Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 171027, 12 pages
doi:10.1155/2010/171027

Research Article

SQNR Estimation of Fixed-Point DSP Algorithms

Gabriel Caffarena (EURASIP Member),[1] Carlos Carreras,[2] Juan A. López,[2] and Ángel Fernández[2]

[1] *Departamento de Ingeniería de Sistemas de Información y de Telecomunicación, Universidad CEU-San Pablo, Urbanización Montepríncipe, Boadilla del Monte, 28668 Madrid, Spain*
[2] *Departmento de Ingeniería Electrónica, Universidad Politécnica de Madrid, C. Universitaria, 28040 Madrid, Spain*

Correspondence should be addressed to Gabriel Caffarena, gabriel.caffarenafernandez@ceu.es

Received 1 August 2009; Revised 10 February 2010; Accepted 13 April 2010

Academic Editor: Jar Ferr Yang

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 word-length optimizations. The proposed estimator is based on the use of Affine Arithmetic (AA) and it is presented in two versions: (i) a general version suitable for differentiable nonlinear algorithms, and Linear Time-Invariant (LTI) algorithms with and without feedbacks; and (ii) an LTI optimized version. The process 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 word-length optimization process using as a precision metric the Signal-to-Quantization Noise Ratio (SQNR). The estimator is tested using different LTI filters and transforms, as well as a subset of non-linear operations, such as vector operations, adaptive filters, and a channel equalizer. Fixed-point optimization times are boosted by three orders of magnitude while keeping the average estimation error down to 4%.

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. Word-length optimization (WLO) aims at the selection of the variables' word-lengths of an algorithm to comply with a certain output noise constraint while optimizing the characteristics of the implementation (e.g., area, speed or power consumption). Normally, the precision loss committed is computed by using a double precision floating-point arithmetic description of the algorithm as a reference and, although there are some works on quantization for custom floating-point arithmetic [1–3], the common approach is to implement the system using fixed-point (FxP) arithmetic, since this leads to lower cost implementations in terms of area, speed, and power consumption [4–7].

WLO is a slow process due to the fact that the optimization is very complex (NP-hard [8]) and also because of the necessity of a continuous assessment of the algorithm accuracy which may involve a high computational load. This estimation is normally performed adopting a simulation-based approach [7, 9, 10] 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. These approaches can be applied to Linear Time-Invariant (LTI) systems [6, 11] and to differentiable nonlinear systems [12–15]. As for the noise metric used, they are based on the peak value [15] and on the computation of SQNR [6, 11–14]. Since SQNR is a very popular error metric within DSP systems, our work aims at fast SQNR estimation techniques for LTI and differentiable nonlinear systems.

This paper contains the following contributions:

(i) a novel Affine-Arithmetic (AA) SQNR estimator optimized for LTI algorithms,

(ii) a novel AA-based SQNR estimator for LTI and differentiable algorithms. Previous approaches were not able to deal with feedback systems, or produced overestimations.

Our approach enables addressing complex WLO techniques, since the computation times are drastically reduced while providing high levels of accuracy.

The paper is structured as follows. In Section 2, related work is discussed. Section 3 deals with fixed-point optimization. Section 4 presents the grounds of the novel SQNR estimation proposal. In Section 5, the benchmarks used for validation are described. Performance results are collected in Section 6. And finally, Section 7 draws the conclusions.

2. Related Work

In this section, we focus on those approaches aiming at estimating the quantization noise to avoid the execution of time-consuming simulations [7, 9, 18] and, therefore, that support fast WLO. We disregard those that are not fully automated [19–22], but consider those that, even though are not implemented within an automatic WLO engine, could be easily integrated within one. Also, we do not consider in this analysis approaches that focus on error-free implementations [23–25].

The Signal-to-Quantization Noise Ratio (SQNR) is a popular quality metric in DSP systems. However, only recently it has been considered in the development of fast quantization noise estimators. Approaches such as [19–25] and also the fully automated [15, 26–28] aim basically at peak-value estimates. Most of these works are based on the use of (i) interval arithmetic (IA) [29], which produces significant overestimations in general, and intolerable overestimations in the presence of loops; (ii) multi-interval arithmetic (MIA) [30], which improves the results of IA but it still performs poorly in the presence of loops; (iii) affine arithmetic [31], which solves the cancellation problem of IA, and can alleviate overestimation by applying confidence intervals; and (iv) the computation of first-order derivatives [15, 28], mostly combined with a worst-case analysis, that leads again to overestimation. Due to its interest for DSP applications, only approaches that consider SQNR as a quality metric are fully analyzed in this section.

Table 1 contains information about the main approaches regarding quantization noise fast estimation under the mentioned premises. The first column holds the reference to the approach. The second column indicates if LTI or nonlinear (NL) algorithms are supported. Column 3 shows if the algorithms are cyclic (i.e., containing loops) or not. The computational complexity of the noise parameterization stage, if applicable, is shown in the fourth column. Also, the computational complexity of the noise estimation itself is presented in column 5. The last two columns contain information about the accuracy of the estimates and comments highlighting interesting features or drawbacks of the approaches.

The approaches in the table have been grouped according to the type of algorithm being addressed. The first three rows correspond to approaches aimed at LTI algorithms, the next three rows to those addressing nonlinear algorithms (also valid for LTI systems), and the last two rows describe the features of the two approaches proposed in this paper.

