Kriging-Assisted Ultra-Fast Simulated-Annealing Optimization of a Clamped Bitline Sense Amplifier

Oghenekarho Okobiah[1], Saraju P. Mohanty[2], Elias Kougianos[3], and Oleg Garitselov[4]
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu)
University of North Texas, Denton, TX 76207, USA.
E-mail ID: oo0032@unt.edu[1], saraju.mohanty@unt.edu[2], eliask@unt.edu[3], and omg0006@unt.edu[4]

***Abstract*—Simulations using SPICE provide accurate design exploration but consume a considerable amount of time and can be infeasible for large circuits. The continued technology scaling requires that more circuit parameters are accounted for along with the process variation effects. Regression models have been widely researched and while they present an acceptable accuracy for simulation purposes, they fail to account for the strong correlation effect between parameters on the design. This paper presents an ultra-fast design-optimization flow that combines correlation-aware Kriging metamodels and a simulated annealing algorithm that operates on them. The Kriging-based method generates metamodels of a clamped bitline sense amplifier circuit which take into account the effects of correlation among the design and process parameters. A simulated annealing based optimization algorithm is used to optimize the circuit through the Kriging metamodel. The results show that the Kriging metamodels are very accurate with very low error. The optimization algorithm finds an optimized precharge time while keeping power consumption as constraint in an average execution time of 2.78 ms, as compared to a 45 minutes for an exhaustive search of the design space; i.e. close to $10^6\times$ faster. To the best of the authors' knowledge this is the first paper that uses Kriging and simulated annealing for nano-CMOS design.**

***Keywords*-Kriging Methods, Metamodeling, DRAM, Sense Amplifier, Fast Design Optimization, Simulated Annealing**

I. Introduction and Contributions

Computer simulations for the design and optimization of analog/mixed-signal (AMS) circuits often consumes a considerable amount of time. The continued scaling and increasing complexity of nanoscale technology increases the number of design factors and process parameters that affect the performance of AMS circuits. In addition, the effects of process variation now has to be taken into consideration during the design process. These effects increase the already enormous time for an exhaustive simulation search of the design space and makes design optimization a very time consuming task. To increase the speed of design space exploration, designers resort to other alternatives such as interpolating functions, fast algorithms and metamodeling.

Metamodels are approximations of the behavior, output, or figure-of-merit (FoM), of a simulated design model in response to inputs or design parameters [1]. In essence, a metamodel is an abstraction of the design model itself. The use of metamodels in circuit design allows the designer to efficiently explore the design space. With metamodels, the time for design optimization is significantly reduced while providing a reasonably close output when compared to an exhaustive search for an optimal design. Commonly used metamodeling techniques include linear and low-order polynomial regressions [1], [2], [3], [4], and neural networks [5], [6], [7], [8], [9]. Interpolating functions, which include linear and low-order polynomial regression techniques, are one of the most popular methods used by designers. They provide an accurate description of the local design space but are not effective when applied globally [5], [10]. Regression based techniques assume that errors due to process variation across the design space are random, and they approximate this error equally over the surface points on the metamodel. For designs and processes in which the error due to variation is significantly correlated between the design parameters across the local and global space, regression based metamodels do not provide an accurate fit. The technology scale into deep nanometer regions significantly increases the correlation effects between parameters, hence there is a need for design methods which accurately capture and model these effects in the design process.

This paper presents a design methodology that uses a metamodeling technique based on Kriging prediction methods and uses a simulated annealing based optimization algorithm for design optimization. The Kriging based metamodel takes into consideration the error correlation between design inputs. Kriging prediction techniques were originally used in the geostatistics field and have now been used in other fields [2], [11], [12] and only recently in VLSI [13]. In generating the metamodel, the Kriging technique predicts responses based on regression with observed data from surrounding data points. This differs from conventional regression techniques because for each predicted point, a new set of weights is calculated based on the correlations and variance of the design points in the local space. As a case study, a Kriging based metamodel is generated for a clamped bitline sense amplifier. The generated metamodel is then optimized using a simulated annealing based optimization algorithm. This methodology improves process aware design optimization reducing computational expense while providing an optimized result.

