Geostatistics Inspired Fast Layout Optimization of Nanoscale CMOS Phase Locked Loop

Oghenekarho Okobiah[1], Saraju P. Mohanty[2], and Elias Kougianos[3]
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], and eliask@unt.edu[3]

***Abstract*—In this paper, we present a geostatistical method for design and optimization of analog and mixed signal circuits design illustrated with the design of phase locked loop (PLL) systems used in Wide Area Network (WAN) and Private Mobile Radio (PMR) applications. The proposed method incorporates the use of a geostatistic based metamodeling technique (Kriging) and optimization algorithm (gravitational search algorithm) and is compared to similar approaches. The results show that the geostatistical methods provide more accurate metamodels and more efficient optimization design techniques. To the best of the authors' knowledge, *this is the first geostatistical method for metamodeling and optimization of PLL designs*. The proposed optimization could achieve 79% reduction in PLL power with 4% reduction in locking time without any area penalty.**

I. Introduction

The design of custom analog mixed signal circuits continues to pose a problem for designers as the scale of technology decreases. With improved technologies and reduced feature size, designs become more complex as more circuits and sub-systems are packed into single Integrated Chips (IC). Furthermore, the scaling of technology deeper into the sub-micro region increases the number of parameters that must be considered in the exploration of the design space. The complexity of designs and the increase in the number of design parameters exponentially increases the time time and computer resources for an exhaustive exploration of the design space. For example the simulation of a circuit system on a CAD tool for a parasitic netlist could take many days. This is not an entirely new problem as possible solutions have been proposed by [1] in the form of metamodeling which in itself is also not new. Metamodeling techniques have also been known as response surface modeling and loosely as macro-modeling. Metamodeling is the mathematical representation and approximate description of the design performance with respect to design parameters [2], [1]. This is different from macromodeling which is a simplified approximation of the design [3]. Metamodeling techniques that have been proposed and are commonly used include polynomial regression [1], [4], neural networks [5], [6] and Kriging [7], [8].

For the fast optimization of designs, optimization algorithms are often used with metamodels. Heuristic algorithms such as simulated annealing (SA), genetic algorithms (GA), geometric programming (GP) and more recently the family of swarm intelligent algorithms have been applied to metamodels. The accuracy of the designs however is strongly dependent on the accuracy of the metamodels. Based on the technique used and the parameter range, the generated metamodel may not be efficient for global optimizations as is the case with polynomial regression [9]. Also as the number of parameter increases, the cost of creating the metamodel could also increase [3], thus negating the goal of metamodeling. In this paper, we present a design methodology incorporating Kriging metamodeling techniques and the gravitational search algorithm for the design optimization of an 180 nm phase locked loop (PLL) system for Wide Area Network (WAN) and Private Mobile Radio (PMR) applications. We show the accuracy and efficiency of Kriging for high dimensional designs (21 design parameters).

The rest of this paper is organized as follows: Section II details the novel contributions of this paper. In Section, III, a brief description of Kriging is presented. A background of the the gravitational search algorithm is given in Section IV. The experimental results for this study are presented and discussed in Section V-C. In Section VI, a brief discussion of selected related works and some comparison to this paper is presented. Finally a summary and conclusion is given in Section VII.

II. Major Contributions of this Paper

In this paper, we present a design methodology that illustrates the accuracy of a geostatistic based metamodeling technique (Kriging) for high dimensional analog mixed signal circuits. The design is is optimized with a gravitational search algorithm. The ability of Kriging to accurately predict performance response for high dimensional designs is explored. Kriging prediction techniques are based on weighting schemes calculated with the aid of correlated effects of parameters and thus are able to capture precise models reflecting the inter-dependence of parameters even at high dimensional designs. The recently proposed GSA algorithm with explorative and exploitative features based on gravitational laws is used for optimization of the design. A PLL circuit for WAN and PMR applications is presented as a demonstration of the proposed methodology.

