Analysis of Finite Word-Length Effects in Fixed-Point Systems

Daniel Ménard, Gabriel Caffarena, Juan Antonio Lopez, David Novo, Olivier Sentieys

Analysis of Finite Word-Length Effects in Fixed-Point Systems

D. Menard[*], G. Caffarena[†], J.A. Lopez[‡],
D. Novo[§], O. Sentieys[¶]

February 4, 2019

Contents

[*]Univ Rennes, INSA Rennes, IETR, Rennes, France
[†]University CEU-San Pablo, Madrid, Spain
[‡]ETSIT, Universidad Politécnica de Madrid, Spain
[§]LIRMM, Université de Montpellier, CNRS, Montpellier, France
[¶]Univ Rennes, Inria, Rennes, France

Abstract

Systems based on fixed-point arithmetic, when carefully designed, seem to behave as their infinite precision analogues. Most often, however, this is only a macroscopic impression: finite word-lengths inevitably approximate the reference behavior introducing quantization errors, and confine the macroscopic correspondence to a restricted range of input values. Understanding these differences is crucial to design optimized fixed-point implementations that will behave "as expected" upon deployment. Thus, in this chapter, we survey the main approaches proposed in literature to model the impact of finite precision in fixed-point systems. In particular, we focus on the rounding errors introduced after reducing the number of least-significant bits in signals and coefficients during the so-called quantization process.

1 Introduction

The use of *fixed-point (FxP)* arithmetic is widespread in computing systems. Demanding applications often force computing systems to specialize their hardware and software architectures to reach the required levels of efficiency (in terms of energy consumption, execution speed, etc.). In such cases, the use of fixed-point arithmetic is usually not negotiable. Yet, the cost benefits of fixed-point arithmetic are not for free and can only be reached through an elaborated design methodology able to restrain finite word-length – or quantization – effects.

Digital systems are invariably subject to nonidealities derived from their finite precision arithmetic. A digital operator (e.g., an adder or a multiplier) imposes a limited number of bits (i.e., word-length) upon its inputs and outputs. As a result, the values produced by such an operator suffer from (small) deviations with respect to the values produced by its "equivalent" (infinite precision) mathematical operation (e.g., the addition or the multiplication). The more the bits allocated the smaller the deviation – or quantization error – but also the larger, the slower and the more energy hungry the operator. The so-called word-length optimization – or quantization – process determines the word-length of every signal (and corresponding operations) in a targeted algorithm. Accordingly, the best possible quantization process needs to select the set of word-lengths leading to the cheapest implementation while bounding the precision loss to a level that is tolerable by the application in hand. The latter can formally be defined as the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & C(\mathbf{w}) \\ \text{subject to} \quad & D(\mathbf{w}) \leq \Omega, \end{aligned} \tag{1}$$

where $\mathbf{w}$ is a vector containing the word-lengths of every signal, $C(\cdot)$ is a cost function that propagates variations in word-lengths to design objectives such as energy consumption, $D(\cdot)$ computes the degradation in precision caused by a particular $\mathbf{w}$ and Ω represents the maximum precision loss tolerable by the application.

From a methodological perspective, the word-length optimization process can be approached in two consecutive steps: (1) range selection and (2) precision optimization. The *range selection* step defines the left hand limit – or *Most-Significant Bit (MSB)* – and the subsequent *precision optimization* step fixes the right hand limit – or *Least-Significant Bit (LSB)* – of each word-length. Typically, the range selection step is designed to avoid overflow errors altogether, and therefore, the precision optimization step becomes the sole responsible for precision loss. Figure 1 gives a pictorial impression of the word-length optimization process and divides the precision optimization step into four interacting components, namely the optimization engine, the cost estimation, the constraint selection and the error estimation.

- The *optimization engine* basically consists of an algorithm that iteratively converges to the best word-length assignment. It has been shown that the constraint space is non-convex in

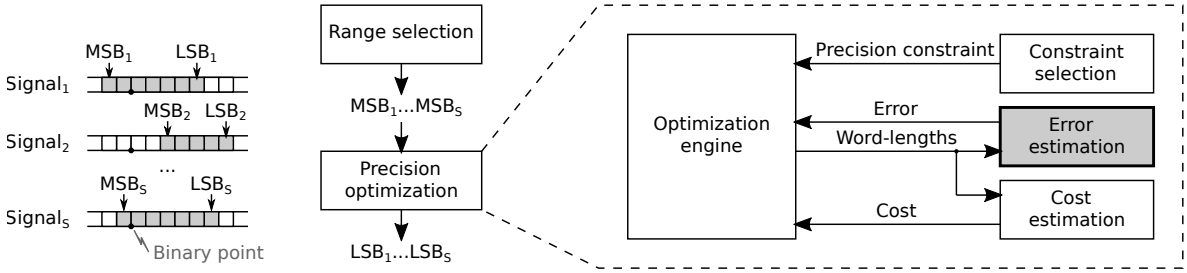


Figure 1: Basic components of a word-length optimizaton process

nature [29] – it is actually possible to have a lower quantization error at a system output by reducing the word-length at an internal node –, and that the optimization problem is NP-hard [35]. Accordingly, existing practical approaches are of a heuristic nature [32, 21, 22].

- A precise *cost estimation* of each word-length assignment hypothesis leads to impractical optimization times as such heuristic optimization algorithms involve a great number of cost and error evaluations. Instead, word-length optimization processes use fast abstract cost models, such as the hardware cost library introduced in the chapter [132] of this book or the fast models proposed by Clarke et al. [28] to estimate the power consumed in the arithmetic components and routing wires.

- The *precision constraint selection* block is responsible of reducing the abstract sentence "the maximum precision loss tolerable by the application" into a magnitude that can be measured by the error estimation. Practical examples have been proposed for audio [103] or wireless applications [109].

- Existing approaches for *error estimation* can be divided into simulation-based and analytical methods. Simulation-based methods are suitable for any type of application but are generally very slow. Alternatively, analytical error estimation methods can be significantly faster but often restrict the domain of application (e.g., only linear time-invariant systems [32]). There are also hybrid methods [122] that aim at combining the benefits of each method.

While the chapter presented in [132] covers in breadth most of the blocks in Figure 1, this chapter takes a complementary in-depth approach and focuses on arguably the most important block in the word-length optimization process: the error estimation. The latter is crucial to ensure correctly behaving fixed-point systems and has received considerable attention in the research literature. Thus, in this chapter, we survey the main approaches proposed to model quantization errors. To understand their similarities and differences, we present a classification of the reviewed approaches based on their assumptions and coverage. We believe that this chapter will shed some light on the word-length optimization process as a whole and help readers choose the most convenient available approach to model quantization errors in their word-length optimization process.

The rest of the chapter is organized as follows. Section 2 introduces the main concepts regarding quantization. The next section deals with signal quantization. Noise metrics and both simulation-based and analytical techniques for the evaluation of quantization noise are explained. Regarding the analytical evaluation, this covers both the estimation of noise power and noise bound. Section 4 addresses the quantization of coefficients. The different meausurement parameters used to evaluate coefficient quantization are explained, with special enphasis on the use of the L_2-sensitivity. System stability is described in section 5, again focusing on simulation-based and analytical approaches. Finally, a summary is presented in the last section.

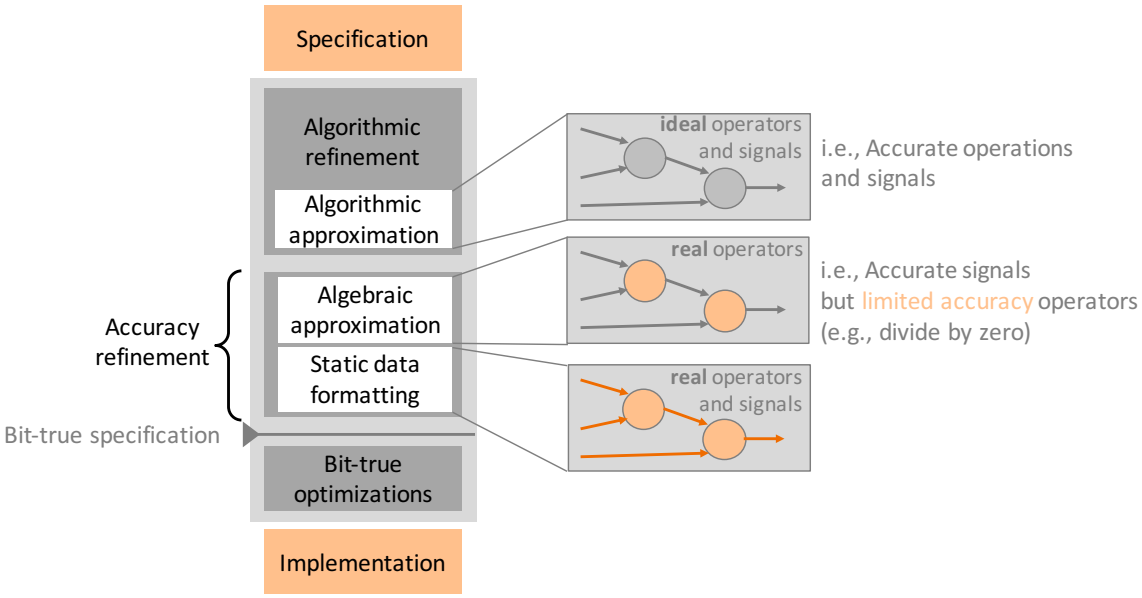


Figure 2: Basic DSP design flow

2 Background

A typical *Digital Signal Processing (DSP)* design flow begins with a design specification and follows a number of steps to produce a satisfactory implementation as illustrated in Figure 2. The original specification serves as a functional reference and is typically implemented in frameworks that prioritize software productivity, such as MATLAB, in floating-point or double precision. For instance to illustrate, such a specification can include a 64-point *Discrete Fourier Transform (DFT)*. Firstly, a skillful designer will reduce the algorithmic complexity in the *algorithmic refinement* step. The DFT matrix can be factorized into products of sparse factors (i.e., Fast Fourier Transform), which reduces the complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. Additionally, the algorithmic refinement step can make use of approximations to further reduce the complexity – e.g., the *Maximum Likelihood (ML)* detector is approximated by a near-ML detector [109]. Once the algorithm structure is fixed, operators and signals are defined in the subsequent *algebraic transformation* and *static data formatting* steps, respectively. An algebraic approximation can for instance reduce a reciprocal square root operator to a scaled linear function [109]. Finally, the static data formatting step is the responsible of finalizing the bit-true specification that will constrain all succeeding (bit-true) optimizations, such as loop transformations, resource binding, scheduling, etc.

Algorithmic and algebraic approximations are integrating parts of what is known as *approximate computing* [107]. Instead, data formatting is equivalent to the word-length optimization process introduced in the previous section. Although some prior work targets implementations that do not add quantization error to those of the inputs [9, 130, 84], *lossy* static data formatting [34] – i.e., reduction of implementation cost by introducing additional quantization noise in intermediate nodes – is the common practice and the main focus of this chapter.

2.1 Floating-Point vs. Fixed-Point Arithmetic

The IEEE-754 standard [60] for *floating-point (FlP)* arithmetic – particularly the 64 bit double-precision format – is commonly used in implementations requiring high mathematical precision. However, many applications tolerate the use of less precise arithmetic modules in both FxP [34,

120] and non-standard FlP [51] formats. As introduced in Chapter [132], the FlP format represents numbers by means of two variables: an exponent e and a mantissa m. Given the pair (m,e), the value of the represented FlP number, V_{FlP}, is

$$V_{FlP} = m \cdot 2^e. \tag{2}$$

The combined use of mantissa and exponent provides the finest level of scaling: each number includes its own scaling factor. Thereby, FlP digital systems can effectively operate numbers with a very wide dynamic range. However, FlP arithmetic often involves overheads in terms of area, delay and energy consumption. Firstly, FlP requires wider bit-widths than FxP arithmetic to operate with equivalent precision on variables with low to moderate dynamic range [57], which is the typical case in most applications. Furthermore, FlP operators are more complex as they implement in hardware the alignment of the fractional point of the operands and the normalization of the output besides the actual operator.

Alternatively, FxP arithmetic constrains the exponent e to be a design time constant. Equation (2) remains valid but only the mantissa m changes at run time – and thus needs to be stored in memory. Accordingly, describing an implementation employing FxP arithmetic is more complex and tedious as the designer is responsible of handling explicitly in the source code the scaling of variables.