The **novel contribution of this paper** is a fast Kriging based metamodel design flow which is optimized with a simulated annealing based algorithm.

The rest of this paper is organized as follows. A brief discus-

[0]This research is supported in part by NSF awards CNS-0854182 and DUE-0942629 and SRC award P10883.

sion of selected related research is presented in Section II. The proposed design methodology is introduced in Section III. A brief background and fundamentals of Kriging metamodels are presented in Section IV. Section V briefly describes the design and characterization of the clamped bitline sense amplifier. Section VI describes the Gaussian Kriging based metamodels used in this work. Section VIII presents the conclusion and future research directions.

II. Related Prior Research

The use of metamodels for design simulation has been well researched. The most popular metamodeling technique has been the low order polynomial regression technique [12], [4], [1], [3]. In [4], a comparison of different sampling techniques used for metamodel creation is presented. While low order polynomial regression techniques are capable of generating accurate models for local optimizations, they are not very accurate in a global design space [5], [9], [14]. The weighting systems used in regression techniques are independent and are averaged over the design space. This fails to account for the spatial autocorrelation effects between input design variables. In [14] circuit designs are expressed as equations in polynomial forms. These circuit equations are reduced to form convex problems which are solved by geometric programming. This method ensures global optimization but does not result in accurate surfaces due to approximations for the circuit equations.

Neural networks (NN) have also been used to generate metamodels which have been shown to outperform regression techniques [8], [9], [7], [6]. Neural networks use a learning approach to train and adjust the weights in developing metamodels for the underlying design system. In [8], [7], well known simulation problems are used to test the accuracy of NN metamodels. Optimal metamodel generation based on neural networks is still researched actively particularly for determining the optimal network structures and the application of neural network metamodeling for point targets.

III. The Proposed Kriging-Assisted Accurate and Ultra-Fast Design Optimization Methodology

Computationally intensive simulations are very expensive. To reduce this cost, metamodels are generated to aid the design process and its optimization. Commonly used metamodel functions do not take into account the error correlation between design parameters, which is increasingly becoming significant in the deep nanometer technology range. A new methodology, Kriging Assisted Ultra-Fast Simulated Annealing Optimization design flow is proposed. Kriging techniques take into consideration the error correlation effects between design parameters. The generated metamodel is then optimized using a simulated annealing based algorithm. The methodology is incorporated in the design flow shown in Fig. 1. The design flow can be broken into 4 steps as described below.

A. Design and Netlist Optimization

The first step in the design flow is to create a model of the circuit design that meets the design specifications. The circuit

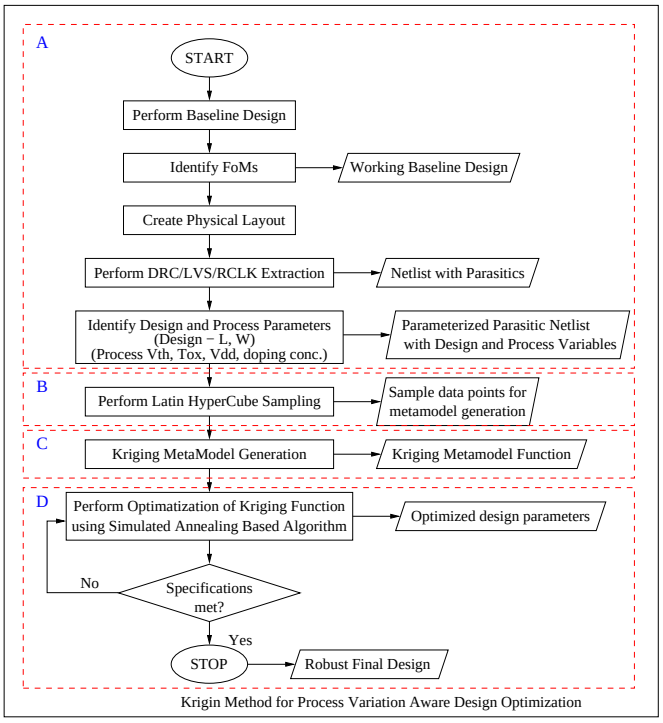


Fig. 1. The proposed Kriging assisted ultra-fast design optimization flow.

