Exploring Kriging for Fast and Accurate Design Optimization of Nanoscale Analog Circuits

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

***Abstract*—The increasing complexity of modern electronic devices driven by consumer demand and technological advancements presents significant challenges for designers. The reduced feature size and increased capabilities lead to more complex designs as more sub-systems are packed into a single chip. Traditional synthesis and optimization methods which involve CAD tools for accurate simulation are computationally time expensive and even become infeasible especially in designs using nanoelectronic technology due to increased design factors and the exponentially increasing design space. The current objective is to explore techniques that produce optimal designs while reducing the design effort. Metamodeling techniques have been used in this respect to reduce the cost of manual iterative circuit sizing during synthesis. Existing metamodeling techniques however are unable to capture the effects of process variation which are dominant in deep nanometer regions. This work explores Kriging techniques for fast and accurate design optimization of nanoscale analog circuits.**

***Keywords*—Analog Mixed-Signal (AMS), Nano-CMOS, Process Variation, Geostatistics, Kriging, Neural Network, Optimization**

I. Introduction

The design of Analog Mixed-Signal Systems-on-a-Chip (AMS-SoCs) presents a difficult challenge given the number of design specifications that must be met. The demand for smaller and yet more powerful electronic devices along with the progressive scaling of the technology only aggravates the problem. The reduced feature sizes and increased capabilities lead to more complex designs as more sub-circuit systems are packed into a single chip. The continuous progression of semiconductor technology in itself further worsens the design process as designers now need to deal with issues of subthreshold leakage and power density. A more prominent issue which arises in designs using nanoelectronic technology is the issue of process variation. With more complex designs, and the increase in the number of design and process parameters that must be considered to mitigate the issues of subthreshold leakage and process variation, circuit sizing synthesis for optimal analog designs presents a major burden to designers. The use of existing electronic design automation (EDA) tools for exhaustive design space exploration incurs expensive computer SPICE simulation time and continues to grow exponentially. For example the complete silicon aware (parasitic) simulation of a circuit system like a Phase Locked Loop (PLL) on a CAD tool could take days or even weeks.

The current objective is to explore techniques that produce optimal designs comparable to results using CAD tools while reducing the design effort of designers. Conventional techniques include the use of metamodels to create a quick behavioral response of the circuit design which can be used for design analysis and optimization before completing the final sizing of the physical design. A metamodel is an abstraction of the design, approximating the behavior of the circuit response to the most sensitive design parameters [1]. The aim of metamodeling designs is to reproduce as accurately as possible simulation results close to ones produced by CAD tools while abstracting the time intensive cost efforts of computer simulations, providing the designers a fast, accurate and efficient way to explore the design space during circuit sizing synthesis without manually resizing the layout schematics. Common metamodeling techniques include response-surface modeling, linear and low-order polynomial regression functions, and artificial neural networks (ANNs) [2]–[4].

In creating metamodels based on low-order polynomial regression for nanoCMOS designs, the regression models assume the effects of process variation are purely random and thus approximate the error equally across the design space. However, the effects are not random but are strongly correlated. Kriging prediction techniques take into account the correlation effects between input parameters and also incorporate a stochastic component for performance point prediction that produces a more statistical accurate circuit description. The weighting system for each point predicted is unique and calculated using the variogram for characterizing the autocorrelation effects between the design parameters. This unique weighting property however can potentially be time consuming for large design spaces. This work proposes Kriging based metamodel designs and also combines a Kriging-bootstrapped ANN metamodel that mitigates the potential time cost of Kriging on large design spaces. The use of Kriging based metamodels thus provides a more robust process variation aware design increasing the yield and reducing design effort. The **novel contributions** of this work are the following: (1) A layout-accurate method for geostatistics-based Kriging metamodel generation, (2) A novel ultra-fast but accurate layout design optimization flow for AMS components that incorporates layout and process-aware metamodels into different levels of the design process, and (3) A layout optimization technique for AMS-SoCs blocks with novel Ant Colony (ACO), Simulated Annealing (SA), and Gravitational Search Algorithm (GSA) algorithms.

