ANALYTICAL ACCURACY EVALUATION OF FIXED-POINT SYSTEMS

Romuald Rocher, Daniel Menard, Olivier Sentieys and Pascal Scalart

ENSSAT/IRISA
University of Rennes I
6 rue de Kerampont, BP447
22300 Lannion, France
name@enssat.fr

ABSTRACT

To satisfy cost constraints, application implementation in embedded systems requires fixed-point arithmetic. Thus, applications defined in floating-point arithmetic must be converted into a fixed-point specification. This conversion requires accuracy evaluation to ensure algorithm integrity. Indeed, fixed-point arithmetic generates quantization noises due to some bits elimination during a cast operation. These noises propagate through the system and modify computing accuracy. In this paper, an accuracy evaluation model based on an analytical approach is presented and valid for all systems including arithmetic operations. The LMS algorithm example is developed and its validity is verified through experimentations.

1. INTRODUCTION

Digital signal processing applications are specified in floating-point to prevent problems due to computing accuracy. However, to satisfy cost constraints, application implementation in embedded systems requires fixed-point arithmetic. Thus, the application defined in floating-point arithmetic must be converted into a fixed-point specification. To reduce application time-to-market, tools to automate floating-point to fixed-point conversion are needed. In these tools, an important stage corresponds to accuracy evaluation of fixed-point specification. Indeed, fixed-point arithmetic generates quantization noises due to some bits elimination during cast operations. These noises propagate through the system and modify computing accuracy. Computing accuracy damages must be contained to ensure algorithm integrity and application performances.

Application accuracy can be evaluated through different manners. On one hand, accuracy can be evaluated with fixed-point simulations [1, 5]. However, these methods require high computing time since a new simulation is required as soon as a fixed-point format changes in the system. So, these approaches lead to very significant optimisation time inside the fixed-point conversion process. On the other hand, a fixed-point specification accuracy can be evaluated with analytical methods. These approaches determine a mathematical expression for the accuracy metric. These methods require very short computing time compared to methods based on simulation. In this domain, existing approaches are only valid for linear and time invariant (LTI) systems [3] or non-LTI and non recursive systems [6] or need restrictive hypothesis about noises [2]. Thus, the aim of this paper is to propose a method which evaluates the fixed-point accuracy of any system based on arithmetic operations (additions, subtractions, multiplications and divisions). Especially, non-LTI systems with a recursion as adaptive filters are supported. The accuracy is determined through the Signal to Quantization Noise Ratio (SQNR) of the considered application for any quantization law (rounding or truncation).

This paper is organized as follows. First, quantization noises are introduced. The modelization of noise source is presented and the noise propagation through the system is summarized. A general model to determine analytically noise propagation through arithmetic operation is deduced. This model takes into account the different noise source types : the noises can be scalar, vector or matrix. Then, the considered system is modelized. This system is general (LTI, non-LTI, recursive or non-recursive) and is modelized through an expression of its transfert function and impulse response. Given that the system can be non-LTI, the transfert function and its impulse response are time-varying. This expression lets us compute the noise power at the system output with an analytical relation based on noise source statistical parameters and the system time-varying impulse response. This expression is unbiased and leads to infinite sums. Finally, the method is applied to different systems such as the LMS algorithm and its quality is evaluated by experimentations. Model execution times have been measured on Matlab. The approach reduces dramatically the noise power computing time compared to approaches based on fixed-point simulations. These results show the ability of our methodology to reduce fixed-point system development time.

2. QUANTIZATION NOISES

2.1 Quantization noises model

A data quantization can be modelized by the sum of the data and a uniformly distributed white noise [7]. This white noise (or quantization noise) is uncorrelated with the signal and other noise sources. According to the type of quantization, the noise distribution will differ. Three quantization modes can be considered. It corresponds to truncation, conventional rounding and convergent rounding [3].

Let n be the number of bits for the fractional part after the quantization process and k the number of bit eliminated during the quantization. The quantization step q after the quantization is equal $q = 2^{-n}$. The quantization noise mean and variance are presented in Table 1 for the three quantization modes.

