Verification of Analog and Mixed-Signal Circuits using Hybrid System Techniques*

Thao Dang, Alexandre Donzé, Oded Maler

VERIMAG
Centre Equation, 2 avenue de Vignate
38610 Gières, France
{tdang,donze,maler}@imag.fr

Abstract. In this paper we demonstrate a potential extension of formal verification methodology in order to deal with time-domain properties of analog and mixed-signal circuits whose dynamic behavior is described by differential algebraic equations. To model and analyze such circuits under all possible input signals and all values of parameters, we build upon two techniques developed in the context of hybrid (discrete-continuous) control systems. First, we extend our algorithm for approximating sets of reachable sets for dense-time continuous systems to deal with differential algebraic equations (DAEs) and apply it to a biquad low-pass filter. To analyze more complex circuits, we resort to bounded horizon verification. We use optimal control techniques to check whether a Δ-Σ modulator, modeled as a discrete-time hybrid automaton, admits an input sequence of bounded length that drives it to saturation.

1 Introduction

Formal verification has become part of the development cycle of digital circuits. Its advantage relative to more traditional simulation methods lies in its *exhaustiveness*: It can guarantee that a system behaves correctly in the presence of *all* its possible inputs, whose number can be infinite or too large to be covered by individual simulations. Of course, this advantage does not come for free and verification algorithms are more complex and costly than simple simulation. The extension of verification methodology to deal with analog and mixed-signal circuits is far from being straightforward due to the following reason. Digital circuits are modeled as discrete event dynamical systems (automata, transition systems) where the inputs are sequences of binary values, and the behaviors of the circuit induced by these inputs are binary sequences corresponding to paths in the transition graph. Hence digital verification can be realized using graph search algorithms. In contrast, the mathematical model of an analog circuit is

* This work was partially supported by the European Community projects IST-2001-33520 CC (Control and Computation) and IST-2003-507219 PROSYD (Property-based System Design) and by the US Army Research Office (ARO) contract no. DAAD19-01-1-0485.

that of a continuous dynamical system defined typically by differential algebraic equations where inputs are real-valued signals defined over the real time axis, and the behaviors they induce are trajectories in the continuous state space of the system.

A typical verification task is to prove that a circuit behaves correctly for all possible input signals and that none of them drives the system into a bad state, for example a state where one of the components reaches saturation. Even if we are satisfied with checking the circuit against a *finite* number of typical input signals, something that can be done using numerical simulation, we still have a problem because of the possible variations in system parameters which are determined only *after* low-level synthesis is complete. To account for such variations during high-level design, symbolic analysis methods that compute symbolic expressions characterizing the behavior of a circuit have been developed and successfully applied to linear or linearized circuits [14]. Extensions of these methods to non-linear circuits are mainly based on simplification and reduction to linear or weakly non-linear systems, which are often limited by accuracy trade-offs (see, for example, [32, 31]). Consequently, numerical simulation with a finite number of input signals is the commonly used validation tool, albeit its inadequacy for systems with under-specified parameters.

In this paper we focus on verifying *time-domain properties*[1] of analog and mixed-signal circuits with dynamics described by a system of differential algebraic equations with parameters. To analyze such a circuit under all possible input signals and parameter values, we use techniques developed in the context of hybrid (discrete-continuous) control systems (see the conference proceedings [17, 25, 2] for a sample of recent hybrid systems research). In particular we extend the forward reachability analysis technique that we have developed for linear [5] and non-linear [4] *ordinary* differential equations to deal with differential *algebraic* equations. The case of mixed-signal circuits is investigated through the modeling and analysis of a Δ-Σ modulator, a widely used circuit, for which stability analysis remains a challenging problem [29, 20, 12]. We tackle this problem using the approach advocated in [8] for the verification discrete-time hybrid systems. The idea is to formulate bounded horizon reachability as a hybrid constrained optimization problem that can be solved by techniques such as mixed-integer linear programming (MILP), in the same sense that bounded verification of digital systems can be reduced to solving a Boolean satisfiability (SAT) problem.

There have been several previous works on formal verification of analog circuits (see [22, 15, 19, 27] and references there in). The work closest to this paper is [22, 19], in which an analog system is approximated by a discrete system in which classical model-checking algorithms can be applied. The discrete system is obtained by partitioning the state space into boxes (each of which corresponds to a state of the discrete model). In [19] the transition relation is determined

[1] Frequency-domain properties which are often used in analog design are outside the scope of this paper. Properties used in digital verification, such as those specified in *temporal logic* are, using this terminology, time-domain properties.

by reachability relation between the boxes which is approximated by simulating trajectories from some test points in each box.

The rest of the paper is organized as follows. In Section 2 we present our approach to the verification of non-linear analog circuits using reachability computation for differential algebraic equations. The approach is then illustrated with a low-pass filter circuit. In Section 3 we formulate the bounded horizon verification problem for a mixed-signal Δ-Σ modulator and solve it using an MILP solver. Some discussions and future research directions close the paper.

