Proceedings of the 2010 Forum on specification & Design Languages

Southampton, UK
September 14th-16th, 2010

General Chair:

Prof. Tom Kazmierski
School of Electronics and Computer Science
University of Southampton

FDL is an ecsi event!

Table of Contents

Towards Abstract Analysis Techniques for Range Based System Simulations

Florian Schupfer[1], Michael Kärgel[2], Christoph Grimm[1], Markus Olbrich[2], Erich Barke[2]
[1]Institute of Computer Technology
Vienna University of Technology
Email: {schupfer,grimm}@ict.tuwien.ac.at
[2]Institute of Microelectronic Systems
Leibniz Universität Hannover
Email: {mk,mo,eb}@ims.uni-hannover.de

***Abstract*—In recent times, range based modeling and simulation techniques have emerged for systems with parameter tolerances and deviations. They are used to perform a semi-symbolic simulation and to analyze the examined systems for their time domain behavior. The system quantities in such simulations are represented as range based signals using the concept of Affine Arithmetic. Transforming the range based signals from a time domain to a frequency domain representation significantly increases the analysis capabilities and provides a broader insight into the system's behavior. Such a transformation enriches the expressiveness of semi-symbolic system quantities and simultaneously allows frequency based analysis techniques to be applicable. We use the Discrete Fourier Transform to compute a range based frequency representation and finally discuss the method and interpretation of the frequency spectrum on two examples.**

I. Introduction

The increased complexity of electronic systems increasingly interferes with the desire to thoroughly verify the system behavior prior to the actual implementation. Currently, time consuming Monte Carlo simulations or sensitivity analysis are performed to get an understanding of the system behavior [1]. Variations and uncertainties of system parameters also increase the parameter space which influences the number of necessary simulation runs. Performing such a simulation with a substantial coverage quality, quickly ends up in a dead end where the simulation time restricts the verification process. Even when the computational effort is manageable, numerical simulations only provide snapshots of the system behavior. They can not provide a formal result, which is not based on an statistical approach. In recent years, a novel class of system simulation methodologies emerged which address these drawbacks. They are referred to as semi-symbolic simulations. That means that classical numerical simulation techniques are extended by symbolic descriptions of specific parameters, which is also descibed in [2]. A commonly used method for a semi-symbolic simulation is based on the concept of Affine Arithmetic [3]. Sets of uncertain values are modeled by sets of ranges enclosing all of the deviations and additionally labeling them with symbols for keeping track of the range correlations throughout the computation process. Exploiting the advantages of semi-symbolic simulation techniques provides a manifold improvement over traditional numeric approaches. Firstly, the formally infinite set of continuous parameter variations can be modeled in a single symbol, reducing the number of necessary simulation runs to one. Secondly, in contrast to numeric simulations, sources of undesired behavior can be tracked back to their origin, giving a better understanding of the system parameter correlations. This is accomplished by labeling intervals by symbols, which allows an identification of the correlated sources throughout the whole simulation process. Thirdly, and most importantly, a semi-symbolic simulation provides a self-contained, formally guaranteed result of the simulation process, giving the full set of possible output values caused by the input stimuli.

This work enhances the basic time domain semi-symbolic analysis techniques to be also applicable in the frequency domain. We utilize the linear nature of traditional system analysis techniques like the Discrete Fourier Transform (DFT) to compute its range based representation and in this way to transform range based signal quantities into the frequency domain.

Section II provides an overview about related work in this field. Section III describes the underlying concepts like Affine Arithmetic and the simulation environment. The performed Fourier Transform technique is presented in Section IV more precisely. Finally Section V gives two examples and shows the capabilities of the presented techniques. Section VI concludes the paper and identifies future work.