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 1: Quantization noise first and second order moment for the three quantization modes.

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

Table 2: Terms values α_1 and α_2 of equation (1) for different operations $\{+,-,\times,\div\}$

2.2 Quantization noises propagation

The aim of this part is to define the noise propagation models. The propagation of two scalar noises b_x and b_y associated with two input operator X and Y generates an output noise b_z expressed as the sum of the two input noises b_x and b_y multiplied by signal terms as explained in [6]. The terms α_i are summarized in Table 2 for the different arithmetic operations.

$$b_z = \alpha_1 b_x + \alpha_2 b_y \tag{1}$$

In the case of non-scalar noise sources (vectors or matrix), the last model is not valid since terms commutativity doesn't exist. Indeed, each noise source on the operation input can be multiplied by signal term on the left or on the right. Thus, the general model for noise source propagation is expressed by the multiplication of each input noise by two signal terms (A et D)

$$b_z = A_x b_x D_x + A_y b_y D_y \tag{2}$$

The terms A and D are defined by the different operations crossed by the noise source.

3. SYSTEM MODELIZATION

In this section, the system crossed by the noise sources is characterized. This characterization lets us compute the system output noise power. Let N_e be the number of noise sources. In the expression (2), the crossed noise terms do not appear. So, each noise source $b_i(n)$ at time n propagates through the system and contributes to the generation of system output noise $b'_i(n)$. The system output noise $b_y(n)$ is the sum of all contributions as expressed in equation (3) and presented in Figure 1.

$$b_y(n) = \sum_{i=1}^{N_e} b'_i(n) \tag{3}$$

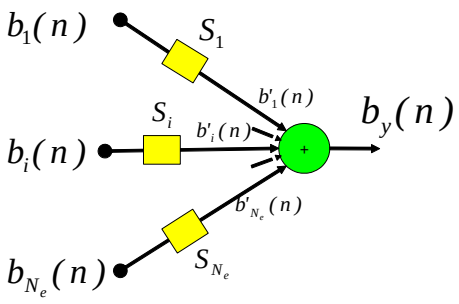

Figure 1: System modelization

Each contribution $b'_i(n)$ comes from noise source $b_i(n)$ propagation through the system S_i. Thus, to determine completely the output noise $b_y(n)$, each block S_i must be analytically characterized.

3.1 System characterization

The noise source $b_i(n)$ leads to a noise contribution $b'_i(n)$ on the system output. This contribution depends on $b_i(n)$ but also on the previous samples $b_i(n-k)$ for $k \in [1:Q_i]$ because of delays inserted in the system. Moreover, the contribution $b'_i(n)$ depends on its previous samples $b'_i(n-m)$ for $m \in [1:P_i]$ due to recursions in the system. Then, $b'_i(n)$ can be analytically written by the following expression

$$b'_i(n) = \sum_{k=0}^{Q_i} g_i(n-k) b_i(n-k) + \sum_{m=1}^{P_i} f_i(n-m) b'_i(n-m) \tag{4}$$

where $g_i(n-k)$ represents the contribution of noise source b_i at time $(n-k)$ to system output noise and $f_i(n-m)$ that of noise b'_i at time $(n-m)$. These terms f_i and g_i are time-varying and depend on system implementation. For LTI systems, these terms correspond to filter coefficients. The expression (4) lets us introduce the time-varying transfert function $H_i(z)$ defined as

$$H_i(z) = \frac{\sum_{k=0}^{Q_i} g_i(n-k) z^{-k}}{1 - \sum_{m=1}^{P_i} f_i(n-m) z^{-m}} \tag{5}$$

This equation modelizes the system crossed by the noise source $b_i(n)$. Nevertheless, the aim is to express output noise power using only input noises statistics and system characteristics. Then, expression (4) must be developed to express contribution $b'_i(n)$ with only input noises terms b_i introducing time-varying impulse response.

