POLYNOMIAL-METAMODEL ASSISTED FAST POWER OPTIMIZATION OF NANO-CMOS PLL COMPONENTS

Saraju P. Mohanty[1], *Elias Kougianos*[2], *Oleg Garitselov*[3], *Javier Moreno Molina*[4]

NanoSystems Design Laboratory (http://nsdl.cse.unt.edu)[1,2,3]
Dept. of Computer Science and Engineering [1,3] and Dept. of Engineering Technology [2]
University of North Texas, Denton, TX, USA.[1,2,3]
Institute of Computer Technology, Vienna University of Technology, Vienna, Austria.[4]
Email-ID: saraju.mohanty@unt.edu[1], elias.kougianos@unt.edu[2], omg0006@unt.edu[3], and moreno@ict.tuwien.ac.at[4]

ABSTRACT

As the complexity of mixed-signal systems grows, the challenges of their design becomes exponentially more difficult. In order to mitigate this problem, this paper proposes a two stage approach that uses accurate *metamodels* and efficient algorithms for fast mixed-signal system optimization. The different components of a Phase-Locked Loop (PLL) are considered as case study. First, the metamodel creation process is presented. A simulated annealing based optimization algorithm is then introduced for power optimization of the components. It is shown that the metamodel approach speeds up the optimization phase by 2000× with very good accuracy. The power consumption of the circuits is decreased by 22% for the baseline design and is within 8% of the circuit netlist-based, but computationally expensive approach.

Index Terms— Nano-CMOS, Mixed-Signal Circuit Design, Fast Optimization, Metamodeling.

1. INTRODUCTION AND MOTIVATION

Analog/Mixed Signal (AMS) component design is a complex and time consuming process especially at the optimization and physical design stages. In addition, the presence of parasitics after the layout stage has a very dramatic effect on the output and performance thus making numerical simulation methods inefficient [1]. Hence, there is a pressing need for fast and accurate design flows and optimization approaches for mixed-signal circuit and system design exploration.

The mitigation of this problem can be achieved by one or more of the following approaches: reduction of the simulation time, reduction of the optimization time, reduction of the number of layouts needed. The research introduced in this paper proposes a new design flow for fast and accurate optimization of complex mixed-signal circuits and systems. The proposed design flow creates and uses accurate and fast metamodels of the actual circuit and performs optimization on the metamodel, not the actual circuit. Metamodels (which are models of models) are used in many different fields to simplify the design process, especially when the sampling of the design space for optimization is very costly or time consuming [2]. A metamodel is essentially a predictive mathematical algorithm for a given figure of merit (FoM) such as power, frequency, jitter, leakage, phase noise, etc [3]. Each circuit can obviously have more than one metamodel if the optimization is multiobjective. Using mathematical functions in the optimization step speeds up the process since each iteration of the calculations does not require analog simulation of the circuit or recreation of the physical design. In this paper we address the power metric to simplify and explain the metamodel creation process. Creating a metamodel is a very crucial step, since the manufacturing price of the circuit is very high and it is essential to produce the most accurate metamodel possible, given a fixed simulation time budget.

Metamodeling has been applied on IP reuse for SoC integration and microprocessor design in [4]. This approach covers a higher level design flow, is purely digital and does not create metamodels specifically for CMOS circuits. In [5], the authors propose the use of metamodels for creating an inductor for CMOS circuits. The technique that the author proposes does not use sampling techniques but rather uses mathematical formulas for the model estimation and optimization.

An order reduction technique, macromodeling is discussed in [6]. Some authors use the terms “macromodel” and “metamodel” interchangeably but they are very distinct. A macromodel is simply a reduced complexity (order) representation of the circuit but is still a netlist, necessitating the use of an analog (SPICE) simulator. On the other hand, a metamodel is a language and simulator independent model of the original model (hence the term meta). In this context, a VCO parametric metamodeling approach is given in [7]. Design space exploration approaches from high-level descriptions of analog circuits are given in [8]. The use of neural networks in the automatic synthesis of OP-AMPs is explored in [9]. RF-

specific transistor sizing with explicit parasitic estimates is given in [10]. A layout-aware modeling approach for analog synthesis is given in [11].