2.1. Linear Time-Invariant Algorithms. Let us start with LTI-oriented methods. Given an algorithm with $|S|$ signals where each signal is quantized to n_i bits, it is possible to relate the number of bits to the power of the noise at the output of the algorithm in steady state by means of the following expression:

$$P_o = \sum_{i=0}^{|S|-1} \sigma_i^2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| G_i\left(e^{j\Omega}\right) \right|^2 d\Omega + \left(\sum_{i=0}^{|S|-1} \mu_i \cdot G_i(1) \right)^2, \tag{1}$$

where G_i is the transfer function from signal s_i to output o, and σ_i^2 and μ_i are the variance and mean of the quantization noise associated to signal s_i—which is related to n_i. This expression can be rewritten more compactly using vectors $\boldsymbol{\sigma}^2, \mathbf{v}, \boldsymbol{\mu},$ and $\mathbf{m}$ as,

$$P_o = \boldsymbol{\sigma}^2 \cdot \mathbf{v}^T + \left(\boldsymbol{\mu} \cdot \mathbf{m}^T\right)^2, \tag{2}$$

$$\boldsymbol{\sigma}^2 \equiv \left\langle \sigma_0^2, \ldots, \sigma_{|S|-1}^2 \right\rangle, \tag{3}$$

$$\boldsymbol{\mu} \equiv \langle \mu_0, \ldots, \mu_{|S|-1} \rangle, \tag{4}$$

$$\begin{aligned} \mathbf{v} \equiv \Bigg\langle & \frac{1}{2\pi} \int_{-\pi}^{pi} \left| G_0\left(e^{j\Omega}\right) \right|^2 d\Omega, \ldots, \\ & \frac{1}{2\pi} \int_{-\pi}^{pi} \left| G_{|S|-1}\left(e^{j\Omega}\right) \right|^2 d\Omega \Bigg\rangle, \end{aligned} \tag{5}$$

$$\mathbf{m} \equiv \langle G_0(1), \ldots, G_{|S|-1}(1) \rangle. \tag{6}$$

Note that $\mathbf{v}$ and $\mathbf{m}$ can be computed by means of a graph analysis, and once they are determined, the output noise power can be estimated from $\boldsymbol{\sigma}^2$ and $\boldsymbol{\mu}$.

In [6] a two-step method is applied where, first, vectors $\mathbf{v}$ and $\mathbf{m}$ are computed, and then, expression (2) is used to estimate the output noise variance during WLO. The Parseval Theorem [32] is applied in order to compute expression (5), since it is possible to obtain an equivalent expression that makes use of the impulse response from signal s_i to the output of the systems ($g_i[n]$), instead of using G_i. This highly simplifies the computational cost. If the length of the input vectors is long enough, expression (1) can be estimated with high precision leading to highly accurate quantization noise estimations.

An AA-based approach is presented in [11]. The approach is based again on the computation of $g_i[n]$ for each signal. Due to the characteristics of AA, it is possible to compute all $g_i[n]$ simultaneously. The process has not been divided into parameterization (extraction of vectors $\mathbf{v}$ and $\mathbf{m}$) and noise estimation. Instead, everything is computed at once. It can be seen in Table 1 that the computational cost is similar to the total cost of [6] (e.g., parameterization plus estimation times). Also, the quality of the estimates is high, since they are based on (1). This approach is further developed in Section 4.3 in order convert it into a two-step

Table 1: Fast quantization noise estimation approaches.

Approach	Type	Cyclic complexity	Parameterization complexity	Estimation complexity	Accuracy	Comments
Constantinides et al. [6]	LTI	YES	$\lvert S\rvert \times$ cgn	$2 \times \lvert S\rvert$-dot product	High	Steady state
López et al. [11]	LTI	YES		$\lvert S\rvert \times$ cgn + $2 \times \lvert S\rvert$-dot product	High	Affine arithmetic steady state
Menard [16]	LTI	YES	$\approx \lvert S\rvert \times$ cgn	$2 \times \lvert S\rvert$-dot product	High	Graph analysis steady state
Constantinides [12]	NL	YES	$\lvert S\rvert \times$ MC	$2 \times \lvert S\rvert$-dot product	Variance overestimated	Differentiable operations 1st order approx.
Menard [13] and Rocher et al. [17]	NL	NO	$\lvert S\rvert \times$ MC	$\lvert S\rvert$-dot product + matrix-vector mult.	High	Differentiable operations 1st order approx.
Shi and Brodersen [14]	NL	YES	$\lvert S\rvert^2/2 \times$ MC+ $\lvert S\rvert^2/2$-coeff. curve-fitting	$\lvert S\rvert$-dot product + matrix-vector mult.	High	Differentiable operations 1st order approx.
This work (Section 4.3)	LTI	YES	$\lvert S\rvert \times$ cgn	$2 \times \lvert S\rvert$-dot product	High	Affine arithmetic Steady state
This work (Section 4.2)	NL	YES	Acycilic: $\approx \lvert S\rvert \times$ MC cycilic: It depends on *amount* of loops and stimuli size	dot product + matrix-vector mult.	High	Affine arithmetic Differentiable op. 1st-order approx.

$|S| \equiv$ number of signals in algorithm.
cgn $\equiv$ computation of $g[n]$.
MC $\equiv$ Monte Carlo simulation.

method, thus, allowing faster noise estimation (see Table 1, *this work—LTI*).

The approach in [16] also relies on (1) to present a two-step estimation method. The parameterization is based on the application of graph transforms that allow to obtain the vectors **v** and **m** (5) and (6). As it can be seen in Table 1, the performance in terms of computation time and accuracy is equivalent to the other two approaches.

2.2. NonLinear Systems. The approaches aimed at nonlinear 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 first-order Taylor expansion to each nonlinear 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 us to obtain an expression similar to (2) that relates the word-lengths of signals to the power—also mean and variance—of the quantization noise at the output. This will be further explained in Section 4.2 (19).

In [12] a hybrid method which combines simulations and analytical techniques to estimate the variance of the noise is proposed. The estimator is suitable for nonrecursive and recursive algorithms. The parameterization phase is relatively fast, since it requires $|S|$ simulations for an algorithm with $|S|$ variables. The noise model is based on [33] and second order effects are neglected by applying first order Taylor expansions. However, the paper seems to suggest that the contributions of the signal quantization noises at the output can be added, assuming that the noises are independent. In nonlinear systems, this is a strong assumption that leads to variance underestimation. The accuracy of the method is not supported with any empirical data.

In [14] another method suitable for nonrecursive 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. On the one hand, the noise produced by each signal is modeled following the traditional quantization noise model from [34, 35], which is less accurate than [33], 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 [12]. The method is tested with an LMS adaptive filter and the accuracy is evaluated graphically. There is no information about computation times.

Finally, in [13] the parameterization is performed by means of $|N|$ simulations and the estimator is suitable only for nonrecursive systems. The accuracy of this approach seems to be the highest since it uses the model from [33] and it accounts for noise interdependencies. Although, the information provided about accuracy is more complete, it is still not sufficient, since the estimator is tested in only a few SQNR scenarios.