3.2 Time-varying impulse response

In this part, the time-varying impulse response of the system is determined. Developing recurrence in equation (4), the next expression is obtained

$$b'_i(n) = \sum_{k=0}^{n} h_i(k) b_i(k) \tag{6}$$

In this relation, contribution $b'_i(n)$ is expressed by noise source $b_i(n)$ and all its previous samples where h_i represents the time-varying impulse response of the system S_i. This impulse response is recursively obtained with the following relation

$$h_i(k) = \sum_{j=1}^{P} f_i(j) h_i(k+j) + g_i(k) \quad (7)$$

This time-varying impulse response represents system influence on noise source $b_i(n)$ and must be determined using system characteristics. The system is composed by arithmetic operations. In the section 2, the propagation of a noise through a system including arithmetic operations has been modelized by the multiplication of two signal terms A et D. as shows equation (2). Thus, the time-varying impulse response, modelizing input noise $b_i(n)$ propagation through the system, is equivalent to the multiplication of the noise source $b_i(k)$ by two terms $A_i(k)$ and $D_i(k)$.

$$h_i b_i <=> A_i b_i D_i \quad (8)$$

So, the contribution $b'_i(n)$ presented in equation (6) is equal to

$$b'_i(n) = \sum_{k=0}^{n} A_i(k) b_i(k) D_i(k) \quad (9)$$

The output noise $b_y(n)$ is the sum of all noise source contributions

$$b_y(n) = \sum_{i=1}^{N_e} \sum_{k=0}^{n} A_i(k) b_i(k) D_i(k) \quad (10)$$

More generally, the considered system can be composed by different serial/parallel blocks. In that case, the previous expression is still valid. However, signal terms A and D are more complex because they are made-up of different signal terms.

4. OUTPUT NOISE POWER EXPRESSION

4.1 Noise power

The output noise power P_b is got using second order moment of expression (10). The non correlation between signal terms and noises allows to obtain the following expression for output noise power P_b.

$$\begin{aligned} P_b &= E[b_y^2(n)] \\ &= \sum_{i=1}^{Ne} \sigma_{b_i}^2 Ka_i + \sum_{i=1}^{Ne} \sum_{j=1}^{Ne} m_{b_i} m_{b_j} Km_{ij} \end{aligned} \quad (11)$$

where m_{b_i} and $\sigma_{b_i}^2$ represent input noises $b_i(n)$ mean and variance. Moreover, Ka_i and Km_{ij} are signal terms defined by the following expression

$$Ka_i = \sum_{k=0}^{n\to\infty} E\left[Tr\big(D_i(k) D_i^t(k)\big) Tr\big(A_i(k) A_i^t(k)\big) \right] \quad (12)$$

$$Km_{ij} = \sum_{k=0}^{n\to\infty} \sum_{m=0}^{n\to\infty} E\left[Tr\big(A_i(k) 1_N D_i(k) D_j^t(m) 1_N A_j^t(m)\big) \right] \quad (13)$$

Systems	**Average relative error**	**Maximum relative error**
IIR 8	0.8%	3.3%
MP3 coder/decoder	6.62%	20.57%
Volterra filter	1.79%	3.22%
Correlator	1.35%	5.78%

Table 3: Average and maximum relative error committed on different systems

with 1_N the N-size matrix composed by 1. The expressions of Ka and Km are obtained by a floating-point simulation. These terms are independent from noise sources and lead to constants in the output noise power expression. Noise statistics m and σ^2 depend on fixed-point formats and are variables of output noise power expression.

The expression (11) is unbiased since no hypothesis has been done about the system. The terms Ka and Km are defined by infinite sums. In practice, these sums are truncated after a number p representative of the infinite sums. This number p depends on signal correlation inside the terms Ka and Km. Nevertheless, according to the different carried out experimentations, a number p equal to 500 leads to very realistic results. Moreover, this expression includes average terms computing. These terms require N_t samples to get realistic results. In practice, a number N_t equal to 100 leads to satisfying modelizations.
Another approach has been developed to modelize the infinite sums and to reduce our approach complexity. This model is based on linear prediction. Relation (7) between impulse response terms is linearized with coefficients minimizing quadratic error between impulse response terms and estimated terms. This approach lets us modelize infinite sums with prediction coefficients. The introduced bias has been measured.

