Towards more Dependable Verification of Mixed-Signal Systems

Florian Schupfer, Christoph Grimm

Vienna University of Technology, Institute of Computer Technology
1040, Vienna, Gusshausstrasse 27-29, Austria
[schupfer|grimm]@ict.tuwien.ac.at

Abstract. The verification of complex mixed-signal systems is a challenge, especially considering the impact of parameter variations. Besides the established approaches like Monte-Carlo or Corner-Case simulation, a novel semi-symbolic approach emerged in recent years. In this approach, parameter variations and tolerances are maintained as symbolic ranges during numerical simulation runs by using affine arithmetic. Maintaining parameter variations and tolerances in a symbolic way significantly increases verification coverage. In the following we give a brief introduction and an overview of research on semi-symbolic simulation of both circuits and systems and discuss possible application for system level verification and optimization.

Keywords. Affine Arithmetic, Range based methods, Verification, Semi-symbolic simulation

1 Introduction

Today's embedded systems become more and more heterogeneous. Beside digital hardware parts performing signal processing operations and software tasks running on dedicated processors, also analog components used for RF front ends contribute to the system functionality. The fact that all these subsystems are functionally interwoven requires an overall system verification. This is done usually by overall system simulation over long periods (e.g. seconds) of simulated time in order to observe complex interactions. Two contradictory approaches are taken in order to increase dependability of verification:

More accurate models: For dependable verification, accurate models at transistor level are mandatory. The demand for more accurate models has steadily increased the number of considered device parameters in the last years [1]. Furthermore, increasing process variations required application of multi-run simulations such as Monte Carlo and Worst Case methods. The approach to increase dependability of verification by higher accuracy leads to simulation times that are often within days for a single simulation run.

Dagstuhl Seminar Proceedings 10271
Verification over discrete-continuous boundaries
http://drops.dagstuhl.de/opus/volltexte/2010/2791

More abstract models: For dependable verification, a high number of simulation runs is required to achieve a high verification coverage. Considering that a single simulation run of a complex embedded system requires at least hours or even days, abstraction is mandatory to be able to simulate many possible inputs.

In order to overcome the required tradeoff between accuracy and abstraction, new approaches are required. In recent years, a new "semi-symbolic" approach has evolved which wants to combine both accuracy *and* abstraction in a single model. In the "semi-symbolic" approach, the accurate behavior (which would be inefficient to simulate) is included into ranges of a more abstract model [2]. For semi-symbolic simulation, the concept of Affine Arithmetic [3] is used.

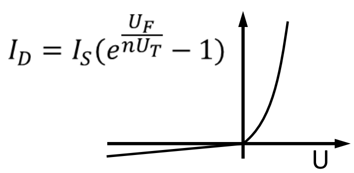

Fig. 1. Detailed description

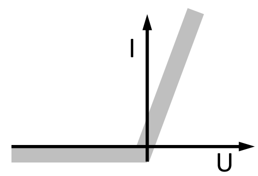

Fig. 2. Abstracted range description

As a simple example for dependable and abstract modeling, Fig.1 shows a commonly used (still simple) diode model. Accurate diode models are much more complex, and, due to process variations, Monte Carlo and Worst Case techniques must validate the overall application against impact of process variations. Nevertheless, for most applications the simple fact that a diode conducts electric current in only one direction is fully sufficient for many applications. Fig.2 describes this characteristic using ranges. Note, that the model is considered as abstract, but also accurate in the sense that it *includes* the real physical behavior.

The fact that the real physical behavior is included (even considering worst case parameter variations) allows getting an over-approximation of all possible outputs (and reachable states). This enables a dependable verification of safety properties: If simulation runs do not reach an insecure output or state, any model included in the abstract, range-based model will also not do this. However, the verification of lifeness properties (e.g. guarantee of reachability) is out of scope in this approach.

2 State of the Art

The traditional approach to verification of mixed-signal system is to perform as many simulation runs as possible to achieve a high coverage. In order to validate the impact of parameter variations, similar input stimuli are applied with different parameter sets (multi-run simulations). The objective of Monte Carlo simulation [4] is to reason about impact of parameter variances and distribution functions of the system properties. In addition, corner cases are simulated to identify potential worst case parameter sets. The number of parameters influences the number of necessary simulation runs exponentially in order to simulate all corner values. Corner cases are not necessarily worst case parameters [5]. The number of simulation runs can be reduced by statistical approaches such as 'Design of Experiments' [6], [7] for finding worst case parameter sets, or 'importance sampling' [8] for more accurate estimation of statistical properties.

