ACCURACY EVALUATION BASED ON SIMULATION FOR FINITE PRECISION SYSTEMS USING INFERENTIAL STATISTICS

Justine Bonnot, Karol Desnos, Daniel Menard

Univ. Rennes, INSA Rennes, CNRS, IETR - UMR 6164, F-35000 Rennes, France

ABSTRACT

The conversion of an algorithm to fixed-point arithmetic is commonly achieved with a large and fixed-number of simulations. Nevertheless, when simulating a fixed and arbitrary large number of samples, no confidence information is given on the characterization, and this method is often time-inefficient. To overcome this limitation, we propose a new method for noise evaluation. The error induced by fixed-point coding is statistically characterized to compute the noise power with an adaptive and reduced number of simulations. From user-defined confidence requirements, the proposed method computes the minimal number of simulations to obtain a confidence interval of the noise power. Experiments on varied signal-processing elementary blocks show that the proposed method requires on average the simulation of only 0.04% of the simulation set required by State of the Art techniques to estimate the noise power of a 64^{th} order FIR filter with a relative error less than 0.01%.

***Index Terms*—** Fixed-point, Statistics, Error, Noise Power, Signal Processing

1. INTRODUCTION

The competition to design faster, cheaper and more energy-efficient electronic systems is becoming not only economical but also an urging answer in the need to save the available energy resources. According to the Semiconductor Industry Association and Semiconductor Research Corporation, the total energy required by computing systems will exceed the estimated world's energy production by 2040, if no significant improvement is obtained in terms of energy-aware computing systems [1]. In this context, approximate computing is an active field of research that trades-off the output quality of a system for its energy consumption. Approximate computing benefits from the error resilience of applications in signal, image or video processing and artificial intelligence fields. Approximations can be applied at three different levels: at the hardware, computation and/or data levels. When applying an approximation at the data level, the volume of data to process can be reduced, and/or the format to encode the data modified.

This project has received funding from the French Agence Nationale de la Recherche under grant ANR-15-CE25-0015 (ARTEFaCT project).

For instance, fixed-point coding offers efficient arithmetic operators in terms of hardware resources, latency and energy. Fixed-point arithmetic becomes inevitable for ultra-low power system for which data precision is a way to reduce the energy consumption. In the fixed-point conversion process, the more complex task corresponds to the optimization of the word-length of each data, which is known to be a NP-hard combinatorial problem [2]. The word-length optimization process requires numerous iterations and at each iteration, the error induced by the finite precision has to be characterized and its impact on the output Quality of Service (QoS) has to be measured. Classically, instead of measuring the impact of the finite precision on the output QoS, an intermediate metric is used [3]. In the rest of the paper, the intermediate metric used is the quantization noise power, which measures the loss of accuracy due to finite precision.

The error induced by fixed-point coding is generally evaluated by measuring the loss of QoS between the algorithm implemented in infinite precision and in finite precision. The error can be measured with two types of approaches: 1) Analytical methods [4–8] have been proposed to avoid time-consuming simulations, by mathematically expressing error statistics. The system in fixed-point coding is then replaced by the system in infinite precision to which is added a noise b characterized by statistical parameters. Nevertheless, analytical techniques are complex, hard to automate and cannot be applied to systems with un-smooth operators, i.e. non-continuous operators. 2) Functional simulation techniques simulate the application using infinite and finite precision, and compute the Peak Signal-to-Noise Ratio in both cases at the output of the system. Given the obtained results, the noise power due to the considered data formats is obtained. This later method is widely employed since it is not limited by the applicability of analytical techniques. Nevertheless, to mimic the finite precision effects, fixed-point coding has to be emulated. Several commercial high-level tools to design digital signal processing applications can be used to emulate fixed-point coding, as CoCentric (Synopsis) [9], or C++ classes proposed in SystemC [10, 11]. SystemC fixed-point data-types are particularly slow to simulate since they can be two to three orders of magnitude slower than the execution of floating-point data-types from the Arithmetic-Logic Unit (ALU). In the rest of the article, the simulation of fixed-point

variables using SystemC data-types is considered. When using simulation-based techniques, the noise power is generally computed by simulating an arbitrary, and large, number of random inputs [12]. However, the quality of the estimated statistics is not evaluated, and this method can be ineffective in terms of simulation time. Sedano et al. [13] proposed, similarly as our approach, to use inferential statistics to infer the number of inputs to simulate. They derived that for a noise constraint of 10^{-k}, 10^{k+1} input points were required for a fair evaluation of the noise power. In our approach, a different expression is derived for the number of input points.