schematic is drawn and simulated using a CAD tool. After the design is verified for key performance characteristics, the physical layout design is created using Design Rule Checks (DRC) as a guide. Once the DRC is complete, a layout vs. schematic (LVS) verification is also completed to ensure that the physical design matches the circuit schematic. A parasitic netlist, including resistance, capacitance and self and mutual inductance (RCLK) is then extracted from the physical design and used for further simulations to give a more accurate description of the design. The design and process parameters are identified in the netlist which is then parameterized and used for sample point generations in the next step. In this flow, the design parameters chosen are the transistor gate length L and width W. For process parameters, threshold voltage (V_{th}), oxide thickness (T_{ox}), supply voltage (V_{DD}), and doping concentration are considered.

B. Latin HyperCube Sampling (LHS)

Latin Hypercube Sampling (LHS) techniques are one of the commonly used methods for generating sample data points for Kriging based metamodels. LHS generates n random sample points based on a range of specified inputs. The LHS technique divides the input range into n intervals of equal length, from which it randomly selects points from each interval such that the interval appears once in each row and column of a design matrix. Data points may be selected uniformly, randomly, from midpoints or in any distribution form in each interval. When the distribution used to sample points from each interval is the midpoint, the technique is called Middle Latin HyperCube Sampling (MLHS). The design points L and W are used as

the sampling corners while the process parameters are varied to model the effects of process variation.

C. Kriging Based MetaModel Generation

The sample design points generated by LHS are used with the Kriging based algorithm to generate the metamodel surface. The Kriging technique generates predicted output response points of design inputs based on observations from the sampled data. The generated metamodel is a function of the design parameters L and W, and process parameters. Two Kriging methods, ordinary Kriging and simple Kriging, are used to generate metamodels for each of the FoMs (precharge time T_{PC}, sense delay T_{SD}, and sense margin V_{SM}), of the clamped bitline sense amplifier. A total of 8 metamodels are generated and are compared to an accurate model generated by exhaustive simulation.

D. Optimization using Simulated Annealing Based Algorithm

A simulated annealing based algorithm is used to optimize the Kriging metamodels. The metamodels can be optimized for each of the identified FoMs. In this paper, the precharge time (T_{PC}) is used as the objective while the average power consumption (P_{SA}) is used as a design constraint.

IV. Fundamentals of Kriging Metamodels

Kriging methods were originally proposed in the early 1950's by Daniel Krige (hence the term "Kriging") for use in geostatistical methods. Its application has since spread into many other fields. The fundamental idea behind Kriging is that the predicted outputs are weighted averages of sampled data. The weights are unique to each predicted point and are a function of the the distance between the point to be predicted and observed points. The weights are chosen so that the prediction variance is minimized [15], [2].

The general expression of a Kriging model is as follows:

$$y(\mathbf{x_0}) = \sum_{j=1}^{L} \lambda_j B_j(\mathbf{x}) + z(\mathbf{x}), \tag{1}$$

where $y(\mathbf{x_0})$, is the predicted response at design point $(\mathbf{x_0})$ $\{B_j(\mathbf{x}), j = 1, \cdots, L\}$ is a specific set of basic functions over the design domain D_N, λ_j are fitting coefficients (also known as weights) to be determined and $z(\mathbf{x})$ is the random error. Kriging differs from common least squares based approaches in that $z(\mathbf{x})$ is assumed to be a random process and not independent, unique to each weight and not distributed identically. It is assumed that the process has a known mean, variance σ^2, and correlation function. The correlation function, called the *variogram* in geophysics, is expressed as follows:

$$r(\mathbf{s}, \mathbf{t}) = \mathrm{Corr}(z(\mathbf{s}), z(\mathbf{t})). \tag{2}$$

The variogram is used to derive the Kriging weights, λ_j. The autocorrelation of the design points is characterized by the covariance function [16]. The weights are chosen so that the Kriging variance is minimized. There are different variations of Kriging models. Two methods explored in this paper are the ordinary and simple Kriging techniques. Ordinary Kriging assumes a mean that is constant in the local domain of a predicted point, while simple Kriging assumes a constant and known mean over the global domain.

