Fast performance evaluation of fixed-point systems with un-smooth operators

Karthick Parashar, Daniel Menard, Romuald Rocher, Olivier Sentieys, David Novo, Francky Catthoor

Fast performance evaluation of fixed-point systems with un-smooth operators

K. Parashar, D. Menard, R. Rocher, O. Sentieys
INRIA/IRISA, University of Rennes
6 rue de Kerampont
F-22300 Lannion
parashar@irisa.fr

D. Novo, F. Catthoor
IMEC vzw
Kapeldreef 75
B-3001 Leuven
novo@imec.be

ABSTRACT

Fixed-point refinement of signal processing systems is an essential step performed before implementation of any signal processing system. Existing analytical techniques to evaluate performance of fixed-point systems are not applicable to the errors due to quantization in the presence of un-smooth operators. Thus, it is inevitable to use simulation to evaluate performance of fixed-point systems in the presence un-smooth operators. This paper proposes a hybrid technique which can be used in place of pure simulation to accelerate the performance evaluation. The principle idea in the proposed hybrid approach is to selectively simulate parts of the system only when un-smooth errors occur but use analytical results otherwise. The acceleration thus obtained reduces the performance evaluation time which can be used to explore a wider word-length design space or speedup the optimization process. This method has been tried on a complex MIMO sphere decoding algorithm and the results obtained show several orders of magnitude improvement in terms of evaluation time.

Keywords

Fixed-point system design, Word-length Optimization, Quantization noise, Performance evaluation

1. INTRODUCTION

Signal processing algorithms are rapidly finding use in mobile and hand-held electronic gadgets. While these gadgets cater to applications in niche areas, they are severely constrained by cost, power and the response time. The choice of software platform with sub-word parallelism ASIP(application specific instruction set processor) or custom hardware platforms over general purpose platforms is inevitable especially when considering stringent power and execution time specifications. An important step in mapping a signal processing algorithm on DSP or hardware platform is to refine the signal representation in fixed-point. This step is popularly referred to as the floating-point to fixed-point conversion. The choice of appropriate wordlength essentially trades off processing accuracy of the system to the implementation cost. In fact, use of fixed-point operations has been one of the driving forces in cutting edge processor technology (e.g. the MMX extension on Intel processors). Many practical systems have known to be benefited from the use of fixed-point operations in place of floating-point operations. It has become common knowledge that the use of fixed-point arithmetic is truly beneficial and it is prevalent even in hardware design paradigms. Recent attempts such as [1] have been made to even automate the process of data-path optimization with fixed-point systems to aid hardware designers.

Word-length optimization process is known to be NP-hard [2] and thus time consuming. Some experiments [3] have shown that the manual fixed-point conversion process can consume 25% to 50% of the total product development time. In a survey [4], the fixed-point conversion was identified as one of the most difficult aspects of hardware implementation on FPGA platforms. Algorithms used to solve this problem involve repeated cost and performance evaluation of the system under consideration in an iterative fashion. Defining the system level performance metric and evaluating the same requires a thorough understanding of the functioning of the system. Indeed, the performance evaluation step happens to be the bottleneck resulting in long optimization times which increases with system complexity.

One way of evaluating performance of DSP algorithms with different fixed-point formats is by fixed-point simulation. Though simulation can theoretically be performed on any kind of system, the long simulation time is a limitation and simulation is not always practical especially when it is required to explore the entire fixed-point design space. Analytical models have been successful in providing a closed form analytical expression for the quantization noise power or bounds on the quantization noise amplitude for any given fixed-point specification in case of certain kinds of signal processing systems. Performance metric is derived from the quantization noise power or bounds on quantization noise from relevant signals. The analytical models assume the popular Widrow quantization noise model [9] to characterise and linear models to propagate the noise generated from every operator source across the system. The linear approximation for noise propagation has proven to be accurate when the noise power is small in comparison to signal power and

it need not be true always. Hence, those operators whose noise generation confirm to the Widrow quantization model are referred to as smooth operators and the ones that do not are referred to as un-smooth operators. Decision-making operators or saturation operators are typical un-smooth operators. A good example is the decision-making operator which is vastly used to identify the symbol transmitted at the receiving end in digital communication systems. The errors from decision-making operators are un-smooth and exhibit non-linearity in the sense of discontinuity in its output. Other examples that occur commonly are the saturation and over-flow arithmetic errors.