2 Verification of Non-Linear Analog Circuits

2.1 Approach

Mathematically, the behavior of a non-linear analog circuit can be described by a set of differential algebraic equations (DAE):

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

where $x \in \mathbb{R}^n$ denotes the state variables (internal voltages, currents, and outputs), $\dot{x}$ denotes their time derivatives, $p \in P \subset \mathbb{R}^m$ is the parameter vector, and $u : \mathbb{R}^+ \to U$ is the input signal. We assume a set $\mathcal{U}$ of admissible input signals consisting of piecewise-continuous functions taking values in a bounded and convex set $U \subset \mathbb{R}^l$. In this model the input is uncertain, which allows one to model external disturbance and noise. A parameter can be a resistor value, a transistor saturation current, etc. The equations (1) result from applying Kirchhoff laws to the whole circuit and the characteristics equations to the basic elements. Such circuit equations can be automatically generated by techniques such as Modified Nodal Analysis (MNA) [13].

To verify time-domain properties of the circuit, such as those related to the transient behavior, one needs to characterize the set of solutions of (1) under all possible inputs $u(\cdot)$ and all parameter values p. For reachability properties (the circuit never reaches a bad state), it suffices to compute the set of states reachable by all possible trajectories of the system that we define formally below.

We denote by $\gamma(t, x_0, u(\cdot), p)$ the value at time t of the solution of (1) with the initial condition $x(0) = x_0$ under the input signal $u(\cdot) \in \mathcal{U}$ and a parameter $p \in P$. Given a set of initial conditions X_0 and $T > 0$, the reachable set from X_0 during the time interval $[0, T]$ is defined as:

$$\begin{aligned} \Phi(X_0, T) = \{\gamma(t, x_0, u(\cdot), p) \mid t \in [0, T] \ \wedge \ x_0 \in X_0 \\ \wedge \, u(\cdot) \in \mathcal{U} \ \wedge \ p \in P\}. \end{aligned}$$

Note that, unlike simulation, reachability computations can also handle uncertainty in initial conditions. The extension of reachability techniques for ordinary differential equations (ODEs) to handle DAEs is not straightforward since these classes of equations differ in both theoretical and numerical properties, and this is captured by the *index* concept (for an introduction see [9]). The differential

index of (1) is the minimal number of differentiations required to solve for the derivatives $\dot{x}$. In general the problem of numerically solving DAEs with index 2 or higher is ill-posed [7]. DAEs that model practical electronic circuits are typically of index 1 or 2 and in this work we focus on the former. In particular, we will study the equivalent semi-explicit form of (1):

$$\dot{x}(t) = f(x(t), y(t), p), \tag{2}$$

$$0 = g(x(t), y(t), p). \tag{3}$$

Note that the implicit DAE system (1) can be trivially transformed into the above form as follows: By letting $z(t) = \dot{x}(t)$, $y = (z, u)$ and substituting in (1) we obtain $0 = F(x(t), y(t), p)$. Thus the resulting system in the above semi-explicit form is: $\dot{x}(t) = z(t)$ and $0 = F(x(t), y(t), p)$.

Coupling the ODE (2) with the non-linear equation (3) means that the solution of (2) has to lie on the manifold defined by (3). If the Jacobian $g_y(x, y) = \partial g/\partial y$ is invertible in a neighborhood of the solution, then by differentiating the algebraic equation we obtain

$$\dot{y} = -g_y^{-1} g_x f, \tag{4}$$

and in this case, the DAE system is of index 1. In a simpler case, where $\partial F/\partial \dot{x}$ in (1) is regular, the algebraic equation (3) disappears, and (1) is a DAE of index 0, i.e. an ODE [9].

A trivial way to compute reachable sets for index 1 DAEs is to transform it into an ODE composed of (2) and (4) using the above-described differentiation and then apply the existing techniques for ODEs. However, the drawback of this approach is that the solution may drift away from the algebraic constraint. We will retain the algebraic constraint (3) and interpret the original DAE as the ODE, composed of (2) and (4), on the manifold defined by (3). We will combine the commonly-used technique of geometric integration using projection [11], with our reachability algorithm, to compute the reachable set.

2.2 Computing Reachable Sets of ODEs on Manifolds

We summarize below the approach we have developed over the years for reachability computation of ODEs and hybrid automata. We start with an algorithm for linear ODEs of the form $\dot{x}(t) = Ax(t)$, first presented in [5] and implemented in the tool **d/dt** [6]. Many other techniques for computing reachable sets can be found in the hybrid systems literature. In particular, those developed independently by Chutinan and Krogh and implemented in the tool CheckMate [10] are very similar to ours. We use $\gamma(t, x_0)$ for the solution and $\Phi(X_0, T)$ for the states of the solutions at any $t \in T$ starting from any $x_0 \in X_0$. Basically we compute a polyhedral over-approximation of the reachable states on a step-by-step basis as in numerical integration, that is, we compute a sequence of polyhedra that over-approximates the sequence

$$\Phi(X_0, r), \Phi(\Phi(X_0, r), r) \ldots$$

Given a convex polyhedron R, the set R' of states reachable from R at time r can be computed as the convex hull of the points reachable at time r from the vertices of R. Then, the set of states reachable during the whole time interval $[0, r]$ is approximated by the convex hull $conv(R \cup R')$ which is enlarged by an appropriate amount to ensure conservative approximation.[2]

This basic algorithm is then extended in various directions. When the system admits an input and is of the form $\dot{x}(t) = Ax(t) + bu(t)$, the computation can still be done by applying optimization techniques to find "extremal" values for u that push the set "outwards" [30]. Another important extension handles systems that admit mode switching and are modeled by *hybrid automata* [1], automata that have a distinct differential equation in each state, depending on the values of the continuous state variables. As long as the automaton remains in the same discrete state, the reachability computation proceeds as for simple ODEs but when the reachable polyhedron intersects the switching surface, it needs to be split and parts of it undergo reachability computation under the new dynamics. We will come back to this in the next section and the reader is referred to [5, 6] for more details.