Formal and symbolic methods [9], [10] for overall system verification are still in early stages and not able to handle complexity and heterogeneity of today's embedded systems. The semi-symbolic approach [11], [12], [13], [14], [2] [15] provides an interesting compromise between the efficiency and completeness of formal methods with the usability of simulation based verifications. The completeness of verification does not hold in a full formal sense but can be shown for the effects of the range modeled parameters and signals on the system at least for verification of safety properties.

3 Semi-symbolic Modeling and Simulation

Many parameters in models of technical systems are subject to statistical deviations due to aging, process variations, or simply noise, for example. Such parameters can be abstracted by using regions (or subspaces in case of multiple dimensions) which include the varying parameter. In the following we show some examples for modeling typical uncertainties or deviations by using ranges. For formally describing ranges, we use symbols ϵ_i (range symbols) that represent ranges $[-1, 1]$.

Production tolerances: Analog implementations often have a static deviation $\pm e$ from the ideal behavior. This can be modeled by adding a range with the radius e to the ideal value:

$$\mathrm{tol}(\hat{y}, e) = \hat{y} + \epsilon_i e$$

Quantization: Quantization of a continuous quantity can be modeled by adding a range with a radius of a half quantization unit $Q/2$, which models the worst case deviation:

$$\mathrm{quant}(\hat{y}, Q) = \hat{y} + \epsilon_j[n]Q/2$$

Truncation of numerical operations, such as multiplication can be handled in the same way.

An interesting approach to perform a range based simulation of complex, heterogeneous systems is the combination of Affine Arithmetic [3] with SystemC AMS for modeling/simulation [14] at system level, and a numerical SPICE-like environment for transistor-level simulations [16]: The SystemC AMS environment can easily be extended by using an Affine Arithmetic library, which overloads certain computation related operations, introducing the semi-symbolic methodology for modeling/simulation as described in [2].

3.1 Affine Arithmetic

Affine Arithmetic extends the concept of Interval Arithmetic [17] with symbolic range identifiers in order to overcome the problem of over-approximation. It is also denoted as a semi-symbolic technique because it describes ranges by a central value (that is a numerical value) and a sum of interval valued partial deviations and noise symbols (that represented in a symbolic way) [3]. The major advantage is that maintaining corellation information in a symbolic way totally avoids over-approximation for linear operations, and enables a number of different techniques for defining linear inclusions for non-linear operations with very limited over-approximation. Each affine expression represents the influence of independent sources of uncertainty by a sum of partial deviations $x_i \epsilon_i$. The noise symbol ϵ_i represents the range $[-1, 1]$ which is scaled by the deviation value x_i. Affine expressions are referred to by ˜ in the following.

$$\tilde{x} = x_0 + \sum_{i=1}^{length} x_i \epsilon_i \qquad \epsilon_i \in [-1, 1] \tag{1}$$

$$\tilde{x} \pm \tilde{y} = (x_0 \pm y_0) + \sum_{i=1}^{length} (x_i \pm y_i)\epsilon_i \tag{2}$$

$$c\tilde{x} = cx_0 + \sum_{i=1}^{length} cx_i \epsilon_i \tag{3}$$

Equation 1 gives the composition of an Affine Arithmetic symbol which is also referred to as Affine Arithmetic Form. The number i of partial deviations correlates with the sources of uncertainty which affect this particular quantity. Equation 2 and 3, specify the so called affine operations which result in exact solutions. All other operations can solely be solved by approximating the exact result, which is performed by computing the approximation solution and adding an additional $x_{i+1}\epsilon_{i+1}$ to enclose the remaining residual.

Thus, for nonlinear operations affine arithmetic provides (over-) approximations that always include the correct result but might be pessimistic, e.g.:

$$\begin{aligned} \tilde{x} \cdot \tilde{y} := (x_0 \cdot y_0) + \sum_{i=1}^{length} (x_0 y_i + x_i y_0)\epsilon_i \\ + \ \mathrm{rad}(\tilde{x}) \cdot \mathrm{rad}(\tilde{y})\epsilon_{m+1} \end{aligned}$$

where $\mathrm{rad}(\tilde{x})$ is the radius of an affine variable $\tilde{x}$.

