

Nano-CMOS Mixed-Signal Circuit Metamodeling Techniques: A Comparative Study

Oleg Garitselov¹, Saraju P. Mohanty², Elias Kougianos³, and Priyadarsan Patra⁴

NanoSystem Design Laboratory (NSDL, <http://nsdl.cse.unt.edu>), University of North Texas, Denton, TX 76203, USA.^{1,2,3}

Intel Architecture Group, Intel Corporation, USA.⁴

E-mail ID: omg0006@unt.edu¹, saraju.mohanty@unt.edu², elias@unt.edu³, and priyadarsan.patra@intel.com⁴

Abstract—Fast design space exploration of complex nano-CMOS mixed-signal circuits is an important problem. In this paper, a design process flow that uses metamodels is introduced. In this flow the most important task is the sampling of the design space. In this paper, different sampling techniques for producing an accurate metamodel are investigated to minimize the number of samples required by using a nano-CMOS ring oscillator (RO) as an example. Through SPICE simulations, it is shown that the parasitics have a drastic effect on performance metrics, such as the frequency of oscillation. Alternative sampling techniques, both random, such as Monte Carlo (MC), and uniform, such as Latin Hypercube Sampling (LHS), and Design of Experiments (DOE), are considered as and compared for speed and accuracy. Due to the time constraints of the circuit design process, this paper can be used as a guideline for which sampling technique will produce the most accurate result to minimize the design time. All experimental results are presented for a 45 nm technology.

Keywords—Nanoscale CMOS, Mixed-Signal Circuits, Metamodeling, Statistical Sampling, Circuit Simulation

I. INTRODUCTION AND MOTIVATION

The design cycle for typical analog circuits is very long since accurate, circuit-level simulation is very CPU intensive. This situation is further aggravated when such circuits are designed using nano-CMOS technology where the transistors are modeled using 100's of parameters. It is also very difficult to accurately predict the performance of analog circuits in high frequency applications due to the many parasitic effects [1], [2]. To meet the desired design specifications, the original design is iteratively adjusted by attempting different values of the design variables. A large number of design variables results in an enormous amount of different possibilities for alternative design tradeoffs. Exhaustive search of the design space to obtain an optimal solution is quite time consuming and, for typical complexity circuits, it is an impossible task to exhaustively search the space. An alternative of exhaustive search in the design space of the *actual circuit* is performing a fast search using a *metamodel*. A typical circuit design already consists of a hierarchy of models at different levels of abstraction. At the lowest level, BSIM4 models of transistors are used to create SPICE netlists of small design units which are further assembled into subsystems and finally, into complete systems. The *model* of the circuit is thus a very large,

hierarchical SPICE netlist and typically includes parasitics which are extracted from the final layout. Optimal values for the design specifications can be obtained by creating an accurate metamodel for that design by sampling data from the simulated circuit. The metamodel is a mathematical model that acts as a substitute (surrogate) for the original model. Since it is a very expensive process to manufacture the circuit it is essential to create the closest possible result to generate a circuit that can be manufactured with the lowest tolerance of error. SPICE simulation tools are used to simulate circuits in different design steps. Very complex circuits can take days if not weeks to simulate. Hence, the amount of simulation iterations need to be as low as possible to minimize the time of the design process. The use of metamodels introduces a simpler way of understanding the behavior of the circuit and an easier model to conduct simulations and apply optimization techniques. To generate an accurate metamodel a designer needs to take samples from the simulated circuit's response only a limited number of times, sufficient to construct the metamodel. This paper provides guidelines on which sampling technique works best, using a sample mixed-signal nano-CMOS circuits with full parasitics as well as estimates on the number of samples that produce the desired result with the least amount of simulations.

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

- 1) This paper proposes technology independent metamodeling sampling techniques.
- 2) Five distinct random and uniform metamodeling sampling techniques are introduced. They include Monte Carlo (MC), Latin Hypercube Sampling (LHS), Middle Latin Hypercube Sampling (MLHS), and Design of Experiments (DOE), and are applied to nano-CMOS.
- 3) The use of these sampling techniques in metamodeling is demonstrated for a 45 nm CMOS ring oscillator. The oscillator is characterized for frequency, power and jitter. The full RCLK (resistance, capacitance, and self and mutual inductance) parasitic extraction is performed and compared to the schematic of the oscillator. The metamodels are generated on the parasitic netlist.