5. EXPERIMENTATIONS

In this section, experimentations are carried out to validate our model. LTI and non LTI systems are studied to apply our model in all cases.

5.1 Experimentation on LTI and non-LTI non recursive systems

In this section, the proposed model is evaluated on LTI systems (Infinite Impulse Response filter and MP3 coder/decoder) and non-LTI and non recursive systems (Volterra system and correlator). Average and maximum relative error obtained between noise power estimated with our model and real noise power got by simulations is presented on Table 3.

For the 8-order IIR filter, relative error depends on chosen structure (Direct or Transposed Form). Nevertheless, relative error is always less than 3.3%. The MP3 coder/decoder is made-up of a polyphase filter and a Discrete Cosine Transform (DCT). It leads to a maximum error equal to 20.57%.

This error represents a difference less than 2 dB between the real noise power and our model estimation. For the 2^{nd} order Volterra filter and the correlator, relative error is less than 5.78%. Thus, in all cases, relative error is low with an average value about 2.64% for these four systems. These results let us validate our model for LTI and non-LTI and non-recursive systems. The estimation quality is definitely sufficient for the fixed-point design process.

5.2 LMS Experimentations

5.2.1 *Fixed-point LMS Algorithm*

To illustrate previous results and experiment model on a non-LTI system with recursion, the Least Mean Square (LMS) example is under consideration. The LMS adaptive algorithm addresses the problem of estimating a sequence of scalars $y(n)$ from a N length vector $X(n) = [x(n), x(n-1)...x(n-N+1)]^t$ [4]. The linear estimate of $y(n)$ is $W^t(n)X(n)$ where $W(n)$ is a N length vector which converges to the optimal vector W_{opt} in the Mean-Square Error (MSE) sense according to the following equation

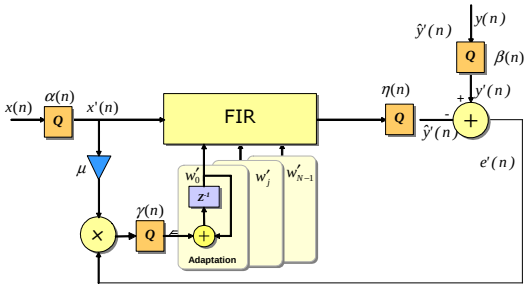

Figure 2: LMS algorithm

$$W(n+1) = W(n) + \mu X(n)(y(n) - W^t(n)X(n)) \qquad (14)$$

where μ is a positive constant representing the adaptation step. In fixed-point implementation, four noise sources are introduced (figure 2). The noises $\alpha(n)$ and $\beta(n)$ are generated by input data $X(n)$ quantization and desired signal $y(n)$ quantization. The term $\gamma(n)$ comes from product between $\mu X(n)$ and error $e(n)$ equal to $y(n) - W^t(n)X(n)$. The noise $\eta(n)$ is generated by the inner product $W^t(n)X(n)$. The terms m and σ^2 represent mean and variance of each noise source.

5.2.2 *Accuracy model*

The system crossed by each noise is determined. The noise $\gamma(n)$ is analyzed in details to illustrate our approach. The noise term $\gamma(n)$ propagation is shown on figure 3. The transfer function of its propagation is given by the following expression

$$H_\gamma(z) = X^t(n)\frac{z^{-1}}{1-(I_N - \mu X(n-1)X^t(n-1))z^{-1}} \qquad (15)$$

where I_N is the N size identity matrix. Its contribution $\gamma'(n)$ is written using its time-varying impulse response h_γ as follows