3.2 Semi-Symbolic System Level Simulation

The idea of semi-symbolic system level simulation is as follows [11]: In an existing simulator, numerical operations are replaced by affine operations as sketched in Section 3.1. This allows to model uncertainties, and to use the simulator e.g. for standard transient simulations. However, as simulation results we get the ideal behavior of the system quantities described by the central values x_0 and the potential deviations modeled by the sum of all x_i with $i > 0$.

The implementation is based on SystemC AMS [18]. In SystemC AMS the behavior of blocks is specified by a C++ method called *processing*. This method computes the outputs of the block depending on given inputs. Furthermore, a C++ class library is used that provides the abstract data type AAF and overloads the mathematical operations as described in Section 3.

The cental model of computation used in SystemC AMSs is *Timed Synchronous Dataflow* (TDF) which is also used for the semi-symbolic simulation. It is a timed version of the *Synchronous Dataflow* (SDF) which allows to determine a schedule of process executions before simulation. The fact that data flow simulation does not requires solving equations that depend on stimuli makes it well-suited for the semi-symbolic approach. Furthermore, the C++ based nature of SystemC AMS allows easy integration of additional libraries, such as the Affine Arithmetic library. This extensibility makes SystemC AMS to the best choice for semi-symbolic simulations.

For a first proof of concept a control loop with uncertainties modeled by symbolic terms at three positions [11] has been simulated as shown in Figure 3. The simulation of this control loop has been carried out by using SystemC AMS and an AAF class library with a step signal as input $x(t)$.

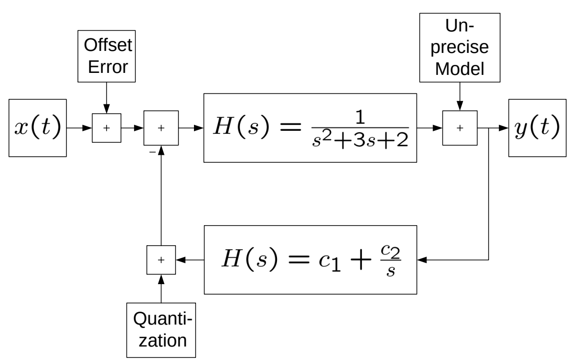

Fig. 3. Block diagram of the control loop with PI controller

A semi-symbolic simulation provides a sequence of AAF samples $\tilde{y}[nT]$ describing the ideal transient simulation result $y_0[nT]$, and the potential contribution of all uncertainties by the values $y_i[nT]$. In $\tilde{y}$ the ϵ_i are still symbols. From this symbolic representation the particular output signals are given by assigning the symbols ϵ_i real values. Figure 4 shows areas containing the step response of the control loop example.

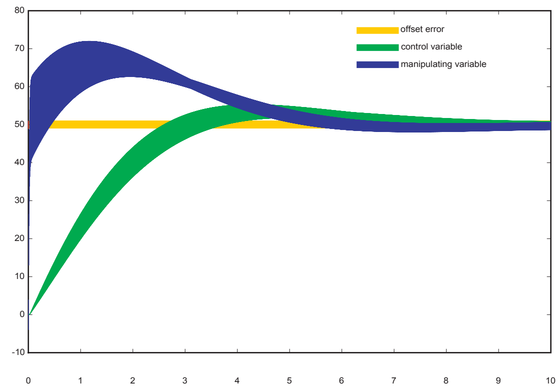

Fig. 4. Graphical representation of potential output signals

With different, both linear and nonlinear system models following observations have been made [14], [2]:

1. For systems containing only affine operations, the output signals are computed accurately. There is no over-approximation, and the number of noise symbols remains constant.
2. For nonlinear systems, there is an over-approximation that is handled accurately by new noise terms. The over-approximation depends on the kind of nonlinearity. Mild nonlinearities usually lead to quite small over-approximations. Non-contiguous functions such as a comparator cannot be handled in a reasonable way.

The fact that each nonlinear operation increases the number of noise terms – and thereby also simulation time – might be a problem. To overcome this a kind of garbage collection [2] has been added that collects the smallest terms and replaces them by new noise terms.

