Statistical Model Checking of Mixed-Signal Circuits with an Application to $\Delta - \Sigma$ Modulators

Alexandre Donzé

Joint work with

Edmund M. Clarke and Axel Legay

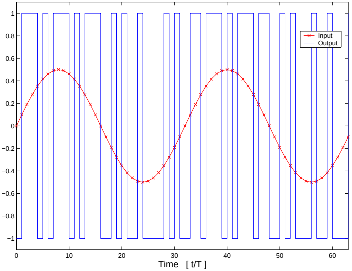
School of Computer Science, Carnegie Mellon University

October 27, 2008

$\Delta - \Sigma$ Modulators for Dummies

Analog to Digital converters (ADC)

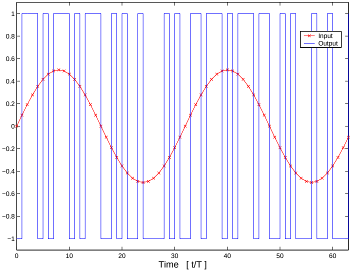
- Converts analog signals into digital signals
- Used in many electrical devices interfacing with a physical environment



$\Delta - \Sigma$ Modulators for Dummies

Analog to Digital converters (ADC)

- Converts analog signals into digital signals
- Used in many electrical devices interfacing with a physical environment

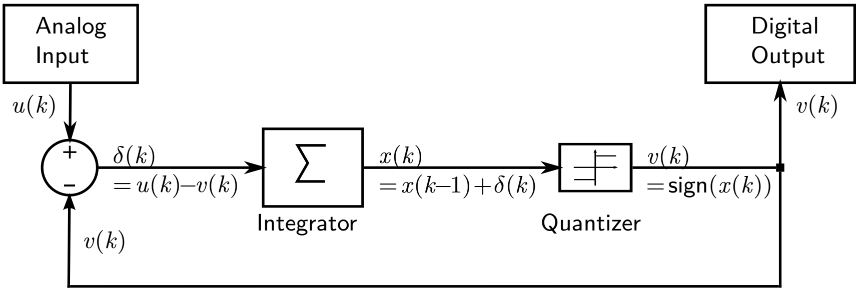


$\Delta - \Sigma$ *modulators*

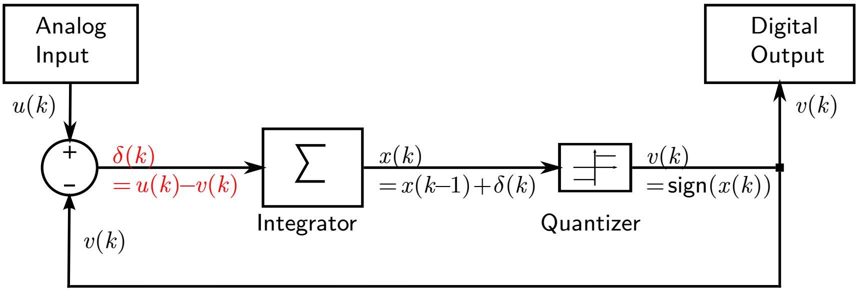
- Widely used family of ADCs
- Efficient processing of the *quantization error*, i.e., the difference between the analog input and the digital output