We propose an efficient methodology using inferential statistics to determine a reduced number of simulations to compute an accurate enough estimation of the power of the noise induced by fixed-point conversion. The proposed method iteratively adapts the number of samples to simulate depending on accuracy and confidence constraints. Our approach exploits the statistical properties of the approximation error. The number of needed simulations and the characterization time are drastically reduced compared to [13]. This method is demonstrated on three signal processing use-cases using varied accuracy and confidence constraints.

The remainder of this paper is organized as follows: Section 2 details the proposed estimation method for the noise power. Section 3 presents the experimental setup and the obtained results in terms of number of simulated samples.

2. ESTIMATION OF THE NOISE POWER AT THE OUTPUT OF AN APPLICATION

2.1. Motivations

The noise power induced by finite precision at the output of an application can be expressed as the second order moment of the random variable e_x, where e_x represents the error distance and is expressed as:

$$e_x = |x_Q - x_\infty| \tag{1}$$

where x_Q and x_∞ are the variable x expressed in finite and infinite precision, respectively. The impact of finite precision may be induced by custom floating-point [14] or fixed-point coding. In the following, only fixed-point coding will be considered.

Usually, to compute the noise power P induced by fixed-point conversion using simulations, an arbitrary large number of samples N_{Samples} is taken. In the literature, N_{Samples} ranges from 10^5 [12] to 10^{12} [15]. For determining the noise power P induced by finite precision, two different versions of the application are simulated as presented in Figure 1. The distance e_x between the output of the application with infinite precision x_∞ and the output of the application with finite precision x_Q is measured and squared for each simulated sample. The expected value of these distances is then computed to obtain P. The slow software simulation of fixed-point data-types as well as the high number of samples to simulate makes generally fixed-point conversion a long and tedious task. While a lot of work has been done to propose more efficient and faster data-types, we show that a fixed and large number of samples to simulate is inefficient and that number can drastically be reduced to speed up the exploration.

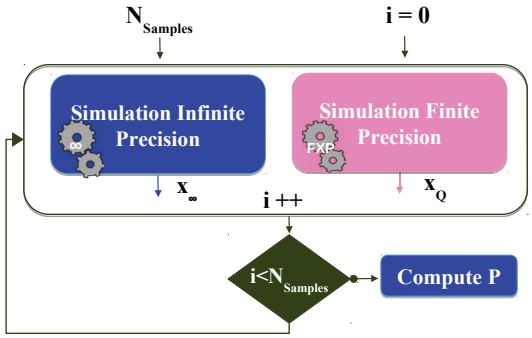


Fig. 1: Simulation-based determination of the noise power P

The noise power P is a statistical parameter expressed as in Equation 2 and can be estimated using inferential statistics. The noise power is computed by averaging all the $e^2_{x,i}$ values with $i \in \mathcal{I}$, where $\mathcal{I}$ represents the input set.

$$P = E[e_x^2] \; \forall \; x \in \mathcal{I} \tag{2}$$

Inferential statistics [16] aim at predicting the behavior of a large population $\mathcal{I}$ using a subset of this population. This statistical analysis is particularly interesting in the case of large simulation set, where the exhaustive characterization of the error and of the noise power is not economically viable since not in line with time to market constraints. Using inferential statistics, the input set is sampled to give an estimation of an interval with an accuracy h and a probability p that the real value is contained within the estimated confidence interval, instead of simulating exhaustively all the possible inputs x in $\mathcal{I}$.

The objectives of the proposed method are: 1) to estimate the noise power P induced by fixed-point conversion, more efficiently, using a reduced but sufficient number of samples, 2) to provide the estimated error characteristics within a given confidence constraint, which is normally not the case with a fixed amount of samples. The proposed method computes the minimal number of samples to simulate, to estimate the noise power P according to (h, p), where h is the accuracy on the estimation and p the probability that the estimated interval contains the real value. N_P represents the minimal number of samples to estimate P according to (h, p).

2.2. Minimal number of samples to estimate P, N_P

As expressed in Equation 2, the power of the noise induced by finite precision at the output of an algorithm whose inputs are in $\mathcal{I}$ can be expressed as:

$$P = \frac{1}{N} \sum_{i \in \mathcal{I}} e_{x,i}^2 \tag{3}$$

where $e_{x,i}^2$ is the squared Error Distance of the i^{th} stimuli on a sample $\mathcal{I}$ of size N.