2.2 Finite Word-Length Effects

Quantized systems suffer from two types of errors: *overflow* and *precision* errors. On the one hand, overflow errors result from variable values growing beyond the limits of the word-length. They are related to the lack of scaling and saturation and wrap-around [119, 97, 116] are the most common techniques used to handle them at the operator output. Saturation employs extra hardware to detect and reduce overflow error. Instead, wrap-around is hardware-free but leads to intolerably huge errors in underdimensioned word-lengths. On the other hand, precision errors are due to the unavoidable limited precision of quantized digital implementations [119, 97, 116]. Rounding and truncation are the most common techniques used to handle precision errors at the operator output. Rounding employs extra hardware to reduce the maximum error magnitude resulting from the removal of LSBs. Instead, truncation is hardware-free but often accumulates larger precision errors. The technique leading to the most implementation is application dependent: even though rounding requires more complex operators, they can generally operate shorter word-lengths to achieve the same precision error as truncation [98].

The limited precision effects of the DSP realizations have been studied extensively since the raise of digital systems, particularly in *Linear Time Invariant (LTI)* systems [119, 97, 116]. They are commonly divided in four different types: round-off noise, coefficient quantization, limit cycles and system stability. *Round-Off Noise (RON)* refers to the probabilistic deviation of the results of a quantized implementation with respect to the error-free reference [119, 97, 116]. *Coefficient Quantization (CQ)* refers to the deterministic deviation of the parameters of the transfer function [71, 119, 97]. *Limit Cycles (LC)* are the parasitic oscillations that appear in quantized system under constant or zero inputs due to the propagation of the quantization errors through feedback loops [27, 119]. Finally, in the case of digital filters, the coefficient quantization modifies the position of the poles of the transfer function, which might jeopardize the system stability when approached carelessly [110]. Table 1 summarizes the classification of these effects attending to linearity and whether they result from the quantization of signals or coefficients.

RON is the prominent finite precision effect during normal operation of FxP systems [71, 119, 97, 116]. It introduces stochastic variations around the system's nominal operation point. Complementary, CQ effects modify the actual nominal operation point of the system and can lead to instability when such deviation is not carefully conducted. While RON and CQ effects apply to any FxP system, LCs effects are only relevant to particular types of systems (e.g., DSP filters) as they are the result of correlated quantization errors in feedback loops [119, 116]. For this reason,

Table 1: Classification of the finite WL quantization effects

Type of effect	Quantization object	Name of effect
Linear	Signals	Round-Off Noise (RON) (Section III)
	Coefficients	Coefficient Quantization (CQ) (Section IV)
Nonlinear	Signals	Limit Cycle Oscillations
	Coefficients	System Instability

in this chapter we focus mainly on RON (most of Section 3) and CQ effects (Section 4) while also covering LCs for the sake of completeness but in much less detail (end of Section 3).

3 Effect of Signal Quantization

Finite precision arithmetic leads to unavoidable deviations of the finite precision values from the infinite precision ones. Such deviations, due to signal quantizations, modify the quality of the application output. Thus, they must be evaluated and maintained within reasonable bounds. In most cases these deviations are accurately modeled as additive white noise, or quantization noise. The quantization noise can be evaluated through analytical or fixed-point simulation based approaches. In the case of analytical approaches, a mathematical expression of a metric is determined. Computing an expression of an quality metric for every kind of application is generally an issue. Thus, the quality degradations are not analyzed directly in the quantization process, but an intermediate metric measuring the fixed-point accuracy is used instead.

Word-length optimization is split into two main steps. Firstly, a computational accuracy constraint is determined according to application quality and, secondly, the word-length optimization is carried out using this constraint. Interestingly, fixed-point simulation approaches enable the direct evaluation of the effect of quantization on application quality. But, in many cases, an intermediate accuracy metric is used because less samples are required to estimate this metric in contrast to directly computing or simulating application quality under quantization effects.

The different approaches available to analyze quantization noise effects that are covered in this section are displayed in Fig. 3. The techniques are first divided into the three main major groups: simulation-based, analytical and mixed (that combines the two previous ones) approaches. The graph include all techniques covered in the subsequent subsections and also the main related publications.

Fig. 4 shows the main classification of systems used by the different techniques devoted to RON evaluation: LTI systems, *smooth* systems and *all* systems. *Smooth* systems are those whose operations are differentiable and can be linearized without commiting a significant error. This classification also distinguishes between recursive systems – systems with loops or cyclic – and non-recursive systems – systems without loops or acyclic. The different regions displayed in the graph are related to different techniques that are only able to handle a particular type of systems.

Section 3.1 introduces the different noise metrics used. Section 3.2 covers the analytical evaluation of the quantization noise effect, embracing both the noise power and noise bound

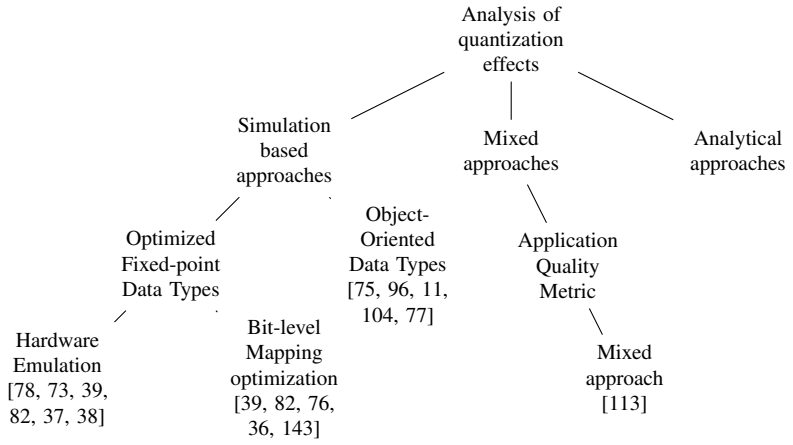


Figure 3: Classification of the different approaches to analyze the quantization noise effects

computation. Then the techniques based on fixed-point simulation and the hybrid techniques are presented in Section 3.3.

3.1 Error Metrics

Different metrics can be used to measure the accuracy of a fixed-point realization. This accuracy can be evaluated through the bounds of the quantization errors [43, 2], the number of significant bits [24], or the power of the quantization noise [102, 126, 18]. The shape of the power spectral density (PSD) of the quantization noise is used as metric in [7] or in [31] for the case of digital filters. In [20], a more complex metric able to handle several models is proposed.

Regarding the metric that computes the bounds of the quantization errors, the maximum deviation between the exact value and the finite precision value is determined. This metric is used for critical systems when it is necesary to ensure that the error will not surpass a maximum deviation. In this case, the final quality has to be numerically validated.

As for the noise power computation, the error is modeled as a noise, and the second order moment is computed. This metric analyzes the dispersion of the finite precision values around the exact value and the mean behaviour of the error. The noise power metric is used in applications which tolerate sporadic high-value errors that do not affect the overall quality. In this case, the system design is based on a trade-off between application quality and implementation cost.

3.2 Analytical Evaluation of the Round-Off Noise

The aim of analytical approaches is to determine a mathematical expression of the fixed-point error metric. The error metric function depends on the word-length of the different data inside the

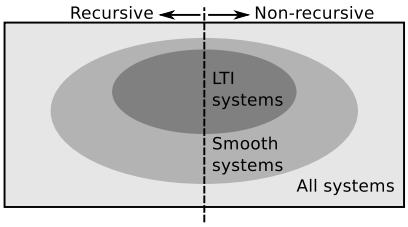


Figure 4: Classification of systems targeted by RON evaluation techniques

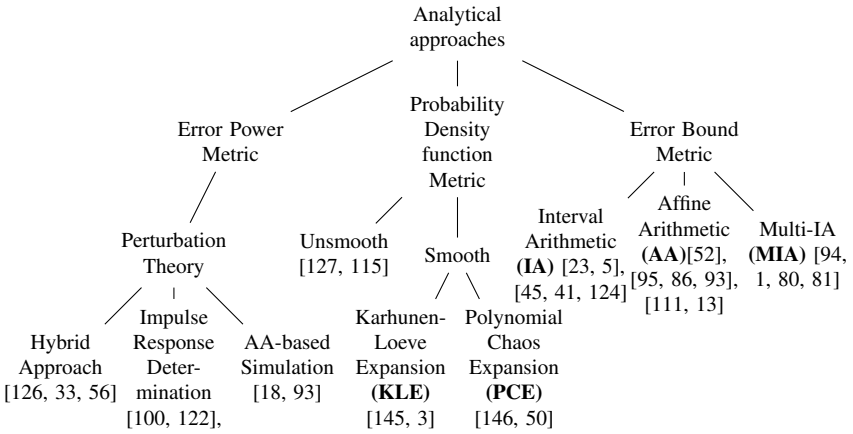


Figure 5: Classification of the different analytical approaches to analyze the quantization noise effects

application. The main advantage of these approaches is the short time required for the evaluation of the accuracy metric for a given set of word-lengths. The time required to generate this analytical function can be more or less important but this process is done only once, before the optimization process. Then, each evaluation of the accuracy metric for a given WL sets corresponds to the computation of a mathematical expression. The main drawback of these analytical approaches is that they do not support all kinds of systems. Figure 5 depicts a classification of existing analytical approaches to analyze the quantization noise effects. This classification depends on the type of metric used (bound, power or probability density function), on the smooth/unsmooth nature of the noise, and on the technique used. In this section, we review the different analytical approaches for computing: RON bounds, RON power, and the effect of RON on any quality metric in the presence of unsmooth operators.

3.2.1 Quantization Noise Bounds

There are a number of techniques and methods that have been suggested in the literature to measure the bounds of the quantization noise. Since the numerical techniques typically lead to exceedingly long computation times, different alternatives have been proposed to obtain results faster.

Table 2 shows the most relevant techniques related to the evaluation of noise bounds. The first column indicates the name of the technique. The second column displays the main characteristics of the technique, while the third column shows particular features of the cited approaches. The next three columns contain information about the type of systems that the approaches can be applied to (all, polynomial, based on smooth operations and LTI systems), the existence of loops and the computational speed of the approach.

The analytical techniques used to evaluate the noise bounds can be classified in two major groups: (i) interval-based computation (Interval Arithmetic (IA), Multi-IA (MIA), Affine Arithmetic (AA) and satisfiability modulo theory) and (ii) polynomial representation with interval remainders (sensitivity analysis and Arithmetic Transformations (AT)). Principal techniques are described in the following paragraphs.

Interval-based computations

In the last decade, *interval-based computations* have emerged as an alternative to simulation-based techniques. A high number of simulations are required in order to cover a significant set of possible values of the inputs, so traditional simulation-based techniques imply very long computation times. As an alternative, interval-based methods have been suggested to speedup

Table 2: Techniques for the evaluation of the quantization noise bounds

General features	Particular features	System	Loops	Speed	References
Interval Arithmetic and Range propagation					
Forward-Backward Propagation: Reduces some overestimation but the results are still oversized.	Combines three methods to reduce oversize: number of bits, range of each variable, and logic value of each bit. Integrated in the *Bitwise* tool.	All	No	Fast	Stephenson [130]
	Inspired by [130], combines constraint propagation, simulation, range evaluation and slack analysis. Integrated in the *Précis* tool.	All	No	Medium	Chang [23]
Forward propagation	User annotations. Integrated in the *Match* compiler and the *AccelFPGA* tool.	All	No	Medium	Nayak [108] Banerjee [4, 5]
	Precision analysis stage based on error propagation.	All	No	Medium	Doi [45]
IA overestimation reduction	Integrated in the *Gappa* tool.	All	No		De Dinechin [42]
Multi-Interval Arithmetic					
More accurate results than IA, but still oversized (splitting does not solve the dependency problem)	Evaluates the propagation of the intervals due to the quantization operations through the feedback loops. Integrated in the *Abaco* set of tools.	LTI	Yes	Very fast	Lopez [94]
	Symbolic Noise Analysis (SNA) by splitting the intervals. They take into account the probabilities in the propagation of the error.	LTI	No	Fast	Ahmadi [1]
	Based on the Satisfiability Modulo Theory (SMT) the intervals are iterativelly reduced by splitting them and selecting which parts are valid.	All	Yes	Medium	Kinsman [79], [80, 81]
Affine Arithmetic					
More accurate results than IA and MIA	It provides guaranteed bounds.	LTI	No	Very fast	Fang [53]
	It provides estimates of the bounds. Integrated in the *Abaco* tool	LTI	Yes	Very fast	Lopez [92, 95, 93]
	It provides guaranteed bounds. Implemented on *Minibit* and *Lengthfinder* tools	Polynomial	No	Medium Fast	Lee [84]
Sensivity Analysis					
Based in automatic differentiation. It provides fast results.	It computes the maximum deviation for each noise source and performs propagation by means of signal derivatives. It provides guaranteed bounds, yet oversized.	Smooth	No	Very fast	Gaffar [58]
Arithmetic transformations					
Analytical approach that follows a similar concept to the Taylor Models. AT provides a canonical representation of the propagation functions	The output is described as a polynomial function of the inputs. The WLs are optimized by considering the imprecision allowed for the quantizations	Polynomial	No	Fast	Pang [112, 125, 124]
	AA is used for range analysis, and (AT, IA) for WL analysis and optimization. Small overestimation.	LTI Polynomial	Yes	Very fast Fast	Sarbishei [125, 124]

the computation process. The results are obtained much faster, but they have to deal with the continuous growth of the intervals (oversizing) through the sequence of operations. Thus, these techniques are restricted to a limited subset of systems (mostly LTI or quasi-LTI), or combined with other techniques to reduce the oversize.

