SQNR ESTIMATION OF NON-LINEAR FIXED-POINT ALGORITHMS

Gabriel Caffarena[*], *Juan A. López*[+], *Angel Fernández-Herrero*[+] *and Carlos Carreras*[+],

[*] Universidad CEU-San Pablo, Urbanización Montepríncipe, 28668, Boadilla del Monte, Madrid, Spain
email: gabriel.caffarenafernandez@ceu.es
[+]Universidad Politécnica de Madrid, C. Universitaria, 28040, Madrid, Spain.
e-mail: {carreras—juanant—angelfh}@die.upm.es

ABSTRACT

In this paper, a fast and accurate quantization noise estimator aiming at fixed-point implementations of Digital Signal Processing (DSP) algorithms is presented. The estimator enables significant reduction in the computation time required to perform complex wordlength optimizations.

The proposed estimator is based on the use of Affine Arithmetic (AA) and it is aimed at differentiable non-linear algorithms with and without feedbacks. The estimation relies on the parameterization of the statistical properties of the noise at the output of fixed-point algorithms. Once the output noise is parameterized (i.e. related to the fixed-point formats of the algorithm signals), a fast estimation can be applied throughout the wordlength optimization process using as a precision metric the SQNR.

The estimator is tested using a subset of non-linear algorithms such as vector operations, adaptive filters and a channel equalizers. wordlength optimization times are boosted by three orders of magnitude while keeping the average estimation error down to 13%.

1. INTRODUCTION

The original infinite precision of an algorithm based on the use of real arithmetic must be reduced to the practical precision bounds imposed by digital computing systems. wordlength optimization (WLO) aims at the selection of the variables' wordlengths of an algorithm to comply with a certain output noise constraint while optimizing the characteristics of the implementation. Fixed-point (FxP) arithmetic is commonly used in VLSI implementations since it leads to lower cost implementations in terms of area, speed and power consumption [1, 2, 3].

WLO is a slow process due to the fact that the optimization is very complex (NP-hard) and also because of the necessity of a continuous assessment of the algorithm accuracy. This estimation is normally performed by adopting a simulation-based approach [4, 3] which leads to exceedingly long design times. However, in the last few years, there have been attempts to provide fast estimation methods based on analytical techniques to reduce design times. Among the different quality metrics used, SQNR is of most interest for DSP systems. In this work, we focus on fast SQNR estimation of FxP algorithms covering non-linear mathematical operations. Thus, a wide range of DSP applications can be targeted (i.e. adaptive filtering, matrix operations, etc.). Our approach tries to overcome the limitations of previous approaches, which are: the application to non-recursive algorithms only, the use of non-accurate quantization noise models, extremely long noise parameterization times, etc.

This paper contains the following contributions:

- A novel AA-based SQNR estimator for differentiable non-linear algorithms with and without loops. The estimator is fast and accurate and it enables fast WLO.
- Performance results of a set of DSP benchmarks covering algorithms with and without loops, under different application scenarios (e.g. 4G equalization, adaptive filtering, etc.).

The paper is structured as follows: In section 2, related work is discussed. Section 3 deals with the novel SQNR estimation proposal. Performance results are collected in section 4. And finally, section 5 draws the conclusions.

2. RELATED WORK

In this section, we focus on those approaches aiming at the automatic SQNR estimation of non-differentiable algorithms. SQNR estimators applicable only to LTI systems are not included in the discussion ([2, 5]).

2.1 Non-linear approaches

The approaches aiming at non-linear systems are mainly based on perturbation theory, where the effect of the quantization of each algorithm's signal on the quality of the output signal is supposed to be *small*. This allows to apply a first-order Taylor expansion to each non-linear operation in order to characterize the effect of the quantization of the inputs of the operations. This constrains the application to algorithms composed of differentiable operations. The existent methods enable to obtain an expression that relates the wordlengths of signals to the power – also mean and variance – of the quantization noise at the output (see subsection 3.3).

In [6] a hybrid method which combines simulations and analytical techniques to estimate the variance of the noise is proposed. The estimator is suitable for non-recursive and recursive algorithms. The parameterization phase is fast, since it requires N simulations for an algorithm with N variables. The noise model is based on [7] and second order effects are neglected by applying first order Taylor expansions. The paper suggests that the contributions of the signal quantization noises at the output can be added, assuming that the noises are independent. In non-linear systems this is a strong assumption that might lead to variance underestimation. The accuracy of the method is not supported with any empirical data, so it cannot be inferred the quality of the method.

