

# Statistical Model Checking of Mixed-Analog Circuits with an Application to a Third Order $\Delta - \Sigma$ Modulator\*

Edmund Clarke, Alexandre Donzé, Axel Legay

School of Computer Science  
Carnegie Mellon University, Pittsburgh, PA 15213  
{emc|adonze|a.legay@cs.cmu.edu}

**Abstract.** In this paper, we consider verifying properties of mixed-signal circuits, i.e., circuits for which there is an interaction between analog (continuous) and digital (discrete) quantities. We follow the statistical Model Checking approach of [You05,You06] that consists of evaluating the property on a representative subset of behaviors, generated by simulation, and answering the question of whether the circuit satisfies the property with a probability greater than or equal to some value. The answer is correct up to a certain probability of error, which can be pre-specified. The method automatically determines the minimal number of simulations needed to achieve the desired accuracy, thus providing a convenient way to control the trade-off between precision and computational cost. We propose a logic adapted to the specification of properties of mixed-signal circuits, in the temporal domain as well as in the frequency domain. Our logic is unique in that it allows us to compare the Fourier transform of two signals. We also demonstrate the applicability of the method on a model of a third order  $\Delta - \Sigma$  modulator for which previous formal verification attempts were too conservative and required excessive computation time.

## 1 Introduction

Given a property  $\phi$ , the *Probabilistic Model Checking Problem* consists of checking whether a stochastic system satisfies  $\phi$  with a probability greater than or equal to a certain threshold  $\theta$ . This problem is generally solved with a *numerical approach* that consists of computing the *exact* probability for the system to satisfy  $\phi$  and by comparing the result to  $\theta$ . The way the probability is computed

---

\* This research was sponsored by the GSRC (University of California) under contract no. SA423679952, National Science Foundation under contracts no. CCF0429120, no. CNS0411152, and no. CCF0541245, Semiconductor Research Corporation under contract no. 2005TJ1366, Air Force (University of Vanderbilt) under contract no. 18727S3, International Collaboration for Advanced Security Technology of the National Science Council, Taiwan, under contract no. 1010717, and a grant from the Belgian American Educational Foundation.

depends on the nature of the system as well as on the property that is considered. Successful results (see e.g. [BHHK03,CY95,CG04]) and tools (see e.g. [KNP04,CB06]) exist for various classes of systems, including (continuous time) Markov Chains and Markov Decision Processes. The drawback behind numerical approaches is that they compute the probability by considering all the executions of the system, which may not scale up for systems of large size. Another way to solve the probabilistic Model Checking problem is to use a *statistical approach* based on hypothesis testing and simulation (e.g., [You05,You06] or [SVA04,SVA05]). The key idea is to deduce whether or not the system satisfies the property by observing some of its executions. Of course, in contrast to a numerical approach, a test-based solution does not guarantee a correct result. However, it is possible to bound the probability of making an error. Statistical approaches are known to be far less memory and time intensive than numerical ones, and are sometimes the last resort [YKNP06].

In this paper, we consider applying the statistical procedure proposed by Younes in [You05,You06] to verify properties of *mixed-signal circuits*, i.e., circuits for which there is an interaction between analog (continuous) and digital (discrete) quantities. Our first contribution is to propose a version of stochastic discrete-time event systems that fits into the framework of [You05,You06] with the additional advantage that it explicitly handles analog and digital signals. We also introduce *probabilistic signal linear temporal logic*, a logic adapted to the specification of properties for mixed-signal circuits in the *temporal* domain and in the *frequency* domain.

Our second contribution is the analysis of a  $\Delta - \Sigma$  modulator. A  $\Delta - \Sigma$  modulator is an efficient *Analog-to-Digital Converter circuit*, i.e., a device that converts analog signals into digital signals. A common critical issue in this domain is the analysis of the *stability* of the internal state variables of the circuit. The concern is that the values that are stored by these variables can grow out of control until reaching a maximum value, causing the circuit to *saturate*. Saturation is commonly assumed to compromise the quality of the analog-to-digital conversion. In [DDM04] and [GKR04] reachability techniques developed in the area of hybrid systems were used to analyze the stability of a third-order modulator. The idea was to use these techniques to guarantee that for *every* input signal in a given range, the states of the system remain stable. While this reachability-based approach is strictly precise, it has important drawbacks such as (1) signals with long duration cannot be practically analyzed and (2) there are interesting properties that cannot be checked. Our results show that a statistical Model Checking approach makes it possible to handle properties and signals that are beyond the scope of the reachability-based approach. As an example, in our experiments, we have been able to analyze discrete signals with more than 24000 sampling points in seconds, while the approach in [DDM04] was limited to 31 points in hours. We are also able to provide insight on an open question in [DDM04] by observing that saturation does not always imply an improper signal conversion.

The latter can be done by comparing the Fourier transform of each of the input analog signals with the Fourier transform of its corresponding digital signal. Such a property can easily be expressed in our logic and model checked with our statistical-based approach. We are unaware of any other formal verification technique that can solve this problem.

## 2 Statistical Probabilistic Model Checking

The following section introduces the technique of *Statistical Probabilistic Model Checking*. We assume the reader is familiar with elementary concepts in probability theory.

### 2.1 The Probabilistic Model Checking Problem

We use  $Pr(E)$  to denote the probability of event  $E$ . We consider a stochastic system  $\mathcal{S}$  whose executions are *observable* and a property  $\phi$ . We assume that one can decide whether an execution of  $\mathcal{S}$ , denoted by  $\sigma$ , satisfies  $\phi$ . The *Probabilistic Model Checking Problem* consists of deciding whether the executions of  $\mathcal{S}$  satisfy  $\phi$  with a probability greater than or equal to a given threshold  $\theta$ . The latter is denoted by  $\mathcal{S} \models Pr_{\geq\theta}(\phi)$ . This problem is well-defined if and only if one can assign a probability to the set of executions of  $\mathcal{S}$  that satisfy  $\phi$ . One way to solve the Probabilistic Model Checking Problem is to use a numerical approach (see the introduction). The drawback with such an approach is that it computes the probability for all the executions of the system and may not scale up for systems of large size. Another way to solve the probabilistic Model Checking problem is to use a *statistical model checking algorithm*. In the rest of this section, we recap the statistical Model Checking technique proposed by Younes in [You05, You06].