The most classical approach is the computation using interval arithmetic (IA), also called *forward propagation*, *value propagation* or *range propagation* techniques. Given the ranges of the inputs of a system, represented by intervals, IA computes the guaranteed ranges of the outputs. The main drawback of these techniques is the so-called *dependency problem*, which is produced when the same variable is used in several places within the algorithms under analysis, since IA is not able to track dependency between variables, ranges are overestimated. To alleviate this situation, some authors have suggested splitting the intervals in a number of sections, generating a *Multi-IA* approach.

One of the earliest work that applied value propagation to the computation of the noise bounds was developed by Stephenson *et al.* in the *Bitwise* project [130]. They perform forward and backward range propagation, and combine three different types of analysis to optimize the WLs with guaranteed accuracy: analysis of the number of bits, the ranges of the operands, and the logic value of each bit. The analysis of the number of bits provides larger WLs than the analysis of ranges, but limits the LSB of the result. In combination with backward propagation, the evaluation of the logic values of the operands enables some optimization, but it is not significant in the general case. Since the oversizing of these techniques rapidly increases along the sequence of operations, this approach does not provide practical results in complex systems. However, it provides fast and guaranteed results for smaller blocks.

Chang *et al.* have applied a similar approach in the *Précis* tool [23]. By including fixed-point annotations in Matlab code, they perform fixed-point simulation, range analysis, forward and backward propagation, and slack analysis. The annotations are based on the routine *fixp*, which allows modelling different integer and fractional WLs, as well as overflow and underflow quantization strategies. They indicate that the combined application of range analysis (MSB) and propagation analysis (LSB) provides accurate WLs, and that the propagation based on the number of bits is more conservative than range analysis for the MSBs. Slack analysis uses the difference between these two results to provide an ordered list of signals that provide better results when their LSBs are optimized [23].

Nayak [108] and Banerjee *et al.*, [4, 5] have applied the propagation techniques to the computation of the noise bounds. They have developed an automatic quantization environment that has been included in the *Match* project and the *AccelFPGA* tool.

In [45], Doi *et al.*, present a WL optimization method that estimates the optimum WLs using noise propagation. They propagate the noise ranges using IA, and apply it in combination with a nonlinear programming solver to estimate the optimum WLs in LTI blocks without loops. Due to the oversizing of the interval-based computations, the bounds provided in this process are conservative in most cases, but the difference with the optimum result is not significant in blocks without loops.

The *Gappa* tool [42, 41] uses a different approach to deal with the oversizing associated to the interval computations. It creates a set of theorems to rewrite the most common expressions into similar ones that are less affected by the correlations in the interval computations. This approach provides guaranteed and accurate results, but up to now its application is limited to systems without loops and branches [41], and requires a very good knowledge of the target system [42].

Multi-IA (MIA) has also been applied by several authors to reduce the width of the bounds of the quantization noise. In [94], the authors suggest a method to reduce the overestimation of IA and use it to provide refined bounds in the impulse response and the transfer function of an IIR filter. Although MIA provides less conservative bounds than IA, MIA does not solve the dependency problem and is therefore not a good option for systems with loops [95].

The *Symbolic Noise Analysis* (SNA) method presented in [1] splits the noise intervals into smaller parts and performs IA propagation of each part. At the output, intervals are combined

according to their probabilities to provide the histogram of the output noise. When there is small or no oversizing, this approach provides accurate estimates of the PDF of the output noise. However, in the general case, this only provides bounds associated to each part, and less conservative global bounds than IA or range propagation methods.

Kinsman and Nicolici [80, 81] propose to use *Satisfiability Modulo Theory* (SMT). This approach initially performs IA propagation of the values of all the signals and noise sources, and provides an initial (conservative) estimate of the bounds at the output. After that, all the sources are successively split using the bisection method to provide less conservative ranges in each iteration. The process finishes after reaching a given constraint or when all the intervals have zero width (degenerated intervals). The authors indicate that this method is particularly useful in presence of discontinuities (such as in systems with divisions or inverse functions) and that it provides more accurate results than AA in non-linear systems [79]. In later work, the authors have generalized this idea to handle floating- and fixed-point descriptions using the same solver [80] and have introduced vectors to reduce the amount of terms in the splitting process [81].

Affine Arithmetic (AA) [131] was proposed to optimize the bounds of signals and noise sources in LTI fixed-point realizations [53]. The authors propose to apply AA for feed-forward systems to obtain guaranteed bounds and also to obtain a practical estimation based on a confidence interval. Moreover, an iterative method is proposed for systems with feedback and is proved to always converge although the bounds are overestimated. A more detailed analysis about the application of AA to characterize quantized LTI systems has been carried out in [92, 95, 93]. The authors have evaluated the source and propagation models of AA in fixed-point LTI systems with feedback loops, and have concluded that AA propagates the exact results in systems described by sequences of affine operations (i.e., LTI systems). In [92] and [95], they propose a variation of the description of the quantization operations of AA that provides more accurate estimates of the noise bounds. A comparison between IA, MIA, AA and the proposed approach shows that IA and MIA are affected by the dependency problem in most LTI systems with feedback loops (whenever the filter has complex poles), and do not provide useful results [95]. In [93], the expressions for the generation of the affine sources, the propagation of the noise terms, and the computation of the output results are provided. Although they are oriented to the computation of the MSE statistics, the derivation of the corresponding expressions to obtain the minimum guaranteed bounds is very easily obtained.

AA has also been suggested in combination with Adaptive Simulated Annealing (ASA) to perform WL optimization of fixed-point systems without feedback loops in the tool *Minibit* [85].

Polynomial representations with interval remainders

The *polynomial representations with interval remainders* are based on the perturbation theory and follow a similar idea to the Taylor Models. They perform a polynomial Taylor series decomposition and the smallest uncertainties can be merged in one or more terms, or simply they can be neglected. These approaches have been suggested, in particular in recent years, to perform efficient evaluation of polynomial sequences of operations.

Perturbation theory is based on a Taylor series decomposition of a given order and can include intervals to provide guaranteed bounds of the results. This idea was first presented by Wadekar and Parker [140], but the implementation details of the computation were not given. The most relevant contributions are those based on sensitivity analysis (using first-order derivatives) and arithmetic transformations (canonical polynomial representations with an error interval remainder). Handelman representations [12] can handle more detailed representations of the internal descriptions, they are out of the scope of this paper since their application so far is to floating-point systems.

Gaffar *et al.* [58] have suggested an approach based on an *automatic differentiation* method and have applied it to linear or quasi-linear systems. The noise bounds are computed as the sum of the maximum deviation of each noise signal multiplied by its corresponding sensitivity. The main advantage of this approach is that the bounding expression is very easily obtained, since in this type of systems the sensitivities are the operands of the multiplications and the other terms of the Taylor series are considered negligible. However, since it is aimed at providing guaranteed

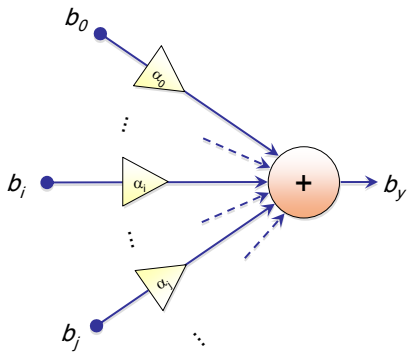


Figure 6: Model for the computation of output RON power based on noise sources b_i and gains α_i

bounds of the results, the provided WLs are usually overestimated even for small blocks [58].

Another interesting approach which acquired relevance in the latest years is the optimization of systems using *Arithmetic Transformations* (AT) [112, 125, 124]. ATs are polynomials that represent pseudo-boolean functions. Their extensions also include word-level inputs and sequential variables in the representations. AT representations are canonical, so the propagation of the polynomial terms is guaranteed to be accurate. In addition, due to their origin, they are particularly well suited to describe and optimize the operations of a given circuit.

In [112], authors distinguish three sources of error: approximation by the finite-order polynomial, quantization of the input signals, and optimization of the WLs of coefficients and result [112]. The combination of these three sources must be less than the specified error bound to provide a valid implementation. They initially determine the order of the Taylor series and the amount of input quantization. After that, a branch and bound algorithm, tuned for this application and guided by the sensitivity, is used for the optimization process [112]. In [125] and [124], the authors extend this approach to evaluate systems containing feedback loops. In [125], they provide the analytical expressions for the analysis of IIR filters, taking into account both MSE statistics and bounds as the target measurements. In [124], they extend this analysis to polynomial systems with loops, and show that AT paired with IA is more efficient than AA to provide the noise bounds. One of the main features of this approach is that it does not require numerical simulations, unlike other similar approaches.

3.2.2 Round-Off Noise Power

Existing approaches to compute the analytical expression of the quantization noise power are based on perturbation theory, which models finite precision values as the addition of the infinite precision values and a small perturbation. At node i, a quantization error signal b_i is generated when some bits are eliminated during a fixed-point format conversion (quantization). This error is assimilated to an additive noise which propagates inside the system. This noise source contributes to the output quantization noise b_y through the gain α_i, as shown in Fig. 6.

The aim of this approach is to define the output noise b_y power expression according to the noise source b_i parameters and the gains α_i between the output and a noise source.

Table 3 summarizes the main techniques to compute the RON power. The first column indicates the type of technique used. The second column displays the main characteristic of the technique, while the next column shows particular features of the cited approaches. The next three columns contain information about the type of systems that the approches handle (*All*, based on *smooth* operations and *LTI*), the existance of loops and the computational speed of the approach. The last columns shows the references to the published works.

The next paragraphs focus on the model used for the quantization process, which has three phases: (i) *noise generation*, (ii) *noise propagation*, and (iii) *noise aggregation*.

Noise Generation
In finite precision arithmetic, signal quantization leads to an unavoidable error. A commonly used model for the continuous-amplitude signal quantization has been proposed in [141] and refined in [129]. The quantization of signal x is modeled by the sum of this signal and a random variable b (quantization noise). This additive noise b is a uniformly distributed white noise that is uncorrelated with signal x and any other quantization noise present in the system (due to the quantization of other signals). The validity conditions of the quantization noise properties have been defined in [129]. These conditions are based on characteristic function of the signal x, which is the Fourier transform of the probability density function (PDF). This model is valid when the dynamic range of signal x is sufficiently greater than the quantum step size and the signal bandwidth is large enough.

Table 3: Techniques for the analytical evaluation of the quantization noise power

General features	Particular features	System	Loops	Speed	References
Hybrid Techniques					
Based on statistical expressions. Requires large matrix computations.	Coefficients K_i and L_{ij} are computed using fixed-point simulations and then subtituted in the statistical matrix equations.	Smooth	Yes	Medium	Shi [126]
		Smooth	Yes	Medium	Constantinides [33]
		Smooth	Yes	Medium	Fiore [56]
Impulse Response Determination					
Based on system transformations. Provides fast results.	Coefficients K_i and L_{ij} are computed from the impulse response between the noise sources and the output. Integrated in the *ID.Fix* tool.	LTI	Yes	Very fast	Menard [100]
		Smooth	Yes	Fast	Rocher [122]
Affine Arithmetic Simulations					
Based on AA simulations. Provides fast results.	Coefficients K_i and L_{ij} are computed from the results of the AA simulations. Integrated in the *Abaco* and *Quasar* tools.	LTI	Yes	Very fast	Lopez [93]
		Smooth	Yes	see note[1]	Caffarena[18]
Combines MAA and PCE.	Provides accurate results in strongly nonlinear systems.	Poly-nomial	Yes	Medium/ Fast	Esteban [50]

[1]: Fast for LTI & non-linear acyclic systems and slow for non-linear cyclic systems

This model has been extended to include the computation noise in a system resulting from some bit elimination during a fixed-point format conversion. More especially, the round-off error resulting from the multiplication of a constant by a discrete amplitude signal has been studied in [6]. This study is based on the assumption that the PDF is continuous. However, this hypothesis is no longer valid when the number k of bits eliminated during a quantization operation is small. Thus, in [30], a model based on a discrete PDF is suggested and the first and second-order moments of the quantization noise are given. In this study, the probability value of each eliminated bit to be equal to 0 or 1 is assumed to be $1/2$.