For ordinary Kriging techniques, the weights are chosen to minimize the Kriging variance under the unbiasedness constraint that $E(\widehat{Z}(x) - Z(x)) = 0$. Hence the weights are chosen so that the following expression is satisfied:

$$\sum_{j=1}^{n} \lambda_j = 1. \tag{3}$$

This condition is not required for simple Kriging. The weights then for ordinary Kriging are given by the following:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{pmatrix} = \Gamma^{-1} \begin{pmatrix} \gamma(e_1, e_0) \\ \vdots \\ \gamma(e_n, e_0) \\ 1 \end{pmatrix}, \tag{4}$$

where μ is a Lagrange multiplier used to ensure equation (3). Γ is the covariance matrix of the observed points and for ordinary Kriging is given by:

$$\Gamma = \begin{pmatrix} \gamma(e_1, e_1) & \cdots & \gamma(e_1, e_n) & 1 \\ \vdots & \ddots & \vdots & 1 \\ \gamma(e_n, e_1) & \cdots & \gamma(e_n, e_n) & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \tag{5}$$

where

$$\gamma(e_1, e_2) = E\left(|z(e_1) - z(e_2)|^2\right). \tag{6}$$

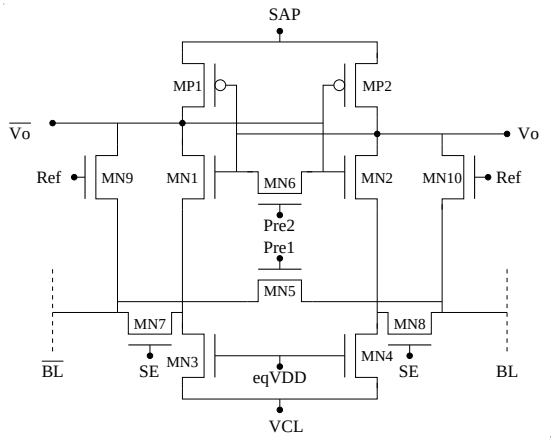
The last row and column are absent in simple Kriging method.

V. The 45 nm Clamped Bitline Sense Amplifier: A Case Study Circuit

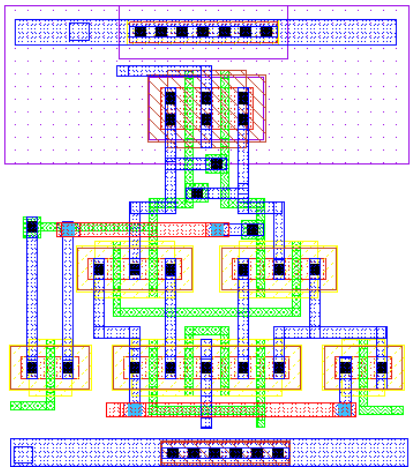
A. The Clamped Bitline Sense Amplifier Circuit Design

The clamped bitline sense amplifier is a variation of the conventional sense amplifier used in DRAMs. The advantage of the clamped bitline is that it is clamped to a stable voltage after a sensing operation. This reduces the capacitive effect of the bitlines during the sensing operation, hence resulting in a decreased dynamic power consumption and sense delay time [17], [18]. Fig. 2(a) shows the circuit schematic design of the clamped bitline sense amplifier. Transistors MP1, MP2, MN1 and MN2 form the cross-coupled inverters, while transistors MN3 and MN4 provide a low impedance between the bitlines through V_{CLAMP}.

The initial design parameters for the transistors are length L_n, L_p = 45 nm, width W_n = 120 nm, and W_p = 240 nm. These dimensions are based on the nominal 45 nm technology node values and similar designs in [19]. The clamped bitline sense amplifier needs matched transistors for optimal performance, making it a good test circuit to model the effects of process variation. The physical layout design is shown in Fig. 2(b). The extracted SPICE netlist from the layout includes the parasitics of the design which impact the its performance as seen in Table I.



(a) Schematic design.



(b) Physical design.

Fig. 2. Circuit and layout for the clamped bitline sense amplifier.

