Accuracy vs Speed Tradeoffs in the Estimation of Fixed-Point Errors on Linear Time-Invariant Systems

David Novo, Sara El Alaoui, and Paolo Ienne
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
CH–1015 Lausanne, Switzerland
Email: {david.novobruna, sara.elalaoui, paolo.ienne}@epfl.ch

***Abstract*—Fixed-point format is essential to most efficient *Digital Signal Processing (DSP)* implementations. The conversion of an algorithm specification to fixed-point precision targets the minimization of the implementation cost while guaranteeing a minimal processing accuracy. However, measuring such processing accuracy can be extremely time consuming and lead to long design cycles. In this paper, we study reference approaches to measure fixed-point errors of *Linear Time-Invariant (LTI)* systems without feedback. Unsurprisingly, we find the existing analytical approach significantly faster than a straightforward simulation-based estimation. However, we also show that such analytical approach can incur high estimation errors for some particular bitwidth configurations. Accordingly, we propose a new hybrid approach, which is able to reduce by up to 4 times the error of the analytical estimation, while still being more than 10 times faster than the simulation-based estimation.**

I. Introduction

The reduction of signal bitwidths is an effective way to reduce area, delay and energy consumption of fixed-point systems. However, such a reduction also decreases processing accuracy, which can result in functional failure when performed carelessly. Thus, every time a signal bitwidth is reduced, the system needs to be checked for functional correctness (i.e., if the system still provides a minimum *Signal-to-Quantization-Noise Ratio, SQNR*). Such an iterative process of finding the right bitwidth configuration is known as *fixed-point refinement* and can take up to 30% of the total implementation time [1].

The elimination of *Least-Significant Bits (LSB)* of a signal is performed by a quantizer, $Q[\cdot]$. The latter is an operator uniquely described by its fractional bitwidth, W_F, and its quantization mode, which can be a simple truncation or some form of roundoff. This quantizer, when acting on an input signal x, adds a quantization noise e to the signal, $e = x - Q[x]$. Such quantization noises propagate forward through system. The aggregate power of the different quantization noises at the output of the system is the magnitude typically monitored throughout the fixed-point refinement process [2], [3].

Different methods have been proposed to estimate the quantization noise (also known as fixed-point error) power at the output of a system. On the one hand, the straightforward simulation-based method computes the fixed-point error as the difference between the outputs produced by a reference high-precision implementation and the particular fixed-point implementation under evaluation [4], [5]. The latter, also known as statistical method, can be very accurate but suffers from very long simulation times. On the other hand, analytical methods, such as the one described by Menard et al. [6] or Shi and Brodersen [7], can deliver much faster estimates, but for a limited scope of applications, e.g., LTI systems.

In this paper, we study these two reference approaches to measure fixed-point errors of LTI systems without feedback and propose a new one. Our hybrid approach is able to reduce the estimation error of the reference analytical approach by up to 4 times while still being more than 10 times faster than simulation-based approach.

II. Analytical Quantization Noise Power model

In this section we survey the analytical approach to estimate quantization noise power used as reference in the rest of the paper. Menard et al. [6] define the expectation of quantization noise power at the output of a LTI system as

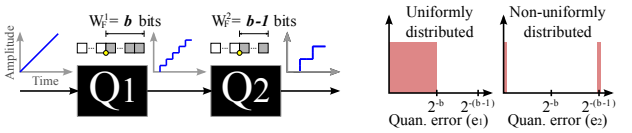
$$E({e_o}^2) = \sum_{i=1}^{N} k_i^2 \cdot \sigma_i + \left(\sum_{i=1}^{N} k_i \cdot \mu_i \right)^2 . \qquad (1)$$

The model includes two different types of components. On the one hand, σ_i and μ_i are the variance and mean of the quantization noise, e_i, that is injected when a quantizer i eliminates some LSB from a particular signal. On the other hand, k_i is the constant that defines the propagation of quantization noise e_i to the output of the system. Accordingly, the process of modeling analytically the quantization noise power at the output of a system can be divided in two phases: (1) noise injection, and (2) noise propagation and aggregation.