2. NOVEL CONTRIBUTIONS OF THIS PAPER

This paper proposes a novel design approach, presented in Fig. 1, which shows the flow up to the optimization stage. In this approach, the baseline circuit SPICE netlist with parasitics is extracted from the layout. The netlist is then parameterized in terms of the design variables, and metamodels are constructed for each FoM. If the desired optimization is a single FoM, then an algorithm is used to optimize it. Otherwise specification constraints are set by using one or more metamodels and the FoMs are then optimized for specified constraints. One of the main benefits of using metamodels is that they are reusable, as long as the FoMs and the process constraints are selected judiciously by the designer. Once metamodels are created the specification constraints can be adjusted and the optimization time using metamodels is very short in comparison to the same type of optimization done via simulations on the original model (SPICE netlist).

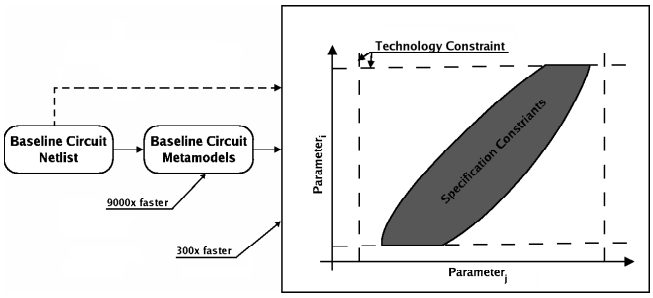

Fig. 1. The proposed metamodel-assisted multiobjective flow.

The *novel contributions of this paper* are as follows:

1. A technology-independent metamodeling design flow is presented.
2. The metamodeling approach is used to optimize different components of a PLL for power (including leakage dissipation).
3. A 22% decrease in power is achieved when using the metamodeling approach to the initial baseline design and 8% decrease in power is reached over the traditional netlist-based approach.
4. The proposed polynomial-metamodeling approach is 2000$\times$ faster compared to a netlist-based optimization.

3. GENERATION OF FAST AND ACCURATE METAMODELS FOR PLL COMPONENTS

In this section, a metamodel generation technique is presented for different AMS circuit or system components with a 180nm CMOS PLL as a case study. Due to space limitations for a more detailed discussion of the PLL components readers are referred to [12]. The layout of the LC-VCO which is the heart of a PLL is shown in Fig. 2.

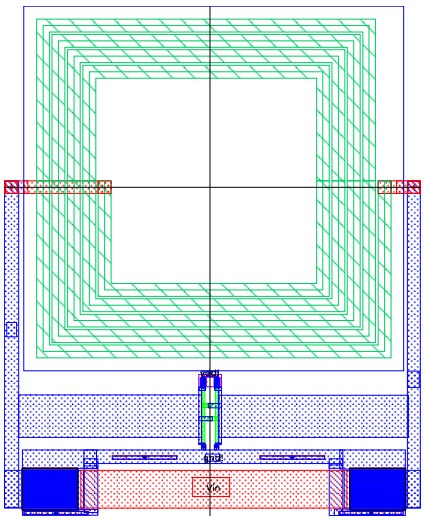

Fig. 2. Physical design of the LC-VCO for 180 nm CMOS.

The creation of metamodels is the crucial step in this design flow. Selecting the right prediction function that has low error variance compared to the actual simulated analysis is crucial. Since it is not possible to exhaustively sample the design space when a large number of degrees of freedom (design variables) are present, the accuracy of the model varies based on the form of the metamodel. To decrease the complexity of metamodel creation, this paper targets polynomial functions due to the small number of variables for each subcircuit. Polynomial functions of order 1 through 6 are considered for metamodeling. This process can be extended to other functions such as splines, artificial neural networks etc., which can be generated from the same sample data. To make this process robust the flow for metamodel creation has been described in Algorithm 1. In this algorithm, RMSE refers to the Root-Mean-Square-Error between the predicted value at a sample point and the actual result from the netlist simulation.