Noise Propagation
Each noise source b_i propagates to the system output and contributes to the noise b_y at the output. The propagation noise model is based on the assumption that the quantization noise is sufficiently small compared to the signal to consider that the finite precision values can be modeled by using the addition of the infinite precision values and a small perturbation. A first-order Taylor approx-

imation [33, 121] is used to linearize the operation behavior around the infinite precision values. This approach allows obtaining a time-varying linear expression of the output noise according to the input noise [99]. In [126], a second-order Taylor approximation is used directly on the expression of the output quantization noise. In [93] and [18], affine arithmetic is used to model the propagation of the quantization noise inside the system. Affine expression allows obtaining directly a linear expression of the output noise according to the input noises. For non-affine operations, a first order Taylor approximation is used to obtain a linear behaviour. These models, based on the perturbation theory, are only valid for smooth operations. An operation is considered to be smooth if the output is a continuous and differentiable function of its inputs.

Noise Aggregation

Finally, the output noise b_y is the sum of all the noise source contributions. The second order moment of b_y can be expressed as a weighted sum of the statistical parameters of the noise source:

$$E(b_y^2) = \sum_{i=1}^{N_e} K_i \sigma_{b_i}^2 + \sum_{i=1}^{N_e} \sum_{j=1}^{N_e} L_{ij} \mu_{b_i} \mu_{bj} \tag{3}$$

where μ_{b_i} and $\sigma_{b_i}^2$ are respectively the mean and the variance of noise source b_i, and N_e is the total number of error sources. These terms depends on the fixed-point formats and are determined during the evaluation of the accuracy analytical expression. The terms K_i and L_{ij} are constant and depend on the computation graph between b_i and the output. Thus, these terms are computed only once for the evaluation of the accuracy analytical expression. These constant terms can be considered as the gain between the noise source and the output.

For the case of Linear Time-Invariant systems, the expressions of K_i and L_{ij} are given in [101]. The coefficient L_{ij} can now be computed by the multiplication of terms L_i and L_j, which can be calculated independently. The coefficients K_i and L_{ij} are determined from the transfer function $H_i(z)$ or the impulse response $h_i(n)$ of the system having b_i as input and b_y as output. In [102, 100], a technique is proposed to compute these coefficients from the SFG (Signal Flow Graph) of the application. The recurrent equation of the output contribution of b_i is computed by traversing the SFG representing the application at the noise level. To support recursive systems, for which the SFG contains cycles, this SFG is transformed into several Directed Acyclic Graphs (DAG). The recurrent equations associated to each DAG are computed and then merge together after a set of variable substitutions. The different transfer functions are determined from the recurrent equations by applying a Z transform.

In [18], AA is used to keep track of the propagation of every single noise contribution along the datapath, and from this information the coefficients K_i and L_i are extracted. The method has been proposed for LTI in [93] and for non-LTI systems in [18]. An affine form, defined by a central value and an uncertainty term (error term in this context), is assigned to each noise source. These terms depend on the mean and variance of the noise source. Then, the central value and the uncertainty terms associated to each noise source are propagated inside the system through an affine arithmetic based simulation. The values of the coefficients K_i and L_{ii} are extracted from the affine form of the output noise. In the case of recursive systems, it is necessary to use a large number of iterations to ensure that the results converge to stable values. In some cases, this may lead to large AA error terms and therefore to long computation time.

In the method proposed in [122], an analytical expression of the coefficients K_i and L_{ij} is determined. For each noise source b_i, the recurrent equation of the output contribution of b_i is determined automatically from the application SFG with the technique presented in [100]. A time-varying impulse response h_i is computed from each recurrent equation. The output quantization noise b_y is the sum of the noise source b_i convolved with its associated time varying impulse response. The second-order moment of b_y is determined. The expression of the coefficients is proposed in [122]. These coefficients can be computed directly from their expression by approximating an infinite sum, or a linear prediction approach can be used to obtain more quickly the value of these coefficients. The statistical parameters of the signal terms involved in the expres-

sion of the coefficients are computed from a single floating-point simulation, leading to reduced computation times. The analysis to compute coefficients K_i and L_{ij} is done on an SFG representing the application and where the control flow has been removed. To avoid loop unrolling which can lead to huge graph, a method based on polyhedral analysis has been proposed in [44].

Different hybrid techniques [126, 33, 56] that combine simulations and analytical expressions have been proposed to compute the coefficients K_i and L_{ij} from a set of simulations. In [126], these $N_e(N_e+1)$ coefficients are obtained by solving a linear system in which K_i and L_{ij} are the variables. The way to proceed is to carry out several fixed-point simulations where a range of values for σ_{b_i} and μ_{b_i} is covered for each noise source. The fixed-point parameters of the system are set carefully to control each quantizer and to analyze its influence on the output. For each simulation, the statistical parameters of each noise source b_i are known from the fixed-point parameter and the output noise power is measured. At least $N_e(N_e+1)$ fixed-point simulations are required to be able to solve the system of linear equations. A similar approach is used in [56] to obtain the coefficients by simulation. Each quantizer is perturbed to analyze its influence at the output to determine K_i and L_{ii}. To obtain the coefficients L_{ij} with $i \neq j$, the quantizers are perturbed in pairs. This approach requires again $N_e(N_e+1)$ simulations to compute the coefficients, which requires long computation times.

During the last fifteen years, numerous work on analytical approaches for RON power estimation have been conducted and interesting progresses have been made for the automation of this process. These approaches allow for the evaluation of the RON power and are very fast compared to simulation-based approaches. Theoretical concepts have been established enabling the development of automatic tools to generate the expression of the RON power. The limit of the proposed methods have been identified. Analytical approaches based on perturbation theory are valid for systems made-up of only smooth operations.

3.2.3 Probability Density Function

The probability density function (PDF) of the quantization noise has been used as a metric to analyze the effect of signal quantization. This metric provides more information than the quantization error bounds or the quantization noise power. They are of special interest if applied to the analysis of unsmooth operations since error bounds or noise power are mainly suitable for differentiable operations.

There are two types of measures used to optimize quantized systems: statistical analysis of the quantization noise, and guaranteed bounds of the results. In most cases, statistical analysis techniques only compute the mean and variance of the quantization noise (or, alternatively, the noise power) at the output signal. Since the number of noise sources is usually high, these techniques assume that the Central Limit Theorem is valid, and the output noise follows a Gaussian distribution. Consequently, these two parameters fully characterize the distribution of the quantization noise. However, in systems with non-linear blocks (such as slicers) the Central Limit Theorem can no longer be valid, and a more detailed analysis is required. In this sense, some work focused on evaluating the PDF of the quantization noise.

In the context of guaranteed bounds, the objective is to ensure that the maximum distortion introduced in the quantization process is below a given constraint. Some techniques select the WLs and perform the computations to ensure that the bounds of the quantization noise are below this constraint. Other techniques focus on ensuring that the output of the quantized system is equal to a valid reference (e.g., the floating-point one). In both cases, to obtain efficient implementations, it is important to ensure that the provided bounds are close to the numerical ones, and that the oversizing included in the process (if any) is small.

Stochastic approaches, based on Karhunen-Loève Expansion (KLE) and Polynomial Chaos Expansion (PCE), have been used to model the quantization noise at the output of a system. The output quantization noise PDF can be extracted from the coefficients of the KLE or PCE. In the domain of fixed-point system design, these techniques have been previously proposed to

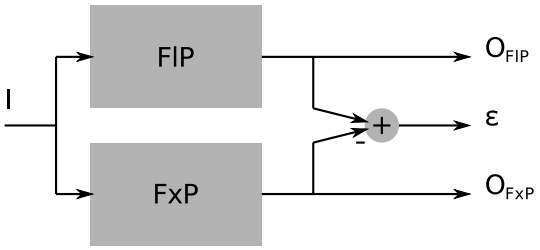


Figure 7: Simulation-based computation of quantization error

determine the signal dynamic range in LTI [145] and non-LTI systems [146]. In [3], a stochastic approach using KLE is used to determine the quantization noise PDF of an LTI system output. The KLE coefficients associated to a noise source are propagated to the output by means of the impulse response between the noise source and the system output. In [50], a stochastic approach based on a combination of Modified Affine Arithmetic (MAA) and Polynomial Chaos Expansion (PCE) is proposed to determine the output quantization noise PDF. Compared to KLE based approach, PCE allows supporting non-LTI systems. This technique is based on decomposing the random variables into weighted sums of Legendre orthogonal polynomials. The Legendre polynomial bases are well suited to represent uniformly distributed random variables, thus, they are very efficient to model quantization noise.

The determination of the PDF is required to handle unsmooth operations. In [127], the effect of quantization noise on the signum function is analyzed. This work has been extended in [115] to handle more complex decision operations which have specific contours like in QAM (Quadrature Amplitude Modulation) constellation diagrams. These two models are defined for one single unsmooth operation. Handling systems with several unsmooth operations is still an open issue for purely analytical approaches.

3.3 Simulation-based and Mixed Approaches

3.3.1 Fixed-point Simulation-based Evaluation

The quantization error can be obtained by extracting the difference between the outputs of simulation when the system has a very large precision (e.g. simulation with double-precision floating-point) and when there is quantization (bit-true fixed-point simulation), as shown in Fig. 7. Floating-point simulation is considered to be the reference given that the associated error is definitely much smaller than the error associated to fixed-point computation. Different error metrics can be computed from the quantization error obtained from this simulation. The main advantage of simulation-based approaches is that every kind of application can be supported. Fixed-point simulation can be performed using tools such as [40, 75, 96, 47].

Different `C++` classes, to emulate the fixed-point mechanisms have been proposed, such as `sc_fixed` (*SystemC*) [11], `ac_fixed` (*Algorithm C Data Types*) [104] or `gFix` [77]. The `C++` class attributes define the fixed-point parameters associated to the data: integer and fractional word-lengths, overflow and quantization modes, signed/unsigned operations. For `ac_fixed`, the fixed-point attributes can be parametrized through template parameters. For `sc_fixed`, these attributes can be static to obtain fast simulations or dynamic so they can be modified at run-time. Bit-true operations are performed by overloading the different arithmetic operators. During the execution of a fixed-point operation, the data range is analyzed and the overflow mode is applied if required. Then, the data is cast with the appropriate quantization mode. Thus, for a single fixed-point operation, several processing steps are required to obtain a bit true simulation. Therefore, these techniques suffer from a major drawback which is the extremely long simulation time [39]. This becomes a severe limitation when these methods are used in the data word-length optimization process where multiple simulations are needed. The simulations are made on floating-point

machines and the extra-code used to emulate fixed-point mechanisms increases the execution time between one to two orders of magnitude compared to traditional simulations with native floating-point data types [76, 36]. Besides, to obtain an accurate estimation of the statistical parameters of the quantization error, a great number of samples must be taken for the simulation. This large number of samples combined with the fixed-point mechanism emulation lead to very long simulation time.

Different techniques have been proposed to reduce this overhead. The execution time of the fixed-point simulation can be reduced by using more efficient fixed-point data types. In [77], the aim is to reduce the execution time of the fixed-point simulation by using efficiently the floating-point units of the host computer. The mantissa is used to compute the integer operations. Thus, the word-length of the data is limited to 53 bits for double data types. The execution time is one order of magnitude greater than the one required for a fixed-point simulation. This technique is also used in *SystemC* [11] for the fast fixed-point data types.

The fixed-point simulation can be accelerated by executing it on a more adequate machine like a fixed-point DSP [78, 73, 39, 82, 37] or an FPGA [38] through hardware acceleration. In the case of hardware implementation, the operator word-length, the supplementary elements for overflow and quantization modes are adjusted to comply exactly with the fixed-point specification which has to be simulated. In the case of software implementation, the operator and register word-lengths are fixed. When the word-length of the fixed-point data is lower than the data word-length supported by the target machine, different degrees of freedom are available to map the fixed-point data into the target storage elements. In [39], to optimize this mapping, the execution time of the fixed-point simulation is minimized. The cost integrates the data alignment and the overflow and quantization mechanism. This combinatorial optimization problem is solved by a divide and conquer technique and several heuristics to limit the search space are used. In [82] a technique is proposed to minimize the execution time due to scaling operations according to the shift capabilities of the target architecture. In the same way, the aim of the Hybris simulator [76] [36] is to optimize the mapping of the fixed-point data described with *SystemC* into the target architecture register. All compile-time information are used to minimize the number of operations required to carry-out the fixed-point simulation. The overflow and quantization operations are implemented by conditional structures, a set of shift operations or bit mask operations. Nevertheless, to obtain fast simulation, some quantization modes are not supported. In [143], the binary point alignment is formulated as a combinatorial optimization problem and an integer linear programming approach is used to solve it. But, this approach is limited to simple applications to obtain reasonable optimization times. These methods reduce the execution time of the fixed-point simulation but, this optimization needs to be performed every time that the fixed point configuration changes. Accordingly, it might not compensate for the execution time gain of the fixed-point simulation when involving complex optimizations.

