Analytical Approach for Numerical Accuracy Estimation of Fixed-Point Systems Based on Smooth Operations

Romuald Rocher, Daniel Menard, Pascal Scalart, and Olivier Sentieys *Members, IEEE*
University of Rennes 1-INRIA/IRISA
6 rue de Kerampont - 22305 Lannion, France
Email: name@irisa.fr

[1]

***Abstract*—In embedded systems using fixed-point arithmetic, converting applications into fixed-point representations requires a fast and efficient accuracy evaluation. This paper presents a new analytical approach to determine an estimation of the numerical accuracy of a fixed-point system, which is accurate and valid for all systems formulated with smooth operations (e.g. additions, subtractions, multiplications and divisions). The mathematical expression of the system output noise power is determined using matrices to obtain more compact expressions. The proposed approach is based on the determination of the time-varying impulse-response of the system. To speedup computation of the expressions, the impulse response is modelled using a linear prediction approach. The approach is illustrated in the general case of time-varying recursive systems by the Least Mean Square (LMS) algorithm example. Experiments on various and representative applications show the fixed-point accuracy estimation quality of the proposed approach. Moreover, the approach using the linear-prediction approximation is very fast even for recursive systems. A significant speed-up compared to the best known accuracy evaluation approaches is measured even for the most complex benchmarks.**

***Index Terms*—Fixed-point arithmetic, quantization noises, adaptive filters, accuracy evaluation**

I. Introduction

Digital image and signal processing applications are mostly implemented in embedded systems with fixed-point arithmetic to satisfy energy consumption and cost constraints [14], [16], [33], [15]. Fixed-point architectures manipulate data with relatively small word-lengths, which could offer the advantages of a significantly lower area, latency, and power consumption. Moreover, floating-point operators are more complex in their processing of exponent and mantissa and, therefore, chip area and latency are more important than for fixed-point operators.

Nevertheless, applications in their earliest design cycle are usually designed, described and simulated with floating-point data types. Consequently, application must be converted into a fixed-point specification where implementation is considered. To reduce application time-to-market, tools to automate fixed-point conversion are needed. Indeed, some experiments [7] have shown that the manual fixed-point conversion process can represent from 25% to 50% of the total development time. Likewise, in a survey conducted by a design tool provider [19], the fixed-point conversion was identified by a majority of respondents as one of the most difficult aspects during FPGA implementation of a DSP application. The aim of fixed-point conversion tools is to determine the optimized fixed-point specification that minimizes the implementation cost and provides a sufficient computation accuracy to maintain the application performance.

The fixed-point conversion process is divided into two steps to determine the number of bits for both integer and fractional parts. The first step is the determination of the binary-point position from the data dynamic range [36]. The integer part word-length of all data inside the application is determined in order to avoid overflow. The second step is the determination of the fractional part. The limitation of the number of bits for the fractional part leads to an unavoidable error resulting from the difference between the expected value and the real finite precision value. Obviously, numerical accuracy effects must be limited as much as possible to ensure algorithm integrity and application performance. Therefore, this stage of the fixed-point conversion flow corresponds to an optimization process in which the implementation cost is minimized under an accuracy constraint. The fractional part word-length is optimized with an iterative process requiring to evaluated the numerical accuracy numerous times. Therefore precise and fast numerical accuracy estimator is a key element of any efficient fixed-point conversion method.

The numerical accuracy of an application can be obtained using two approaches: fixed-point simulations and analytical approaches. Accuracy evaluation based on fixed-point simulations [2], [20] is very time consuming because a new simulation is needed as soon as a fixed-point format changes in the system resulting in prohibitive optimization time for the fixed-point conversion process. Analytical approaches attempt to determine a mathematical expression of an estimation of a numerical accuracy metric, which leads to short evaluation times compared to fixed-point simulation based approaches. Consequently, the rest of the paper focuses on improving the quality and the application range of analytical approaches for numerical accuracy evaluation.

Numerical accuracy can be evaluated by determining one of the following metrics: quantization error bounds [13], [1], number of significant bits [6] or quantization noise power [27], [31], [5]. The error bound metric (or the number of significant bits) is used for critical systems where the quantization effects could have catastrophic consequences [21], [3]. The error has to be kept small enough to ensure the system behavior within acceptable bounds. The quantization error power is preferred when a slight degradation of application performance due to quantization effects is acceptable. This design approach concerns a great number of digital signal processing applications. Moreover, the quantization error can be linked to the degradation of the application performance using for example the technique presented in [26]. In this paper, the quantization noise power is used as the numerical accuracy metric.

This paper proposes an efficient and accurate analytical approach to estimate the numerical accuracy of all types of fixed-point systems based on smooth operations (an operation is considered to be smooth if the output is a continuous and differentiable function of its inputs, as in the case of arithmetic operations). From the signal flow graph of the application, the associated tool automatically generates the source code implementing the analytical expression of the output quantization noise power. In this expression, the gains between the different noise sources and the output are computed from the impulse response (IR), which can be time-varying. This expression

is unbiased but introduces infinite sums. To reduce the computation time of the gains, an approach based on linear prediction is also proposed. The approach is based on a matrix model which leads to a compact expression in the case of algorithms expressed with vectors and matrices, such as the Fast Fourier Transform (FFT) and the Discrete Cosine Transform (DCT). Compared to approaches based on fixed-point simulations, the time required to evaluate the fixed-point accuracy is also dramatically reduced. Existing analytical approaches are only valid for linear and time invariant (LTI) systems [22], [23], or for non-LTI and non recursive systems [24]. In [4], non-LTI systems with feedbacks are handled. But, as mentioned in [4], the execution time can be quite long for systems containing feedbacks. Indeed, to obtain an accurate estimation, this execution time depends on the length of the system impulse response. Compared to the last-mentioned approach, a significant speed-up is obtained with the proposed approach thanks to the proposed linear prediction approach.

This paper is organized as follows. In Section II, existing methods to evaluate the numerical accuracy of fixed-point applications are detailed. The comparison of simulation-based and analytical approaches highlights the benefits of analytical approaches. In Section III, the proposed approach for numerical accuracy evaluation is defined. The quantization noise propagation model is presented and the output noise power expression is expressed. A technique based on linear prediction is also presented to reduce the time to obtain the power expression. In Section IV, the approach is tested on several LTI and non LTI systems. The quality of the estimation and the time to obtain the power expression are evaluated on various applications and benchmarks. Finally, Section V concludes the paper.

II. Existing Approaches for Numerical Accuracy Evaluation

In the fixed-point conversion process, the numerical accuracy evaluation is the most critical part, mainly because this evaluation is successively performed in the iterative word-length optimization process. The numerical accuracy can be evaluated using analytical or simulation-based approaches.

A. Accuracy Evaluation Based on Fixed-Point Simulations

In fixed-point simulation-based approaches, the application performance and the numerical accuracy are directly determined from fixed-point (bit-true) simulations. To obtain accurate results, the simulations must use a high number of samples N_{samp}. Let N_{ops} denote the number of operations in the algorithm description, N_I—the number of times the accuracy is evaluated and τ—the simulation time for one sample. Therefore, the total system simulation time is

$$T_{sim} = \tau N_{samp} N_{ops} N_I. \tag{1}$$

To reduce word-length optimization time, approaches based on fixed-point simulations aim at reducing the values of some parameters of (1). First, fixed-point simulators have to minimize the computing time τ by using efficient data types to simulate fixed-point arithmetic. Most of them use object oriented language and operator overloading [20] to implement fixed-point arithmetic. However, the fixed-point execution time of an algorithm is significantly longer than the one of a floating-point simulation [12]. To reduce this overhead, new data types have been introduced in [20]. They use more efficiently the floating-point units of general-purpose processors. With this new data type, the fixed-point simulation time is still 7.5 times longer than the one obtained with floating-point data types for a simple application like a fourth-order Infinite Impulse Response (IIR) filter [20]. However, the optimization time to accelerate the fixed-point simulations is not given in [20], which could actually annihilate the obtained gain for the fixed-point simulation. The approach proposed in [30] was developed for MATLAB code but its simulation time is still significantly higher.

Finally, some methods [2] attempt to reduce N_i, the number of iteration of the optimization process by limiting the search space, and thus leading to sub-optimal solutions.

B. Analytical Accuracy Evaluation

The general idea of analytical approaches is to determine a mathematical expression that provides estimations of the output quantization noise power. The computation of this analytical expression could be time-consuming, but it is executed only once. Then, the numerical accuracy is determined very quickly by evaluating the mathematical expression for a given word-length of each data. Existing analytical approaches for numerical accuracy evaluation are based on perturbation theory [8], [34]. The quantization of a signal x generates a noise b_x considered as a signal perturbation. Let x_i and y be respectively the n inputs and the output of an operation f, and b_{x_i} and b_y their perturbations. The perturbation of the output value of a differentiable operation is given by

$$b_y = b_{x_1} \frac{df}{dx_1} + ... + b_{x_n} \frac{df}{dx_n}. \tag{2}$$

Then, the expression of the noise power P_b corresponding to the second-order moment of the output quantization noise can be expressed as

$$P_b = \sum_{i=1}^{N_e} \sigma_{b_i}^2 K_i + \sum_{i,j}^{N_e} m_{b_i} m_{b_j} L_{ij}, \tag{3}$$

where N_e is the number of quantization noises, m_{b_i} and $\sigma_{b_i}^2$ are respectively the mean value and the variance of the noise b_i and the terms K_i and L_{ij} depend on system characteristics. The expressions of K_i and L_{ij} differ according to the considered approach and will be presented in the following subsection. Determining the terms K_i and L_{ij} is crucial for the analytical accuracy evaluation process.

1) Simulation-Based Computation: Approaches based on hybrid techniques [31], [10], [17] have been proposed to compute the coefficients K_i and L_{ij} of (3) from a set of simulations. In [31], these $N_e \times (N_e + 1)$ coefficients are obtained by solving a linear system in which K_i and L_{ij} are the variables. The other elements of (3) are determined by carrying out fixed-point simulations. The word-lengths of the data are set to control each quantizer and to analyze its influence on the output. For each simulation, the output noise power is evaluated. The statistical parameters (mean and variance) of each noise source b_i are extracted from the data word-length. At least $N_e \times (N_e + 1)$ fixed-point simulations are required to solve the system of linear equations. A similar approach to [31] is used in [17] to obtain the coefficients by simulation.

2) Affine Arithmetic Based Simulations: In approach based on affine arithmetic [4], the coefficients K_i and L_{ij} are computed with an affine arithmetic based simulation. This approach was proposed for LTI in [22], [23] and recently, for non-LTI systems in [4].

In affine arithmetic, a data x is defined by a linear relation

$$x = x_0 + \epsilon_1 x_1 + ... + \epsilon_n x_n, \tag{4}$$

where ϵ_i is an uncertainty term included in the interval $[-1, 1]$ and such that ϵ_i is independent of ϵ_j $\forall i \neq j$. Each noise source is defined by an affine form whose behavior is controlled by the mean and the variance of the noise source. Then, noise source expressions are propagated throughout the system thanks to an affine arithmetic based

simulation. The values of K_i and L_{ii} are extracted from the affine form of the output noise. The approach proposed in [4] is general and can handle systems based on smooth operations. This estimation is accurate and the execution time is low for non-recursive systems. Nevertheless, in the case of recursive systems, several iterations for the affine arithmetic based simulation are required to converge to stable values. Thus, the execution time depends on the length of the impulse response and on the number of noise sources. As mentioned in [4], the execution time can be quite long for systems containing feedback loops.

