process. The average number of important parameters is $d_r = 23$. Before converging the probability up to the required accuracy, we rebuild the Kriging model 4 times by adding some samples from the region near to the failure threshold. Table 4 represents the results of the estimation phase of the ISMC and HISMC algorithms. The mean probability, the average coefficient of variation (CV(%)) and the mean squared error (MSE) are computed for both of the techniques. We also computed the relative bias ($\epsilon_{\mathrm{rel}}(\%)$) and the central coverage probability (CCP) for the HISMC technique. The last column indicates the total number (#Runs) of true simulations used in the estimation phase.

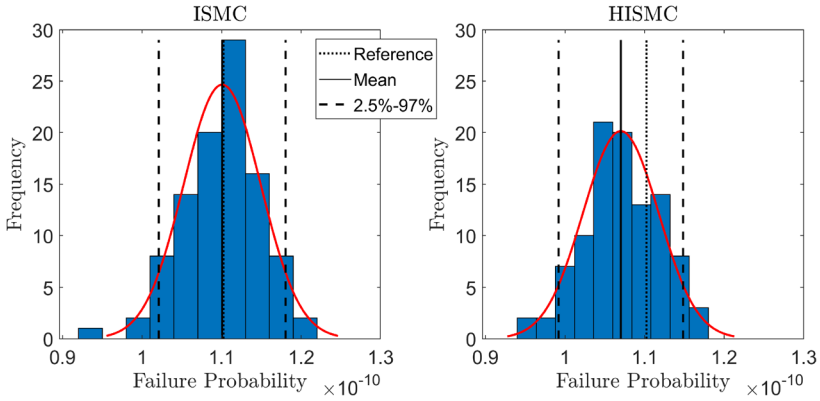

Figure 4 VCO: Empirical probability distributions of ISMC and HISMC from $N_{\text{rep}} = 100$ experiments with mean 1.10×10^{-10} and 1.07×10^{-10}, respectively

The results in Table 4 are visualized by Fig. 4. The left and right-hand side plots show the probability distributions from $N_{\text{rep}} = 100$ experiments (repetitions) of ISMC and HISMC methods, respectively. The vertical solid 'black' line in the center represents the mean of the N_{rep} random values of the failure probability. The dotted line in the center is the reference probability $\hat{p}_{\text{fail}}^{\text{ref}}$. The two dashed lines around the center represent the 95% confidence interval. Clearly, the reference probability $\hat{p}_{\text{fail}}^{\text{ref}}$ lies within the 95% empirical CI for both the methods. Moreover, the difference between mean and the reference probability gives the bias in the probability estimator. For the HISMC estimator (right-hand side plot), a bias ($\epsilon_{\text{rel}} = 2.92\%$) can be noticed. Because of this bias, the CCP is equal to 0.92 (see Table 4) which is smaller than 0.95 (the desired probability for a 95% CI). We assume that for $N_{\text{rep}} = 100$ a CCP with less than and equal to 5% error is acceptable. For more accurate estimation of the bias ϵ_{rel} and CCP we need to perform a larger number N_{rep} (say 1000) of experiments. Finally, we estimate the efficiency of the HISMC with respect to the ISMC method. Recalling formula (5.6), we require the mean squared error and the average CPU time for both methods. The mean squared error is given in Table 4. The average CPU times required for ISMC and HISMC are $\bar{T}_{\text{ISMC}} = 82{,}239$ s and $\bar{T}_{\text{HISMC}} = 9676$ s, respectively. Thus, we get

$$\widehat{\text{Eff}}(\text{ISMC}, \text{HISMC}) = \frac{\text{MSE}(\hat{p}_{\text{fail}}^{\text{IS}}) \times \bar{T}_{\text{ISMC}}}{\text{MSE}(\hat{p}_{\text{fail}}^{\text{E}}) \times \bar{T}_{\text{HISMC}}} = \frac{2.34 \times 10^{-23} \times 82{,}239}{3.28 \times 10^{-23} \times 9676} \approx 6.$$

This means that ISMC requires approximately 6 times more CPU time than HISMC to achieve the same accuracy. Hence, we get a speedup of factor 6.

5.2 The memory cell

5.2.1 Results of the exploration phase

Similar to the VCO, both the ISMC and HISMC algorithms were repeated $N_{\text{rep}} = 100$ times with a different seed of the random generator each time. The mean results (average of 100 experiments) of the exploration phase are given in this section.

Figure 5 indicates the computation of the mean-shift $\boldsymbol{\theta}^{(k)}$ and the intermediate failure thresholds γ_k at each iteration of the exploration phase for the memory cell. Here the target threshold failure $\gamma = 902$ is reached in 6 and 4 iterations (right plot) per method.