3.3.2 Mixed Approach

To handle systems made-up of unsmooth operations, a mixed approach which combines analytical evaluations and simulations has been proposed in [113, 114]. The idea is to evaluate directly the application performance metric with fixed-point simulation and to accelerate drastically the simulation with analytical models. In this technique the analytical approach is based on the perturbation theory and the simulation is used when the assumptions associated with perturbation theory are no longer valid (i.e. when a decision error occurs). In this case, the quantization noise at the unsmooth operation input can modify the decision at the operation output compared to the one obtained with infinite precision.

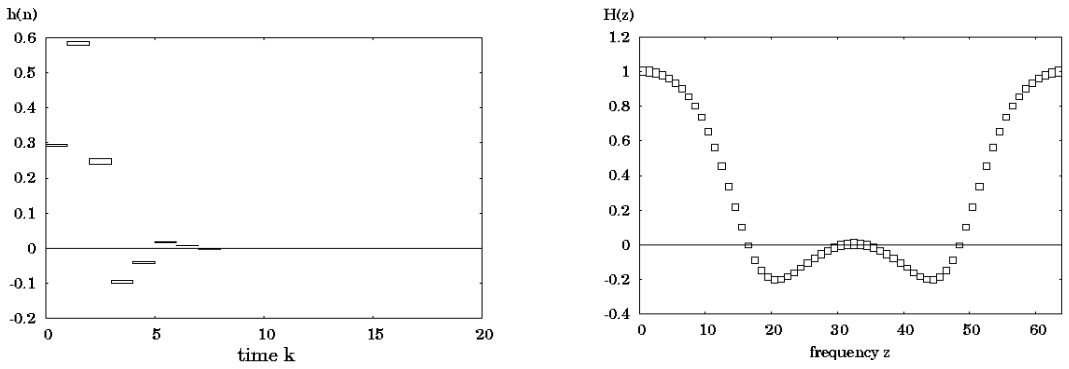
This technique selectively simulates parts of the system only when an decision error occurs [114]. Given that decision errors are rare event the simulation time is not so important as for classical fixed-point simulations. The global system is divided into smooth clusters made-up of smooth operations. These smooth clusters are separated by unsmooth operations. The single

source noise model [103] is used to capture the statistical behavior of quantization noise accurately at the output of each smooth cluster. In [103], The authors propose to model the output quantization noise of a LTI system with a weighted sum of a Gaussian random variable and a uniform random variable. In [123], the output quantization noise of a smooth system is modeled by a generalized Gaussian random variable, whose parameters define the shape of the PDF. These parameters are analytically determined from the output quantization noise statistics (mean, variance and kurtosis). The general expression of the noise moments are given in [123], and are computed from the impulse responses between the noise sources and the system output.

4 Effect of Coefficient Quantization

Coefficient Quantization (CQ) is the part of the implementation process that describes the degradation of the system operation due to the finite WL representation of the constant values of a system. Especially this problematic is of high importance for LTI systems with the quantization of the coefficients. Opposite to RON, CQ modifies the impulse and frequency responses for LTI system and the functionality for other systems. In the analysis of the quantization effects for LTI systems, this parameter is the first to be determined, since it involves two major tasks: (i) the selection of the most convenient filter structure to perform the required operation, and (ii) the determination of the actual values of the coefficients associated to it.

Figure 8 illustrates the amount of deviation due to CQ by means of interval simulations. A butterworth filter has been realized in DFIIt (Direct Form II transposed) form, and each coefficient has been replaced by a small interval that describes the difference between the ideal coefficient and the quantized one using 7 fractional bits. Figure 8.a shows the impulse response of the realization, where the size of each interval reveals how sensitive is each sample to this quantization of coefficients. Figure 8.b shows the transfer function associated to it, where in this case the intervals reveal the most sensitive frequencies to the same set of quantizations.



(a) (b)

Figure 8: Effect of CQ on a given filter realization: (a) Evolution in time of the impulse response of the differences in the output response. (b) Distribution of the effects in the frequency domain. The intervals represent the deviation between the quantized and unquantized samples of the impulse response and the transfer function.

In LTI systems, CQ has been traditionally measured using the so-called Coefficient Sensitivity (CS). Although this parameter was originally defined for LTI systems, whose operation is described by $H(z)$, its current use has also been extended to non-linear systems.

Table 4 summarizes the most important techniques and groups related to the computation of the CS. The first column indicates the type of technique used to compute this parameter (residues,

geometric sum of matrices, Lyapunov equations, perturbation theory). The second and third columns respectively provide the most important work in this area, and the most relevant features in each case. The last two columns provide the main advantages and disadvantages of the different approaches. First, an overview of the different parameters used in the literature to measure the CS is presented, before discussing in more detail the L_2-sensitivity. Second, the most commonly-used L_2-sensitivity computation procedures are described. Finally, a generic algorithm that perform fast computation of the L_2-sensitivity is described.

Table 4: Measurement techniques for the computation of the Coefficient Sensitivity (CS)

Features		Advantages	Disadvantages	References
Evaluation of the Residues				
General analytical procedure based on complex mathematical equality.		General method. Provides exact results.	Very complex to develop. Different analysis for each structure.	Roberts [119]
Geometric Sum of Matrices				
Analytical procedure that approximates S_{L12} by using infinite sums in state-space realizations.		The analytical expressionis easier to obtain.	Limited to state-space realizations. Provides an upper bound.	Hinamoto [66]
Lyapunov Equations				
Provides the analytical expression for families of filter structures, mainly state-space realizations.		Fast and exact results (without infinite sums).	Iterative method. Limited to certain filter structures.	Li [89] Hilaire [64]
Perturbation Theory				
Compute the sum of deviations of all the coefficients.	Analytical approach based on Lyapunov Equation.	Extremely fast, if the analytical expression is obtained	Limited to state-space realizations.	Xiao [147]
	Interval-based procedure.	Fast and automatic. Valid for all types of systems.	Approximated value. Requires interval computations support.	Lopez [91]

4.1 Measurement Parameters

A number of procedures have been initially suggested to minimize the degradation of $H(z)$ with respect to the quantization of all coefficients of the realization under different constraints [133, 134, 135]. In these procedures, the coefficients of the realization have been obtained by minimizing the so-called L_1/L_2-*sensitivity*, $S_{L_{12}}$ [133, 134, 135, 69, 59, 144, 68, 67]. The main feature of this parameter is that its upper bound is easily obtained [59, 144, 88]. However, two different norms are applied to obtain the result. Therefore, its physical interpretation is not clear.

Instead, it is more natural to measure the deviations of $H(z)$ using only the L_2-norm [68, 88]. For this reason, the so-called L_2-*sensitivity*, S_{L2}, is currently applied [68, 67]. The main feature of this parameter is that it is proportional to the variance of the deviation of $H(z)$ due to the quantization of all the coefficients of the realization [59, 144, 68, 67]. However, the computation of its analytical expression requires performing extremely complex mathematical operations [144, 68, 89]. Due to this fact the computation of the L_2-sensitivity has been limited to simple

linear structures, typically SSR (State-Space Representation) forms. Since each analytical expression only characterizes one family of filter structures, it requires developing a new mathematical expression to optimize or compare each new structure. The most recent work in this area are focused on minimizing the L_2-sensitivity of two-dimensional (2-D) SSR filter structures [68, 67], and of structures based on the generalized delta operator [89, 148].

In [136, 137, 138], the authors have compared the performance of the filter structures by computing the maximum value of the magnitude sensitivity, S_{mag}, or the relative sensitivity, S_{mag}. The main feature of S_{mag} and S_{rel} is that their numerical values are more easily computed than the analytical expressions of $S_{L_{12}}$ or S_{L2}. For this reason, they have been used in combination with simulated annealing or genetic algorithms that perform automated search of the most robust structures against the quantization of coefficients [136, 137, 138]. However, S_{mag} and S_{rel} only provide information of the maximum deviations of $H(z)$. In contrast, the L_2-sensitivity provides global information about the deviations of $H(z)$. For this reason, this parameter is widely preferred [59, 68, 88, 89, 66].

In [64], the authors introduce a unified algebraic description able to represent the most widely used families of filter realizations. They focus on the fixed- and floating-point deviation of the transfer function and pole measures using CS parameters. They apply Adaptive Simulated Annealing to obtain the optimal realization among these structures. In particular, they introduce the S^W_{L2} measure, which considers the individual quantization of coefficients into the traditional L_2-sensitivity parameter. This work has been further expanded in [62] to include L_2-scaling constraints, and in [63] to include the evaluation of MIMO filters and controllers.

Table 5 summarizes the parameters introduced in this Section. In each column, the representation of the different parameters, the main references associated to them, and their most important features, advantages and disadvantages are also briefly outlined.

Table 5: Measurement parameters for coefficient quantization

Parameter	Features	Advantages	Disadvantages	References
S_{L12}	Initial measure of the coefficient sensitivity, based on the L_1 and the L_2-norms.	It has a simple expression in some filter structures	Only provides an upper bound, based on two different norms.	[133, 134, 135, 69]
S_{L2}	Advanced measurement, based only on the L_2-norm. Development of the expressions associated to each filter structure.	Global measurement. It has statistical meaning.	Complex to develop.	[59, 144, 68, 67, 88, 89, 148]
S_{mag}, S_{rel}	Information about the magnitude of the quantizations.	Computationally simple.	Only provides information about the maximum deviations.	[136, 137, 138]
S^W_{L2}	Measures the actual deviations of coefficients with different amount of quantizations.	More accurate than S_{L2}.	Requires complex analytical developments.	[64, 62, 63]

4.2 L_2-Sensitivity

Since the L_2-Sensitivity is much more commonly used than the others CQ measurement parameters, in this section its mathematical definition and physical interpretation are described in more detail.

Definition The L_2-sensitivity is the parameter that quantitatively measures the influence of the variations of all the coefficients of the realization in the transfer function. Its mathematical definition is as follows

$$S_{L_2} = \sum_{i=1}^{n_c} S_{c_i} = \sum_{i=1}^{n_c} \left\| \frac{\partial H(z)}{\partial c_i} \right\|_2^2 \tag{4}$$

where S_{c_i} is the sensitivity of the transfer function with respect to coefficient c_i, and $\|X(z)\|_2^2$ represents the L_2-norm of $X(z)$ [89, 66]. This definition considers that all the coefficients of the set $i = \{1, \ldots, n_c\}$ are affected by quantization [45]. Coefficients not affected by quantization operations (i.e., those that are exactly represented with the assigned number of bits) are excluded from this set.

Statistical interpretation Using a first-order approximation of the Taylor series, the degradation of $H(z)$ due to the quantization of the coefficients follows

$$\Delta H(z) = H_{Q_c}(z) - H(z) = \sum_{i=1}^{n_c} \frac{\partial H(z)}{\partial c_i} \Delta c_i \tag{5}$$

where $H_{Q_c}(z)$ is the transfer function of the realization with quantized coefficients. From a statistical point of view, the variance of the degradation of $H(z)$ due to these quantization operations is given by

$$\sigma_{\Delta H}^2 = \sum_{i=1}^{n_c} \left\| \frac{\partial H(z)}{\partial c_i} \right\|_2^2 \sigma_{\Delta c_i}^2 = \sum_{i=1}^{n_c} S_{c_i} \sigma_{\Delta c_i}^2 \tag{6}$$

When all the coefficients are quantized to the same number of bits, $\sigma_{\Delta c_i}^2$ is equal to the common value $\sigma_{\Delta c}^2$. In this case, eq. (6) is simplified to

$$\sigma_{\Delta H}^2 = \sum_{i=1}^{n_c} S_{c_i} \sigma_{\Delta c}^2 = S_{L_2} \sigma_{\Delta c}^2 \tag{7}$$

where $\sigma_{\Delta c}^2$ is the variance of the coefficients affected by the quantization operations.

Therefore, S_{L2} provides a global measure of the degradation of $H(z)$ with respect to the quantization of all the coefficients of the realization. Consequently, in the comparison of the different filter structures, the L_2-sensitivity indicates the most robust realizations against the quantization of coefficients. However, it must be noted that once the final realization has been chosen, the quantization of coefficients has deterministic effects on the computation of the output samples, and the behaviour of the filter structure is completely determined by $H_{Q_c}(z)$.

4.3 Analytical Approaches to Compute the L_2-Sensitivity

The analytical computation of the L_2-sensitivity is based on calculating the individual sensitivities of the coefficients of the realization. There are three different types of techniques: (i) evaluation of residues, (ii) geometric series of matrices, or (iii) Lyapunov equations. However, since all of them are based on developing expressions for the different realizations, they are only valid for particular structures, mainly SSR (State-Space Realization) and DFIIt (Direct Form II transposed) forms.