A. Noise Injection

Oppenheim and Weinstein [8] presented standard models for quantization errors based on linearizing the truncation of signals. Error signals, assumed to be uniformly distributed, white and uncorrelated, are injected whenever a truncation occurs. This approximate model has served very well for studying the quantization of continuous signals, such as in

978-3-9815370-0-0/DATE13/©2013 EDAA



(a) Removing bits with quantizers (b) Equivalent noise injection

Fig. 1. **Local correlation of quantization noises.** Q_1 quantizes a continuous signal producing the typical uniformly distributed quantization noise. However, Q_2 removes one bit of the output of Q_1 and the resulting quantization error is discrete, being either 0 or 2^{b-1}.

analog-to-digital converters. However, Constantinides et al. [9] provided a more accurate model for the truncation of signals that have already been discretized, which is normally the case in fixed-point system of multiple bitwidths. The variance and mean of the quantization noise injected when truncating an already quantized signal are defined as follows:

$$\mu_2 = \frac{1}{2}(2^{-W_F^1} - 2^{-W_F^2}), \sigma_2 = \frac{1}{12}(2^{-2\cdot W_F^1} - 2^{-2\cdot W_F^2}), \quad (2)$$

where W_F^1 and W_F^2 correspond to the fractional bitwidth of a quantizer Q_1 and a subsequent quantizer Q_2, as shown in Fig. 1. Note that this violates the original assumption of uncorrelated quantization noise sources, as the quantization noise generated in Q_2 actually depends on Q_1. We refer to this correlation as *Local Correlation*, since it is manifested at the output of the quantizer and only relates successive quantizers.

B. Noise Propagation and Aggregation

Since we are interested in measuring the quantization noise power at the output of the system, the noise injected needs to be propagated. The quantization noise travels through the *Data Flow Graph (DFG)* of the system as any other signal. Thus, in order to propagate the quantization noise to the output, this needs to be convolved with the transfer function defined from the injection point to the system output. In the case of LTI systems, the mean and variance of the quantization noise just need to be multiplied by a constant parameter defined by the magnitude of the noise transfer function. This parameter corresponds to k_i in Eq. 1. Once propagated, all the noise variances are added together and to the squared accumulation of propagated noise means. The result is the average quantization noise power at the output of the system.

Next, we show that this analytical model produces erroneous results for some particular bitwidth configurations, where the assumption of independence in the variances of the different quantization noises does not hold.

III. Distributed Correlations of Quantization Noises

The reference analytical approach described in the previous section can successfully model local correlations but fails at capturing other types of correlations that can occur between quantization noises. Fig. 2 shows an example DFG exhibiting correlations that are not captured by the reference approach. Based on Eq 1, the average output noise power at O_3 corresponds to:

$$E({e_{O_3}}^2) = \sigma_3 + \sigma_4 + (\mu_3 + \mu_4)^2. \quad (3)$$

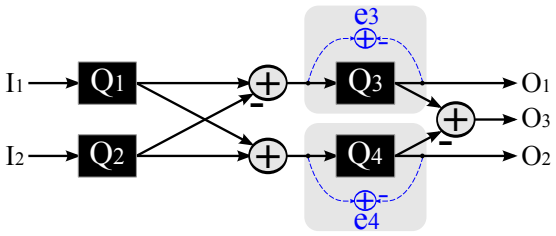


Fig. 2. **Synthetic DFG exhibiting distributed correlation.** The quantizers Q_3 and Q_4 produce correlated quantization noises.

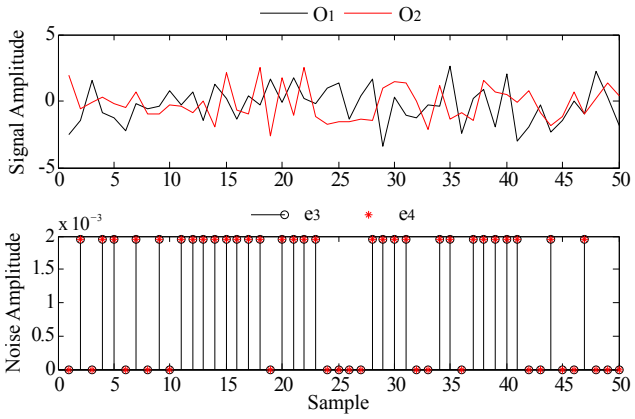


Fig. 3. **Distributed quantization noise correlation.** The upper plot shows the outputs O_1 and O_2 of Fig. 2, which are clearly different. The lower plot shows the quantization noises generated by the Q_3 and Q_4 quantizers, which, despite having different inputs, produce identical quantization errors. Thus, the analytical model that assumes uncorrelated quantization errors will not be accurate in this case. The corresponding bitwidth configurations are indicated in the first line of Table I.