In this paper, a hybrid technique which makes use of the analytical models to accelerate performance evaluation by simulation is proposed. This method reduces the time taken for performance evaluation in comparison with technique which employs only simulation by several orders of magnitude. The principle used here is to simulate the algorithm in entirety or in parts only when un-smooth errors occur and use the analytical results otherwise. Moreover, floating-point simulation is used even in case of un-smooth errors thereby avoiding fixed precision simulation completely. The acceleration achieved by this technique can be used to reduce the total optimization time or explore a larger fixed point search space so as to improve the quality of the solution.

In the rest of the paper, the next section provides a brief background on the works related to accelerating simulation-based performance and other performance evaluation approaches. Section 3 describes the hybrid method which makes use of analytical models to accelerate simulation-based performance. In section 4 a case study of a highly parallel MIMO ML-decoder is used to study the efficacy of the proposed hybrid method. Section 5 shows the results obtained by using this method on the MIMO decoder. The paper concludes with discussion on the results obtained and prospects the use of this technique in various other scenarios.

2. RELATED WORKS

Measuring the impact on performance of systems with fixed-point arithmetic is a two fold problem. The dynamic range and the quantization noise due to limited precision are two different parts of the fixed-point specification. Improper dynamic range estimates lead to un-smooth errors due to saturation or wrap-around. Such fixed-point word-length induced un-smoothness can be controlled, localised or even eliminated by allowing for large dynamic range or by choosing appropriate fixed-point formats. On the other hand, there are errors due to lack of precision in the fixed-point format which get magnified due to non-linearity of the un-smooth operators such as decision-making operators.

Many approaches for performance evaluation based on simulation and analytical models have been proposed for design and analysis of fixed-point effects. The quantization noise error characteristics are explained in the Widrow [9] noise model. Attempts such as [7] use the standard linear systems theory to propagate the quantization noise power across the system such that quantization noise on any signal in the system can be estimated analytically. These analytical models for quantization noise propagation are restricted to smooth operators and are not usable when un-smooth errors occur. A decision operator which essentially defines a boundary between any two regions (for e.g. less than zero or greater than equal to zero) can be thought of as the basic form of un-smoothness due to discontinuity in the range of the output. A good example for this type of decision maker is the QAM (Quadrature Amplitude Modulation) constellation mapper. Two approaches [8] and [6] attempt to study the generation of such errors. While the former approach defines a bound on error probability in the simple BPSK case, the latter is capable of generating a probability mass function corresponding to error for any kind of decision boundaries associated with the operator. However, both approaches do not handle propagation of decision errors and do not add value when error needs to be propagated.

When none of the analytical techniques are usable, it is inevitable that the performance of fixed-point systems are analysed by way of simulation. Many techniques such as [10] have been proposed in the past to accelerate simulation by efficient code generation. This approach attempts to generate efficient binary for the given system that needs to be simulated such that simulation takes the least possible time. It is easy to see that such simulation acceleration efforts deliver when large parts of the system are implemented in low level software languages and are hence costly. One way of solving this problem is by rapid prototyping which essentially means automatic native code generation. Even this idea is essentially that of simulation acceleration by generating code in a language which can later be compiled to native binary code. Popular tools such as Matlab [11] have this process fully automated and is hence quite .

To the best of the authors knowledge, there has been no effort to use the analytical models to accelerate the process of simulation. This paper proposes a hybrid methodology which selectively simulates the system. The single noise source model which is derived using analytical techniques are used to accelerate simulation of fixed-point behavior for smooth operations. The un-smooth operators are simulated only when deemed necessary.

3. HYBRID TECHNIQUE DESCRIPTION

The proposed hybrid can be thought of as a technique for fixed-point simulation acceleration by using analytical methods. As with any simulation-based techniques, the system is simulated with infinite precision just once and the results are stored in a suitable database to serve as reference for comparison. The single noise source model abstracts away the quantization noise behavior at the sub-system level. They are used in place of simulation of the actual system for every simulation point for all smooth operations thus simplifying simulation of smooth operators which can be handled analytically.

3.1 Single noise source model

The single noise source model abstracts away the noise generated from within a block of smooth operations as a single additive noise source at the output. Consider a sub-system B in a hierarchically defined system $\mathbb{B}$ with input x and output s as shown in Figure 1. The noises b_x and b_s are associated with signals x and s respectively.