3) Approaches Based on System Characteristics: The approaches presented in [11], [25] provide an approximation of the mathematical expression of the output quantization noise power in the case of Linear and Time Invariant (LTI) systems. The principle of [25] is based on the system transfer functions and the expressions of the coefficients K_i and L_i are

$$K_i = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_i(e^{j\Omega})|^2 d\Omega, \quad (5)$$

$$L_{ij} = H_i(1)H_j^*(1), \quad (6)$$

where $H_i(z)$ is the transfer function between the white noise source b_i and the system output. In [25], the authors proposed a technique to automatically obtain the transfer function from the application description.

For non-LTI and non-recursive systems, an extended approach was presented in [24]. Each noise source $b_i(n)$ is propagated through M_i operations υ_{M_i}, which leads to the calculation of the output noise $b_i'(n)$. The noise $b_i'(n)$ is the product of each noise source $b_i(n)$ and of different signals α_k associated to operations υ_{M_i} included in noise source $b_i(n)$ propagation. So, K_i and L_{ij} terms are equal to

$$K_i = E[\prod_{k=1}^{M_i} \alpha_k^2] \quad (7)$$

$$L_{ij} = E\left[\prod_{k=1}^{M_i}\prod_{g=1}^{M_j} \alpha_k\alpha_g\right]. \quad (8)$$

It is noteworthy that this approach is adequate for both non-LTI and non-recursive systems, but not to non-LTI and recursive systems such as the clan of adaptive filters.

C. Summary

While approaches based on fixed-point simulations lead to very long computing time and thus require to limit the search space during word-length optimization, analytical approaches are significantly less time-consuming. Unfortunately, they are limited to a reduced class of systems and are not always efficient in terms of execution time. To cope with the lack of efficient and general approaches, an analytical approach for all types of systems based on smooth operations is presented in the next section.

III. Proposed Accuracy Evaluation Approach

In this section, a new approach for evaluating the quantization noise at the output of any system based on smooth operations (addition, subtraction, multiplication and division) is detailed. First, the models associated with the quantization process and the quantization noise propagation model are introduced. Then, the modelling of the system is presented and the output noise power expression is detailed.

A. Quantization Noise Models

1) Quantization Noise Source: The quantization process can be modelled by the sum of the original signal and of a uniformly distributed white noise [35], [32]. This quantization noise is uncorrelated with the signal and the other noise sources. Such a model is valid provided that the signal has a sufficiently large dynamic range compared to the quantization step. According to the quantization mode, the noise Probability Density Function (PDF) will differ. Three quantization modes are usually considered: truncation, conventional rounding, and convergent rounding. In the truncation mode, the Least Significant Bits (LSB) are directly eliminated. The resulting number is always smaller than or equal to the number obtained before quantization, and, therefore, the quantization noise is always positive. Consequently, the mean of the quantization noise is not equal to zero. To reduce the bias due to truncation, the rounding quantization mode is often used. In conventional rounding, the data are rounded to the nearest value representable in the reduced-accuracy format. For numbers located at the midpoint between two consecutive representable values, the data are rounded-up always to the larger output value. This technique leads to a (small) bias for the quantization noise. To eliminate this quantization noise bias, the convergent rounding can be used as well. In this case, the numbers located at the midpoint between two consecutive representable values are, with equal probability, rounded to the higher or lower output value.

Let w_{FP} denote the number of bits for the fractional part after the quantization process and k—the number of bits eliminated during the quantization. The quantization step q after the quantization is equal to $q = 2^{-w_{FP}}$. The quantization noise mean and variance are given in Table I for the three considered quantization modes [9], [28].

Quantization mode	Truncation	Conventional rounding	Convergent rounding
Mean	$\frac{q}{2}(1-2^{-k})$	$\frac{q}{2}(2^{-k})$	0
Variance	$\frac{q^2}{12}(1-2^{-2k})$	$\frac{q^2}{12}(1-2^{-2k})$	$\frac{q^2}{12}(1+2^{-2k+1})$

TABLE I
First- and second-order moments for the three considered quantization modes.

2) Quantization Noise Propagation: The aim of this part is to define the model for the propagation of the quantization noise for smooth operations. We specifically concentrate on an operation f with two inputs x and y and on the calculation of its output $z = f(x, y)$. The finite precision values $\widehat{x}$, $\widehat{y}$ and $\widehat{z}$ are respectively the sum of infinite precision values x, y and z with their associated quantization noises b_x, b_y and b_z. The propagation model is then defined such as b_z is a linear combination of b_x and b_y

$$b_z = \alpha_1 b_x + \alpha_2 b_y. \quad (9)$$

α_1 and α_2 are obtained from a first order Taylor development at the output of the differentiable operator f defined as

$$z = f(\widehat{x},\widehat{y}) = f(x,y) + (\widehat{x}-x)\frac{\partial f}{\partial x}(x,y) + (\widehat{y}-y)\frac{\partial f}{\partial y}(x,y). \quad (10)$$

Thus, the expressions of α_1 and α_2 are

$$\alpha_1 = \frac{\partial f}{\partial x}(x,y) \quad \text{and} \quad \alpha_2 = \frac{\partial f}{\partial y}(x,y). \quad (11)$$

Operation	α_1	α_2
$z = x \pm y$	1	$\pm$ 1
$z = x \times y$	y	x
$z = \frac{x}{y}$	$\frac{1}{y}$	$-\frac{x}{y^2}$

TABLE II
VALUES α_1 AND α_2 OF EQUATION (9) FOR ARITHMETIC OPERATIONS

Operation	$\mathbf{A_x}$	$\mathbf{D_x}$	$\mathbf{A_y}$	$\mathbf{D_y}$
$\mathbf{Z = X \pm Y}$	1	1	$\pm$1	1
$\mathbf{Z = X \times Y}$	1	$\mathbf{Y}$	$\mathbf{X}$	1

TABLE III
VALUES $\mathbf{A}$ AND $\mathbf{D}$ OF EQUATION (12) FOR OPERATIONS $\{+, -, \times\}$

The values of α_1 and α_2 are summarized in Table II for the arithmetic operations.

For non-scalar noise sources (vectors or matrix), the model of (9) is not valid anymore. For example, in the case of the multiplication of two matrices $\mathbf{X}$ and $\mathbf{Y}$ associated with noise matrices $\mathbf{B_x}$ and $\mathbf{B_y}$, the result $\mathbf{Z}$ is equal to $\mathbf{Z} = \mathbf{XY}$. Then, the output noise matrix $\mathbf{B_z}$ is given by $\mathbf{B_z} = \mathbf{B_x}.\mathbf{Y} + \mathbf{X}.\mathbf{B_y}$.

In the rest of the paper, upper-case letters in bold will represent matrices, lower-case letters in bold will represent vectors, and italic will represent scalars.

Since commutativity is not valid for non-scalar multiplications and since we seek to obtain a general model, each noise source on the operation input is multiplied by a signal term on the left or on the right. Therefore, in the general case, the noise source propagation is expressed as the multiplication of each noise input by two matrices ($\mathbf{A}$ on the left and $\mathbf{D}$ on the right) as

$$\mathbf{B_z} = \mathbf{A_x}\mathbf{B_x}\mathbf{D_x} + \mathbf{A_y}\mathbf{B_y}\mathbf{D_y}. \tag{12}$$

Table III defines values of $\mathbf{A}$ and $\mathbf{D}$ according to the type of operation. For the mathematical expressions of the general approach, $\mathbf{A}$ and $\mathbf{D}$ will be considered as matrices. However, when applied to different examples, depending on the application context, these terms will be noted $\mathbf{A}$ and $\mathbf{D}$ for matrices, $\mathbf{a}$ and $\mathbf{d}$ for vectors and a and d for scalars.

To illustrate the proposed approach, the example of a FIR (Finite Impulse Response) filter is firstly considered. In this case, $\mathbf{x}(n)$ is an N-size input vector, $\mathbf{b_x}(n)$ is its associated noise, and $\mathbf{h}$ is the filter coefficient vector. So, the output noise $b_y(n)$ due to input noise propagation is equal to

$$b_y(n) = \mathbf{h}^t\mathbf{b_x}(n). \tag{13}$$

The terms $\mathbf{a_x}$ and d_x are therefore given by

$$\mathbf{a_x} = \mathbf{h}^t, \tag{14}$$
$$d_x = 1. \tag{15}$$

To illustrate the need of the two matrices $\mathbf{A}$ and $\mathbf{D}$, the adaptive filter based on the Affine Projection Algorithm (APA) is then considered. APA has been proposed in [29] to have a faster convergence compared to the Least Mean Square (LMS) algorithm and to reduce complexity compared to Recursive Least Square (RLS) algorithm. Let $\mathbf{X}(n)$ denote an $N \times K$ matrix made-up of the K last observation vectors of the input, $\mathbf{w}(n)$—the coefficient vector and $\mathbf{e}(n)$ the error vector. The updated coefficient equation is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{X}(n)(\mathbf{X}^t(n)\mathbf{X}(n))^{-1}\mathbf{e}(n). \tag{16}$$

The noise matrix $\mathbf{B_x}(n)$ associated to the inversion propagation matrix $(\mathbf{X}^t(n)\mathbf{X}(n))^{-1}$ is then multiplied by two terms

$$\mathbf{A} = \mu\mathbf{X}(n), \tag{17}$$
$$\mathbf{d} = \mathbf{e}(n). \tag{18}$$

B. System Model

Each noise source propagates throughout the system and contributes to the global output noise. In this section, the system characterizing each noise propagation is defined to compute afterward the expression of the system output noise power. Let N_e denote the number of noise sources inside the system. As shown in Table II, the noise expression does not contain crossed noise terms. So, each noise source $b_i(n)$ at time n contributes to the output noise by a noise $b_i'(n)$ that is independent of all other noise terms. The system output noise $b_y(n)$ is the sum of all contributions as presented in Figure 1 and expressed by

$$b_y(n) = \sum_{i=1}^{N_e} b_i'(n). \tag{19}$$

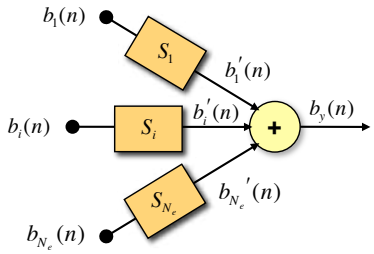

Fig. 1. System noise model for N_e input noises $b_i(n)$ and output noise $b_y(n)$

Each noise contribution $b_i'(n)$ is obtained from the propagation of noise sources $b_i(n)$ through the system S_i. Thus, to determine the complete expression of the output noise $b_y(n)$, each subsystem S_i must be characterized analytically.