Principle Control of quantization error using a feedback loop

Principle Control of quantization error using a feedback loop

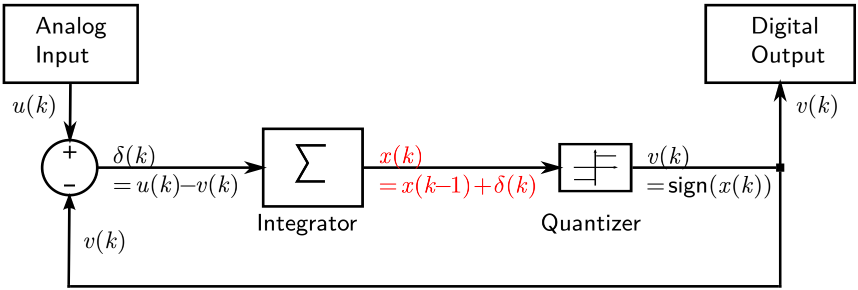


Principle Control of quantization error using a feedback loop



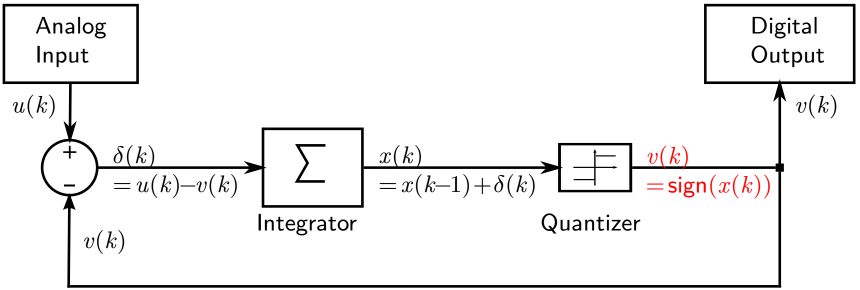
- The quantization error is the difference between the input and the output

Principle Control of quantization error using a feedback loop



- The quantization error is the difference between the input and the output
- The integrator stores the summation of δs in a state variable x

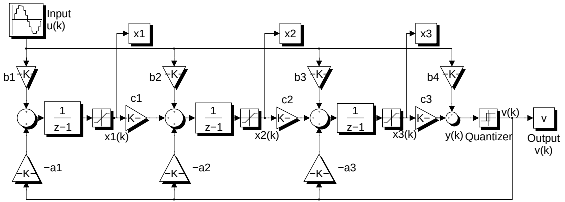
Principle Control of quantization error using a feedback loop



- The quantization error is the difference between the input and the output
- The integrator stores the summation of δs in a state variable x
- The quantizer produces the output based on the sign of x

Higher Order $\Delta - \Sigma$ Modulators

- More complex designs use more than one integrator:



- The *order* of a $\Delta - \Sigma$ modulator is the number of integrators used
- The integrator *stability* becomes an issue when order ≥ 3.
- *Saturation* can compromise the analog to digital conversion

Question For a given design and a class of input signals, how do we verify the correctness of the circuit ?

Model Checking circuit designs

- Mature for digital circuits but still new for analog and mixed-signal
- Difficult due to continuous and hybrid state variables

Model Checking circuit designs

- Mature for digital circuits but still new for analog and mixed-signal
- Difficult due to continuous and hybrid state variables

Probabilistic Model Checking

- We take a *randomized* approach and reduce the problem to a Probabilistic Model Checking (PMC) problem
- Exact solution for PMC problems is still difficult in general

Model Checking circuit designs

- Mature for digital circuits but still new for analog and mixed-signal
- Difficult due to continuous and hybrid state variables

Probabilistic Model Checking

- We take a *randomized* approach and reduce the problem to a Probabilistic Model Checking (PMC) problem
- Exact solution for PMC problems is still difficult in general

Statistical Approach

- Use of numerical simulation
- Provides an approximate solution with error bounds

Outline

Statistical Probabilistic Model Checking

Systems and Logics with Signals

Application to $\Delta - \Sigma$ Modulators

Outline

Statistical Probabilistic Model Checking

Systems and Logics with Signals

Application to $\Delta - \Sigma$ Modulators

Probabilistic Model Checking (PMC)

Given

- a property ϕ
- a *stochastic* system $\mathcal{S}$ for which each execution σ satisfies ϕ with probability p
- a number θ in $[0, 1]$

Do we have

$$p \geq \theta \quad \text{noted:} \quad \mathcal{S} \models Pr_{\geq\theta}(\phi) \quad ?$$

Statistical Approach to PMC: Hypothesis Testing

([Younes et al 02,05,06], [Sen et al 04, 05])