Assuming the first bitwidth configuration of Table I, Eq. 3 produces a 25% underestimation of the quantization noise power. The reference value can be obtained by measuring the actual quantization noise power in a long, and thus precise, Monte Carlo simulation. Such an estimation error comes from the wrong assumption that the quantization error of Q_3 and Q_4 are uncorrelated. Fig. 3 shows that, despite having different inputs, Q_3 and Q_4 produce identical quantization noises, which obviously, are totally correlated. Accordingly, we propose to modify the original analytical expression to account for this correlation effect as follows:

$$E({e_{O_3}^+}^2) = E({e_{O_3}}^2) + 2\gamma_{34}\sqrt{\sigma_3\sigma_4}, \quad (4)$$

where a correlation term is added to Eq. 3, in which γ_{34} corresponds to a correlation coefficient between Q_3 and Q_4. Table I shows how this correlation coefficient varies depending on the bitwidth configurations. The table also shows the average quantization noise power according to Eq. 3, the measured value and the corresponding estimation error. Note that this type of correlation is different from the local correlation described earlier in Section II-A. The new type does not manifest at the output of the quantizer but once the two quantization errors are added in their propagation

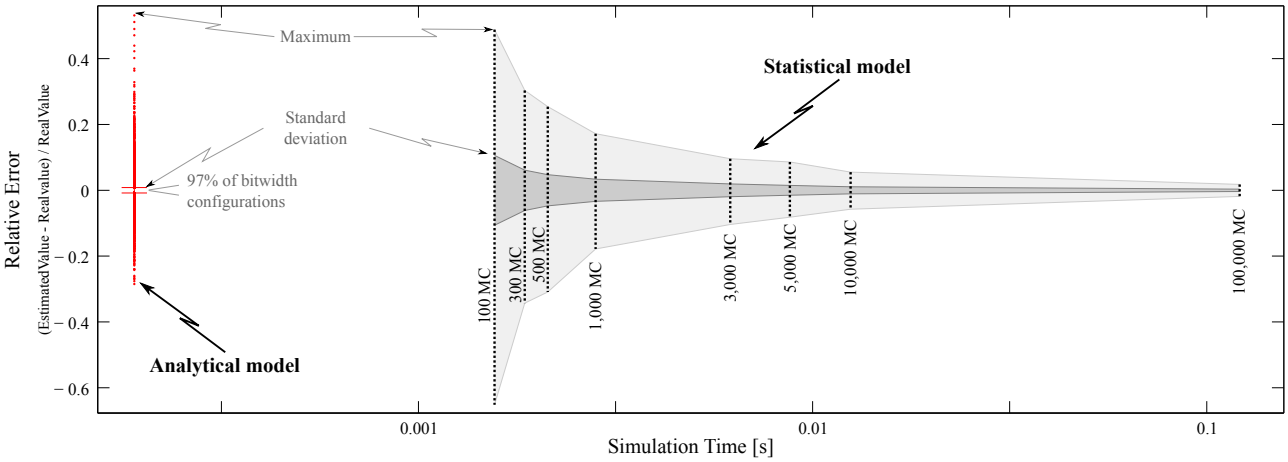


Fig. 4. **Relative error of the Analytical vs Statistical estimation approaches on a 16-point FFT.** The statistical model experiences a consistent reduction of the estimation error as the number of *Monte Carlo (MC)* runs increases. The analytical model produces faster estimates but it can incur relatively high errors, up to 50%. Note that the simulation times reported here are for a single estimation. In practice, even the simplest fixed-point refinement optimization requires a huge number (thousands to millions) of these estimations [10]. Thus short estimation times are crucial to effectively reduce implementation time.