II. Related Work

Recently several semi-symbolic system analysis techniques based on Affine Arithmetic have emerged in the academic field [2], [4], [5]. They use ranges, defined by the Affine Arithmetic, to model and simulate parameter uncertainties in conservative and non-conservative systems. One special application is to estimate the optimal bit widths of a system to suffice a given system quality. Errors introduced during quantization, rounding and truncation operations are modeled by ranges and resulting system quantities are calculated [6], [7]. Although the most published work concentrates on the system level also transistor level circuits are simulateable when solving the non-linear differential equations by using

Affine Arithmetic [4]. [8] uses semi-symbolic simulation to analyze the convergence behavior of control loops in presence of uncertainties. [9] and [10] additionally enhance the semi-symbolic simulation for simulating non-linear analog circuits and obtaining refinement information to improve the system quality. The problem of over approximation is addressed in recent works where the Affine Arithmetic is modified to also provide exact results for multiplication operations which avoids the additional approximation terms [11], [12].

III. Analysis Environment

Parameters and system models of technical systems can not always be fully specified from the beginning. They are frequently described as being inside of certain ranges of possible values. In typical design processes such uncertainties are covered in a statistical way by performing Monte Carlo or Worst Case analysis simulations. These analysis techniques not only demand for a high number of simulation runs and consequently consume a high computation power they also rely on approaching the worst cases when using a sufficiently high number of simulation runs. This is where a semi-symbolic simulation provides its full advantage of conservatively including the worst case scenario together with a considerably reduced simulation effort. A convenient way to perform a range based semi-symbolic simulation uses the SystemC AMS modeling and simulation environment [10] for high level and a numerical SPICE-like environment for transistor-level simulations. All necessary modeling structures and simulation engines are provided for describing circuits on all required abstraction levels. The SystemC AMS environment can easily be extended using an Affine Arithmetic library, which overloads the computation related operations, introducing the semi-symbolic methodology. In such environments initially uncertain parameters or quantities will be modeled using ranges instead of single functional values which allows a semi-symbolic computation of the system quantities. The quantities comprise a numerical value which scales the interval and a range symbol which forms the interval itself. This explains the name semi-symbolic simulation, simply referring to the mixed representations.

A. SystemC AMS

SystemC AMS uses C++ based language constructs to model and simulate analog and/or mixed-signal systems [13]. Its main *Model of Computation* (MoC) is *Timed Synchronous Dataflow* (TDF) which is also used for the semi-symbolic simulation. It is a timed version of the original *Synchronous Dataflow* (SDF) which allows to precalculate the schedule of process executions. This characteristic offers a high simulation performance in combination with a powerful modeling expressiveness. On the other hand the C++ based nature of SystemC AMS allows easy integration of additional libraries, like the Affine Arithmetic library in our case. This extensibility makes SystemC AMS an efficient choice for the ever increasing functionality of the range based approach.

B. Transistor level solver

While digital systems of today are well modeled on higher abstraction layers through 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 through SystemC AMS, a verification step on lower levels is often desired to verify the behavior. An affine transistor-level simulator gives the opportunity to use results of an affine system-level simulation as stimuli. This has been developed prior [9] to this paper and the proposed methodologies of this paper have been added. The flow of the given simulator is divided into two parts. At first the given netlist with the corresponding device-models is transformed into a mathematical representation through the well known Modified-Node-Analysis (MNA). This converts the netlist to the according differential equation system (Eqn. 1).

$$F(\underline{x}(t), \underline{\dot{x}}(t), \underline{p(t)}, t) = \underline{0} \tag{1}$$

$\underline{x}(t)$ is the vector of time dependent variables and $\underline{p(t)}$ describes the circuit parameters. This step is performed in Maple, given a SPICE-like netlist and all device models through their symbolic equations. The intermediate result is a complete symbolic equation system. Then 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 C++-Solver, which performs DC, AC [14] and Transient-Simulations [4]. Equations are solved through numerical integration, either forward, backward Euler or trapezoidal.

C. Affine Arithmetic

Affine Arithmetic is a semi-symbolic technique describing ranges by a nominal central value and a superimposed sum of interval valued partial deviations [3]. Affine Arithmetic bases on the original concept of Interval Arithmetic [15] but enhances it with symbolic range identifiers to overcome the dependency problem preventing the usability of the Interval Arithmetic. This correlation enhancement provides Affine Arithmetic the flexibility to be applied in realistic systems. A feedback loop, for instance, is simulateable considering the correlated summation and is not ending up in a considerable overapproximation the Interval Arithmetic would result [10]. For modeling purposes, each affine expression represents the influence of independent sources of uncertainty by a sum of partial deviations $x_i\epsilon_i$. The symbol ϵ_i represents the range $[-1, 1]$ which is scaled by the deviation value x_i. Affine expressions are referred to by $\tilde{}$ in the following.