The rest of this paper is organized as follows: A brief discussion on the proposed design flow approach is presented in section II. Section III presents the background of Kriging

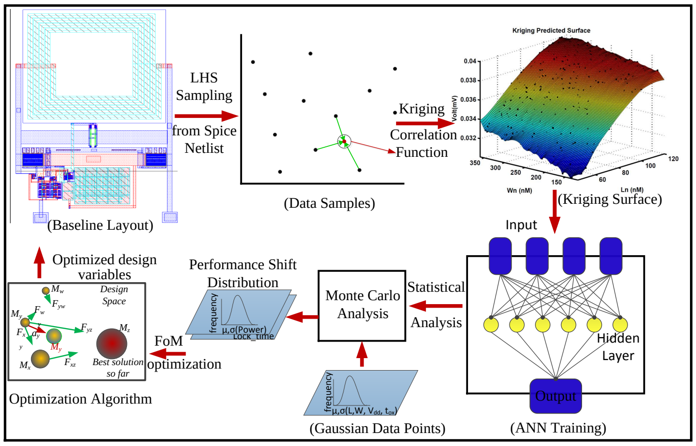

Fig. 1. Proposed design flow for metamodel-assisted ultra-fast optimization.

techniques. An overview of the novel optimization algorithms is presented in section IV. A discussion of the experimental results, analysis and comparisons is presented in section V and finally, a summary conclusion and future research ideas are presented in section VI.

II. Proposed Kriging-ANN Metamodel Method

Kriging techniques achieve modeling of process variation effects by taking into account the correlation between design parameters especially for designs in the deep nanometer range. The details of Kriging metamodeling are discussed in Section III. The drawback of Kriging is the calculation of the weighting matrix equations that are used are potentially time expensive for very large design spaces. In order to limit this time factor, the characteristics of Kriging are combined with ANN based metamodels. ANN metamodels have been explored for nanoscale analog design and are reported to be robust, fast and accurate for high dimensional designs [3]. ANN metamodels are still not able to effectively model the effects of process variation. The proposed process combines the Kriging based metamodels by producing an intermediate set of sample points (bootstrapping) which are then used to train an ANN metamodel. The bootstrapped sample data generated by Kriging techniques incorporates the process aware component into the ANN generated metamodel. Optimization algorithms are then used to explore the design space over the generated Kriging-ANN metamodel resulting in final optimized designs which are used for the final sizing of the physical layout. A high level general design flow methodology of this process is shown in Fig. 1. The proposed technique produces more process aware accurate metamodels than plain ANN metamodels which are computationally less time expensive than plain Kriging metamodels.

A more detailed process is shown in Fig. 2. The first phase, labeled A, constitutes the baseline logical and physical design and functional verification of the circuit. The schematic and physical designs are simulated for functional verification of the performance objectives of the circuit design. The next phase labeled B involves the generation of the process aware metamodel. A full blown (RLCK) netlist is used to ensure silicon accuracy of the design. The sample points are generated using LHS techniques which are then fed into the Kriging function generator to generate an intermediate set of data points infusing process aware characteristics. The bootstrapped points are used to generate the Kriging-ANN metamodels using an ANN trainer. Finally the last phase is the design optimization by exploration with optimization algorithms.

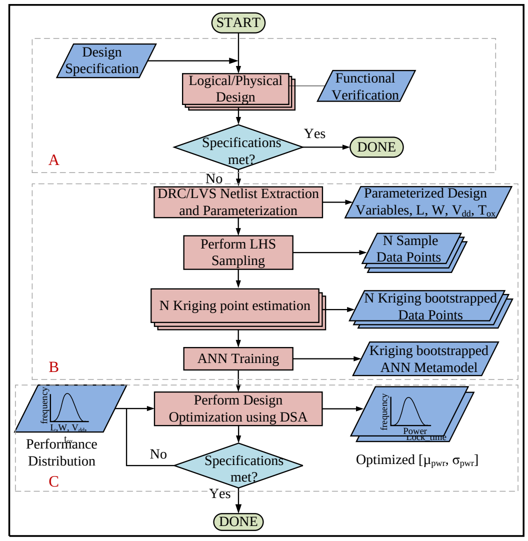

Fig. 2. Proposed process variation design optimization flow.