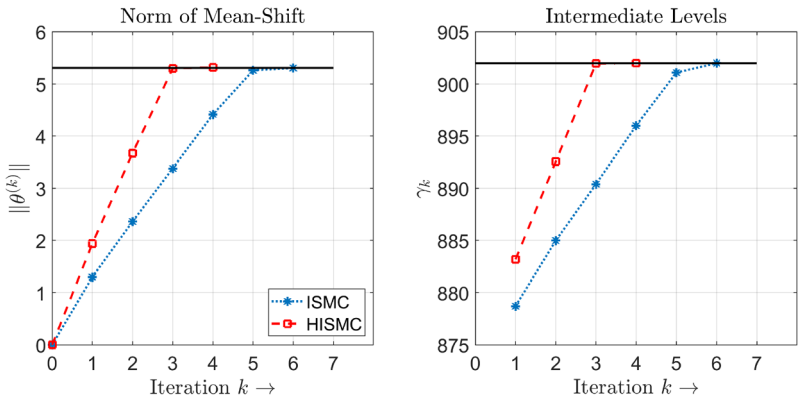

Figure 5 Memory cell: the mean-shifts $\boldsymbol{\theta}^{(k)}$ and intermediate levels γ_k at iteration k

Table 5 Memory cell: ISMC exploration phase

Iteration (k)	#Runs	γ_k	$\|\boldsymbol{\theta}_k\|$
1	1000	878.72	1.30
2	1000	884.99	2.36
3	1000	890.40	3.38
4	1000	896.01	4.14
5	1000	901.10	5.26
6	1000	902.00	5.31
Total	6000		

Table 6 Memory cell: HISMC exploration phase

Iteration (k)	#Runs	γ_k	$\|\boldsymbol{\theta}_k\|$	d_r	LOO-MCR (%)
1	500	883.38	1.91	35	2.0
2	350	892.40	3.64	34	1.0
3	340	901.75	5.30	33	1.0
4	330	902.00	5.32	33	1.0
Total	1520				

The last $\|\boldsymbol{\theta}^{(k)}\|$ which are known as optimal-mean shift have the values 5.31 and 5.32 for the ISMC and HISMC methods, respectively. Thus, $\|\boldsymbol{\theta}^{(k)}\|$ for both methods converge to the same line. To this end, we can say that HISMC gives a good estimation of the mean-shift computed with ISMC.

Tables 5 and 6 show the numerical results from the exploration phase of the ISMC and HISMC, respectively. It can be seen from the tables that HISMC requires 1520 full simulations (used for training the Kriging models) for estimating the optimal mean-shift. On the other hand, ISMC requires 6000 simulations.

5.2.2 Results of the estimation phase

The reference probability $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$ is shown in Table 7. Again, the $\hat{\mathrm{p}}_{\mathrm{fail}}^{\mathrm{ref}}$ is computed using ISMC algorithm with a small coefficient of variation.

The empirical results of the estimation phase with $N_{\mathrm{rep}} = 100$ experiments of ISMC and HISMC methods are shown in Table 8. It is worth noting here that, in HISMC, we perform the dimension reduction (using LASSO) before fitting the initial Kriging model in the estimation phase. The average number of important parameters is $d_r = 33$. Moreover, we

Table 7 Memory cell: reference probability estimation

Method	Probability ($\hat{p}_{fail}^{ref}$)	Variance ($\widehat{\sigma}_{ref}^2$)	CV (%)	#Runs
ISMC	6.01×10^{-8}	9.26×10^{-20}	0.99	315,000

Table 8 Memory cell: ISMC versus HISMC probability estimation

Method	Mean probability	CV (%)	ϵ_{rel} (%)	CCP	MSE	#Runs
ISMC	6.02×10^{-8}	7.85	−0.17	0.95	3.65×10^{-18}	5000
HISMC	5.98×10^{-8}	6.83	0.51	0.96	3.70×10^{-18}	1130

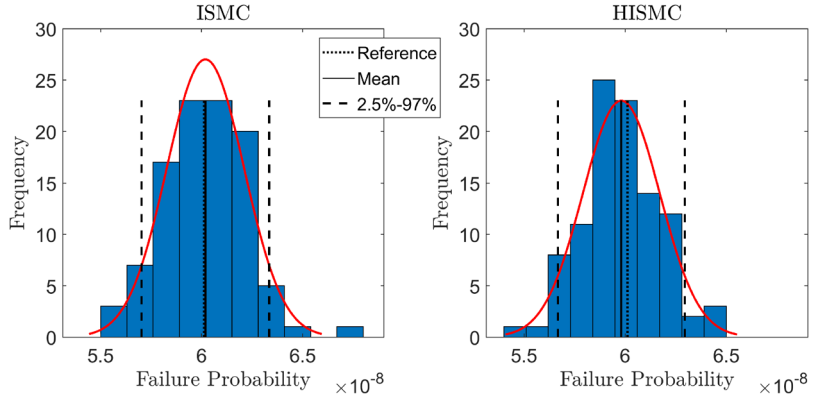

Figure 6 Memory cell: probability distributions of ISMC and HISMC methods from $N_{rep} = 100$ experiments with mean 6.02×10^{-8} and 5.98×10^{-8}, respectively