In [8] another method suitable for non-recursive and recursive algorithms is presented. Here, $N^2/2$ simulations as well as a curve fitting technique (with $N^2/2$ variables) are required to parameterize quantization noise, leading to high parameterization times. On the one hand, the noise produced

by each signal is modeled following the traditional quantization noise model from [9], which is less accurate than [7], and, again, second order statistics are neglected. On the other hand, the expression of the estimated noise power accounts for noise interdependencies, which is a better approach than [6]. The method is tested with an LMS adaptive filter and the accuracy is evaluated graphically. There is no information about computation times.

Finally, in [10] the parameterization is performed by means of N simulations and the estimator is suitable only for non-recursive systems. The accuracy of this approach seems to be the highest since it uses the model from [7] and it accounts for noise interdependencies. Although the information provided about accuracy is more complete, it is still not sufficient, since the estimator is only tested in a few SQNR scenarios. This approach was successfully extended to recursive systems in [11], with reasonably short parameterization times due to the use of linear-prediction techniques.

2.2 This work

Our approach tries to overcome most of the drawbacks of the works presented above: it deals with non-recursive and recursive systems, with an accurate noise model [7] and also accounting for noise interdependencies. The parameterization time can be relatively long for algorithms that contain loops. However, the computation times are within standard times (see section 4), and the benefits of fast estimates make up for the sometimes slow parameterization process. Also, the use of AA enables reconstructing the error probability density function at each time step, thus allowing a powerful tool for error analysis of DSP algorithms, and proposed technique can be integrated with other AA-based techniques that enable range estimation, limit cycles analysis [12], etc.

3. SQNR ESTIMATION

3.1 Wordlength optimization

Wordlength optimization is commonly divided into two sequential tasks:

- *Scaling*.
  Its objective is to compute the number of integer bits of each signal. It can be computed by means of a single simulation that collects the signals ranges or by using interval arithmetic (IA) based approaches.
- *Wordlength selection*.
  Here, the number of fractional bits is calculated. It is a time-consuming task since it involves the application of optimization techniques. During the optimization process, thousands of estimates of the quality of the fixed-point version of the algorithm (i.e. SQNR estimation) are required. Traditionally, this estimation is based on simulations, leading to extremely long optimization times.

The fast SQNR estimation technique that we present in this paper is intended to replace those simulations, thus reducing the wordlength selection time dramatically.

3.2 Affine Arithmetic

Affine Arithmetic (AA) [13] is an extension of Interval Arithmetic (IA) aimed at the fast and accurate computation of the ranges of an algorithm signals. Its main feature is that it automatically cancels the linear dependencies of the included uncertainties along the computation path, thus avoiding the oversizing produced by IA approaches.

The mathematical expression of an affine form is

$$\hat{x} = x_0 + \sum_{i=1}^{N} x_i \varepsilon_i \tag{1}$$

where x_0 is the central value of $\hat{x}$, and ε_i and x_i are its i-th noise term identifier and amplitude, respectively. In fact, $x_i\varepsilon_i$ represents the interval $[-x_i, +x_i]$, so an affine form describes a numerical domain in terms of a central value and a sum of intervals with different identifiers. Affine operations are those which operate affine forms and produce an affine form as a result. Given the affine forms $\hat{x}$, $\hat{y}$ and $\hat{c} = c_0$, the affine operations are

$$\hat{x} \pm \hat{c} = x_0 \pm c_0 + \sum_{i=1}^{N} x_i \varepsilon_i \tag{2}$$

$$\hat{x} \pm \hat{y} = x_0 \pm y_0 + \sum_{i=1}^{N} (x_i \pm y_i) \varepsilon_i \tag{3}$$

$$\hat{c} \cdot \hat{x} = c_0 x_0 + \sum_{i=1}^{N} c_0 x_i \varepsilon_i \tag{4}$$

These operations suffice to model any LTI algorithm. Differentiable operations can be approximated using a first-order Taylor expansion:

$$f(\hat{x},\hat{y}) \approx f(x_0,y_0) + \sum_{i=1}^{N} \left(\frac{\delta f(x_0,y_0)}{\delta \hat{x}} \cdot x_i + \frac{\delta f(x_0,y_0)}{\delta \hat{y}} \cdot y_i \right) \varepsilon_i \tag{5}$$

3.3 AA-based SQNR Estimation

Here, we present a method able to estimate the quantization noise power of recursive and non-recursive non-linear algorithms from an AA simulation.