2.3. This Work. As aforementioned, we present two approaches: one exclusive for LTI algorithms in steady state, and the other for differentiable algorithms which are a subset of nonlinear algorithms. The LTI-oriented approach is based on [11] and it basically enables the division of the estimation process into two steps. One step is devoted to parameterization, while the other is dedicated to perform fast estimations. This method is equivalent to the other methods present in the literature. The advantage that it offers is that now it is possible to analyze the most important finite word-length effects (SQNR analysis, peak value analysis, dynamic range, limit cycles) using the very same AA simulation engine.

Regarding nonlinear systems, our approach tries to overcome most of the drawbacks of the works presented above. It deals with nonrecursive and recursive systems, using the accurate noise model from [33] and also accounting for noise interdependencies. The parameterization time can be quite long for algorithms that contain loops. However, as we will see in Section 6, the computation times are within standard times, and the benefits of fast estimations make up for the sometimes slow parameterization process.

3. Word-Length Optimization

The starting point of WLO is a signal flow graph $G(V,S)$ that contains information about the signal FxP formats and the data dependencies. The FxP formats of signals enable the computation of the statistical parameters of the quantization noises introduced by them, and the data dependencies are essential to obtain a noise model that relates the signals' noise parameters with the overall noise at the output of the algorithm. Set V holds the operations of the algorithm: additions/subtractions, constant multiplications, multiplications, divisions, and unit delays. Set S contains the signals that interconnect these operations. The FxP format of a number is defined by means of pair (p,n), where p represents the number of bits from the most significant bit (MSB) to the binary point, and n is the number of total bits (see Figure 1). The FxP format of a signal requires two FxP formats: the format before quantization—$(p_{\text{pre}}, n_{\text{pre}})$—and the format after quantization—(p,n) (see [6]). The quantization of the signal is performed only if these two formats are not equal. Initially, the FxP format of signals is unknown and it is the task of WLO to find a suitable set that minimizes the total cost. The FxP format, not only determines the quantization error generated by a quantized signal, but also the number of bits of each signal, and, therefore, the size of the required hardware resources. The size of a resource ultimately determines its area, delay and, power. During WLO, the optimization is guided by means of the cost and the output error obtained from the different FxP formats tried through successive iterations.

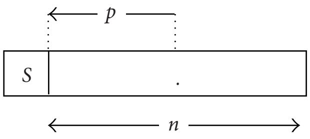

Figure 1: Fixed-point format.

Figure 2 depicts the WLO approach adopted in this work. WLO is composed of the stages of *scaling*, which determines the set of p, and *word-length selection*, which determines the set of n. This subdivision allows to simplify WLO, while still providing significant cost reductions.

A wrap-around scaling strategy is adopted since it requires less hardware than other approaches (i.e., saturation techniques). After scaling, the values of p are the minimum possible values that avoid the overflow of signals or, at least, those that reduce the likelihood of overflow to a negligible value. A simulation-based approach is used to carry out scaling [7].

Once scaling is performed, the values of p can be fixed during word-length selection. The right side of Figure 2 shows basic blocks for word-length selection. The main idea is to iterate trying different word-length (i.e., n) combinations until the cost is minimized. Each time the word-length of a signal or a group of signals is changed, the word-lengths must be propagated throughout the graph, task referred to as graph *conditioning* [6], in order to update the rest of word-lengths. The *optimizer control* block selects the size of the word-lengths using the values of the previous error and cost estimations and decides when the optimization procedure has finished. The first block in the diagram is the extraction of the quantization noise model (parameterization). The role of this block is to generate a model of the quantization noise at the output due to the FxP format of each signal. This enables to perform a quick error estimation within the optimization loop. The implications of using a fast error estimator are twofold. On the one hand, it is possible to reduce WLO time. On the other hand, more complex optimization techniques can be applied in standard computation times.

4. Quantization Noise Estimation

4.1. Affine Arithmetic. Affine Arithmetic (AA) [31] is an extension of Interval Arithmetic (IA) [29] aimed at the fast and accurate computation of the ranges of signals in a particular mathematical description of an algorithm. 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 [36]. It has been applied to both, scaling computation [15, 36, 37], and word-length allocation [1, 15, 36]. Also, a modification called Quantized Affine Arithmetic (QAA) has been applied to the computation of limit cycles [38] and dynamic range analysis of quantized LTI algorithms [37].

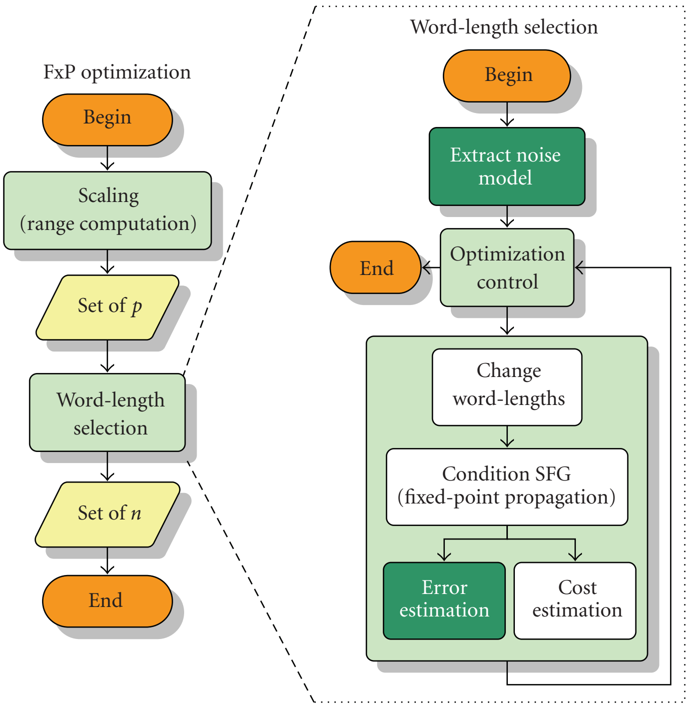

Figure 2: Fixed-point optimization diagram.