3.3 Semi-Symbolic Circuit Simulation

While today's digital systems are commonly modeled on higher abstraction levels through abstracted languages like Verilog, VHDL or SystemC, analog systems are still often designed by hand on transistor level. Even if analog circuits have been simulated on system level using SystemC AMS, a verification step on lower levels is often desired to verify the behavior. Therefore, a semi-symbolic circuit simulator has been developed recently [19], [12] which allows the analysis of a deviated system model even on lower levels. Compared with data flow simulation, circuit simulation requires to set up a system of (semi-symbolic) equations, and to solve it using numerical methods.

Affine circuit simulation is divided into two parts. At first a netlist of a given circuit with the corresponding device models is transformed into a mathematical representation by the Modified-Node-Analysis (MNA). MNA converts the netlist to the according system of differential and algebraic equations (Equation. 4).

$$F(\underline{x}(t), \underline{\dot{x}}(t), \underline{p(t)}, t) = \underline{0} \quad (4)$$

$\underline{x}(t)$ is the vector of time dependent variables and $\underline{p(t)}$ describes the circuit parameters. In [19], [12], MNA is performed using Maple. The intermediate result is a complete symbolic equation system. Following, all static parameters are applied and the resulting semi-symbolic system is calculated. In the second step the semi-symbolic equation system is passed to a numerical equation solver (implemented in C++), which performs DC, AC [20] and Transient-Analysis [16]. Equations are solved by applying numerical integration with either forward, backward Euler or trapezoidal methods.

The numerical computation of the outputs from the states and input quantities (solving of the non-linear DAE) follow the well known solving strategies of linearization and discretization in the current solution point. In order to handle affine arithmetic forms instead of real values, the simulation kernel uses the following method:

1. Compute x_0 by existing Newton-Raphson iteration.
2. Compute $x_i\epsilon_i$ by sensitivity analysis in each ASP using a solver.
3. Compute $NL\epsilon_{i+1}$ by numerical algorithm that minimizes over approximation, but includes potential error due to numerical integration.

In the third item an additional $NL\epsilon_{i+1}$ range is introduced which holds the numerical solving errors and together with the previous solution gives the formal solution to the circuit simulation. As every calculation introduces a new error approximation term, the number of ranges increases steadily in time and therefor a garbage collection process is activated at certain times. This garbage collection summarizes ranges which are already heavily attenuated and in this way reduces the number of ranges considerably.

Fig.5 and Fig.6 show schematics of a bandpass filter and its OpAmp transistor level structure, respectively. This system is used to demonstrate the applicability of the above procedure. The W/L parameters are considered to be deviating, the remaining parameters are constant. A Monte Carlo simulation was additionally carried out to compare the resulting range to the simulation based approach.

Fig.7 and Fig.8 show a step response with 5% correlated and uncorrelated W/L deviations. The solid line represents the Affine Arithmetic ranges whereas the dotted lines show the Monte Carlo results which reside inside the calculated range.

4 Advanced Analysis Techniques

In the first approaches, the analysis of semi-symbolic simulations concentrated on transient simulations and simulation interpretations in the time domain. However, most properties of signal processing systems are specified in the frequency

