Ultra-Fast Variability-Aware Optimization of Mixed-Signal Designs using Bootstrapped Kriging

Saraju P. Mohanty*, Elias Kougianos†, and Venkata P. Yanambaka‡
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu) *†‡
Department of Computer Science and Engineering*‡ Department of Electrical Engineering Technology†
University of North Texas, Denton, TX 76207, USA. *†‡
Email: saraju.mohanty@unt.edu*, elias.kougianos@unt.edu†, and VenkataPrasanthYanambaka@my.unt.edu‡

***Abstract*—Analog/Mixed-Signal (AMS) circuits present significant challenges to designers with the increase of design complexity and aggressive technology scaling. Design optimization techniques that account for process variation while presenting an accurate and fast design flow which can perform design optimization in reasonable time are still lacking. As a trade-off of the accuracy and speed, this paper presents a process-variation aware design flow for ultra-fast variability-aware optimization of nano-CMOS based physical design of analog circuits. It combines Kriging bootstrapped Neural Network (KBNN) metamodels with a Particle Swarm Optimization (PSO) algorithm in the design optimization flow. The KBNN provides a trade-off between analog-quality accuracy and scalability and can be effectively used for large and complex AMS circuits while capturing correlations in process variations. The effectiveness of the design flow is demonstrated using a 180nm CMOS based PLL as a case study with 21 design parameters. The KBNN metamodel is 24× faster than NN metamodeling.**

***Index Terms*—Nano-CMOS, Process variation, Mixed-Signal Circuit, Geostatistics, Kriging, Particle swarm optimization**

I. Introduction

The design of Analog/Mixed signal (AMS) systems continues to present significant challenges, especially in design optimization. Metamodeling design techniques are used to aid the design process by reducing the design time while maintaining accuracy. There still is a need to improve the metamodels currently used to increase time efficiency and accuracy. Another issue is the effect of process variation. Metamodeling techniques can be improved to account for these effects early in the design process to ensure designs are process variation-aware. Particle Swarm Optimization (PSO) is one of many optimization techniques that have been explored to improve accuracy in deep nanometer regions [1], [2]. While PSO and Neural Network (NN) metamodels have demonstrated increased accuracy [3], certain design factors such as device parameter variations continue to pose a significant concern to circuit performance estimation.

In this paper we present a process aware design flow that is incorporated into different levels of the design process. It combines a Kriging bootstrapped NN (KBNN) metamodel with a PSO algorithm in the design optimization flow. The effectiveness of the design flow is shown using a PLL as a case study.

II. Novel Contributions

This paper presents the following *novel contributions* to the state-of-the art of analog/mixed-signal CAD: 1) Fast and accurate physical design and optimization flow incorporating process awareness in analysis, characterization and optimization of performance measures. 2) Process-variations aware accurate and scalable metamodeling using Kriging bootstrapped Neural Networks. 3) Adaptation of Particle Swarm Optimization (PSO) algorithm for nano-CMOS based process-variation aware optimization. 4) A case study exploration using a 180 nm CMOS based PLL design. It may be noted that a generic overview of Kriging metamodeling is presented in [4]. Kriging metamodel for process variation analysis is presented in [5]. The current paper presents a natural progression of our research to ultra-fast physical design optimization of large analog blocks through the use of Kriging metamodeling.

III. Related Research

Polynomial regression methods which include response surface methodology (RSM) [6], [7], [8] are one of the most common and reliable methods explored. Non-polynomial based metamodels, particularly built from Neural Network (NN) training have also been reported to surpass polynomial regression techniques [9], [10], [11], [12]. NN techniques use a learning process to continuously train weights used in approximating these models. The weight training process is critical in the development of NN models and research in exploring techniques for optimizing this process is currently active. A technique popularly used is applying optimization algorithms to optimize the weight training of NN models [3]. Use of Kriging training for the NN architecture provides a trade-off between the accuracy of Kriging and scalability of the NN method [5]. In the current paper, we propose to infuse the characteristics of Kriging based techniques by bootstrapping the sample data points which are then used for the NN training process. We believe that the bootstrapped data points will enhance the modeling of process variation effects.