The total quantization noise b_s at the output of the sub-

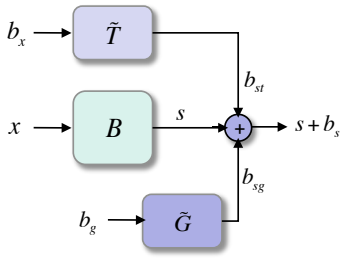


Figure 1: Modelling with single noise source

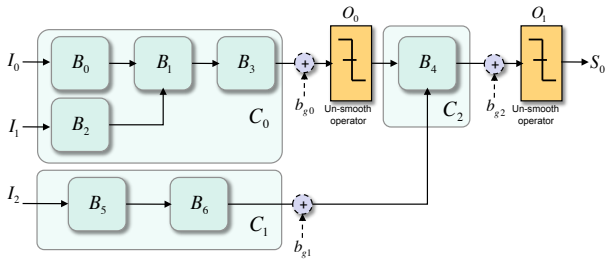


Figure 2: A representative signal processing system

system consists of two components b_{sg} and b_{st} for the noise generated within the system and noise transmitted through the system respectively. The formulation of the single noise source model is based on the analytical techniques developed for the purpose of smooth operations. Naturally, all kinds of smooth operations including non-linear (but still smooth) operations can be captured in the realm of the single noise source model. Indeed, the proposed hybrid technique exploits this feature to make good use of the analytical techniques by way of clustering all smooth operators in the system.

The noise $b_x(n)$ associated with the input signal $x(n)$ is assumed to be uncorrelated with the signal. Further, it is also assumed that the noise power is very small in comparison to the signal. The effect of this noise at the output b_{st} is obtained by passing it through the noise propagation filter $\tilde{T}$ which modifies the power spectrum of b_x like the sub-system B. The noise b_{sg}, generated in the sub-system B is modeled by passing the single noise source b_g through the noise generation filter $\tilde{G}$. The noise generation filter shapes the spectral characteristics of the noise to represent the effect of quantization noise generated within the sub-system B. It can been shown that the output PDF is not uniform and is in fact closer to being a Gaussian due to central limit theorem. Hence, noise source b_g is conveniently modeled as white Gaussian whenever the single noise source model is used.

The setting up of single noise source model for each cluster of smooth operators in a given system is a one-time effort. This model can be re-used by plugging in different word-length values during the course of system evaluation.

3.2 Clustering smooth operators

Consider a system $\mathbb{B}$ with predefined subsystems B_i as shown in Figure 2. The system is made up of N_b subsystems B_i each of them consisting of only smooth operations and N_o un-smooth operators O_j. The smooth sub-systems can be grouped together at the boundaries of un-smooth operators to form smooth clusters such that large parts of the system may be handled analytically.

The sub-systems are grouped to form clusters as shown in Figure 2. The sub-systems B_0 through B_3 are combined together to make the cluster C_0. A single noise source b_{g0} which can mimic the fixed-point behavior of blocks B_0,through B_3 is used to simulate the fixed-point behavior of the smooth cluster thus formed and presented at the input of the un-smooth operator O_0. Similarly, cluster C_1 is formed which includes sub-blocks B_5 and B_6 with the single noise source b_{g1}. One of the inputs to B_4 is sourced from the un-smooth operator O_0. As mentioned earlier, the existing analytical error models do not allow propagation of the errors due to un-smooth operators. Therefore the block B_4 is a separate cluster C_2 and is not made a part of the cluster C_1. The single noise source model can capture any kind operators including non-linear as long as they are smooth.

It has to be noted that in the absence of un-smooth errors, the presence of un-smooth error does not impact quantization noise propagation across the system essentially rendering the system smooth as far as noise propagation is concerned.

3.3 Evaluation strategy

A pure simulation based approach for evaluating performance of the system essentially consists of simulating all the operators in the system for every sample of input data. Floating point simulation is typically used as reference to compare the fixed-point performance degradation and it is a one time effort which can be considered a pre-processing step. Floating-point simulation is used to capture the signals of interest which are stored in the signal data base.The proposed hybrid technique which can be used for accelerated simulation is described in Algorithm 1.

To begin with, the given system is divided into smooth clusters and un-smooth operators and represented in a convenient cluster graph structure $\mathbf{G}(\mathbf{V}, \mathbf{E})$. In this graph representation, the smooth clusters and un-smooth operators form the set of nodes $\mathbf{V}$ and the signals connecting them are the set of edges $\mathbf{E}$. The graph is a directed edge graph whose direction is determined by the direction of the flow of signal. It is proposed to use an evaluate and propagate strategy by following the precedence imposed by the graph structure to evaluate the system. The idea is to follow the precedence of the graph structure to evaluate the given node either by simulation or analytical means and propagate the accumulated noise to subsequent nodes connected to the out-edges of the evaluated node. A limitation imposed on the cluster graph $\mathbf{G}$ due to the evaluate and propagate strategy is that it can be applied on directed acyclic graph (DAG) only.On every smooth node in the cluster graph $\mathbf{G}$, the single noise source model is derived analytically so as to mimic the fixed-point word-length effects at the output of the node. The single noise source model augments the cluster graph and participates only during fixed-point performance evaluation of the system. Given the bit widths of the operators inside the smooth clusters, the noise powers at the input of

every un-smooth operator node is calculated and the respective single noise source models initialized with the calculated noise powers.