$$\tilde{x} = x_0 + \sum_{i=1}^{length} x_i\epsilon_i \qquad \epsilon_i \in [-1, 1] \tag{2}$$

$$\tilde{x} \pm \tilde{y} = (x_0 \pm y_0) + \sum_{i=1}^{length} (x_i \pm y_i)\epsilon_i \tag{3}$$

$$c\tilde{x} = cx_0 + \sum_{i=1}^{length} cx_i\epsilon_i \tag{4}$$

Equation 2 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 3 and 4, 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.

IV. Enhanced Analysis

Semi-symbolic simulations provide an important and expressive methodology to analyze the behavior and parameter sensitivity of conservative and non-conservative systems. Studying the interdependencies reduces to a backtracking of the single range contributions to their creating sources. A behavior analysis can be performed by examining the transient simulation which not only shows the behavior for one stimulus but provides a set of results for the whole range. However, the analysis is currently restricted to a time domain view of the system behavior. For a basic, elementary class of systems a time domain simulation based behavior analysis may suffice. When moving towards realistic systems for a wider field of applications, an analysis gap emerges reflecting the time restricted possibilities. Currently available methodologies lack to fully cover the traditional analysis techniques available in the field which also utilize frequency domain representations. This is the focus of our work. As first, but certainly powerful intermediate step, the Discrete Fourier Transform has been enhanced to be processable on range based affine arithmetic forms. The DFT has been chosen because it represents the discrete variant of the widely used Fourier Transform. The Discrete Fourier Transform allows a transition from time to frequency domain and consequently enables a broad field of frequency domain analysis techniques to be applicable on systems modeled by Affine Arithmetic.

A. Traditional Fourier Transform

Basically a Fourier Transform is an operation that transforms complex valued time representatives into their frequency domain counterparts. It is defined for time continuous signals as

$$F(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \tag{5}$$

resulting in a complex valued frequency domain quantity. We prefer the time discrete transformation, the Discrete Fourier Transform, which is more suitable for the simulation technique used. The environment consists of the timed synchronous data flow models for SystemC AMS and variable discrete time-steps in the SPICE-like transistor-level-solver. The following considerations would also identically apply to the time continuous transformation which indicates no loss of generalization when using the discrete operation.

$$F[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k}{N}n} \qquad k = 0, ..., N-1; \tag{6}$$

B. Range based Fourier Transform

To allow the transformation operation to be applicable on affine forms, we expand the Discrete Fourier Transform to handle Affine Arithmetic symbols as in Equation 7. Because the DFT is a linear operation the transformation of the Affine Arithmetic symbols simply splits up in a frequency domain superposition of the transformed nominal and partial deviation parts.

$$\tilde{F}[k] = \sum_{n=0}^{N-1} \left(x_0[n] + \sum_{i=1}^{length} x_i[n]\epsilon_i[n] \right) e^{-j\frac{2\pi k}{N}n} \tag{7}$$
$$k = 0, ..., N-1;$$

Equation 7 shows the structure of the implicit Discrete Fourier Transform operation. The input symbol split up into their nominal values x_0 and the appertaining number of i partial deviations $x_i\epsilon_i$. For better representation of the affine transformation, the number of partial deviations is reduced to one in this abstract. This assumption does not restrict the transformation process itself but allows to focus on the range related issues. The number of deviations is not limited in the results shown in the following chapters. Expanding the equation simplifies the calculation to a sum of two Fourier transforms, one giving the transformation operation of the nominal value and the second one giving the frequency domain representation of the partial deviation.

$$\tilde{F}[k] = \sum_{n=0}^{N-1} x_0[n] e^{-j\frac{2\pi k}{N}n} + \sum_{n=0}^{N-1} x_1[n]\epsilon_1[n] e^{-j\frac{2\pi k}{N}n} \tag{8}$$
$$k = 0, ..., N-1;$$