The rest of the paper is organized as follows: section II briefly discusses previous works relevant to metamodeling. The 45 nm CMOS based ring oscillator that is used in this research to provide sampling data is discussed in Sec-

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

tion III. Section IV shows the proposed design flow. Section V introduces five different sampling techniques and they are compared in Section V. The paper is concluded with directions for future research in Section VI.

II. RELATED PRIOR RESEARCH

The general theory of metamodeling, associated sampling techniques, and computer experiments, as applied in various fields of science and engineering can be found in [3] and [4], but these works do not address nano-CMOS technologies. In [5] the author proposes the use of metamodels for modeling inductors in CMOS circuits. The technique that the author proposes does not use sampling techniques but rather uses mathematical formulas for the model estimation and optimization. A technique for the automated creation of surrogate multivariate mathematical models by using CAD-Model Construction for microwave components is developed and tested in [6]. This technique is compatible with SPICE. Considerable work on metamodeling and surrogate techniques has been done for digital VLSI but not analog or mixed-signal circuits. In [7] support vector machine (SVM)-based machine learning is proposed as a surrogate for expensive circuit-level simulation. A statistical wire-length estimation approach using surrogate modeling is proposed in [8]. The application of statistical techniques in timing analysis of critical paths using small-scale Monte-Carlo is presented in [9].

The design and characterization of ring oscillators covering jitter, power and frequency can be found in [10], [11] and [12]. These are design research which do not deal with metamodeling rather perform design cycle on the actual circuit which are time consuming.

III. DESIGN AND CHARACTERIZATION OF THE CASE STUDY CIRCUIT: A 45 NM CMOS RING OSCILLATOR

A Ring Oscillator (RO) consists of an odd amount of inverters connected in series with the output fed back to the input to create oscillations which are derived from the propagation delay of each inverter. Ring oscillators are useful in die and new technology testing and are commonly to find the delay times of logic gates. Figure 1 shows the schematic diagram of a three inverter RO. For a given technology node, the designer can adjust the widths of the NMOS and PMOS to obtain the desired frequency. For this simple circuit, the design space is spanned by the two widths.

A. Logical Design of the 3 Inverter Ring Oscillator

The design variable chosen for this design are: length of transistors is $L_n = L_p = 45\text{nm}$, width of NMOS $W_n = 4L = 120\text{nm}$ and width of PMOS $W_p = 8L = 240\text{nm}$ at a nominal operating voltage $V_{dd} = 1\text{V}$, as shown in figure 1. All simulations will also assume that the ambient temperature of 27 degrees Celsius is constant and will not change, since it can affect the output dramatically. Self-heating effects are not taken into account in this work. Assuming equal fall and rise

times of each inverter, the frequency of oscillation for the RO is calculated by the following expression [13]:

$$f = \left(\frac{1}{2Nt_p} \right), \quad (1)$$

where N is the (odd) number of inverters and t_p is the propagation delay of each inverter.

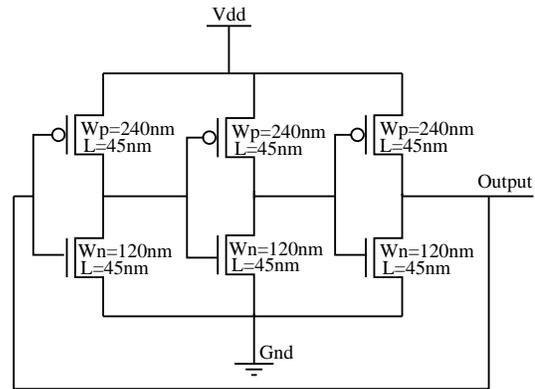


Fig. 1. Transistor-level schematic of the ring oscillator.

B. Physical Design of the 3 Inverter Ring Oscillator

At nano-CMOS technologies, where the frequency is in the GHz range, parasitics have a dramatic effect on performance. It is difficult to estimate these parasitics without actually performing the layout, which is shown in figure III-B. RCLK parasitic extraction on this layout provide the full SPICE netlist. The physical design which involves tedious manual work, by using *metamodeling* will only need to be done twice, i.e., once for the initial design and one final time after obtaining the optimized data from the metamodel.