The analysis of hybrid automata with under-specified inputs is part of the more recent methodology [4] for analyzing non-linear systems of the form $\dot{x}(t) = f(x(t)) + bu(t)$ using a piecewise-affine approximation. This method is based on partitioning the state space into simplices and assigning a distinct discrete state to each of them. The dynamics at each state is specified by an affine function obtained by interpolation on the values of f on the vertices the corresponding simplex. This approximation is conservative since the interpolation error is included in the model as an input. In addition, if the derivative $\dot{x}$ is a C^2 function, the reachable set approximation error is quadratic in the size of the underlying simplicial partition.

Before proceeding let us remark that the potential contribution of ideas coming from hybrid systems to the design of analog circuits is not restricted to verification. In particular, the modeling of non-linear systems by piecewise-linear ones, called *hybridization* in [4, 3], offers an alternative modeling style that was often avoided because it does not fit into the analytical and numerical framework of continuous systems. On the contrary, many discontinuous phenomena that could have been modeled naturally as discrete transitions, are often "smoothened" to avoid numerical instability. Hybrid modeling and analysis can treat such phenomena directly.

We can now return to ODEs on a manifolds and combine reachability with projection. For the sake of clarity we omit u and p and work with

$$\dot{x}(t) = f(x(t)), \tag{5}$$

$$0 = g(x(t)). \tag{6}$$

The following algorithmic scheme, illustrated in Figure 1, computes an approximation of the reachable states where Φ is the reachability operator and $\Pi_{\mathcal{M}}$ denotes projection onto the manifold $\mathcal{M}$ defined by (6).

[2] Note that this part of the algorithm is not needed for discrete-time systems.

Algorithm 1 Computation of $\Phi(X_0, .)$ with time step r.

$R_0 = X_0$
repeat $k = 0, 1, \ldots,$
  $\hat{R}_{k+1} = \Phi(R_k, r)$
  $R_{k+1} = \Pi_{\mathcal{M}}(\hat{R}_{k+1})$
until $R_{k+1} = \bigcup_{i=1}^{k} R_i$

The projection of a point $x \in \mathbb{R}^n$ onto the manifold $\mathcal{M}$ is computed as

$$\Pi_{\mathcal{M}}(x) = \arg\min_{\bar{x}} |x - \bar{x}| \quad \text{subject to } g(\bar{x}) = 0,$$

where $|\cdot|$ is the Euclidean norm. In the special case where g is linear, this optimization problem can be easily solved using linear algebra. The projection of a convex polyhedron $\hat{R} = conv(v^1, \ldots v^m)$ is approximated as the convex hull of the projected vertices, $R = conv(\bar{v}^1, \ldots \bar{v}^m)$. Although R does not always lie entirely on $\mathcal{M}$, its distance to $\mathcal{M}$ can be made as small as desired. Indeed, we can prove that the convergence order of this approximate reachability method for DAEs is that of the reachability method for ODEs used to compute Φ, which is *quadratic* [6, 4]. To see this consider a point $x \in \mathcal{M}$ and its successor $\hat{x} = \Phi(\{x\}, r)$ computed by Φ. The distance between $\hat{x}$ and $\mathcal{M}$ is bounded by the local error of the method for computing Φ. Hence, the distance between $\Pi_{\mathcal{M}}(\hat{x})$ and the exact successor of x is of the same order.

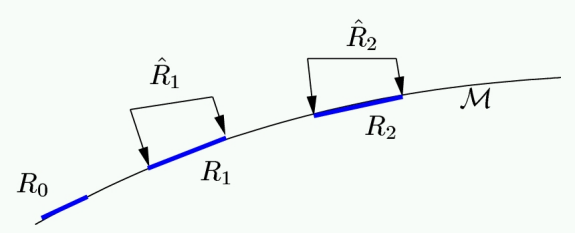

Fig. 1. Combining projection and reachability computations for ODEs.

2.3 Example: a Biquad Low-pass Filter

We now illustrate the approach with a second order biquad low-pass filter circuit, shown in Figure 2. This example is taken from [19]. The circuit equations are as

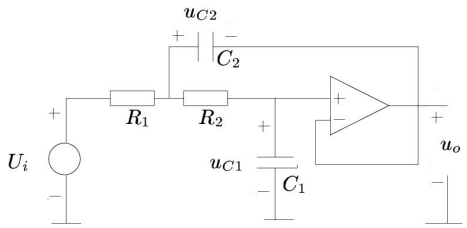

Fig. 2. A low-pass filter.

follows:

$$\dot{u}_{C1} = \frac{u_{C2} + u_o - u_{C1}}{C_1 R_2}, \tag{7}$$

$$\dot{u}_{C2} = \frac{U_i - u_{C2} - u_o}{C_2 R_1} - \frac{u_{C2} + u_o - u_{C1}}{C_2 R_2}, \tag{8}$$

$$u_o - V_{max} \tanh(\frac{(u_{C2} - u_o)V_e}{V_{max}}) + U_{om} = 0, \tag{9}$$

$$i_o = -C_2\, \dot{u}_{C2}, \tag{10}$$

$$U_{om} = \mathcal{V}(i_0), \tag{11}$$

$$\mathcal{V}(i_o) = K_1 i_o + 0.5\sqrt{K_1 i_o^2 - 2K_2 i_o I_s + K_1 I_s^2 + K_2} - 0.5\,\sqrt{K_1 i_o^2 + 2K_2 i_o I_s + K_1 I_s^2 + K_2}. \tag{12}$$