Solving this equation allows a calculation of the generalized frequency spectrum of Affine Arithmetic symbols during a semi-symbolic simulation. The generalization refers to the $\epsilon[n]$ symbols. In this description they are time dependent, which means they can hold every value within this interval, independently from its predecessor or successors in time. This behavior perfectly corresponds with the idea of Affine Arithmetic where the partial deviations represent a range of allowed values. The ranges are considered to model the area in where the possible resulting signal values are expected to reside in. Solving Equation 8 is performed in two consecutive steps. First, the frequency spectrum of the nominal value is computed. This correlates to the traditional DFT and will be superimposed later on with the range based contribution. The second part of the transformation equation treats the range symbols. The ϵ symbols are considered as time dependent $\epsilon[n]$ which indicates an uncorrelated characteristic of these symbols. Accordingly the $x_1[n]\epsilon_1[n]$ have to be summarized independently, which means composing the uncorrelated, Interval Arithmetic like, range result. This results in a substantial widening of the frequency based range interval which is also scaled by the number of discrete samples to be transformed. A typical Discrete Fourier Transform of 2048 points already widens the ranges that much that reliable

statements about the frequency domain behavior can not be made any more.

One measure to allow a meaningful frequency domain representation of range based signals is to restrict the ϵ symbol to being time independent. The partial deviation envelopes the signal anyway, but the imagined realization which would be a specific value inside the range, stays constant over time. We explain this property more detailed in the next section, at this point we concentrate on the derivation of the corresponding transformation operation. Restricting the transformation to the time independent ϵ_1 case, results in the simplified Discrete Fourier Transform given by:

$$\tilde{F}[k] = \sum_{n=0}^{N-1} x_0[n] e^{-j\frac{2\pi k}{N}n} + \epsilon_1 \sum_{n=0}^{N-1} x_1[n] e^{-j\frac{2\pi k}{N}n} \quad (9)$$

$$k = 0, ..., N-1;$$

All ϵ_1 symbols in time can be treated as correlated and therefore the partial deviation contribution reduces to a sum over the deviation transformed by the scalar multiplication of the exponential function. For simplification the ϵ_1 is moved in front of the sum, which illustrates the remaining computation. As a result of such a range based transformation we get the Fourier Transform of the nominal signal, superimposed by the frequency representation of the partial deviation. Having a Discrete Fourier Transform being applicable on range based Affine Arithmetic forms provides a powerful analysis technique often used in system theory and signal processing applications.

C. Applied range based Fourier Transform

Technical systems are usually modeled by creating partial deviations for every source of uncertainty. Whenever a range symbol is created it is labeled by a symbol which identifies the range for the further operations. These partial deviations are basically divided into two groups, static and dynamic deviations. Static uncertainties represent time independent parameters like production tolerances and are modeled in the simulation environment to add a constant deviation to a signal. In contrast dynamic deviations model time dependent behavior, like a quantization error. The uncertainty is different at every simulation time point and is introduced by creating a new deviation symbol at every point in time. This modeling strategy preserves the source correlation of the symbols as for instance a quantization operation adds uncertainty to the system every time it is performed. In the last section it was defined that a Discrete Fourier Transform is applicable in its simplified form, when the single ϵ symbols represent time independent deviations. For most of our modeled systems this assumption can be proved. A time independence is the source of our technical modeling strategy to allow operations on correlated intervals. Thus, identical intervals are correlated even when delayed in time by the system model.

D. Runtime of range based Fourier Transform

Using DFT in combination with Affine Arithmetic has a huge impact on runtime dependency. While small DFTs (e.g. 2^{10} point) are still computable in decent time on modern CPUs using e.g. double precision floating point values, the additional runtime caused by affine symbols makes it hardly applicable anymore. The runtime complexity for a non optimized DFT is $O(n^2)$, but through the use of Affine Arithmetic each operation actually consists internally of up to $m+1$ operations for m deviation symbols. This results in a runtime complexity of $O(n^2 \cdot m^2)$, causing also a high dependency on the number of deviation symbols and through that on the applicability on circuits. While on system-level the expected number of deviation symbols is quite small, even small transistor-level circuits will have a few thousand deviation symbols through the simulation methodology. Thus, the number of symbols increases the runtime similar to increasing the number of transformation points. In order to reduce this effect an FFT is essential, having a runtime of about $O(n \cdot log(n))$. With this measure not only the dependency on the DFT depth is decreased but also the influence of the circuit size on the computation performance can be reduced.