Algorithm 1 Metamodel Creation Algorithm

```
1:  sample = generate_sample_data().
2:  verify = generate_verify_data().
3:  centered = center(sample).
4:  for degree = 1 to 6 do
5:      sample_mm(degree)=stepwise(sample, degree).
6:      centered_mm(degree)=stepwise(centered, degree).
7:      sample_RMSE(degree)=verify(sample_mm, verify).
8:      centered_RMSE(degree)=verify(centered_mm, verify).
9:  end for
10: pick_lowest(sample_RMSE, centered_RMSE).
```

The final outcome is a multivariate polynomial function of degree p in the n design variables $\mathbf{x} = x_1, x_2, \cdots, x_n$ for the predicted response $\hat{y}$:

$$\hat{y}(\mathbf{x}) = \sum_{|\alpha| \leq p} \mathbf{c}_\alpha \mathbf{x}^\alpha. \tag{1}$$

Where the multi-index $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ takes values in $\mathbb{N}_0^n$, $|\alpha| \equiv \alpha_1 + \alpha_2 + \cdots + \alpha_n$, $\mathbf{x}^\alpha \equiv (x^{\alpha_1}, x^{\alpha_2}, \cdots, x^{\alpha_n})$ and $\mathbf{c}_\alpha \equiv (c_{\alpha_1}, c_{\alpha_2}, \cdots, c_{\alpha_n})$ are the undetermined fitting coefficients. Since $0 \leq |\alpha| \leq p$ there are a total of:

$$N(p, n) = \sum_{k=0}^{p} \binom{n+k-1}{k} \tag{2}$$

such terms. As the number of design variables (n) and/or the degree of the multinomial (p) increase, $N(p, n)$ increases exponentially.

The final metamodel degree p is selected so that it provides the least RMSE to the data points and the lowest RMSE for verification points. Since the RMSE value for fitted data points shows the error for points that are being sampled, it is not a good metric to use since the function can be possibly over fitted: it can pass through all sample points but can have a very large error between samples. Points in-between the sampled data are used to verify the accuracy of function and based on the RMSE value calculated from that step, the metamodel function degree is selected. Algorithm 1 generated the best fitted polynomial function within the given criteria. If the accuracy still does not satisfy the specifications, the number of samples has to be increased. The initial number of samples can be quite small to reduce the sampling time and can be gradually increased for higher accuracy.

3.1. Data Sampling

The starting step for creating the metamodel is to perform simulations for a sample of the design space, determined by the available simulation budget. In this paper all metamodels have been generated using Middle Latin Hypercube Sampling (MLHS) [13] which divides the design space into Latin Hypercubes [14] and then samples the middle point. Our previous research has shown that data is distributed more evenly, even though MLHS does not sample the edges where the parameters have minimum and maximum values, but due to the polynomial function behavior it is compensated by the slope of the function as long as the behavior of the sample data is not sporadic at the edges.All metamodels have been generated by 1000 MLHS samples and 100 MLHS samples were used for their verification purposes.

A matrix L is generated to code each row as a term in the polynomial power for each parameter except when the value is 0 as it is implied that the term is not present. As an example, L is the matrix for second order code for 2 parameters:

$$L = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 0 & 1 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} \tag{3}$$

Lets assume that $X \equiv [x_1, x_2]$ where x_1, x_2 are design parameters. Then the design matrix template (DMT) for X using L code becomes the following:

$$DMT(X, L) = [0, x_1, {x_1}^2, x_2, {x_2}^2, x_1 x_2]. \tag{4}$$

The design matrix (DM) is created by using MLHS of the x_1 and x_2 parameters, resulting in a matrix of the following:

$$DM^T(X, L) = \begin{bmatrix} 0 & \cdots & 0 \\ \mathrm{mlhs}(x_1)_1 & \cdots & \mathrm{mlhs}(x_1)_n \\ \mathrm{mlhs}(x_1)_1^2 & \cdots & \mathrm{mlhs}(x_1)_n^2 \\ \mathrm{mlhs}(x_2)_1 & \cdots & \mathrm{mlhs}(x_2)_n \\ \mathrm{mlhs}(x_2)_1^2 & \cdots & \mathrm{mlhs}(x_2)_n^2 \\ \mathrm{mlhs}(x_1)_1 \mathrm{mlhs}(x_2)_1 & \cdots & \mathrm{mlhs}(x_1)_n \mathrm{mlhs}(x_2)_n \end{bmatrix} \tag{5}$$

The corresponding Y matrix is then sampled from the original design and is based on the parameters x_1 and x_2 for each row in the DM matrix.

3.2. Data Centering

Mean data centering is used to alleviate abrupt changes in the data. It centers and scales each column of X before fitting. Z_{score} for each parameter in the circuit is calculated by using the following expression:

$$Z_{score} = \left(\frac{x_i - \mu_i}{\sigma_i} \right), \tag{6}$$

where x_i is the i-th design parameter, μ_i is the mean of the data points for that parameter, and σ_i is its standard deviation. This approach is mostly used when the parameter range is very high and the data is not distributed evenly.