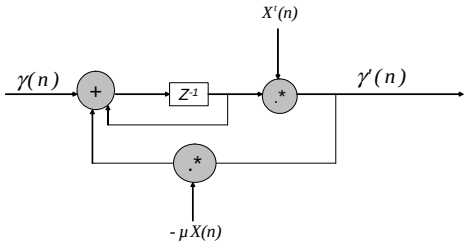

Figure 3: Noise source $\gamma(n)$ propagation

$$\begin{aligned}\gamma'(n) &= \sum_{k=0}^{n-1} h_\gamma(k)\gamma(k) = \sum_{k=0}^{n-1} A_\gamma(k)\gamma(k) \\ &= \sum_{k=0}^{n-1} X^t(n)F(n,k)\gamma(k) \qquad (16)\end{aligned}$$

with

$$F(n,k) = \prod_{m=k+1}^{n-1}\Big(I_N - \mu X(m)X^t(m)\Big)$$

The time-varying impulse response h_γ is defined as the product of two signal terms A_γ on the left and D_γ on the right. The term D_γ doesn't appear in the expression since all multiplications are made on the left. The contributions of the three other terms can be obtained with the same method. The noises $\eta(n)$ and $\beta(n)$ are scalar. Then, the terms A modelizing their propagation through the system are also scalars which lets us write $Tr(AA^t) = A^2$ for input noises $\eta(n)$ and $\beta(n)$. The output noise power is computed using expression (11)

$$\begin{aligned}E[b_y^2(n)] &= \sum_{k=0}^{n}\sigma_\alpha^2 E\Big[Tr\big(A_\alpha(k)A_\alpha^t(k)\big)\Big] + \sum_{k=0}^{n}\sigma_\eta^2 E[A_\eta^2(k)] \\ &+ \sum_{k=0}^{n}\sigma_\beta^2 E[A_\beta^2(k)] + \sum_{k=0}^{n}\sigma_\gamma^2 E\Big[Tr\big(A_\gamma(k)A_\gamma^t(k)\big)\Big] \\ &+ \sum_{k=0}^{n}\sum_{l=0}^{n} E\Big[Tr\big(M(k)M^t(l)\big)\Big] \qquad (17)\end{aligned}$$

with

$$\begin{aligned}M(k) &= A_\alpha(k)m_\alpha + A_\beta(k)m_\beta + A_\eta(k)m_\eta + A_\gamma(k)m_\gamma \\ A_\alpha(k) &= \mu X^t(n)F(n,k)\Big(e(k) - X(k)W^t(k)\Big) + W^t(n)\Delta(n-k) \\ A_\beta(k) &= \mu X^t(n)F(n,k)X(k) \\ A_\eta(k) &= -\mu X^t(n)F(n,k)X(k) + \Delta(n-k)\end{aligned} \qquad (18)$$

with Δ the Kronecker symbol.

5.2.3 *Estimation quality*

To evaluate our model quality, experimentations have been made. The relative error between the noise power estimated

with our model and its real value obtained by simulation is evaluated. Figure 4 shows relative error committed by our model on the $N = 32$ size LMS. The results are presented versus the number p chosen to represent infinite sums and the correlation of input signal $x(n)$. The signal can be white ($\delta = 0$), fairly correlated ($\delta = 0.5$) or very correlated ($\delta = 0.95$). As p increases, relative error decreases. Indeed, higher p is, more terms are included in sums computing which leads to a better result. Moreover, relative error convergence speed depends on input data correlation. For non correlated input data, relative error convergence is slower than the one for very correlated input data. In fact, relative error is less than 20% after 300 points for very correlated input data, after 350 points for fairly correlated data and after 550 points for uncorrelated input data.

Thus, number p determining points number in infinite sums computing depends on input data correlation. Nevertheless, with experimentation presented after 500 points, relative error is less than 25% in all cases wich represents a difference less than 1 dB between noise power obtained with our model and real noise power. For linear prediction model, obtained relative error is equal to 21%. For the other size of LMS algorithm, same results are obtained.