V. Fourier Analysis Demonstration

For demonstrating the applicability of the Discrete Fourier Transform deduced in Section III semi-symbolic simulation examples are presented in this section. As a semi-symbolic simulation can be applied on different levels of abstraction its usage is shown firstly on system level and subsequently on transistor level. To generally motivate the necessity of a Fourier Transform a widely used mixer system e.g. for use in a software defined radio, has been chosen.

A mixer model on system level reduces 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. Our example illustrates a down conversion in a communication receiver and therefore only the lower sideband is of relevance.

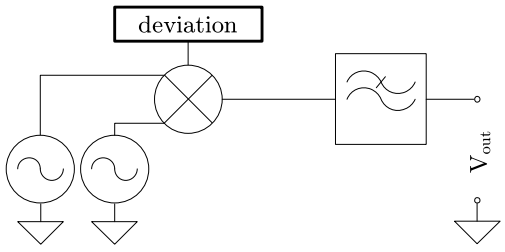

Fig. 1. Mixer with deviations on system level

For enhancing the system model to additionally handle ranges of imprecise parameters we added a block called “deviation“ to the model description. The possibilities for such a deviation definition are numerous. When using semi-symbolic simulations, they are based on Affine Arithmetic. 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 system simulation is finally transformed into the frequency domain by applying the Discrete Fourier Transform on the range based output quantity V_{out}.

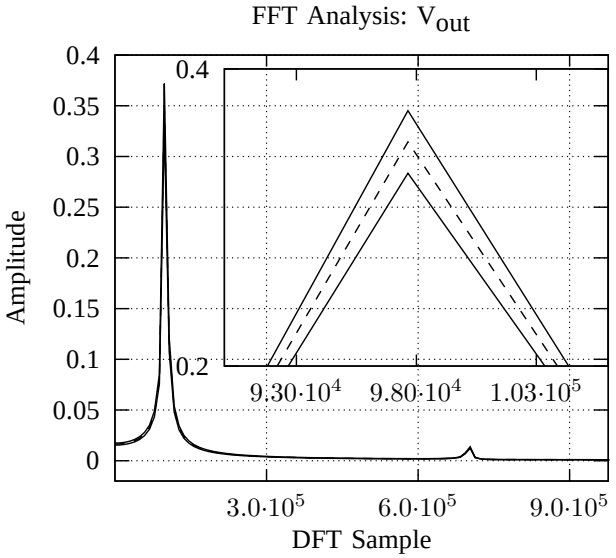

Fig. 2. Amplitude spectrum of mixer output on system level

The resulting frequency spectrum shown in Fig. 2 gives the spectral components remaining in the system model output signal. The spectrum does not only provide the frequency behavior of the nominal system model but it also delivers an envelope which forms the boundaries of the range defined system behavior.

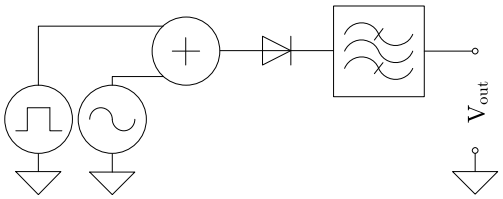

Fig. 3. Simulated transistor level mixer circuit

The second example circuit is a mixer as shown in Fig. 3, which is often used as a frequency selection stage in communication receivers. The circuit was built on transistor-level from an inverting adder as input stage, a diode as non-linearity, followed by a buffer stage and a passive bandpass filter. The parameters had tolerances of 1 % to 10 % including width and length of the transistors as well as the resistors, capacitors and inductors. These tolerances were expressed directly during the modeling process as deviations of the parameters using affine symbols, e.g. $R_1 = 10\ k\Omega + 1\ k\Omega \cdot \epsilon_{R1}$. The result of the transient simulation was used as input for the range based FFT. In order to avoid deviations through the settling time of the mixer's output filter the simulation runtime was chosen to be 1 ms, which holds 1000 cycles of the slowest input signal. Only the last 0.1 ms of the resulting output signal have been used as input for the FFT, which is plotted in Fig. 4.