TABLE I

Distributed correlation coefficient. Different bitwidth configurations of the DFG of Fig. 2 result in different correlation coefficients, γ_{34}. Accordingly, the estimation of the average quantization noise power $E(e_{O_3}{}^2)$, described in Eq. 3, introduces a sizable error that decreases with the correlation value.

Bitwith Conf.							
Q_1	Q_2	Q_3	Q_4	γ_{34}	$E(e_{O_3}{}^2)$	Measured	Error
9	9	8	8	1	5.72	7.63	25%
9	10	8	8	0.2	10.96	11.44	4%
9	9	7	8	0.45	20.98	22.88	8%
9	9	7	7	0.2	43.86	45.78	4%
9	9	6	6	0.05	226.97	228.91	1%

path towards the outputs. The latter can happen much later in the DFG, and thus, we refer to this type of correlation as *Distributed Correlation*.

The distributed correlation, illustrated with the synthetic example of Fig. 2, also appears in real systems. For instance, Fig. 5 shows a portion of a 16-point Fast Fourier Transform (FFT) DFG that exhibits distributed correlations between Q_1 and Q_2, Q_3 and Q_4, and Q5 and Q6. Actually, such distributed correlations are responsible for the up to 50% estimation error of the naive analytical noise model, as shown in Fig. 4. Instead, the reference statistical model, based on Monte Carlo simulations, is able to provide tighter estimates at expenses of significantly longer simulation times. Clearly, an alternative estimation method which can offer tighter estimations than the analytical approach in a much faster time than the statistical approach is of great interest.

For this reason, we propose to upgrade the reference analytical approach with the notion of distributed correlations. In order to capture such correlations, the analytical model of Eq. 1 can be completed as follows:

$$E(e_o^{+2}) = E(e_o{}^2) + \sum_{j=1}^{N}\sum_{i=1}^{N} 2\gamma_{ij}\sqrt{\sigma_i\sigma_j}, \qquad (5)$$

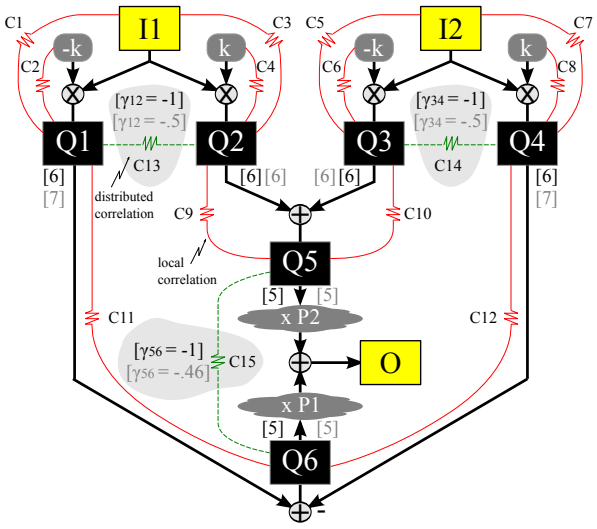


Fig. 5. **Portion of a 16-point FFT DFG that exhibits distributed correlations.** The local correlations are indicated as solid resistors, while the distributed correlations are indicated with dashed resistors. P1 and P2 correspond to two different subDFGs that propagate the quantization noises before being added. This last addition triggers the distributed correlations which are responsible of the estimation error of the analytical model shown in Fig. 4. The values of the distributed correlations under two different bitwidth configurations (black and gray) are indicated between brackets.

where γ_{ij} is the pair-wise distributed correlation coefficient of a quantizer i and a quantizer j. These new correlation coefficients are very difficult to be computed analytically, as their exact value will depend not only on the actual bitwidth configuration of quantizers i and j but also on predecessor quantizers, as shown in the second line of Table I. Accordingly, the following section introduces a solution to discover such correlation coefficients based on selective Monte Carlo simulations.