3.3. Stepwise Regression

Stepwise regression [15] is conducted on the sample data to create the polynomial of first to sixth order. The resulting polynomial may not include all terms since stepwise regression iteratively removes coefficients that are not statistically significant, resulting in metamodels with fewer coefficients without losing accuracy.

This method may build different models from the same set of potential terms depending on the terms included in the initial model and the order in which terms are moved in and out. The function stepwise(), which is embedded in MATLAB, tests different initial models and outputs the coefficients for the best model for that order.

3.4. Verification of Metamodel

It has been mentioned above that the RMSE value that is calculated from metamodel fitting for sampling points is not a good metric for the metamodel fit. Extra verification points are required to ensure that the fit is also good at other points than the ones being sampled. Verification samples are checked to make sure that the data points are not the same as the sampling points. The lowest RMSE value for different metamodel functions is then selected as the best function.

A total of 12 metamodels, 6 metamodels using the sample points with data centering and 6 metamodels without using data centering, are generated and then compared. The lowest RMSE verification value is used to pick the most accurate metamodel. Table 1 compares the various metamodels.

4. OPTIMIZATION ALGORITHM

Simulated annealing optimization is an extension of the Monte Carlo algorithm and simulates the annealing process which is used in metallurgy. This paper explored the algorithm for circuit optimization as the number of parameters can be quite large. By nature, the algorithm has a random component and two successive runs can produce different results. The steps of simulated annealing based PLL component optimization are presented in Algorithm 2.

Algorithm 2 Proposed Simulated Annealing based Optimization for PLL Components.

Initialize iteration counter: $Counter = 0$.
Initialize first feasible solution $S_i = mid(P_n)$ for each parameter P.
Determine initial $Power_i$ for the solution S_i.
Initialize temperature T and cooling rate α_T.
while ($\Delta_{Power}! = 0$) **do**
  $Counter$ =Maximum number of iterations.
  while ($Counter > 0$) **do**
    Generate random transition from S_i to S_i^*.
    if (S_i^* is acceptable solution) **then**
      $result = S_i^*$.
      break both while loops.
    else
      $\Delta_{Power} = Power_S - Power_i^*$
      if $\left(\Delta_{Power} < 0 \text{ random}(0,1) < e^{\left(\frac{\Delta Power}{T}\right)} \right)$ **then**
        $S \leftarrow S_i^*$.
      end if
    end if
    $Counter = Counter - 1$.
  end while
  $T = \alpha_T \times T$.
end while
return $result$

Parameter S calculations are based on the FoM function which takes frequency into constraint and returns circuit power value if the design is within the frequency specifications. If the design is not within the frequency specifications the returned value is high, making the algorithm ignore the parameter values of that step.

5. EXPERIMENTAL RESULTS

The algorithm implementations and simulations were performed in an integrated environment of Cadence and MATLAB. A Graphic User Interface (GUI) is developed for easy use of the design flow, as shown in Fig. 3.

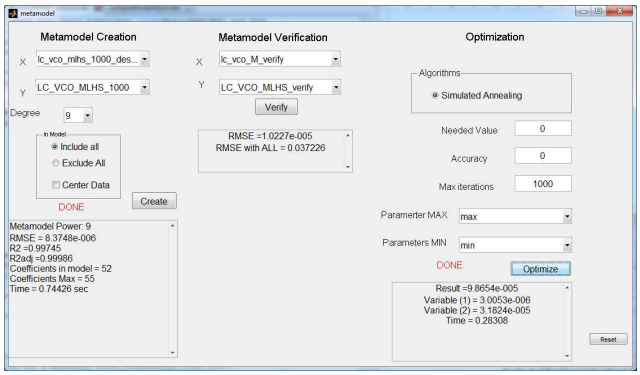

Fig. 3. The Graphic User Interface (GUI) of the metamodeling toolbox developed as a part of the on-going research.

The results for LC-VCO multiobjective optimization which has been performed using two metamodels, one for power and one for frequency, are shown in Table 2. The frequency metamodel is used as constraint in the optimization process and is set to within 1% of 2.25 GHz. The simulated annealing algorithm is used to minimize the power with the given constraint. As a result, the chosen metamodels are of order 4 and 5 for power and frequency respectfully. The accuracy of the metamodel prediction values are verified using SPICE simulations and show that the metamodel error to actual simulation at the near optimal point is 0.047 mW for power and 0.02 GHz for frequency. It is also interesting to note that the verification simulation achieved closer results to the target values with lower power and frequency being directly on target.