Evaluation of the Residues The reference procedure to compute the value of S_{L2} is to analytically develop the expressions of the derivatives of $H(z)$ [119]. This approach separately computes the L_2-norms of the sensitivities of the coefficients. The derivatives involved in this process are extremely complex, even in simple LTI systems. Therefore, this procedure is only applicable to compute the reference values in some low-complexity LTI systems.

Geometric Series of Matrices (GSM) In this case, the expression to compute the S_{L2} is transformed into an equivalent expression that computes the sensitivity of all the coefficients of the

same group [66]. This procedure computes an upper bound of S_{L2}, which is equal to the real value if all the coefficients of the SSR filter are quantized [147]. Its main advantage is that it is easily extended to n-D filters [66]. However, it has two important drawbacks: (i) its application to non-SSR structures or sparse realizations has not been defined; and (ii) due to the infinite sums involved, the results are only approximated up to a given degree of accuracy. The approximations can be made as accurate as required by adding a large number of terms, but in such cases the computation times involved to provide the results can be very high.

Lyapunov Equations (LEs) In this procedure, the computation of the infinite sum of matrices of the GSM method is replaced by the computation of the solutions of their associated LEs. This procedure is very accurate and fast, but requires performing iterative computations, and the involved equations must be solved for each non-zero coefficient [147]. Its main drawback is that these expressions are only applicable to 1-D SSR filters. This procedure has also been used in [89] to develop the expressions of the L_2-senstivity of DFIIt structures with generalized delta operators, and in [64, 65] to include different amounts of quantization in each coefficient of the realization.

Perturbation methods The existing analytical techniques to compute the S_{L2} have the drawback of being only valid for each family of filter structures, and the required expressions are in most cases very difficult to develop. Moreover, these techniques cannot be extended to evaluate the sensitivity of a given signal in non-linear systems.

In [147], the author suggests an analytical approach based on an improved S_{L2} measure that separately computes the sensitivities of all the coefficients of the realization. Using this improved measure, an analytical expression to compute the S_{L2} based on LEs for state-space realizations is derived. This measure is more accurate, and the computation of S_{L2} as the sum of contributions of the individual coefficients facilitates the automatization. The author also develops the analytical expressions for the state-space realizations, but these expressions cannot be generalized.

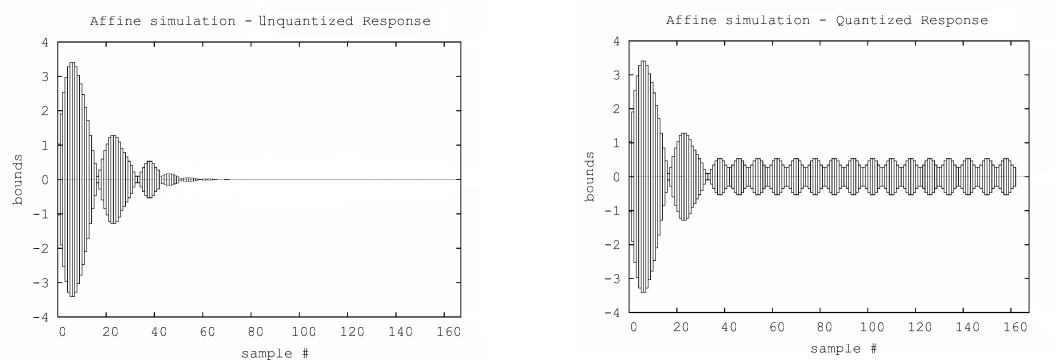
5 System Stability due to Signal Quantization

Although most of existing techniques to evaluate the quantization effects are based on substituting the quantizers by additive noise sources, this aproximation is only valid under certain assumptions (see Section 3.2) [10, 117, 6, 71, 142]. In particular, when the quantization operations in the feedback loops significantly affect the behavior of the system, oscillations of a given frequency and amplitude may appear, provoking an unstable behaviour at the output. These oscillations are called Limit Cycles [27, 119, 97, 106, 116].

Fig. 9 shows an example of the existence of LCs. In unquantized systems, the output response tends to zero, since it is a requirement of the stability of the LTI systems (Fig. 9-a). In quantized systems, due to the nonlinear effect of the quantization operations, the output response may present self-sustained oscillations of a given amplitude and frequency (Fig. 9-b). These two parameters vary according to the quantized realization and the values of the input signals, although certain conditions have been provided in the literature to keep them under a given limit.

To detect the oscillations, the actual behavior of the quantizers must be evaluated, instead of substituting them by their respective equivalent linear models (i.e. noise sources) [27, 117, 6, 119]. In LTI systems these oscillations have been extensively analyzed in the second-order sections [26, 25, 27, 16, 87], and sufficient conditions that ensure the absence of LCs have also been developed [72, 46, 128], particularly in regular filters structures [54, 61, 27, 48, 49, 55, 116, 70, 17, 119].

In this Section, a classification of the procedures most commonly used to guarantee the absence of LCs in digital filters is first presented, followed by a description of the automatic techniques to detect LCs.



(a) (b)

Figure 9: Detection of LCs in a filter using AA-based computations. The joint simulation of all the input values allows fast detection of system instabilities and self-sustained oscillations: (a) The unquantized response of the reference interval [-1,1] at sampled time k=0 tends to zero. (b) The quantized system generates LCs due to the nonlinear effects of the quantization operations and the feedback loops.

5.1 Analysis of Limit Cycles in Digital Filters

Limit Cycles (LCs) are self-sustained oscillations that appear due to the propagation of the nonlinear effects of the quantization operations through the feedback loops [119, 97, 106, 116]. The techniques aimed to detect LCs primarily intend to bound the maximum amplitudes of these oscillations [19], and, in particular, their effect at the output signal.

Similarly to the computation of the RON and the CQ, the techniques used to detect and bound the LCs are classified into analytical and simulation-based. *Analytical techniques* provide three different types of results [19]: (i) they give sufficient conditions to ensure asymptotic stability of filters after quantization [105, 119, 15, 14]; (ii) they present requirements for the absence of LCs [139]; or (iii) they describe strategies to eliminate zero-input and constant-input LCs [83]. These techniques have been used to select realizations where the absence of LCs is guaranteed. However, they are not able to evaluate all the possible values of the coefficients, so in general they must be combined with simulation-based procedures for a detailed analysis of the target structure. Moreover, these techniques have focused on obtaining the analytical expressions of the coefficients of the second-order sections and SSR filters, but there are few results about factored-SSR filters of arbitrary order [119], and they do not consider arbitrary number of quantizers. Consequently, this type of technique is not suitable to perform automated analysis of LCs of generic filter structures.

Simulation-based techniques perform exhaustive evaluations of all the possible sets of values of the state variables [8, 118, 74, 90, 92, 19]. They provide precise results, but they require exceedingly long computation times [8, 118, 74]. Consequently, this type of technique allows automated analysis of LCs in generic filter structures, but requires a bounding stage to perform these computations in realistic computation times.

The application of AA-based simulations reduces by several orders of magnitude the computation time required to bound the LCs of generic filter structures. Moreover, they can be used in combination with numerical simulations to detect the presence or to guarantee the absence of LCs [90, 92].

5.2 Simulation-based LC Detection Procedures

Existing simulation-based LC detection procedures perform the computation in two stages [8, 118, 74]: (i) they compute the bounds of the maximum amplitude and frequency of the LCs; and (ii) they perform exhaustive search for LCs among all the possible combinations of values of the state variables (SVs) contained within these bounds. Since the SVs have a finite number of bits, the number of possible combinations of values of the SVs is also finite, i.e.,

$$n_{st} = \prod_i (2^{q_i}), \qquad i = 1, \ldots, n_{SV} \tag{8}$$

where n_{SV} is the number of SVs of the target structure, and q_i is the number of bits of state variable i.

From (8), it is clear that the number of combinations, n_{st}, is huge even for small-order filters. Consequently, the aim of the first step is to reduce the number of combinations to be tested for LCs. This reduction is obtained by limiting the maximum values of the SVs, M, or the maximum period of oscillation, T_{max} [118, 74]. The expressions of M and T_{max} are difficult to obtain, and they are dependent on the filter structure. The interested reader is referred to [118] for the expressions of these parameters in SSR filters, and to [74] for their expressions in second-order DFIIt forms with delta operators.

The exhaustive search is performed by evolving the values of the SVs. In each iteration, four possible cases may occur [74]: (i) The state vector is repeated, which means that a LC is found. (ii) The state converge to a point that produces zero output. This situation occurs when the values of the SVs are below a given threshold. (iii) The state vector has grown out of the search space. (iv) The maximum number of steps has expired. If none of these situations occur, the state vector evolves to the values of the next iteration. The most recent algorithms make use of alternative procedures to speed up the required computations, but they still follow the basic principles explained above [74]. They consider that: (a) the large values of the SVs do not need to be tested due to condition (iii); (b) the small values of the SVs converge to zero output in short time; and (c) most LCs have short period of oscillation, so they are quickly identified.

In summary, the existing simulation-based procedures are based on performing exhaustive searches among the values of the SVs but they need a binding stage, which depends on the target structure. This type of procedure can be accelerated in combination with AA [90, 92], since it is capable of evaluating a large number of states in a single execution of the algorithm.

6 Summary

Fixed-point design plays a major role in the VLSI implementation of state-of-the-art multimedia and communication applications. This paper surveys the major works related to the automated evaluation of fixed-point quantization effects, focusing on signal quantization, coefficient quantization and system stability. The main approaches in the field have been explained and classified covering simulation-based, analytical and hybrid techniques. The paper is intended to provide digital designers with a useful guide while facing the design of fixed-point systems.

When assessing the effect of signal quantization the designer can use general approaches such as simulation-based techniques but at the expense of expending a long time in the quantization process. For particular types of systems it is possible to apply analytical and hybrid automatic techniques that reduce computation time considerably. As a general remark, all the available techniques are not suitable to the optimization of high-complexity systems, so a system-level approach to quantization is most nedded.

Regarding, coefficient quantization the designer has to check the impact of finite WL coefficient on the system properties (i.e. frequency response). The majority of the available techniques are system-specific and require the manual development of analytical expressions, so there are still research opportunities in this problem.

Finally, the detection of LCs, the main approaches are based on simulations and exahustive search, so the computation time can be high for complex systems. Also, the starting condition of the algorithms are system dependant so the results depend on user experience. There are preliminary works on finding a general and fast approach to LC detection, so there is still room for research in this area.

Quantization is an intriguing field of research which has been open for more than 30 years, an the most impacting contributions are still to come, as no general solution exists yet in practice.

References

[1] A. Ahmadi and M. Zwolinski. Symbolic noise analysis approach to computational hardware optimization. In *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, pages 391–396, 2008.

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

[3] A. Banciu, E. Casseau, D. Menard, and T. Michel. Stochastic modeling for floating-point to fixed-point conversion. In *Proc. IEEE International Workshop on Signal Processing Systems, (SIPS)*, Beirut, october 2011.

[4] P. Banerjee, D. Bagchi, M. Haldar, A. Nayak, V. Kim, and R. Uribe. Automatic conversion of floating point matlab programs into fixed point fpga based hardware design. In *Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on*, pages 263–264, 2003.

[5] P. Banerjee, M. Haldar, A. Nayak, V. Kim, V. Saxena, S. Parkes, D. Bagchi, S. Pal, N. Tripathi, D. Zaretsky, R. Anderson, and J. Uribe. Overview of a compiler for synthesizing matlab programs onto fpgas. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(3):312–324, 2004.

[6] C. Barnes, B. N. Tran, and S. Leung. On the Statistics of Fixed-Point Roundoff Error. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3):595–606, june 1985.

[7] B. Barrois, K. Parashar, and O. Sentieys. Leveraging Power Spectral Density for Scalable System-Level Accuracy Evaluation. In *IEEE/ACM Conference on Design Automation and Test in Europe (DATE)*, page 6, Dresden, Germany, Mar. 2016.

[8] P. Bauer and L.-J. Leclerc. A computer-aided test for the absence of limit cycles in fixed-point digital filters. *Signal Processing, IEEE Transactions on*, 39(11):2400–2410, 1991.

[9] A. Benedetti and P. Perona. Bit-Width Optimization for Configurable DSPs by Multi-interval Analysis. In *IEEE Asilomar Conf. on Signals, Systems and Computers*, 2000.

[10] W. Bennett. Spectra of quantized signals. *Bell System Tech. J.*, 27:446–472, 1948.

[11] 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.

[12] D. Boland and G. Constantinides. Bounding variable values and round-off effects using handelman representations. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(11):1691–1704, 2011.