IV. Our Solution

In this section we describe a new approach to model average quantization noise power capturing the distributed correlations described in the previous section. We propose to combine the strengths of the existing analytical and statistical approaches in a hybrid one. On the one hand, the analytical approach brings fast simulation times but limited precision. On the other hand, the statistical approach brings high precision but slow simulations. Therefore, our solution improves the accuracy of the existing analytical analysis with the result of the statistical simulation of selected parts of the application DFG.

A. Hybrid Noise Estimation

Fig. 6 sketches the proposed hybrid approach. For each input kernel (i.e., an LTI digital signal processing algorithm), an analytical model covering the whole input DFG is generated as proposed by Menard et al. [6]. Besides, a pattern matching algorithm traverses the input DFG searching for subDFGs that are known to include distributed correlations, as the one shown in Fig. 5. These subDFGs are prestored in a library. The algorithm visits all the DFG nodes walking from inputs to outputs. For each node, the matching algorithm produces a small subDFG that only includes the node and immediate neighboring nodes. This subDFG is pattern matched with the DFGs stored in the library. If the matching is not successful, the algorithms moves on to the next node. If there is a partial matching, meaning that the subDFG corresponds to a cluster of a stored DFG, more neighboring nodes are gradually added to the subDFG until the partial matching disappears or there is a perfect match. Every subDFG found in the library, i.e., it includes distributed correlations, is annotated and kept for further analysis.

The library is common to all the kernels and it is built based on the profiling of multiple kernels that have shown large estimations errors of the analytical model, such as the 16-point FFT shown in Fig 4. Out of these kernels, a sensitivity analysis is run in order to detect and isolate the smallest clusters (subDFGs) that are responsible for the correlations. For example, the library used to produce the experiments presented in Section V includes 12 DFGs, where the most complex one contains 16 nodes.

Then, for each correlated subDFG found, a separate statistical model, comprising only the subDFG, is generated. These statistical models can be simulated with random input values to extract the correlation values without any loss of precision. This obeys to the fact that quantization noise is independent from the actual signal value in the targeted LTI systems [8].

Once all the models have been generated for a particular kernel, these need to be evaluated for every different bitwidth configuration. To combine both models, the contribution of each of the correlated subDFGs to the overall power needs to be first removed from the analytically result. Then, the power values obtained in the local statistical simulations need to simply be aggregated and added to remaining analytical noise power.

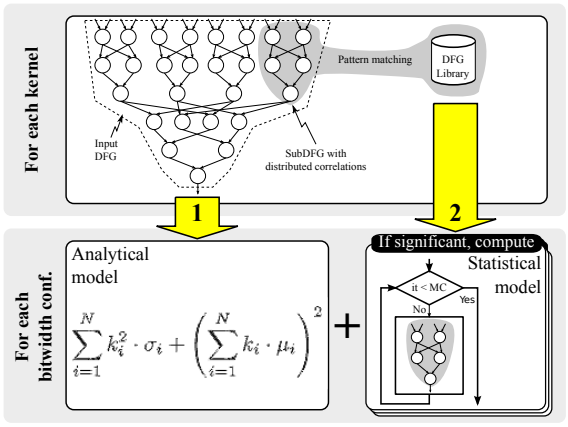


Fig. 6. **Hybrid method.** First, a conventional analytical model of the input DFG is generated (arrow 1). Then the subgraphs of the input DFG that exhibit distributed correlations are carved out and embedded in a statistical model (arrow 2). For each bitwidth configuration, the analytical model is evaluated and corrected by the correlation values provided by a selective simulation (i.e., only if its contribution can be significant) of the statistical models.

However, as illustrated in Fig. 4, the execution of the analytical model is significantly faster than the execution of all the statistical models, and therefore, such a naive hybrid method will suffer from excessively long simulation times. Therefore, in our hybrid approach a metric based on the fast analytical estimation is used to select which statistical models of all the correlated subDFGs need to be executed. Thereby, a great number of statistical simulations can be spared. The selection criteria is detailed next.

B. Conditional Statistical Estimation

A conditional execution of the statistical model is proposed in order to reduce the overall estimation time of our hybrid approach. This basically consists in evaluating, based on an upper bound estimate of the correlation value, whether the execution of the statistical model can affect in a "noticeable" way the current analytical noise estimate. Only then, the statistical model will be executed.