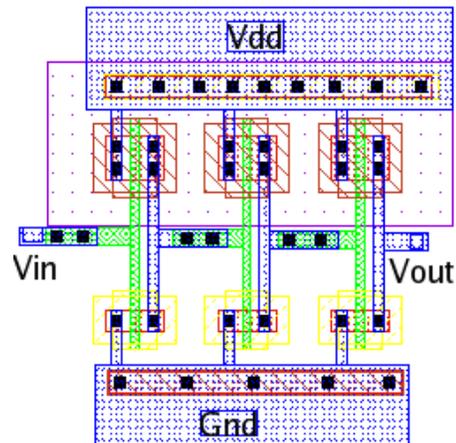


Fig. 2. RO layout for a 45nm CMOS technology.

The parasitics from this layout result in a dramatic decrease in frequency versus the regular schematic simulations. The presence of parasitics for this simple circuit also increases the simulation run time a factor of 3. For a present day complex circuit with thousands of transistors, the simulation

time will be performed in days, if not weeks, depending on the complexity of the circuit. Table I compares the number of components between the regular schematic and the parasitic netlist.

TABLE I
NUMBER OF COMPONENTS IN THE RING OSCILLATOR CIRCUIT

Simulation	Transistors	Capacitors	Resistors	Total
Without parasitics	6	0	0	6
With parasitics	6	82	19	107

The creation of a different layout by adjusting the widths of the CMOS components to $W_n = 360\text{nm}$ and $W_p = 720\text{nm}$ did not show a big effect of width on the output frequency. On the other hand, the change between the schematic and parasitic outputs is of the order of 40%. Table II compares the simulation results with and without parasitics.

TABLE II
SIMULATION COMPARISON

Extraction	Power	Frequency
Schematic	27.17 μW	16.21 GHz
120nm-240nm Parasitic	26.96 μW	9.88 GHz

C. Simulation and Characterization of the Ring Oscillator

Table II shows the results for both simulations from the original schematic and from the RLCK extracted netlist runs. The frequency has dropped dramatically by approx. 40% due to the presence of the parasitics. It is observed that the total power consumption has not been altered, and only changed by merely 1%. This data shows that the extraction of parasitics is necessary to calculate the desired output such as the frequency for this circuit. The eye diagram in figure 3 shows that the jitter effect for a 100 ns period is negligible, even when full parasitics are taken into account.



Fig. 3. Eye diagram of parasitic netlist.

IV. METAMODEL DESIGN FLOW

The proposed design flow is shown in figure 4. Once the logical design is done and meets the required specifications, an initial physical design is implemented. The physical design is then subjected to Design Rule Check (DRC), Layout vs. Schematic (LVS) and parasitic (RCLK) extraction. If the specifications are not met, the parasitic Netlist is then created with the design variables used as parameters. This netlist is then used by our automated process to create a metamodel by applying a sampling technique described in this paper. Once the metamodel is created it can be optimized to find the parameters for the variables that were chosen before. The final physical design is created by using the parameters from the optimization. By using this approach, the designers only need to create the physical design two times, one before the creation of the metamodel and its optimization (initial design) and one after the optimization of the metamodel (final design). In this approach the production of a very accurate metamodel is essential for this algorithm to work properly. Hence this paper covers metamodel sampling techniques and their accuracy.