### 2.2 Statistical Approach

The approach in [You05, You06] is based on hypothesis testing. The idea is to check the property  $\phi$  on a sample set of simulations and to decide whether the system satisfies  $Pr_{\geq\theta}(\phi)$  based on the number of executions for which  $\phi$  holds compared to the total number of executions in the sample set. With such an approach, we do not need to consider all the executions of the system. To determine whether  $\mathcal{S}$  satisfies  $\phi$  with a probability  $p \geq \theta$ , we can test the hypothesis  $H : p \geq \theta$  against  $K : p < \theta$ . A test-based solution does not guarantee a correct result but it is possible to bound the probability of making an error. The *strength*  $(\alpha, \beta)$  of a test is determined by two parameters,  $\alpha$  and  $\beta$ , such that the probability of accepting  $K$  (respectively,  $H$ ) when  $H$  (respectively,  $K$ ) holds, called a Type-I error (respectively, a Type-II error) is less or equal to  $\alpha$  (respectively,  $\beta$ ).

A test has *ideal performance* if the probability of the Type-I error (respectively, Type-II error) is exactly  $\alpha$  (respectively,  $\beta$ ). However, these requirements

make it impossible to ensure a low probability for both types of errors simultaneously (see [You05] for details). A solution to this problem is to relax the test by working with an *indifference region*  $(p_1, p_0)$  with  $p_0 \geq p_1$  ( $p_0 - p_1$  is the *size of the region*). In this context, we test the hypothesis  $H_0 : p \geq p_0$  against  $H_1 : p \leq p_1$  instead of  $H$  against  $K$ . If the value of  $p$  is between  $p_1$  and  $p_0$  (the indifference region), then we say that the probability is sufficiently close to  $\theta$  so that we are indifferent with respect to which of the two hypotheses  $K$  or  $H$  is accepted. The threshold  $p_0$  and  $p_1$  are generally defined in term of the single threshold  $\theta$ , e.g.,  $p_1 = \theta - \delta$  and  $p_0 = \theta + \delta$ .

### 2.3 An Algorithmic Scheme

Younes proposed a procedure to test  $H_0 : p \geq p_0$  against  $H_1 : p \leq p_1$  that is based on the *sequential probability ratio test* proposed by Wald [Wal45]. The approach is briefly described below.

Let  $B_i$  be a discrete random variable with a Bernoulli distribution. Such a variable can only take 2 values 0 and 1 with  $Pr[B_i = 1] = p$  and  $Pr[B_i = 0] = 1 - p$ . In our context, each variable  $B_i$  is associated with one simulation of the system. The outcome for  $B_i$ , denoted  $b_i$ , is 1 if the simulation satisfies  $\phi$  and 0 otherwise. In the sequential probability ratio test, one has to choose two values  $A$  and  $B$ , with  $A > B$ . These two values should be chosen to ensure that the strength of the test is respected. Let  $m$  be the number of observations that have been made so far. The test is based on the following quotient:

$$\frac{p_{1m}}{p_{0m}} = \prod_{i=1}^m \frac{Pr(B_i = b_i | p = p_1)}{Pr(B_i = b_i | p = p_0)} = \frac{p_1^{d_m} (1 - p_1)^{m - d_m}}{p_0^{d_m} (1 - p_0)^{m - d_m}}, \quad (1)$$

where  $d_m = \sum_{i=1}^m b_i$ . The idea behind the test is to accept  $H_0$  if  $\frac{p_{1m}}{p_{0m}} \geq A$ , and  $H_1$  if  $\frac{p_{1m}}{p_{0m}} \leq B$ . An algorithm for sequential ratio testing consists of computing  $\frac{p_{1m}}{p_{0m}}$  for successive values of  $m$  until either  $H_0$  or  $H_1$  is satisfied. This has the advantage of minimizing the number of simulations. In each step  $i$ , the algorithm has to check the property on a single execution of the system, which is handled with a new Bernoulli variable  $B_i$  whose realization is  $b_i$ . In his thesis [You05], Younes proposed a logarithmic based algorithm (Algorithm 2.3 page 27) SPRT that given  $p_0, p_1, \alpha$  and  $\beta$  implements the sequential ratio testing procedure. Computing ideal values  $A_{id}$  and  $B_{id}$  for  $A$  and  $B$  in order to make sure that we are working with a test of strength  $(\alpha, \beta)$  is a laborious procedure (see Section 3.4 of [Wal45]). In his seminal paper [Wal45], Wald showed that if one defines  $A_{id} \geq A = \frac{(1-\beta)}{\alpha}$  and  $B_{id} \leq B = \frac{\beta}{(1-\alpha)}$ , then we obtain a new test whose strength is  $(\alpha', \beta')$ , but such that  $\alpha' + \beta' \leq \alpha + \beta$ , meaning that either  $\alpha' \leq \alpha$  or  $\beta' \leq \beta$ . In practice, we often find that both inequalities hold.

The SPRT algorithm can be extended to handle Boolean combinations of probabilistic properties as well as much more complicated probabilistic Model checking problems than the one considered in this paper [You05].

### 3 Signals, Systems and Logics

#### 3.1 Signals Definition

We use  $\mathbb{N}$ ,  $\mathbb{R}$ , and  $\mathbb{C}$  to denote the sets of natural, real, and complex numbers, respectively. Let the *time set*  $\mathcal{T}$  be a finite set of non-negative reals  $\{t_0, t_1, \dots, t_{N-1}\}$ , where  $N \in \mathbb{N}$ . To simplify the presentation, we assume that  $t_{i+1} - t_i = \delta t$ , where  $\delta t \in \mathbb{R}_{>0}$ . A *digital set* is a set consisting of  $2^b$  elements, which can be encoded in terms of  $b$  bits. A *frequency set* is a subset of  $\mathbb{R}$ . An *analog signal* is a mapping  $\xi : \mathcal{T} \rightarrow \mathbb{R}$ . A *digital signal* is a mapping  $\xi : \mathcal{T} \rightarrow \mathcal{D}$ , where  $\mathcal{D}$  is a digital set. A *frequency-domain signal* is a mapping  $\hat{\xi} : \mathcal{F} \rightarrow \mathbb{C}$ , where  $\mathcal{F}$  is a frequency set. The value at time  $t \in \mathcal{T}$  of a signal  $\xi$  is denoted by  $\xi[t]$ . Let  $t, t' \in \mathcal{T}$ , the *restriction* of a signal  $\xi$  to  $[t, t']$ , denoted by  $\xi_{|[t, t']}$ , is a signal such that:

$$\xi_{|[t, t']}[\tau] = \begin{cases} \xi[\tau] & \text{if } \tau \in [t, t'] \\ 0 & \text{else.} \end{cases}$$

The restriction of a frequency-domain signal to an interval of frequencies is defined similarly.