1) System Characterization: The contribution $b_i'(n)$ of the noise source $b_i(n)$ depends on the previous samples $b_i(n-k)$ of the noise source with $k \in \{1, 2, ..Q_i\}$. If the system S_i is recursive, the noise $b_i'(n)$ depends on its previous samples $b_i'(n-m)$ with $m \in \{1, 2, ..P_i\}$. Thus, the general expression of the quantization noise $b_i'(n)$ is

$$b_i'(n) = \sum_{k=0}^{Q_i} g_i(k)b_i(n-k) + \sum_{m=1}^{P_i} f_i(m)b_i'(n-m), \tag{20}$$

where $g_i(k)$ represents the contribution of noise source b_i at time $(n-k)$ to the output noise and $f_i(m)$—the contribution of noise b_i' at time $(n-m)$. The terms f_i and g_i may be time-varying and

depend on the system implementation. For LTI systems, the terms f_i and g_i are constant and correspond to filter coefficients.

Nevertheless, to compute the output noise power expression from the statistical parameters of the noise $b_i(n)$, (20) must be developed to express contribution $b'_i(n)$ with only input noise terms b_i. This development introduces the time-varying impulse response associated with the system S_i.

2) Time-Varying Impulse Response: In this part, the time-varying impulse response h_i of the system S_i is determined. By developing the recurrence in (20), the noise term becomes

$$b'_i(n) = \sum_{k=0}^{n} h_i^{(n)}(k) b_i(n-k) \tag{21}$$

where $h_i^{(n)}(k)$ is the time-varying impulse response of the system S_i which formulates the noise $b'_i(n)$ only from the samples of the noise source $b_i(k)$. The term $h_i^{(n)}(k)$ represents the contribution of the noise source $b_i(n-k)$ at time $n-k$ to generate $b'_i(n)$ at time n. For the sake of clarity, $h_i^{(n)}(k)$ is noted by $h_i(k)$ since output time is always considered at time n. This impulse response is obtained from the terms f_i and g_i with

$$h_i(k) = \sum_{j=1}^{P_i} f_i(j) h_i(k-j) + g_i(k) \tag{22}$$

(22) was obtained in the case of scalar noise sources. The model can be extended to vector (or matrix) noise sources by using vector (or matrix) terms for the time-varying impulse response. In this case, the time-varying impulse response is equivalent to the multiplication of the noise source $\mathbf{B_i}(k)$ (to be more general, noises are considered as matrices in the following) by two terms $\mathbf{A_i}(k)$ and $\mathbf{D_i}(k)$. In case of non-scalar noise source, (21) can be rewritten as

$$\mathbf{B'_i}(n) = \sum_{k=0}^{n} \mathbf{A_i}(k)\mathbf{B_i}(n-k)\mathbf{D_i}(k). \tag{23}$$

The output noise $\mathbf{B_y}(n)$ is finally the sum of all noise source contributions and can be expressed as

$$\mathbf{B_y}(n) = \sum_{i=1}^{Ne} \sum_{k=0}^{n} \mathbf{A_i}(k)\mathbf{B_i}(n-k)\mathbf{D_i}(k). \tag{24}$$

The output noise $\mathbf{B_y}(n)$ and input noises $\mathbf{B_i}(n)$ have formal dimensions of M_yxN_y and M_ixN_i. So, $\mathbf{A_i}(k)$ and $\mathbf{D_i}(k)$ are M_yxM_i and N_ixN_y matrices. Their numerical values depend on the system characteristics and can be computed from a analysis of the considered application.

C. Output Noise Power Expression

1) Noise Power: The output noise power P_b is defined as the second order moment of (24) and can be expressed as

$$P_b = E[Tr(\mathbf{B_y}(n)\mathbf{B_y}^t(n))], \tag{25}$$

which is equivalent to

$$P_b = \sum_{i=1}^{Ne}\sum_{j=1}^{Ne}\sum_{k=0}^{n}\sum_{m=0}^{n} E\Big[Tr\big(\mathbf{A_i}(k)\mathbf{B_i}(n-k)\mathbf{D_i}(k) \mathbf{D_j}^t(m)\mathbf{B_j}^t(n-m)\mathbf{A_j}^t(m)\big)\Big]. \tag{26}$$

Equation (26) can be split into three parts by separating the noise $\mathbf{B_i}$ from the other noises $\mathbf{B_j}$ when $j \neq i$ and the terms $\mathbf{B_i}(n-k)$ from $\mathbf{B_i}(n-m)$ when $m \neq k$.

Theorem 1: Each term of $\mathbf{D_i}(k)\mathbf{D_i}^t(k)$ is less or equal than its trace value.

Proof: The matrix $\mathbf{D_i}(k)\mathbf{D_i}^t(k)$ is real, symmetric and positive and therefore diagonalizable. Thus, eigenvalues of $\mathbf{D_i}(k)\mathbf{D_i}^t(k)$ are positive, and each of them is less than the trace value. Indeed, this matrix $\mathbf{D_i}(k)\mathbf{D_i}^t(k)$ can be approximated by its trace $Tr(\mathbf{D_i}(k)\mathbf{D_i}^t(k))$. In the case where the noises are vectors, then $\mathbf{D_i}(k)\mathbf{D_i}^t(k)$ is a scalar and is equal to its trace. ■

Moreover, by applying Theorem 1, (26) can be reformulated and leads to (27) given at page 6.

Following the method described in [5], the expected value of the noise and signal terms are computed separately. For the two matrices $\mathbf{A}$ and $\mathbf{D}$ made-up of signal terms and for a noise $\mathbf{B}$ non-correlated with these two signal terms, the following relation is used: $E[\mathbf{ABD}] = E[\mathbf{A}E[\mathbf{B}]\mathbf{D}]$. Moreover, the different quantization noises $\mathbf{B_i}$ are assumed to be white and independent so the intercorrelation between two noises $\mathbf{B_i}$ and $\mathbf{B_j}$ is equal to

$$\begin{aligned}\phi_{\mathbf{B_iB_j}}(k-m) &= E[\mathbf{B_i}(n-k)\mathbf{B_j}(n-m)] \\ &= \sigma_{b_i}^2\delta(i-j)\delta(k-m)\mathbf{I_N} + m_{b_i}m_{b_j}\mathbf{1_N}\end{aligned} \tag{28}$$

where m_{b_i} and $\sigma_{b_i}^2$ represent mean and variance of each element of input noises $\mathbf{B_i}(n)$, $\mathbf{I_N}$ is the identity matrix and $\mathbf{1_N}$ is the N-size matrix composed of 1. Thus, the output noise power P_b can be summarized as (29) given at page 6.

By introducing the terms K_i and L_{ij}, (29) is reformulated as

$$P_b = \sum_{i=1}^{Ne} \sigma_{b_i}^2 K_i + \sum_{i=1}^{Ne}\sum_{j=1}^{Ne} m_{b_i}m_{bj}L_{ij}. \tag{30}$$

K_i represents the gain on the variance for the noise source b_i and is equal to

$$K_i = \sum_{k=0}^{n} E\Big[Tr\big(\mathbf{D_i}(k)\mathbf{D_i}^t(k)\big)Tr\big(\mathbf{A_i}(k)\mathbf{A_i}^t(k)\big)\Big]. \tag{31}$$

L_{ij} represents the gain on the mean for the couple of noise sources b_i and b_j and is equal to

$$L_{ij} = \sum_{k=0}^{n}\sum_{m=0}^{n} E\Big[Tr\big(\mathbf{A_i}(k)\mathbf{1_N}\mathbf{D_i}(k)\mathbf{D_j}^t(m)\mathbf{1_N}^t\mathbf{A_j}^t(m)\big)\Big]. \tag{32}$$

The expressions of K_i and L_{ij} are obtained by a single floating-point simulation since they only involve signal terms. All the terms included in these expressions are stored from this single floating-point simulation and the terms K_i and L_{ij} are then computed. These terms are independent from noise sources and lead to constants in the output noise power expression. Noise statistics m_{b_i} and $\sigma_{b_i}^2$ depend on fixed-point formats and are the variables describing the output noise power expression. They are computed from the data word-length of each data and operation in the application graph.

It should be noted that (29) is unbiased since no restrictive assumption was made about the system. K_i and L_{ij} are defined by infinite sums. However, different cases must be taken into account. If the system is LTI, the matrices $\mathbf{A}$ and $\mathbf{D}$ are constant, (31) and (32) can be simplified, and it leads to the results obtained in equation (5). If the system is not recursive, the length of the time-varying impulse responses is finite and the sums in (29) are finite and can be computed directly.

For the recursive case, these infinite sums are approximated by computing the first p elements of the sum, and therefore (31) and (32) are computed from $k = 0$ to p with p depending on the signal correlation inside the terms K and L. Nevertheless, according to the different experimentations that have been carried out, a number p equal to or less than 500 leads to very realistic and accurate results. Moreover, to compute (29), the ergodic assumption is used so that

$$\begin{aligned} P_b &= \sum_{i=1}^{Ne}\sum_{k=0}^{n} E\Big[Tr(\mathbf{D_i}(k)\mathbf{D_i}^t(k))Tr\big(\mathbf{A_i}(k)\mathbf{B_i}(n-k)\mathbf{B_i}^t(n-k)\mathbf{A_i}^t(k)\big)\Big] \\ &+ \sum_{i=1}^{Ne}\sum_{\substack{k=0\\ m\neq k}}^{n} E\Big[Tr\big(\mathbf{A_i}(k)\mathbf{B_i}(n-k)\mathbf{D_i}(k)\mathbf{D_i}^t(m)\mathbf{B_i}^t(n-m)\mathbf{A_i}^t(m)\big)\Big] \\ &+ \sum_{\substack{i=1\\ j\neq i}}^{Ne}\sum_{\substack{k=0\\ m=0}}^{n} E\Big[Tr\big(\mathbf{A_i}(k)\mathbf{B_i}(n-k)\mathbf{D_i}(k)\mathbf{D_j}^t(m)\mathbf{B_j}^t(n-m)\mathbf{A_j}^t(m)\big)\Big] \end{aligned} \quad (27)$$

$$\begin{aligned} P_b &= \sum_{i=1}^{Ne}\sum_{k=0}^{n} \sigma_{b_i}^2 E\Big[Tr(\mathbf{D_i}(k)\mathbf{D_i}^t(k))Tr\big(\mathbf{A_i}(k)\mathbf{A_i}^t(k)\big)\Big] \\ &+ \sum_{i=1}^{Ne}\sum_{j=1}^{Ne} m_{b_i}m_{b_j}\sum_{k=0}^{n}\sum_{m=0}^{n} E\Big[Tr\big(\mathbf{A_i}(k)\mathbf{1_N}\mathbf{D_i}(k)\mathbf{D_j}^t(m)\mathbf{1_N}^t\mathbf{A_j}^t(m)\big)\Big] \end{aligned} \quad (29)$$

the statistical mean is replaced by a time average over N_t samples to get a realistic result. In practice, a number N_t equal to 100 leads to accurate modelling properties, as it will be shown in Section IV. The complexity of the proposed method has been analyzed and leads to a number of multiplications N_{mult} and additions N_{add} equal to

$$N_{mult} = N_t.p\sum_i (2M_yM_iN_i + M_yN_y(M_i+N_i)) \quad (33)$$

$$N_{add} = N_t.p\sum_i (2M_yM_iN_i + M_yN_y(M_i+N_i) + M_yN_y) \quad (34)$$

where M_yxN_y corresponds to the output noise matrix $\mathbf{B_y}(n)$ size and M_ixN_i those of input noise $\mathbf{B_i}(n)$. In the next paragraph, an approach to reduce the complexity of the infinite sum computation without a significant loss of accuracy is presented.