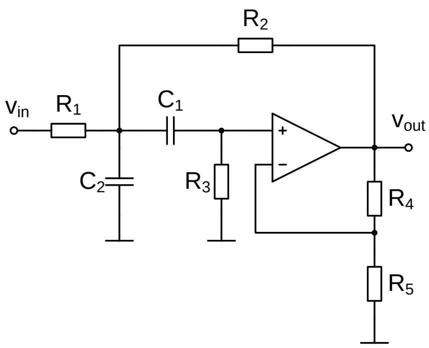

Fig. 5. Active Bandpass Filter

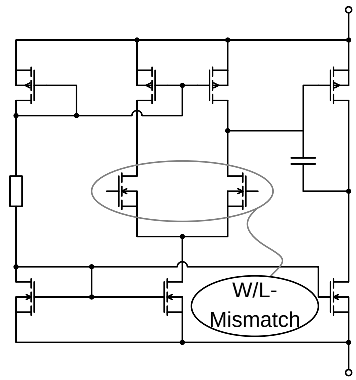

Fig. 6. MOS OpAmp

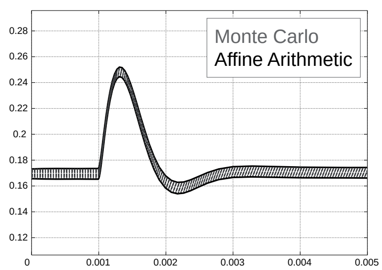

Fig. 7. Correlated W/L

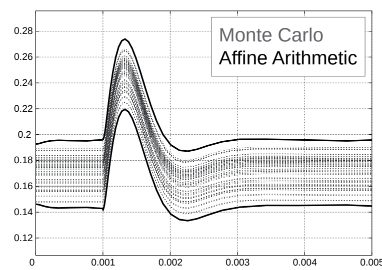

Fig. 8. Uncorrelated W/L

domain. Therfore, first steps have been taken to broaden the analysis capabilities of range based methodologies towards the frequency domain. The Discrete Fourier Transform (DFT) has been adapted to allow a transformation from range based signal representations in time domain to its frequency domain counterpart [21]. The linear nature of the DFT is used to divide the transformation into the transformation of the central value and the contribution of the range on the spectrum. Equation 5 shows the mathematical representation of such a range based DFT with $k = 0, \ldots, N-1$:

$$\tilde{F}[k] = \sum_{n=0}^{N-1} x_0[n] e^{-j\frac{2\pi k}{N} n} + \epsilon_1 \sum_{n=0}^{N-1} x_1[n] e^{-j\frac{2\pi k}{N} n} \tag{5}$$

To demonstrate the applicability of this transformation operation, a mixer circuit example has been chosen and is simulated with added abstract deviations to the mixing operation. The mixer model on the system level is reduced to a simple algebraic multiplication of its input signals. For simplification we chose sine shaped input quantities and added a low pass filter for damping the unwanted upper sideband for our demonstration. This example illustrates a down conversion in a communication receiver. Therefore only the lower sideband is of relevance.

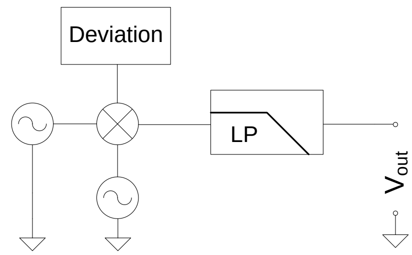

Fig. 9. Mixer with deviations on system level

In order to enhance the system model to additionally include a range, we added a block called 'Deviation' to the model description. We chose to add a dynamic partial deviation to reflect a steadily uncertain mixing property. For simplification, the deviation quantity is kept constant but could be identically implemented as a function of certain model parameters. The resulting (semi-symbolic) output of semi-symbolic simulation is finally transformed into the frequency domain by applying the Discrete Fourier Transform on the range based output quantity V_{out}.

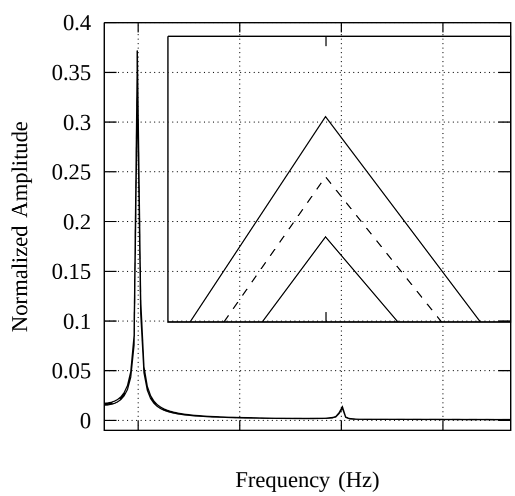

Fig. 10. Spectrum of the output signal

The resulting frequency spectrum shown in Fig. 10 gives the spectral components remaining in the output signal. The spectrum does not only provide the frequency behavior of the nominal system model – it also delivers an envelope which forms the boundaries of the range defined system behavior.

5 System Optimization with Semi-Symbolic Simulation (MARC)

A semi-symbolic simulation provides the major advantage that the different ranges, which contribute to the system output are labeled by symbolic identifiers. Thereby, a back tracing of all (possible) deviations to their cause is permanently possible, because the composition of the contributions to the deviation is maintained symbolically in the partial deviations and noise symbols ($x_i\epsilon_i$). Such a back tracing allows us a sensitivity analysis even in presence of DSP operations. Therefore, the system output can be analyzed for the influencing parameters.