III. Kriging Metamodeling

A. Kriging Based Metamodeling

Kriging prediction techniques, originally applied in geostatistical fields, have since been explored for other applications such as circuit design [5], [6]. Kriging combines polynomial regression with a stochastic approach to mitigate the deterministic nature of computer expressions. Kriging equations can be expressed in the form of:

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

where $y(\mathbf{x_0})$ is a stochastic function which predicts the response of the design point $(\mathbf{x_0})$. $\{B_j(\mathbf{x}), j = 1, \cdots, L\}$ is a specific set of basis functions over the design domain D_N, λ_j are fitting coefficients (also known as weights) to be determined based on the Kriging method applied.

In calculating the weights λ_j for estimating Kriging functions, the autocorrelation between the input parameters is accounted and characterized by the covariance function [7]:

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

This property of Kriging prediction techniques is explored to model the effects of process variation on circuit design metamodels where the correlation between the process variation of the design and process parameters is taken into consideration. The Kriging metamodel process is shown in Fig. 3.

B. Artificial Neural Network Metamodeling

ANN models consist of simple computational elements with a rich interconnection between each element. They are

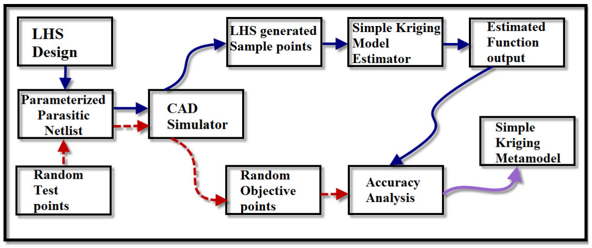

Fig. 3. Geostatistical Kriging metamodel generation process.

modeled after biological neural networks which operate in a parallel and distributed fashion. The neural networks create models over a set of inputs by training the weights of the interconnections. Multilayer and radial neural networks are few of the commonly employed neural networks. The multilayer network which is used in this work uses a combination of non-linear activation function in a hidden layer and a linear activation function in the output layer. The linear layer of the function output can be expressed as follows:

$$v_i = \sum_{i=1}^{s} w_{ji} x_i + w_{j^0}, \tag{3}$$

where w_{ji} is the weight of the connection between the jth element in the hidden layer and the ith component in the input layer x_i. The input layer is represented using a sigmoid function such as follows:

$$b_j\left(\nu_i\right) = \tanh\left(\lambda v_j\right) \tag{4}$$

For this work, the ANN metamodel was created using a MATLAB toolbox which implements the Levenberg-Marquardt optimization algorithm [8]. The Kriging-ANN metamodel generation flow is shown in Fig. 4.

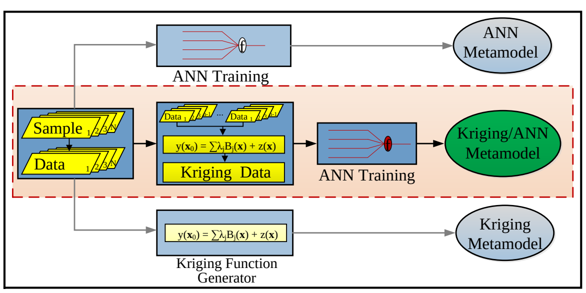

Fig. 4. Proposed Kriging based ANN metamodel generation flow

IV. Optimization Algorithm: Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) was introduced as a new heuristic optimization algorithm based on the Newtonian laws of gravity [9]. The proposed GSA algorithms models the search agents as mass objects varying in design points by the location of the masses. At the same time the performance object is modeled as the mass of the object. The heavier masses correspond to better performing agents As the agent masses become heavier, they attract other agents towards them by gravity force. In the process pull the search agents towards an area with a likely optimal solution. Agents which attract other masses become heavier and move slower. Thus, concentrating in a search area with a likely optimal solution while lighter masses are able to move faster exploring other search locations. The selection of the rate of attraction is an important step in order to fully utilize the exploration and exploitation features of this algorithm. The exploration feature is the capability of actively stratifying the design space while exploitation is the efficiency of locating optimal solution in a likely optimal area.