The mathematical expression of an affine form is

$$\hat{x} = x_0 + \sum_{i=1}^{N} x_i \epsilon_i, \tag{7}$$

where x_0 is the central value of $\hat{x}$, and ϵ_i and x_i are its ith noise term identifier and amplitude, respectively. In fact, $x_i\epsilon_i$ 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

$$\begin{gathered} \hat{x} \pm \hat{c} = x_0 \pm c_0 + \sum_{i=1}^{N} x_i \epsilon_i, \\ \hat{x} \pm \hat{y} = x_0 \pm y_0 + \sum_{i=1}^{N} (x_i \pm y_i) \epsilon_i, \\ \hat{c} \cdot \hat{x} = c_0 x_0 + \sum_{i=1}^{N} c_0 x_i \epsilon_i. \end{gathered} \tag{8}$$

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

$$\begin{aligned} & f(\hat{x}, \hat{y}) \\ & \quad \approx f(x_0, y_0) + \sum_{i=1}^{N} \left(\frac{\delta f(x_0, y_0)}{\delta \hat{x}} \cdot x_i + \frac{\delta f(x_0, y_0)}{\delta \hat{y}} \cdot y_i \right) \epsilon_i. \end{aligned} \tag{9}$$

4.2. Proposed Estimator: General Expression. Here, we present a method able to estimate the quantization noise power from a single AA simulation. The noise estimation is not based on (1), since this equation only applies to LTI algorithms in steady state and our proposal is more general, since it covers both LTI algorithms and nonlinear algorithms. Also, the parameterization method does not lead to (2)–(6), since these are aimed at LTI algorithms in steady state.

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 [33]:

$$\begin{aligned} \sigma_i^2 &= \frac{2^{2p_i}}{12} \left(2^{-2n_i} - 2^{-2n_i^{\text{pre}}} \right), \\ \mu_i &= -2^{p_i - 1} \left(2^{-n_i} - 2^{-n_i^{\text{pre}}} \right). \end{aligned} \tag{10}$$

This noise model, which is referred to as the *discrete noise model*, is an extension of the traditional modeling of quantization error as an additive white noise [34, 35] (*continuous noise model*). In [33], it is shown that the continuous model can produce an error of up to 200% in comparison to the discrete model.

In [11] it was proved that the effect of the deviation from the original behavior of an algorithm with feedback loops can be modelled 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, and it can be expressed as

$$\hat{n}_i[n] = \mu_i + \sqrt{12\sigma^2}\epsilon_{i,n} = \epsilon'_{i,n}. \tag{11}$$

Table 2: Properties of benchmarks.

Benchmark	LTI	Cyclic	Inputs	Outputs	Z^{-1}	$+/-$	$*$	$*K$	$\div$	$\vert S\vert$	Input signals
RGB	YES	NO	3	3	0	4	0	6	0	16	Uniform noise
IDCT$_8$	YES	NO	8	8	0	37	0	11	0	48	Uniform noise
IIR$_2$	YES	YES	1	1	2	2	0	2	0	8	Uniform noise
LAT$_3$	YES	YES	1	1	3	9	0	10	0	24	Uniform noise
DELTA$_6$	YES	YES	1	1	6	18	0	29	0	62	Uniform noise
VEC$_{3\times3}$	NO	NO	3	3	0	3	3	0	0	12	Uniform noise
VEC$_{8\times8}$	NO	NO	8	8	0	8	8	0	0	32	Uniform noise
EQ	NO	YES*	2	2	64	2	3	2	4	81	MIMO channel Tx [41]
POW	NO	YES	1	1	1	1	1	2	0	7	Synthetic tone
LMS$_1$	NO	YES	2	1	3	5	6	3	0	23	Synthetic tone
LMS$_2$	NO	YES	2	1	5	7	8	4	0	30	Synthetic tone
LMS$_5$	NO	YES	2	1	11	13	14	7	0	51	Synthetic tone
VOL$_3$	NO	YES	2	1	2	4	6	4	0	19	Gaussian noise

* MAC operations applied to 32-data chunks.

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 (7)). 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 $\epsilon'_{i,n}$ by the original expression that accounts for the mean and variance ($\mu_i + \sqrt{12\sigma^2}\epsilon_{i,n}$), thus enabling a fast estimation of the quantization error. We will see all the process in the next paragraphs.

The expression of a given output $\widehat{Y}$ of the algorithm with $|S|$ noise sources is

$$\widehat{Y}[n] = Y_0[n] + \sum_{i=0}^{|S|-1}\sum_{j=0}^{n} Y_{i,j}[n]\epsilon'_{i,j}, \tag{12}$$

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 $\widehat{\mathrm{Err}}_Y$ at the output is

$$\widehat{\mathrm{Err}}_Y[n] = Y_0[n] - \widehat{Y}[n] = -\sum_{i=0}^{|S|-1}\sum_{j=0}^{n} Y_{i,j}[n]\epsilon'_{i,j}. \tag{13}$$

The value of the error 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 (14). The estimation is performed using an AA simulation during K time steps,

$$\begin{aligned} P\left(\widehat{\mathrm{Err}}_Y[n]\right) &= \frac{1}{K}\sum_{m=0}^{K-1} E\left[\left(\widehat{\mathrm{Err}}_Y[m]\right)^2\right] \\ &= \frac{1}{K}\sum_{m=0}^{K-1}\left(\mathrm{Var}\left(\widehat{\mathrm{Err}}_Y[m]\right) + E\left[\widehat{\mathrm{Err}}_Y[m]\right]^2\right). \end{aligned} \tag{14}$$

This equation relies on the fact that error terms $\epsilon'_{i,n}$ are uncorrelated to each other, which is a sensible assumption in quantized DSP systems [34, 35]. Also, the uncorrelation 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 (14) are developed as follows

$$\begin{aligned} E\left[\widehat{\mathrm{Err}}_Y[m]\right] &= \mathrm{Var}\left(-\sum_{i=0}^{|S|-1}\sum_{j=0}^{m} Y_{i,j}[m]\epsilon'_{i,j}\right) \\ &= \sum_{i=0}^{|S|-1}\sum_{j=0}^{m}\mathrm{Var}\left(-Y_{i,j}[m]\epsilon'_{i,j}\right) \\ &= \sum_{i=0}^{|S|-1}\sigma_i^2\sum_{j=0}^{m} Y_{i,j}^2[m], \end{aligned} \tag{15}$$

$$\begin{aligned} E\left[\widehat{\mathrm{Err}}_Y[m]\right] &= E\left[-\sum_{i=0}^{|S|-1}\sum_{j=0}^{m} Y_{i,j}[m]\epsilon'_{i,j}\right] \\ &= -\sum_{i=0}^{|S|-1}\mu_i\sum_{j=0}^{m} Y_{i,j}[m]. \end{aligned} \tag{16}$$