A **summary of contributions** of this paper is as follows:

1) Exploring Kriging metamodeling for high dimensional analog mixed signal circuits using a PLL as case study.
2) Exploring the gravitational search algorithm (GSA) features for metamodel design optimization.

3) A design flow methodology for 180 nm PLL incorporating Kriging with GSA optimization techniques.

III. Ordinary Kriging Metamodel

A. Kriging Background

Kriging was originally used in geostatistics and has now been explored in various fields [10], [11], [12]. Kriging as a metamodeling technique was proposed in [13] as a combination of a polynomial model with a stochastic approach to compensate for the deterministic nature of computer experiments. A Kriging model is of the form:

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

$y(\mathbf{x_0})$, is a stochastic function that predicts the response of the design at 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 (weights) to be determined and $z(\mathbf{x})$ is a stochastic process with zero mean and based on a spatial correlation function. The weights λ_j are a function of the correlation between the set of sampled data points to be used for prediction and the response points to be predicted. Hence, the weighting average of each predicted response point is unique. The correlation function, usually called the *variogram*, is expressed as follows:

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

The autocorrelation of the design points to predicted is characterized by the covariance function [14]. The weights are chosen so that the Kriging variance is minimized [15], [10]. Ordinary Kriging method, which assumes a constant mean in the local domain of the predicted point, is explored in this paper. To achieve this condition, the weights are chosen such as to minimize the Kriging variance with the unbiasedness constraint $E(\widehat{Z}(x) - Z(x)) = 0$. Thus λ_j are chosen to satisfy $\sum_{j=1}^{n} \lambda_j = 1$. Assuming there are n sampled points, of variable x, to predict a new point $y(x_0)$, the weights λ are estimated by the following:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{pmatrix} = \Gamma^{-1} \begin{pmatrix} \gamma(x_1, x_0) \\ \vdots \\ \gamma(x_n, x_0) \\ 1 \end{pmatrix}, \tag{3}$$

Γ is the covariance matrix of the observed points and is given by:

$$\Gamma = \begin{pmatrix} \gamma(x_1, x_1) & \cdots & \gamma(x_1, x_n) & 1 \\ \vdots & \ddots & \vdots & 1 \\ \gamma(x_n, x_1) & \cdots & \gamma(x_n, x_n) & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \tag{4}$$

where

$$\gamma(x_1, x_2) = E\left(|z(x_1) - z(x_2)|^2\right). \tag{5}$$

The variograms are usually modeled on some empirical correlation functions which include spherical, linear, exponential and gaussian models. The spherical model is expressed as follows:

$$\gamma(h) = C_0 + C\left(\frac{3h}{2a} - \frac{1}{2}\left(\frac{h}{a}\right)^3\right) \text{ for } 0 < h \leq a, \tag{6}$$

where C_0, C and a are shape parameters.

For this paper, where we seek to model the power consumption of the PLL, the generated metamodel is expressed as:

$$\widehat{P_{PLL}}(\mathbf{Wn_0}) = \sum_{j=1}^{L} \lambda_j B_j(\mathbf{wn}) + z(\mathbf{wn}), \tag{7}$$

where $\widehat{P_{PLL}}(\mathbf{Wn_0})$ is the predicted power consumption at design point $\mathbf{Wn_0}$. The metamodels are generated with the aid of the MATLAB toolbox mGstat [16]. It takes in as input n design points $\mathbf{Wn_0}$ to be predicted, sample design points $\mathbf{wn}$, and a variogram model and outputs the predicted response for the corresponding prediction points.

B. Data Sampling and Metamodel Verification

Latin Hypercube Sampling (LHS) was used in this work as it is proven to achieve smaller variance than random sampling techniques [17]. A comparison of sampling techniques presented in [1], shows that sample point generation using LHS designs over random sampling points results in more accurate models. The design response at the sample points is produced from SPICE simulations. These responses are fed into the Kriging metamodel generator along with the design points to be predicted. The performance of the metamodel should be verified for accuracy to ensure design validation. This can be done using statistical metrics such as the Root Mean Square (RMSE) and the correlation coefficient R^2. A lower value for RMSE and a higher value of R^2 indicate a more accurate model.