The *Fourier transform* (see [Smi97]) is a functional  $F$  that maps a time-domain signal  $\xi : \mathcal{T} \rightarrow \mathbb{R}$  to a *frequency-domain signal*  $\hat{\xi} = F(\xi)$ . The *inverse Fourier transform* is used to “reconstruct”  $\xi$  from  $\hat{\xi}$ , i.e.,  $\xi = F^{-1}(\hat{\xi})$ . Formally, for all  $\nu$  in  $\mathcal{F}$  and for all  $t$  in  $\mathcal{T}$  we have

$$F(\xi)[\nu] = \int_{\mathcal{T}} \xi[t] e^{-i2\pi\nu t} dt \quad \text{and} \quad F^{-1}(\hat{\xi})[t] = \xi[t] = \int_{\mathcal{F}} \hat{\xi}[\nu] e^{i2\pi\nu t} d\nu.$$

An efficient algorithm known as the *Fast Fourier Transform algorithm* (see, e.g., [FJ97]) is used to compute a discrete approximation of the Fourier transform.

#### 3.2 Model

Our main motivation is to verify properties of mixed-signal circuits. For this purpose, we define *stochastic signal discrete-time event systems*, which extend the classical stochastic discrete-time event systems with information about signals. During an execution, these systems have to remain in the same state between the occurrence of two events. The signals associated with each execution are thus piecewise-constant.

**Definition 1.** Let  $\mathcal{B}$  be a finite set of Boolean propositions. A stochastic signal discrete-time event system (*SSDES*) is a tuple  $\mathcal{S} = (\mathcal{T}, S, s_0, \rightarrow, \pi_a, \pi_d, L)$  where

- $\mathcal{T}$  is a finite set of non-negative reals  $\{t_0, t_1, \dots, t_{N-1}\}$ , with  $t_{i+1} - t_i = \delta t$ ;
- $S$  is the set of states, defined as  $S = A_s \times D_s$ , where  $A_s \subset \mathbb{R}^{n_a}$  and  $D_s \subset \mathcal{D}^{n_d}$ ,  $n_a$  and  $n_d$  being the number of analog and digital signals associated with  $\mathcal{S}$ , respectively. These signals will be denoted by  $\xi_a^1, \dots, \xi_a^{n_a}$  and  $\xi_d^1, \dots, \xi_d^{n_d}$ ;
- $s_0 \in S$  is the initial state;

- The relation  $\rightarrow: S \times S$  is the transition relation of the system. We assume a probability distribution on  $\rightarrow$ , i.e.,

$$\forall s \in S, \sum_{s' \in S} Pr(s \rightarrow s') = 1;$$

Our model is assumed to have the Markovian property;

- $\pi_a: S \times \{1, \dots, n_a\} \rightarrow A_s$  is a projection operator such that for all  $s = (s_a^1, \dots, s_a^{n_a}, s_d^1, \dots, s_d^{n_d})$  and  $1 \leq j \leq n_a$ ,  $\pi_a(s, j) = s_a^j$ ;
- $\pi_d$  is defined in a similar manner to  $\pi_a$ .
- $L$  is a mapping from  $S$  to  $2^{\mathcal{B}}$ , which assigns to each state the elements in  $\mathcal{B}$  that are true in that state. If  $p \in L(s)$ , then we say that  $s$  satisfies  $p$ .

Let  $\omega = s_1 \dots s_k$  be a finite sequence of states of  $\mathcal{S}$ . We use  $\omega(i)$  and  $\omega^i$  to denote the  $i$ -th state of  $\omega$  and the sequence  $s_i \dots s_k$ , respectively. The length of  $\omega$ , denoted  $|\omega|$ , is the number of states in  $\omega$ . An *execution* of an SSDES  $\mathcal{S} = (\mathcal{T}, S, s_0, \rightarrow, \pi_a, \pi_d, L)$  is a sequence of  $N$  states  $\sigma = s_0 s_1 \dots s_{N-1}$  such that for each  $i \in 0 \dots N-1$ ,  $s_i \in S$  and  $s_i \rightarrow s_{i+1}$ . Each state  $s_k$  (with  $k < N$ ) of  $\sigma$  assigns to each analog signal  $\xi_a^i$  (respectively, digital signal  $\xi_d^i$ ) its constant value between  $t_k$  and  $t_{k+1}$ , i.e.,  $\xi_a^i[t] = \pi_a(s_k, i)$  (respectively,  $\xi_d^i[t] = \pi_d(s_k, i)$ ) for  $t \in [t_k, t_{k+1}]$ . The  $i$ -th *suffix* of  $\sigma$  is the sequence  $s_i, \dots, s_{N-1}$ . An SSDES is thus an infinite-state Markov Chain equipped with information and operations on analog and digital signals.

### 3.3 Probabilistic Signal Linear Temporal Logic

We introduce the *probabilistic signal linear temporal logic* (SLTL) to reason on the set of executions of an SSDES. In the rest of the section, we assume a set of atomic propositions  $\mathcal{B}$  and an SSDES  $\mathcal{S} = (\mathcal{T}, S, S_0, \rightarrow, \pi_a, \pi_d, L)$  with  $L$  being a mapping from the set of states  $S$  to  $2^{\mathcal{B}}$ . Before introducing SLTL, we first recall the syntax and the semantics for linear temporal logic (LTL). The syntax of LTL is given by the following grammar:

$$\phi ::= \mathbf{T} \mid \mathbf{F} \mid b \in \mathcal{B} \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \neg \phi \mid \bigcirc \phi \mid \phi_1 \mathcal{U} \phi_2 \mid \phi_1 \tilde{\mathcal{U}} \phi_2.$$

We now present the semantics of LTL, which here is defined with respect to finite sequences of states of  $\mathcal{S}$ . The fact that a finite sequence of states  $\omega$  of  $\mathcal{S}$  satisfies the LTL property  $\phi$  is denoted by  $\omega \models \phi$ . We have the following:

- $\omega \models \mathbf{T}$  and  $\omega \not\models \mathbf{F}$ ;
- $\omega \models b$  with  $b \in \mathcal{B}$  if and only if  $b \in L(\omega(0))$ ;
- $\omega \models \phi_1 \vee \phi_2$  if and only if  $\omega \models \phi_1$  or  $\omega \models \phi_2$ ;
- $\omega \models \phi_1 \wedge \phi_2$  if and only if  $\omega \models \phi_1$  and  $\omega \models \phi_2$ ;
- $\omega \models \neg \phi$  if and only if  $\omega \not\models \phi$ .
- $\omega \models \bigcirc \phi$  if and only if  $|\omega| > 1$  and  $\omega^1 \models \phi$ ;
- $\omega \models \phi_1 \mathcal{U} \phi_2$  if and only if there exists  $0 \leq i \leq |\omega| - 1$  such that  $\omega^i \models \phi_2$ , and for each  $0 \leq j < i$ ,  $\omega^j \models \phi_1$ ;

- $\omega \models \phi_1 \tilde{\mathcal{U}} \phi_2$  if and only if for each  $0 \leq i \leq |\omega| - 1$  such that  $\omega^i \not\models \phi_2$  there exists  $0 \leq j < i$  such that  $\omega^j \models \phi_1$ ;

Two additional temporal operators are used, that are  $\diamond\psi = \mathbf{T}\mathcal{U}\psi$  and  $\square\psi = \mathbf{F}\tilde{\mathcal{U}}\psi$ .

Note that we consider LTL properties on finite executions. Thus, we can only specify bounded LTL properties. As in [You05], we thus stay in the class of safety properties. It is easy to decide whether a finite execution satisfies a LTL formula.

We now introduce the notion of an *execution predicate*.

**Definition 2 (Execution Predicate).** Let  $\Sigma(\mathcal{S})$  be the set of all the executions of an SSDES  $\mathcal{S}$ . An execution predicate  $p$  for  $\mathcal{S}$  is a predicate on  $\Sigma(\mathcal{S})$ .

*Example 1.* Consider an execution predicate  $p$  that decides whether the mean value of the first analog signal associated with an execution  $\sigma$  of an SSDES is greater than 0. Such predicate can be defined as

$$p(\sigma) = \mathbf{T} \quad \text{iff} \quad \frac{1}{N} \sum_{k=0}^{N-1} \pi_a(\sigma(k), 1) \geq 0.$$

This example shows that the rather general definition of execution predicate makes it easy to define properties on entire executions that cannot easily be defined with temporal operators. In Section 5, we will consider a more complex execution predicate that compares the Fourier transforms of two signals.

We add a new clause to the grammar of LTL for execution predicates. Let  $\mathcal{P}$  be a set of execution predicates, our new grammar for LTL is

$$\phi ::= \mathbf{T} \mid \mathbf{F} \mid p \in \mathcal{P} \mid b \in \mathcal{B} \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc\phi \mid \phi_1 \mathcal{U}\phi_2 \mid \phi_1 \tilde{\mathcal{U}}\phi_2,$$

with the restriction that execution predicates cannot be under the scope of temporal operators.

We can now define probabilistic signal linear temporal logic.

**Definition 3 (SLTL Formula).** An SLTL formula is a formula of the form  $\psi = Pr_{\geq\theta}(\phi)$ , where  $\phi$  is a LTL formula with execution predicates.

We say that  $\mathcal{S}$  satisfies  $\psi$ , denoted by  $\mathcal{S} \models \psi$  if and only if the probability for an execution of  $\mathcal{S}$  to satisfy  $\phi$  is greater or equal than  $\theta$ . The problem is well-defined since, as is shown in the following theorem, one can always assign a unique probability measure to the set of executions that satisfy an LTL formula with execution predicates.

**Theorem 1.** Let  $\mathcal{S}$  be an SSDES and  $\phi$  be a LTL property with execution predicates. One can always associate a unique probability measure to the set of executions of  $\mathcal{S}$  that satisfy  $\phi$ .

*Proof.* We first introduce the definition of *history expansion* for a Markov chain.

**Definition 4.** Consider the Markov chain  $\mathcal{S} = (S, s_0, \rightarrow, L)$ ; its *history expansion* is the Markov chain  $\mathcal{S}' = (S', s'_0, \rightarrow', L')$ , where

- Each state in  $S'$  is a prefix of one of the executions of  $\mathcal{S}$ ;
- $s'_0 = s_0$ ;
- The transition relation  $\rightarrow'$  is defined as follows :  $(s_x, s_y) \in \rightarrow'$  iff  $s_x = s_0 \dots s$ ,  $s_y = s_0 \dots ss'$ , and  $(s, s') \in \rightarrow$ . The probability distribution from  $s_x$  is derived from the one defined on  $s$ .
- Given a state  $s = s_0s_1 \dots s_i$  in  $S'$ ,  $L'(s) = L(s_i)$ .

An SSDES is an infinite-state Markov Chain whose executions can be viewed as infinite executions by considering their last state to be an absorbing state, i.e., a state in which the system stays forever. In [You05], it is shown that one can assign a unique probability measure to sets of infinite executions of such a Markov Chain using a *probability space* and the classical notion of *basic cylinder*. In [You05], it is also shown that this probability distribution is sufficient to assign a probability to the set of executions that satisfy an LTL formula without execution predicates. We are now left with the case where  $\phi$  can reference execution predicates  $P_1, \dots, P_n$ . In such situation, we first derive from  $\mathcal{S}$  its corresponding history expansion<sup>1</sup>  $\mathcal{S}'$ . It is easy to see that there is a one-to-one correspondence between the executions of  $\mathcal{S}$  and those of  $\mathcal{S}'$ . We then introduce a new Boolean variable  $p_i$  for each execution predicate  $P_i$ . Given a state  $s_{S'}$  of  $\mathcal{S}'$ , we have  $p_i \in L(s_{S'})$  if and only if (1)  $s_{S'}$  is an execution of  $\mathcal{S}$  and (2) this execution satisfies  $P_i$ . Let  $\phi'$  be the formula  $\phi$  where each Boolean predicate  $P_i$  has been replaced by the LTL formula  $\diamond p_i$ . The formula  $\phi'$  is a LTL formula. Observe also that an execution of  $\mathcal{S}$  satisfies  $\phi$  if and only if its corresponding execution in  $\mathcal{S}'$  satisfies  $\phi'$ . Since  $\phi'$  is a LTL formula, one can always assign a probability to the set of executions of  $\mathcal{S}'$  that satisfy it. By construction, we know that this probability is also the one assigned to the set of executions of  $\mathcal{S}$  that satisfy  $\phi$ .