Noise estimation is based on the assumption that the quantization of a signal s_i from n_{pre} bits to n bits can be modeled by the addition of a uniformly distributed white noise with the following statistical parameters [7]:

$$\sigma_i^2 = \frac{2^{2p_i}}{12} \left(2^{-2n_i} - 2^{-2n_i^{pre}} \right) \tag{6}$$

$$\mu_i = -2^{p_i - 1} \left(2^{-n_i} - 2^{-n_i^{pre}} \right). \tag{7}$$

This noise model is an extension of the traditional modeling of quantization error as an additive white noise [9]. In [7] is shown that the traditional continuous model can produce an error of up to 200% in comparison to this modified version.

In [5] was proved that the effect of the deviation from the original behavior of an algorithm with feedback loops can be modeled by adding an affine form $\hat{n}_i[n]$ to each signal i at each simulation time instant n. The affine form $\hat{n}_i$ models a quantization noise with mean μ_i and variance σ_i^2, if each error term ε is assigned a uniform distribution:

$$\hat{n}_i[n] = \mu_i + \sqrt{12\sigma^2}\varepsilon_{i,n} = \varepsilon'_{i,n} \tag{8}$$

Thus, it is possible to know at each moment the origin of a particular error term (i) and the moment when it was generated (n). The AA-based simulation can be made independent on the particular statistical parameters of each quantization thanks to error term ε'. This is desirable in order to obtain a parameterizable noise model. This error term encapsulates the mean value and the variance of the error term ε, and now it can be seen as a random variable with variance σ_i^2 and mean μ_i. This is a reinterpretation of AA, since the error terms are not only intervals, but they also have a probability distribution associated. Once the simulation is finished, it is possible to compute the impact of the quantization noise produced by signal s_i on the output of the algorithms by checking the values of $x_{i,n}$ (see eqn.(1)). This enables the parameterization of the noise. Once the parameterization is performed, the estimation error produced by any combination of (p,n) can be easily assessed replacing all $\varepsilon'_{i,n}$ by the original expression that accounts for the mean and variance ($\mu_i + \sqrt{12\sigma^2}\varepsilon_{i,n}$), thus enabling a fast estimation or the quantization error. We will see all the process in the next paragraphs.

The expression of a given output $\hat{Y}$ of an algorithm with $|S|$ quantized signals is

$$\hat{Y}[n] = Y_0[n] + \sum_{i=0}^{|S|-1}\sum_{j=0}^{n-1} Y_{i,j}[n]\varepsilon'_{i,j}, \tag{9}$$

where $Y_0[n]$ is the value of the output of the algorithm using floating-point arithmetic and the summation is the contribution of the quantization noise sources. Note that $Y_{i,j}[n]$ is a function that depends on the inputs of the algorithm.

The error $\hat{E}rr_Y$ at the output is in eqn.(10), and it is formed by a collection of affine forms at each time step n. The power of the quantization noise of the output can be approximated by the Mean Square Error (MSE), which is estimated as the mean value of the expectancy of the power of the summations of the uniform distributions at each time step m as in eqn.(11). The estimation is performed using an AA simulation during K time steps.

$$\hat{E}rr_Y[n] = Y_0[n] - \hat{Y}[n] = -\sum_{i=0}^{|S|-1}\sum_{j=0}^{n-1} Y_{i,j}[n]\varepsilon'_{i,j}, \tag{10}$$

$$\begin{aligned} P\left(\hat{E}rr_Y[n]\right) &= \frac{1}{K}\sum_{m=0}^{K-1} E\left[\left(\hat{E}rr_Y[m]\right)^2\right] \\ &= \frac{1}{K}\sum_{m=0}^{K-1}\Big(Var(\hat{E}rr_Y[m]) + \\ &\qquad E\left[\hat{E}rr_Y[m]\right]^2\Big) \end{aligned} \tag{11}$$

This equation relies on the fact that error terms $\varepsilon'_{i,n}$ are uncorrelated to each other, which is a sensible assumption in quantized DSP systems [9]. Also, the non-correlation between quantization noises enables to express the variance of a summation of random variables as the summation of the variance of each random variable. The two main terms in eqn.(11) are developed in (12) and (13).