IV. Gravitational Search Algorithm

The gravitational search algorithm (GSA) used for optimization was introduced in [18]. The GSA is based on the interaction of masses with respect to the phenomenon of gravity and the laws of motion. The basic theory of the GSA algorithm is that search agents characterized in the form of mass objects, search and explore the design space by moving according to the laws of motion. The search agents interact with each other converging on a solution by attracting each other. The agents with better solutions acquire a heavier mass and thus attract other search agents towards them. The general flow of the GSA algorithm is shown in Fig. 1.

Assuming a system with N search agents, the location (design point) of the i-th agent can be expressed as follows:

$$X_i = (x_i^1, x_i^2, \ldots, x_i^d, \ldots, x_i^n) \text{ for } i = 1, 2, \ldots, N, \tag{8}$$

where x_i^d, presents the position of the i-th search agent in the d-th dimension, and n is the number of dimensions. Each search agent is initially associated with an arbitrary mass. At the start of the flow, random search agents are evaluated and the best and worst solutions are stored. A gravitational constant

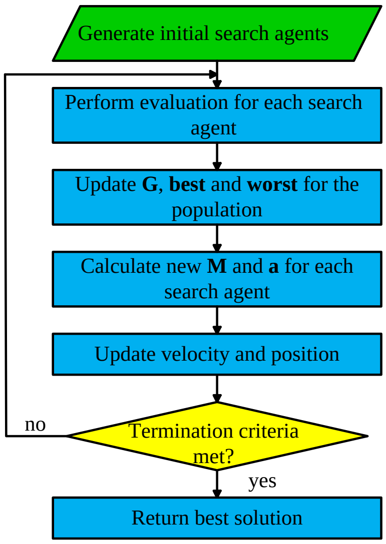

Fig. 1. Flow for a typical gravitational search algorithm.

$G(t) = G(G_o, t)$ is also updated, with G_o being an initial value.

After the best and worst solutions have been updated, the mass of the search agents M_i are also updated using the following equations:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \quad (9)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}, \quad (10)$$

where $fit_i(t)$ represents the best solution found in each iteration and $M_i(t)$ is the mass of an agent in iteration t.

The attractive force and acceleration are calculated by the following equations based on force and gravity laws:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon}(x_j^d(t) - x_i^d(t)), \quad (11)$$

$$F_i^d(t) = \sum_{j=1.j\neq i}^{N} rand_j F_{ij}^d(t), \quad (12)$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}, \quad (13)$$

where $F_{ij}^d(t)$ is the attractive force on mass 'i' from mass 'j' and M_{aj} and M_{pi} are the active and passive gravitational masses of objects 'j' and 'i', respectively. R_{ij} is the Euclidean distance between the objects. $F_i^d(t)$ is the total force acting on an object, and $rand_j$ is a random number between 0 and 1 used to introduce the stochastic property of GSA.

After the masses have been updated, the search agents are moved to a different search location by updating the velocity and position with the following equations:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t), \quad (14)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \quad (15)$$

The flow repeats until a termination criterion is met, where the best solution is returned. One appealing feature of the GSA is that it is memoryless, as it does not need to store the previous best solutions. The ability of good search agents to acquire mass, slowing them down ensures they remain in areas of best solutions. A summary of the implementation of the GSA algorithm is shown in Algorithm 1.

Algorithm 1 Gravitational Search Algorithm.