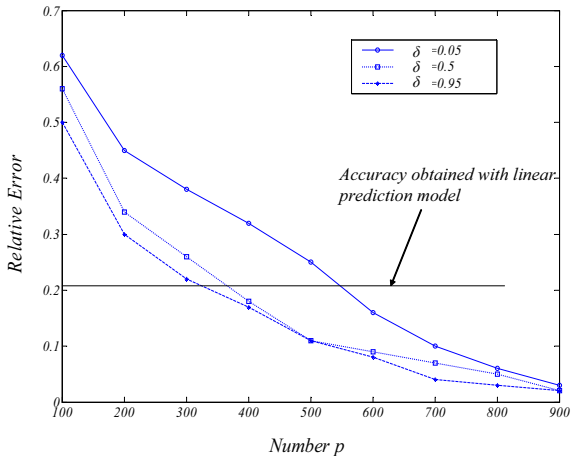

Figure 4: Relative error for the 32 size LMS

5.3 Fixed-point specification optimization

The model has been compared in terms of execution time to methods based on simulation for the fixed-point optimization process. The experiments have been conducted on Matlab and the results are given in figure 5. For our analytical approach, first the analytical expression of Pb must be computed and it represents the most time consuming part equal to 46 seconds for method based on infinite sums and 4 seconds for linear prediction model. Then, each iteration of this optimization process corresponds to noise power expression evaluation whose computing time is negligible. For the LMS algorithm, our method leads to time gain after less than 100 iterations which represents an execution time equal to 46 seconds. For an optimization process with about 30 variables, between 10000 and 100000 iterations are required. With the model based on linear prediction, our approach leads to time gain after only 10 iterations compared to methods based on simulations leading to an execution time equal to 4 seconds. The interest of our model is demonstrated to reduce fixed-point systems development time.

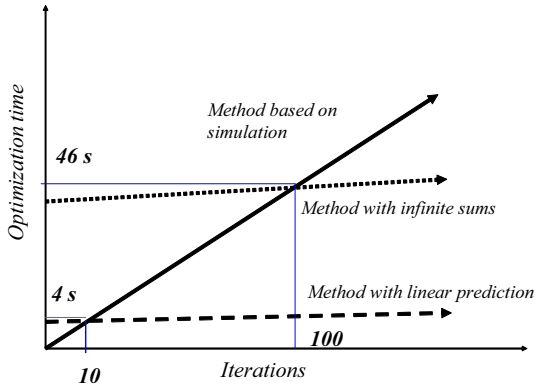

Figure 5: Optimization time for our approach and method based on simulations

6. CONCLUSION

In this paper, a model to determine analytically the accuracy of a fixed-point system is presented. The model is developed for all systems made-up of arithmetic operations and is valid for all quantization laws. The method is unbiased and leads to infinite sums to compute the output noise power. A method based on linear prediction has been introduced to reduce our method complexity. It has been applied to different systems such as LMS algorithm to verify its validity. This method allows to reduce conversion time of floating-point to fixed-point systems.

REFERENCES

[1] P. Belanovic and M. Rupp, "Automated Floating-point to Fixed-point Conversion with the fixify Environment," *IEEE Rapid System Prototyping*, pp. 172-178, 2005.

[2] J.M. Cheneaux and L.S. Didier and F. Rico, "The Fixed CADNA Library," *Real Number and Computers*, Sep. 2003.

[3] G.A. Constantinides and P.Y.K. Cheung and W.Luk "Synthesis and Optimization of DSP Algorithms," *Kluwer Academic Publishers*, 2004.

[4] S. Haykin, "Adaptive Filter Theory, " *Englewood Cliffs, NJ:Prentice-Hall*, 2nd edition, 1991.

[5] S. Kim and K. Kum and S. Wonyong, "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. 1453-1464, Nov. 1998.

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

[7] B. Widrow and I. Kollár and M.-C. Liu, "Statistical Theory of Quantization," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 353-361, Apr. 1996.