rebuild the Kriging model 4 times (during the process) by adding some samples from the region near to the failure threshold. Furthermore, we see in the table that the relative bias (ϵ_{rel}) for HISMC is small, and therefore we have a good estimation of the CCP.

Similar to VCO, the probability distribution plots from $N_{rep} = 100$ experiments (repetitions) of ISMC and HISMC methods are shown in Fig. 6. Clearly, the reference probability $\hat{p}_{fail}^{ref}$ lies within the 95% empirical CI for both the methods. Moreover, the difference between mean and the reference probability gives the bias in the probability estimator. Compared to VCO we here see a smaller bias in the HISMC estimator. The reason is the small number $N_{rep} = 100$ of experiments. For accurate estimation of the bias ϵ_{rel} and CCP we need to perform a higher number N_{rep} (say 1000) of experiments.

Finally, we estimate the efficiency of the HISMC with respect to the ISMC method. The mean squared error is given in Table 8. The average CPU times required for ISMC and HISMC are $\bar{T}_{ISMC} = 53{,}448$ s and $\bar{T}_{HISMC} = 7922$ s, respectively. Thus, we get

$$\widehat{\text{Eff}}(\text{ISMC}, \text{HISMC}) = \frac{3.65 \times 10^{-18} \times 53{,}448}{3.70 \times 10^{-18} \times 7922} = 6.7.$$

This means that ISMC requires approximately 6.7 times more CPU time than HISMC to achieve the same accuracy. Hence, we get a speedup of factor 6.7, and therefore we prefer the HISMC over the ISMC.

6 Conclusion and future work

In this paper we proposed a HISMC approach for yield optimization of circuits having a very large number of input variables and scalar response. Moreover, we assume that only a few (say less than 35) of the input variables are important. The HISMC approach uses a feature selection method (LASSO) that reduces the dimension of the input variables of an underlying problem that allows us to fit the Kriging model on the reduced dimension. The Kriging model is used for most of the simulations and makes a significant reduction on runs from the expensive to use circuit model. Although it is hard or even impossible to quantify the bias in the probability estimator, the Emulator prevents loss of accuracy by using the true simulations near to the failure threshold. For future work we will try to compare the HISMC approach (in terms of efficiency and robustness) with a hybrid approach proposed by [24] and with commercially available methods (e.g., Solido® and MunEDA®). More focus will be on multi-input and multi-output circuits, especially in considering output $H(\mathbf{x})$ with more constraints involved.

Acknowledgements
The authors would like to thank Cyril Desclèves, Joost Rommes, Pascal Bolcato from Mentor Graphics, Grenoble for valuable discussions to improve some aspects of the work. The first author is grateful to the financial support from the Marie Curie Action.

Funding
This research is completely supported by the European Union in the FP7-PEOPLE-2013-ITN Program under Grant Agreement Number 608243 (FP7 Marie Curie Action, Project ASIVA14—Analog Simulation and Variability Analysis for 14nm designs).

Abbreviations
CCP, Central Coverage Probability; CI, Confidence Interval; CV, Coefficient of Variation; DOE, Design Of Experiments; HISMC,Hybrid Importance Sampling Monte Carlo; IC, Integrated Circuit; iid, independent and identically distributed; IPs, Intellectual Properties; ISMC, Importance Sampling Monte Carlo; LASSO, Least Absolute Shrinkage and Selection Operator; LHS, Latin Hypercube Sampling; MC, Monte Carlo; MCR, Misclassification Error; MLE, Maximum Likelihood Estimation; MSE, Mean Squared Error; pdf, probability density function; PLSR, Partial Least Squares Regression; r.v., random vector.

Availability of data and materials
The data that support the findings of this study are available from Mentor Graphics but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of Mentor Graphics.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The first author AKT wrote this manuscript and performed all the experiments. XJ did help to find the research direction, arranged the circuit examples and reviewed this work closely together with TGJB who also supported with fine-tuning and proofreading of the manuscript. WHAS followed the work closely, arranged the meetings with the expertise in the field and made useful suggestions. All the authors read and approved the final manuscript.