2) Linear Prediction Approach: To reduce complexity, a linear prediction approach is proposed to replace the infinite sums. (22) has shown that impulse response terms $h_i(k)$ are defined by a recurrent equation where the terms P_i corresponds to the maximal delay for the recursive part. For LTI systems, terms f and g are directly equal to transfer function coefficients. Therefore, the rest of the section will focus on non-LTI recursive systems.

As the impulse response is time-varying, the relation (22) is non linear. The aim is then to linearize this expression with prediction coefficients λ_i to obtain an estimate $\widehat{h}_i(k)$ of the impulse response $h_i(k)$ equal to

$$\widehat{h}_i(k) = \sum_{m=1}^{P_i} \lambda_{i_m} h_i(k-m). \quad (35)$$

Computing the coefficient vector $\lambda_i = [\lambda_{i_1}...\lambda_{i_{P_i}}]^t$ gives a model of the impulse response term $h_i(P_i+Q_i-1)$ with the aim to minimize the mean square error e_i between $h_i(P_i+Q_i-1)$ and its estimate value $\widehat{h}_i(P_i+Q_i-1)$ defined as

$$E[e_i^2] = E[(h_i(P_i+Q_i-1) - \widehat{h}_i(P_i+Q_i-1))^2]. \quad (36)$$

For the sake of clarity on e_i, a term $k = P_i+Q_i-1$ is introduced and the index i is removed. Thus, the mean square error is defined as

$$E[e^2] = E[(h(k) - \widehat{h}(k))^2]. \quad (37)$$

The derivative of the mean square error, with respect to the coefficients λ_j, $\forall j \in [1,P]$, is

$$\frac{\partial E[e^2]}{\partial \lambda_j} = -2E\Big[h(k)h(k-j) - \sum_m \lambda_m h(k-m)h(k-j)\Big]. \quad (38)$$

These P prediction coefficients λ_j can be determined by setting (38) to zero, leading to the solution λ_{opt} defined as

$$\lambda_{opt} = \mathbf{U}^{-1}\mathbf{J}, \quad (39)$$

with

$$\mathbf{U} = \begin{pmatrix} E[h^2(k-1)] & \dots & E[h(k-1)h(k-P)] \\ E[h(k-1)h(k-2)] & \dots & E[h(k-2)h(k-P)] \\ \vdots & & \\ E[h(k-1)h(k-P+1)] & \dots & E[h(k-P+1)h(k-P)] \\ E[h(k-1)h(k-P)] & \dots & E[h^2(k-P)] \end{pmatrix} \quad (40)$$

and

$$\mathbf{J} = \begin{pmatrix} E[h(k)h(k-1)] \\ E[h(k)h(k-2)] \\ \dots \\ E[h(k)h(k-P+1)] \\ E[h(k)h(k-P)] \end{pmatrix} \quad (41)$$

To determine $\mathbf{J}$ and $\mathbf{U}$, the first values of the impulse response must be known. Therefore, the approach to determine the prediction coefficients is as follows:

1) The first $P+Q-1$ terms of the time-varying impulse response are determined using (22), with Q and P representing respectively the maximal number of delays for the non-recursive and the recursive part of the system S.
2) The two matrices $\mathbf{U}$ et $\mathbf{J}$ are determined.
3) The prediction coefficients $\lambda_{opt} = [\lambda_1....\lambda_P]$ are determined by $\lambda_{opt} = \mathbf{U}^{-1}\mathbf{J}$.

The coefficients λ_{opt} are determined according to Appendices A and B and the infinite sums can therefore be written as

$$\sum_{m=0}^{\infty} h(m) = \frac{1}{P}\sum_{j=1}^{P}\Big(\sum_{m=0}^{Q-2}\boldsymbol{\omega}_m + (\mathbf{I_P}-\mathbf{\Lambda})^{-1}\boldsymbol{\omega}_{Q-1} + \boldsymbol{\theta}\Big)_{(j)} \quad (42)$$

$$\sum_{m=0}^{\infty} h(m)^2 = \frac{1}{P}Tr\left(\sum_{m=0}^{Q-2}\boldsymbol{\omega}_m\boldsymbol{\omega}_m^t + \mathbf{S} + \mathbf{\Phi}\right) \quad (43)$$

where

$$\boldsymbol{\omega}_m = \begin{pmatrix} h(m) \\ h(m+1) \\ \vdots \\ h(m+P-1) \end{pmatrix} \tag{44}$$

$$\boldsymbol{\Lambda} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & \dots & 0 & 1 \\ \lambda_P & \lambda_{P-1} & \dots & \dots & \lambda_1 \end{pmatrix} \tag{45}$$

$$\boldsymbol{\theta}_r = \sum_{k=0}^{r-2} h(k)^2 \tag{46}$$

$$\boldsymbol{\Phi}_{r,r} = \sum_{k=0}^{r-2} h(k)^2 \tag{47}$$

and with matrix $\mathbf{S}$ being the solution of

$$\mathbf{S} - \boldsymbol{\Lambda}\mathbf{S}\boldsymbol{\Lambda}^t = \boldsymbol{\omega}_{Q-1}\boldsymbol{\omega}_{Q-1}^t. \tag{48}$$

In practice, this model is applied to matrices $\mathbf{A}$ and $\mathbf{D}$ for each noise source. The complexity of this method is computed and leads to a number of multiplications N_{mult} and additions N_{add} equal to

$$\begin{aligned} N_{mult} &= \sum_i N_t \Big[(P_i + Q_i - 1)\big(2M_y M_i N_i + M_y N_y (M_i + N_i)\big) \Big] \\ &+ P_i(M_y + N_y)(M_i + N_i) + P_i^2 \end{aligned} \tag{49}$$

$$\begin{aligned} N_{add} &= \sum_i N_t \Big[(P_i + Q_i - 1)\big(2M_y M_i N_i + M_y N_y (M_i + N_i)\big) \Big] \\ &+ 4P_i \end{aligned} \tag{50}$$

D. Accuracy Evaluation Tool

The proposed approach has been implemented in a software tool to automate this process, its synoptic is given in Figure 2. Our numerical accuracy evaluation tool generates the analytical expression of the output quantization noise from the signal flow graph (SFG) of the application. This analytical expression is implemented through a C function having the fractional part $\mathbf{w_{FP}}$ of all data word-lengths as input parameters. This C code can be compiled and dynamically linked to the fixed-point conversion tool for the optimization process. The accuracy evaluation tool uses a technique similar to the one proposed in [25]. The first step transforms the application SFG G_s into an SFG G_{ns} representing the application at the quantization noise level. All the potential noise sources are included in the graph. The expression of the variance $\sigma_{b_i}^2$ and the mean m_{b_i} of each noise source b_i according to $\mathbf{w_{FP}}$ are generated in order to be included in the output C function. The expression of the number of bits eliminated between two operations can be found in [25]. Then, each operation is replaced by its propagation noise model, as presented in Section III-A2. The aim of the second step is to determine the expression of the system impulse response between the output and the different noise sources. To handle recursive systems, cycles in the graph are detected and then dismantled. The third step computes the value of the gains K_i and L_{ij} given in (31) and (32). The linear prediction approach presented in Section III-C2 is used to approximate the infinite sums. The values of the different terms of the matrices $\mathbf{A}$ and $\mathbf{D}$ are collected during a single floating-point simulation. The reference simulation and the third step of the tool are performed with the Matlab tool.

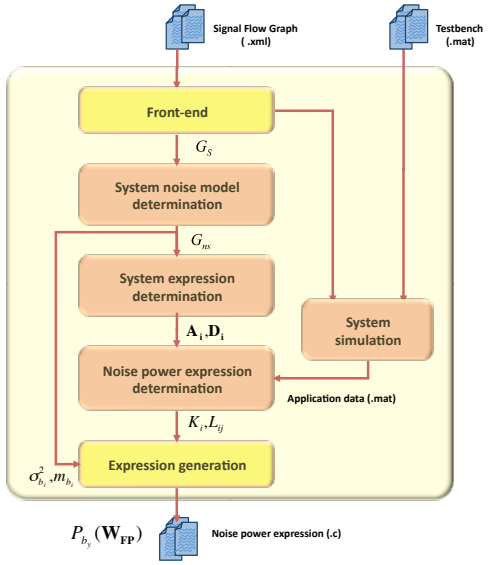

Fig. 2. Synoptic of the Accuracy Evaluation Tool

IV. Results

In this section, experimentations are carried out to validate the proposed approach on LTI and non-LTI systems. Results are given on various applications: vector normalization illustrates non-LTI non-recursive systems, adaptive filters such as LMS, NLMS and APA illustrate non-LTI recursive systems, and various benchmarks illustrate all kinds of systems. For these applications the Signal-to-Quantization Noise Ratio (SQNR) values are chosen from 30 dB to 90 dB to ensure that the assumptions of the Widrow model are satisfied (the noise variance is significantly smaller than the signal power). Moreover, the optimization time of the proposed approach is compared with approaches based on fixed-point simulations.

A. A non-LTI non-recursive system: Vector Normalization

The first example considers an N-size vector $\mathbf{x}(n)$ normalized according to the relation

$$\mathbf{y}(n) = \frac{\mathbf{x}(n)}{\mathbf{x}^t(n)\mathbf{x}(n)}. \tag{51}$$

Here to simplify the final expression of the power, only two noises are considered: the noise $\mathbf{b_x(n)}$ due to the quantization of $\mathbf{x}(n)$ and the noise $\mathbf{b_g}(n)$ generated by the normalization computation. By using the noise model propagation defined in Section III-A, the output noise vector $\mathbf{b_y(n)}$ is equal to

$$\mathbf{b_y}(n) = \frac{\mathbf{b_x}(n)}{\mathbf{x}^t(n)\mathbf{x}(n)} + 2\frac{\mathbf{x}(n)\mathbf{x}^t(n)\mathbf{b_x}(n)}{\mathbf{x}^t(n)\mathbf{x}(n)^2} + \mathbf{b_g}(n), \tag{52}$$

which can be simplified as

$$\mathbf{b_y}(n) = \underbrace{\left(\frac{\mathbf{I_N}}{\mathbf{x}^t(n)\mathbf{x}(n)} + 2\frac{\mathbf{x}(n)\mathbf{x}^t(n)}{\mathbf{x}^t(n)\mathbf{x}(n)^2}\right)}_{\mathbf{A_x}(n)} \mathbf{b_x}(n) + \mathbf{b_g}(n), \tag{53}$$

with $\mathbf{I_N}$ the N-size identity matrix. This example has no delay in the graph so only one impulse term $\mathbf{A_x}(n)$ has to be determined. The other terms $\mathbf{A_x}(k)$ are then equal to zero for $k \neq n$ and $d_x(n)$ is equal to one.