1: Initialize counter: $counter \leftarrow 0$.
2: Initialize max iteration Max_{iter}
3: Initialize optimization constants gravity constant G, and velocity ν
4: Generate η random search agents (design parameter sets) $X_i = (x_i^1, x_i^2, \ldots, x_i^d, \ldots, x_i^n)$
5: Consider the objective of interest P_{PLL_i}.
6: **while** $(counter < Max_{iter})$ **do**
7:   Evaluate P_{PLL_i} for each search node using Kriging Metamodel
8:   Update best and worst solution P_{PLL_i}.
9:   Update G (gravity constant)
10:   Calculate M and a for each search agent.
11:   Update ν for each search agents
12:   Update search agents location by applying ν on M.
13:   $counter \leftarrow counter + 1$.
14: **end while**
15: **return** $best\ solution$

The input to the algorithm is an initial set of random search agents (design points to be used) and the generated Kriging metamodel. In this case, the design variables used are the transistor widths of the PLL circuit components. The design has 21 variables so each search agent is a vector set of 21 variables defined in Line 4 of the algorithm. The search agents are then evaluated for the design objective, the power consumption of the PLL (P_{PLL_i}). Steps 8-10 of the algorithm are performed, and then the new locations (design points) for the search agents are calculated using equations 14 and 15 in steps 11 and 12, respectively. The optimization constraint used is the locking time, while the termination criterion for the algorithm is the number of iterations.

V. Experimental Results

A. Overall Design Optimization Flow

The proposed design flow methodology is shown in Fig. 2. It incorporates Kriging and the GSA algorithm for the design optimization of a 180nm PLL system. The design process begins with the design (sizing of transistors) for the schematic level with respect to the input specifications. After the logical design is finished, an initial simulation is done to make sure it meets the input specifications. If the input specifications are met, the layout of the design is drawn, otherwise the transistors are resized to meet specifications. After the layout is drawn, Design Rule Check (DRC), and Layout vs Schematic (LVS) checks are performed. Subsequently a full parasitic netlist is extracted to capture the parasitic effects and thus improve the accuracy. The next step entails parameterizing the parasitic netlist with the design variables. The parameterized netlist is then sampled for data points that are used to create the metamodel. In this design LHS is applied. The generated sample points are used to create the Kriging metamodel using the mGstat toolbox on MATLAB. After the accuracy of the metamodel has been verified, a design optimization is performed on the metamodel using the GSA algorithm. After the optimization algorithm is completed, the final layout of the circuit can be drawn using the optimized parameter set obtained from the optimization phase.

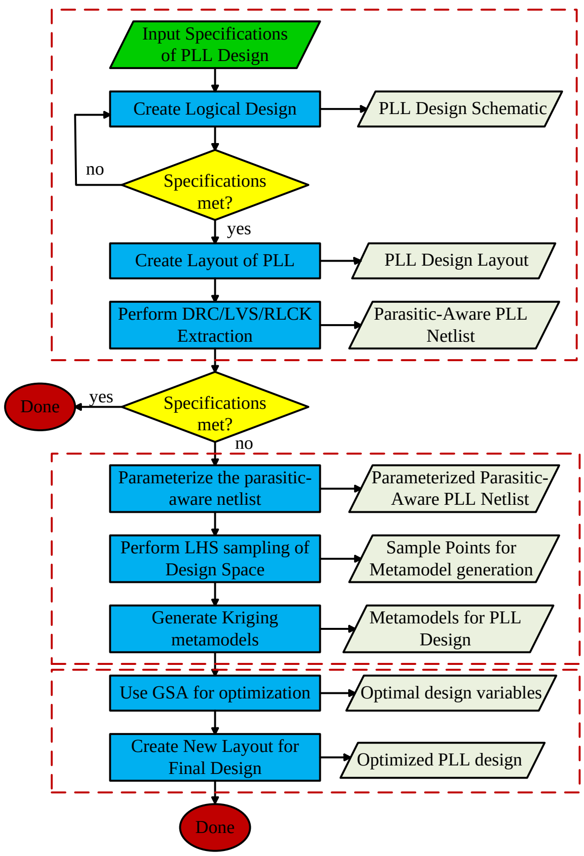

Fig. 2. Proposed overall design flow.