To estimate the real value of P, the empirical mean $\overline{\mu_{e_x^2}}$, a punctual estimator of the expected value of the squared error distances P is used. That is to say, $\overline{\mu_{e_x^2}}$ is an estimation of $P = E[e_x^2]$ computed over a subset of $\mathcal{I}$. $\overline{\mu_{e_x^2}}$ is used to compute the theoretical number of samples N_P to simulate to get an estimation according to the constraints (h, p). To estimate N_P, the standard deviation of the squared error distances is also required. The empirical mean $\overline{\mu_{e_x^2}}$ and the empirical standard deviation $\tilde{S}^2$, a biased estimator of the standard deviation σ_e, are computed over $T \leq N$ samples as:

$$\overline{\mu_{e_x^2}} = \frac{1}{T} \sum_{i=1}^{T} e_{x,i}^2 \tag{4}$$

$$\tilde{S}^2 = \frac{1}{T} \sum_{i=1}^{T} (e_{x,i}^2 - \overline{\mu_{e_x^2}})^2 \tag{5}$$

The estimators $\overline{\mu_{e_x^2}}$ and $\tilde{S}^2$ are associated to confidence intervals $\text{IC}_{\mu_{e_x^2}}$ and $\text{IC}_{\sigma_{e_x^2}}$ respectively, defined such that they include $\mu_{e_x^2}$ and $\sigma_{e_x^2}$ with a probability p. Then, according to the Central Limit Theorem [16], since $(y_1, y_2, ..., y_T) = (e_1^2, e_2^2, ..., e_T^2)$ are belonging to the same probability set, are independent and identically distributed, Equation 6 is verified if the number of samples N_P is higher than 30. No assumption has to be made on the distribution of the population. In Equation 6, $\mathcal{N}(0, \sigma_{e_x^2})$ represents a gaussian distribution whose mean is 0 and standard deviation is $\sigma_{e_x^2}$.

$$\sqrt{N_P}(\overline{\mu_{e_x^2}} - \mu_{e_x^2}) \xrightarrow{\text{law}} \mathcal{N}(0, \sigma_{e_x^2}) \tag{6}$$

The confidence interval $\text{IC}_{\mu_{e_x^2}}^p$ is developed in Equation 7 and contains $\mu_{e_x^2}$ with a probability p. The term $a_{\mu_{e_x^2}}^{\alpha}$ embodies the accuracy on the estimation and is computed as in Equation 8. $z_\alpha(p)$ is given by the table of the standard normal distribution given p, and $\alpha = 1 - p$. N_P is the minimal number of samples to simulate to get an estimation respecting the constraints (h, p).

$$\text{IC}_{\mu_{e_x^2}}^p = [\overline{\mu_{e_x^2}} - a_{\mu_{e_x^2}}^{\alpha}; \overline{\mu_{e_x^2}} + a_{\mu_{e_x^2}}^{\alpha}] \tag{7}$$

$$a_{\mu_{e_x^2}}^{\alpha} = z_\alpha(p) \cdot \frac{\tilde{S}}{\sqrt{N_P - 1}} \tag{8}$$

The desired accuracy h on the estimation of the noise power impacts the number of samples to simulate as expressed in Equation 9. To get a desired accuracy of h, $a_{\mu_{e_x^2}}^{\alpha}$ must be less than or equal to h.

$$N_{\mu_{e_x^2}} > \frac{z_\alpha^2 \cdot \tilde{S}^2}{h^2} \tag{9}$$

Algorithm 1 presents the computation of N_P with the described method. The population of the squared error distances, on which inferential statistics are applied is the set $\mathcal{E} = \{e_{x,i}^2 / i \in \mathcal{I}\}$. To sample the population $\mathcal{E}$, a random sampling method is used. To converge towards the minimal number of samples to simulate, a refreshment period $T \geq 30$ is used. Every T samples, the punctual estimators $\overline{\mu_{e_x^2}}$ and $\tilde{S}^2$ are computed to estimate the number of simulations N_P to compute P with the confidence constraints (h, p).

Algorithm 1 Proposed Computation of N_P

procedure COMPUTE$\text{N}_P(\mathcal{E}, h, p, T)$
  $E \leftarrow \emptyset$
  $n \leftarrow 0$
  repeat
    $(e_n^2, .., e_{n+T}^2) \leftarrow \text{sampling}(\mathcal{E}, T)$
    $E \leftarrow E \bigcup (e_n^2, .., e_{n+T}^2)$
    $\overline{\mu_{e_x^2}} \leftarrow \text{computeMean}(E, n + T)$ ▷ Equation 4
    $\tilde{S}^2 \leftarrow \text{computeSD}(E, n + T, \overline{\mu_{e_x^2}})$ ▷ Equation 5
    $N_P \leftarrow \text{computeN}(\tilde{S}^2, h)$ ▷ Equation 9
    $n \leftarrow n + T$
  until $n \geq N_P$
return N_P
end procedure