Combining (14), (15) and (16):

$$P\left(\hat{\mathrm{E}}\mathrm{rr}_Y[n]\right) = \frac{1}{K}\sum_{m=0}^{K-1}\left(\sum_{i=0}^{|S|-1}\sigma_i^2\sum_{j=0}^{m}Y_{i,j}^2[m] + \left(\sum_{i=0}^{|S|-1}\mu_i\sum_{j=0}^{m}Y_{i,j}[m]\right)^2\right). \tag{17}$$

Expressions for the mean and variance can be obtained in a similar fashion:

$$\mu_{\hat{\mathrm{E}}\mathrm{rr}_Y[n]} = \frac{1}{K}\sum_{m=0}^{K-1}\left(\sum_{i=0}^{|S|-1}\mu_i\sum_{j=0}^{m}Y_{i,j}[m]\right), \qquad \sigma^2_{\hat{\mathrm{E}}\mathrm{rr}_Y[n]} = P\left(\hat{\mathrm{E}}\mathrm{rr}_Y[n]\right) - \mu^2_{\hat{E}rr_Y[n]}. \tag{18}$$

The output noise power (17), as well as the mean and the variance, can be expressed more compactly by using vectors **v**, **m**, and matrix **M** as shown in (19)–(23). Once vectors **v**, **m**, and matrix **M** are computed, the estimation of the quantization noise does not require a simulation but the computation of expressions (19)–(21), which is a much faster process,

$$P_o = \frac{1}{K}\left(\boldsymbol{\sigma}^2\cdot\mathbf{v}^T + \boldsymbol{\mu}\cdot\mathbf{M}\boldsymbol{\mu}^T\right), \tag{19}$$

$$\mu_o = \frac{1}{K}\left(\boldsymbol{\mu}\cdot\mathbf{m}^T\right), \tag{20}$$

$$\sigma_o^2 = P_o - \mu_o^2, \tag{21}$$

$$\mathbf{v} \equiv \left\langle \sum_{n=0}^{M-1}\sum_{j=0}^{n}Y_{0,j}^2[n],\ldots,\sum_{n=0}^{M-1}\sum_{j=0}^{n}Y_{|S|-1,j}^2[n]\right\rangle, \tag{22}$$

$$\mathbf{m} \equiv \left\langle \sum_{n=0}^{M-1}\sum_{j=0}^{n}Y_{0,j}[n],\ldots,\sum_{n=0}^{M-1}\sum_{j=0}^{n}Y_{|S|-1,j}[n]\right\rangle, \tag{23}$$

$$\mathbf{M} \equiv \begin{bmatrix} m_{0,0} & \cdots & m_{|S|-1,0} \\ & \ddots & \\ m_{0,|S|-1} & \cdots & m_{|S|-1,|S|-1} \end{bmatrix}, \tag{24}$$

$$m_{i_1,i_2} = \sum_{n=0}^{M-1}\left(\sum_{j_1=0}^{n}Y_{i_1,j_1}[n]\sum_{j_2=0}^{n}Y_{i_2,j_2}[n]\right). \tag{25}$$

The parameterization process is composed of the following steps:

(1) perform a K-step AA simulation adding an affine form $\hat{n}_i$ to each signal i,

(2) compute (22)–(24) using previously collected $Y_{i,j}[n]$.

The error estimation phase can now be executed very quickly by applying (19)–(21).

Please note that

(i) expressions (17)–(22) can be applied to DSP algorithms including differentiable operations (e.g. multiplications, divisions, etc.) by mean of (9) due to the 1st order approximation,

(ii) they are exact for LTI systems in steady state (see the appendix).

4.3. Particularization for LTI Systems. The expressions and the algorithms from the previous subsection can be applied to LTI algorithms, but with a high computational load. In this subsection, we present new expressions to compute the power, mean and variance of the output error for LTI systems in steady state that enable fast estimations.

It is possible to simplify the noise estimation by modifying the expression of the noise terms:

$$\hat{n}_s[n] = \begin{cases} \mu_s + \sqrt{12\sigma^2}\epsilon_{s,n} = \epsilon'_{s,n}, & \text{if } n = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

It can also be inferred that

$$Y_{i,0}[n] = g_i[n]. \tag{27}$$

Therefore, it is possible to rewrite the set of (A.1)–(A.3) in order to relate them to the amplitudes of the error terms at the output of the system $\hat{Y}[n]$ as shown in the following

$$\begin{aligned} {\sigma_o^{\mathrm{LTI}}}^2 &= \boldsymbol{\sigma}^2_{\mathrm{LTI}}\cdot\mathbf{v}^T_{\mathrm{LTI}} \\ \mu_o^{\mathrm{LTI}} &= \boldsymbol{\mu}_{\mathrm{LTI}}\cdot\mathbf{m}^T_{\mathrm{LTI}}, \\ P_o^{\mathrm{LTI}} &= {\sigma_o^{\mathrm{LTI}}}^2 + {\mu_o^{\mathrm{LTI}}}^2, \\ \mathbf{v}_{\mathrm{LTI}} &= \left\langle \sum_{j=0}^{M-1}Y_{0,0}^2[j],\ldots,\sum_{j=0}^{M-1}Y_{0,0_{|S|-1,0}}^2[j]\right\rangle, \\ \mathbf{m}_{\mathrm{LTI}} &= \left\langle \sum_{j=0}^{M-1}Y_{0,0}[j],\ldots,\sum_{j=0}^{M-1}Y_{|S|-1,0}[j]\right\rangle. \end{aligned} \tag{28}$$

5. Benchmarks

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

(i) RGB to YCrCb converter (RGB) [6],

(ii) 8-point IDCT (IDCT_8) [26],

(iii) 2nd-order IIR filter (IIR_8) [26],

(iv) 3rd-order Lattice filter (LAT_3) [39],

(v) 6th-order transposed direct form II delta-operator filter (DEL_6) [40],

(vi) 3 × 3 vector scalar multiplication ($\mathrm{VEC}_{3\times3}$),

(vii) 8 × 8 vector scalar multiplication ($\mathrm{VEC}_{8\times8}$),

(viii) MIMO channel equalizer (EQ) [41],

(ix) a mean power estimator based on a 1st IIR filter (POW),

(x) 1st-order LMS filter (LMS_1) [12],

(xi) 2nd-order LMS filter (LMS_2) [12],

Table 3: Performance of the estimation method: Precision.