The state variables are (u_{C1}, u_{C2}), the voltages across the capacitors C_1 and C_2 (the reference directions of which are indicated by the $+$ and $-$ signs in Figure 2). The algebraic constraints (9-12) come from the characteristics of the operational amplifier where u_o is the output voltage and U_{om} corresponds to the output voltage decrease caused by the output current i_o. In this circuit, U_i (input voltage), V_{max} (maximal source voltage), V_e, I_s, C_1, C_2, R_1, and R_2 are parameters. Denoting $x = (u_{C1}, u_{C2})$ and $y = u_o$, the circuit equations can be put in the semi-explicit form (2-3). Assuming that the Jacobian $g_y(x, y)$ has bounded inverse in a neighborhood of the solution (which can indeed be verified for a concrete circuit), by differentiating (9) the circuit equations can be transformed into a non-linear ODE on a manifold as in (5-6) with state variables $z = (u_{C1}, u_{C2}, u_o)$.

As mentioned earlier, to reduce the complexity of reachability computation, we will use the hybridization idea. First, the non-linear characteristics $U_{om} = \mathcal{V}(i_o)$ in equation (11) can be approximated by a piecewise-affine function of the form:

$$\mathcal{V}(i_o) = \begin{cases} K_1 i_o + K_3 & \text{if } i_o \le I_s, \\ 0 & \text{if } -I_s < i_o < I_s, \\ K_1 i_o - K_3 & \text{if } i_o \ge I_s. \end{cases} \tag{13}$$

Therefore, the original system is approximated by a hybrid automaton with 3 discrete states (modes). The conditions for staying in a mode and for switching between modes are determined by the value interval of i_o. For example, in order to stay in the mode corresponding to the first equation of (13) the state variables (u_{C1}, u_{C2}, u_o) should satisfy $i_o \leq I_s$. Using (10) this condition becomes: $-C_2\,\dot{u}_{C2} \leq I_s$, which together with (8) gives

$$-\frac{U_i - u_{C2} - u_o}{R_1} - \frac{u_{C2} + u_o - u_{C1}}{R_2} \leq I_s.$$

Note that the hyperbolic tangent function in (9) is retained because it can be observed from simulation results that this non-linearity is important for the accuracy of the model. In general, the designer's knowledge of the the circuit can help to choose appropriate simplifications and approximations. As a result, the continuous dynamics of each mode is defined by a DAE which remains non-linear and is transformed automatically to a piecewise-affine dynamics using the hybridization technique of [4].

The property to verify is the absence of overshoots. For the highly damped case (where $C_1 = 0.5e-8$, $C_2 = 2e-8$, and $R_1 = R_2 = 1e6$), Figure 3 shows the projection of the reachable set on u_{C1} and u_{C2}. The initial set is defined by a box: $u_{C1} \in [-0.3, 0.3]$, $u_{C2} \in [-0.3, 0.3]$ and $u_o \in [-0.2, 0.2]$. From the figure, one can see that u_{C1} indeed remains in the range $[-2, 2]$. This computation took **d/dt** 3 minutes until termination. We are currently working on making this process more systematic and efficient. In particular we investigate the automatic transformation of circuit equations into ODEs on a manifold.

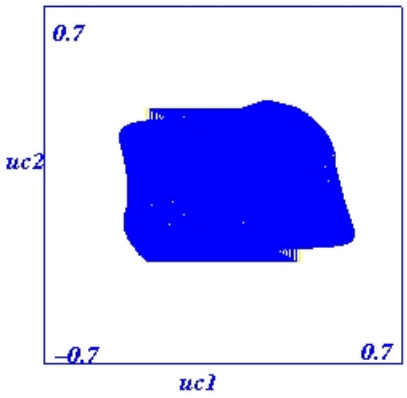

Fig. 3. The reachable set for the filter projected on variables u_{C1} and u_{C2}.

3 Verification of Mixed-Signal Circuits

3.1 Optimal Control based Verification Approach

Mixed-signal circuits that exhibit both logical and continuous behaviors can be naturally modeled as hybrid automata and verified using the reachability techniques described in the previous section. However, as it also happens in digital verification, reachability algorithms may explode in time and space before termination and less exhaustive methods should sometimes be used. One popular approach is to restrict the verification to behaviors of *bounded* length and ask whether the set of such behaviors contains one that violates the property in question. A positive answer demonstrates a "bug" in the system while a negative one is generally not a proof of correctness unless the length bound is very large. Bounded horizon reachability for digital systems is typically formulated as Boolean satisfiability. For dynamical systems over a continuous state space, bounded horizon problems were used in the context of optimal control, where one looks for a finite input signal that induces a behavior which is optimal according to some performance criterion. In discrete-time, this problem reduces to a finite-dimensional optimization of a continuous function subject to continuous constraints (see a unified treatment of the discrete and continuous case in [23]). The application of these ideas to the verification of hybrid systems has been advocated by Bemporad and Morari [8]. In verification the input is interpreted as a disturbance and the search for a bad behavior becomes a search for the *worst* input sequence with respect to the property in question (see also [30] for the applicability of optimal control to reachability-based verification). The discrete part of the system makes the optimization problem hybrid, and one of the popular methods for solving it is mixed integer-linear programming (MILP).