The results of optimization conducted with the metamodel and the circuit itself are shown in Table 3. The metamodel generation time is generally in the order of seconds and depends on the maximum number of coefficients that can be present in the model, since stepwise regression process is very much dependent on the number of coefficients. Higher order polynomials add more calculation time, but still the longest 6th order polynomial with 9 parameters took roughly 2 minutes to create. The optimization time is also in seconds in comparison to simulation optimization.

Table 1. Metamodel polynomial degree comparison for PLL component circuits

PLL Component	Degree	R^2	R^2 Adjusted	Coefficients Total	Coefficients in Model	RMSE Fit (W)	RMSE Verification (W)
Phase Detector	1	0.8610	0.9997	7	3	2.1×10^{-10}	6.6×10^{-10}
	2	0.9603	0.9993	28	18	1.1×10^{-10}	7.1×10^{-10}
	3	0.9663	0.9989	84	34	1.0×10^{-10}	7.0×10^{-10}
	4	0.9748	0.9974	210	94	9.2×10^{-11}	7.1×10^{-10}
	5	0.9926	0.9955	462	206	8.2×10^{-11}	7.1×10^{-10}
	6	0.9968	0.9938	924	662	5.4×10^{-11}	9.7×10^{-10}
Charge Pump (with data centering)	1	0.9905	1	5	5	2.3×10^{-6}	8.4×10^{-4}
	2	0.9968	1	15	14	1.3×10^{-6}	8.4×10^{-4}
	3	0.9987	1	35	31	8.6×10^{-7}	8.5×10^{-4}
	4	0.9995	1	70	54	5.3×10^{-7}	8.4×10^{-4}
	5	0.9999	1	126	87	2.8×10^{-7}	8.2×10^{-4}
	6	1	1	210	151	1.4×10^{-7}	8.3×10^{-4}
LC-VCO	1	0.9237	0.9985	3	3	4.5×10^{-5}	4.1×10^{-5}
	2	0.9769	0.9999	6	6	2.5×10^{-5}	2.1×10^{-5}
	3	0.9878	0.9999	11	11	1.8×10^{-5}	1.7×10^{-5}
	4	0.9927	0.9999	16	16	1.4×10^{-5}	1.4×10^{-5}
	5	0.9942	0.9999	21	20	1.2×10^{-5}	1.2×10^{-5}
	6	0.9946	0.9999	28	20	1.2×10^{-5}	1.1×10^{-5}
Divider	1	0.2954	0.9943	10	9	7.6×10^{-6}	7.7×10^{-6}
	2	0.3503	0.9901	55	16	7.3×10^{-6}	5.9×10^{-6}
	3	0.4504	0.9618	220	66	6.9×10^{-6}	5.8×10^{-6}
	4	0.6787	0.8828	715	268	6.0×10^{-6}	9.1×10^{-6}
	5	1	1	2002	999	0	1.1×10^{-4}

Table 2. Multiobjective optimization using power and frequency metamodels for LC-VCO with 1% accuracy for 2.25 GHz.

	Power	Frequency	Error to Verification	Error to Target
Prediction	0.153 mW	2.23 Ghz	0.047 mW	0.02 Ghz
Verification	0.110 mW	2.25 Ghz	–	0 Ghz
Polynomial order	4	5	–	–

The final results for power consumption for all PLL components are shown in Table 4. The initial power has been measured at the lowest minimal values for each parameter considered in all circuits. The order of each metamodel varies and was selected from having the lowest RMSE values. It is easy to see that on average the metamodeling approach has reached better results than the simulation approach, even though simulation optimization provides better results for the divider.

6. CONCLUSION AND FUTURE RESEARCH

From this research we can conclude that the second order polynomial, which forms the basis for Response Surface Methodologies (RSM), does not always accurately capture the complexity of the response in a multidimensional design space. For the PLL circuit, higher order polynomials provide better results. The metamodeling design flow lowers the design optimization phase by roughly 2000× compared to the simulation based optimization approach if used for IP reuse. The metamodel generation is the slowest step of the proposed design process and provides speedup that is roughly equal to (number of samples)/(netlist optimization sampling). One can argue that the metamodeling approach sampling and model generation stage is time consuming, but considering that the metamodels are reusable and can be used for circuit verification in addition to optimization, this approach becomes very attractive for AMS SoC design and verification.