	Estimation error							
Benchmark	[120,100)[1] dB		[100,80) dB		[80,60) dB		[60, 40] dB	
	(dB)[2]	(%)[3]	(dB)	(%)	(dB)	(%)	(dB)	(%)
RGB	0.11	0.24	0.09	0.09	0.07	0.17	0.07	0.44
IDCT$_8$	0.11	0.11	0.08	0.42	0.21	0.88	0.27	0.68
IIR$_2^*$	0.04	0.03	0.04	0.04	0.06	0.74	0.04	0.09
LAT$_3^*$	0.24	0.69	0.18	0.33	0.20	0.15	0.19	0.46
DEL$_6^*$	0.03	0.01	0.02	0.03	0.03	0.16	0.16	1.16
VEC$_{3\times3}$	0.07	0.54	0.07	0.11	0.06	0.50	0.09	0.72
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$_1^*$	0.09	0.41	0.14	0.90	0.16	1.74	0.82	6.96
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
VOL$_3^*$	1.14	3.33	0.49	1.84	0.81	6.70	1.43	16.67
All	1.14	0.20	0.49	0.09	0.81	1.26	1.43	3.52

[1] Error constraint, [2] $|10\log(P_{\text{ref}}/P_{\text{est}})|$(max), [3] $|100((P_{\text{ref}}-P_{\text{est}})/P_{\text{ref}})|$ (average),
* It contains loops.

(xii) 5th-order LMS filter (LMS$_5$) [12],

(xiii) 3rd-order Volterra adaptive filter (VOL$_3$)[42].

The main features of the benchmarks are summarized in Table 2, which contains the type of algorithm (LTI or nonlinear, with or without loops), the number of inputs/outputs, the number and type of operations involved, and the total number of signals ($|S|$). The set of benchmarks covers both LTI and nonlinear algorithms, as well as cyclic and acyclic ones. It must be noted that the set of operations is quite complete since it 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, but they also address 4 G MIMO channel equalizing, vector multiplications and adaptive filtering for both linear and nonlinear system identification.

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.

6. Results

The procedure to carry out the tests is as follows:

(1) compute scaling by means of a floating point simulation,

(2) extract noise parameters (22)–(24) performing an AA-based simulation,

(3) perform a WPO as in Figure 2 using a gradient-descent approach,

(4) perform a single FxP bit-true simulation and use it as reference to compute the performance and accuracy of the estimator.

The accuracy obtained by means of a gradient-descent optimization [6] under different SQNR constraints—80 in total, from 40 dB to 120 dB—for the different benchmarks is presented in Table 3. 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 extremely accurate for LTI algorithms. The mean percentage error is smaller than 1.16%, and the maximum relative error is smaller than 0.24 dB. The quality of the estimates is homogenous within the range (40, 120) dB.

The accuracy for nonlinear algorithms shows some degradation. This is expected, since a 1st-order Taylor approximation has been applied (9) in the computation of the quantization noise. Moreover, the presence of loops increases the error in the estimation, since the error due to neglecting Taylor series terms is amplified through the feedback loops. The nonlinear algorithms without loops perform significantly well. The mean percentage error is smaller than 1.52%, and the maximum relative error is smaller than 0.3 dB. This performance is similar to that of LTI algorithms.

The nonlinear algorithms that contain loops have a clearly different behaviour. The mean percentage error is smaller than 16.7%, and the maximum relative error is smaller than 1.43 dB. Now, the accuracy decreases as long

Table 4: Performance of the estimation method: Computation time.

Bench.	FxP Samples	Param. Samples	Param. time (secs)$^+$	No. of estimates (mean)	Estimation-based optim. (secs)$^+$	Simulation-based optim. (secs)$^+$	Speed-up
RGB	20000	1	0.00016	141	0.03	76	x3205
IDCT$_8$	20000	1	0.00031	4575	5.77	13774.81	x2468
IIR$_2^*$	20000	5000	0.88	19	0.02	4.41	x270
LAT$_3^*$	20000	20000	10.80	2276	0.74	2381.51	x3222
DEL$_6^*$	20000	5000	6.31	3930	3.47	11206.08	x3235
VEC$_{3\times3}$	20000	20000	59	150	0.03	66.86	X2122
VEC$_{8\times8}$	20000	20000	330	1739	1.72	2331.79	x1377
EQ	16000	16000	61.64	231	0.12	105.78	x904
POW*	20000	20000	546.14	97	0.02	21.93	x1048
LMS$_1^*$	5000	5000	908.02	712	0.42	163.73	x394
LMS$_2^*$	5000	5000	592.11	1032	0.94	310.93	x331
LMS$_5^*$	5000	5000	1646.38	2547	7.26	1611.46	x221
VOL$_3^*$	5000	5000	212.72	673	0.29	151.13	x526
All	—	—	—	—	—	—	x1486

* It contains loops,

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

as the error constraints get looser. This is due to the aforementioned amplification of the Taylor error terms and also to the fact that the uniformly distributed model for the quantization noise does not remain valid for small SQNRs. The errors due to the quantization noise model introduced by the SQNR ranges used for these experiments are minimum, but, after being propagated through the feedback loops and amplified due to nonlinearities, they become much more noticeable. Anyway, the quality of the estimates is still very high.

The average percentage error is 3.52% which 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 fifth 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, in fact, 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 presented in the last column.

The parameterization time goes from 160 μsecs. 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 born 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 x221 to x3235 are obtained. The average boost is x1486 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 both LTI and nonlinear DSP algorithms.

7. Conclusions

A novel noise estimation method based on the use of Affine Arithmetic has been presented. This method allows to obtain fast and 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 time, leading to significant hardware cost reductions. The method can be applied to differentiable nonlinear DSP algorithms with and without feedbacks.