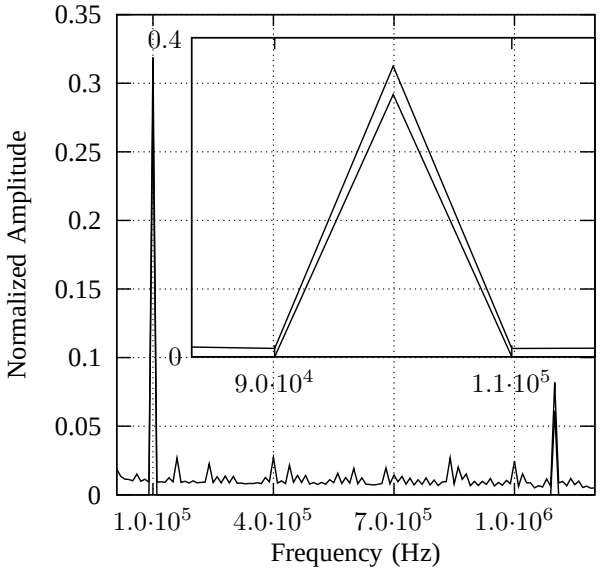

Fig. 4. Amplitude spectrum of mixer output on transistor level

The solid lines represent the upper and lower bounds of the transformed affine output signal. The input frequencies of 1 MHz and 1.1 MHz are well suppressed, resulting in a high amplitude of the down sampled 100 kHz signal. As low-level simulation includes solving a highly non-linear differential equation system, many new deviation symbols are generated during runtime. Each call of a non-linear arithmetic function creates one new deviation symbol. This causes generation of about one million symbols in total. Because at each time step new deviations are generated, these uncorrelated symbols create frequencies throughout the whole spectrum. This again results in a high FFT-noise, which is shown in Fig. 4.

While the DFT and FFT are completely linear Operations, the illustration of the result may not. Using a representation in the complex plane plots the output of the FFT directly, not introducing any further approximation. But it is not very commonly used. The most common representation through magnitude and phase introduces further non-linear functions. Especially the magnitude, which calculated through $\sqrt{re^2 + im^2}$, shows a issue in the range expression. Using Affine Arithmetics, it cannot be guaranteed, that $re^2 + im^2$ is always positive, which means that the square root may not be computable. To avoid this, at first the complex values are converted back to their interval expressions. In this expression the absolute value is calculated. If both bounds are greater or smaller than zero, the new bounds are their absolute values. If one bound is greater than zero while the other is smaller than zero, the lower bound is zero and the upper bound is the greatest absolute value. The magnitude is then calculated using the new intervals.

The simulation was conducted on a 16-core AMD Opteron system with 2.8 GHz and 32 GB RAM. The verification of the simulation results as well as of the FFT has been conducted

through a nominal simulation using 1000 Monte-Carlo samples. The runtime of the affine simulation and FFT required 14:59 minutes compared to 549:14 minutes for the Monte-Carlo-simulation. A total amount of around 19 GB RAM was required, mainly caused by the high amount of deviations symbols. As worst case each affine value contains all deviation symbols resulting in a total amount of required RAM to store only the values of the deviations of $N_{FFTpoints} \cdot N_{Deviations} \cdot length(deviations) = 2048 \cdot 10^6 \cdot 8\ Byte \approx 16GB$.

VI. Conclusion and Future Work

In this paper, we introduced a methodology to apply a Discrete Fourier Transform on range based system simulations. Semi-symbolic simulations are used to model and simulate range based system descriptions. Initially undefined or varying system parameters are modeled as ranges by using the mathematical concept of Affine Arithmetic. Currently a semi-symbolic simulation is restricted to the time domain. When postulating a time invariance of the interval symbol ϵ an efficient and meaningful Fourier Transform can be calculated. We demonstrated our work by two examples modeling a mixer circuit on system level as well as on transistor level. Although the introduced deviations are not identical on both levels the frequency spectrum calculated is comparable. For future work several subjects for improvement have been identified. The vast number of partial deviations in realistic systems limits the applicability of the methodology. A garbage collection introduced in [16] would address this restriction. Currently the circuit simulation on system and transistor level is performed by two independent simulation environments. An integration of the transistor level solver into the SystemC AMS environment is subject of current research and will provide the advantage of a full mixed level simulation environment.