$$\begin{aligned} Var(\hat{E}rr_Y[m]) &= Var(-\sum_{i=0}^{|S|-1}\sum_{j=0}^{m-1} Y_{i,j}[m]\varepsilon'_{i,j}) \\ &= \sum_{i=0}^{|S|-1}\sum_{j=0}^{m-1} Var(-Y_{i,j}[m]\varepsilon'_{i,j}) \\ &= \sum_{i=0}^{|S|-1}\sigma_i^2\sum_{j=0}^{m-1} Y_{i,j}^2[m] \end{aligned} \tag{12}$$

$$\begin{aligned} E[\hat{E}rr_Y[m]] &= E[-\sum_{i=0}^{|S|-1}\sum_{j=0}^{m-1} Y_{i,j}[m]\varepsilon'_{i,j}] \\ &= -\sum_{i=0}^{|S|-1}\mu_i\sum_{j=0}^{m} Y_{i,j}[m] \end{aligned} \tag{13}$$

Combining (11), (12) and (13):

$$\begin{aligned} P\left(\hat{E}rr_Y[n]\right) &= \frac{1}{K}\sum_{m=0}^{K-1}\Bigg(\sum_{i=0}^{|S|-1}\sigma_i^2\sum_{j=0}^{m} Y_{i,j}^2[m] + \\ &\qquad \left(\sum_{i=0}^{|S|-1}\mu_i\sum_{j=0}^{m} Y_{i,j}[m]\right)^2\Bigg) \end{aligned} \tag{14}$$

The output noise power (eqn. 14) can be expressed more compactly by using vector $\vec{v}$ and matrix M as shown in equations (eqns. 15-17). The statistical parameters of the quantization signals are in vectors $\vec{\sigma}^2 = \left\langle \sigma_0^2 \ldots \sigma_{|S|-1}^2 \right\rangle$ and $\vec{\mu} = \left\langle \mu_0 \ldots \mu_{|S|-1} \right\rangle$. Once vector $\vec{v}$ and matrix M are computed, the estimation of the quantization noise does not require a simulation but the computation of (15), which is a much faster process.

$$P_o = \frac{1}{K}\left(\vec{\sigma}^2 \cdot \vec{v}^T + \vec{\mu} \cdot M\vec{\mu}^T\right) \tag{15}$$

$$\begin{aligned} \vec{v} \equiv \Bigg\langle &\sum_{n=0}^{K-1}\sum_{j=0}^{n-1} Y_{0,j}^2[n], \ldots, \\ &\sum_{n=0}^{K-1}\sum_{j=0}^{n-1} Y_{|S|-1,j}^2[n] \Bigg\rangle \end{aligned} \tag{16}$$

$$M \equiv \begin{bmatrix} m_{0,0} & \cdots & m_{|S|-1,0} \\ & \ddots & \\ m_{0,|S|-1} & \cdots & m_{|S|-1,|S|-1} \end{bmatrix} \tag{17}$$

$$m_{i_1,i_2} = \sum_{n=0}^{K-1}\left(\sum_{j_1=0}^{n-1} Y_{i_1,j_1}[n]\sum_{j_2=0}^{n-1} Y_{i_2,j_2}[n]\right) \tag{18}$$

The parameterization process is composed of the following steps:

1. Perform a $K-$step AA simulation adding a affine forms $\hat{n}_i$ to each signal i

2. Compute eqns. (16-18) using previously collected $Y_{i,j}[n]$.

During WLO the wordlengths of signals are used to compute vectors $\vec{\sigma}^2$ and $\vec{\mu}$, and then the error can be estimated very quickly by using eqn.(15). Please note that expression (14) can be applied to DSP algorithms including differentiable operations (e.g. multiplications, divisions, etc.) by means of eqn.(5) due to the 1st order approximation.

4. RESULTS

This section presents the performance results of our fast estimator. The benchmarks used to test the performance of the SQNR estimator are:

- 8x8 vector scalar multiplication (VEC_{8x8})
- MIMO channel equalizer (EQ) [14]
- a mean power estimator based on an 1st IIR filter (POW)
- 2nd-order LMS filter (LMS_2) [6]
- 5th-order LMS filter (LMS_5) [6]

The set of benchmarks covers cyclic and acyclic algorithms and that the set of operations includes additions, multiplications, and also divisions, usually neglected in similar research studies. In addition to that, it is interesting to highlight that the algorithms are not limited to linear filtering; they address 4G MIMO channel equalizing, vector multiplications and adaptive filtering.

All benchmarks are fed with 16-bit inputs and 12-bit constants and the noise constraint is an SQNR ranging from 40 to 120 dB. The inputs used to perform the noise parameterization as well as the fixed-point simulation are summarized in the last column of the table.