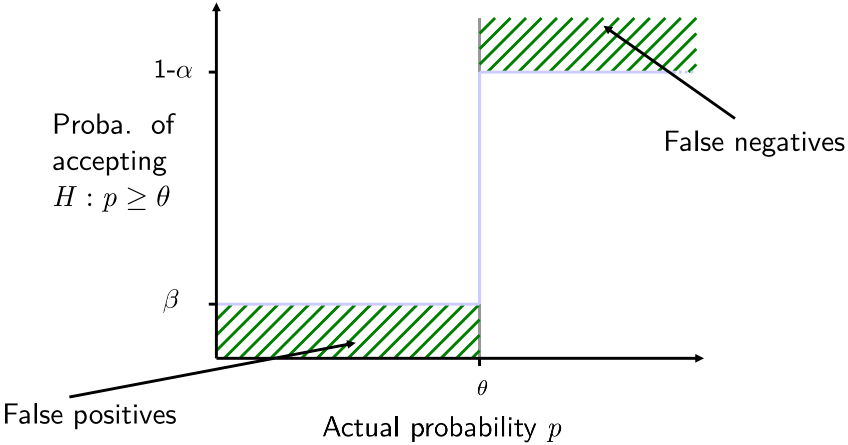
Idea: test hypothesis $H : p \geq \theta$ by

- Generating sample executions σ using simulation
- Verifying ϕ for each σ
- Using sequential acceptance sampling to accept or reject H

While providing *Error bounds* α *and* β such that

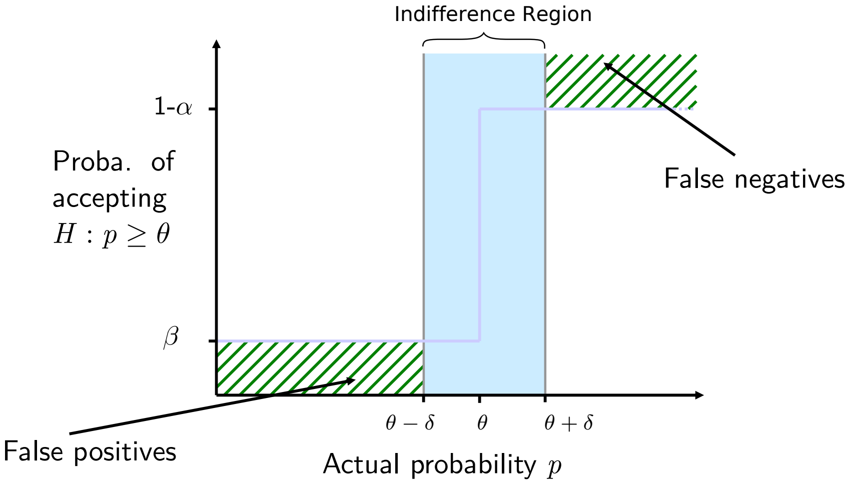
- Probability of false negative (Reject H whereas it is true) $\leq \alpha$
- Probability of false positive (Accept H whereas it is false) $\leq \beta$

Performance of Test



Needs an infinite number of samples to get ideal performances !

Performance of Test



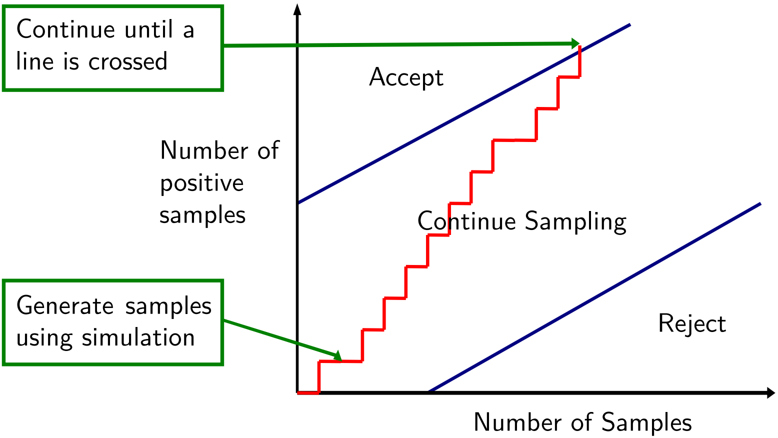
If $p \in [\theta - \delta, \theta + \delta]$, we say we are *indifferent* to know if $p \geq \theta$

Sequential Hypothesis Testing

- Check hypothesis after each sample and stop as soon as possible
- We can find an acceptance line and a rejection line given α, β, θ, δ.