B. Characterization of 45 nm Clamped Bitline Sense Amplifier

In characterizing the performance of the sense amplifier design, the following figures of merit (FoM) were selected based on previous publication [19].

Precharge and Voltage Equalization Time is the time required to equally precharge both bitlines BL and $\overline{BL}$. This reduces power consumption during the sense operation by reducing the voltage swing. The capacitance of the bitline significantly affects the precharge time.

Power Consumption is the average power consumed by the clamped bitline sense amplifier. The average power measured includes dynamic power, subthreshold leakage and gate oxide leakages. With technology scaling now in the deep nanometer regions, the leakage power components now contribute significantly to power consumption.

Sense Delay is the minimum amount of time required for sufficient voltage to appear on the bitlines that can be correctly detected by the sense amplifiers. The cell data value affect the sense delay. The impact of the bitline capacitance on the sense delay is reduced by the design of the clamped bitline.

Sense Margin is the minimum voltage that can be correctly detected by the clamped bitline.

The circuit schematic and the physical design were both simulated for verification and characterization. The performance was characterized based on the selected FoMs. Table I shows a summary of these values. The last column also shows the area of the physical design.

VI. Kriging Metamodeling of the Clamped Bitline Sense Amplifier

A. Kriging Model Generation for the FoMs

The extracted netlist from the physical layout is parameterized and used to generate sample data points using the LHS technique. Two Kriging methods are used to generate the metamodels: (1) Simple Kriging and (2) Ordinary Kriging. As discussed in section III-C, each Kriging predicted point is calculated with a different weight. The weights are based on the empirical semivariogram. Hence, the covariance functions were determined to obtain the spatial autocorrelation of the design parameters. For this paper, to simplify the analysis, only W_n has been used as a design parameter. A parametric analysis varying W_n and W_p shows that the FoMs are dominated by W_n. The topology of the circuit supports this trend: there are 10 NMOS transistors compared to 2 PMOS transistors. The use of only W_n has been used to illustrate the proposed methodology and in future work, the approach will be extended to designs with multiple design parameters.

The empirical variogram is estimated from the created variogram. It is then fitted with the theoretical spherical model, which was the best fit for the sampled data points. Each FoM can be expressed based on the general form of the Kriging function. For example, the predicted precharge time $\widehat{Y}_{pr}$ at an unknown design point W_n^* is expressed as:

$$\widehat{Y}_{pr}\left(W_n^*\right) = \sum_{i=1}^{N} \lambda\left(W_n^*\right)_i Y_{pr}\left(W_{n_i}\right), \tag{7}$$

where $Y_{pr}(W_{n_i})$ are the observed precharge values for the given N W_{n_i} $(i = 1, 2, \ldots, N)$ sample points. The weights $\lambda(W_n^*)$ are unique for each predicted point W_n^* and are calculated from Eqn. (4). Using similar equations, the values for the other FoMs of the sense amplifier are predicted.

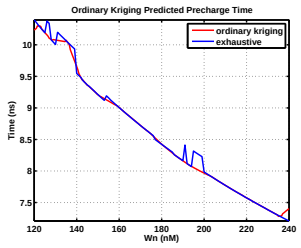
B. Kriging Metamodels and Accuracy Analysis

The generated metamodels for the FoMs are presented in this section. An exhaustive baseline simulation was also done to compare the accuracy of the Kriging predicted models. A total of 1000 design points were simulated to densely capture the design space compared to the 20 and 100 LHS points used to generate the Kriging surfaces.

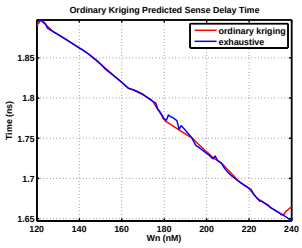
The predicted curves for the ordinary Kriging based metamodels are shown in Fig. 3 with W_n as the design input. The results for simple Kriging are very similar and are omitted due

TABLE I
FIGURES OF MERIT OF THE OPTIMAL CLAMPED BITLINE SENSE AMPLFIER.

