Formal Approaches to Analog Circuit Verification

Erich Barke*, Darius Grabowski*, Helmut Graeb†, Lars Hedrich‡,
Stefan Heinen§, Ralf Popp¶, Sebastian Steinhorst‡, Yifan Wang§
*Institute of Microelectronic Systems, Leibniz Universitaet Hannover, Germany
†Institute for EDA, Technische Universitaet Muenchen, Germany
‡Institute of Computer Science, Goethe University of Frankfurt/Main, Germany
§Chair of Integrated Analog Circuits, RWTH Aachen University, Germany
¶edacentrum GmbH, Germany

***Abstract*—For a speed-up of analog design cycles to keep up with the continuously decreasing time to market, iterative design refinement and redesigns are more than ever regarded as showstoppers. To deal with this issue, referred to as design and verification gap, the development of a continuous and consistent verification is mandatory. In digital design, formal verification methods are considered as a key technology for efficient design flows. However, industrial availability of formal methods for analog circuit verification is still negligible despite a growing need. In recent years, research institutions have made considerable advances in the area of formal verification of analog circuits. This paper presents a selection of four recent approaches in analog verification that cover a broad scope of verification philosophies.**

I. Introduction

Within the last decade, verification has become one of the most important topics in circuit and system design. Up to 70 % of the design time can be consumed by the verification process. Therefore, successful design in terms of correctness, handling complexity, and time to market requires large efforts in the development of verification methodologies. It was no surprise that verification in digital design has gained enormous ground due to the lessons learned from the Pentium bug. Especially formal methods have been developed that boosted the verification of digital circuits.

Verification of analog circuits is completely different, still being a matter of intensive research. Not only the continuity of an analog signal in time and value and its comparability is responsible for that; additionally, decreasing device sizes and increasing process variability lead to enormous design and verification challenges, exacerbated by an ever growing number of parameters and physical effects that have to be considered. At the end of the day, verification of analog circuits is a difficult job that mostly has to be done manually by analyzing simulation results.

In recent years, research institutions have made considerable advances in the area of formal verification of analog circuits. This paper presents a selection of four recent approaches in analog verification that cover a broad scope of verification philosophies.

This work was partly developed within the project VeronA (project label 01 M 3079) which is funded within the Research Program ICT 2020 by the German Federal Ministry of Education and Research (BMBF).

According to Figure 1, verification of analog circuits can be divided into simulation methods and formal methods. Simulation methods for analog verification aim at providing the possibility of simulating a whole system. This involves behavioral and structural modeling and simulation methods across different levels of abstraction for system components from the analog, digital or other domains. The first approach presented in this paper is such a mixed-signal/mixed-domain approach. Formal methods for analog verification on the other hand aim at providing a systematic coverage of the property space. This can be done by formulating properties based on classical modified nodal analyses or by deriving circuit characteristics from the state space. The second and third approach presented in this paper deal with analog verification in the nodal space. Specifically, they consider manufacturing tolerances and operating variations. The fourth approach deals with formal verification of analog circuits in the state space. Based on circuit characteristics that are derived from an entire coverage of the state space, verification results hold for all possible input signals and any state the system can adopt.

In the following sections, these four approaches to analog circuit verification are presented. Afterwards, a conclusion about the practicability and maturity of these approaches is drawn.

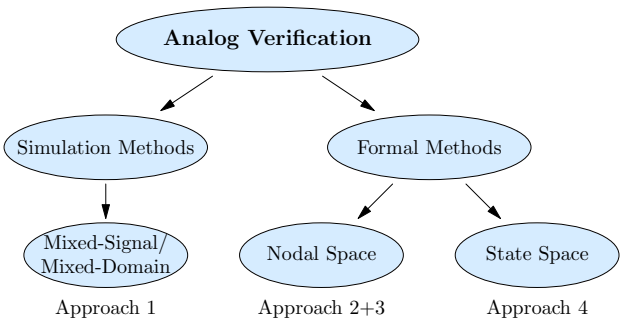


Fig. 1. Verification approaches for analog circuits.