A high-level overview of the GSA algorithm is shown in Fig. 5. The search agents, for example a set of design parameters, are denoted by their locations and masses as M_w, M_x, M_y, and M_z in the design space. The location of each agent at any particular time is shown and the quality of solution is denoted by the mass size of the agent. M_z, currently has the best quality while M_w has the worst. The underlying principle of the algorithm is shown using the forces acting on search agent M_y as an example.

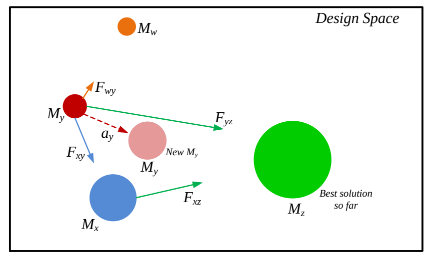

Fig. 5. Search agents are attracted towards locations with possible solutions.

V. Results and Analysis

In this section, the experimental results of the proposed design flow methodology which was illustrated with the design of an 180 nm Phase Locked Loop (PLL) design [10]. The PLL which is a closed feedback loop circuit system, is an ideal circuit for this study. It is widely used in many analog/mixed signal systems including processors, Field-Programmable Gate Arrays (FPGAs) and in telecommunication applications. The major components of the PLL are the phase detector, charge pump, voltage controlled oscillator (VCO) and frequency divider. A system diagram is shown in Figure. 6.

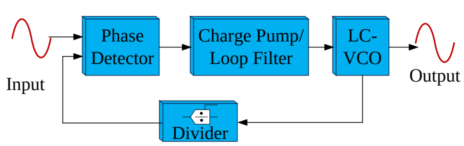

Fig. 6. High level system diagram for the PLL.

The optimization result of the proposed Kriging-ANN metamodel is shown in Table I. The power consumption is reduced by approximately 79%. The locking time is also reduced by 4 %. The result of the algorithm operation is shown in Fig. 7. It is seen from the figure that an optimal power consumption of 1.67 mW is obtained after 377 iterations. It can also be seen that the algorithm has very fast convergence rate due to its strong attractive features. On average, the GSA is able to converge to an optimal power consumption in about 400 iterations.

TABLE II. STATISTICAL ANALYSIS FOR ACCURACY OF NEURAL NETWORK METAMODEL FOR PLL FOMS.

		CircuitCircuit	Kriging-ANNKriging-ANN		KrigingKriging		ANNANN	
		Value	Value	Error (%)	Value	Error (%)	Value	Error (%)
P_{PLL}	Mean	2.4 mW	2.4 mW	3.2	2.5 mW	0.8	2.5 mW	0.8
	STD	0.4 mW	0.3 mW	19.0	0.5 mW	21.4	0.7 mW	64.3
F_{PLL}	Mean	2.6 GHz	2.5 GHz	5.6	2.6 GHz	0.1	2.7 GHz	5.4
	STD	10.9 MHz	41.9 MHz	282.9	3.7 MHz	66.0	51.9 MHz	373.9
Lck_{PLL}	Mean	5.5 μs	5.1μs	7.2	5.5 μs	0.07	5.2 μs	5.6
	STD	0.7 μs	0.4 μs	38.9	0.6 ns	10.2	1.0 μs	40.3
J_{PLL}	Mean	16.8 ns	14.7ns	10.2	16.8ns	0.1	17.9 ns	6.6
	STD	1.3 ps	4.5 ps	240.9	0.7ps	48.5	19.1 ps	1352.2

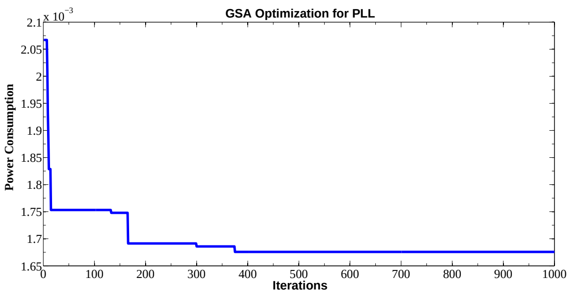

Fig. 7. Optimization steps of the PLL.

