Semi-Symbolic Analysis of Mixed-Signal Systems including Discontinuities

Carna Radojicic, Christoph Grimm, Javier Moreno and Xiao Pan
Kaiserslautern University of Technology
{radojicic, grimm, moreno, pan}@cs.uni-kl.de

***Abstract*—The paper describes an approach for semi-symbolic analysis of mixed-signal systems that contain discontinuous functions, e.g. due to modeling comparators. For modeling and semi-symbolic simulation, we use extended Affine Arithmetic. Affine Arithmetic is currently limited to accurate analysis of linear functions and mild non-linear functions, but not yet discontinuities. In this paper we extend the approach to also handle discontinuities. For demonstration, we symbolically analyze a $\Sigma\Delta$-modulator.**

I. Introduction

Embedded systems include an increasing number of mixed-signal circuits. Verification is a challenge, because system properties result from interaction of analog circuits with parameter variations and SW/DSP systems that compensate these variations. For verification of corner cases multi-run simulation techniques like Monte-Carlo (e.g. [1]) and Worst Case analysis (e.g. [2], [3]) are used. However, they often require a very high number of simulation runs to validate all corners.

Formal methods for analog or mixed-signal systems are a rather new field of intensive research that could offer interesting alternatives to multi-run simulations. Formal approaches for analog/mixed-signal circuits and systems allow equivalence checking [4], model checking [5], or symbolic analysis [6] deal with pure continuous behavior of analog circuits. At a much higher level of abstraction, Henzinger [7] suggests modeling hybrid systems with linear hybrid automata on which algorithmic analysis can be performed. However, this method is limited to very abstract and simple systems.

A less formal approach is semi-symbolic simulation [8] based on Affine Arithmetic [9]) that we use in this work. For simulation we use (symbolic) affine expressions instead of real values. Affine terms describe the impact of different sources of uncertainty/deviation in a symbolic way. The method has two advantages:

1) Modeling parameter uncertainties and deviations as ranges allows us to simulate all possible corner cases in a single simulation run
2) Representation of uncertainties with deviation symbols allows clear traceability of a violation in system performances back to contributions of corners of (e.g. technological) parameters.

In previous work, the semi-symbolic simulation has been used successfully with block diagram level models [8], and even circuits [10]. Affine Arithmetic is also used in static analysis of floating point errors of DSP algorithms [11]. A problem not yet solved within such semi-symbolic simulation is the transition between discrete and continuous behavior. In [8], an ADC between discrete controller and continuous plant is modeled by adding quantization noise to a linear transfer function. Modeling/simulation of discontinuous functions with considering two (or more) separate cases was not possible. In this work we extend semi-symbolic simulation to handle discontinuities in such a way. In Section II we describe simulation based on Affine Arithmetic Forms (AAF). In Section III we describe a new method for modeling/simulation of discontinuities. In Section IV we demonstrate applicability by analysis of a $\Sigma\Delta$-modulator.

II. Semi-Symbolic Simulation with AAF

Affine Arithmetic [9] allows accurate computation with ranges. In each affine form, the influence of uncorrelated sources of uncertainty i to a value with the 'ideal', central value x_0 is represented by a sum of terms $x_i\varepsilon_i$. Noise (also:deviation) symbols ε_i are unknown values from the interval $[-1, 1]$ that are scaled by partial deviations x_i:

$$\tilde{x} = x_0 + \sum_{i\in\mathcal{N}_{\tilde{x}}} x_i\varepsilon_i \qquad \varepsilon_i \in [-1, 1]\,.$$

$\mathcal{N}_{\tilde{x}}$ represents a set of natural numbers identifying deviation terms $x_i\varepsilon_i$ in $\tilde{x}$. Linear operations are accurate, because noise terms keep correlation information. However, non-linear operations introduce more or less over-approximation that guarantees safe inclusions.