Assuming that we are only working with execution predicates that we can compute, we observe that SSDES and SLTL are in the scope of the class of systems and logics that can be handled with the SPRT algorithm. In our experiments, we will thus use the statistical model checking approach proposed by Younes for verifying SLTL properties of SSDESs.

## 4 A Class of Mixed-Signal Circuits: $\Delta - \Sigma$ Modulators

This section is a brief introduction to the principles of  $\Delta - \Sigma$  modulation and the related design issues. The reader can consult [MPVRV01] for more details on this topic in Signal Processing.

<sup>1</sup> Note that we only use this construction for the purpose of the proof. For our experiments, we only need to generate executions of  $\mathcal{S}$  which can be done without a complete representation of either  $\mathcal{S}$  or  $\mathcal{S}'$ .

## 4.1 Analog to Digital Conversion via $\Delta - \Sigma$ Modulation

A  $\Delta - \Sigma$  modulator is an *Analog-to-Digital Converter circuit*, i.e., a circuit that takes an analog value  $u \in \mathbb{R}$  as input and encodes it into a digital value  $v \in \mathcal{D}$ . Since digital signal processing is more widely used than analog signal processing, such converters are found in many electrical devices, which motivates their study. The challenge with Analog-to-Digital conversion is to represent the uncountable set of analog values using a finite set of digital values  $\mathcal{D}$ . The direct approach, which is called *quantization*, consists in mapping  $u$  to the digital value  $v$  that minimizes the *quantization error* defined as  $\delta = u - v$ , i.e., it chooses  $v = \operatorname{argmin}_{v \in \mathcal{D}} |\delta|$ . Obviously, one way to decrease the remaining quantization error is to increase the number of bits used to encode  $\mathcal{D}$  and thus the number of possible digital values. Another approach, which is implemented by  $\Delta - \Sigma$  modulation, is to measure and compensate for the accumulation of quantization errors during time. As an example, consider the following simple instance of a discrete time  $\Delta - \Sigma$  modulator. Let  $u(k)$ ,  $v(k)$ ,  $\delta(k) = u(k) - v(k)$  be the analog input, the digital output, and the quantization error at step  $k$ , respectively. The modulator uses an *integrator* to store the accumulation of errors in a variable  $x(k) = \sum_0^k \delta(k)$ , so that  $x(k+1) = x(k) + \delta(k)$ , and determines the next digital output  $v(k+1)$  based on the sign of  $x(k+1)$ , i.e.,  $\mathcal{D} = \{-1, 1\}$  and  $v(k+1) = 1$  if  $x(k+1) \geq 0$  and  $v(k+1) = -1$  otherwise. A  $\Delta - \Sigma$  modulator thus basically consists of a feedback loop controlling the quantization error. To improve the performance, more complex feedback loops can be designed involving more than one integrator. The *order* of a modulator is given by the number of integrators used.

The benefit of the  $\Delta - \Sigma$  modulation approach is clearly apparent in the frequency domain. Indeed, the Fourier transform of the digital signal is the Fourier transform of the analog signal composed with some error due to the quantization. The feedback loop in the  $\Delta - \Sigma$  modulator is designed to “push” this error towards high frequencies, where it can be isolated and removed, e.g. by using a low-pass filter. The original signal can then be retrieved by using the inverse Fourier transform (see appendix A for examples of behaviors of a modulator). Note that  $\Delta - \Sigma$  modulators can achieve good performance using a limited number of bits.

## 4.2 Verification Issues

Modulators with more than two integrators are known to exhibit better performance but also introduce a *stability* issue [ASS96]. An integrator memorizes its input and adds it to the sum of all the previously read inputs during the execution. Consequently, an important issue is whether the integrators are stable, i.e., whether or not the values stored in the integrators can grow indefinitely. Because integrators have limited capacity, the values of these states would then reach a *saturation level*. Saturation can compromise the quality of the analog-to-digital conversion. The stability analysis of the feedback loop is made difficult by the nonlinearity (in this case, a discontinuity) induced by quantization. This

invalidates the direct application of classical linear stability theory which makes the stability analysis of  $\Delta - \Sigma$  modulators a challenging problem (see [SH93]). In the next section, we investigate several issues related to stability by using a statistical Model Checking approach.

## 5 Experimental Results

We implemented a prototype in the MATLAB environment. Our procedure takes as input a Simulink model and a property  $\phi$  that is a LTL formula with execution predicates. To apply our statistical approach, we combine the Simulink model with a stochastic input generator. At each time instant  $t_i$ , this generator randomly chooses an input value for the analog signal and the Simulink engine uses this value to compute the next state of the system. The result is an SSDES whose executions can easily be observed without building the entire state-space of the system.

We now discuss the experimental results we obtained when applying our prototype to a third-order  $\Delta - \Sigma$  modulator. We start with the encoding of the simulator into a SSDES model.

### 5.1 SSDES for a Third Order Modulator

We work with the instance of a third order  $\Delta - \Sigma$  modulator that was considered in [DDM04]. A Simulink model is available in appendix A (a full description can be found in [DDM04]). It is combined with a stochastic input generator to give an SSDES  $\mathcal{S}=(\mathcal{T}, S, s_0, \rightarrow, \pi_a, \pi_d, L)$ , where

- **Time.** We set  $\mathcal{T} = \{t_0, t_1, \dots, t_{N-1}\}$  with  $t_0 = 0, t_{N-1} = 3$  and  $\delta t = t_{i+1} - t_i = \frac{1}{8000}, N = 24000$ .
- **Set of States.** The Simulink model contains three integrators such that each contains one real-valued (or analog) variable. A state  $s \in S$  can thus be described as a tuple  $(u, x_1, x_2, x_3, v)$ , where
  - $x_1, x_2$  and  $x_3$  are analog variables storing the integrators' states;
  - $u$  is an analog variable storing values for the input signal  $\xi^u$ ;
  - $v$  is a digital variable storing values for the output signal  $\xi^v$ .