By applying (29) to the output noise expression, its power P_b is

$$\begin{aligned} P_b &= E[Tr(\mathbf{A_x}^t(n)\mathbf{A_x}(n))]\sigma_x^2 + N\sigma_g^2 + m_x^2 E[Tr(\mathbf{A_x}^t(n)\mathbf{A_x}(n))] \\ &+ Nm_g^2 + 2m_g m_x E[Tr(\mathbf{A_x}(n))] \end{aligned} \tag{54}$$

Gains	Expression	Correlation coefficient c 0.1	0.9
K_1	$E[Tr(\mathbf{A_x}^t(n)\mathbf{A_x}(n))]$	21622.45	11723.46
K_2	$Tr(\mathbf{I_4}\mathbf{I_4})$	4	4
L_{11}	$E[Tr(\mathbf{A_x}^t(n)\mathbf{A_x}(n))]$	21622.45	11723.46
L_{12}	$E[Tr(\mathbf{A_x}(n)]$	287.78	137.44
L_{21}	$E[Tr(\mathbf{A_x}(n)]$	287.78	137.44
L_{22}	$Tr(\mathbf{I_4}\mathbf{I_4})$	4	4

TABLE IV
EXPRESSION AND VALUES OF THE GAINS K AND L IN THE CASE OF THE VECTOR NORMALIZATION EXAMPLE

where m_x and m_g are the means of $\mathbf{b_x}(n)$ and $\mathbf{b_g}(n)$ and σ^2 their variances.

For this example, N is set to 4, and the truncation mode is considered. The fractional part word-length of the input $\mathbf{x}$ and for the output $\mathbf{y}$ are respectively w_{in} and w_{out}. So, using (30) and by introducing values of mean and variance of the two considered noises, (54) can be written as :

$$\begin{aligned} P_b &= K_1\frac{2^{-2w_{in}}}{12} + K_2\frac{2^{-2w_{out}}}{12}(1-2^{-2(k_{out})}) \\ &+ L_{11}\frac{2^{-2w_{in}}}{4} + L_{22}\frac{2^{-2w_{out}}}{4}(1-2^{-(k_{out})})^2 \\ &+ L_{12}\frac{2^{-w_{out}}}{2}(1-2^{-(k_{out})})\frac{2^{-w_{in}}}{2} \\ &+ L_{21}\frac{2^{-w_{out}}}{2}(1-2^{-(k_{out})})\frac{2^{-w_{in}}}{2} \end{aligned} \quad (55)$$

where k_{out} represents the number of bits eliminated at the output of the normalization. The expression of the gain K and L are given in Table IV.

The chosen synthetic input signal $x(n)$ is a first-order autoregressive signal defined by

$$x(n+1) = c.x(n) + u(n), \quad (56)$$

with $c \in [0,1]$ and $u(n)$ a zero-mean white noise of variance 0.01.

The numerical values of the terms K and L are given in Table IV. As shown by the results, the terms K and L depend on signal characteristics and correlation between them.

B. A non-LTI recursive system: the Least Mean Square (LMS) adaptive filter

1) Fixed-Point LMS Algorithm: To illustrate the proposed model and to verify its accuracy on a non-LTI and recursive system, the Least Mean Square (LMS) example is detailed. The LMS adaptive algorithm [18] addresses the problem of estimating a sequence of scalars $y(n)$ from an N-length vector $\mathbf{x}(n) = [x(n), x(n-1)\dots x(n-N+1)]^t$. The linear estimate of $y(n)$ is $\mathbf{w}^t(n)\mathbf{x}(n)$ where $\mathbf{w}(n)$ is an N-length vector which converges to the optimal vector $\mathbf{w}_{opt}$ in the mean-square error (MSE) sense. This optimal vector is equal to $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{v}$ where $\mathbf{R}$ corresponds to the correlation matrix of the input signal equal to $E[\mathbf{x}(n)\mathbf{x}^t(n)]$ and where $\mathbf{v}$ represents the intercorrelation vector between $\mathbf{x}(n)$ and $y(n)$ equal to $E[\mathbf{x}(n)y(n)]$. The vector $\mathbf{w}(n)$ is updated according to

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_{LMS}.\mathbf{x}(n)(y(n) - \mathbf{w}^t(n)\mathbf{x}(n)) \quad (57)$$

where μ_{LMS} is a positive constant representing the adaptation step. In a fixed-point implementation of the LMS, four equivalent noise sources have to be considered as presented in Figure 3. $\mathbf{b_x}(n)$ and $b_r(n)$ are noises associated respectively with the input data $\mathbf{x}(n)$ quantization and the desired signal $y(n)$ quantization. The vector $\mathbf{b_w}(n)$ is the noise vector due to the quantization of the products between μ, $\mathbf{x}(n)$ and the error $e(n) = y(n) - \mathbf{w}^t(n)\mathbf{x}(n)$. Finally, $b_f(n)$ groups together all noises generated inside the filtering part during the computation of the inner product $\mathbf{w}^t(n)\mathbf{x}(n)$. The terms μ_i and σ_i^2 represent the mean and variance of each noise source b_i.

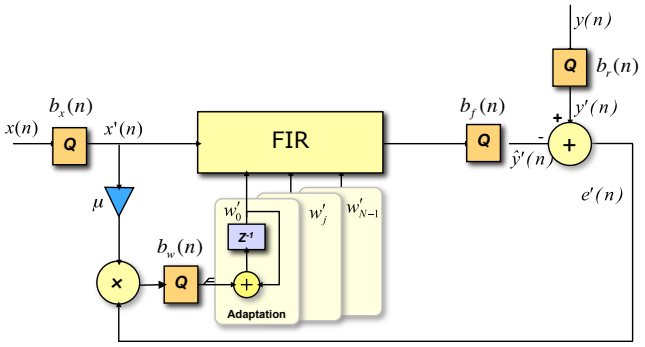

Fig. 3. Fixed-point LMS algorithm and noise model for input data $x'(n)$, estimate data $\widehat{y}'(n)$ and coefficients $\mathbf{w}'(n)$

2) Noise Model for Adaptive Coefficient Computation: In this section, the output noise power expression is determined from the four noise sources: $\mathbf{b_x}(n)$, $b_f(n)$, $\mathbf{b_w}(n)$ and $b_r(n)$. For each one, the impulse response between the source and the output is determined. Then, the output noise power can be directly expressed using (30) where $N_e = 4$ in that case. Also, the output noise power can be determined using the linear prediction approach (as presented in Section III-C2). Each impulse response is modelled using the prediction coefficients, leading to a simpler expression of the output noise power.

These two approaches are presented on the LMS example in the following. Only the propagation of the noise $\mathbf{b_w}(n)$ is detailed. The contribution of the other noises can be obtained in the same way. The propagation of noise term $\mathbf{b_w}(n)$ is shown on Figure 4 and is considered for validating the accuracy of the proposed approach and detailed hereafter. As shown on Figure 3, the noise $\mathbf{b_w}(n)$ is first delayed with a first recursion. Then, the noise goes through the FIR filter where it is multiplied by the signal $\mathbf{x}^t(n)$. After, the noise propagates through the second recursion to the system output and comes back to the multiplication by $\mu_{LMS}.\mathbf{x}(n)$.

Noise propagation can be modelled using the following set of recurrent equations

$$\begin{aligned} b'_w(n) &= \mathbf{x}^t(n)\mathbf{b1_w}(n-1) & (58)\\ \mathbf{b1_w}(n) &= \mathbf{b1_w}(n-1) + \mathbf{b_w}(n) - \mu\mathbf{x}(n)b'_w(n) & (59) \end{aligned}$$

where $\mathbf{b1_w}(n)$ is the noise after the delay. By developing these two previous expressions, the scalar output noise $b'_w(n)$ is written using its time-varying impulse response h_w as

$$b'_w(n) = \sum_{k=0}^{n-1} h_w(k)\mathbf{b_w}(k). \quad (60)$$

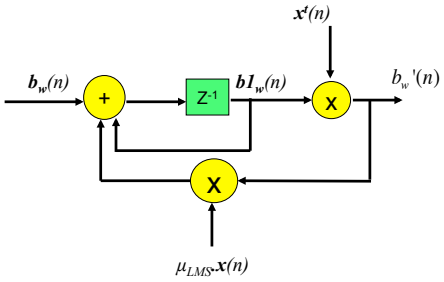

Fig. 4. Propagation through the LMS algorithm of the noise source $\mathbf{b_w}(n)$ due to filter coefficient computing

The time-varying impulse response h_w is defined through two equivalent matrices as defined in Section III. For this noise, the matrix $\mathbf{A_w}$ corresponds to an N-size vector $\mathbf{a_w}$. This vector is introduced in (61) and is defined in (62).

$$b'_w(n) = \sum_{k=0}^{n-1} \mathbf{a_w}(k)\mathbf{b_w}(k) \tag{61}$$

$$\mathbf{a_w}(k) = \mathbf{x}^t(n)\mathbf{F}(n,k) \tag{62}$$

with

$$\mathbf{F}(n,k) = \prod_{m=k+1}^{n-1} \Big(I_N - \mu_{LMS}\mathbf{x}(m)\mathbf{x}^t(m)\Big). \tag{63}$$

The expression of $\mathbf{a_w}$ can be modelled using linear prediction coefficient as presented in Section III-C2. The first impulse response terms of $\mathbf{a_w}(k)$ are

$$\begin{aligned} \mathbf{a_w}(n) &= 0, \\ \mathbf{a_w}(n-1) &= \mathbf{x}^t(n), \\ \mathbf{a_w}(n-2) &= \mathbf{x}^t(n)\big(I_N - \mu_{LMS}\mathbf{x}(n-1)\mathbf{x}^t(n-1)\big). \end{aligned} \tag{64}$$

The LMS algorithm is a first-order recursive system. Thus, only one prediction coefficient has to be determined using the approach defined before. The matrix U_w, defined by (40), is a scalar in that case and is equal to

$$U_w = E(\mathbf{a_w}(n-1)\mathbf{a_w}^t(n-1)) = E\big(\mathbf{x}^t(n)\mathbf{x}(n)\big). \tag{65}$$

The term J_w, defined by (41), is equal to

$$\begin{aligned} J_w &= E(\mathbf{a_w}(n-2)\mathbf{a_w}^t(n-1)) \\ &= E\Big(\mathbf{x}^t(n)\big(I_N - \mu_{LMS}\mathbf{x}(n-1)\mathbf{x}^t(n-1)\big)\mathbf{x}(n)\Big). \end{aligned} \tag{66}$$

So, from (39), the prediction coefficient λ_w is expressed as

$$\begin{aligned} \lambda_w &= \frac{J_w}{U_w} = \frac{E[\mathbf{a_w}(n-2)\mathbf{a_w}^t(n-1)]}{E[\mathbf{a_w}(n-1)\mathbf{a_w}^t(n-1)]} \\ &= \frac{E[\mathbf{x}^t(n)(I_N - \mathbf{x}(n-1)\mathbf{x}^t(n-1))\mathbf{x}(n)]}{E[\mathbf{x}^t(n)\mathbf{x}(n)]}. \end{aligned} \tag{67}$$

Thus, with this coefficient, the impulse response term $\mathbf{a_w}$ can be expressed as

$$\mathbf{a_w}(k) = \lambda_w \mathbf{a_w}(k+1), \tag{68}$$

for $k < n-1$ since $\mathbf{a_w}(n) = 0$. The infinite sum is obtained using (42) as

$$\sum_{k=0}^{n\to\infty} \mathbf{a_w}(k) = \frac{1}{1-\lambda_w}\mathbf{a_w}(n-1). \tag{69}$$

The contributions of the three other terms can be obtained with the same method. The time-varying impulse response associated with these three noises are given in the following expressions.

$$\begin{aligned} \mathbf{a_x}(k) &= \mu_{LMS}\mathbf{x}^t(n)\mathbf{F}(n,k)\Big(e(k) - \mathbf{x}(k)\mathbf{w}^t(k)\Big) \\ &+ \mathbf{w}^t(n)\delta(n-k) \\ a_r(k) &= \mu_{LMS}\mathbf{x}^t(n)\mathbf{F}(n,k)\mathbf{x}(k) \\ a_f(k) &= -\mu_{LMS}\mathbf{x}^t(n)\mathbf{F}(n,k)\mathbf{x}(k) + \delta(n-k) \end{aligned} \tag{70}$$

where $\delta(k)$ is the Kronecker operator. It is noteworthy that, as a consequence, the noises $b_f(n)$ and $b_r(n)$ are scalar. Then, the terms A modelling their propagation through the system are also scalars, which let us write $Tr(AA^t) = A^2$ for input noises $b_f(n)$ and $b_r(n)$. The output noise power is finally computed using (30) and leads to (71) given at page 10.