Acknowledgment

This work has been supported by the Austrian WWTF project MARC under contract no. ICT08_012 and the German BMBF under project no. 01 M 3087.

References

[1] R. Rubinstein, *Simulation and the Monte Carlo Method*. New York, NY, USA: John Wiley&Sons, Inc., 1981.

[2] W. Heupke, C. Grimm, and K. Waldschmidt, "Semi-symbolic simulation of nonlinear systems," in *Forum on Specification and Design Languages 2005 (FDL05)*. Lausanne: ECSI, Sep 2005.

[3] M.Andrade, J.Comba, and J.Stolfi, "Affine arithmetic (extended abstract)," in *INTERVAL '94*, St. Petersburg, Russia, 1994.

[4] D. Grabowski, M. Olbrich, and E. Barke, "Analog circuit simulation using range arithmetics," in *ASP-DAC '08: Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. Seoul, Korea: IEEE Computer Society Press, 2008, pp. 762–767.

[5] D. Grabowski, C. Grimm, and E. Barke, "Semi-Symbolic Modeling and Simulation of Circuits and Systems," in *International Symposium on Circuits and Systems 2006 (ISCAS 2006)*. Kos, Greece: IEEE Press, May 2006.

[6] C. F. Fang, T. Chen, and R. A. Rutenbar, "Floating-Point Error Analysis Based On Affine Arithmetic," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, 2003, pp. 561–564.

[7] J. A. Lopez, C. Carreras, and O. Nieto-Taladriz, "Improved Interval-Based Characterization of Fixed-Point LTI Systems With Feedback Loops," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, vol. 26, no. 11, pp. 1923–1933, 2007.

[8] C. Grimm, W. Heupke, and K. Waldschmidt, "Analysis of Mixed-Signal Systems with Affine Arithmetic," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, vol. 24, no. 1, pp. 118– 123, 2005.

[9] D. Grabowski, C. Grimm, and E. Barke, "Ein Verfahren zur Effizienten Analyse von Schaltungen mit Parametervariationen," in *Tagungsband GI/ITG/GMM - Workshop Modellierung und Verifikation '03*, Dresden, Mar 2006.

[10] C. Grimm, W. Heupke, and K. Waldschmidt, "Refinement of Mixed-Signal Systems with Affine Arithmetic," in *Design, Automation and Test in Europe 2004 (DATE 04)*. IEEE Press, Feb 2004.

[11] H. Shou, H. Lin, R. Martin, and G. Wang, "Modified Affine Arithmetic Is More Accurate than Centered Interval Arithmetic or Affine Arithmetic," in *Martin(Eds.), Lecture Notes in Computer Science 2768, Mathematics of Surfaces, Springer-Verlag*. Springer, 2003, pp. 355–365.

[12] F. Messine and A. Touhami, "A General Reliable Quadratic Form: An Extension of Affine Arithmetic," *Reliable Computing*, vol. 12, no. 3, pp. 171–192, 2006.

[13] A. Vachoux, C. Grimm, and K. Einwich, "Analog and Mixed-Signal Modeling with SystemC-AMS," in *International Symposium on Circuits and Systems 2003 (ISCAS '03)*. Bangkok, Thailand: IEEE Press, May 2003.

[14] D. Grabowski, M. Olbrich, and E. Barke, "Ac-analyse analoger schaltungen mit affiner arithmetik," in *Analog 2008*, GMM/ITG, Ed., vol. Entwicklung von Analogschaltungen mit CAE-Methoden. VDE, 2008, pp. 63–68, 6 VG-Wort pages.

[15] R. E. Moore, *Interval Analysis*. Eaglewood Cliffs, NJ: Prentice-Hall, 1966.

[16] W. Heupke, C. Grimm, and K. Waldschmidt, *Advances in Specification and Design Languages for SoC*. Springer-Verlag, 2006, ch. Modeling Uncertainty in Nonlinear Systems with Affine Arithmetic, pp. 198–213.