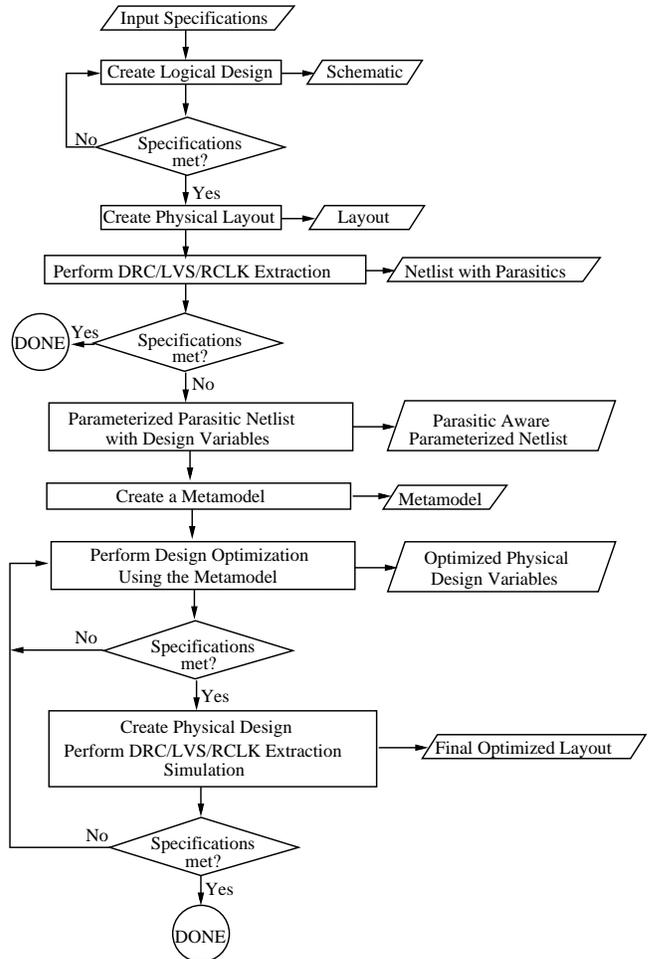


Fig. 4. Metamodel-based design flow.

V. STATISTICAL SAMPLING TECHNIQUES

An accurate metamodel provides designers with a good understanding of the design's behavior as the design space is traversed. The requirement of obtaining accurate results with a small number of samples will minimize the time for design development and circuit generation, which includes the generation of the final physical layout. By creating an accurate metamodel one can optimize the design to the needed specifications. We divided the sampling techniques into three different categories: random, uniform and Design of Experiments (DOE). The generated sample data can be fitted in many different ways to generate a metamodel. Of course, the choice of fitting algorithm can affect the accuracy of the metamodel. For comparison purposes, all future data will be fitted into polynomial regression models in powers of four except the DOE which is fitted in powers of two due to the lack of sampling points. Thus the metamodel has the following form:

$$y = \sum_{i,j=0}^k \left(\alpha_{ij} \times x_1^i \times x_2^j \right), \quad (2)$$

where y is the response being modeled (frequency in our case), $\mathbf{x} = [W_n, W_p]$ is the vector of design variables and α_{ij} are the coefficients determined by the polynomial regression. $k=4$ except in the case of DOE where $k=2$.

Of course, the *true* response of the circuit is typically unknown because we are working with a limited number of samples. However, since the test circuit is intentionally simple (to allow exhaustive sampling), we can use 100,000 points to generate an extremely accurate "golden" response surface which can be used for validation and evaluation of the various metamodels. In the following discussion, the "golden" response will be taken as the true circuit response. In more complex circuits, the actual verification will probably use under 100 sample points and, for very large circuits, substantially less points. The square root of mean square error (RMSE), shown in equation 3, is used to compare the sampled data response of the parasitic netlist to the "true" response. RMSE shows the departure of the metamodel from the true model. The smaller the RMSE value, the better the metamodel [4]. Since we are also interested in the accuracy of the metamodel, we calculated the standard deviation (σ) for all 100,000 verification points of the exhaustive sampling, which is calculated as shown in equation 4. The generation of the sampling points, SPICE runs and post-processing calculations are done automatically using a combination of commercial and in-house tools by using the following expressions:

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(x_k) - \hat{y}(x_k))^2}, \quad (3)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (|y(x_k) - \hat{y}(x_k)| - RMSE)^2}. \quad (4)$$

$N=100,000$ random points x_k are selected in the design domain T to evaluate the metamodels. These points are

checked to ensure that they are not the same points used in the generation of the golden model. This would generate artificially small values of the RMSE. $y(x_k)$ and $\hat{y}(x_k)$ are the responses at point x_k of the golden model and the metamodel, respectively.

A. Exhaustive Sampling

Exhaustive sampling could be used if the simulation time is not an issue. With m as the number of variables and n as the number of runs for each variable the amount of samples it will take to create a metamodel will be n^m . The RMSE for a large amount of m is very small. In our case, taking in consideration the width of PMOS and NMOS as variables and running the simulation for these two variables 100 different times each, we obtain 10,000 simulation results for the given RO. The RMSE for a metamodel of that amount of number of runs is minimal. Figure 5 shows the surface of the frequency output on the z -axis with W_n and W_p on the x - and y -axis, accordingly. Since the calculated RMSE is very small for this metamodel, we can conclude that we can use the generated metamodel's data as the golden model for a comparison as the actual results for future simulations. Running this many simulations to receive an almost perfect metamodel is usually not practical in the design process. We will try to minimize the sampling amount for these two variables to receive the best metamodel that fits the design for the given ring oscillator.