The number of analog signals is thus  $n_a = 4$  and the number of digital signals  $n_d = 1$ . We assume that the states of the integrators cannot go beyond certain values that are fixed by the model. When this value is reached, we say that the integrators saturate. In practice,  $x_i \in [-1, 1]$  for  $i \in \{1, 2, 3\}$  and  $-1, 1$  are the *saturation values*. Assuming also that  $u \in [-u_{\max}, u_{\max}]$ , we get  $A_s = [-1, 1]^3 \times [-u_{\max}, u_{\max}]$  and  $D_s = \{-1, 1\}$ . Given an execution  $\sigma = s_0 s_1 \dots s_{N-1}$ , we use  $u(k) = \pi_a(s_k, 1), x_1(k) = \pi_a(s_k, 2), x_2(k) = \pi_a(s_k, 3), x_3(k) = \pi_a(s_k, 1)$  and  $v(k) = \pi_d(s_k, 1)$ . For all  $k \in \{0, \dots, N-1\}$ , we have  $\xi^u[t_k] = u(k)$  and  $\xi^v[t_k] = v(k)$ ;

- **Transition relation.** When  $u(k)$  is given, the Simulink engine computes<sup>2</sup>  $x_1(k+1)$ ,  $x_2(k+1)$ ,  $x_3(k+1)$  and  $v(k+1)$ . Thus the probability distribution  $Pr(s_k \rightarrow s_{k+1})$  for all  $(s_k, s_{k+1}) \in S \times S$  is induced by the probability distribution of the input value  $u(k+1)$ . For our experiments, we consider uniform random inputs: for all  $k$ ,  $u(k)$  is chosen in a set  $[-u_{\max}, u_{\max}]$  with a uniform random distribution;
- **Initial state.** Initially, the values of the integrator states are 0 and by convention the digital output  $v(0)$  is set to 1 and the input value  $u(0)$  to 0. Thus the initial state is  $s_0 = (0, 0, 0, 0, 1)$ ;
- **Boolean variables.** We define a Boolean variable *Satur* which is true iff one of the analog values, i.e., either the input or an integrator state, saturates. Formally,  $L(s) = \mathbf{T}$  iff there exist  $i$  in  $\{1, \dots, 4\}$  such that  $\pi_a(s, i) = 1$  or  $-1$ ,  $L(s) = \mathbf{F}$  otherwise.

The choice of the probability distribution to generate input signals influences the statistical result we obtain. A simple choice is the *uniform* distribution, which gives the same probability for every possible input signal to occur. By doing so, we make as few assumptions as possible on the nature of the input signal. We can thus compare our results with those obtained on the corresponding non-stochastic model.

## 5.2 Experiments

**Saturation** We first considered the formula  $Pr_{\geq \theta}(\diamond Satur)$ , i.e., whether saturation occurs with a probability greater or equal to  $\theta$  for different values of  $u_{\max}$ . We applied the SPRT algorithm for several values<sup>3</sup> of  $\theta$ . We set the two error bounds  $\alpha$  and  $\beta$  to 0.001 and used an indifference region  $(p_1, p_0) = (\theta - 0.01, \theta + 0.01)$ . We tested  $H_0 : p \geq \theta + 0.01$  against  $H_1 : p \leq \theta - 0.01$ . Our results are reported in Table 2. The first and second column report the value of  $u_{\max}$  and the value of  $\theta$  chosen, respectively. Column 3 reports the number of simulations performed.  $H_0$  was rejected for the first line and accepted for the others. Our results show that saturation will occur with probability 1 when the maximum amplitude  $u_{\max}$  of the input signal is greater than 0.3.

In [DDM04] and [GKR04] reachability techniques developed in the area of hybrid systems were used to guarantee that for *every* input signal in a given range, the integrator state will never saturate. While this approach is clearly sound for proving stability, its computational cost is prohibitive. As an example, in [DDM04], stability was only proved for a small number of steps, i.e.,  $N = 31$ .

<sup>2</sup> The way this computation is performed can be deduced from the Simulink model given in Appendix A.

<sup>3</sup> The values for  $u_{\max} = 0.1$  and  $u_{\max} = 0.3$  were chosen to validate the experiments in [DDM04], while the others were chosen with some trial and error process to get closer to true probability.

$u_{\max}$	Probability $\theta$ checked	Number of exec.
0.1	0	416
0.15	0.09	4967
0.2	0.64	17815
0.25	0.98	416
0.3	1	688

**Table 1.** Table of results for  $Pr_{\geq\theta}(\diamond Satur)$ .  $H_0$  was rejected for the first line and accepted for the others.

Our results can be compared with those reported<sup>4</sup> in [DDM04]. In particular, we confirmed the fact that for signals with a maximum amplitude of 0.1, the circuit never saturates whereas if  $u_{\max}$  is more than 0.3, the circuit always does. In our case, though, the length  $N$  of the executions considered was much larger.

**Frequency Domain Predicate** In addition to improving the computation time, our approach makes it possible to verify more complex properties than those that can be handled with a reachability-based technique. In particular, by defining execution predicates involving the Fourier transform, we can check reliably whether an analog signal was properly converted to a digital one. We can also investigate the relation between saturation and wrong behaviors of the modulator without assuming a priori, as is the case in [DDM04], that the latter implies the former.

We checked the formula  $Pr_{\geq\theta}(p_F)$ , where  $p_F$  is a frequency-domain execution predicate that compares the Fourier transform of the input analog signal  $u$  with the one of its corresponding digital signal  $v$ . Formally,  $p_F$  is defined as follows. Let  $d_F$  be a metric on frequency-domain signals such that for two signals  $\hat{\xi}_1$  and  $\hat{\xi}_2$ ,

$$d_F(\hat{\xi}_1, \hat{\xi}_2) = \frac{1}{N} \sum_{0 \leq k \leq N-1} |\hat{\xi}_1[\nu_k] - \hat{\xi}_2[\nu_k]|. \quad (2)$$

Let  $\sigma$  be an execution of  $\mathcal{S}$ . The value of the execution predicate  $p_F$  on  $\sigma$  is given by

$$p_F(\sigma) = \mathbf{T} \text{ iff } d_F(\hat{\xi}_{[0,\nu]}^u, \hat{\xi}_{[0,\nu]}^v) \leq \epsilon,$$

where  $\hat{\xi}^u$  and  $\hat{\xi}^v$  are the Fourier transforms of the input analog signal  $u$  and its corresponding digital signal  $v$ , respectively. It is easy to derive a MATLAB routine that can decide whether or not an execution provided by Simulink satisfies  $p_F$ . We worked with  $\nu = 100Hz$  and  $\epsilon = 0.1$ , since we observed that for those values the predicate efficiently discriminates between executions for which

<sup>4</sup> Recall that the results in [DDM04] are obtained from the Simulink model, while we work with the corresponding stochastic model.

the digital output has a correct Fourier transform (see Figure 2 of Appendix B) against executions when this is not the case (see Figure 3 of Appendix B). We used the same indifference region and the same error types as in the previous experiments.