Algorithm 1 Accelerated Evaluation

```
Identify Smooth Clusters;
Obtain G(V,E) cluster graph (DAG);
Derive analytical Single Noise Source Models;
Set precision bit widths for all operator sources;
Initialise all single noise source models so as to mimic the
set precision bit width effects;
while Traversing G(V,E) in Precedence order do
  for all un-smooth V_i do
    if Input Signal S_{V_i} along E_i in error boundary then
      Traceback and generate noise to evaluate S_{V_i} with
      SNS model;
      Simulate un-smooth operator;
      if Un-smooth Error then
        Calculate erroneous value;
        Simulate the rest of nodes with precedence de-
        pendence on V_i;
      else
        No un-smooth error, propagate noise;
      end if
    else
      No un-smooth error, propagate noise;
    end if
  end for
end while
```

The DAG $\mathbf{G}(\mathbf{V}, \mathbf{E})$ is traversed in the direction of the signal flow from the input to the output to cover all the clusters and un-smooth operators satisfying the precedence constraints in the process. At the input of an un-smooth operator, all the smooth blocks contributing to that input would have been evaluated and the resulting quantization noise power is calculated. When the smooth clusters contain delays and a decision error occurs at the input, enough care has to be taken such that the actual sample where the decision error occurred is simulated. For example, if the smooth cluster contains memory and cycles with memory, the points after decision error is presented at the output after M samples. In case of cycles inside the cluster, enough samples (N) must be simulated such that the effect of a decision error does not prevail after these many samples. In other words, all the samples that are affected by the decision errors need to be simulated.

Consider a signal S_i at the input of an un-smooth operator in an infinite precision system. The corresponding signal $\tilde{S}_i$ in the case of a finite precision system is obtained as the sum of the signal S_i with the noise b_{si} generated until that signal with the help of single noise source models ($\tilde{S}_i = S_i + b_{si}$). An un-smooth error is said to have occurred in case the output of the operator for signals in infinite precision and fixed precision are different (i.e. $O(\tilde{S}_i) \neq O(S_i)$). The function $O()$ corresponds to the un-smooth operator. Evaluation of the un-smooth operator corresponds to the simulation of un-smooth operator. In case of an un-smooth error, the nodes on the path connected to the output of the un-smooth operators (immaterial whether they are smooth or otherwise) all the way to the output have to be simulated. If there is no error, the accumulated quantization noise is suitably propagated (with a gain of 1 or 0) to the subsequent nodes along the path. For example, a decision-making operator has a gain of 0 whereas a saturation or over-flow operation has a gain of 1 in case of no un-smooth errors.

To determine whether an un-smooth error has occurred, the noise characteristics of b_{si} is used to determine a boundary around the un-smooth region for every un-smooth operator O_i in the system. This boundary is with respect to the value of signal S_i. If the signal S_i happens to be confined in that boundary, there is no need to even simulate the un-smooth operator. When $\overline{b_{si}}$ is defined as the maximum tolerable quantization noise power, if $O(S_i \pm \overline{b_{si}}) = O(S_i)$ then it can be guaranteed that no decision error occurs and there is no need to simulate the un-smooth operator $O_i()$. For example, when the noise PDF is known to be a Gaussian ($N(0, \sigma)$), the noise boundary can be defined as $\mu \pm 5 \times \sigma$ where $\overline{b_{si}} = 5 \times \sigma$. If the signal S_i is placed outside the boundary, such that the addition of quantization noise might cause un-smooth errors, the signal $\tilde{S}_i$ is calculated by generating the random value b_{si} from the single noise source model and the un-smooth operator O_i is simulated to check for error. In both cases, the value $O(S_i)$ is obtained from the database of signal values obtained during infinite precision simulation.

3.4 Complexity Analysis

The time taken for optimization using such an approach is of interest when comparing the proposed hybrid technique with pure simulation.