For semi-symbolic analysis of circuits and systems, we model uncertainties and parameter variations of circuits and systems such as noise, aging, drift, offsets, etc. by noise symbols ε_i scaled by x_i. This gives us a symbolic description of all parameters that influence the traces for a given stimuli and parameter ranges. Fig. 1 for example shows semi-symbolic

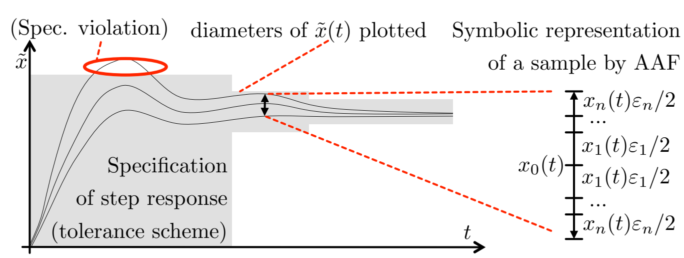

Fig. 1: Semi-symbolic simulation results and it's visualization by plotting diameters of signals.

978-3-9815370-2-4/DATE14/©2014 EDAA

representation of the step response of a control loop. Output of simulation is a symbolic representation of the output signals for a given input stimuli (i.e. step function) representing all parameter combinations. It consists of the ideal output values, and terms $x_i\varepsilon_i$ that - for each time step - describe contributions of uncertainties. The symbolic output can be visualized as shown in Figure 1 by setting ε_i to 1 resp. -1.

III. MODELING DISCONTINUITIES WITH AAF

To handle discontinuities, we split and merge non-contiguous range-based signals and states (spaces for multiple signals/states) . Let $\tilde{y}$ be a sample of a signal represented by AAF (informally, a range). When the signal is compared with a threshold v_{th} which represents a border between two consecutive discontinuous values, we have to consider three cases:

1) The whole signal range is below v_{th}
2) One part of the signal range is below v_{th}, and the other above v_{th}. Hence, we must take both possible results into account.
3) The whole signal range is above v_{th}

Cases (1) and (3) require no changes, because the result remains contiguous. In case (2) we must split system behavior, and handle both cases separately in the following simulation.

In case (2), output is first computed considering a value of discontinuous signal which corresponds to the case for which $\tilde{y} \geq v_{th}$. In the following, we name this case *above*. Then, the case for which $\tilde{y} < v_{th}$ must be considered and the output for this value must be computed. We name this case *below*. Both cases are handled currently by separate semi-symbolic simulation runs.

Implementation note: For implementation, we handle both cases separately by repeating time steps in SystemC AMS. Both computations cannot be performed at the same time step of simulation. However, SystemC AMS 2.0 Dynamic Timed Data Flow (DTDF) [12] model of computation allows repeating a time step, e.g. to handle crossing of borders. We (mis-)use this feature of SystemC-AMS 2.0 to allow us to repeat time steps by calling the function *request_next_activation(SC_ZERO_TIME)* with zero time step as an argument. A zero time step repeats the time step at which the value for *below* case was written to the output, but we use input for *above* case to compute the output.

When both outputs are available they can be merged in order to generate a single result represented by affine terms. The idea used to merge both outputs is described below.

In the following the symbols $\tilde{y}_a$ and $\tilde{y}_b$ are used to assign the outputs computed for *above* and *below* discontinuous signal values, respectively. Since deviations of system parameters are modeled as affine terms, $\tilde{y}_a$ and $\tilde{y}_b$ are also affine terms:

$$\tilde{y}_a = y_{a0} + \sum_{i \in \mathcal{N}_{\tilde{y}_a}} \varepsilon_i y_{ai} \qquad \varepsilon_i \in [-1, 1]$$

and

$$\tilde{y}_b = y_{b0} + \sum_{i \in \mathcal{N}_{\tilde{y}_b}} \varepsilon_i y_{bi} \qquad \varepsilon_i \in [-1, 1].$$

The basic idea to merge $\tilde{y}_a$ and $\tilde{y}_b$ is to treat them as bounds of a range which is also modeled with Affine Arithmetic. Hence, the following holds:

$$[\tilde{y}_b, \tilde{y}_a] = \tilde{y}_c + \varepsilon_{\mathcal{N}_{\tilde{y}}+1} \tilde{y}_{diam} \qquad \varepsilon_{\mathcal{N}_{\tilde{y}}+1} \in [-1, 1] \tag{1}$$

where $\tilde{y}_c$ represents the central value of the range $[\tilde{y}_b, \tilde{y}_a]$ and it is calculated as:

$$\tilde{y}_c = \frac{(\tilde{y}_a + \tilde{y}_b)}{2} = \frac{y_{a0} + y_{b0}}{2} + \sum_{i \in \mathcal{N}_{\tilde{y}}} \varepsilon_i (\frac{y_{ai} + y_{bi}}{2}).$$