3. EXPERIMENTAL STUDY

In this section, the proposed characterization method is evaluated on several elementary blocks of signal processing applications. The convergence of the estimated intervals towards the accurate value of P is demonstrated with a 64^{th} order Finite Impulse Response (FIR) filter converted to 16-bit fixed-point coding. Then, the approach is applied to two different types of quantization. The proposed method does not depend on the complexity of the implemented application but on the distribution of the noise power. For the different elementary blocks presented, the considered block is implemented in C++ both in floating-point and fixed-point using SystemC [10] dynamic data-types. The accuracy of estimation has been measured by computing the relative error between the mean estimated noise power, and the accurate noise power computed over all the input samples in $\mathcal{I}$. The size of the input set $\mathcal{I}$ is set up to 10^5 as in [12]. The goal of the experimental study is, according to different values of p, to compute the number of simulations required to obtain an evaluation of the noise power with a required accuracy h and confidence p.

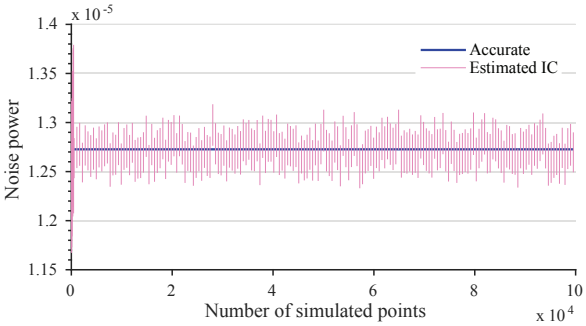
3.1. Example of a FIR Filter

The conversion of a 64^{th} order FIR filter is under consideration. The proposed characterization method is presented with $p = 98\%$. Figure 2 represents the estimated confidence intervals on the noise power and the relative error of estimation

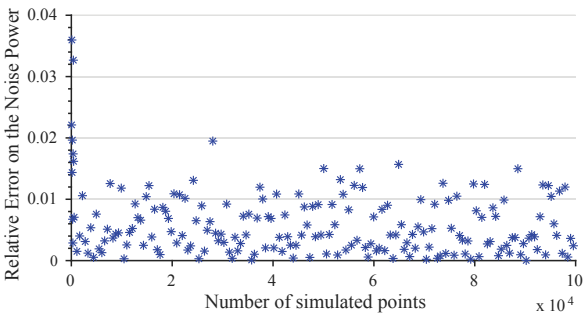
depending on the number of simulated points. The more samples are taken, the more accurate the estimation is since the width of the estimated interval is reduced. The refreshment period T has been set to 50 for the characterization done with less than 550 points, and afterwards to 500. The proposed characterization method estimates the noise power with a high accuracy ($h \leq 2\%$) from 150 simulated points. To apply our approach, the noise must be stationary. The delays of the 64 taps must contain relevant values. Thus, the noise can be analyzed only when the system has reached a steady-state. The approach can also be used on individual nodes, as long as the noise power at the output of the node can be measured.

3.2. Number of points to simulate to obtain a given precision

Table 1 lists for different elementary blocks of signal processing (FIR filter, quantization from floating-point to 8-bit fixed-point and quantization from 8-bit to 6-bit fixed-point) the number of points to simulate (the maximum is set to 10^5 [12]) to get an estimation for several constraints (h, p). For each block, the proposed method has been tested 1000 times with the constraints (h, p), IC% and accuracy of estimation respectively. Over the 1000 runs, ICExpe is the probability that the estimated interval contains the real noise power value and has been measured. The average number of simulated samples is also indicated as N_P.



(a) Confidence intervals on the noise power P.



(b) Relative error of estimation of the noise power P.

Fig. 2: Estimation of the noise power P for $p = 98\%$.

		Accuracy of estimation %			
		0.01		0.001	
	IC%	N_P	ICExpe	N_P	ICExpe
FIR(64)	68	45	68.4	55	69.7
	95	45	95	55.5	95.3
	98	30	98.7	77.63	98.8
	99	55	99	93.8	98.8
Quantization 8-bit to 6-bit	68	65	68	474	69
	95	145	94.5	1785.8	95.4
	98	145	98.4	2973	99
	99	205	99.2	3012	99.4
Quantization float to 8-bit	68	18	69.6	857.8	70.3
	95	42.7	95.1	3253.6	95.3
	98	39	98	5766	98.9
	99	54	99	5565	99.1

Table 1: N_P for varied elementary blocks and (h, p).