In a pure simulation-based scenario where N_p number of input samples are present, if the optimization demands N_i number of iterations and if t_{sim} is the time taken for one fixed-point sample simulation, the total time for optimization can be estimated as

$$T_{fopt} = N_i.N_p.t_{sim}. \tag{1}$$

In the proposed mixed approach, a finite time is spent performing analytical estimation of the noise powers followed by choosing the points that need to be simulated as shown in Algorithm 1 is followed. Therefore, the total time for optimization can be written as

$$T_{mopt} = t_{sns} + N_i(N_a.t_{ana} + N_s.\tilde{t}_{sim}) \tag{2}$$

where t_{sns} is the time for performing the single noise source analysis, $\tilde{t}_{sim}$ is the average time taken for partial or full simulation of the system averaged over N_s samples. The value of $\tilde{t}_{sim}$ takes on the value anywhere between 0 and t_{sim} and is influenced by the number of decision errors when evaluating the input set. t_{ana} is the time for analytical evaluation. t_{ana} is very small as it involves generation of random numbers which is quite simple in comparison to the actual computation which is performed only if the signal happens to be in the error boundary according to the single noise source model and the generated signal. Moreover, it is typical to think of $N_s << N_a$ since decision errors are not very common. Some of the samples N_s are simulated while the others N_a are handled analytically. In other words, the total

number of samples is split between simulation and analytical modelling based evaluation ($N_p = N_s + N_a$). Also, the time taken for simulation $\tilde{t}_{sim}$ is lesser than or equal to t_{sim}. Even when there is a decision error very early in the system which requires the entire system to be simulated, $\tilde{t}_{sim} < t_{sim}$ on some evaluation platforms as the floating-point simulation is used without wasting time for emulation of fixed-point data.

The time spent for analytical estimation t_{sns} is a one-time effort performed once for a given signal flow graph. Thus in successive optimization efforts, this factor does not have any contribution to the time taken. The benefit obtained by following a mixed approach can be quantified by an improvement factor (IF) in the context of Equation 2 as

$$T_{mopt} = t_{sns} + N_i.N_p.\frac{t_{sim}}{IF}. \quad (3)$$

where, the improvement factor IF is defined as the ratio

$$IF = \frac{N_p.t_{sim}}{N_a.t_{ana} + N_s.\tilde{t}_{sim}}. \quad (4)$$

It can be seen that the improvement factor is a positive quantity with potentially large values. It deteriorates when there are too many decision errors under noisy conditions. It is expected that such pathological situations seldom arise.

4. CASE STUDY

The proposed method is generic and it can be applied to any system with un-smooth operators. To study the efficiency of the proposed hybrid technique,a Multiple Input, Multiple Output (MIMO) decoding algorithm is chosen to showcase the effectiveness of the proposed method.

The MIMO sphere decoding algorithm uses a decision operator (QAM mapper) to explore different combination of symbols transmitted. Therefore, it is possible to have different configurations of the decoder algorithm with desired number of un-smooth decision-making operators and to analyse the effects of a number of decision operators. Apart from a number of decision operators, there is a fair amount of computations to be performed at every antenna with a smooth cluster always separating any two decision operators. Interestingly, it can also be seen from the signal flow graph in Figure 3, the decision errors are correlated with one another. Moreover, it also contains a $min()$ operator which is again un-smooth and always requires simulation. Thus with the MIMO decoder, it is possible to have a good mixture of smooth and un-smooth operations together with a lot of control in choosing this ratio. The correlation between decision errors from different operators makes it even more complicated and impossible to be handled with existing analytical techniques. We believe it is more relevant to explore the chosen case study to demonstrate the proposed method than benchmark algorithms on which there is few control.

4.1 SSFE Description

In most recent standards, such as IEEE802.11n, Mobile WiMAX or 3GPP-LTE, the drastic increase in throughput comes at the expense of complex MIMO detectors. Although linear detection methods are mostly implemented so far, its BER(Bit Error Rate) performance is rather poor. At the other extreme, the Maxim Likelihood (ML) detection solves

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\Omega^{N_T}}{\arg\min} \|\mathbf{y} - \mathbf{H}.\mathbf{s}\|^2 \quad (5)$$

where $\mathbf{H}$ denotes the N_R x N_T channel matrix, N_R and N_T correspond to the number of receive and transmit antennas and $\mathbf{s} = [s_1, s_2, ..., s_{N_T}]^T$ is the N_T-dimensional transmit signal vector. The entries of $\mathbf{s}$ are chosen independently from a complex constellation Ω. The set of all possible transmission vector symbols is denoted by Ω^{N_T}. ML detection provides the optimum detection method and minimizes BER. A straightforward way to solve Equation 5 is an exhaustive search. However, the corresponding computational complexity grows exponentially with the number of transmit antennas and with the number of bits per constellation symbol, making its implementation unfeasible for the mentioned high throughput standards.