IV. Process Variation Aware Ultra-Fast Design Optimization Flow for Mixed-Signal Circuits

We propose a novel design flow that integrates a Kriging bootstrapped metamodeling process with the PSO algorithm

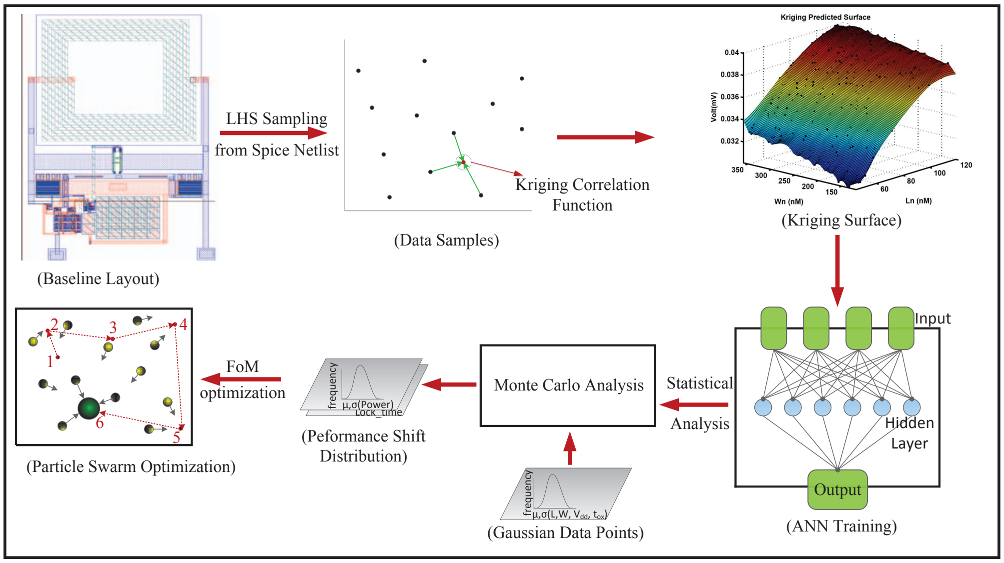

Fig. 1. Proposed high level design flow.

for the design optimization of nano-CMOS circuits as, depicted in Fig. 1. The key idea is to generate a Kriging surface using a small number of analog simulations with latin hypercube sampling (LHS) of the variables. An NN architecture is then trained to create a metamodel of the baseline circuits. Statistical analysis and optimization is performed over the metamodel instead of its SPICE netlist. The use of metamodels for design optimization iterations significantly speeds up the the design-optimization process and analog-level accuracy is maintained by the use of accurate metamodels which are generated from the parasitic-aware netlist.

The overall flow of the design process shown in Fig. 2 highlights the major phases of the design flow. The first phase labeled "A" consists of the baseline logical and physical design. In this phase, the baseline design is drawn both as a circuit schematic and the associated layout. The baseline is simulated for functional verification of the performance objectives. The functional verification also serves to characterize the circuit design objectives. The next phase involves the creation of the process variation aware metamodel of the circuit design. The first step in this phase is the identification and parameterization of the variables used to create the metamodel from the extracted parasitic netlist. Incorporating the process parameters early on in the design phase ensures a process variation aware metamodel. An LHS of the circuit from the parasitic netlist is then used by Kriging techniques to bootstrap the sample data points infusing process variation characteristics. We detail this process in Section V. The Kriging bootstrapped points are used for the NN Training. The final phase is the process aware design optimization. The optimization algorithm is used together with the created metamodel and design objectives as an input to optimize the design. The final design parameters are then used to update the physical design for an optimal design of the circuit. The process aware design optimization phase is discussed in detail in Section VI.

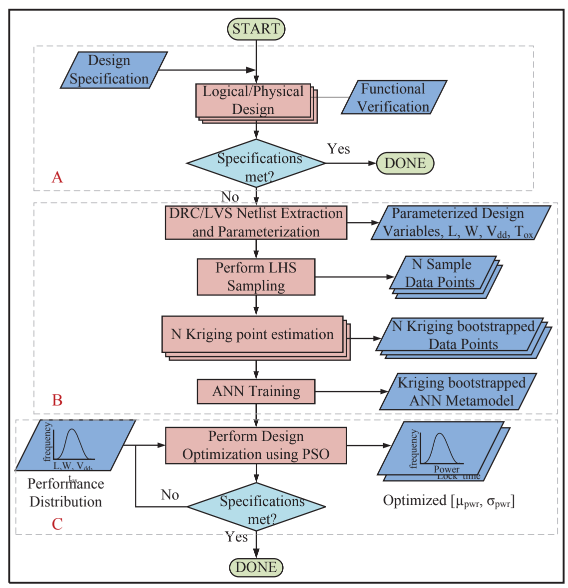

Fig. 2. Proposed design optimization flow.

V. Process-Variation Aware Kriging Bootstrapped Metamodeling

Kriging has been successfully used for metamodel generation with high accuracy [7]. The property of Kriging which makes it very appealing and lends to its high accuracy is its ability to take into account the correlation between the input parameters in performance point prediction.