In our experiments, which are reported in Table 2, we observed that  $p_F$  is true with probability  $\geq 1$  for  $u_{\max} = 0.8$  and that this probability decreases when the value of  $u_{\max}$  increases<sup>5</sup>. This means that saturation does not always imply a wrong behavior. Indeed, as an example, for values of  $u_{\max}$  greater than 0.3, the property  $Pr_{\geq 1}(\diamond Satur)$  holds (see previous experiment) and for values of  $u_{\max}$  smaller than 0.8, the property  $Pr_{\geq 1}(p_F)$  also holds. We can thus infer that between 0.3 and 0.8, the property  $\diamond Satur \wedge p_F$  holds with probability 1. In [DDM04], it is assumed that the absence of saturation is necessary for  $p_F$  to be true. Our experiments show that this may be an overly conservative assumption.

$u_{\max}$	Probability $\theta$ checked	Number of exec.
0.8	1.	688
0.9	0.98	612
1.0	0.98	1248
1.1	0.875	6388
1.2	0.575	15507

**Table 2.** Table of results for  $Pr_{\geq \theta}(p_F)$ .  $H_0$  was accepted for each experiment.

**Additional Experiments** Finally, we performed a few experiments on characterizing the computation time with respect to the strength parameters  $\alpha$  and  $\beta$ , and the size of the indifference region. These experiments show that the number of simulations needed by the algorithm increases logarithmically with respect to the decrease of  $\alpha$  and  $\beta$  and linearly with respect to the decrease of  $p_0 - p_1$  (see Table 3). This indicates that one can verify that  $Pr(\mathcal{S} \models \phi) \geq \theta$  with a very low probability of error whereas it is more difficult to estimate precisely the actual value of  $p$  by narrowing the indifference region. These results corroborate those reported in [You05].

## 6 Future Work

This paper presents the first attempt to apply the statistical Model Checking techniques introduced in [You05,YS06] to verifying non-trivial properties of

<sup>5</sup> We also observed that for values of  $u_{\max}$  greater or equal to 0.9, the probability for a good conversion to occur was strictly inferior to 1. The values of  $\theta$  reported in lines 2 – 5 of Table 2 have been found with some trial and error process.

Test strength $\alpha(=\beta)$	Number of executions	Indifference region $p_0 - p_1$	Number of executions
$1e^{-2}$	335	0.1	55
$1e^{-4}$	502	0.05	106
$1e^{-6}$	857	0.02	228
$1e^{-8}$	1301	0.01	627
$1e^{-10}$	1467	0.005	1056

Number of trajectories against  $\alpha$  ( $\beta$  was set equal to  $\alpha$  and  $p_0 - p_1 = 0.02$ ).

Number of trajectories against the size of the indifference region  $p_0 - p_1$  ( $\alpha = \beta = 0.02$ ).

**Table 3.** Computational costs for different test strengths and different indifference regions for  $Pr_{\geq 0.875}(p_F)$ .

mixed-signal circuits. In comparison to [DDM04], our technique allows us to obtain better performance results as well as to handle a larger class of properties. Our results are correct up to a prespecified probability of an error, while those of [DDM04] are exact.

Our work requires the ability to monitor properties of discrete-time signals, which can easily be done with existing techniques [LS06,dR]. In a series of recent papers [NM07,MNP08], Nickovic et al. proposed techniques for monitoring properties of *dense-time* analog signals. An interesting direction would be to adapt the procedure of Younes to work in this latter, more demanding context.

In our experiments, the choice of the value for  $\theta$  has been driven by the previous observations reported in [DDM04]. In future work, we plan to use the estimation-based method of [?] to approximate the value of  $\theta$  for which the property holds.

We also intend to consider extensions of SLTL incorporating past temporal operators [?] and a better correlation between execution predicates and temporal operators. We plan to define more complex specifications for frequency domain properties based on the needs of designers of mixed signal circuits. Our ultimate goal is to provide them with a general framework for specifying and verifying properties of mixed-signal circuits.

## Acknowledgements

We thank H. Younes for answering many email questions on his work and C. J. Langmead for fruitful discussions and proof-reading drafts of the paper. We also thank the participants and the referees of the *Formal Verification of Ana-*

*log Circuits workshop 2008* (where a preliminary version of this work has been presented) for their helpful comments and suggestions.