In this section we focus on circuits that can be modeled by a discrete-time hybrid system of the form:

$$F(x(k), x(k+1), u(k), \delta(k), p) = 0, \ k \in \mathbb{N}, \tag{14}$$

where $\delta(k) \in \{0,1\}^s$ is a binary vector of dimension s describing the logical part of the dynamics. For convenience, we will use notation similar to the continuous-time case. We use $x(k) = \gamma(k, x(0), u(\cdot))$ to denote the state at time k of the solution of (14) with initial state $x(0) \in X_0$ and input $u(\cdot) \in \mathcal{U}$ which is a sequence ranging over a closed bounded set $U \subset \mathbb{R}^l$ (i.e., $u(\cdot) = (u(k))_{k\in\mathbb{N}}$).

To prove safety over a finite horizon $N \in \mathbb{N}$ we compute a set of *worst* trajectories whose safety implies the safety of all the other trajectories. The formulation of verification as an optimal control problem is done via an objective function J such that $J(x)$ is positive iff x is outside the safe set. Then for each $k \leq N$, we maximize J for the trajectory $x(t)$ with $t = 0, \ldots, k$ by solving the

following constrained optimization problem:

$$\max\ J(x(k)), \tag{15}$$

$$s.t.\ \ F(x(t), x(t+1), u(t), \delta(t), p) = 0, \tag{16}$$

$$u(t) \in \mathcal{U},\ t \in \{0, 1, \ldots, k-1\}, \tag{17}$$

$$x(0) \in X_0. \tag{18}$$

We then check whether the worst trajectories obtained satisfy $J(x(k)) \leq 0$ for all $k \leq N$, meaning that the property is true over horizon N. We illustrate this approach through the stability analysis of a Δ-Σ modulator.

3.2 The Δ-Σ Modulation: Principles and Hybrid Modeling

We describe briefly the principles of Δ-Σ modulation, a very popular technique for analog to digital conversion. Basically, a Δ-Σ modulator processes an analog input through four steps [29]: (1) *Anti-aliasing* in order to be sure that the signal bandwidth lies within a given range $[-f_b, f_b]$; (2) *Oversampling* or sampling at a frequency greater than the Nyquist rate $2 \times f_b$; (3) *Noise shaping* so that the quantization error is "pushed" toward high frequencies outside the bandwidth of interest; (4) *Quantization*, typically on few bits. In the following examples quantization is done on one bit. We use an input-output plot of a simple model,

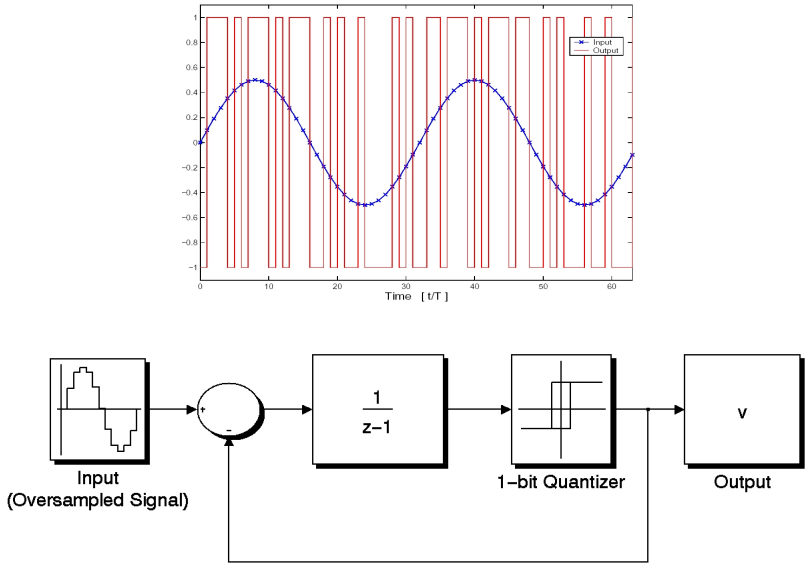

Fig. 4. A first order Δ-Σ modulator and an example of an input-output plot.

shown in Figure 4, to explain intuitively how Δ-Σ modulation works. When

the input sinusoid is positive and its value is less than 1, the output takes the $+1$ value more often and the quantization error which is the difference between the input and the output of the quantizer is fed back with negative gain and "accumulated" in the integrator $\frac{1}{z-1}$. Then, when the accumulated error reaches a certain threshold, the quantizer switches the value of the output to -1 for some time, which reduces the mean of the quantization error. This model is called a first order Δ-Σ modulator since it uses a first order filter to process noise.