The parameter $\tilde{y}_{diam}$ is the diameter of the range $[\tilde{y}_b, \tilde{y}_a]$ and $\mathcal{N}_{\tilde{y}}$ is the set of deviation symbols of the merged output ($\mathcal{N}_{\tilde{y}} = \mathcal{N}_{\tilde{y}_b} \cup \mathcal{N}_{\tilde{y}_a}$).

The diameter of the range is calculated as:

$$\tilde{y}_{diam} = \frac{|\tilde{y}_a - \tilde{y}_b|}{2} = \frac{1}{2} \left| y_{a0} - y_{b0} + \sum_{i \in \mathcal{N}_{\tilde{y}}} \varepsilon_i (y_{ai} - y_{bi}) \right|.$$

Note that since for calculation of $\tilde{y}_{diam}$ the absolute value of $\tilde{y}_a - \tilde{y}_b$ is used, it will not effect the merged result either $\tilde{y}_b$ and $\tilde{y}_a$ are treated as lower and upper bounds of the range $[\tilde{y}_b, \tilde{y}_a]$ or via versa.

One can note that in (1) multiplication of the parameter $\tilde{y}_{diam}$ with $\varepsilon_{\mathcal{N}_{\tilde{y}}+1}$ will beside affine terms also contain quadratic ones. Hence in order to compute the merged result using pure affine terms, $\tilde{y}_{diam}$ will be over approximated by the following:

$$\tilde{y}_{diam} = \frac{|\tilde{y}_a - \tilde{y}_b|}{2} \subseteq [0, l].$$

The value l represents maximum of $\frac{|\tilde{y}_a - \tilde{y}_b|}{2}$ and is calculated as:

$$l = max(\frac{|\tilde{y}_a - \tilde{y}_b|}{2}) = \frac{1}{2}(|y_{a0} - y_{b0}| + \sum_{i \in \mathcal{N}_{\tilde{y}}} |y_{ai} - y_{bi}|)$$

since it holds:

$$\begin{aligned} |\tilde{y}_a - \tilde{y}_b| &= \left| y_{a0} - y_{b0} + \sum_{i \in \mathcal{N}_{\tilde{y}}} \varepsilon_i (y_{ai} - y_{bi}) \right| \\ &\leq |y_{a0} - y_{b0}| + \sum_{i \in \mathcal{N}_{\tilde{y}}} |y_{ai} - y_{bi}|. \end{aligned}$$

The deviation symbol $\varepsilon_{\mathcal{N}_{\tilde{y}}+1}$ in (1) is a new deviation symbol added to ensure the conservativeness of the final result and has a value lying in interval $[-1, 1]$. The actual value of this symbol strongly depends on the values of deviation symbols modeling uncertainties. This dependency is for every new time step taken into account and hence if system behaviors computed at current and previous time steps share in complete deviation symbols, the same deviation symbol modeling over approximation is added. On the other side, if there is independency between system behaviors, an over approximation term will contain new deviation symbol introduced to model noncorrelation.

IV. CASE STUDY: THIRD-ORDER $\Sigma\Delta$ MODULATOR

For proof-of-concept we analyze a third-order $\Sigma\Delta$ modulator shown by Fig. 3. The objective is to get worst-case and best-case performance figures, and ability to symbolically reason about impact of variations, noise, integrator saturation, etc. For simplicity, we focus on (simple) noise and process variations of capacities whose impact and partial cancellation is well-known, but that shall become visible in outputs of semi-symbolic simulation. Adding more sophisticated models, variances and deviations is straightforward.

A. Modeling the one bit quantizer

For semi-symbolic modeling/simulation we use basically SystemC AMS simulation environment. All modules are implemented using SystemC AMS TDF (Timed Data Flow). The implementation of the quantizer, which simply compares its input signal with a specified threshold, is extended to support range-based signals using DTDF as it is explained in Section III.