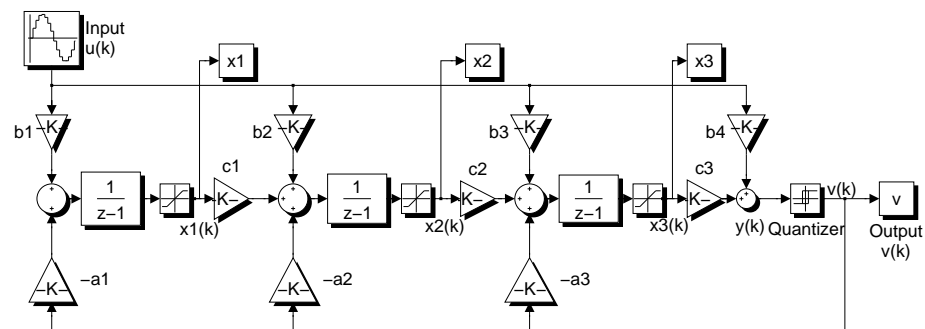
## References

- ASS96. Pervez M. Aziz, Henrik V. Sorensen, and Jan Van Der Spiegel. An overview of sigma-delta converters. *IEEE Signal Processing Magazine*, pages 61–84, January 1996.
- BHHK03. Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.
- CB06. F. Ciesinski and C. Baier. Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems. In *QEST*, pages 131–132. IEEE, 2006.
- CG04. F. Ciesinski and M. Größer. On probabilistic computation tree logic. In *Validation of Stochastic Systems*, LNCS, 2925, pages 147–188. Springer, 2004.
- CY95. Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
- DDM04. T. Dang, A. Donze, and O. Maler. Verification of analog and mixed-signal circuits using hybrid systems techniques. In Alan J. Hu and Andrew K. Martin, editors, *FMCAD’04 - Formal Methods for Computer Aided Design*, LNCS 3312, pages 21–36. Springer-Verlag, 2004.
- dR. Marcelo d’Amorim and Grigore Roşu. Efficient monitoring of  $\omega$ -languages. In *CAV*, LNCS 3576, pages 364 – 378. Springer.
- FJ97. Matteo Frigo and Steven G. Johnson. The fastest Fourier transform in the west. Technical Report MIT-LCS-TR-728, Massachusetts Institute of Technology, September 1997.
- GKR04. Smriti Gupta, Bruce H. Krogh, and Rob A. Rutenbar. Towards formal verification of analog designs. In *ICCAD*, pages 210–217, 2004.
- KNP04. M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism 2.0: A tool for probabilistic model checking. In *QEST*, pages 322–323. IEEE, 2004.
- LS06. Andreas Bauer 0002, Martin Leucker, and Christian Schallhart. Monitoring of real-time properties. In *FSTTCS*, volume 4337 of *LNCS 4337*, pages 260–272. Springer, 2006.
- MNP08. Oded Maler, Dejan Nickovic, and Amir Pnueli. Checking temporal properties of discrete, timed and continuous behaviors. In *Pillars of Computer Science*, pages 475–505, 2008.
- MPVRV01. Fernando Medeiro, Belen Pérez-Verdú, and Angel Rodríguez-Vázquez. *Top-Down Design of High-Performance Sigma-Delta Modulators*, chapter 2. Kluwer Academic Publishers, 2001.
- NM07. Dejan Nickovic and Oded Maler. Amt: A property-based monitoring tool for analog systems. In *FORMATS*, pages 304–319, 2007.
- SH93. Avideh Zakhor Soren Hein. On the stability of sigma delta modulators. *IEEE Transactions on Signal Processing*, 41, July 1993.
- Smi97. Steven W. Smith. *The scientist and engineer’s guide to digital signal processing*. California Technical Publishing, San Diego, CA, USA, 1997.
- SVA04. Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In *CAV*, LNCS 3114, pages 202–215. Springer, 2004.

- SVA05. Koushik Sen, Mahesh Viswanathan, and Gul Agha. On statistical model checking of stochastic systems. In *CAV*, LNCS 3576, pages 266–280, 2005.
- Wal45. A. Wald. sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.
- YKNP06. Håkan L. S. Younes, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Numerical vs. statistical probabilistic model checking. *STTT*, 8(3):216–228, 2006.
- You05. Håkan L. S. Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Carnegie Mellon, 2005.
- You06. Håkan L. S. Younes. Error control for probabilistic model checking. In *VMCAI*, LNCS 3855, pages 142–156. springer-verlag, 2006.
- YS06. Håkan L. S. Younes and Reid G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9):1368–1409, 2006.

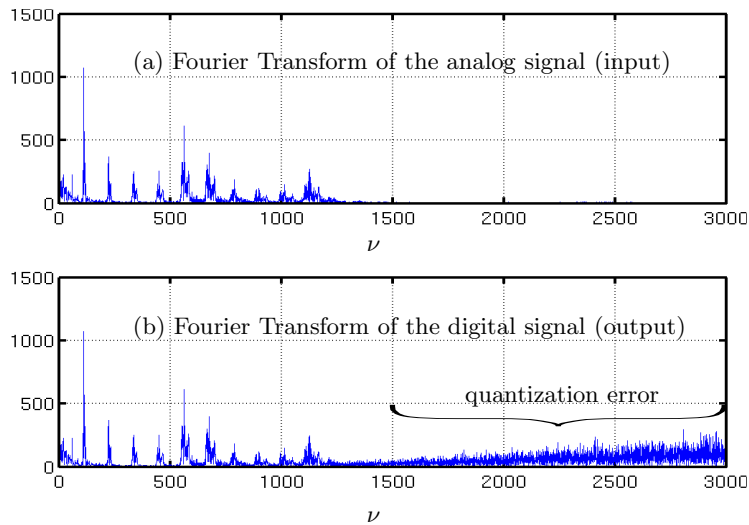
## A Simulink Model of the $\Delta - \Sigma$ Modulator

This appendix contains additional figures and details concerning the  $\Delta - \Sigma$  modulator studied in the experiments. More details can be found in [DDM04].

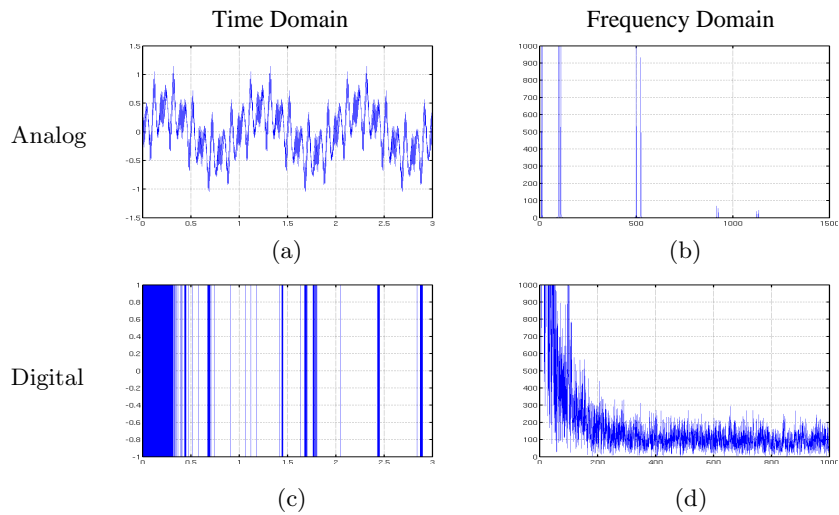


**Fig. 1.** Simulink model of a third order  $\Delta - \Sigma$  modulator. The three blocks  $\frac{1}{z-1}$  followed by saturation blocks represent the saturated integrators. The values of the coefficients  $a_i, b_i$  and  $c_i$  were obtained using the `delsig` toolbox. They are  $a_1 = b_1 = 0.0440$ ,  $a_2 = b_2 = 0.2881$ ,  $a_3 = b_3 = 0.7997$ ,  $b_4 = 1$ , and  $c_1 = c_2 = c_3 = 1$ ;  $x_1, x_2$  and  $x_3$  are the analog variables storing the integrators' states.

## B Experiment Illustrating Frequency Domain Predicates



**Fig. 2.** A sample behavior of the  $\Delta-\Sigma$  modulator. The Fourier transform of the output signal (b) matches the Fourier transform of the input signal (a) on the interval  $[0, 1500Hz]$ . The quantization error is pushed toward frequencies higher than  $1500Hz$ .



**Fig. 3.** An example where the  $\Delta-\Sigma$  fails. We observe that the Fourier transform of the digital signal (d) is clearly different from the Fourier transform of the analog signal (b).