The noise mean term M is equal to

$$M(k) = \mathbf{a_x}(k)\mathbf{m_x} + a_r(k)m_r + a_f(k)m_f + \mathbf{a_w}(k)\mathbf{m_w}$$

where the mean values $\mathbf{m_x}$ and $\mathbf{m_w}$ are N-size vectors and m_r and m_r are scalars. The complexity of the computation of the gains K_i and L_{ij} of (71) are obtained from (33) and (34). In this example, input noises are N size vectors for $\mathbf{b_w}$ and $\mathbf{b_x}$ (so $M_i = N$ and $N_i = 1$) and scalar for b_f and b_r ($M_i = N_i = 1$), the output noise b_y is a scalar ($M_y = N_y = 1$). Moreover, the number p is set to 500 and $N_t = 100$. The number of multiplications N_{mult} is equal to

$$N_{mult} = 50000(6N + 8) \tag{72}$$

So the complexity is in $\mathcal{O}(N)$. The expression of the output noise power can be modelled using prediction coefficients that can be computed for noises $\mathbf{b_x}(n)$, $b_f(n)$ and $b_r(n)$ using the same method as the one presented before for $\mathbf{b_w}(n)$. Using the prediction coefficients, the expression (71) is simplified and leads to (73) given at page 10 in which σ^2 refers to scalar variance value of each noise.

By applying (49), the number of multiplications is then equal to

$$N_{mult} = 100(8N + 16) \tag{75}$$

The complexity is divided by a factor 375. Thus, the output noise power can be computed by two different ways: using the impulse response terms of ((71) derived from (30)) or using prediction coefficients using (73) presented in Section III-C2. The quality of these two approaches is compared in the next section.

3) Quality of the Estimation: To analyze the quality of the proposed approach, experiments have been performed. This quality has been evaluated through the measurement of the relative error Er between our analytical estimation P_b and the estimation $P_b^{(sim)}$ based on fixed-point simulations which is considered to be the reference. The relative error Er is defined as

$$E_r = \left| \frac{P_b - P_b^{(sim)}}{P_b^{(sim)}} \right|. \tag{76}$$

The reference value is obtained by simulations from the difference between a floating-point and a fixed-point simulation. The floating-point simulation is used as the high-precision reference, since floating-point arithmetic generates negligible noises compared to fixed-point arithmetic.

For these experiments, the chosen synthetic input signal $x(n)$ is a first-order autoregressive signal defined by (56). Figure 5 shows the relative error of the proposed approach on a 32-tap LMS adaptive filter. The results are presented according to the number of elements p chosen to compute the infinite sums and the correlation coefficient

$$E[b_y^2(n)] = \sum_{k=0}^{n} \sigma_x^2 E\Big[Tr\big(\mathbf{a_x}(k)\mathbf{a_x}^t(k)\big)\Big] + \sum_{k=0}^{n} \sigma_f^2 E[a_f^2(k)] + \sum_{k=0}^{n} \sigma_r^2 E[a_r^2(k)] + \sum_{k=0}^{n} \sigma_w^2 E\Big[Tr\big(\mathbf{a_w}(k)\mathbf{a_w}^t(k)\big)\Big] + \sum_{k=0}^{n}\sum_{l=0}^{n} E\Big[Tr\big(M(k)M^t(l)\big)\Big] \quad (71)$$

$$E[b_y^2(n)] = \frac{\sigma_x^2 E\big[Tr(\mathbf{a_x}(n)\mathbf{a_x}^t(n))\big]}{1-\lambda_x^2} + \frac{\sigma_f^2 E[a_f^2(n)]}{1-\lambda_f^2} + \frac{\sigma_{b_r}^2 E[a_{b_r}^2(n)]}{1-\lambda_r} \frac{\sigma_w^2 E\big[Tr(\mathbf{a_w}(n)\mathbf{a_w}^t(n))\big]}{1-\lambda_w^2} + Tr\big(TT^t\big) \quad (73)$$

with

$$T = \mathbf{m_x}\frac{E[\mathbf{a_x}(n)]}{1-\lambda_x} + m_r\frac{E[a_r(n)]}{1-\lambda_r} + m_f\frac{E[a_f(n)]}{1-\lambda_f} + \mathbf{m_w}\frac{E[\mathbf{a_w}(n)]}{1-\lambda_w}. \quad (74)$$

c of the input signal $x(n)$. The input signal can be a white noise ($c = 0$), a fairly correlated noise ($c = 0.5$) or a very correlated noise ($c = 0.95$). As p increases, the relative error decreases. Indeed, when p increases, more terms are included in the computation of the sums, which leads to a better estimation quality. Moreover, the relative error convergence speed depends on input data correlation. For non-correlated input data, the relative error convergence is slower than the one for very-correlated input data. In fact, the relative error is less than 20% after 300 samples for very correlated input data, after 350 samples for fairly correlated data and after 550 samples for uncorrelated input data.

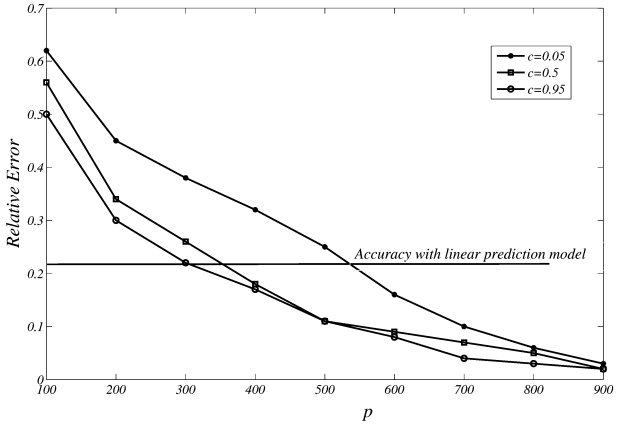

Fig. 5. Relative error for the 32-tap LMS according to the number of samples p in the computation of the infinite sums

To explain this result, we recall that the terms $\mathbf{A}$, representing the noise propagation through the system, include the terms $\mathbf{F}(n,k)$ defined as

$$\mathbf{F}(n,k) = \prod_{m=k+1}^{n-1} \Big(I_N - \mu_{LMS}\mathbf{x}(m)\mathbf{x}^t(m)\Big). \quad (77)$$

For very correlated input data, the term $I_N - \mu_{LMS}\mathbf{x}(m)\mathbf{x}^t(m)$ decreases faster than for uncorrelated data. Thus, the number of elements p required to compute the infinite sums depends on the input data correlation. Nevertheless, with experimentation presented for $p > 500$, the relative error is less than 25% in all cases, which represents a difference equivalent to less than 1 bit between the noise power obtained with the proposed approach and the real noise power. For the linear prediction approach, the obtained relative error is equal to 21%. For other sizes N of the LMS algorithm, same kinds of results are obtained, thus indicating that the proposed approach is accurate for the LMS algorithm.

Benchmark	Infinite sums with $p = 500$	Infinite sums with $p = 1000$	Linear prediction	N_e, number of noise sources
LMS32	14.4%	3.76%	22.6%	66
NLMS	14.24%	4.24	20.68%	67
APA	17.65%	6.54	25.46%	812

TABLE V
RELATIVE ERROR OBTAINED WITH THE PROPOSED APPROACHES FOR DIFFERENT ADAPTIVE FILTERS

Experiments have also been conducted on other adaptive algorithms. The results are presented in Table V for the previous LMS, a 32-tap Normalized LMS (NLMS) and an Affine Projection Algorithm (APA) of size 32×12. The expression of the coefficient update for the APA algorithm is given in (16). The APA application is suitable to show the case where the general model presented in Section III include the two matrices $\mathbf{A}$ and $\mathbf{D}$. Both NLMS and APA algorithms moreover include division operations for the normalization module. The floating-point simulation is executed on $N_t + p$ samples where N_t is the number of samples to compute the statistical mean and p is the number of terms to compute the sums. In these experiments, N_t is set to 100 and different values of p are tested.

The approach based on linear prediction leads to a relative error of around 20% (equal to 1 dB). However, the method using the computation of the sums can be really accurate since the relative error is less than 17.65% in mean with $p = 100$ and less than 6.24% with $p = 1000$. The accuracy of the estimation can be controlled by adjusting the number of elements p used to compute these sums. We can therefore see the term p as a way to trade-off between the accuracy of the estimation and the time required to compute this estimation.

C. Experiments on LTI and Non-LTI Non-Recursive Systems

In this section, the proposed approach is first evaluated on LTI systems such as FIR and IIR filters, an MP3 coder/decoder and an FFT. Then, the approach is applied to non-LTI and non-recursive systems (square, Volterra filter, correlator). The average and maximum relative errors between the noise power estimated with the proposed approach and the noise power obtained by fixed-point simulations are provided.

Quantization	**Average relative error**	**Maximum relative error**
FIR32(R)	2.49%	7.9%
FIR32(T)	1.14%	2.41%
Polyphase filter	5.24%	20.57%
DCT	8.42%	24.8%
FFT128	3.68%	7.32%
FFT256	4.53%	7.89%
IIR8	1.68%	3.3%

TABLE VI
RELATIVE ERROR FOR LTI SYSTEMS

1) LTI Systems: First, non-recursive systems, such as FIR filter, MP3 coder/decoder and FFT, are evaluated for different fixed-point specifications. The results are presented in Table VI. For the 32-tap FIR filter, rounding and truncation have been tested. The average relative error is 2.49% for rounding and 1.14% for truncating. Moreover, the maximum relative errors are very low (about 8%).

For the MP3 coder/decoder, the average error is equal to 5.24% for the polyphase filter and to 8.42% for the DCT. The maximum errors are 20.57% for the polyphase filter and 24.8% for the DCT. These values represent a difference of $2dB$ (less than 1 bit) between the real value of the noise power and the one estimated by the proposed approach. For the FFT the maximal relative error is less than 7.89%, which also illustrates the accuracy of the proposed approach on relevant applications such as the FFT. To test the method on a recursive LTI system, we have considered an 8-order IIR filter divided into four 2-order cells. The relative error is less than 3.3% with an average value of 1.68%.

2) Non-LTI and Non-Recursive systems: For the case of non-LTI systems we have tested three applications: a 2^{nd}-order polynomial evaluation, a 2^{nd}-order Volterra filter and a *correlator*. The results are presented in Table VII. For the 2^{nd}-order polynomial, the estimation is highly accurate, since the error is lower than 0.8%. For the Volterra filter and the correlator, the average relative error is equal to 1.79% and 1.35% with the maximum values of 3.22% and 5.78% respectively. For these three applications, the error is less than 5.78% , thus confirming the accuracy of the proposed approach.