We now describe a hybrid model of a third-order Δ-Σ modulator (shown in Figure 5), generated using the standard MATLAB **delsig** toolbox [28] which provides practical models used by designers. Higher order Δ-Σ modulators achieve

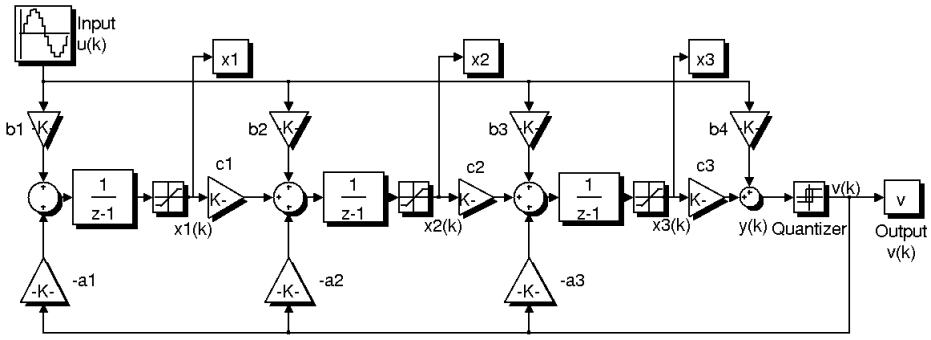

Fig. 5. A model of a third-order modulator with integrators that may saturate.

better performance but induce stability problems [12]. A modulator is said to be stable if its integrators values remain bounded under a bounded input. This property is of a great importance since integrator saturation can deteriorate circuit performance. Stability analysis for such circuits is still a challenging research problem [12] due to the presence of two sources of non-linearities: saturation and quantization.

The circuit is modeled as a discrete-time hybrid automaton. When none of the integrators saturates, the dynamics of the system can be represented in the following state-space form:

$$x(k+1) = Ax(k) + bu(k) - sign(y(k))a, \tag{19}$$

$$y(k) = c_3 x_3(k) + b_4 u(k), \tag{20}$$

where matrix A, vectors a and b are constants depending on the various gains of the model, $x(k) \in \mathbb{R}^3$ represents the integrator states, $u(k) \in \mathbb{R}$ is the input and $y(k) \in \mathbb{R}$ is the input to the quantizer. The output of the quantizer $v(k) = sign(y(k))$ is the only discrete state variable and as long as it remains constant, the dynamics is continuous and affine. Figure 6 gives the usual graph representation of the corresponding hybrid automaton. Note that the discrete

state variable can be made Boolean by letting $\delta(k) = \frac{sign(y(k))+1}{2}$ which transforms (19-20) to the general form of (14).

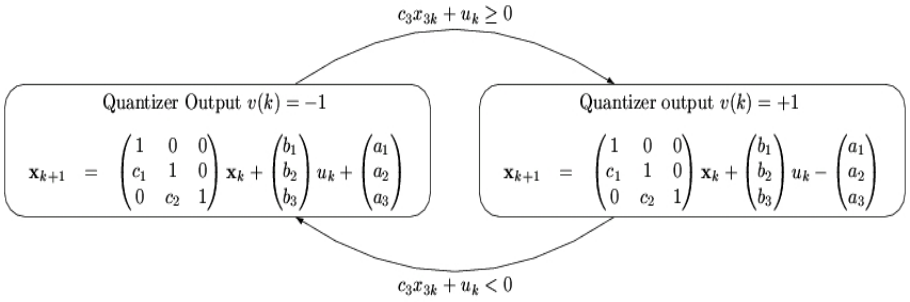

Fig. 6. A hybrid automaton model of the Δ-Σ modulator.

3.3 Stability Analysis: Formulation and Results

The stability property of the modulator is stated as follows: For a given bounded set $X_0 \subset \mathbb{R}^3$ of initial states and a range of input values $U = [u_{min}, u_{max}]$, the system is stable if and only if for any $x(0) \in X_0$ and any sequence $u(\cdot) \in \mathcal{U}$ the sequence $\gamma(k, x(0), u(\cdot))$ is bounded, that is, there exists a *bounded* set $\mathcal{S}$ such that

$$\forall k \in \mathbb{N}\ \forall x(0) \in X_0\ \forall u(\cdot) \in \mathcal{U} : \gamma(k, x(0), u(\cdot)) \in \mathcal{S}.$$

In the following, we apply the method described in Section 3.1 to check this property over a horizon N where the safe set $\mathcal{S}$ is the rectangular set $[-x_1^{sat}, x_1^{sat}] \times [-x_2^{sat}, x_2^{sat}] \times [-x_3^{sat}, x_3^{sat}]$, the set of states where no integrator saturates. Since we want x to remain inside $\mathcal{S}$, we define the objective function J as:

$$J(x(k)) = \max_{i=1,2,3}(|x_i(k)| - x_i^{sat}).$$

Solving this optimization problem means finding an input sequence that drives the integrators as close as possible to their saturation limits. By symmetry it can be easily shown that if $\gamma(\cdot, x(0), u(\cdot))$ is a sequence obtained from $x(0)$ with input $u(\cdot)$, then we have $\gamma(k, x(0), u(\cdot)) = -\gamma(k, -x(0), -u(\cdot))$ for all k. Thus, if X_0 and U are symmetric sets with respect to the origin, maximizing $|x_i(k)|$ is the same as maximizing $x_i(k)$; hence, we can define J as: $J(x(k)) = \max_{i=1,2,3}(x_i(k) - x_i^{sat})$.

We transform this problem into 3 MILP problems (one for each i) by rewriting the function F, given by (19-20), as a set of linear constraints over real and binary variables. To get rid of the *sign* function, we use the standard "big-M" trick. Given bounds $m < 0$ and $M > 0$ on $y(k)$, we introduce two new constraints for

all k:

$$y(k) \leq \delta(k)M, \tag{21}$$
$$y(k) > (1 - \delta(k))m. \tag{22}$$

Thus, it holds that $\delta(k) = 1 \Leftrightarrow\ y(k) \geq 0$, and we can replace $sign(y(k))$ in (19) by $2\delta(k) - 1$. With the above definitions and constraints, the problem (15-18) is put in MILP form.