TABLE I. FINAL OPTIMIZATION RESULTS FOR THE PLL.

Metric	Power (mW)	Locking Time (ns)	Area (μm^2)
Baseline Design	8.27	2.74	525 $\times$ 326
Optimal Optimal	1.67	2.63	525 $\times$ 326
Reduction	79 %	4 %	0 %

A comparison of the Kriging, ANN, Kriging ANN metamodels is shown in Table II. The mean (μ) and standard deviation (σ) for the FoMs in each of the metamodels are presented. From the results, the Kriging metamodels are shown to be most accurate on both the mean (μ) and (σ) values for all FoMs. The Kriging bootstrapped neural network metamodel on the other hand is shown to be more accurate on the (σ) values than the plain neural network metamodel but less accurate on the (μ) values. This difference is expected because while bootstrapping infuses the autocorrelation property of Kriging based techniques, some error is also introduced as well.

Table III shows the time cost for the Monte Carlo Analysis on each metamodel. The Table shows a speedup of approximately 25 times in time cost for the Monte Carlo Simulation of 1000 runs for the Kriging bootstrapped model over traditional Kriging. The significant improvement in time cost is large enough to mitigate the minimal error incurred in the metamodel. It may be noted that the Monte Carlo simulation time on the SPICE models is approximately 5 days, which highlights the huge time gain with the use of metamodels.

TABLE III. METAMODEL TIME ANALYSIS COMPARISON.

Model	Kriging-ANN	Kriging	ANN
Time	19 s	468 s	19 s
Speedup	24.63$\times$	1	24.63$\times$

VI. CONCLUSION AND FUTURE RESEARCH

In this work, a novel design flow methodology has been proposed to reduce the effort involved in current nano-CMOS IC design. The proposed methodology uses geostatistical Kriging-ANN metamodeling techniques for fast and accurate design space exploration. Kriging based techniques generate metamodels that accurately capture the global design space along with the effects of process variation and are combined with fast ANN techniques. Comparisons with exhaustive simulations show that Kriging-ANN predicted metamodels are more process aware accurate than ANN metamodels. The accurate Kriging based metamodels have been combined with optimization algorithms which efficiently explore the design space for optimal design parameters. An illustrative design with an 180 nm PLL shows a power consumption improvement of about 79 % and significantly reducing simulation time compared to plain Kriging techniques by about 25 times. For future research, the proposed methodology could be extended to 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] 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.

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

[4] O. Okobiah, S. P. Mohanty, and E. Kougianos, "Geostatistical-Inspired Fast Layout Optimization of a Nano-CMOS Thermal Sensor," *IET Circuits, Devices Systems*, vol. 7, no. 5, pp. 253–262, Sept 2013.

[5] 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.

[6] G. Yu and P. Li, "Yield-Aware Analog Integrated Circuit Optimization Using Geostatistics Motivated Performance Modeling," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2007, pp. 464–469.

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

[8] MATLAB, *MATLAB and Neural Network Toolbox Release 2012b*. Natick, Massachusetts, United States: The MathWorks Inc., 2012.

[9] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232 – 2248, 2009.

[10] O. Garitselov, S. P. Mohanty, and E. Kougianos, "Accurate Polynomial Metamodeling-Based Ultra-Fast Bee Colony Optimization of a Nano-CMOS PLL," *J. Low Power Electronics*, vol. 8, no. 3, pp. 317–328, Jun 2012.