Sequential Hypothesis Testing

- Check hypothesis after each sample and stop as soon as possible
- We can find an acceptance line and a rejection line given $\alpha, \beta, \theta,\ \delta$.



Outline

Statistical Probabilistic Model Checking

Systems and Logics with Signals

Application to $\Delta - \Sigma$ Modulators

Stochastic Signal Discrete Time Event System (SSDES)

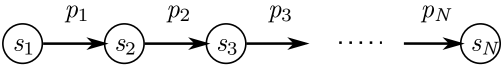
Set of states S and a probability distribution on transitions $s \rightarrow s'$

Stochastic Signal Discrete Time Event System (SSDES)

Set of states S and a probability distribution on transitions $s \rightarrow s'$

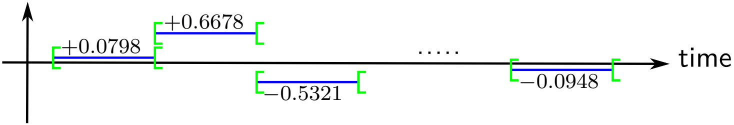
Stochastic Signal Discrete Time Event System (SSDES)

Set of states S and a probability distribution on transitions $s \to s'$



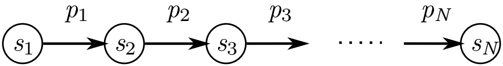
execution σ

Analog signal: $t \in [t_k, t_{k+1}[,\ \xi_a[t] = \pi_a(s_k) \in \mathbb{R}$



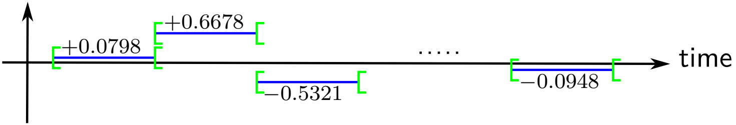
Stochastic Signal Discrete Time Event System (SSDES)

Set of states S and a probability distribution on transitions $s \to s'$

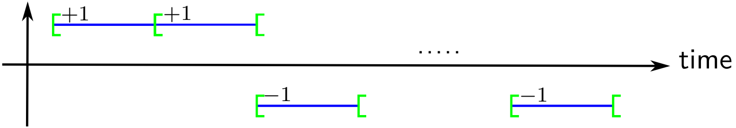


execution σ

Analog signal: $t \in [t_k, t_{k+1}[, \ \xi_a[t] = \pi_a(s_k) \in \mathbb{R}$



Digital signal: $t \in [t_k, t_{k+1}[, \ \xi_d[t] = \pi_d(s_k) \in \{-1, +1\}$



Logics: LTL formulas

Let $\mathcal{B}$ be a set of predicates. The following defines an LTL formula:

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

Logics: LTL formulas

Let $\mathcal{B}$ be a set of predicates. The following defines an LTL formula:

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

Let $\omega = s_1 s_2 ... s_k$, $|\omega| = k$, $\omega^i = s_i s_{i+1} ... s_k$, $\omega(i) = s_i$ and L be a mapping from S to $2^{\mathcal{B}}$. We have:

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

Logics: LTL formulas

Let $\mathcal{B}$ be a set of predicates. The following defines an LTL formula:

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

Let $\omega = s_1 s_2 \ldots s_k$, $|\omega| = k$, $\omega^i = s_i s_{i+1} \ldots s_k$, $\omega(i) = s_i$ and L be a mapping from S to $2^{\mathcal{B}}$. We have:

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

Additionally, we use the *eventually* operator $\Diamond$ defined as $\Diamond\phi = \mathbf{F}\mathcal{U}\phi$.