978-3-9810801-5-5/DATE09 © 2009 EDAA

II. Mixed-Signal and Mixed-Domain Modeling for RF SoC Functional Verification

The verification of highly integrated, state of the art RF-SoCs requires new simulation and modeling methods. While there are enhanced simulation possibilities for purely periodic time-varying circuits (PSS, HB) [1], they all suffer from the fact that they are not applicable to large mixed-signal circuits, which in general do not provide those basic periodic conditions. In these cases, it is essential to go back to straight transient simulations, which should adequately present the system's behavior. This requires solving problems like unacceptable simulation time and the need for tremendous computation power, especially for the verification of RF front-ends with very high carrier frequencies [2], [3]. Therefore, the modeling of the subsystems and circuits becomes an indispensable task for the functional verification flow [4]. Hardware description languages offer the possibility to model the behavior of the circuits, reducing the number of equations by substituting the abstract coherences between the components with simpler mathematical descriptions of the circuits' behavior.

The event driven simulation approach (1) excludes the high frequency signal path from the analog domain, using the new *wreal* datatype introduced by Verilog-AMS. With this approach, the high frequency signal path can be extracted from the analog matrix and put into the digital event driven domain, while keeping other low frequency components like baseband polyphase filters and biasing concepts in the analog domain. Keeping high frequency signals in the digital domain does not lead to less calculations, but the size of the equation package that has to be solved for each of the time steps is reduced dramatically. Since only isolated portions are calculated and not the whole analog matrix, the simulation time is much shorter [2].

Similar to analog and mixed-signal assertions in the test benches of simulation-based verification flows, the event driven modeling and simulation approach proposes an explicit mapping of all analog specifications including an implementation of the digital control. This is necessary to allow a verification of large mixed-signal systems on top level in the time domain. Accordingly, only with a precise mapping of the analog circuit specification, the requirement of switching between different abstraction levels of the modeling (view switching) can be guaranteed. The benefit of the view switching is to exploit the influence of the newly designed blocks on the whole system. While the top level simulation is performed with highly abstract models, a small portion of the newly designed block is simulated on transistor level. This approach can relieve the entire debugging processes during the functional top level verification procedure. The models have to ensure pin compatibility while switching between different abstraction levels. This is the main reason why the baseband modeling [3], [5] approach is limited within contemporary tools and is not widely applied in the industrial verification flow. It is therefore crucial to firstly define the partitioning possibilities to estimate the necessary analog specifications like e.g. nonlinearity (which includes the 1 dB compression point ICP and the intercept point at least 3rd order IIP3), frequency behavior, and noise figures for a nonlinear block like the RF frontend. Most other common figures of merit (like blocking performance, co-channel interference) arise from the nonlinearities of the building blocks. Nowadays, nonlinearity models typically do a precise calculation only for the ICP or the IIP3 [6]. In the authors' opinion, it is vital to include both figures of merit in the model to map the analog specification for verification purposes. This leads to the following additional calculation. Using a taylor series approach like : $y(t) = \alpha_1 x(t) - \alpha_3 x^3(t) - \alpha_5 x^5(t)$. The linear gain is donated as $\alpha_1 = 10^{G_{dB}/20}$ with the influence of the 3rd order intercept point $\alpha_3 \approx -\frac{2}{3}\frac{20\cdot\alpha_1}{10^{IIP3_{dBm}/10}}$ and for the corresponding factor α_5 arising from a specified ICP:

$$\alpha_5 \approx \frac{0.16\alpha_1\left(10^{\frac{IIP3_{dBm}}{10}+1} - 10^{\frac{IIP3_{dBm}}{10}+\frac{19}{20}} + 100\cdot A_{ICP}^2\right)}{A_{ICP}^4 10^{\frac{IIP3_{dBm}}{10}}}$$

with $A_{ICP} = \sqrt{2\cdot\frac{R}{1000}\cdot 10^{\frac{ICP_{dBm}}{10}}}$.