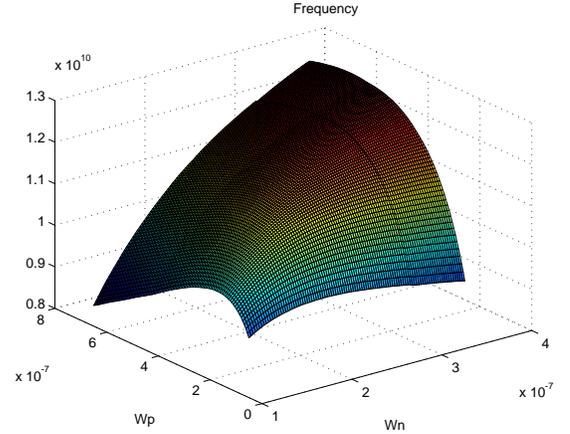


Fig. 5. 10,000 data points exhaustive sampling.

B. Random Sampling: Monte Carlo

Monte Carlo or random sampling is a technique which samples the data for each variable, by picking n random data points for each variable in the domain T . Figure 6 shows the results for creation of multiple metamodels with different number of sample amounts and their RMSE results. Note that the RMSE and its standard deviation will both change each time if the simulation is performed with the same amount of data points, since the data could have some areas of unsampled points or an over-abundant number of points in one area which is caused by the uneven distribution of sampling points.

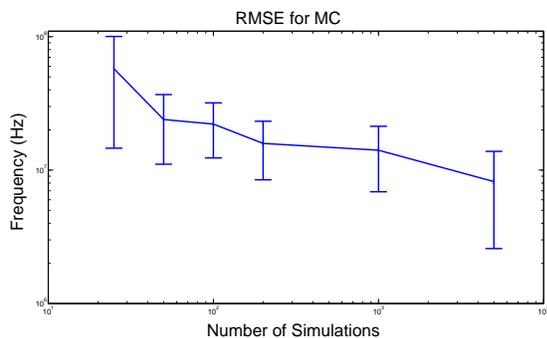


Fig. 6. RMSE data for Monte Carlo sampling. The error bars have unequal lengths due to the logarithmic scale.

C. Uniform Sampling

There are different kinds of uniform sampling techniques. Latin Hypercube Sampling (LHS) and Middle Latin Hypercube Sampling (MLHS) techniques are common. There are also many variants which are derived from these two, such as Orthogonal array-based Latin hypercube design, Symmetric Latin hypercube design, orthogonal column LHS, and Optimal Latin Hypercube design. Uniform sampling results in a distribution that is even. Given that the points are more evenly spaced out in the domain of T this distribution of points produces more effective coverage than random sampling. Uniform sampling techniques can deal with a large number of runs and input variables. They also are computationally cheap to generate. Both LHS and MLHS RMSE results are smaller than simply random sampling technique such as Monte Carlo. Both divide the domain T into n amount of Latin squares, and a data point is then sampled from each square. The drawback for both designs is that the smallest possible variance for the sample mean can never be reached [4].

1) *Latin Hypercube Sampling*: Latin Hypercube Design produces a random point within the generated n amount of Latin squares on the domain T . This technique provides more evenly distributed sampling points than random sampling techniques, but the samples can still be bunched up together as the samples are taken randomly from each Latin square and they can be adjacent to each other. Considering the same number of points as the Monte Carlo generated samples, figure 7 shows the RMSE results for LHS metamodels.

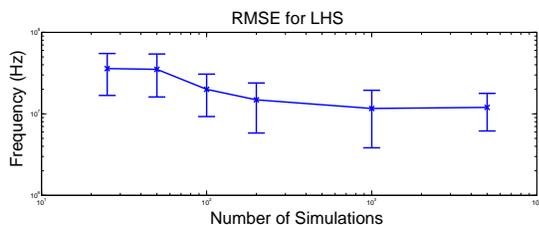


Fig. 7. RMSE data for LHS sampling.