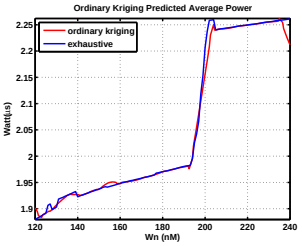
Design	Precharge time, T_{PC} (ns)	Sense delay, T_{SD} (ns)	Power, P_{SA} (μW)	Sense Margin, V_{SM} (mV)	Area μm^2
Schematic	10.31	1.79	1.84	26.91	-
Layout	10.40	1.91	1.88	26.86	6.045
Optimized	8.16	1.68	1.98	28.03	6.356
Change	21.54 %	12.04 %	-5.32 %	-4.36 %	5.15 %



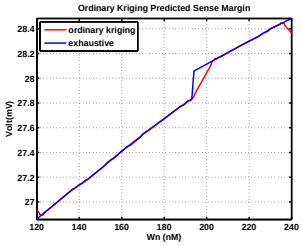
(a) Ordinary Kriging for precharge time



(b) Ordinary Kriging for sense delay



(c) Ordinary Kriging for average power



(d) Ordinary Kriging for sense margin

Fig. 3. Ordinary Kriging responses using W_n as the design parameter.

to space constraints. The plots also show the exhaustive design points simulations. From the plots it is seen that the predicted Kriging metamodels for both the ordinary and simple Kriging techniques closely match the exhaustive simulation.

A statistical analysis on both responses shows that the accuracy of the Kriging method is very high. A summary of the statistical analysis is shown in Table II for both ordinary and simple Kriging metamodels compared to the exhaustive design surface. The metrics used for comparison are the Mean Square Error (MSE), the Root Mean Square Error (RMSE) and the correlation coefficient R^2:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)^2}, \tag{8}$$

where N is the number of design points predicted.

From an analysis of the results in Table II the predicted points have an average R^2 of 0.99. The simulation time for the generation of the metamodels was 3 mins compared to 72 hrs used for exhaustive simulation.

TABLE II
STATISTICAL ANALYSIS OF THE KRIGING PREDICTED VALUES.

FoMs	Ordinary Kriging		Simple Kriging	
Samples	20	100	20	100
Precharge				
MSE	6.02×10^{-21}	3.85×10^{-19}	5.32×10^{-21}	3.63×10^{-19}
$RMSE$	7.76×10^{-11}	6.20×10^{-10}	7.29×10^{-11}	6.02×10^{-10}
R^2	0.9931	0.5560	0.9939	0.5810
STD	6.95×10^{-11}	6.09×10^{-10}	6.60×10^{-11}	5.91×10^{-10}
Sense Delay				
MSE	1.12×10^{-23}	8.27×10^{-24}	7.49×10^{-24}	4.02×10^{-24}
$RMSE$	1.02×10^{-10}	2.88×10^{-12}	2.73×10^{-12}	2.00×10^{-12}
R^2	0.9984	0.9985	0.9987	0.9993
STD	8.62×10^{-11}	2.64×10^{-12}	2.29×10^{-12}	1.79×10^{-12}
Power				
MSE	3.64×10^{-15}	4.35×10^{-15}	3.56×10^{-15}	4.69×10^{-15}
$RMSE$	6.24×10^{-11}	6.60×10^{-08}	5.96×10^{-08}	6.85×10^{-08}
R^2	0.9957	0.8145	0.8486	0.8003
STD	5.75×10^{-11}	6.40×10^{-08}	5.69×10^{-08}	6.66×10^{-08}
Sense Margin				
MSE	2.79×10^{-09}	6.31×10^{-09}	2.56×10^{-09}	4.32×10^{-09}
$RMSE$	5.28×10^{-05}	7.94×10^{-05}	5.06×10^{-05}	6.57×10^{-05}
R^2	0.9987	0.9753	0.9900	0.9831
STD	2.58×10^{-05}	7.73×10^{-05}	4.79×10^{-05}	6.41×10^{-05}

C. Experimental Setup

The Cadence virtuoso platform was used for the initial circuit schematic design and the physical layout. The extracted and parameterized netlists were used to write Ocean Scripts that were used to run the exhaustive simulation and gather LHS sample data points. The Spectre analog simulator was used to perform the simulations. The algorithm used to generate the Kriging metamodels was written using MATLAB with the help of the toolboxes mGstat [20] and SUMO [21].