A disadvantage of Kriging is that it uses a set of matrix equations in calculating the unique weights for point predictions. For large circuits and high dimensional designs, the time cost can become expensive. The use of NN on the other hand can generate metamodels which are ultra-fast and robust in accuracy. The NN models however do not efficiently model

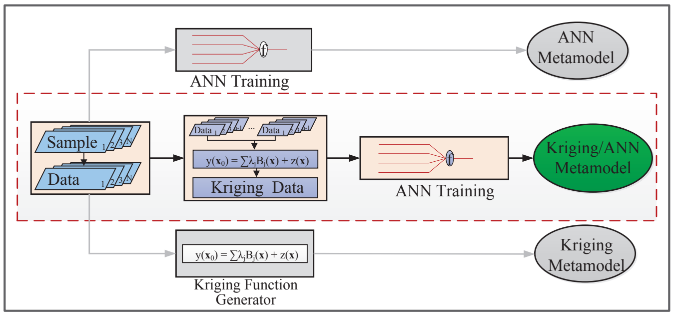

Fig. 3. Proposed Kriging based NN metamodel generation flow.

the effects of process variation. To ensure accuracy and time efficiency as well, we present a Kriging bootstrapped metamodeling technique that combines the accuracy of Kriging with the speed of NN models, as shown in Fig. 3.

The metamodel generation process takes in sample data from the extracted parasitic netlist. The sample data points are fed into the Kriging metamodel generator for resampling of the data (bootstrapping). We generate N Kriging bootstrapped data points by using $N-1$ points and the Kriging method to estimate the Nth point. N iterations of this process will generate N Kriging bootstrapped data points which are then used for the NN training.

VI. Particle Swarm Optimization (PSO) Algorithm for Process-Variation Aware Optimization

PSO is a type of evolutionary swarm intelligence algorithm for numerical optimization problems. Swarm intelligence algorithms are based on the exploitation of social or communal behavior of naturally or artificially occurring agents to collectively search for solutions. While heuristic in nature and based on social behaviors, swarm intelligence algorithms have proved to be very effective in optimization [13], [14], [15], and circuit design [16], [17].

The optimization problem implemented in this flow is to minimize the power consumption of the PLL circuit using the locking time as a design constraint. The process aware optimization of the circuit involves minimizing the mean μ and standard deviation σ of the optimal power consumption. As a an example, optimization function can be expressed as:

$$\text{Minimize}[\mu_{pwr} + 3\sigma_{pwr}], \quad (1)$$

while subjected to locking time constraint. The PSO algorithm for the PLL is shown in Algorithm 1. Fig. 4 presents an illustration of the algorithm.

VII. Experimental Results

The physical layout design of a PLL using a 180 nm CMOS technology were performed on the CADENCE Virtuoso platform, as shown in Fig. 5.

The PLL was characterized for power consumption, frequency output, locking time and jitter. The FoMs selected are Power (P_{PLL}), Frequency (F_{PLL}), Locking time (Lck_{PLL}), and Jitter (J_{PLL}). The design objective is the minimization

Algorithm 1: Particle Swarm Optimization of PLL over the Metamodels.

Input: Tuning parameter set X, Bootstrapped Kriging metamodels, Tuning parameter ranges.
Output: $X = (x_1, x_2, ..., x_n)$ parameter set with optimized statistical performance;

```
begin
    SET: N, number of particles;
    SET: Max_iteration, counter ⟵ 0;
    SET: local best l_{x_i} ⟵ current position;
    SET: global best g_{x_i} ⟵ current position;
    Initialize: weight for swarm effect ϱ;
    Initialize: velocity for swarm effect w;
    while counter < Max_iteration do
        foreach N do
            Monte Carlo analysis over metamodels with nominal x_i;
            v_i = w v_i + ϱ_p τ_p (l x_i − x_i) + ϱ_g τ_g (g x_i − x_i);
            x_i ← x_i + v_i;
            if x_i < l_{x_i} then
                l_{x_i} ← x_i;
                if l_{x_i} < g_{x_i} then
                    g_{x_i} ← l_{x_i};
```

Result: Parameter set X with minimized μ, σ;

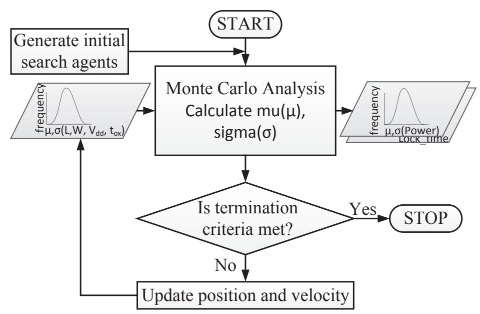

Fig. 4. Flow diagram for the PSO algorithm.

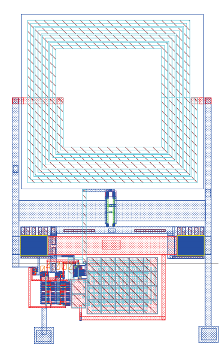

Fig. 5. Physical design of the 180 nm PLL.

of power consumption using the locking time as optimization cost and 21 parameters as design variables.