With the intention of finding a good compromise between implementation complexity and performance, heuristics that approximate ML detection, namely near-ML, have recently gathered relevant attention from the algorithmic community. In this context, the Selective Spanning with Fast Enumeration [5] (SSFE) shows to achieve considerable BER performance with a decent implementation complexity. The SSFE relays in the triangulation of the channel matrix $\mathbf{H}$ using a QR decomposition according to $\mathbf{H} = \mathbf{QR}$, where the N_R x N_T matrix $\mathbf{Q}$ has orthogonal columns and the N_T x N_T matrix $\mathbf{R}$ is upper triangular. Thus,

$$\|\mathbf{y} - \mathbf{Hs}\|^2 = \kappa + \|\hat{\mathbf{y}} - \mathbf{Rs}\|^2 \quad \text{with} \quad \hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y} \quad (6)$$

where κ is a constant independent of the vector symbol $\mathbf{s}$ and can hence it will be consider to be zero in the following.

A tree can be build such that the leaves at the bottom correspond to all possible vector symbols $\mathbf{s}$ and the possible values of the entry s_{N_T} define its top level. Then, each node at level i ($i = 1, 2, ..., N_T$) can uniquely be described by the partial vector symbols $\mathbf{s}^i = [s_i, s_{i+1}, ..., s_{N_T}]^T$. Accordingly, Equation 6 can be rewritten as

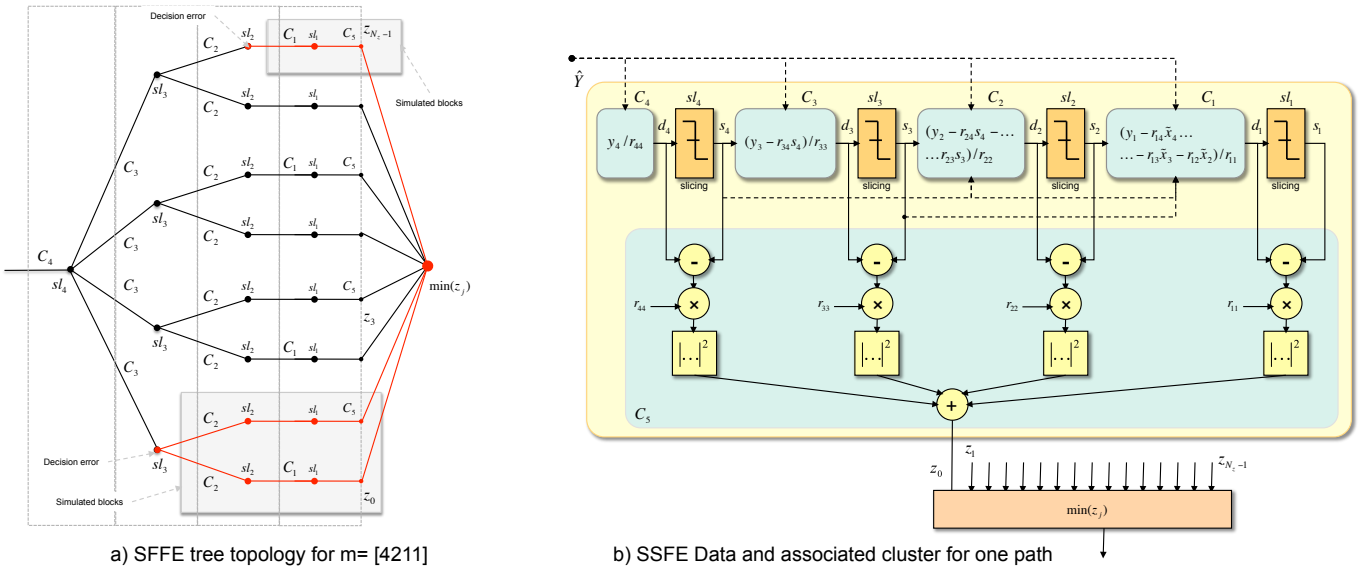
$$\|\hat{\mathbf{y}} - \mathbf{Hs}\|^2 = \sum_{i=1}^{N_T} \| \hat{y_i} - \sum_{j=1}^{N_T} r_{ij}s_j \|^2 = \sum_{i=1}^{N_T} \|e_i(s^i)\|^2 \quad (7)$$

where $\|e_i(s^i)\|^2$ corresponds to the partial squared euclidean distance increment at the node i. In order to reduce the computations, the SSFE algorithm reduces the enumeration of S to a subset of Ω^{N_T}. This selective enumeration happens when traversing the tree to the leaves. To enable the selective enumeration the partial squared euclidean distance is rewritten as:

$$\|e_i(s^i)\|^2 = \| \hat{y_i} - \sum_{j=i+1}^{N_T} r_{ij}s_j - r_{ii}s_i \|^2 = \| c_i - r_{ii}s_i \|^2 \quad (8)$$