Table I shows the resulting performance benefits for the higher order realization of a single nonlinearity. The specifications based on circuit level simulations are: $ICP = -16.9\ dBm$ and $IIP3 = -7.85\ dBm$.

Parameter-Calculation	α_3	α_5	IIP3-Error (dB)	ICP-Error (dB)
Taylor (ICP)	894.00	-	0.59	-0.03
Taylor (IIP3)	1023.15	-	0.00	-0.57
Taylor 5th order	-1023.15	1126810	0.00	-0.05

TABLE I
Comparison of different implementation variants for the nonlinearity parameter.

The explicit mapping of the specification leads to a parameterizable top level test bench [2], [3], which provides the possibility to verify and optimize routines and distributions of specifications, based upon an identical test bench throughout the whole design and verification process. It therefore effectively bridges the gap between system engineer and circuit designer. With the proposed event driven simulation method, the receiver chain of a RF SoC can be simulated using its pin compatible models in roughly 15 minutes for 500 μs system time. This leads to a speed up of about 7900 compared to the simulation at transistor level. During the simulation, the complete mapping of noise, nonlinearity, memory effects and digital controlling have been taken into account. The implementation of the proposed approach into available design flows can lead to a fast and reliable functional verification process.

III. Transistor Level Simulation with Parameter Uncertainties

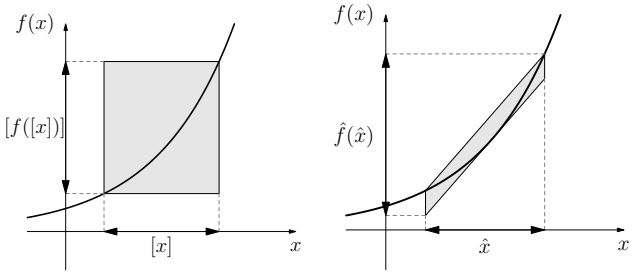
The extensive runtime needed for simulation of AMS-circuits with parameter uncertainties is still a bottleneck in the design process. On the one hand, statistical methods predict the yield in terms of specifications which are met by the circuit quite precisely, however a large number of simulation runs is needed for sufficient confidence. Simulations based on range arithmetic promise reduced simulation time at the expense of pessimistic results. This approach (2) presents a new field entered by the application of affine arithmetic in circuit simulation [7]–[9]. Design aids for integrated circuits based on probabilistic affine forms have already been presented in [10], [11].

The approach is based on affine forms which are an extension of the interval concept. An affine form $\hat{x}$ is made up by a central value and a sum of partial deviations $x_i \in \mathbb{R}$ scaled by formal symbols ϵ_i:

$$\hat{x} := x_0 + \sum_{i \in \mathcal{N}_{\hat{x}}} x_i \epsilon_i, \quad \epsilon_i \in [-1, 1] \ . \tag{1}$$

Each ϵ_i corresponds to a distinct source of uncertainty throughout all computations so that linear correlations are preserved in contrast to interval analysis storing no correlations. $\mathcal{N}_{\hat{x}}$ is a finite set of positive integers.

Nonlinear operations on affine forms are safely included by affine extensions. For this purpose, each nonlinear operation introduces a new deviation $x_i \epsilon_i$ which includes the error of linearization. This error is proportional to the square d^2 where d is the width of the input interval. In contrast, errors from interval arithmetic are proportional to d. For this reason, affine arithmetic will often yield much tighter bounds as shown in Figure 2. The shaded areas indicate the approximation error which is reduced significantly by affine arithmetic.



(a) Interval arithmetic (b) Affine arithmetic

Fig. 2. Extensions for nonlinear functions.

Circuit simulation at transistor level requires the computation of solutions for nonlinear equation systems, which describe the map between parameter space $\mathcal{P}$ and nodal space $\mathcal{X}$ implicitly. $\mathcal{P}$ is specified by parameter uncertainties which are described by affine forms in our approach. For the nonlinear implicit map from $\mathcal{P}$ to $\mathcal{X}$ a new Newton-based method has been implemented. It takes advantage of the linear correlation expressed by affine forms and computes from the inclusion property of affine extensions a safe inclusion of the nodal quantities. Affine analysis has been implemented for the general simulation modes: DC, AC and TR simulation.