If $h = 0.01\%$, on average for the different probabilities p, the FIR filter requires the simulation of 44 samples, the quantization from 8-bit to 6-bit fixed-point 140 samples and the quantization from floating-point to 8-bit fixed-point 38 samples. If $h = 0.001\%$, on average, the FIR filter requires the simulation of 70 samples, the quantization from 8-bit to 6-bit fixed-point 2061 samples and the quantization from floating-point to 8-bit fixed-point 3861 samples, which corresponds to less than 4% of the whole input set (10^5). More points are needed for the quantization since the noise follows a uniform distribution.

The more the distribution of the noise power is centered around the mean, the least points are needed. When converting a massive algorithm to fixed-point, saving numerous simulations can greatly reduce the implementation time. The more an algorithm is massive, the more noise sources. The overall noise tends to follow a gaussian distribution. Besides, the speed of convergence strongly varies depending on the considered signal processing block. An adaptive sample-size method like the proposed one is thus more adapted to the measurement of the noise power rather than naive exhaustive simulations. The proposed method can then be used to ease the design space exploration of an application, but because of its statistical nature, not to verify a safety critical application.

4. CONCLUSION

In this article, we proposed a new method for characterizing the noise power of an application converted in fixed-point. From user-defined confidence requirements, the number of simulations required is determined by using statistical properties of the quantization error. This method is demonstrated for the estimation of the noise power of various signal processing elementary blocks. Validated by its accurate estimation of the noise power, this experimental study has demonstrated that the proposed method overcomes naive random simulations with a fixed number of samples by drastically reducing the amount of samples required for an accurate estimation, saving time and resources.

5. REFERENCES

[1] S. I. Association and S. R. Corporation, "Rebooting the it revolution, a call for action," https://www.src.org/newsroom/rebooting-the-it- revolution.pdf, 2015.

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

[3] D. Menard, R. Serizel, R. Rocher, and O. Sentieys, "Accuracy constraint determination in fixed-point system design," *EURASIP Journal on Embedded Systems*, vol. 2008, p. 1, 2008.

[4] B. Liu, "Effect of finite word length on the accuracy of digital filters–a review," *IEEE Transactions on Circuit Theory*, vol. 18, no. 6, pp. 670–677, 1971.

[5] R. E. Moore, "Interval arithmetic and automatic error analysis in digital computing," Stanford Univ Calif Applied Mathematics And Statistics Labs, Tech. Rep., 1962.

[6] G. Caffarena, J. A. López, A. Fernández-Herrero, and C. Carreras, "Sqnr estimation of non-linear fixed-point algorithms," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 522–526.

[7] D. Menard, R. Rocher, and O. Sentieys, "Analytical fixed-point accuracy evaluation in linear time-invariant systems." *IEEE Trans. on Circuits and Systems*, vol. 55, no. 10, pp. 3197–3208, 2008.

[8] G. Deest, T. Yuki, O. Sentieys, and S. Derrien, "Toward scalable source level accuracy analysis for floating-point to fixed-point conversion," in *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 726–733.

[9] F. Berens and N. Naser, "Algorithm to system-on-chip design flow that leverages system studio and systemc 2.0. 1," *Synopsys Inc., March*, 2004.

[10] T. Grtker, S. Liao, G. Martin, and S. Swan, "System design with systemc," 2010.

[11] W. Müller, W. Rosenstiel, and J. Ruf, *SystemC: methodologies and applications*. Springer Science & Business, 2007.

[12] H. Keding, F. Hurtgen, M. Willems, and M. Coors, "Transformation of floating-point into fixed-point algorithms by interpolation applying a statistical approach," in *9th International Conference on Signal Processing Applications and Technology (ICSPAT 98)*, 1998.

[13] E. Sedano, J. A. López, and C. Carreras, "Acceleration of monte-carlo simulation-based quantization of dsp systems," in *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference On*. IEEE, 2012, pp. 189–192.

[14] B. Barrois and O. Sentieys, "Customizing fixed-point and floating-point arithmetic-a case study in k-means clustering," in *SiPS 2017-IEEE International Workshop on Signal Processing Systems*, 2017.

[15] H. Keding, M. Willems, M. Coors, and H. Meyr, "Fridge: a fixed-point design and simulation environment," in *Proceedings of the conference on Design, automation and test in Europe*. IEEE Computer Society, 1998, pp. 429–435.

[16] R. Lowry, "Concepts and applications of inferential statistics," 2014.