B. Modeling the SC integrator

Integrators in the Fig. 3 are implemented by a switched-capacitor (SC) circuit. For simplicity, we model it using a discrete-time integrator composed of a delay module and a module whose gain is equal to an integrator gain. As the integrator gets input from a one-bit quantizer that maps noncontiguous ranges/states to emulated multi-run simulation (repeating time step), in this place we must also support multi-run simulations. Therefore, the integrator model has an internal state that must be reseted to the old state and recomputed for repeated time step. For the integrator, the saturation effect is implemented as a nonideal behavior. The integrator finally contains a merge module at its output to map the noncontiguous affine outputs to a contiguous representation using AAF (see Fig. 2). The integrator gain b from Fig. 2 is

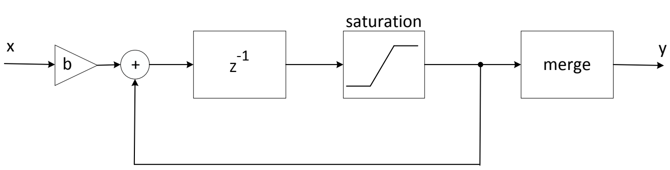

Fig. 2: Block level representation of modulator integrator including merge module

equal to capacitor ratio C_2/C_1; process variations are mostly cancelled because they are partially correlated. The values of integrator capacitors of the third order $\Sigma\Delta$-modulator are given in Table I. Tolerances of capacitor values are modeled by deviation terms. It is supposed that 30% tolerance and 5% of C_2 value are correlated and noncorrelated with C_1 value,

TABLE I: Modulator parameters

Integrator	$C_2[pF]$	$C_1[pF]$
1st Integrator	$0.07333 + \varepsilon_1 30\% + \varepsilon_2 5\%$	$1 + \varepsilon_1 30\% + \varepsilon_3 5\%$
2nd Integrator	$0.2881 + \varepsilon_1 30\% + \varepsilon_2 5\%$	$1 + \varepsilon_1 30\% + \varepsilon_3 5\%$
3rd Integrator	$0.7997 + \varepsilon_1 30\% + \varepsilon_2 5\%$	$1 + \varepsilon_1 30\% + \varepsilon_3 5\%$

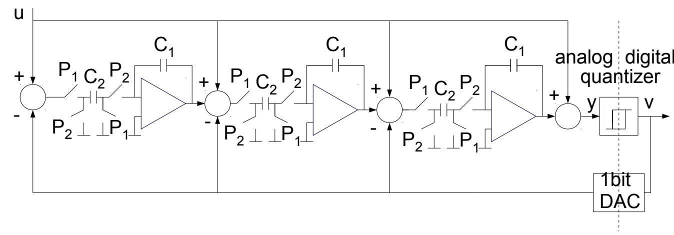

Fig. 3: Third order SC $\Sigma\Delta$-modulator

respectively. Hence, correlated values are modeled such that they share deviation symbols, while noncorrelation is modeled using different deviation symbols for each parameter deviation.

C. Modeling the integrator saturation

In case of single signal samples the implementation of saturation module is simple. However, we use signal samples that are Affine Arithmetic forms. Hence, we cannot simply saturated a single value, since one part signal range can be out of upper and one out of lower saturation value. The affine representation of range-valued signal which may be saturated is as following:

$$\tilde{x} = x_0 + \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i \varepsilon_i \quad \varepsilon_i \in [-1, 1]$$

where x_0 is the nominal signal value and $\mathcal{N}_{\tilde{x}}$ represents a set of natural numbers identifying deviation terms $x_i \varepsilon_i$ in the signal value. Firstly, assume that $\forall \varepsilon_i \in [-1, 1]$ the whole range of signal $\tilde{x}$ lies above upper $\tilde{x} > V_{sathigh}$ or below lower saturation value $\tilde{x} < V_{satlow}$. In these cases saturation procedure is trivial and requires $\tilde{x}$ to be saturated to one of saturation values $V_{sathigh}$ or V_{satlow}. The second case to be considered is the one for which it holds that $\exists \varepsilon_i \in [-1, 1]$ such that $\tilde{x}$ is below V_{satlow} and(or) above $V_{sathigh}$ and via versa. In this case the affine term needs to be modified such that new bounds of $\tilde{x}$ are saturated to $V_{sathigh}$ or V_{satlow} or both of them depending which one is exceeded. For example, if both bounds are exceeded, $\tilde{x}$ is modified such that new bounds are $[V_{satlow}, V_{sathigh}]$. For this purpose, new deviation symbol modeling saturation is introduced:

$$\begin{aligned} \tilde{x}^{(sat)} &= x_0 + \sum_{i \in \mathcal{N}_{\tilde{x}}} x_i \varepsilon_i + \\ &+ ((V_{satlow} + V_{sathigh})/2 - x_0) + \varepsilon_{\mathcal{N}_{\tilde{x}}+1} x_{\mathcal{N}_{\tilde{x}}+1} \end{aligned}$$

where the numerical value $x_{\mathcal{N}_{\tilde{x}}+1}$ scaling new deviation symbol $\varepsilon_{\mathcal{N}_{\tilde{x}}+1}$ is equal to:

$$x_{\mathcal{N}_{\tilde{x}}+1} = \frac{(V_{sathigh} - \tilde{x}_{high}) - (V_{satlow} - \tilde{x}_{low})}{2}.$$

The values $\tilde{x}_{high}$ and $\tilde{x}_{low}$ represent upper and lower bounds of a range represented with affine term $\tilde{x}$:

$$\tilde{x}_{high} = x_0 + \sum_{i \in \mathcal{N}_{\tilde{x}}} |x_i|$$

$$\tilde{x}_{low} = x_0 - \sum_{i \in \mathcal{N}_{\tilde{x}}} |x_i| \,.$$

D. Semi-symbolic analysis of FFT

As an input signal a sinusoid of frequency $f = 3.9kHz$ and amplitude $600mV$ is considered. Fast Fourier Transformation (FFT) of the range-based modulator output was calculated at $N = 8192$ points with the sampling rate $T_s = 0.125us$. For this purpose the extended version of FFT [13] supporting range-based signals is used. The modulator was simulated for $8192 * 0.125us = 1024us$ and simulation run including FFT calculation took $110s$.

The symbolic amplitude spectrum of the modulator output v including quantization error is shown in Fig. 4. The symbolic representation of the amplitude spectrum for each frequency consists of the nominal amplitude values $v_0(f)$ and the terms $v_i(f)\varepsilon_i$ that are superimposed to the nominal value and represent different sources of uncertainties to the final output. For example at $f = 3.90625kHz \simeq 3.9kHz$ a symbolic output a as function of deviation symbols $\varepsilon_1, \varepsilon_2, \varepsilon_3$ is equal to $a = 0.604032 + \varepsilon_1 0 + \varepsilon_2 0.013476 - \varepsilon_3 0.013476.....$ Beside $\varepsilon_1, \varepsilon_2$ and ε_3 the symbolic output also contains the other deviation symbols added by nonlinear operations such as calculation of capacitor ratio C_2/C_1, then merge operation, saturation modeling and finally quantization noise. By setting all deviation symbols ε_i to -1 and 1 the upper and lower bounds of amplitude spectrum can be visualized (blue and red line in Fig. 4). In order to show that lower and upper bounds are safe inclusions of precise result, the impact of one set of deviation terms $v_i(f)\varepsilon_i$ together with nominal amplitude values is extracted and shown in the same figure.

Quantization noise is a dynamic uncertainty which represents a sequence of statistical uncorrelated values with 0 mean value and variation $\sigma^2 = \frac{Q^2}{12}$. Hence, it must be modeled by using uncorrelated noise symbols as it is described in [8]:

$$noise(\tilde{y}, \sigma) = \tilde{y} + \gamma[n]\sigma$$

where γ is uncertainty representing Gauss distribution and σ is standard deviation equal to the square root of variation.