In summary, for all types of systems based on arithmetic operations, the presented results tend to indicate that the method has an overall relative error that is less than 25% even in the worst cases, which can be translated to less than 1 bit of precision.

D. Fixed-Point Specification Optimization Time

The evaluation of the numerical accuracy is used in the word-length optimization based on an iterative process. The optimization time obtained with the proposed approach is compared with those obtained with fixed-point simulations. For the two approaches,

Application	**Average relative error**	**Maximum relative error**
Polynomial eval.	0.65%	0.8%
Volterra filter	1.79%	3.22%
Correlator	1.35%	5.78%

TABLE VII
RELATIVE ERROR FOR NON-LTI AND NON-RECURSIVE SYSTEMS

the optimization times are given in Figure 6 for the 32-tap LMS algorithm. A computer with Intel Core 2 Duo T8300 at 2.4 GHz and 2GB of RAM was used for the experiments. For the approach based on fixed-point simulation, a new simulation is required as long as a fixed-point format is modified. So, the optimization time depends linearly on the number of optimization iterations. For approaches based on fixed-point simulations, experimentations were conducted on Matlab to evaluate their computing time. For the proposed approach, the analytical expression of P_b needs to be computed first. The determination of this analytical expression is the most time-consuming part and its computation takes 46 seconds for the approach based on infinite sum approximation and 4 seconds for the linear prediction approach. Then, each iteration of this optimization process corresponds only to the noise power expression evaluation, whose computing time is negligible. For the LMS algorithm, the proposed approach leads to a gain after only 100 iterations, which represents an execution time equal to 46 seconds. As an example, for an optimization process with about 30 variables, between 10^3 and 10^5 iterations are required. In that case the proposed approach would require between 10 and 10^3 less time to perform the optimization process. The approach based on linear prediction leads to time gain after only 10 iterations compared to approaches based on fixed-point simulations, thus leading to an execution time equal to 4 seconds. These results clearly show that the proposed approach is faster than the best simulation-based techniques for the process of fixed-point optimization.

A comparison with methods from [4] and [31] has also been carried-out. As they are also analytical, only the time to determine the coefficients K_i and L_{ij} is given. For the approach developed in [31], no result on the quality and the execution time were given. However, for an application with N_e noise sources, N_e^2 fixed-point simulations are required. Thus, the execution time grows quadratically with the complexity of the application. For the 32-tap LMS algorithm presented above, 66 noise sources are considered and thus the approaches proposed in [31] would require 4356 fixed-point simulations. Therefore, the proposed method leads to significantly lower optimization time, especially when the application becomes complex.

In the approach based on affine arithmetic simulations proposed in [4], the relative error E_r for the estimation and the time T_{exe} required to determine the coefficients K_i and L_{ij} are given in Table VIII for a second-order IIR filter (IIR2), an 8-order IDCT, and for an LMS filter with 2 and 5 taps. For the case of the IIR2 application, the time to execute the third step corresponding to the determination of the noise power expression (computation of K_i and L_{ij}) corresponds to 70 % of the global execution time for the approach based on infinite sums and to 50 % for the approach based on

Bench-marks	Infinite sums		Linear prediction		Approach from [4]	
	T_{exe}	E_r	T_{exe}	E_r	T_{exe}	E_r
IIR 2	0.15	1.68%	0.08	0.54%	0.88	0.74%
IDCT 8	0.04	0.62%	0.04	0.62%	0.0031	0.88%
LMS 2	0.0587	1.09%	0.0485	4.87%	592	0.92%
LMS 5	0.2142	2.87%	0.1256	10.67%	1646	1.08%

TABLE VIII
COMPARISON OF THE PROPOSED APPROACHES WITH THE ONE PROPOSED IN [4] IN TERMS OF EXECUTION TIME T_{exe} (SECONDS) AND RELATIVE ERROR E_r FOR THE ESTIMATION

linear-prediction. The time to execute the second step corresponding to the determination of system expression (computation of A_i and D_i) represents to 26 % of the global execution time for the approach based on infinite sums. For this approach, the rest of the time (4 %) is dedicated to the other steps (front-end, determination of system noise model and expression generation). For the proposed method based on infinite sums with $p = 1000$, the quality of the estimation is close to the one obtained with the technique of [4]. For our linear prediction based-approach, the quality is slightly lower, but the execution time is reduced. For our two approaches, the execution time is significantly reduced compared to the approach proposed in [4] and a reduction of several order of magnitude is observed.

The results show the benefit of the proposed approaches to reduce the development time of fixed-point systems. We propose two different approaches with different characteristics, as shown in Table VIII. It is seen that the infinite sums approach is really accurate and fast, whereas, the approach based on linear prediction is faster but a little less accurate. So, the proposed framework allows the user to choose between two different approaches with different trade-offs between speed and accuracy.

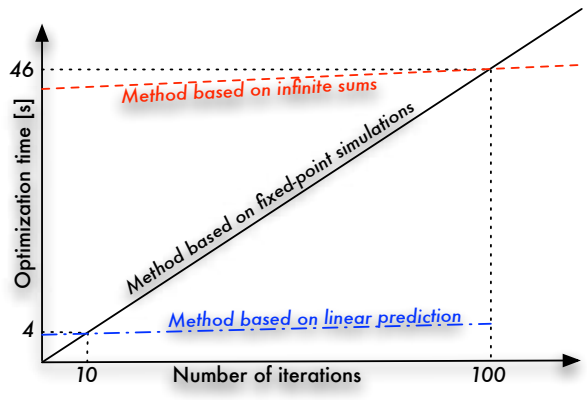

Fig. 6. Optimization time (s) for the LMS32 example of our two approaches (linear prediction, infinite sums with $p = 500$) and the approach based on fixed-point simulations

V. CONCLUSION

In this paper, a fast general and accurate approach to determine the accuracy of a fixed-point system is presented and validated. The method is analytic and provides a closed-form expression of the output noise power for systems composed of smooth operations. The quantization noises due to fixed-point computation are considered through their statistical properties (mean, variance, etc.) and their propagation through arithmetic operations (addition, subtraction, multiplication, and division) is detailed. The system is characterized by its time-varying impulse response and the expression of the output noise power is obtained. The approach is valid for all types of system made-up of smooth operations and for most common rounding modes (truncation, conventional rounding, convergent rounding). The estimation is unbiased and leads to infinite sums inside the expression of the output noise power which have to be approximated by a finite number of terms. To reduce the complexity due to infinite sums, an approach based on linear prediction is also proposed.

The proposed approach was applied to the LMS algorithm, a recursive and non-LTI system, to verify its validity and its quality. Moreover, several LTI and non-LTI systems have been tested to show the accuracy of the proposed approach on various types of applications. The maximal relative errors are less than 25% which corresponds to less than 1 bit. Finally, the approach has been compared to approaches based on fixed-point simulations in terms of execution time. After only 10 iterations, the proposed approach is less time-consuming than other approaches based on fixed-point simulations. Thus, it was shown that the proposed approach leads to accurate estimation and allows a significant reduction of the optimization time in the fixed-point conversion process.

APPENDIX A

In this section, infinite sums are modelled for impulse response terms $h(k)$. The vector $\boldsymbol{\omega}_m$ is defined as

$$\boldsymbol{\omega}_m = \begin{pmatrix} h(m) \\ h(m+1) \\ \vdots \\ h(m+P-1) \end{pmatrix}. \tag{78}$$

The first $P + Q - 1$ terms of impulse response are directly determined. Thus, terms $h(0)$ to $h(P + Q - 2)$ are computed with (22). The term $P + Q - 1$ is present in the matrix $\boldsymbol{\Omega}_Q$ and in the following $P - 1$ matrices. This term can also be computed with prediction coefficient λ_i. The P-size matrix $\boldsymbol{\Lambda}$ is defined by

$$\boldsymbol{\Lambda} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & \dots & 0 & 1 \\ \lambda_P & \lambda_{P-1} & \dots & \dots & \lambda_1 \end{pmatrix} \tag{79}$$

and the following recurrence relation is obtained:

$$\boldsymbol{\omega}_{m+1} = \boldsymbol{\Lambda}\boldsymbol{\omega}_m = \boldsymbol{\Lambda}^{m+2-Q}\boldsymbol{\omega}_{Q-1}. \tag{80}$$

Theorem 2: **The matrix $\boldsymbol{\Lambda}^n$ has a limit equal to 0 when $n \to \infty$.**

Proof:

The characteristic polynomial $\psi_P(X) = det(X\mathbf{I_P} - \boldsymbol{\Lambda})$ of the matrix $\boldsymbol{\Lambda}$ is equal to

$$\psi_P(X) = \begin{vmatrix} X & -1 & 0 & \dots & 0 \\ 0 & X & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \dots & \\ 0 & \dots & \dots & X & -1 \\ -\lambda_P & -\lambda_{P-1} & \dots & \dots & X-\lambda_1 \end{vmatrix}. \tag{81}$$

To compute the determinant, we develop according to the first column, which leads to the recurrence expression

$$\begin{aligned}\psi_P(X) &= X\psi_{P-1}(X) - \lambda_P(-1^{P-1})(-1^{P-1}) \\ &= X\psi_{P-1}(X) - \lambda_P\end{aligned}$$

As for $P = 1$, $\mathbf{\Lambda} = \lambda_1$, $\psi_1(X) = X - \lambda_1$ and by using the previous recursion, $\psi_P(X)$ is equal to

$$\psi_P(X) = X^P - \sum_{k=1}^{P} \lambda_k X^{P-k}. \tag{82}$$

The roots of the previous equation are the eigenvalues of $\mathbf{\Lambda}$. Moreover, these prediction coefficients define a transfer function. Indeed, they linearize the time-varying transfer function. The poles ρ_m of the transfer function satisfy the equality

$$1 - \sum_{k=1}^{P} \lambda_k = \prod_{m=1}^{P} (1 - \rho_m). \tag{83}$$

Thus, comparing solutions of (82) with (83), eigenvalues of $\mathbf{\Lambda}$ are the poles of the transfer function, and $\mathbf{\Lambda}$ can be written as

$$\mathbf{\Lambda} = \mathbf{T} \begin{pmatrix} \rho_1 & 0 & 0 & \dots & \\ 0 & \rho_2 & 0 & \dots & \\ \vdots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & \rho_{P-1} & 0 & \\ 0 & \dots & \dots & \rho_P & \end{pmatrix} \mathbf{T}^{-1} \tag{84}$$

where $\mathbf{T}$ is an invertible matrix and ρ_m can be real or complex. However, since the system is stable, the modulus of ρ_m is less than one so ρ_m^n converges to 0 when $n \to \infty$. So, $\mathbf{\Lambda}^n$ is equal to

$$\mathbf{\Lambda}^n = \mathbf{T} \begin{pmatrix} \rho_1^n & 0 & 0 & \dots & \\ 0 & \rho_2^n & 0 & \dots & \\ \vdots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & \rho_{P-1}^n & 0 & \\ 0 & \dots & \dots & \rho_P^n & \end{pmatrix} \mathbf{T}^{-1}. \tag{85}$$

(85) proves that $\mathbf{\Lambda}^n$ tends to zero when $n \to \infty$. ∎

Therefore:

$$\sum_{m=0}^{\infty} \boldsymbol{\omega}_m = \sum_{m=0}^{Q-2} \boldsymbol{\omega}_m + \sum_{k=0}^{\infty} \mathbf{\Lambda}^k \boldsymbol{\omega}_{Q-1} = \sum_{m=0}^{Q-2} \boldsymbol{\omega_m} + (\mathbf{I_P} - \mathbf{\Lambda})^{-1} \boldsymbol{\omega}_{Q-1} \tag{86}$$

with $\mathbf{I_P}$ the P-size identity matrix.