B. 180nm PLL Case Study

A system level diagram of a PLL is shown in Fig. 3. The details of the design are presented in [omitted for blind review]. The baseline specification for the PLL was PMR and WAN applications with a target frequency of 4 GHz. The layout of the baseline 180 nm design is shown in Fig. 4. The PLL was characterized for power consumption, frequency output and locking time. The design objective was the minimization of power consumption using the locking time as optimization cost and 21 design parameters as variables.

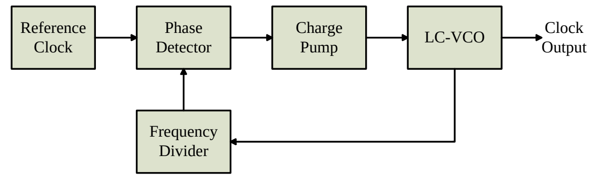

Fig. 3. System level diagram of the PLL

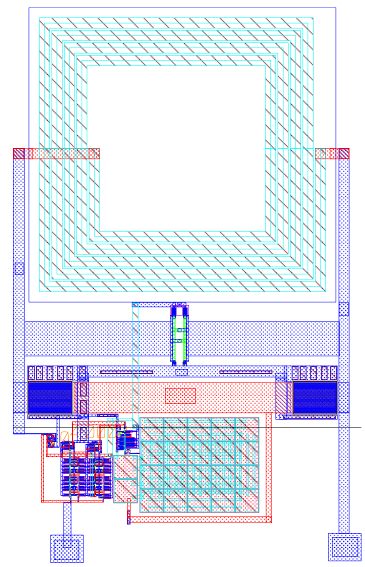

Fig. 4. 180nm layout of the PLL

C. Quantitative Results

The application of the proposed method is illustrated with the design optimization of the PLL described in Section V-B. The list of design variables is shown in Table III with their min-max ranges. A Kriging metamodel, as described in Section III was generated for modeling the power consumption. The analytical evaluation of the generated metamodel is shown in Table I. From the results, the RMSE for the power consumption is 6.46×10^{-10} which shows a very high accuracy. The R^2 value is 0.996 which signifies a very high accuracy also.

TABLE I
STATISTICAL ANALYSIS FOR ACCURACY OF KRIGING GENERATED METAMODEL FOR PLL POWER CONSUMPTION

Metric	Value
$RMSE$	6.46×10^{-10}
R^2	0.9959

With the accuracy of the Kriging metamodel verified, the optimization algorithm is performed to minimize the power consumption using the locking time as a constraint. The GSA algorithm described in Section IV is applied to the metamodel. The maximum iteration is set to 1000 with 50 initial search nodes (masses). The result of the algorithm operation is shown in Fig. 5. 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 in II shows the final results from the optimization algorithm. The power consumption is reduced by approximately 79%. The locking time is also reduced by 4 %. The range of the design variables and the final optimized values are given in Table III.