Figure 3 shows the results of an AC-analysis with range arithmetic applied to an operational amplifier. The solid lines depict the inclusion of the open loop gain computed by affine arithmetic whereas the dashed lines describe the results from interval analysis. Note, that in both cases the circuit is linearized on the affine DC-range computed by affine DC-simulation. Obviously, the results from interval analysis are very pessimistic and therefore unsuitable for verification purposes. The new approach provides more realistic results with reduced overestimations by taking into account linear information throughout the computation process.

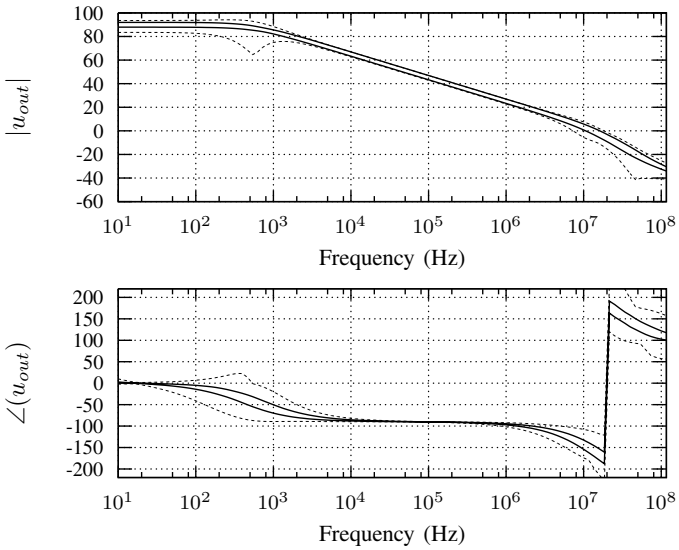


Fig. 3. Bode plot: open loop gain of an operational amplifier.

IV. Tolerance Verification in the Nodal Space

Models for performance properties are the basis for formal verification of analog circuits. These models have to capture the complex physical behavior of integrated circuits. Therefore, performance properties are modeled on top of SPICE-type circuit simulation.

A straightforward approach is to find the property modeling on the same set of system coordinates as the circuit simulation does. These are node voltages and branch currents in the equation system of the modified nodal analysis, which are computed as constants (DC simulation), in dependence of frequency (AC simulation), or in dependence of time (TR simulation). Performance properties like delay, power, gain, and bandwidth are then formulated as functions of node voltages. Even if the design is also based on performance properties formulated in the nodal space, a verification based

on dedicated algorithms can still check the completeness of a design, because the design process is usually not determined but rather is at the discretion of the designer. In addition, available fast and powerful circuit simulation tools can be utilized for verification.

In this work, the following two issues for an approach (3) to tolerance verification in the nodal space have been considered:

- Instead of a specific circuit sizing provided by design, the whole range of achievable trade-offs between competing performance objectives shall be considered. The comparison of the resulting Pareto front with the specific design allows to verify if the design is optimal, if the design process has been performed well, and how conflicting performance objectives have been arbitrated.
- Manufacturing tolerances and operating ranges (e.g. of temperature, supply voltage) shall be considered. This allows to verify the completion of the design with regard to manufacturability and operability.

The details of the approach are described in [12]. Essentially, a nominal Pareto optimization without consideration of parameter tolerances is performed first. It leads to a discretized Pareto front with Pareto points that represent specific trade-offs. On top of each Pareto point, a realistic worst-case analysis is performed, which computes the worst-case performance values achievable under a given minimum yield requirement [13]. The individual worst-case performance values from the worst-case analysis are combined into a new worst-case Pareto point that corresponds to a nominal Pareto point. All worst-case Pareto points form the worst-case Pareto front for a minimum yield requirement. The trade-offs given by the worst-case Pareto front represent combinations of performance value specifications that can be offered for the given yield requirement and the specified operating range.