Note that we only consider finite executions.

Logics: Execution Predicates

Definition (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 mapping $p : \sigma \in \Sigma(S) \mapsto p(\sigma) \in \{\mathbf{T}, \mathbf{F}\}$.

Logics: Execution Predicates

Definition (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 mapping $p : \sigma \in \Sigma(S) \mapsto p(\sigma) \in \{\mathbf{T}, \mathbf{F}\}$.

Example

Execution predicate p that decides whether the mean value of the analog signal associated with σ is ≥ 0:

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

More complex functionals such as the Fourier transform can be used

Logics: Execution Predicates

Definition (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 mapping $p : \sigma \in \Sigma(S) \mapsto p(\sigma) \in \{\mathbf{T}, \mathbf{F}\}$.

Example

Execution predicate p that decides whether the mean value of the analog signal associated with σ is ≥ 0:

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

More complex functionals such as the Fourier transform can be used

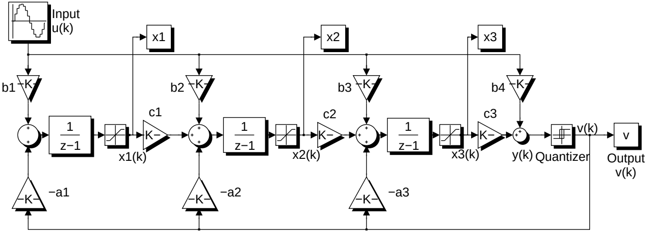
Claim

Let $\mathcal{S}$ be an SSDES and ϕ be a Boolean combination of LTL formulas and execution predicates. One can always associate a probability with the set of executions of $\mathcal{S}$ that satisfy ϕ.

Outline

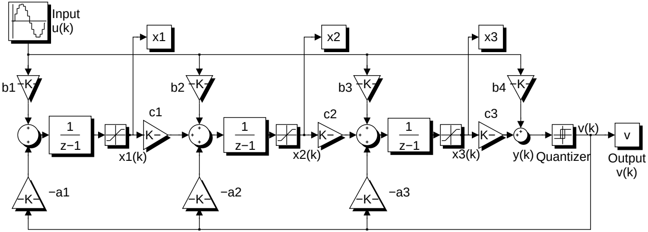
A third order $\Delta - \Sigma$ modulator, Simulink model

[Gupta Krogh Rutenbar 04], [Dang Donze Maler 04]



A third order $\Delta - \Sigma$ modulator, Simulink model

[Gupta Krogh Rutenbar 04], [Dang Donze Maler 04]

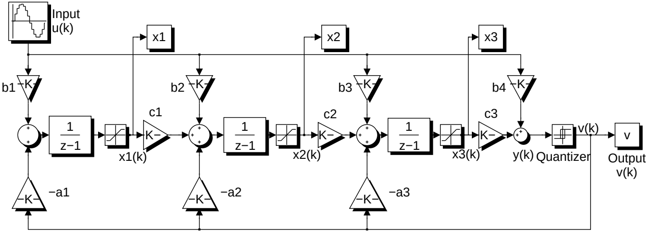


We get a stochastic system by randomly choosing the inputs $u(k)$

- State s_k is the tuple $(u(k), x_1(k), x_2(k), x_3(k), v(k))$
- The next state s_{k+1} is determined by the random choice of $u(k+1)$ and computed by the Simulink engine
- For all k, $u(k)$ is chosen uniformly in $[-u_{\max}, u_{\max}]$

A third order $\Delta - \Sigma$ modulator, Simulink model

[Gupta Krogh Rutenbar 04], [Dang Donze Maler 04]



We get a stochastic system by randomly choosing the inputs $u(k)$

- State s_k is the tuple $(u(k), x_1(k), x_2(k), x_3(k), v(k))$
- The next state s_{k+1} is determined by the random choice of $u(k+1)$ and computed by the Simulink engine
- For all k, $u(k)$ is chosen uniformly in $[-u_{\max}, u_{\max}]$

$\Rightarrow$ Statistical analysis for all input signals of amplitude bounded by $u_{\max}$

Saturation Analysis

Probability of saturation occurrence for different values of $u_{\max}$?