In brief, the main contributions of the paper are

(i) the proposal of a novel AA-based quantization noise estimation for LTI algorithms,

(ii) the proposal of a novel AA-based quantization noise estimation for nonlinear algorithms with and without feedbacks,

(iii) the average estimation error for LTI systems is smaller than 2%,

(iv) the average estimation error for nonlinear systems is smaller than 17%,

(v) the computation time of WLO is boosted up to x3235 (average of x1486),

The reduction of the computation time of the noise parameterization process, specially in the presence of loops, is to be approached in the near future. Also, the improvement of the quantization model for nonlinear operations is perceived as an interesting research line.

Appendix

Validity of General Expression for Steady-State LTI Algorithms

Expressions (17)–(22) can be applied to DSP algorithms including differentiable operations (e.g., multiplications, divisions, etc.) by means of (9), due to the 1st-order approximation. However, they should be exact for LTI systems and match the well-known expressions for LTI algorithms in steady state,

$$\mu_{\text{LTI}} = \sum_{i=0}^{|S|-1} \mu_i \cdot G_i(1) = \sum_{i=0}^{|S|-1} \mu_i \sum_{n=0}^{\infty} g_i[n], \tag{A.1}$$

$$\begin{aligned}\sigma^2_{\text{LTI}} &= \sum_{i=0}^{|S|-1} \sigma_i^2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| G_i(e^{j\Omega}) \right|^2 d\Omega \\ &= \sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{n=0}^{\infty} g_s^2[n],\end{aligned} \tag{A.2}$$

$$\begin{aligned}P_{\text{LTI}} &= \sigma^2_{\text{LTI}} + (\mu_{\text{LTI}})^2 \\ &= \sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{n=0}^{\infty} g_s^2[n] + \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{n=0}^{\infty} g_i[n] \right)^2,\end{aligned} \tag{A.3}$$

where $G_i(Z)$ and $g_i[n]$ are the transfer function and the impulse response from signal i to the output of the algorithm, respectively. The LTI system is supposed to be causal (for all $n < 0$, $g_i[n] = 0$) and stable ($g_i[n]|_{\to\infty} = 0$).

In LTI systems, the coefficients $Y_{i,j}[n]$ multiplying each $\epsilon'_{i,j}$ depend only on $g_i[n]$ and are equal to

$$Y^{\text{LTI}}_{i,j}[n] = \begin{cases} g_i[n-j], & \text{if } n > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{A.4}$$

Equation (17) turns into

$$\begin{aligned}&P^{\text{LTI}}_{\hat{\text{E}}\text{rr}_Y} \\ &\quad = \frac{1}{K} \sum_{m=0}^{K-1} \left(\sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{j=0}^{n} g_i^2[n-j] + \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{j=0}^{n} g_i[n-j] \right)^2 \right).\end{aligned} \tag{A.5}$$

Note that (A.1)–(A.3) assume that the LTI system is in steady state. Therefore, the transient must be removed from the computation of the MSE. Hereby, (A.5) only matches (A.3), if the affine simulation is performed during M iterations ($M \gg K + K'$), where K' is such that for all $n > K'$, $g_i[n] \approx 0$, and the first K iterations ($K > K'$) are removed from the computation. Thus,

$$\begin{aligned}P^{\text{LTI}}_{\hat{\text{E}}\text{rr}_Y} &= \frac{1}{M-K} \sum_{n=K}^{M-1} \left(\sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{j=0}^{n} g_i^2[n-j] \right. \\ &\qquad \left. + \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{j=0}^{n} g_i[n-j] \right)^2 \right) \\ &\approx \frac{1}{M-K} \sum_{n=K}^{M-1} \left(\sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{j=0}^{K'} g_i^2[n-j] \right. \\ &\qquad \left. + \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{j=0}^{K'} g_i[j] \right)^2 \right) \\ &= \frac{1}{M-K} (M-K) \\ &\quad \cdot \left(\sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{j=0}^{K'} g_i^2[j] \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{j=0}^{K'} g_i[j] \right)^2 \right) \\ &= \sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{j=0}^{K'} g_i^2[j] + \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{j=0}^{K'} g_i[j] \right)^2 \\ &\approx \sum_{i=0}^{|S|-1} \sigma_i^2 \sum_{j=0}^{\infty} g_i^2[j] + \left(\sum_{i=0}^{|S|-1} \mu_i \sum_{j=0}^{\infty} g_i[j] \right)^2 \\ &= P_{\text{LTI}}.\end{aligned} \tag{A.6}$$

Similarly, (18) can be matched to (A.1) and (A.2), respectively, thus validating the approach for LTI algorithms.

Acknowledgments

The authors would like to thank the work of the anonymous reviewers. This work was supported by the Spanish Ministry of Science and Innovation under project TEC2009-14219-C03-02.

References

[1] C. F. Fang, T. Chen, and R. A. Rutenbar, "Floating-point error analysis based on affine arithmetic," in *IEEE International Conference on Accoustics, Speech, and Signal Processing*, pp. 561–564, Hong Kong, April 2003.

[2] A. Gaffar, O. Mencer, W. Luk, P. Cheung, and N. Shirazi, "Floating-point bitwidth analysis via automatic differentiation," in *International Conference on Field Programmable Technology*, pp. 158–165, 2002.

[3] A. A. Gaffar, O. Mencer, W. Luk, and P. Y. K. Cheung, "Unifying bit-width optimisation for fixed-point and floating-point designs," in *Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM '04)*, pp. 79–88, April 2004.

[4] G. Caffarena, G. A. Constantinides, P. Y. K. Cheung, C. Carreras, and O. Nieto-Taladriz, "Optimal combined word-length allocation and architectural synthesis of digital signal processing circuits," *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 5, pp. 339–343, 2006.

[5] F. Catthoor, H. De Man, and J. Vandewalle, "Simulated annealing based optimization of coefficient and data word-lengths in digital filters," *International Journal of Circuit Theory and Applications*, vol. 16, no. 4, pp. 371–390, 1988.

[6] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, pp. 1432–1442, 2003.

[7] W. Sung and K. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 3087–3090, 1995.

[8] G. A. Constantinides and G. J. Woeginger, "The complexity of multiple wordlength assignment," *Applied Mathematics Letters*, vol. 15, no. 2, pp. 137–140, 2002.