A total of 21 design parameters were used for the optimization simulation. Further statistical analysis was carried out using MATLAB. The results from the optimization simulation displayed in Table I show an improved statistical variation of

TABLE I
STATISTICAL OPTIMIZATION FOR KRIGING NEURAL NETWORK METAMODEL FOR PLL FOMS

		SPICE Netlist	Kriging-ANN Metamodel			
			Before Optimization		After Optimization	
		Value	Value	Error (%)	Value	Error (%)
Power (P_{PLL})	Mean (μ)	2.48 mW	2.40 mW	3.33	2.35 mW	2.08
	St. Dev. (σ)	0.42 mW	0.34 mW	19.05	0.39 mW	7.14
Frequency (F_{PLL})	Mean (μ)	2.66 GHz	2.51 GHz	5.64	2.78 GHz	4.51
	St. Dev. (σ)	10.95 MHz	41.93 MHz	282.92	16.92 MHz	54.52
Locking Time (Lck_{PLL})	Mean (μ)	5.51 μs	5.11μs	7.26	5.21 μs	5.44
	St. Dev. (σ)	0.72 μs	0.44 μs	38.88	0.42 μs	41.67
Jitter (J_{PLL})	Mean (μ)	16.80 ns	14.69ns	10.25	17.72ns	5.47
	St. Dev. (σ)	1.32 ps	4.50 ps	240.91	0.33ps	75

the design simulation.

From the results it is observed that the standard deviation for P_{PLL}, Lck_{PLL}, and jitter (J_{PLL}) are all minimized with the frequency (F_{PLL}) having an increased deviation. The mean power consumption was also reduced while the other FoMs were increased. This is expected since the statistical optimization started off with the design parameters for optimal performance.

VIII. Conclusion and Future Research

This paper presented a statistical optimization design flow technique that incorporates into different levels of the design process statistical awareness through the combination of Kriging and Neural Network based metamodeling with a Particle Swarm Optimization (PSO) based algorithm. The design technique was illustrated through an 180 nm Phase Locked Loop circuit. Experimental simulation results show improved standard deviations of most of the Figures-of-Merits (FoMs) considered. This demonstrates promising results in increasing performance yield in the presence of strong and correlated process variation. The research will be expanded for design optimization of a variety of analog blocks in future.

Acknowledgments

Authors would like to thank UNT graduate Dr. Oghenekarho Okobiah for his help and inputs on this paper.

References

[1] N. Jin and Y. Rahmat-Samii, "Advances in Particle Swarm Optimization for Antenna Designs: Real-Number, Binary, Single-Objective and Multiobjective Implementations," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 556–567, 2007.

[2] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling Multiple Objectives With Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[3] I. Vilović, N. Burum, and D. Milić, "Using Particle Swarm Optimization in Training Neural Network for Indoor Field Strength Prediction," in *51st International Symposium ELMAR*, 2009, pp. 275–278.

[4] O. Okobiah, S. P. Mohanty, and E. Kougianos, "Exploring Kriging for Fast and Accurate Design Optimization of Nanoscale Analog Circuits," in *Proceedings of the 13th IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2014, pp. 244–247.

[5] ——, "Kriging Bootstrapped Neural Network Training for Fast and Accurate Process Variation Analysis," in *Proceedings of the 15th IEEE International Symposium on Quality Electronic Design (ISQED)*, 2014, pp. 365–372.

[6] 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.

[7] 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.

[8] 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.

[9] L. Wang, "A Hybrid Genetic Algorithm- Neural Network Strategy for Simulation Optimization," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1329–1343, 2005.

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

[11] 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.

[12] 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.

[13] R. Poli, "Analysis of the Publications on the Applications of Particle Swarm Optimisation," *Journal of Artificial Evolution and Applications*, vol. 2008, pp. 4:1–4:10, January 2008.

[14] P. Civicioglu, "Transforming Geocentric Cartesian Coordinates to Geodetic Coordinates by Using Differential Search Algorithm," *Computers and Geosciences*, vol. 46, no. 0, pp. 229–247, 2012.

[15] M. Dorigo, M. Birattari, and T. Stutzle, "Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, 2006.

[16] S. Bennour, A. Sallem, M. Kotti, E. Gaddour, M. Fakhfakh, and M. Loulou, "Application of the PSO technique to the Optimization of CMOS Operational Transconductance Amplifiers," in *Proceedings of the 5th International Conference on Design and Technology of Integrated Systems in Nanoscale Era*, 2010, pp. 1–5.

[17] O. Garitselov, S. Mohanty, E. Kougianos, and G. Zheng, "Particle Swarm Optimization over Non-Polynomial Metamodels for Fast Process Variation Resilient Design of Nano-CMOS PLL," in *Proceedings of the great lakes symposium on VLSI*, 2012, pp. 255–258.