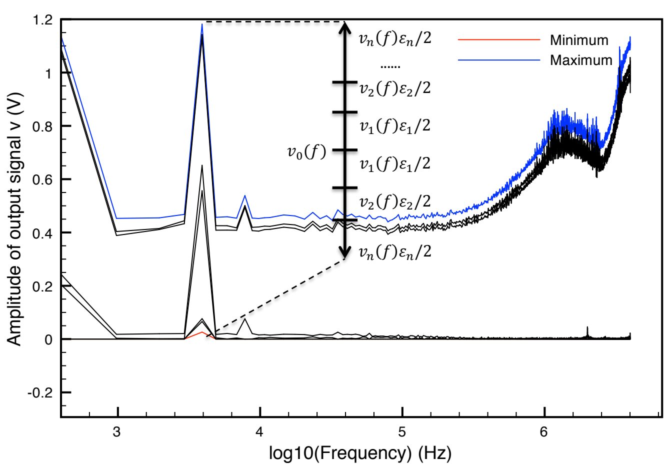

Fig. 4: Amplitude spectrum of the output signal v

V. Conclusion

We have shown a first approach that enables handling functions with discontinuities such as quantizers in Affine Arithmetic. This enables application of symbolic methods to mixed-signal circuits that had not been possible before: circuits that contain comparators, or limitation such as the $\Sigma\Delta$-modulator analyzed for demonstration.

Compared with conventional single-valued simulation there are still some open issues and limitations:

- First of all, the discrete domain is still limited to manually modeling the possible simulation runs. Symbolic simulation at one hand might provide much more general and powerful results; at the other hand we might loose ease-of-use as in the current SystemC AMS implementation.
- We limited ourself to block diagram level; although semi-symbolic circuit simulation has been implemented. This is rather a tradeoff between simulation performance and accuracy than a real limitation of the approach.

References

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

[2] 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*, vol. 4(3), pp. 261–282, 1993.

[3] M. Rafaila, C. Grimm, C. Decker, and G. Pelz, "Sequential Design of Experiments for Effective Model-Based Validation of Electronic Control Units," *e&i Elektrotechnik und Informationstechnik*, vol. 127, pp. 164–170, 2010.

[4] L. Hedrich and E. Barke, "A Formal Approach to Nonlinear Analog Circuit Verification," in *Proceedings of the 1995 IEEE/ACM International Conference on Computer-Aided Design*, 1995, pp. 123–127.

[5] W. Hartong, L. Hedrich, and E. Barke, "Model Checking Algorithms for Analog Verification," in *Proceedings of the 39th annual Design Automation Conference*, 2002, pp. 542–547.

[6] G. Gielen, *Trade-Offs in Analog Circuit Design.* Springer US, 2002, ch. Techniques and Applications of Symbolic Analysis for Analog Integrated Circuits, pp. 953–984.

[7] T. Henzinger and H. Wong-Toi, "Linear Phase-Portrait Approximations for Nonlinear Hybrid Systems," *In Hybrid Systems III: Verification and Control*, vol. LNCS 1066, pp. 377–388, Springer, 1996.

[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] M. Andrade, J. Comba, and J. Stolfi, "Affine Arithmetic (extended abstract)," *Interval '94, St.Petersburg, Russia*, 1994.

[10] D. Grabowski, M. Olbrich, and E. Barke, "Analog Circuit Simulation Using Range Arithmetics," in *Proceedings of the 2008 Asia and South Pacific Design Automation Conference (ASP-DAC '08).* Seoul, Korea: IEEE Computer Society Press, 2008, pp. 762–767.

[11] C. F. Fang, R. A. Rutenbar, M. Püschel, and T. Chen, "Toward Efficient Static Analysis of Finite-Preicision Effects in DSP Applications via Affine Arithmetic Modeling," in *Proceedings of the 40th annual Design Automation Conference (DAC '03)*, no. 496-501, 2003.

[12] M. Barnasconi, K. Einwich, C. Grimm, T. Maehne, and A. Vachoux, "Advancing the SystemC Analog/Mixed-Signal (AMS) Extensions: Introducing Dynamic Timed Data Flow," *Open SystemC Initiative (OSCI), White Paper*, 2011.

[13] F. Schupfer, M. Kärgel, C. Grimm, M. Olbrich, and E. Barke, "Towards Abstract Analysis Techniques for Range Based System Simulations," in *Forum on Specification and Design Languages 2010 (FDL '10)*, 2010, pp. 1–6.