An operational amplifier (Fig. 4) serves to illustrate the approach. The basis has been an industrial 180nm technology with a corresponding statistical parameter distribution. On a dual-quadcore 1.86 GHz Xeon PC with 4 GB RAM, the computation time for the nominal and worst-case Pareto front has been 1h 21min.

The upper curve represents the nominal Pareto front, and the individual nominal Pareto points that have actually been computed are marked with a plus. This curve shows that a correct design should have, for instance, a slew rate of $13\frac{V}{\mu s}$ and a DC gain of $75dB$, or a slew rate of $3\frac{V}{\mu s}$ and a DC gain of $95dB$, or any other combination on this curve. If a nominal design under verification features a combination of values that corresponds to a point below the curve, this means that the design is not complete.

The lower curve represents the worst-case Pareto front, the individual worst-case Pareto points that have actually been computed are marked with a cross. The minimum yield

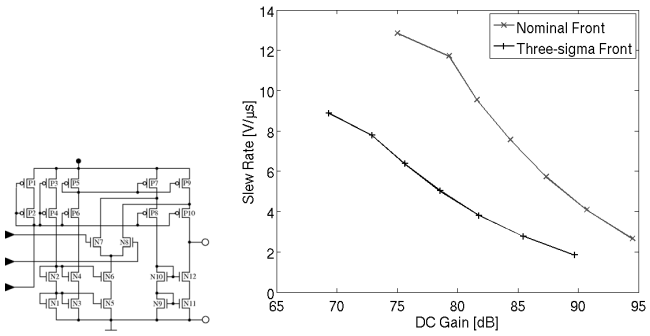


Fig. 4. Folded-cascode operational amplifier and 3-sigma Pareto fronts for slew rate and DC gain.

requirement is three sigma, which corresponds to a yield of 99.9% This curve shows that a correct design for 3 sigma manufacturability should specify for instance a slew rate of $9\frac{V}{\mu s}$ and a DC gain of $70dB$, or a slew rate of $2\frac{V}{\mu s}$ and a DC gain of $90dB$, or any other combination on this curve. If a tolerance design under verification features a combination of specification values that corresponds to a point below the curve, this means that the design for manufacturability is not complete.

This example shows how a verification can be performed based on performance space exploration. The significant difference between the two curves shows the necessity to consider parameter tolerances.

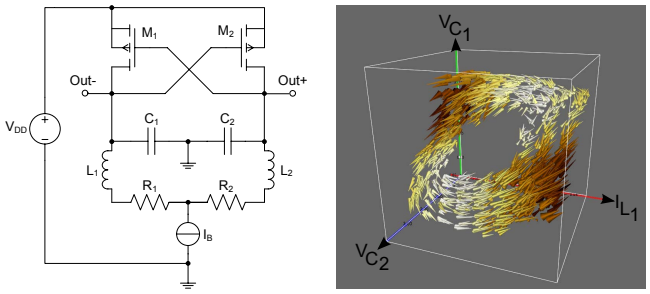
V. Analog Formal Verification in the State Space

Conventional circuit characterization relies on the circuit designer's expertise and experience to find the right test benches and input stimuli in order to assure compliance to the specification. In future automated analog design flows, such manual verification approaches will be too time consuming and incomplete, as specification conformance cannot be proven.

In contrast, formal verification approaches (4) guarantee their results to hold for all possible states and all possible input signals of the system under verification. The key to the approaches of formal verification of analog systems presented in the following is to create a state space representation of the circuit's behavior. By applying modified nodal analysis (MNA) to the circuit netlist, the circuit is represented by a nonlinear first order differential algebraic equation (DAE) system

$$f(\dot{x}(t), x(t), u(t)) = 0.$$

The state space is spanned by the input vector $u(t)$ and the vector of the system variables $x_e(t)$, representing the energy-storing elements of the circuit, such as capacitances and inductances. An example is given in Figure 5, visualizing a state space representation for the oscillation area of a



(a) (b)

Fig. 5. Circuit schematic of an LC oscillator ($V_{DD} = 3.3V$, $I_B = 0.05A$, $R_1 = R_2 = 20\Omega$, $L_1 = L_2 = 10nH$, $C_1 = C_2 = 10pF$) (a) and the vector field of the oscillation area representing the circuit's dynamic behavior in state space (b).