The procedure to carry out the tests is as follows:

1. Compute scaling by means of a floating point simulation.
2. Extract noise parameters (eqns. 16-18) performing an AA-based simulation.
3. Perform a WL selection based on the fast estimator (eqn. 15), using a gradient-descent approach.
4. Perform a single FxP bit-true simulation and use it as reference to assess the performance of the estimator.

The accuracy obtained by means of a gradient-descent optimization [2] under different SQNR constraints – 80 in total, from 40 dB to 120 dB – for the different benchmarks is presented in Table 4. The first column indicates the benchmark used. The remaining columns show the accuracy of the estimations measured in terms of the maximum absolute value of the relative error in dB, and the average of the absolute value of the percentage error, for four SQNR ranges: [120,100) dB, [100,80) dB, [80, 60) dB and [60,40] dB (see the expressions of the metrics at the bottom of the table).

The results yield that the estimator is very accurate. The mean percentage error is smaller than 4.3 %, and the maximum relative error is smaller than 1.12 dB. Note that the accuracy decreases as long as the error constraints get looser. This is due to the amplification of the Taylor error terms (specially in the presence of loops) and also to the fact that the uniformly distributed model for the quantization noise does not remain valid for small SQNRs. Anyway, the quality of the estimates is still very high. The average percentage error confirms the excellent accuracy obtained by our estimator.

Table 4 holds the performance results in terms of computation times. The first column shows the names of the benchmarks. The second and third columns show the length of the input vectors required for a fixed-point simulation and for the parameterization process. The parameterization time is in the fourth column. The average number of iterations required during the optimization process is in the sixth column. The next two columns present the computation time required to perform the gradient-descent optimization using our estimation-based proposal and using a classical simulation-based approach. The computation time for the simulation-based approach is an estimation based on multiplying the average number of optimization iterations by the computation time of a single fixed-point simulation. Finally, the speed-up obtained by our estimation-based approach is in the last column.

The parameterization time goes from 330μsecs. to 28 mins. (1646 secs.) and it depends on the size of the input data, the complexity of the algorithm (i.e. number and types of operations) and the presence of loops. The LMS benchmarks clearly show how the parameterization time is increased as long as the number of delays, and therefore loops, increases. These times might seem quite long, but it must be bared in mind that the parameterization process is performed only once, and after that the algorithm can be assigned a fixed-point format as many times as desired using the fast estimator.

The mean number of estimates in the fifth column is shown to give an idea of the complexity of the optimization process. A simulation-based optimization approach would require that very same number of simulations,thus taking a very long time. For instance, the optimization of LMS_5 would approximately require 2500 FxP simulations of 5000 input data. Considering the number of estimations required, the optimization times are extremely fast, ranging from 0.02 secs to 7.26 secs. The speedups obtained in comparison to a simulation-based approach are staggering: boosts from $\times 331$ to $\times 1377$ are obtained. The average boost is $\times 776$ which proves the advantage of our approach in terms of computation time.

In summary, results show that our approach enables fast and accurate WLO of non-linear DSP algorithms.

5. CONCLUSIONS

A novel noise estimation method based on the use of Affine Arithmetic has been presented. This method allows to obtain fast an accurate estimates of the quantization noise at the output of the FxP description of a DSP algorithm. The estimator can be used to perform complex WLO in standard times leading to significant hardware cost reductions. The method can be applied to differentiable non-linear DSP algorithms with and without feedbacks. In brief, the main contributions of the paper are:

- The proposal of a novel AA-based quantization noise estimation for non-linear algorithms with and without feedbacks
- The average estimation error for non-linear systems is smaller than 12% in general, and smaller than 5% for most cases.
- The computation time of WLO is boosted up to $\times 1377$ (average of $\times 766$)

The reduction of the computation time of the noise parameterization process in the presence of loops, is to be approached in the near future, as well as the improvement of the quantization model for non-linear operations.

Table 1: Performance of the estimation method: Precision.

Benchmark	Estimation error [120,100)[1] dB (dB)[2]	(%)[3]	[100,80) dB (dB)	(%)	[80,60) dB (dB)	(%)	[60,40] dB (dB)	(%)
$VEC_{8\times8}$	0.05	0.57	0.04	0.40	0.04	0.57	0.13	1.19
EQ^*	0.27	0.98	0.24	0.71	0.29	0.17	0.18	1.52
POW^*	0.39	5.00	0.17	1.55	0.76	5.96	1.12	12.12
LMS_2^*	0.09	0.46	0.08	0.24	0.15	0.78	0.92	3.73
LMS_5^*	0.09	0.46	0.08	0.07	0.13	1.08	1.09	5.51
All	0.39	1.27	0.24	0.05	0.76	1.48	1.12	4.21

* Recursive [1] Error constraint [2] $|10log(\frac{P_{ref}}{P_{est}})|$ (max) [3] $|100(\frac{P_{ref}-P_{est}}{P_{ref}})|$ (average)

Table 2: Performance of the estimation method: Computation time.