7. REFERENCES

[1] J. Park, K. Choi, and D. J. Allstot, "Parasitic-Aware Design and Optimization of a Fully Integrated CMOS

Table 3. Power Metamodel versus netlist circuit optimization comparison

Circuit (parameters)	Metamodel			Netlist	
	Order	Prediction (W)	Time	(W)	Time
Phase Detector (6)	1	7.35×10^{-9}	0.07s	6.81×10^{-9}	$\approx$ 19 min
Charge Pump (4) (without data centering)	2	3.01×10^{-5}	0.11s	3.11×10^{-5}	$\approx$ 18 min
LC-VCO (2)	5	8.84×10^{-5}	1.22s	9.54×10^{-5}	$\approx$ 18 min
Divider (1)	3	1.20×10^{-5}	1.05s	$3.43 \times 10^{--5}$	$\approx$ 8 min

Table 4. Power optimization using metamodels for different components.

Components	Phase Detector	Charge Pump/Filter	LC-VCO	Divider	Total Power	Total Time
Initial Power	9.31 nW	21.84 μW	75.5 μW	60.5 μW	157.9 μW	–
Metamodel Optimization	6.80 nW	3.10 μW	71.4 μW	48.1 μW	122.6 μW	7.48 sec
Simulation Optimization	6.87 nW	3.11 μW	95.4 μW	34.3 μW	132.8 μW	81 min

Wideband Amplifier," in *Proceedings of the 8th Asia South Pacific Design Automation Conference*, 2003, pp. 904–907.

[2] K. T. Fan, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*, Chapman and Hall/CRC, Boca Raton, FL, 2006.

[3] O. Garitselov, S. P. Mohanty, and E. Kougianos, "A Comparative Study of Metamodels for Fast and Accurate Simulation of Nano-CMOS Circuits," *IEEE Transactions on Semiconductor Manufacturing*, vol. 25, no. 1, pp. 26–36, February 2012.

[4] D. A. Mathaikutty and S. Shukla, *Metamodeling Driven IP Reuse for System-on-chip Integration and Microprocessor Design*, Artech House, 2007.

[5] A. Lamecki, L. Balewski, and M. Mrozowski, "Towards Automated Full-Wave Design of Microwave Circuits," in *17th International Conf. on Microwaves, Radar and Wireless Communications*, 2008, pp. 1–2.

[6] A. Agarwal, G. Wolfe, and R. Vemuri, "Accuracy Driven Performance Macromodeling of Feasible Regions During Synthesis of Analog Circuits," in *Proceedings of 15th ACM Great Lakes Symposium on VLSI*, 2005, pp. 482–487.

[7] W. Dong, Z. Feng, and P. Li, "Efficient vco phase macromodel generation considering statistical parametric variations," in *Proc. IEEE/ACM international conference on Computer-aided Design*, 2007, pp. 874–878.

[8] A. Doboli and R. Vemuri, "Exploration-based High-level Synthesis of Linear Analog Systems Operating at low/medium Frequencies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 11, pp. 1556–1568, 2003.

[9] G. Wolfe and R. Vemuri, "Extraction and Use of Neural Network Models in Automated Synthesis of Operational Amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198–212, 2003.

[10] A. Agarwal and R. Vemuri, "Layout-aware RF Circuit Synthesis Driven by Worst Case Parasitic Corners," in *2005 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 2005.

[11] A. Pradhan and R. Vemuri, "A Layout-aware Analog Synthesis Procedure Inclusive of Dynamic Module Geometry Selection," in *Proceedings of the 18th ACM Great Lakes symposium on VLSI*, 2008, pp. 159–162.

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

[13] F. H. Lesh, "Multi-Dimensional Least-Squares Polynomial Curve Fitting," *Commun. ACM*, vol. 2, pp. 29–30, September 1959.

[14] Bo. Tang, "Orthogonal Array-Based Latin Hypercubes," *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1392–1397, December 1993.

[15] A. T. McCray, J. McNames, and D. Abercrombie, "Stepwise Regression for Identifying Sources of Variation in a Semiconductor Manufacturing Process," in *IEEE Conference and Workshop on Advanced Semiconductor Manufacturing*, May 2004, pp. 448 – 452.