LC oscillator circuit as a discrete vector field, projected to state space dimensions V_{C_1}, V_{C_2}, and I_{L_1}. On the state space representation, different algorithmic approaches target special verification tasks:

- *Model Checking*, i.e. check automatically whether or not specified circuit properties meet the specification, written in a machine-readable specification language.
- *Equivalence Checking*, i.e. prove the input-output equivalence of two systems, for example enabling the exchange of transistor netlists and behavioral models on block level with certainty regarding the modeling error introduced into the top-level system simulation.
- *Complete State Space-Covering Input Stimuli Generation*, i.e. generate input stimuli for simulation of the circuit covering every state that the system can adopt. The usage of such stimuli will enhance the results of conventional transient simulation to cover all existing corner cases.

In the following, the aforementioned three approaches are presented.

A. Model Checking

To apply model checking algorithms, the state space representation is transferred into a discrete graph structure. Therefore, the continuous state space is divided into hypercubes of homogeneous behavior by comparing length and angle of the system variables' derivatives $\dot{x}_e(t)$. These derivatives also define the transition relation of the hypercubes and the transition time. By considering each hypercube as a vertex of a graph, connected according to the transition relation with the corresponding transition times, a graph data structure is generated from this information. The vertices of the graph are labeled with the state space variable values at the center of the corresponding hypercube.

To the generated graph structure, analog model checking algorithms in conjunction with a special Analog Specification Language (ASL) [14] are applied. Compared to approaches using temporal logic specification [15], the number of verifiable circuit properties such as Offset, Gain, CMRR, PSRR, Slew Rate, Overshoot, Startup Time, Oscillation, and VCO Input Sensitivity is increased significantly.

B. Equivalence Checking

Equivalence checking of analog systems can be performed by a comparison of dynamic behavior in the state spaces of the systems under verification [15]. Generally, the compared circuit implementations differ in the number of system variables, hence a direct comparison is impossible. However a nonlinear transformation to a canonical representation for each system can be calculated. Due to the complexity of nonlinear analog circuit descriptions by a system of nonlinear differential algebraic equations, symbolic approaches to this transformation are not feasible.

Therefore, a numerical approach is performed by sampling the state spaces and comparing the system behavior in each of these sample points. A local linear transformation function to a canonical representation (Kronecker's canonical form) for each sample point of the systems has to be calculated, and by numerically integrating these linear local transformations, an approximation of the nonlinear transformation for the system is obtained. The numerical differences between both internal transformed dynamics and the output variables give a direct measure for the equality/difference of both systems.

C. Complete State Space-Covering Input Stimuli Generation

Transient simulation results are highly depending on the circuit designer to find the right input stimuli to identify critical system behavior. Obviously, model or equivalence checking can resolve this problem. However, transient simulation is the state of the art in industrial design flows and introduction of completely new methodologies will take some time. Therefore, profiting of formal techniques by generation of input stimuli for transient simulation that cover the complete state space of an analog circuit is possible without fundamentally changing the design flow.

Working on the same graph structure obtained by the discrete modeling process for model checking as described above, generating stimuli that cover the complete reachable state space of the system is done by a complete traversal of the graph structure [16]. The algorithm starts at a vertex representing a DC operating point of the circuit and then follows the edges of the graph structure from vertex to vertex, storing the corresponding input signal values at the visited vertices and the transition times. Therewith, tuples of value and time are obtained, representing a piecewise linear input stimulus for each input of the system. These input stimuli can now be used in a circuit simulator, bringing the circuit to every reachable state during simulation, resulting in complete state space coverage of the simulation. Moreover, assertion-based verification approaches could be enhanced by usage of complete state space-covering input stimuli to provide formal verification results.

All of the three approaches presented in this section can contribute to more design efficiency by introducing more automation and formalization to the verification process.