[13] D. Boland and G. Constantinides. A scalable precision analysis framework. *Multimedia, IEEE Transactions on*, 15(2):242–256, 2013.

[14] T. Bose and M. Chen. Overflow oscillations in state-space digital filters. *IEEE Trans. Circuits and Systems*, 38(7):807–810, 1991.

[15] T. Bose and M. Chen. Stability of digital filters implemented with two's complement truncation quantization. *IEEE Trans. Signal Process.*, 40(1):24–31, 1992.

[16] H. Butterweck, A. van Meer, and G. Verkroost. New second-order digital filter sections without limit cycles. *Circuits and Systems, IEEE Transactions on*, 31(2):141–146, 1984.

[17] M. Buttner. Elimination of limit cycles in digital filters with very low increase in the quantization noise. *Circuits and Systems, IEEE Transactions on*, 24(6):300–304, 1977.

[18] G. Caffarena, C. Carreras, J. Lopez, and A. Fernandez. SQNR Estimation of Fixed-Point DSP Algorithms. *Int. J. on Advances in Signal Processing*, 2010:1–11, 2010.

[19] J. Campo, F. Cruz-Roldan, and M. Utrilla-Manso. Tighter limit cycle bounds for digital filters. *Signal Processing Letters, IEEE*, 13(3):149–152, 2006.

[20] M. Cantin, Y. Savaria, D. Prodanos, and P. Lavoie. A Metric for Automatic Word-Length Determination of Hardware Datapaths. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10):2228–2231, october 2006.

[21] M.-A. Cantin, Y. Savaria, and P. Lavoie. A comparison of automatic word length optimization procedures. In *IEEE International Symposium on Circuits and Systems*, volume 2, pages II–612–II–615 vol.2, 2002.

[22] F. Catthoor, H. de Man, and J. Vandewalle. Simulated-annealing-based optimization of coefficient and data word-lengths in digital filters. *International Journal of Circuit Theory and Applications*, I:371390, 1988.

[23] M. Chang and S. Hauck. Precis: a usercentric word-length optimization tool. *Design Test of Computers, IEEE*, 22(4):349–361, 2005.

[24] J.-M. Chesneaux, L.-S. Didier, and F. Rico. Fixed CADNA library. In *Proc. conference on Real Number Conference (RNC)*, pages 215–221, Lyon, France, september 2003.

[25] T. Claasen and L. Kristiansson. Necessary and sufficient conditions for the absence of overflow phenomena in a second-order recursive digital filter. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 23(6):509–515, 1975.

[26] T. Claasen, W. Mecklenbrauer, and J. Peek. Second-order digital filter with only one magnitude-truncation quantizer and having practically no limit-cycles. *Electronics Letters*, 9(22):531–532, 1973.

[27] T. Claasen, W. Mecklenbrauker, and J. Peek. Effects of quantization and overflow in recursive digital filters. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 24(6):517–529, 1976.

[28] J. A. Clarke, G. A. Constantinides, and P. Y. K. Cheung. Word-length selection for power minimization via nonlinear optimization. *ACM Trans. Des. Autom. Electron. Syst.*, 14(3):1–28, 2009.

[29] G. Constantinides. *High Level Synthesis and Wordlength Optimization of Digital Signal Processing Systems*. PhD thesis, PhD. Thesis, Electr. Electron. Eng., Univ. London, 2001.

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

[31] G. Constantinides, P. Cheung, and W. Luk. Roundoff-noise shaping in filter design. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 4, pages 57–60, Geneva, may 2000.

[32] G. Constantinides, P. Cheung, and W. Luk. Wordlength optimization for linear digital signal processing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(10):1432– 1442, october 2003.

[33] G. A. Constantinides. Word-length optimization for differentiable nonlinear systems. *ACM Transactions on Design Automation of Electronic Systems*, 11(1):26–43, 2006.

[34] G. A. Constantinides, P. Y. K. Cheung, and W. Luk. Wordlength Optimization for Linear Digital Signal Processing. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 22(10):1432–1442, 2003.

[35] G. A. Constantinides and G. J. Woeginger. The complexity of multiple wordlength assignment. *Applied Mathematics Letters*, 15(2):137–140, 2002.

[36] M. Coors, H. Keding, O. Luthje, and H. Meyr. Fast Bit-True Simulation. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pages 708–713, Las Vegas, june 2001.

[37] M. Coors, H. Keding, O. Luthje, and H. Meyr. Integer Code Generation For the TI TMS320C62x. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Sate Lake City, may 2001.

[38] L. D. Coster. *Bit-True Simulation of Digital Signal Processing Applications*. PhD thesis, KU Leuven, 1999.

[39] L. D. Coster, M. Ade, R. Lauwereins, and J. Peperstraete. Code Generation for Compiled Bit-True Simulation of DSP Applications. In *Proc. IEEE International Symposium on System Synthesis (ISSS)*, pages 9–14, Hsinchu, december 1998.

[40] Coware. Coware SPW. Technical report, Coware, 2010.

[41] M. Daumas and G. Melquiond. Certification of bounds on expressions involving rounded operators. *ACM Trans. Math. Softw.*, 37(1):2:1–2:20, Jan. 2010.

[42] F. de Dinechin, C. Q. Lauter, and G. Melquiond. Assisted verification of elementary functions using gappa. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1318–1322, New York, NY, USA, 2006. ACM.

[43] L. de Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37(1):147–158, 2004.

[44] G. Deest, T. Yuki, O. Sentieys, and S. Derrien. Toward scalable source level accuracy analysis for floating-point to fixed-point conversion. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '14, pages 726–733, Piscataway, NJ, USA, 2014. IEEE Press.

[45] N. Doi, T. Horiyama, M. Nakanishi, and S. Kimura. Minimization of fractional wordlength on fixed-point conversion for high-level synthesis. In *Design Automation Conference, 2004. Proceedings of the ASP-DAC 2004. Asia and South Pacific*, pages 80 – 85, 27-30 2004.

[46] P. Ebert, J. Mazo, and M. Taylor. Overflow oscillations in digital filters. *Bell System Tech. J.*, 48:2999–3020, 1969.

[47] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming Heterogeneity, the Ptolemy Approach. *Proceedings of the IEEE*, 91, 2003.

[48] K. Erickson and A. Michel. Stability analysis of fixed-point digital filters using computer generated lyapunov functions- part i: Direct form and coupled form filters. *Circuits and Systems, IEEE Transactions on*, 32(2):113–132, 1985.

[49] K. Erickson and A. Michel. Stability analysis of fixed-point digital filters using computer generated lyapunov functions- part ii: Wave digital filters and lattice digital filters. *Circuits and Systems, IEEE Transactions on*, 32(2):132–142, 1985.

[50] L. Esteban, J. Lopez, E. Sedano, S. Hernandez-Montero, and M. Sanchez. Quantization analysis of the infrared interferometer of the tj-ii for its optimized fpga-based implementation. *Nuclear Science, IEEE Transactions on*, page accepted, 2013.

[51] C. Fang, T. Chen, and R. Rutenbar. Lightweight Floating-Point Arithmetic: Case Study of Inverse Discrete Cosine Transform. *EURASIP J. on Applied Signal Processing*, 2002(2002):879–892, 2002.

[52] C. Fang, T. Chen, and R. Rutenbar. Floating-point error analysis based on affine arithmetic. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2:561–564, 2003.

[53] C. Fang, R. Rutenbar, and T. Chen. Fast, accurate static analysis for fixed-point finite-precision effects in dsp designs. In *Int. Conf. on Computer-Aided Design, 2003 (ICCAD '03).*, pages 275–282, 2003.

[54] A. Fettweis. Some principles of designing digital filters imitating classical filter structures. *Circuits and Systems, IEEE Transactions on*, 18(2):314–316, 1971.

[55] A. Fettweis. Wave digital filters: Theory and practice. *Proceedings of the IEEE*, 74:270–327, 1986.

[56] P. Fiore. Efficient Approximate Wordlength Optimization. *IEEE Transactions on Computers*, 57(11):1561 –1570, november 2008.

[57] A. Gaffar, O. Mencer, and W. Luk. Unifying Bit-Width Optimisation for Fixed-Point and Floating-Point Designs. In *IEEE Symp. on Field-Programmable Custom Computing Machines*, pages 79–88, 2004.

[58] A. Gaffar, O. Mencer, W. Luk, P. Cheung, and N. Shirazi. Floating-point bitwidth analysis via automatic differentiation. In *Field-Programmable Technology, 2002. (FPT). Proceedings. 2002 IEEE International Conference on*, pages 158–165, 2002.

[59] M. Gevers and G. Li. *Parametrizations in control, estimation, and filtering problems : accuracy aspects*. Communications and control engineering series. Springer-Verlag, London ; New York, 1993. Michel Gevers and Gang Li.

[60] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.*, 23(1):5–48, 1991.

[61] A. Gray and J. Markel. Digital lattice and ladder synthesis. *IEEE Trans. Audio Electroacoust.*, 21:491–500, 1973.

[62] T. Hilaire. Low-parametric-sensitivity realizations with relaxed L_2-dynamic-range-scaling constraints. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 56(7):590–594, 2009.

[63] T. Hilaire and P. Chevrel. Sensitivity-based pole and input-output errors of linear filters as indicators of the implementation deterioration in fixed-point context. *EURASIP Journal on Advances in Signal Processing*, 2011(1):893760, 2011.

[64] T. Hilaire, P. Chevrel, and J. Whidborne. A unifying framework for finite wordlength realizations. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 54(8):1765–1774, 2007.

[65] T. Hilaire, D. Menard, and O. Sentieys. Bit Accurate Roundoff Noise Analysis of Fixed-point Linear Controllers. In *Proc. IEEE International Conference on Computer-Aided Control Systems (CACSD)*, pages 607–612, september 2008.

[66] T. Hinamoto, K. Iwata, and W.-S. Lu. l_2-sensitivity minimization of one- and two- dimensional state-space digital filters subject to l_2-scaling constraints. *Signal Processing, IEEE Transactions on*, 54(5):1804–1812, 2006.

[67] T. Hinamoto, H. Ohnishi, and W.-S. Lu. Minimization of l_2-sensitivity for state-space digital filters subject to l_2-dynamic-range scaling constraints. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 52(10):641–645, 2005.

[68] T. Hinamoto, S. Yokoyama, T. Inoue, W. Zeng, and W.-S. Lu. Analysis and minimization of l_2-sensitivity for linear systems and two-dimensional state-space filters using general controllability and observability gramians. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 49(9):1279–1289, 2002.

[69] L. Jackson. Roundoff noise bounds derived from coefficient sensitivities for digital filters. *Circuits and Systems, IEEE Transactions on*, 23(8):481–485, 1976.

[70] L. Jackson. Limit cycles in state-space structures for digital filters. *Circuits and Systems, IEEE Transactions on*, 26(1):67–68, 1979.

[71] L. Jackson. *Digital Filters and Signal Processing*. Kluwer Academic Publishers, Boston, 1986. by Leland B. Jackson. ill. ; 25 cm. Includes index.

[72] E. Jury and B. Lee. The absolute stability of systems with many nonlinearities. *Automat. Remote Contr.*, 26:943–961, 1965.

[73] J. Kang and W. Sung. Fixed-Point C Compiler for TMS320C50 Digital Signal Processor. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, april 1997.

[74] J. Kauraniemi. Analysis of limit cycles in the direct form delta operator structure by computer-aided test. *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97, 1997 IEEE International Conference on*, 3:2177–2180 vol3, 1997.

[75] H. Keding. Pain Killers for the Fixed-Point Design Flow. Technical report, Synopsys, 2010.

[76] H. Keding, M. Willems, M. Coors, and H. Meyr. FRIDGE: A Fixed-Point Design and Simulation Environment. In *Design, Automation and Test in Europe*, pages 429–435, Paris, France, 1998.

[77] S. Kim, K.-I. Kum, and W. Sung. Fixed-point optimization utility for C and C++ based digital signal processing programs. *IEEE Transactions on Circuits and Systems II - Analog and Digital Signal Processing*, 45(11):1455 –1464, nov 1998.

[78] S. Kim and W. Sung. A Floating-point to Fixed-point Assembly program Translator for the TMS 320C25. *IEEE Transactions on Circuits and Systems*, 41(11):730–739, nov. 1994.

[79] A. Kinsman and N. Nicolici. Bit-width allocation for hardware accelerators for scientific computing using sat-modulo theory. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(3):405–413, 2010.

[80] A. Kinsman and N. Nicolici. Automated range and precision bit-width allocation for iterative computations. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(9):1265–1278, 2011.

[81] A. Kinsman and N. Nicolici. Computational vector-magnitude-based range determination for scientific abstract data types. *Computers, IEEE Transactions on*, 60(11):1652–1663, 2011.