Saturation Analysis

Probability of saturation occurrence for different values of $u_{\max}$?

- Let $Satur$ be a boolean predicate
- For all state $s = (u, x_1, x_2, x_3, v)$, let $L(s) = \{Satur\}$ iff $|x_3| \geq 1$

We can then evaluate the formula $Pr_{\geq\theta}(\diamondsuit Satur)$.

Saturation Analysis

Probability of saturation occurrence for different values of $u_{\max}$?

- Let $Satur$ be a boolean predicate
- For all state $s = (u, x_1, x_2, x_3, v)$, let $L(s) = \{Satur\}$ iff $|x_3| \geq 1$

We can then evaluate the formula $Pr_{\geq\theta}(\Diamond Satur)$.

We implemented

- A routine checking $\sigma \models \Diamond Satur$
- The sequential ratio testing algorithm which decides whether $\mathcal{S} \models Pr_{\geq\theta}(\phi)$ given θ, α, β and δ

Experimental Results

$u_{\max}$	Hypothesis Accepted	Number of executions
0.1	$p \leq 0$	416
0.15	$p \geq 0.1$	4967
0.2	$p \geq 0.6$	17815
0.25	$p \geq 0.98$	416
0.3	$p \geq 1$	688

Table of results for $p = Pr(\sigma \models \diamondsuit Satur)$,
with $\alpha = \beta = 1e^{-3}$ and $\delta = 1e^{-2}$

Experimental Results

$u_{\max}$	Hypothesis Accepted	Number of executions
0.1	$p \leq 0$	416
0.15	$p \geq 0.1$	4967
0.2	$p \geq 0.6$	17815
0.25	$p \geq 0.98$	416
0.3	$p \geq 1$	688

Table of results for $p = Pr(\sigma \models \Diamond Satur)$,
with $\alpha = \beta = 1e^{-3}$ and $\delta = 1e^{-2}$

- Consistent with results formally obtained in [Dang Donze Maler 04] but on a much larger horizon (24000 as compared to 31)

Experimental Results

$u_{\max}$	Hypothesis Accepted	Number of executions
0.1	$p \leq 0$	416
0.15	$p \geq 0.1$	4967
0.2	$p \geq 0.6$	17815
0.25	$p \geq 0.98$	416
0.3	$p \geq 1$	688

Table of results for $p = Pr(\sigma \models \Diamond Satur)$,
with $\alpha = \beta = 1e^{-3}$ and $\delta = 1e^{-2}$

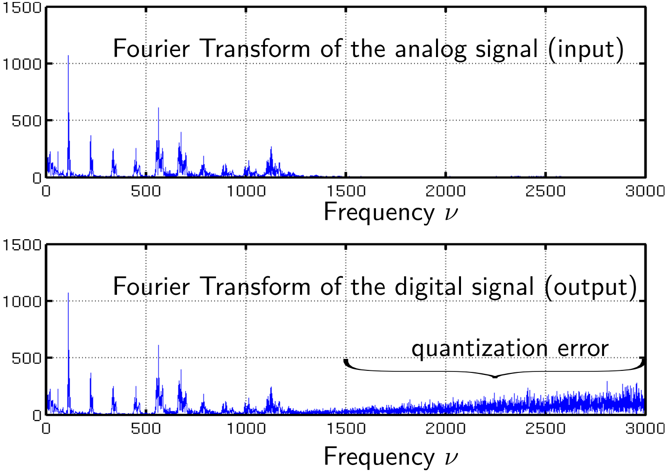
- Consistent with results formally obtained in [Dang Donze Maler 04] but on a much larger horizon (24000 as compared to 31)
- The expected number of simulations grows logarithmically w.r.t. the inverse of α and β and polynomially w.r.t. the inverse of δ

Verification in the Frequency Domain

The signal conversion is correct if the Fourier transforms (FT) of the analog signal and the digital signal are similar in low frequencies.