VII. SIMULATED ANNEALING BASED OPTIMIZATION

Simulated annealing optimization is based on the Monte Carlo algorithm and was originally used to simulate the annealing process used in metallurgy. This gives the simulated annealing algorithm random characteristics. Successive runs of the algorithm will produce different results. The optimization steps are presented in Algorithm 1.

The algorithm takes random walks through the design space starting from the middle point of each design parameter, looking for points with low energies. In each step, the probability of taking a step is determined by the Boltzmann distribution, $p = \left(e^{\frac{\Delta_{T_{PC}}}{T}}\right)$ if $\Delta_{T_{PC}}$ is high, and p = 1 when $\Delta_{T_{PC}}$ is low. Therefore a step will occur if a new value is better than the previous one. If the new value is worse, the transition can still

Algorithm 1 Simulated-Annealing Based Optimization of the Clamped-Bitline Sense Amplifier.

Initialize iteration counter: $counter \leftarrow 0$.
Initialize temperature Θ.
Initialize $Cooling\_Rate$.
Start with an initial solution $\widehat{CBSA}_i$.
Calculate the FoMs for $\widehat{CBSA}_i$ using the Kriging models.
Consider the objective of interest T_{PC_i}.
$result \leftarrow \Delta_{T_{PC}} \leftarrow T_{PC_i}$.
while ($\Delta_{T_{PC}} != 0$) **do**
  $counter \leftarrow max\_Iteration$.
  while ($counter > 0$) **do**
    Generate random transition from solution $\widehat{CBSA}_i$ to $\widehat{CBSA}_j$.
    Calculate the FoMs for $\widehat{CBSA}_j$ using the Kriging models.
    if ($T_{PCj} < result$) **then**
      $result \leftarrow T_{PCj}$.
      $\widehat{CBSA}_i \leftarrow \widehat{CBSA}_j$.
    else
      $\Delta_{T_{PC}} \leftarrow T_{PC_i} - T_{PCj}$.
      if ($\Delta_{T_{PC}} < 0$, random(0,1) $< e^{\frac{\Delta_{T_{PC}}}{T}}$) **then**
        $T_{PC_i} \leftarrow T_{PCj}$.
        $\widehat{CBSA}_i \leftarrow \widehat{CBSA}_j$.
      end if
    end if
    $counter \leftarrow counter - 1$.
  end while
  $\Theta \leftarrow \Theta \times Cooling\_Rate$.
end while
return $result$ and $\widehat{CBSA}_i$.

occur, and its likelihood is proportional to the temperature T and inversely proportional to $\Delta_{T_{PC_i}}$.

The finalized values for the design are shown in Table I. T_{PC} has been reduced by 21.54 % while P_{SA} was increased by 5.32 %. T_{SD} and V_{SM} was also improved by 12.04 % and 4. 36%, respectively. The area for the final layout design was also increased by 5.15%. The simulated annealing based algorithm finds optimized values in 2.78 ms compared to a run of 45 minutes for an exhaustive search optimization. In other words, the proposed design flow could speedup the optimization process by a factor approximately $10^6\times$.

VIII. Conclusions and Future Research

This paper presented a new methodology that uses Kriging metamodels and the simulated annealing algorithm for sense amplifier optimization. Kriging methods generate metamodel functions that accurately capture the global design space while taking into account the spatial autocorrelation of the input design parameters. Comparisons with exhaustive simulations show that Kriging predicted models are very accurate with very low RMSE and high R^2. The simulated annealing based algorithm optimized the generated metamodel function for the precharge T_{PC} FoM, improving it by 21.54%. In future research, the methodology will be extended to multiple design parameters and multi-objective optimization algorithms.

References

[1] W. E. Biles, J. P. C. Kleijnen, W. C. M. van Beers, and I. van Nieuwenhuyse, "Kriging Metamodeling in Constrained Simulation Optimization: An Explorative Study," in *Proceedings of the 39th Winter Simulation Conference*, 2007, pp. 355–362.