Clearly, the minimization of $\|e_i(s^i)\|^2$ is equivalent to the minimization of $\|e_i(s^i)/r_{ii}\|^2$, hence

$$\|e_i(s^i)/r_{ii}\|^2 = \| c_i/r_{ii} - s_i \|^2 = \| d_i - s_i \|^2 \quad (9)$$



a) SFFE tree topology for m= [4211]

b) SSFE Data and associated cluster for one path

Figure 3: SSFE data flow model and associated smooth clusters

The key element of the SSFE algorithm is the special slicer operator, $\mathcal{Q}(d_i, m_i)$, which generates a vector p with the m_i constellation symbols with the shortest euclidean distance (as defined in Equation 9) to the demodulated symbol d_i. SSFE is a distributed and greedy algorithm which splits the minimization problem of Equation 5 into multiple N_T subsequent minimizations of Equation 9. Each subsequent minimization produces a vector z of N_T constellation symbols. Since more than one candidate can be generated at each subsequent minimization, a total of t ($t = \prod_{i=1}^{N_T} m_i$) different z vectors are produced. Thus, the SSFE algorithm includes a final sorting of the accumulative euclidean distance (defined in Equation 7) of all the z vectors. The z vector leading to the minimal accumulative distance corresponds to the SSFE solution. Depending on the vector parameter m ($m = [m_1, m_2, ..., m_{N_T}]$), the SSFE algorithm can reach any BER performance ranging from the V-BLAST to the ML algorithm.

The SSFE algorithm includes many architecturally favorable features such as deterministic and regular data-flow. This has shown to lead to software and hardware based solutions which consume considerably less area, energy and execution time than other near-ML solutions for a similar BER performance. However, the SSFE detector is still responsible for a dominant share of the overall baseband complexity, specially when high BER performance are targeted.

Figure 3.a shows the tree topology of a four antennas SSFE with $m = [1, 1, 2, 4]$ which results in 8 different z vectors. Each circle represents computations performed in an arithmetic cluster followed by a decision-making operator. The un-smooth operator in these paths are QAM de-mappers or decision operators. Figure 3 illustrates the data-flow of the SSFE algorithm with four receiver antennas corresponding to the processing in one path of the tree.

4.2 SSFE Clustering

Before evaluating the performance with our hybrid technique, the smooth operators are grouped together to form clusters. The SSFE tree diagram shows many paths corresponding to the various permutations of the signals. Each path consists of $N_T + 1$ clusters ($C_1..C_{N_T+1}$). The clustered SSFE data flow graph is as shown in Figure 3.b for the case of four antennas.

The computations performed at any given node is shared across the paths diverging from the same and all the way to the corresponding leaves in the tree. In the case depicted in the Figure 3.a, only few of the clusters are simulated in the proposed hybrid approach. While in a pure simulation approach all the clusters would have to be simulated.

Let $\mathbf{d} = [d_1, d_2, ..., d_{N_T}]^T$ be the vector corresponding to the output of the N_T first clusters. The element d_i corresponding to the output of cluster i is equal to the ratio between c_i and r_{ii} where c_i is the i^{th} element of the vector $\mathbf{c} = [c_1, c_2, ..., c_{N_T}]^T$ computed with the following expression

$$\mathbf{c} = \hat{\mathbf{y}} - \mathbf{R}_t\mathbf{s} = \mathbf{Q}^H\mathbf{y} - (\mathbf{R} - diag(\mathbf{R}))\mathbf{s}. \quad (10)$$

The last cluster C_{N_T+1} computes the accumulative euclidean distance $\mathbf{z}_j$ associated to the path j from the two vectors $\mathbf{d}$ and $\mathbf{s}$ corresponding respectively to the output of the N_T first clusters and the demodulated symbols

$$z_j = \sum_{i=1}^{N_T} |r_{ii}.(d_i - s_i)|^2 \quad (11)$$

4.3 Analytical quantization noise model

The analytical model for the smooth clusters associated to the SSFE algorithm is detailed in this section. The propagation model presented in [7] for the case of quantization noise matrix is used.