Verification in the Frequency Domain

The signal conversion is correct if the Fourier transforms (FT) of the analog signal and the digital signal are similar in low frequencies.

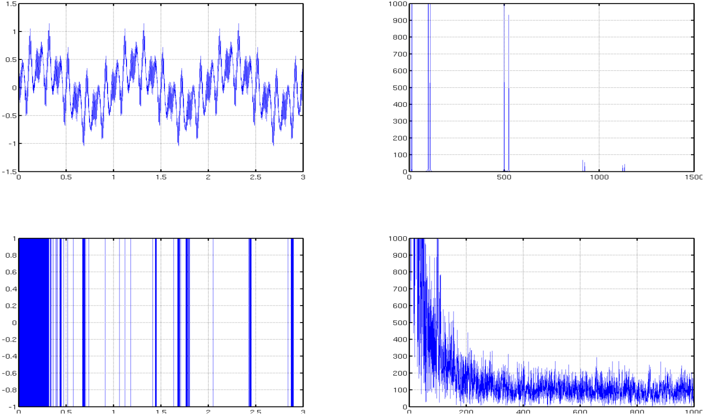


Failed conversion, example

Time domain | Frequency domain

Analog

Digital



Execution Predicate in the Frequency Domain

- Let $F_u(\sigma)$ and $F_v(\sigma)$ be the Fourier Transforms (FTs) of the input signal associated with σ
- Let $d_f^{\nu_0}(\hat{\xi}_1, \hat{\xi}_2)$ be a measure of the distance between two FTs $\hat{\xi}_1$ and $\hat{\xi}_1$ for frequencies smaller than ν_0
- Then we can derive an execution predicate p_f such that

$$p_f(\sigma) = \mathbf{T} \text{ iff } d_f^{\nu_0}(F_u(\sigma), F_v(\sigma)) \leq \epsilon,$$

For $\nu_0 = 100Hz$ and $\epsilon \leq .1$ the predicate discriminates between "correct" and "failed" conversions

Frequency Domain Predicate, Experimental Results

$u_{\max}$	Hypothesis Accepted	Number of Executions
0.8	$p \geq 1$	688
0.9	$p \geq 0.98$	612
1.0	$p \geq 0.98$	1248
1.1	$p \geq 0.875$	6388
1.2	$p \geq 0.55$	15507

Table of results for $p = Pr(p_f)$,
with $\alpha = \beta = 1e^{-3}$ and $\delta = 1e^{-2}$

Experiments Interpretation

The previous results show that

- For $u_{\max} \geq 0.3$ the system satisfies $\Diamond Satur$ with probability 1
- For $u_{\max} \leq 0.8$ the system satisfies p_f with probability 1

Thus we statistically established that for $0.3 \leq u_{\max} \leq 0.8$, the formula $\Diamond Satur \wedge p_f$ is satisfied with probability 1, meaning that saturation can occur without a dramatic decrease in the conversion quality

This extends the results in [Gupta Krogh Rutenbar 04] and [Dang Donze Maler 04] where it was conservatively assumed that the absence of saturation was necessary for a proper behavior

Summary

- A framework for the statistical probabilistic Model Checking of mixed-signal circuits
- The simulation-based approach makes it easier to deal with functionals on executions such as the Fourier transform
- Application to a non-trivial case study for which we improved previous results

Summary

- A framework for the statistical probabilistic Model Checking of mixed-signal circuits
- The simulation-based approach makes it easier to deal with functionals on executions such as the Fourier transform
- Application to a non-trivial case study for which we improved previous results

Future work

- Extension to unbounded execution and dense time using appropriate monitoring techniques
- Logic mixing temporal properties and partial execution predicates
- More precise definitions and specifications for frequency domain properties based on the need of analog designers