Considering that the maximum value of a correlation coefficient γ_{ij} is 1, the following inequality always holds:

$$2\gamma_{ij}\sqrt{\sigma_i \sigma_j} \leq (\sigma_i + \sigma_j). \tag{6}$$

Note that the analytical estimation of the overall output error, $E({e_o}^2)$, and the noise variances, σ_i and σ_j, are already given by the initial analytical estimation. Therefore, the condition to decide on the execution of the statistical model can be formalized as follows:

$$\frac{\beta E({e_o}^2)}{N} < (\sigma_i + \sigma_j), \tag{7}$$

where N is the number of quantizers contributing to the given output and β is the ratio that considers the potential correlation value to be negligible (e.g., 2% is used in the experimental section). Accordingly, β can be used to bound the maximum estimation error produced by the detected distributed correlations.

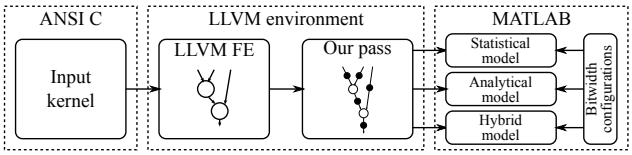


Fig. 7. **Evaluation framework.** The different quantization noise modeling approaches are implemented as a compiler pass in the LLVM compiler framework [11]. A kernel written in ANSI C is input to the system and transformed into the LLVM IR by the LLVM FE. Our pass builds the kernel DFG, runs analysis on it and dumps three MATLAB files: an *Analytical* model based on the algorithm proposed by Menard et al. [6], the *Statistical* model, which can be used as functional reference, and the *Hybrid* model proposed in this paper.

V. Evaluation

In this section the statistical and analytical reference methods to estimate average quantization noise power are evaluated and compared with the hybrid method proposed in Section IV.

A. Methodology

To evaluate the various estimation approaches, we implement three main algorithms: a reference analytical method proposed by Menard et al. [6], the statistical method based on Monte Carlo simulations used by Sung et al. [4] and our new hybrid method. The algorithms are written in C++ and integrated as a compiler pass in the *LLVM* (formerly *Low Level Virtual Machine*) compiler framework [11].

Fig. 7 shows the assembled evaluation framework. First, the LLVM *Front End (FE)* generates the LLVM *Intermediate Representation (IR)* from an input ANSI C description of the kernel of interest. Our new compiler pass builds the DFG of the kernel and inserts quantizer nodes at the inputs and outputs of each operand. Such DFG is used to generate all the models.

The statistical model includes a MATLAB description of the input functionality and a second version which is instrumented with *quantizer* functions that emulate configurable fixed-point precision in each of the signals. Both versions are driven by the same randomly generated input values and the difference between their outputs corresponds to the actual fixed-point error, which is used to compute the average quantization noise power. The generated statistical model function takes as input a vector of bitwidths (bitwidth configurations) and a parameter MC that indicates the number of Monte Carlo simulations used to average the estimate. The higher MC, the more accurate the estimation of the average quantization noise power, but also the longer the estimation time. The statistical model of Fig. 4 illustrates such an effect.

To generate the reference analytical model, the DFG is first traversed from outputs to input to identify locally correlated quantizers (see Fig. 1). For each of these quantizers, noise injection equations based on Eq. 2 are generated. Then, the DFG is once more traversed from each of the quantizer nodes to every kernel output in order to compute their corresponding propagation factor (k_i in Eq. 1). Finally, Eq. 2 is generated for each kernel output. The generated analytical model function takes as input a vector of bitwidths and produces a quantization noise estimate for each of the outputs.

The proposed hybrid analytical model extends the reference analytical model with some extra analysis. Firstly, a new analysis searches for particular subDFGs in the input DFG that are known to produce distributed correlations. The targeted subDFGs are stored in a library, which can be extended once new correlated patterns are found. Then, for each correlated subDFG a new function with the corresponding statistical model is generated. Finally, the global analytical model and the partial statistical model are integrated in a single function with the same interface as in the previous approaches (refer to Section IV for further details on the hybrid approach).