Author details
[1]Department of Mathematics and Computer Science, University of Technology, Eindhoven, The Netherlands. [2]Mentor Graphics, Grenoble, France.

Endnotes
[a] We use the SPICE-like Eldo® simulator from Mentor Graphics® to perform the circuit simulations.
[b] Note that, in this paper, we only consider a scalar response of the circuit. However, for the cases where a circuit has multiple responses, the algorithm proposed in this paper has to be repeated for each output, individually. This process will reduce the overall speedup of the proposed method. For such cases a further research is required.
[c] The columns of $\mathbf{S}^*$ corresponding to irrelevant variables (complement of $\mathbf{x}_r$) are set to zeros before evaluating the outputs $\mathbf{H}^*$.
[d] The purpose of selecting these points is to improve the Kriging predictor in the margin of uncertainty.

Received: 17 January 2018 Accepted: 22 October 2018 Published online: 25 October 2018

References

1. Tyagi AK, Jonsson X, Beelen TGJ, Schilders WHA. Speeding up rare event simulations using Kriging models. In: Proceedings of IEEE 21st workshop on signal and power integrity (SPI). Baveno: IEEE; 2017.
2. Ciampolini L, Lafont J-C, Drissi FT, Morin J-P, Turgis D, Jonsson X, Desclèves C, Nguyen J. Efficient yield estimation through generalized importance sampling with application to NBL-assisted SRAM bitcells. In: Proceedings of the 35th international conference on computer-aided design. ICCAD '16. New York: ACM; 2016.
3. Haldar A, Mahadevan S. Probability, reliability and statistical methods in engineering design. New York: Wiley; 2000.
4. Singhee A, Rutenbar RA. Statistical blockade: very fast statistical simulation and modeling of rare circuit events and its application to memory design. IEEE Trans Comput-Aided Des Integr Circuits Syst. 2009;28(8):1176–89.
5. Santner T, Williams B, Notz W. The design and analysis of computer experiments. Berlin: Springer; 2003.
6. Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge: MIT Press; 2006.
7. Jourdain B, Lelong J. Robust adaptive importance sampling for normal random vectors. Ann Appl Probab. 2009;19(5):1687–718.
8. Homem-de-Mello T, Rubinstein RY. Estimation of rare event probabilities using cross-entropy. In: Proceedings of the winter simulation conference. vol. 1. San Diego, CA, USA. 2002. p. 310–9.
9. Kroese DP, Porotsky S, Rubinstein RY. The cross-entropy method for continuous multi-extremal optimization. Methodol Comput Appl Probab. 2006;8(3):383–407.
10. Alpaydin E. Introduction to machine learning. London: MIT Press; 2010.
11. Rosipal R, Krämer N. Overview and recent advances in partial least squares. In: Saunders C, Grobelnik M, Gunn S, Shawe-Taylor J, editors. Subspace, latent structure and feature selection. Berlin: Springer; 2006. p. 34–51.
12. Li G-Z, Zeng X-Q, Yang JY, Yang MQ. Partial least squares based dimension reduction with gene selection for tumor classification. In: 2007 IEEE 7th international symposium on BioInformatics and BioEngineering. Boston, MA, USA. 2007. p. 1439–44.
13. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3:1157–82.
14. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc, Ser B, Methodol. 1996;58(1):267–88.
15. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Stat. 2004;32(2):407–99.
16. Tibshirani RJ. The lasso problem and uniqueness. Electron J Stat. 2013;7:1456–90.
17. Matheron G. Principles of geostatistics. Econ Geol. 1963;58:1246–68.
18. Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. Stat Sci. 1989;4:409–35.
19. Lophaven SN, Nielsen HB, Sondergaard J. DACE: a Matlab Kriging toolbox, version 2.0. Technical University of Denmark, DK-2800 Kgs. Lyngby—Denmark. 2002.
20. Dubourg V. Adaptive surrogate models for reliability analysis and reliability-based design optimizations [PhD thesis]. Clermont-Ferrand, France: Blaise Pascal University—Clermont II; 2011.
21. McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 1979;21(2):239–45.
22. Dubrule O. Cross validation of Kriging in unique neighborhood. Math Geol. 1983;15(6):687–99.
23. Li J, Xiu D. Evaluation of failure probability via surrogate models. J Comput Phys. 2010;229:8966–80.
24. Li J, Li J, Xiu D. An efficient surrogate-based method for computing rare failure probability. J Comput Phys. 2011;230:8683–97.
25. Butler T, Dawson C, Wildey T. Propagation of uncertainties using surrogate models. SIAM/ASA J Uncertain Quantificat. 2013;1:164–91.