Bench.	FxP Samples	Param. Samples	Param. time (secs)$^+$	No. of estimates (mean)	Estimation-based WLO (secs)$^+$	Simulation-based WLO (secs)$^+$	Speed-up
$VEC_{8\times8}$	20000	20000	330	1739	1.72	2331.79	$\times$1377
EQ^*	16000	16000	61.64	231	0.12	105.78	$\times$904
POW^*	20000	20000	546.14	97	0.02	21.93	$\times$1048
LMS_2^*	5000	5000	592.11	1032	0.94	310.93	$\times$331
LMS_5^*	5000	5000	1646.38	2547	7.26	1611.46	$\times$221
All	-	-	-	-	-	-	$\times$776

* Recursive $^+$ Using 1.66 GHz Intel Core Duo processor and 1 GB of RAM

6. ACKNOWLEDGMENT

This work was supported by Research Projects TEC2006-13067-C03-03 (Spanish Ministry of Education and Science) and TEC2009-14219-C03-02 (Spanish Ministry of Science and Innovation) and by the Universidad CEU-San Pablo.

REFERENCES

[1] G. Caffarena, G. Constantinides, P. Cheung, C. Carreras, and O. Nieto-Taladriz, "Optimal Combined Word-Length Allocation and Architectural Synthesis of Digital Signal Processing Circuits," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 5, pp. 339–343, 2006.

[2] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Wordlength Optimization for Linear Digital Signal Processing," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 10, pp. 1432–1442, 2003.

[3] W. Sung and K.-I. Kum, "Simulation-Based Word-Length Optimization Method for Fixed-Point Digital Signal Processing Systems," *IEEE Trans. Signal Processing*, vol. 43, no. 12, pp. 3087–3090, 1995.

[4] G. Caffarena, A. Fernandez, C. Carreras, and O. Nieto-Taladriz, "Fixed-Point Refinement of OFDM-Based Adaptive Equalizers: A Heuristic Approach," in *Proc. EUSIPCO*, pp. 1353–1356, 2004.

[5] J. Lopez, G. Caffarena, C. Carreras, and O. Nieto-Taladriz, "Fast and accurate computation of the round-off noise of linear time-invariant systems," *IET Circuits, Devices & Systems*, vol. 2, no. 4, pp. 393–408, 2008.

[6] G. Constantinides, "Perturbation Analysis for Word-Length Optimization," in *Proc. FCCM*, pp. 81–90, 2003.

[7] G. Constantinides, P. Cheung, and W. Luk, "Truncation Noise in Fixed-Point SFGs," *IEE Electronics Letters*, vol. 35, no. 23, pp. 2012–2014, 1999.

[8] C. Shi and R. Brodersen, "A Perturbation Theory on Statistical Quantization Effects in Fixed-Point DSP with Non-Stationary Inputs," in *Proc. ISCAS*, vol. 3, pp. 373–376 Vol.3, 2004.

[9] L. Jackson, "Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form," *IEEE Trans. Audio Electroacoust.*, vol. 18, pp. 107–122, Jun 1970.

[10] D. Menard, R. Rocher, P. Scalart, and O. Sentieys, "SQNR Determination in Non-Linear and Non-Recursive Fixed-Point Systems," in *Proc. EUSIPCO*, pp. 1349–1352, 2004.

[11] R. Rocher, Menard, O. Sentieys, and P. Scalart, "Analytical Accuracy Evaluation of Fixed-Point Systems," pp. 999–1003, 2007.

[12] J. Lopez, G. Caffarena, C. Carreras, and O. Nieto-Taladriz, "Analysis of Limit Cycles by Means of Affine Arithmetic Computer-aided Tests," in *Proc. EUSIPCO*, pp. 991–994, 2004.

[13] J. Stolfi and L. H. Figueiredo, "Self-Validated Numerical Methods and Applications," in *Brazilian Mathematics Colloquium: IMPA*, 1997.

[14] A. Fernández, A. Jimenez, G. Caffarena, and J. Casajús, "Design and Implementation of a Hardware Module for Equalisation in a 4G MIMO Receiver," in *Proc. FPL*, pp. 765–768, 2006.