The benchmarks used in our experiments are described in Table II. Gauss3x3 is a 3×3 pixels image smoothing filter; FIR31 is a 31-taps raised-cosine *Finite-Impulse Response (FIR)* filter; DCT8x8 is a two dimensional 8×8 pixels *Discrete Cosine Transform (DCT)*, and FFT16, FFT32, FFT64 and FFT128 are 16-, 32-, 64- and 128-point *Fast Fourier Transforms (FFT)* , respectively. All the benchmarks are fully unrolled to exhibit as many quantizers as possible, which is equivalent to full parallel implementations.

We run the experiments on a dual-core 2.4GHz processor with 4 GB of memory, using the Mac OS X 10.7 distribution and MATLAB R2012a.

B. Results

Table II shows the performance in simulation time (T) and accuracy, in terms of standard deviation (SD) and maximum value (M) of the percentage of estimation error, of the different noise estimation approaches for the selected benchmarks. We report two instances of the statistical model, one representative for short and inaccurate estimations ($MC = 500$), and a second one for long and accurate estimations ($MC = 10{,}000$). We also report two instances of our hybrid method, one where the conditional analytical estimation is disabled ($\beta = 0$) and one where it is enabled ($\beta = 0.02$). The error power reference is generated with the statistical model configured with $MC = 100{,}000$; a value 10 times higher than in the longest evaluation of the statistical approach. For a given benchmark, the same $10{,}000$ random bitwidth configurations are generated for each of the estimation approaches.

To better understand the references, Fig. 4 illustrates the properties of the reference analytical and statistical estimation approaches for the FFT16 benchmark. The analytical model is significantly faster than any of the configurations of the statistical model. However, such an analytical model produces a rather large estimation error for particular bitwidth configurations. In contrast, the statistical model consistently reduces the estimation error with the increase of Monte Carlo iterations.

The hybrid method proposed in this paper is not able to find any correlation subDFG in the first 3 benchmarks, and thus, it behaves exactly like the reference analytical method. For the other benchmarks, the statistical modeling of correlated subDFGs helps increasing the estimation accuracy at expenses of estimation time. Notice, that our conditional analytical estimation can successfully reduce simulation time while marginally affecting the estimation error.

TABLE II

Benchmarks. Each benchmark is characterized by the number of quantizers (Q). The different estimation approaches are characterized by the standard deviation (SD) and maximum value (M) of the relative estimation error, and the estimation time (T) relative to the *Analytical* approach. We include two instances of the *Statistical* approach for different number of Monte Carlo simulation runs (MC) and two instances of our *Hybrid* approach for two different values of β.

		Stat. (MC = 500)			Stat. (MC = 10,000)			Analytical [6]		Hybrid ($\beta = 0$)			Hybrid ($\beta = 0.02$)		
Name	Q	SD	M	T	SD	M	T	SD	M	SD	M	T	SD	M	T
Gaus3x3	23	2.52	10.74	10.85	0.59	2.94	44.00	0.73	5.38	0.73	5.38	1.00	0.73	5.38	1.00
FIR31	91	1.63	8.21	10.80	0.38	1.61	66.74	0.19	1.31	0.19	1.31	1.00	0.19	1.31	1.00
DCT8x8	800	4.73	29.09	2.97	1.11	6.08	16.35	0.41	8.67	0.41	8.67	1.00	0.41	8.67	1.00
FFT16	225	4.76	30.89	10.00	1.12	5.89	44.91	0.88	52.08	0.38	13.16	9.48	0.38	13.43	2.81
FFT32	589	4.94	31.64	3.76	1.16	6.31	14.43	0.67	33.79	0.35	13.56	7.89	0.36	13.72	2.63
FFT64	1485	5.05	32.54	1.18	1.18	6.74	6.18	0.57	17.38	0.29	8.19	4.80	0.31	8.31	2.09
FFT128	3597	5.10	31.87	0.37	1.19	6.82	2.14	0.50	8.50	0.28	5.31	3.91	0.29	5.34	1.91
Geometric Mean		3.82	20.95	3.49	0.89	4.51	16.76	0.51	9.72	0.34	6.47	2.82	0.35	6.52	1.62