VI. Conclusion

In this contribution, four different verification approaches for analog circuits have been presented. Concepts for enhancing the state of the art in mixed-signal functional verification as well as tolerance verification in the nodal space by systematic circuit simulations can contribute to a formalization of the design process. While not being as mature as the aforementioned two approaches, transistor level simulation with parameter uncertainties using affine arithmetic and analog formal verification can on the other hand guarantee certainty about the verification results.

Choosing the appropriate verification concept requires a well considered tradeoff as the availability of formal verification approaches is limited to block level up to now, while approaches with relaxed formalism are less restricted in circuit size and abstraction levels. Hence, the presented approaches fit into the design flow, altogether making the verification process more efficient and reliable.

References

[1] Cadence, *Virtuoso SpectreRF Simulation Option User Guide*, 5th ed. Cadence, 2005.

[2] S. J. H.-W. G. S. Heinen, "Event driven analog modeling of rf frontends," *Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International*, pp. 46–51, 20-21 Sept. 2007.

[3] S. Joeres and S. Heinen, "Functional verification of radio frequency socs using mixed-mode and mixed-domain simulations," in *Behavioral Modeling and Simulation Workshop, Proceedings of the 2006 IEEE International*, 2006, pp. 144–149.

[4] H. Chang and K. Kundert, "Verification of complex analog and rf ic designs," *Proceedings of the IEEE*, vol. 95, pp. 622–639, 2007.

[5] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems: Modeling, Methodology and Techniques*, 2nd ed. Springer, Oct. 2000.

[6] R. Frevert, J. Haase, R. Jancke, U. Knochel, P. Schwarz, R. Kakerow, and M. Darianian, *Modeling and Simulation for RF System Design*, 1st ed. Springer, Dec. 2005.

[7] M. Andrade, J. Comba, and J. Stolfi, "Affine Arithmetic (Extended Abstract)," in *INTERVAL '94, St. Petersburg, Russia*, 1994.

[8] D. Grabowski, C. Grimm, and E. Barke, "Semi-symbolic modeling and simulation of circuits and systems," *IEEE International Symposium on Circuits and Systems, ISCAS 2006*, pp. 983–986, May 2006.

[9] D. Grabowski, M. Olbrich, and E. Barke, "Analog circuit simulation using range arithmetics," *Asia and South Pacific Design Automation Conference, ASPDAC 2008*, pp. 762–767, March 2008.

[10] A. Singhee, C. F. Fang, J. D. Ma, and R. A. Rutenbar, "Probabilistic interval-valued computation: Toward a practical surrogate for statistics inside cad tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 12, pp. 2317–2330, Dec. 2008.

[11] J. Sun, J. Li, D. Ma, and J. Wang, "Chebyshev affine-arithmetic-based parametric yield prediction under limited descriptions of uncertainty," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 10, pp. 1852–1865, Oct. 2008.

[12] H. Graeb, D. Mueller, and U. Schlichtmann, "Pareto Optimization of Analog Circuits considering Variability," in *European Conference on Circuit Theory and Design (ECCTD)*, Aug. 2007.

[13] H. Graeb, *Analog Design Centering and Sizing*. Springer, 2007.

[14] S. Steinhorst and L. Hedrich, "Model Checking of Analog Systems using an Analog Specification Language," in *Proc. of the Conference on Design, Automation and Test in Europe 2008 (DATE'08)*, 2008, pp. 324–329.

[15] W. Hartong, R. Klausen, and L. Hedrich, "Formal Verification for Nonlinear Analog Systems: Approaches to Model and Equivalence Checking," *Advanced Formal Verification, R. Drechsler, ed., Kluwer Academic Publishers, Boston*, pp. 205–245, 2004.

[16] S. Steinhorst and L. Hedrich, "A formal approach to complete state space-covering input stimuli generation for verification of analog systems," in *Analog 2008: 10. ITG/GMM-Fachtagung Entwicklung von Analogschaltungen mit CAE-Methoden*, 2008, pp. 57–62.