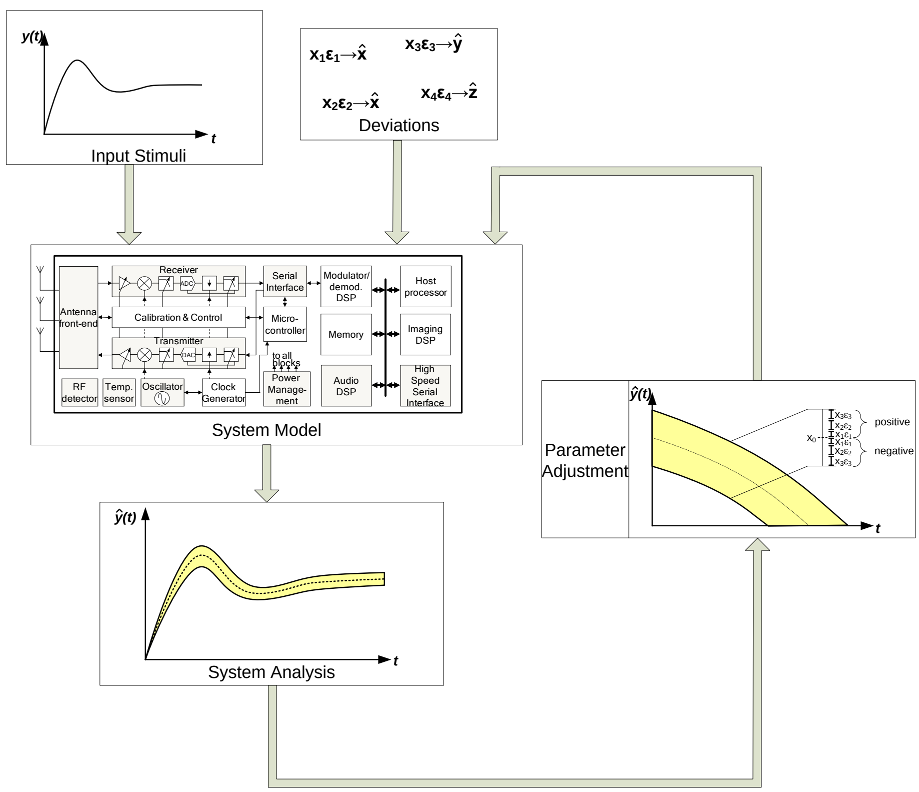

Fig. 11. System optimization within the MARC framework

Fig. 11 shows a system design flow, which uses a semi-symbolic simulation to identify parameter deviations that negatively influence the overall system behavior. First, the system is modeled using the previously described semi-symbolic approach. Deviations of parameters and uncertainties are translated into range descriptions. After that, the model together with its ranges is simulated by applying input stimuli that may also include ranges. Then, the output behavior is analyzed using various system analysis methodologies and finally the most disturbing range sources are refined for improving the system performance.

6 Conclusion, Future Work

Semi-symbolic simulations are capable of determining the worst-case behavior of systems with varying parameters in just one simulation run. The parameter variations are modeled with ranges by using Affine Arithmetic. This range based system model is following simulated performing a transient analysis. The result of such a system simulation provides a signal range which indicates a pessimistic bound for the set of possible output signals, caused by the parameter deviations. Compared to traditional Monte-Carlo and Corner-Case simulations the verification effort can be reduced significantly. A semi-symbolic simulation is currently applicable on transistor level as well as on system level. First steps to enhance the analysis possibilities has been carried out by introducing the Discrete Fourier Transform for range based signals. Further enhancements to increase the usability for industrial projects are considered and will be integrated into the MARC framework in the future.

Although the semi-symbolic techniques have been developed with a background of simulation-based verification, there is also an interesting path towards more formal verification: Formal verification of hybrid systems usually requires symbolic simulation and a representation of sets of possible states, usually using subspaces. The comparison of techniques used for formal verfication and for semi-symbolic simulation using affine arithmetic is a promising field for future work.

Acknowledgment

This work has been supported by the Austrian WWTF project MARC under contract no. ICT08_012.