2) *Middle Latin Hypercube Sampling*: The Middle Latin Hypercube Sampling (MLHS) technique is very similar to

TABLE III
RMSE COMPARISON FOR DIFFERENT SAMPLING TECHNIQUES (IN MHz)

Samples	MC		LHS		MLHS	
	μ	σ	μ	σ	μ	σ
25	57.5	42.9	35.6	19.1	36.0	26.2
50	24.0	12.9	35.2	19.1	27.4	14.8
100	22.1	9.79	20.0	10.7	24.8	14.7
200	15.9	7.39	14.9	9.04	20.5	11.2
1000	14.1	7.21	11.7	7.81	15.4	9.44
5000	8.20	5.62	12.0	5.84	5.99	3.04

regular LHS. It also divides the domain T into n amount of Latin squares, but instead of randomly sampling from each of those squares, it picks the middle value from each one. This technique is more uniform than the LHS, but is not able to sample the regions close to the edge of the domain T . Considering the same number of points as the Monte Carlo generated samples, figure 8 shows the RMSE results for LHS metamodels.

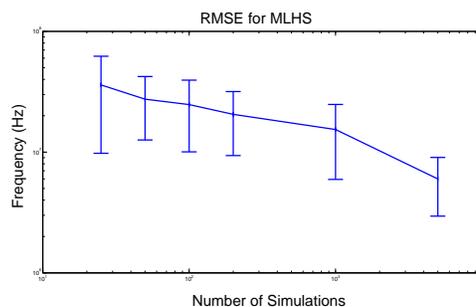


Fig. 8. RMSE data for MLHS sampling.

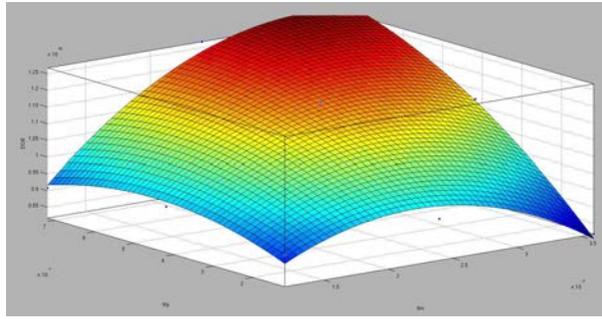
D. Design of Experiments Sampling

Design of Experiments (DOE) is a technique that is most commonly used with a large number of variables. A DOE metamodel was created from 9 points, 3 per axis and their intersections. The metamodel can only be fitted using a 2nd degree polynomial function, instead of the 4th in the other examples, due to the small amount of samples. Therefore the RMSE that was calculated for the DOE metamodel is considerably higher in comparison to the other techniques. The RMSE that was calculated for the DOE sample was 750 MHz with a standard deviation of 410 MHz. The highest variance for the error was 2.11 GHz. This indicates that DOE is not a competitive sampling technique for a small number of variables.

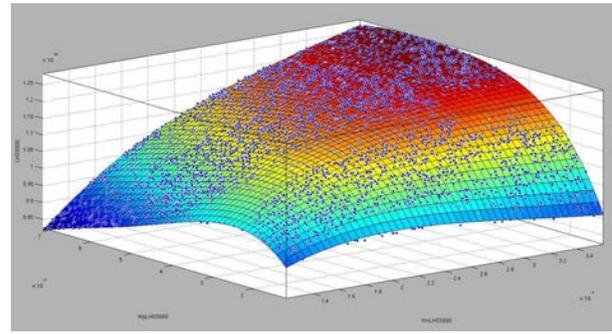
E. Comparative Discussion of Sample Data

The resultant response surfaces for the four sampling techniques discussed previously are shown in figure V-E, while table III shows a quantitative comparison of the RMSE performance for each method.

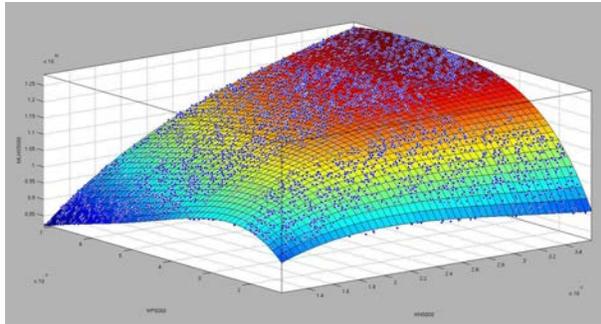
MC sampling produces higher RMSE than uniform sampling because it is random and might not cover the full spectrum of its design variable. It is clear from the data provided