We used the efficient solver MOSEK [26] to solve the resulting MILP problem for various bounds on initial states $x(0)$ and input signals u over a finite horizon N ranging from 1 to 30. The results are shown in Figure 7, where the curves depict the maximal obtained value of $x_1(N)$ as a function of N (note that for each N the maximum might be obtained by another input sequence). The qualitative behavior exhibited by x_2 and x_3 is similar. From these plots one can see, for example, that for $x(0) \in [-0.1, 0.1]^3$ and any *constant* sequence $u(k) = c \in [-0.5, 0.5]$ for all N the maximal value of $x_1(N)$ never leaves the safe set $\mathcal{S}$ and, moreover, it converges quite fast towards a constant value, which shows that the Δ-Σ modulator is stable up to $N = 30$ and most likely forever after. This also holds for $x(0) \in [-0.01, 0.01]^3$ and *any* $u(k) \in [-0.1, 0.1]$ but not for $u(k) \in [-0.5, 0.5]$. Note furthermore that the bad input signals the we found are generally non-trivial, and could not have been found easily by simulation with initial states and input values that are simply on the boundaries of their domains.

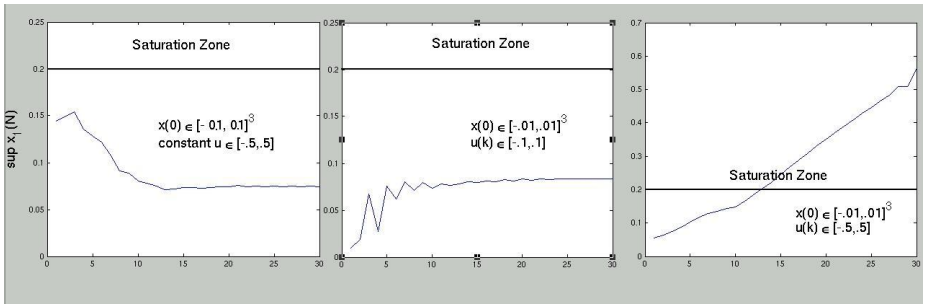

Fig. 7. The curves show $\sup x_1(N)$ as a function of the horizon N for various bounds on initial states $x(0)$ and input signals u.

3.4 Reachability versus Optimal Control.

Safety verification of piecewise affine hybrid systems can also be achieved by computing a so-called *robustly positively invariant set* (RPI), which has already been used in the context of Δ-Σ modulators [21, 16]. An RPI set Ω is such that

if $x(k) \in \Omega$ then $x(k+1) \in \Omega$ regardless of $u(k)$. Indeed, if such a bounded set containing X_0 is found, the boundedness of $x[k]$ is guaranteed. It is easy to see that the reachable set is an RPI set and therefore, we can prove the stability of the Δ-Σ circuit by computing (or over-approximating) the reachable set using a discrete-time version of the algorithm of Section 2. Nevertheless, two characteristics of the Δ-Σ modulator render the reachability computation very expensive:

- Switching between modes is very frequent, which makes the reachable set highly non convex admitting an exponentially growing number of polyhedra.
- The fundamental instability introduced by the integrators makes the system particularly sensitive. Hence, while the use of over-approximations of the reachable set (such as using convex hull) can reduce the computational complexity, the results are often too coarse to prove the property.

One can observe similar phenomena in a recent application of reachability techniques to the same circuit (but under more restricted input signals) reported in [18]. The optimization procedure just described is not immune to these problems although it can reach larger horizons (the computation for horizon 30 took more than two hours on a 2.4GHz machine). On the other hand, efficient algorithms for computing RPI sets, such as those described in [21], require the affine modes to be stable and thus cannot be used for this example. Indeed, the instability of the integrators (the A matrix) implies the instability of each mode of the affine system and the stability of the modulator relies only on switches between the modes.

4 Conclusion and Future Work

We have presented a framework for modeling and verification of analog and mixed-signal circuits using hybrid system techniques. These results are much more modest, both in terms of rigor (approximate computation, non guaranteed termination) and of size (systems with few state variables) than the state-of-the-art in digital hardware verification, but this is not surprising given the inherent complexity of the problems and the current practices in the domain. Fortunately, the accumulated experience in hybrid systems will be useful in accelerating the progress in analog verification, especially by avoiding dead ends that have already been explored.

Some innovative ideas are needed in order to extend the scope of our techniques to larger systems. As in discrete verification, abstraction and model reduction techniques are necessary in order to facilitate compositional reasoning. Although the circuit structure may give hints for useful decompositions, the nature of interaction between analog devices will probably not make this task easy. Another research direction would be to develop reachability techniques that take into account some more refined constraints on the input signals such as bounds on their frequency or on their average value. Current reachability algorithm for

systems with input are essentially "breadth-first" and are not adapted for excluding individual input signals that violate such constraints. Additional work that will be needed in order to integrate formal methods in the design process includes the automatic translation from circuit descriptions (such as those expressed in VHDL-AMS) to hybrid automata, as well as the definition of an expressive formalism for specifying properties of analog signals. First steps in this direction are reported in [24].