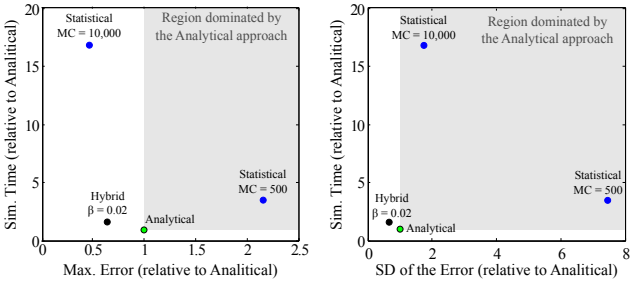


Fig. 8. **Accuracy vs speed Pareto space.** The maximum error and the standard deviation error of the evaluated noise estimation approaches is plotted with their corresponding estimation time. The proposed method is always Pareto optimal can even dominate very time-consuming Monte Carlo simulations.

Fig. 8 plots the average results of the different estimation methods in a Pareto space. On the left-hand side, the Pareto space is defined by the simulation time and the maximum estimation error, both normalized with respect to the values of the analytical method. On the right-hand side, the error axes represents the normalized standard deviation of the estimation error. Note that while the reference analytical method dominates the statistical methods for standard deviation error (SD), this is not the case for maximum estimation error (M). Accordingly, whenever the error of the analytical model is too high for a particular application, the designer will have to fall back to the statistical model and pay a huge simulation time penalty. Interestingly, our proposed hybrid method represents a new Pareto solution which is significantly faster than the statistical method and offers a higher estimation accuracy than the analytical method.

Particularly, our results show that the hybrid method presented in this paper is able to reduce significantly the maximum estimation error of the analytical approach, up to a factor 4. Also, the average standard deviation error of our approach is the smallest of all the studied methods. This is achieved with simulation times that are, in average, 1.62 times slower than in the analytical approach, but more than 10 times faster than in a statistical approach of similar accuracy.

VI. Conclusions

This paper describes a novel approach to monitor the quantization error throughout the fixed-point refinement of LTE systems, which is known to take an important share of the overall design time. Accordingly, we show that our hybrid algorithm is able to improve significantly the accuracy of the existing analytical method, up to a factor 4, while still being more than 10 times faster than an estimation based on Monte Carlo simulations. Therefore, the proposed approach can be used to reduce significantly the design time spent in fixed-point refinement without compromising its optimality.

VII. Acknowledgments

This work was partially supported by the European Union program FP7-PEOPLE-2011-IEF, under Project 299405.

References

[1] T. Grötker, E. Multhaup, and O. Mauss, "Evaluation of HW/SW tradeoffs using behavioral synthesis," *Proceedings of the International Conference on Signal Processing Applications and Technology*, 1996.

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

[3] O. Sarbishei and K. Radecka, "On the fixed-point accuracy analysis and optimization of FFT units with CORDIC multipliers," in *20th IEEE Symposium on Computer Arithmetic*. IEEE, 2011, pp. 62–69.

[4] W. Sung and K.I. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," *IEEETSP*, vol. 43, no. 12, pp. 3087–3090, 1995.

[5] K.I. Kum and W. Sung, "Combined word-length optimization and high-level synthesis of digital signal processing systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 8, pp. 921–930, 2001.

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

[7] C. Shi and R.W. Brodersen, "Automated fixed-point data-type optimization tool for signal processing and communication systems," in *Proceedings of the 41st Design Automation Conference*. ACM, 2004, pp. 478–483.

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

[9] G.A. Constantinides, P.Y.K. Cheung, and W. Luk, "Truncation noise in fixed-point SFGs," *Electronics Letters*, vol. 35, no. 23, pp. 2012–2014, 1999.

[10] M.A. Cantin, Y. Savaria, and P. Lavoie, "A comparison of automatic word length optimization procedures," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2002, vol. 2, pp. II–612.

[11] C. Lattner and V. Adve, "LLVM: A compilation framework for lifelong program analysis & transformation," in *International Symposium on Code Generation and Optimization*. IEEE, 2004, pp. 75–86.