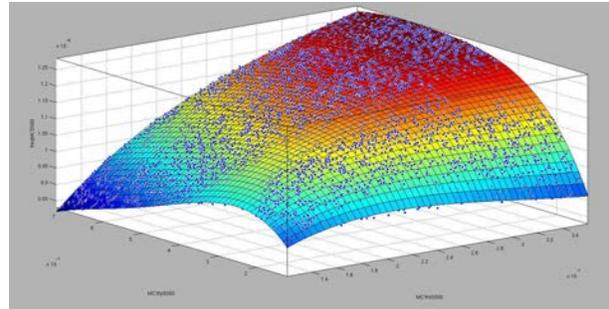
(a) DOE of 9 points



(b) LHS for 5000 points



(c) MLHS for 5000 points



(d) MC for 5000 points

Fig. 9. Response surfaces.

in table III that uniform sampling provides superior accuracy to random sampling. Designers should choose LHS or MLHS over MC but the trend in typical design environments is the opposite. This is probably due to the simplicity of running MC versus LHS or MLHS: most commercial simulators can perform MC with a simple directive. Uniform sampling, on the other hand, requires extensive setup. However, the improved accuracy is well worth the extra effort.

VI. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we presented a novel design flow using metamodels and compared commonly used sampling techniques using a nano-CMOS ring oscillator as case study. The presented design flow can be used to speed up the design process of nanoscale circuits in general. The frequency of the RO was used as the objective function for target specifications. A thorough analysis for various sampling data rates and methods demonstrates that uniform sampling techniques have better overall performance (in terms of accuracy) than the randomized and DOE sampling techniques. Whether LHS or MLHS is more appropriate for a particular design depends on whether edge effects are important or not. In our opinion, LHS is typically preferable over MLHS because it covers the design space uniformly, while, at the same time, providing for a small amount of randomness in the samples. Our future research will include specific optimization techniques as part of the proposed design flow. We will conduct an extensive study of metamodeling-based optimization for a large number of design variables by designing with our flow for more complex nano-CMOS circuits.

REFERENCES

- [1] J. Park, K. Choi, and D. J. Allstot, "Parasitic-aware design and optimization of a fully integrated CMOS wideband amplifier," in *Proc. of the Asia South Pacific Design Automation Conference*, 2003, pp. 904–907.
- [2] D. Ghai, S. P. Mohanty, and E. Kougiianos, "Design of parasitic and process-variation aware nano-CMOS RF circuits: A VCO case study," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 9, pp. 1339–1342, September 2009.
- [3] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. New York: Springer, 2003.
- [4] K.-T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. London: Chapman & Hall/CRC, 2006.
- [5] A. Lamecki and M. Mrozowski, "Design of integrated inductors using parameterized surrogate models," in *Proc. of the International Conference on Computer as a Tool (EUROCON)*, 2007, pp. 102–105.
- [6] A. Lamecki, L. Balewski, and M. Mrozowski, "Towards automated full-wave design of microwave circuits," in *Proc. 17th International Conf. Microwaves, Radar and Wireless Communications*, 2008, pp. 1–2.
- [7] R. Samanta, J. Hu, and P. Li, "Discrete buffer and wire sizing for link-based non-tree clock networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 7, pp. 1025–1035, July 2010.
- [8] J. L. Wong, A. Davoodi, A. Khandarwal, A. Srivastava, and M. Potkonjak, "A statistical methodology for wire-length prediction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1327–1336, July 2006.
- [9] J. L. Wong, A. Davoodi, A. Khandarwal, A. Srivastava, and M. Potkonjak, "Statistical timing analysis using kernel smoothing," in *25th International Conference on Computer Design (ICCD)*, 2007, pp. 97–102.
- [10] J. A. McNeill and D. Ricketts, *The Designer's Guide to Jitter in Ring Oscillators*. New York: Springer, 2009.
- [11] L. C. Rodoni, F. Ellinger, and H. Jackel, "Ultrafast CMOS inverter with 4.7 ps gate delay fabricated on 90nm SOI technology," *Electronic Letters*, vol. 40, no. 20, pp. 1251–1252, Sept. 2004.
- [12] T. C. Weigandt, B. Kim, and P. R. Gray, "Analysis of timing jitter in CMOS ring oscillators," in *Proc. Int. Symp. Circuits Systems*, June 1994, pp. 4.27–4.30.
- [13] S. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits*, 3rd ed. New York: McGraw Hill, 2003.