The vector $\sum_{m=0}^{\infty} \omega_m$ is also equal to

$$\sum_{m=0}^{\infty} \boldsymbol{\omega}_m = \begin{pmatrix} \sum_{m=0}^{\infty} h(m) \\ \sum_{m=1}^{\infty} h(m) \\ \vdots \\ \sum_{m=P-1}^{\infty} h(m) \end{pmatrix}. \tag{87}$$

So, the vector $\boldsymbol{\theta}$ is introduced where each element $\boldsymbol{\theta}_m$ is defined by

$$\boldsymbol{\theta}_m = \sum_{k=0}^{m-2} h(k) \tag{88}$$

$\boldsymbol{\theta}$ contains the $P-1$ first impulse response terms which are computed manually. Thus, each line of $\sum_{m=0}^{\infty} \boldsymbol{\omega}_m + boldsymbol\theta$ is equal to $\sum_{m=0}^{\infty} h(m)$. The sum of impulse response terms is then equal to

$$\sum_{m=0}^{\infty} h(m) = \frac{1}{P} \sum_{j=1}^{P} \Big(\sum_{m=0}^{Q-2} \boldsymbol{\omega}_m + (\mathbf{I_P} - \mathbf{\Lambda})^{-1} \boldsymbol{\omega}_{Q-1} + \boldsymbol{\theta} \Big)_{(j)}. \tag{89}$$

This expression is developed for impulse terms h. In practice, the impulse terms are defined by two matrices $\mathbf{A}$ and $\mathbf{D}$ as shown in (Section III). The previous method is then applied to $\mathbf{A1D}$ where $\mathbf{1}$ is a matrix containing ones (and replacing the noise $\mathbf{B}$).

APPENDIX B

In this appendix, the method modelling the second order infinite sums is presented. Using the same approach as in Appendix A, the following recurrence relation is obtained:

$$\boldsymbol{\omega}_{m+1}\boldsymbol{\omega}_{m+1}^t = \mathbf{\Lambda}\boldsymbol{\omega}_m\boldsymbol{\omega}_m^t\mathbf{\Lambda}^t = \mathbf{\Lambda}^{m+2-Q}\boldsymbol{\omega}_{Q-1}\boldsymbol{\omega}_{Q-1}^t{\mathbf{\Lambda}^t}^{m+2-Q}. \tag{90}$$

Matrix $\mathbf{\Lambda}^m$ has a limit equal to zero when $m \to \infty$ as demonstrated previously, which gives

$$\sum_{m=0}^{\infty} \boldsymbol{\omega}_m\boldsymbol{\omega}_m^t = \sum_{m=0}^{Q-2} \boldsymbol{\omega}_m\boldsymbol{\omega}_m^t + \underbrace{\sum_{k=0}^{\infty} \mathbf{\Lambda}^k\boldsymbol{\omega}_{Q-1}\boldsymbol{\omega}_{Q-1}^t{\mathbf{\Lambda}^t}^k}_{\mathbf{S}}. \tag{91}$$

Yet, the next equality can be determined:

$$\mathbf{S} - \mathbf{\Lambda}\mathbf{S}\mathbf{\Lambda}^t = \boldsymbol{\omega}_{Q-1}\boldsymbol{\omega}_{Q-1}^t. \tag{92}$$

This equality is a Lyapunov equation. So, $\mathbf{S}$ is the solution of the Lyapunov equation and can be obtained using a classical solver. As in Appendix A, we introduce the diagonal matrix $\mathbf{\Phi}$ where each diagonal element is equal to

$$\mathbf{\Phi}_{r,r} = \sum_{k=0}^{r-2} h(k)^2 \tag{93}$$

Each element of the diagonal of $\sum_{m=0}^{\infty} \boldsymbol{\omega}_m\boldsymbol{\omega}_m^t + \mathbf{\Phi}$ is equal to the sum of square impulse terms $\sum_{m=0}^{\infty} h(m)^2$. So, it leads to

$$\sum_{m=0}^{\infty} h(m)^2 = \frac{1}{P} Tr\left(\sum_{m=0}^{Q-2} \boldsymbol{\omega}_m\boldsymbol{\omega}_m^t + \mathbf{S} + \mathbf{\Phi} \right). \tag{94}$$

REFERENCES

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[2] P. Belanovic and M. Rupp. Automated Floating-point to Fixed-point Conversion with the fixify Environment. In *IEEE Rapid System Prototyping (RSP'05)*, pages 172–178, Montreal, Canada, 2005.

[3] M. Blair, S. Obenski, and P. Bridickas. Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia. Technical Report GAO/IMTEC-92-26, United States Department of Defense, General Accounting office, 1992.

[4] G. Caffarena, J.A. Lopez C. Carreras, and A. Fernandez. SQNR Estimation of Fixed-Point DSP Algorithms. *EURASIP Journal on Advances in Signal Processing*, 2010.

[5] C. Caraiscos and B. Liu. A Roundoff Error Analysis of the LMS Adaptive Algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(1):34–41, February 1984.

[6] J.M. Chesneaux, L.S. Didier, and F. Rico. The fixed cadna library. *Real Number and Computers (RNC'03)*, pages 215–229, September 2003.

[7] M. Clark, M. Mulligan, D. Jackson, and D. Linebarger. Accelerating Fixed-Point Design for MB-OFDM UWB Systems. *CommsDesign*, January 2005.

[8] G. Constantinides. Perturbation Analysis for Word-length Optimization. In *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'03)*, pages 81–90, Napa, California, April 2003.

[9] G. Constantinides, P. Cheung, and W. Luk. Truncation Noise in Fixed-Point SFGs. *IEEE Electronics Letters*, 35(23):2012–2014, November 1999.

[10] G.A. Constantinides. Word-length optimization for differentiable nonlinear systems. *ACM Transactions on Design Automation of Electronic Systems*, 26(1):26–43, 2006.

[11] George A. Constantinides, Peter Y. K. Cheung, and Wayne Luk. Wordlength optimization for linear digital signal processing. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 22:1432–1442, 2003.

[12] M. Coors, H. Keding, O. Luthje, and H. Meyr. Integer Code Generation For the TI TMS320C62x. In *IEEE International Conference on Acoustics, Speech, and Signal Processing 2001 (ICASSP'01)*, pages 1133–1136, Sate Lake City, US, May 2001.

[13] L.H. de Figueiredo and J. Stolfi. Affine Arithmetic : Concepts and Applications. *Numerical Algorithms*, pages 1–13, 2003.

[14] B. Evans. Modem Design, Implementation, and Testing Using NI's LabVIEW. In *National Instrument Academic Day*, Beirut, Lebanon, June 2005.

[15] J. Eyre and J. Bier. DSPs court the consumer. *IEEE Spectrum*, 36(3):47–53, 1999.

[16] J. Eyre and J. Bier. The evolution of DSP processors. *IEEE Signal Processing Magazine*, 17(2):43–51, March 2000.

[17] P.D. Fiore. Efficient approximate wordlength optimization. *IEEE Transactions on Computers*, 57(11):1561 –1570, 2008.

[18] S. Haykin. *Adaptive Filter Theory*. Prentice Hall Inc, 2^{nd} edition, 1991.

[19] T. Hill. Acceldsp synthesis tool floating-point to fixed-point conversion of matlab algorithms targeting fpgas. White papers, Xilinx, April 2006.

[20] S. Kim and W. Sung. Fixed-Point Error Analysis and Word Length Optimization of 8x8 IDCT Architectures. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(8):935–940, December 1998.

[21] J.L. Lions, L. Lubeck, J.L. Fauquembergue, G. Kahn, W. Kubbat, S. Levedag, L. Mazzini, D. Merle, and C O'Halloran. ARIANE 5, Flight 501 Failure. Technical report, European Space Agency, july 1996.

[22] J.A. Lopez, G. Caffarena, C. Carreras, and O. Nieto-Taladriz. Fast and accurate computation of the roundoff noise of linear time-invariant systems. *IET Circuits, Devices and Systems*, 2(4):393–408, 2008.

[23] J.A. Lopez, C. Carreras, and O. Nieto-Taladriz. Improved Interval-Based Characterisation of Fixed-Point LTI Systems With Feedbacks Loops. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26:1923–1933, 2007.

[24] D. Menard, R. Rocher, P. Scalart, and O. Sentieys. SQNR determination in non-linear and non-recursive fixed-point systems. In *XII European Signal Processing Conference (EUSIPCO'04)*, pages 1349–1352, Vienna, Austria, September 2004.

[25] D. Menard, R. Rocher, and O. Sentieys. Analytical fixed-point accuracy evaluation in linear time-invariant systems. *IEEE Transactions on Circuits and Systems I*, 55 (10):3197–3208, 2008.

[26] D. Menard, R. Rocher, O. Sentieys, and O. Serizel. Accuracy Constraint Determination in Fixed-Point System Design. *EURASIP Journal on Embedded Systems*, 2008:Article ID 23197, 13 pages, 2008.

[27] D. Menard and O. Sentieys. Automatic Evaluation of the Accuracy of Fixed-point Algorithms. In *Design, Automation and Test in Europe 2002 (DATE'02)*, pages 529–535, Paris, March 2002.

[28] Daniel Menard, David Novo, Romuald Rocher, Francky Catthoor, and Olivier Sentieys. Quantization Mode Opportunities in Fixed-Point System Design. In *18th European Signal Processing Conference (EUSIPCO-2010) (2010)*, pages 542–546, Aalborg Danemark, 08 2010. EURASIP.

[29] K. Ozeki and T.Umeda. An adaptative filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electronics and Communications in Japan*, 67:19–27, 1984.

[30] S. Roy and P. Banerjee. An algorithm for trading off quantization error with hardware resources for Matlab-based FPGA design. *IEEE Transactions on Computers*, 54:886–896, 2005.

[31] C. Shi and R.W. Bodersen. A perturbation theory on statistical quantization effects in fixed-point DSP with non-stationary inputs. In *IEEE International Symposium on Circuits and Systems (ISCAS'04)*, pages 373–376, Vancouver, Canada, May 2004.

[32] A. Sripad and D. L. Snyder. A Necessary and Sufficient Condition for Quantization Error to be Uniform and White. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(5):442–448, October 1977.

[33] W. Strauss. DSP chips take on many forms. DSP-FPGA.com Magazine, March 2006.

[34] S. Wadekar and A. Parker. Accuracy Sensitive Word-Length Selection for Algorithm Optimization. *IEEE/ACM International Conference on Computer Design (ICCAD'98)*, pages 54–61, 1998.

[35] B. Widrow, I. Kollár, and M.-C. Liu. Statistical Theory of Quantization. *IEEE Transactions on Instrumentation and Measurement*, 45(2):353–361, April 1996.

[36] B. Wu, J. Zhu, and F.N. Najm. Dynamic-Range Estimation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25:1618–1636, 2006.