References

1. M. Chan and C. Hu. Engineering BSIM for the Nano-Technology Era and Beyond. In *Technical Proceedings of the 5th international Conference on Modeling and Simulation of Microsystems*, pages 662–665, 2002.
2. Wilhelm Heupke, Christoph Grimm, and Klaus Waldschmidt. Semi-symbolic simulation of nonlinear systems. In *Forum on Specification and Design Languages 2005 (FDL05)*, Lausanne, Sep 2005. ECSI.
3. M.Andrade, J.Comba, and J.Stolfi. Affine arithmetic (extended abstract). In *INTERVAL '94*, St. Petersburg, Russia, 1994.
4. R.Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley&Sons, Inc., New York, NY, USA, 1981.
5. K. Antreich, H. Gräb, and C. Wieser. Practical Methods for Worst-Case and Yield Analysis of Analog Integrated Circuits. *International Journal of High Speed Electronics and Systems*, 4(3):261–282, 1993.
6. Douglas C. Montgomery. *Design and Analysis of Experiments*. Wiley and Sons, 2005.
7. Monica Rafaila, Christoph Grimm, Christian Decker, and Georg Pelz. Design of experiments for effective pre-silicon verification of automotive electronics. In *Forum on Design Languages 09 (FDL09)*. IEEE, 2009.
8. P. J. Smith, M. Shafi, and H. Gao. Quick simulation: A review of importance sampling techniques in communication systems. *IEEE Journal on Selected Areas in Communications*, 15:597–613, 1997.
9. Georges Gielen, Piet Wambacq, and Willy Sansen. Symbolic analysis methods and applications for analog circuits: a tutorial overview. In *Proceedings of the IEEE*, pages 287–304, 1994.
10. L. Hedrich and E.Barke. A formal approach to verification of linear analog circuits with parameter tolerances. In *Proceedings Design, Automation and Test in Europe (DATE)*, pages 649–654, 1998.
11. Darius Grabowski, Christoph Grimm, and Erich Barke. Semi-Symbolic Modeling and Simulation of Circuits and Systems. In *International Symposium on Circuits and Systems 2006 (ISCAS 2006)*, Kos, Greece, May 2006. IEEE Press.
12. Darius Grabowski. *Gebietsarithmetische Verfahren zur Simulation analoger Schaltungen mit Parameterunsicherheiten*. Verlag Dr.Hut, 2009.
13. Christoph Grimm, Wilhelm Heupke, and Klaus Waldschmidt. Analysis of Mixed-Signal Systems with Affine Arithmetic. *IEEE Transactions on Computer Aided Design of Circuits and Systems*, 24(1):118– 123, 2005.
14. Christoph Grimm, Wilhelm Heupke, and Klaus Waldschmidt. Refinement of Mixed-Signal Systems with Affine Arithmetic. In *Design, Automation and Test in Europe 2004 (DATE 04)*. IEEE Press, Feb 2004.
15. Darius Grabowski, Markus Olbrich, Christoph Grimm, and Erich Barke. Range arithmetics to speed up reachability analysis of analog systems. In *Forum on Specification, Verification and Design Languages 2007*, 2007.
16. Darius Grabowski, Markus Olbrich, and Erich Barke. Analog circuit simulation using range arithmetics. In *ASP-DAC '08: Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 762–767, Seoul, Korea, 2008. IEEE Computer Society Press.
17. R. E. Moore. *Interval Analysis*. Prentice-Hall, Eaglewood Cliffs, NJ, 1966.
18. Alain Vachoux, Christoph Grimm, and Karsten Einwich. Analog and Mixed-Signal Modeling with SystemC-AMS. In *International Symposium on Circuits and Systems 2003 (ISCAS '03)*, Bangkok, Thailand, May 2003. IEEE Press.

19. Darius Grabowski, Christoph Grimm, and Erich Barke. Ein Verfahren zur Effizienten Analyse von Schaltungen mit Parametervariationen. In *Tagungsband GI/ITG/GMM - Workshop Modellierung und Verifikation '03*, Dresden, Mar 2006.
20. D. Grabowski, M. Olbrich, and E. Barke. Ac-analyse analoger schaltungen mit affiner arithmetik. In GMM/ITG, editor, *Analog 2008*, volume Entwicklung von Analogschaltungen mit CAE-Methoden, pages 63–68. VDE, 2008. 6 VG-Wort pages.
21. Florian Schupfer, Michael Kaergel, Christoph Grimm, Markus Olbrich, and Erich Barke. Towards Abstract Analysis Techniques for Range Based System Simulations. In *Forum on Specification and Design Languages 2010 (FDL10)*, page To be published., 2010.