Acknowledgment. We thank Lars Hedrich for his help in modeling the biquad filter and for providing us with a lot of information about various techniques used for validating analog circuits. All that we know about Δ-Σ modulation is the result of our interactions with Rob Rutenbar, Bruce Krogh and Smriti Gupta at CMU during the summer of 2003. Goran Frehse provided many useful comments on previous versions of this paper.

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science* 138:3–34, 1995.
2. R. Alur and G.J. Pappas (eds). *Hybrid Systems: Computation and Control*. LNCS 2993, Springer, 2004.
3. E. Asarin and T. Dang. Abstraction by projection. In *Hybrid Systems: Computation and Control*, LNCS 2993, 32–47, Springer, 2004.
4. E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Computation and Control*, LNCS 2623, 20–35. Springer, 2003.
5. E. Asarin, O. Bournez, T. Dang and O. Maler, Reachability analysis of piecewise-linear dynamical systems, In *Hybrid Systems: Computation and Control*, LNCS 1790, 20-31, Springer, 2000.
6. E. Asarin, T. Dang, and O. Maler. The d/dt tool for verification of hybrid systems. In *Computer Aided Verification*, LNCS 2404, 365–370, Springer, 2002.
7. U.M. Ascher and L.R. Petzold. Stability of computational methods for constrained dynamics systems. *SIAM Journal on Scientific Computing* 14:95–120, 1993.
8. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints, *Automatica* 35:407-427, 1999.
9. K.E. Brenan, S.L. Campell, and L.R. Petzold. *Numerical Solution of Initial Value Problems in Ordinary Differential-Algebraic Equations*. North Holland, 1989.
10. A. Chutinan and B.H. Krogh. Verification of polyhedral invariant hybrid automata using polygonal flow pipe approximations. In *Hybrid Systems: Computation and Control*, LNCS 1569, 76-90, Springer, 1999.
11. C. Lubich and E. Hairer and G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations.*, volume 31 of *Series in Comput. Mathematics*. Springer, 2003.
12. B. Pérez-Verdú and F. Medeiro and A. Rodríguez-Vázquez. *Top-Down Design of High-Performance Sigma-Delta Modulators*, chapter 2. Kluwer Academic Publishers, 2001.

13. U. Feldmann and M. Günther. The DAE-index in electric circuit simulation. In *Proc. IMACS, Symposium on Mathematical Modelling* 4:695–702, 1994.
14. H. Floberg. *Symbolic Analysis in Analog Integrated Circuit Design*. Kluwer, 1997.
15. A. Ghosh and R. Verumi. Formal verification of synthesized analog circuits. In *Int. Conf on Computer Design* 4:40–45, 1999.
16. M. Goodson R. Schreier and B. Zhang. An algorithm for computing convex positively invariant sets for delta-sigma modulators. *IEEE Transactions on Circuits and Systems I* 44:38–44, January 1997.
17. M. Greenstreet and C. Tomlin (eds). *Hybrid Systems: Computation and Control*. LNCS 2289. Springer-Verlag, 2002.
18. S. Gupta, B.H. Krogh, and R.A. Rutenbar, Towards formal verification of analog designs, *Proc. ICCAD 2004* (to appear), 2004.
19. W. Hartong, L. Hedrich, and E. Barke. On discrete modelling and model checking for nonlinear analog systems. In *Computer Aided Verification*, LNCS 2404, 401–413, Springer, 2002.
20. S. Hein and A. Zakhor. On the stability of sigma delta modulators. *IEEE Transactions on Signal Processing* 41, 1993.
21. K. Kouramas S.V. Rakovic, E.C. Kerrigan and D.Q. Mayne. Approximation of the minimal robustly positively invariant set for discrete-time LTI systems with persistent state disturbances. In *42nd IEEE Conference on Decision and Control*, 2003.
22. R.P. Kurshan and K.L. McMillan. Analysis of digital circuits through symbolic reduction. *IEEE Trans. on Computer-Aided Design* 10:1350–1371, 1991.
23. O. Maler. On optimal and sub-optimal control in the presence of adversaries. *Workshop on Discrete Event Systems (WODES)*, (to appear), 2004.
24. O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. *Proc. FORMATS/FTRTFT'04*, (to appear), 2004.
25. O. Maler and A. Pnueli (eds). *Hybrid Systems: Computation and Control*. LNCS 2623, Springer, 2003.
26. MOSEK ApS. The Mosek Optimization Toolbox for Matlab version 3.0 (revision 19) user's guide and reference manual, November 2003.
27. A. Salem. Semi-formal verification of VHDL-AMS descriptors. *IEEE Int Symposium on Circuits and Systems*, 5:V–333–V–336, 2002.
28. R. Schreier. The delta-sigma toolbox version 6.0, January 2003.
29. H.V. Sorensen P.M. Aziz and J.V.D. Spiegel. An overview of sigma-delta converters. *IEEE Signal Processing Magazine*, 61–84, January 1996.
30. P. Varaiya. Reach set computation using optimal control. In *Proc. KIT Workshop*, Verimag, Grenoble, 1998.
31. T. Wichmann. Computer aided generation of approximate DAE systems for symbolic analog circuit design. In *Proc. Annual Meeting GAMM*, 2000.
32. T. Wichmann, R. Popp, W. Hartong, and L. Hedrich. On the simplification of nonlinear DAE systems in analog circuit design. In *Computer Algebra in Scientific Computing*. Springer, 1999.