[82] K. Kum, J. Kang, and W. Sung. AUTOSCALER for C: An optimizing floating-point to integer C program converter for fixed-point digital signal processors. *IEEE Transactions on Circuits and Systems II - Analog and Digital Signal Processing*, 47(9):840–848, sept. 2000.

[83] T. Laakso, P. Diniz, I. Hartimo, and J. Macedo, T.C. Elimination of zero-input and constant-input limit cycles in single-quantizer recursive filter structures. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 39(9):638–646, 1992.

[84] D.-U. Lee, A. Gaffar, R. Cheung, W. Mencer, O. Luk, and G. Constantinides. Accuracy-Guaranteed Bit-Width Optimization. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 25(10):1990–2000, 2006.

[85] D.-U. Lee, A. Gaffar, O. Mencer, and W. Luk. Minibit: bit-width optimization via affine arithmetic. In *Design Automation Conference, 2005.*, pages 837–840, 2005.

[86] D.-U. Lee and J. Villasenor. A bit-width optimization methodology for polynomial-based function evaluation. *Computers, IEEE Transactions on*, 56(4):567–571, 2007.

[87] A. Lepschy, G. Mian, and U. Viaro. Stability analysis of second-order direct-form digital filters with two roundoff quantizers. *IEEE Trans. Circuits Syst.*, 33(8):824–826, 1986.

[88] G. Li, M. Gevers, and Y. Sun. Performance analysis of a new structure for digital filter implementation. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 47(4):474–482, 2000.

[89] G. Li and Z. Zhao. On the generalized dfiit structure and its state-space realization in digital filter implementation. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 51(4):769–778, 2004.

[90] J. Lopez. *Evaluacion de los Efectos de Cuantificacion en las Estructuras de Filtros Digitales Utilizando Tecnicas de Cuantificacion Basadas en Extensiones de Intervalos*. PhD thesis, Univ. Politecnica de Madrid, Madrid, 2004.

[91] J. Lopez, G. Caffarena, and C. Carreras. Fast and accurate computation of the l_2-sensitivity in digital filter realizations. Technical report, Univ. Politecnica de Madrid, 2006.

[92] J. Lopez, G. Caffarena, C. Carreras, and O. Nieto-Taladriz. Analysis of limit cycles by means of affine arithmetic computer-aided tests. In *12th European Signal Processing Conference EUSIPCO'04*, pages 991–994, Vienna (Austria), 2004.

[93] J. 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, august 2008.

[94] J. Lopez, C. Carreras, G. Caffarena, and O. Nieto-Taladriz. Fast characterization of the noise bounds derived from coefficient and signal quantization. In *2003 International Symposium on Circuits and Systems (ISCAS '03).*, volume 4, pages IV–309–IV–312 vol4, 2003.

[95] J. A. Lopez, C. Carreras, and O. Nieto-Taladriz. Improved interval-based characterization of fixed-point lti systems with feedback loops. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(11):1923–1933, 2007.

[96] Mathworks. *Fixed-Point Blockset User's Guide (ver. 2.0)*, 2001.

[97] J. McClellan, C. Burrus, A. Oppenheim, T. Parks, R. Schafer, and H. Schuessler. *Computer-Based Exercises for Signal Processing Using Matlab 5*. Matlab Curriculum Series. Prentice Hall, New Jersey, 1998.

[98] D. Menard, D. Novo, R. Rocher, F. Catthoor, and O. Sentieys. Quantization Mode Opportunities in Fixed-Point System Design. In *Proc. European Signal Processing Conference (EUSIPCO)*, pages 542–546, Aalborg, august 2010.

[99] D. Menard, R. Rocher, P. Scalart, and O. Sentieys. SQNR Determination in Non-Linear and Non-Recursive Fixed-Point Systems. In *European Signal Processing Conference*, pages 1349–1352, 2004.

[100] 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,*, 55(1), November 2008.

[101] D. Menard and O. Sentieys. A methodology for evaluating the precision of fixed-point systems. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, may 2002.

[102] D. Menard and O. Sentieys. Automatic Evaluation of the Accuracy of Fixed-point Algorithms. In *Proc. Design, Automation and Test in Europe (DATE)*, Paris, march 2002.

[103] D. Menard, R. Serizel, R. Rocher, and O. Sentieys. Accuracy Constraint Determination in Fixed-Point System Design. *EURASIP Journal on Embedded Systems*, 2008:12, 2008.

[104] Mentor Graphics. *Algorithmic C Data Types*. Mentor Graphics, v.1.3 edition, march 2008.

[105] W. Mills, C. Mullis, and R. Roberts. Digital filter realizations without overflow oscillations. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 26(4):334–338, 1978.

[106] S. K. Mitra. *Digital signal processing laboratory using MATLAB*. WCB/McGraw-Hill, Boston, 1999. Sanjit K. Kumar. ill. ; 24 cm. + 1 computer disk. System requirements for computer disk: IBM pc or compatible, or Macintosh power pc; Windows 3.11 or higher; MATLAB Version 5.2 or higher; Signal Processing Toolbox Version 4.2 or higher.

[107] S. Mittal. A survey of techniques for approximate computing. *ACM Comput. Surv.*, 48(4):62:1–62:33, Mar. 2016.

[108] A. Nayak, M. Haldar, A. Choudhary, and P. Banerjee. Precision and error analysis of matlab applications during automated hardware synthesis for fpgas. In *Design, Automation and Test in Europe, 2001. Conference and Exhibition 2001. Proceedings*, pages 722–728, 2001.

[109] D. Novo, N. Farahpour, U. Ahmad, F. Catthoor, and P. Ienne. Energy efficient mimo processing: A case study of opportunistic run-time approximations. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1–6. ACM, 2014.

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

[111] W. G. Osborne, J. Coutinho, R. C. C. Cheung, W. Luk, and O. Mencer. Instrumented multi-stage word-length optimization. In *Field-Programmable Technology, 2007. ICFPT 2007. International Conference on*, pages 89–96, 2007.

[112] Y. Pang, K. Radecka, and Z. Zilic. Optimization of imprecise circuits represented by taylor series and real-valued polynomials. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(8):1177–1190, 2010.

[113] K. Parashar, D. Menard, R. Rocher, O. Sentieys, D. Novo, and F. Catthoor. Fast Performance Evaluation of Fixed-Point Systems with Un-Smooth Operators. In *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, 11 2010.

[114] K. Parashar, D. Menard, and O. Sentieys. Accelerated performance evaluation of fixed-point systems with un-smooth operations. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 33(4):599–612, April 2014.

[115] K. Parashar, R. Rocher, D. Menard, and O. Sentieys. Analytical Approach for Analyzing Quantization Noise Effects on Decision Operators. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1554–1557, Dallas, march 2010.

[116] K. K. Parhi. *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley, New York, 1999. Keshab K. Parhi. ill. ; 25 cm. "A Wiley-Interscience publication.".

[117] S. Parker and P. Girard. Correlated noise due to roundoff in fixed point digital filters. *Circuits and Systems, IEEE Transactions on*, 23(4):204–211, 1976.

[118] K. Premaratne, E. Kulasekere, P. Bauer, and L.-J. Leclerc. An exhaustive search algorithm for checking limit cycle behavior of digital filters. *Signal Processing, IEEE Transactions on*, 44(10):2405–2412, 1996.

[119] R. A. Roberts and C. T. Mullis. *Digital Signal Processing*. Addison-Wesley series in electrical engineering. Addison-Wesley, Reading, Mass., 1987. Richard A. Roberts, Clifford T. Mullis. ill. ; 24 cm. Includes index.

[120] R. Rocher, D. Menard, N. Herve, and O. Sentieys. Fixed-Point Configurable Hardware Components. *EURASIP Journal on Embedded Systems*, 2006:Article ID 23197, 13 pages, 2006. doi:10.1155/ES/2006/23197.

[121] R. Rocher, D. Menard, P. Scalart, and O. Sentieys. Analytical accuracy evaluation of Fixed-Point Systems. In *Proc. European Signal Processing Conference (EUSIPCO)*, Poznan, September 2007.

[122] R. Rocher, D. Menard, P. Scalart, and O. Sentieys. Analytical approach for numerical accuracy estimation of fixed-point systems based on smooth operations. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, PP(99):1 –14, 2012.

[123] R. Rocher and P. Scalart. Noise probability density function in fixed-point systems based on smooth operators. In *Proc. Conference on Design and Architectures for Signal and Image Processing (DASIP 2012)*, pages 1–8, oct. 2012.

[124] O. Sarbishei and K. Radecka. On the fixed-point accuracy analysis and optimization of polynomial specifications. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 32(6):831–844, 2013.

[125] O. Sarbishei, K. Radecka, and Z. Zilic. Analytical optimization of bit-widths in fixed-point lti systems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 31(3):343–355, 2012.

[126] C. Shi and R. Brodersen. A Perturbation Theory on Statistical Quantization Effects in Fixed-Point DSP with Non-Stationary Inputs. In *IEEE Int. Conf. on Circuits and Systems*, volume 3, pages 373–376 Vol.3, 2004.

[127] C. Shi and R. Brodersen. Floating-point to fixed-point conversion with decision errors due to quantization. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, may 2004.

[128] V. Singh. An extension to jury-lee criterion for the stability analysis of fixed point digital filters designed with two's complement arithmetic. *IEEE Trans. Circuits Syst.*, 33(3):355, 1986.

[129] 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.

[130] M. Stephenson, J. Babb, and S. Amarasinghe. Bitwidth analysis with application to silicon compilation. In *SIGPLAN conference on Programming Language Design and Implementation*, pages 108–120, 2000.

[131] J. Stolfi and L. d. Figueiredo. Self-validated numerical methods and applications. In *21st Brazilian Mathematics Colloquium, IMPA*, Rio de Janeiro, Brazil, 1997.

[132] W. Sung. Optimization of number representations. In S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala, editors, *Handbook of Signal Processing Systems*. Springer, third edition, 2018.

[133] V. Tavsanoglu and L. Thiele. Optimal design of state - space digital filters by simultaneous minimization of sensitivity and roundoff noise. *Circuits and Systems, IEEE Transactions on*, 31(10):884–888, 1984.

[134] L. Thiele. Design of sensitivity and round-off noise optimal state-space discrete systems. *Int. J. Circuit Theory Appl.*, 12:39–46, 1984.

[135] L. Thiele. On the sensitivity of linear state-space systems. *Circuits and Systems, IEEE Transactions on*, 33(5):502–510, 1986.

[136] K. Uesaka and M. Kawamata. Synthesis of low coefficient sensitivity digital filters using genetic programming. In *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, volume 3, pages 307–310 vol3, 1999.

[137] K. Uesaka and M. Kawamata. Heuristic synthesis of low coefficient sensitivity second-order digital filters using genetic programming. *Circuits, Devices and Systems, IEE Proceedings*, 148(3):121–125, 2001.

[138] K. Uesaka and M. Kawamata. Evolutionary synthesis of digital filter structures using genetic programming. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 50(12):977–983, 2003.

[139] P. Vaidyanathan and V. Liu. An improved sufficient condition for absence of limit cycles in digital filters. *IEEE Trans. Circuits and Systems*, 34(3):319–322, 1987.

[140] S. Wadekar and A. Parker. Accuracy sensitive word-length selection for algorithm optimization. In *Computer Design: VLSI in Computers and Processors, 1998. ICCD '98. Proceedings., International Conference on*, pages 54–61, 1998.

[141] B. Widrow. Statistical Analysis of Amplitude Quantized Sampled-Data Systems. *Transaction on AIEE, Part. II: Applications and Industry*, 79:555–568, 1960.

[142] B. Widrow, I. Kollar, and M.-C. Liu. Statistical theory of quantization. *Instrumentation and Measurement, IEEE Transactions on*, 45(2):353–361, 1996.

[143] M. Willems. *A Methodology for the Effiient Design of Fixed-Point Systems*. PhD thesis, Aachen University of Technology, German, 1998.

[144] N. Wong and T.-S. Ng. A generalized direct-form delta operator-based iir filter with minimum noise gain and sensitivity. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 48(4):425–431, 2001.

[145] B. Wu, J. Zhu, and F. Najm. An analytical approach for dynamic range estimation. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pages 472–477, San Diego, june 2004.

[146] B. Wu, J. Zhu, and F. Najm. Dynamic range estimation for nonlinear systems. In *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 660–667, 2004.

[147] C. Xiao. Improved l_2-sensitivity for state-space digital system. *Signal Processing, IEEE Transactions on*, 45(4):837–840, 1997.

[148] Z. Zhao and G. Li. Roundoff noise analysis of two efficient digital filter structures. *Signal Processing, IEEE Transactions on*, 54(2):790–795, 2006.