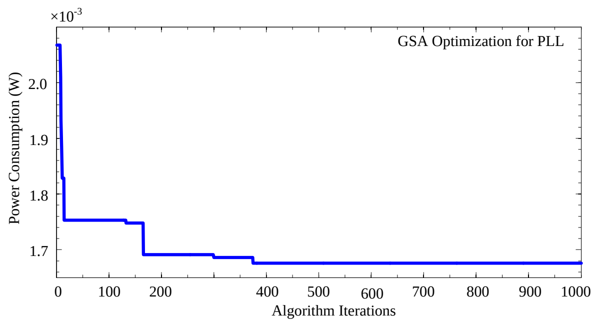

Fig. 5. Optimization Steps of the PLL

TABLE II
FINAL OPTIMIZATION RESULTS FOR THE PLL

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

TABLE III
OPTIMIZED PARAMETER VARIABLES

PLL Components	Parameter	Min (m)	Max (m)	Optimal (m)
Phase Detector	W_{pPD1}	400n	2μ	1.53μ
	W_{pPD1}	400n	2μ	0.95μ
	W_{pPD1}	400n	2μ	1.00μ
	W_{nPD1}	400n	2μ	1.16μ
	W_{nPD1}	400n	2μ	0.52μ
	W_{nPD1}	400n	2μ	1.58μ
Charge Pump	W_{nCP1}	400n	2μ	1.12μ
	W_{pCP1}	400n	2μ	1.32μ
	W_{nCP2}	2μ	4μ	2.07μ
	W_{pCP2}	4μ	4μ	4.72μ
LC-VCO	W_{nLC}	3μ	20μ	12.22μ
	W_{pLC}	6μ	40μ	14.83μ
Divider	W_{pDIV1}	400n	2μ	1.06μ
	W_{pDIV2}	400n	2μ	1.11μ
	W_{pDIV3}	400n	2μ	0.75μ
	W_{pDIV4}	400n	2μ	1.78μ
	W_{nDIV1}	400n	2μ	1.35μ
	W_{nDIV1}	400n	2μ	1.86μ
	W_{nDIV1}	400n	2μ	1.65μ
	W_{nDIV1}	400n	2μ	1.96μ
	W_{nDIV1}	400n	2μ	0.43μ

VI. RELATED PRIOR RESEARCH

In this section we discuss a selected related works in comparison to the proposed design methodology of this paper.

A. Related Research

The use of metamodeling design techniques has been well researched. As designs continue to become complex, the need to devise more accurate metamodels for these designs continues to motivate researchers. In [12], an analysis of various metamodeling techniques is presented detailing their origin and applicability to the design and optimization of analog circuits. In a comparison of selected metamodeling techniques including 2^{nd} order polynomial techniques, Kriging, genetic programming (GP), Multivariate Adaptive Regression Splines (MARS), Artificial Neural Network (ANN), and Support Vector Machine (SVM), Kriging techniques perform admirably but not as well as GP. In similar comparisons reported in [3], [19], a survey of several metamodeling techniques including Response Surface Methodology (RSM), ANN, Polynomial Regression (PR), SVM and Kriging Techniques have been presented. In [11], a comparison of PR, Radial Basis Functions (RBF), (MARS) and Kriging techniques on a variety of test cases are presented. From these results, Kriging and RBF techniques on average have the most accurate metamodels. Kriging techniques are however more suited to highly non-linear response surfaces [3].

For optimization of analog circuit designs, evolutionary algorithms which operate heuristically have generally produced very good results and hence have become very popular. Most commonly used are genetic algorithms (GA)[20] and variations from the family of swarm algorithms (including particle swarm optimization, artificial bee colonies, ant colonies and the recently introduced Gravitational Search algorithm). Most heuristic based algorithms have no scientific or empirical basis and as such do not guarantee optimal results. They are however very fast and efficient when applied on complex analog circuits producing near-optimal results.

B. Qualitative Comparison

Table IV shows a brief comparison of metamodeling based designs and optimization techniques. The comparisons are only qualitative and illustrate the applicability of the proposed design methodology. In [7], [8], [21], Kriging has been used for analog design modeling while in [22], ANN and polynomials are used. In [7], [8], only metamodels have been presented without an optimization algorithm. Both [21], [22] used optimization algorithms on the metamodels for design optimization. The accuracy of the metamodels is shown in column 4 of Table IV. The metric for analysis used is RMSE except in the case of [8] where MSE is used. In this work, Kriging is used to explore the accuracy of metamodeling for high dimensional designs especially designs with a characteristically high non-linear response. The proposed method results in higher accuracy than the compared methods. The selected methods however have been performed on different circuits and different Figures-of-Merit, hence a direct comparison only just shows perspective.