[9] G. Caffarena, A. Fernandez, C. Carreras, and O. Nieto-Taladriz, "Fixed-point refinement of OFDM-based adaptive equalizers: a heuristic approach," in *European Signal Processing Conference*, pp. 1353–1356, 2004.

[10] M.-A. Cantin, Y. Savaria, and P. Lavoie, "A comparison of automatic word length optimization procedures," in *IEEE International Symposium on Circuits and Systems*, pp. 612–615, May 2002.

[11] J. A. López, 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 and Systems*, vol. 2, no. 4, pp. 393–408, 2008.

[12] G. Constantinides, "Perturbation analysis for word-length optimization," in *IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 81–90, 2003.

[13] D. Menard, R. Rocher, P. Scalart, and O. Sentieys, "SQNR determination in non-linear and non-recursive fixed-point systems," in *European Signal Processing Conference*, pp. 1349–1352, 2004.

[14] C. Shi and R. W. Brodersen, "A perturbation theory on statistical quantization effects in fixed-point DSP with non-stationary inputs," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 373–376, Vancouver, Canada, May 2004.

[15] D.-U. Lee, A. A. Gaffar, R. C. C. Cheung, O. Mencer, W. Luk, and G. A. Constantinides, "Accuracy-guaranteed bit-width optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 1990–1999, 2006.

[16] D. Menard and O. Sentieys, "A methodology for evaluating the precision of fixed-point systems," in *IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 3152–3155, May 2002.

[17] R. Rocher, D. Menard, N. Herve, and O. Sentieys, "Fixed-point configurable hardware components," *EURASIP Journal on Embedded Systems*, vol. 2006, Article ID 23197, 13 pages, 2006.

[18] S. Kim, K.-I. I. Kum, and W. Sung, "Fixed-point optimization utility for C and C++ based digital signal processing programs," *IEEE Transactions on Circuits and Systems II*, vol. 45, no. 11, pp. 1455–1464, 1998.

[19] M. Willems, H. Keding, T. Grótker, and H. Meyr, "FRIDGE: an interactive fixed-point code generation environment for Hw/Sw-codesign," in *IEEE Conference on Acoustics, Speech and Signal Processing*, pp. 687–690, Munich, Germany, 1997.

[20] M. Chang and S. Hauck, "Precis: a design-time precision aanalysis tool," in *IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 229–238, 2002.

[21] M. L. Chang and S. Hauck, "Précis: a usercentric word-length optimization tool," *IEEE Design and Test of Computers*, vol. 22, no. 4, pp. 349–361, 2005.

[22] R. Cmar, L. Rijnders, P. Schaumont, S. Vernalde, and I. Bolsens, "A methodology and design environment for DSP ASIC fixed point refinement," in *Proceedings of the Conference on Design, Automation and Test in Europe*, p. 56, 1999.

[23] A. Benedetti and P. Perona, "Bit-width optimization for configurable DSP's by multi-interval analysis," in *Proceedings of the 34th Asilomar Conference on Signals, Systems and Computers*, pp. 355–359, November 2000.

[24] S. Mahlke, R. Ravindran, M. Schlansker, R. Schreiber, and T. Sherwood, "Bitwidth cognizant architecture synthesis of custom hardware accelerators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 11, pp. 1355–1371, 2001.

[25] M. Stephenson, J. Babb, and S. Amarasinghe, "Bitwidth analysis with application to silicon compilation," in *SIGPLAN Conference on Programming Language Design and Implementation (PLDI '00)*, pp. 108–120, June 2000.

[26] C. F. Fang, R. A. Rutenbar, and T. Chen, "Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs," in *IEEE/ACM International Conference on Computer Aided Design (ICCAD '03)*, pp. 275–282, November 2003.

[27] J. A. López, G. Caffarena, C. Carreras, and O. Nieto-Taladriz, "Characterization of the quantization properties of similarity-related DSP structures by means of interval simulations," in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, pp. 2208–2212, November 2003.

[28] S. A. Wadekar and A. C. Parker, "Accuracy sensitive word-length selection for algorithm optimization," in *Proceedings of the IEEE International Conference on Computer Design*, pp. 54–61, October 1998.

[29] B. Hayes, "A lucid interval," *American Scientist*, vol. 91, no. 6, pp. 484–488, 2003.

[30] C. Carreras, J. A. López, and O. Nieto-Taladriz, "Bit-width selection for data-path implementations," in *International Symposium on System Synthesis*, p. 114, 1999.

[31] J. Stolfi and L. H. Figueiredo, "Self-Validated Numerical Methods and Applications," Brazilian Mathematics Colloquium Monograph, IMPA, Rio de Janeiro, Brazil, 1997.

[32] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[33] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Truncation noise in fixed-point SFGs," *Electronics Letters*, vol. 35, no. 23, pp. 2013–2014, 1999.

[34] L. B. Jackson, "Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form," *IEEE Transactions on Audio and Electroacoustics*, vol. 18, no. 2, pp. 107–122, 1970.

[35] A. V. Oppenheim and C. J. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 957–976, 1972.

[36] J. López, *Evaluación de los Efectos de Cuantificación en las Estructuras de Filtros Digitales Mediante Técnicas de Simulación Basadas en Extensiones de Intervalos*, Ph.D. thesis, Universidad Politécnica de Madrid, 2004.

[37] J. A. López, C. Carreras, and O. Nieto-Taladriz, "Improved interval-based characterization of fixed-point LTI systems with feedback loops," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1923–1933, 2007.

[38] J. Lopez, G. Caffarena, C. Carreras, and O. Nieto-Taladriz, "Analysis of limit cycles by means of affine arithmetic computer-aided tests," in *European Signal Processing Conference*, pp. 991–994, 2004.

[39] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, Wiley, New York, NY, USA, 1999.

[40] G. Li and Z. Zhao, "On the generalized DFIIt structure and its state-space realization in digital filter implementation," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 4, pp. 769–778, 2004.

[41] A. Fernandez Herrero, A. Jiménez-Pacheco, G. Caffarena, and J. Casajus Quiros, "Design and implementation of a hardware module for equalisation in a 4G mimo receiver," in *International Conference on Field Programmable Logic and Applications (FPL '06)*, pp. 765–768, August 2006.

[42] T. Ogunfunmi, *Adaptive Nonlinear System Identification: The Volterra and Wiener Approaches*, Springer, 2007.