[2] W. Van Beers, "Kriging Metamodeling in Discrete-Event Simulation: An Overview," in *Proceedings of the Winter Simulation Conference*, 2005, pp. 202–208.

[3] G. Dellino, J. Kleijnen, and C. Meloni, "Robust Simulation-Optimization using Metamodels," in *Proceedings of the Winter Simulation Conference (WSC)*, Dec. 2009, pp. 540–550.

[4] O. Garitselov, S. Mohanty, E. Kougianos, and P. Patra, "Nano-CMOS Mixed-Signal Circuit Metamodeling Techniques: A Comparative Study," in *Proceedings of the International Symposium on Electronic System Design (ISED)*, 2010, pp. 191–196.

[5] B. Ankenman, B. Nelson, and J. Staum, "Stochastic Kriging for Simulation Metamodeling," in *Proceedings of the Winter Simulation Conference*, Dec. 2008, pp. 362–370.

[6] R. A. Kilmer, A. E. Smith, and L. J. Shuman, "Computing Confidence Intervals for Stochastic Simulation Using Neural Network Metamodels," *Computers and Industrial Engineering*, vol. 36, no. 2, pp. 391–407, 1999.

[7] I. Sabuncuoglu and S. Touhami, "Simulation Metamodelling With Neural Networks: An Experimental Investigation." *International Journal of Production Research*,, vol. 40, no. 11, pp. 2483–2505, 2002.

[8] C. W. Zobel and K. B. Keeling, "Neural Network-based Simulation Metamodels for Predicting Probability Distributions," *Computers and Industrial Engineering*, vol. 54, pp. 879–888, May 2008.

[9] A. Khosravi, S. Nahavandi, and D. Creighton, "Developing Optimal Neural Network Metamodels Based on Prediction Intervals," in *International Joint Conference on Neural Networks*, June 2009, pp. 1583–1589.

[10] J. Staum, "Better Simulation Metamodeling: The Why, What, and How of Stochastic Kriging," in *Proceedings of the Winter Simulation Conference (WSC)*, Dec. 2009, pp. 119–133.

[11] B. Harrington, Y. Huang, J. Yang, and X. Li, "Energy-Efficient Map Interpolation for Sensor Fields Using Kriging," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 622–635, May 2009.

[12] M. Zakerifar, W. Biles, and G. Evans, "Kriging Metamodeling in Multi-objective Simulation Optimization," in *Proceedings of the Winter Simulation Conference (WSC)*, 2009, pp. 2115–2122.

[13] H. You, M. Yang, D. Wang, and X. Jia, "Kriging Model Combined with Latin Hypercube Sampling for Surrogate Modeling of Analog Integrated Circuit Performance," in *Proceedings of the International Symposium on Quality of Electronic Design*, 2009, pp. 554–558.

[14] V. Aggarwal, "Analog Circuit Optimization using Evolutionary Algorithms and Convex Optimization," Master's thesis, Massachusetts Institute of Technology, May 2007.

[15] N. A. C. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1993.

[16] G. Bohling, "Kriging," Kansas Geological Survey, Tech. Rep., 2005.

[17] I. Arsovski, "High-Speed Low-Power Sense Amplifier Design," University of Toronto, Tech. Rep., 2001.

[18] T. Blalock and R. Jaeger, "A Subnanosecond Clamped-Bit-Line Sense amplifier for 1T Dynamic RAMs," in *Proceedings of the International Symposium on VLSI Technology, Systems, and Applications*, 1991, pp. 82–86.

[19] O. Okobiah, S. P. Mohanty, E. Kougianos, and M. Poolakkaparambil, "Towards Robust Nano-CMOS Sense amplifier Design: A Dual-Threshold versus Dual-Oxide Perspective," in *Proceedings of the 21st ACM Great Lakes Symposium on VLSI*, 2011, pp. 145–150.

[20] *mGstat: A Geostatistical Matlab Toolbox*. [Online]. Available: mgstat.sourcefourge.net

[21] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design," *J. Mach. Learn. Res.*, vol. 11, pp. 2051–2055, August 2010.