TABLE IV
RELATIVE COMPARISON OF KRIGING METAMODELING TECHNIQUES AND GRAVITATIONAL SEARCH ALGORITHM

Research	Test Circuits	Metamodeling Technique	Accuracy	Optimization Technique
You [7]	Integrated Op-Amp	Kriging	0.5658	-
Yu [8]	Ring Oscillator	Kriging	0.5325% (MSE)	-
	LC-VCO		0.5563% (MSE)	-
Okobiah [21]	Sense Amplifier	Kriging	3.2×10^{-9}	ACO
Garitselov [22]	PLL	Polynomial	0.5658	ABC
		ANN	0.5658	
This Paper	PLL	Kriging	6.46×10^{-10}	GSA

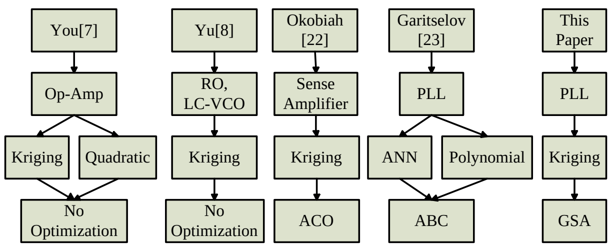

Fig. 6. Comparison of Related Research.

VII. CONCLUSION

This paper presented a design optimization flow methodology that incorporates Kriging with a GSA optimization algorithm. Kriging was used for fast and accurate metamodeling of designs of high dimensions. The efficiency of this methodology was illustrated with the design optimization of an 180 nm PLL for WAN and PMR applications. The metamodel designs were shown to have a very high accuracy with a very low RMSE. The GSA optimization algorithm was also used for optimization and was compared to similar designs. While direct comparison is not possible, the proposed methodology performed comparably to related works. It also speeds up the optimization design from the traditional design flow methodology. The simulation of the 100 sample points used took approximately 10 hours, while the metamodel creation and the optimization process takes approximately 3.5 hours.

ACKNOWLEDGMENTS

This research is supported in part by NSF award DUE-0942629. The authors would like to acknowledge the inputs of UNT graduate Dr. Oleg Garitselov.

REFERENCES

[1] 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.

[2] 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.

[3] M. Yelten, T. Zhu, S. Koziel, P. Franzon, and M. Steer, “Demystifying surrogate modeling for circuits and systems,” *Circuits and Systems Magazine, IEEE*, vol. 12, no. 1, pp. 45 –63, quarter 2012.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] G. Yu and P. Li, “Yield-Aware Analog Integrated Circuit Optimization Using Geostatistics Motivated Performance Modeling,” in *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, Nov. 2007, pp. 464–469.

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

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

[11] R. Jin, W. Chen, and T. W. Simpson, “Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria,” AIAA, Tech. Rep., 2000.

[12] T. McConaghy and G. Gielen, “Analysis of simulation-driven numerical performance modeling techniques for application to analog circuit optimization,” in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, vol. 2, May 2005, pp. 1298–1301.

[13] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn., “Design and Analysis of Computer Experiments.” *Statistical Science,*, vol. 4, no. 4, pp. 409–423, 1989.

[14] G. Bohling, “Kriging,” Kansas Geological Survey, Tech. Rep., 2005.

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

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

[17] R. L. K.-T. Fang and A. Sudjianto, *Design and Modeling for Computer Experiments*. 6000 Broken Sound Parkway NW, Suite 300 Boca Ration, FL, 33487: Chapman and Hall/CRC, 2006, no. 4.

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

[19] D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff, “A Study on Metamodeling Techniques, Ensembles, and Multi-Surrogates in Evolutionary Computation,” in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’07. New York, NY, USA: ACM, 2007, pp. 1288–1295.

[20] T. Binder, C. Heitzinger, and S. Selberherr, “A Study on Global and Local Optimization Techniques for TCAD Analysis Tasks,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 814–822, June 2004.

[21] O. Okobiah, S. Mohanty, and E. Kougianos, “Ordinary Kriging Metamodel-Assisted Ant Colony Algorithm for Fast Analog Design Optimization,” in *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, March 2012, pp. 458–463.

[22] O. Garitselov, S. P. Mohanty, and E. Kougianos, “Fast-Accurate Non-Polynomial Metamodeling for Nano-CMOS PLL Design Optimization,” *VLSI Design, International Conference on*, vol. 0, pp. 316–321, 2012.