Let $\mathbf{b}_{\hat{Y}}$ the N_T-dimensional vector corresponding to the noise associated with the input vector $\hat{\mathbf{y}}$. Let $\mathbf{B_{Rt}}$ be the noise ma-

trix associated with the matrices **Rt** defined in equation 11. The expression of the N_T-dimensional vector $\mathbf{b}_c$ representing the quantization noise obtained after the computation of the vector **c** is as follows

$$\mathbf{b_c} = \mathbf{b}_{\hat{Y}} - \mathbf{B_{R_t}s} + \mathbf{b_{gc}} \tag{12}$$

where $\mathbf{b_{gc}}$ is an N_T-dimensional vector corresponding to the noise generated during the computation of the vector **c**. Given that this model is used only in the case without decision error, no noise is associated to vector **s** corresponding of the output of the QAM de-mappers.

The expression of the N_T-dimensional vector $\mathbf{b}_d$ representing the quantization noise located at the output of the N_T-first clusters is as follows

$$\mathbf{b_d} = \mathbf{R_d}^{-1}\left(\mathbf{b}_{\hat{Y}} - \mathbf{B_{R_t}s} + \mathbf{b_{gc}}\right) + \mathbf{R_d}^{-2}\mathbf{B_{R_d}R_tS} + \mathbf{b_{gd}} \tag{13}$$

where $\mathbf{b_{gd}}$ is a N_T-dimensional vector corresponding to the noise generated during the division operation.

The expression of the quantization noise b_{z_j} located at the output of the last cluster is as follows

$$b_{z_j} = \sum_{i=1}^{N_T} 2Re\left(b_{e_i}\right) Re(d_i) + 2Im\left(b_{e_i}\right) Im(d_i) + b_{ge_i} \tag{14}$$

where $\mathbf{b_{ge}}$ is a N_T-dimensional vector corresponding to the noise generated during the computation of last cluster C_{N_T+1} and where $\mathbf{b_e}$ is a N_T-dimensional vector corresponding to the noise before the square modulus computation

$$b_{e_i} = b_{r_{ii}}.(d_i - s_i) + r_{ii}.b_{d_i} \tag{15}$$

To compute the power expression of the different noises, the technique presented in [7] is used.

5. RESULTS

In any alternative approach to simulation, it has to be taken care that the accuracy of the performance evaluation is not seriously sacrificed. To show that the result of the hybrid technique is close to the result obtained from pure simulation, the experiments were conducted with both Pure simulation (denoted as FP in the Figure) and the proposed Hybrid technique. During the fixed-point simulation, the system was subjected to uniform precision quantization (i.e. all signals have same precision). The results shown in the BER curve in Figure 4 indicate that the quality of BER simulation obtained by the hybrid technique is very high and is only about 10% off even at high channel SNRs. The quality of evaluation is sufficient for the design of fixed-point system. It also follows from the experiments that the number of decision errors due to fixed-point effects measured with simulation and estimated in the hybrid case are very close. It is time consuming to perform pure simulation at low BER conditions. As a proof of concept for the efficiency of the proposed method, high BER (of the order of 10^{-2}) which require relatively less number of samples are used. The trends observed indicate the performance would only increase in low BER conditions.

The time taken by pure simulation and the proposed hybrid method is compared. The actual time for performing clustering and arriving at the closed form expression for single

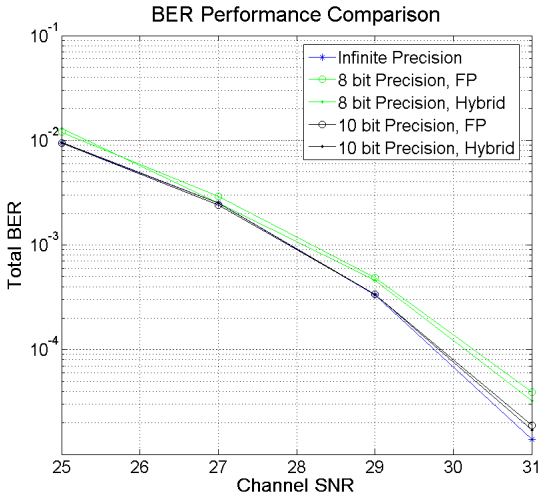


Figure 4: Quality of result obtained with simulation Vs. hybrid technique

noise source models is a one time effort. Therefore, it is not included in the time taken. The time taken to compute the noise power in the single noise source model is seen to be very small. With increasing number of input simulation points, the time taken for performance evaluation grows linearly in both cases according to Equations 1 and 2.

The improvement factor is dependent on the number of decision errors and hence the amount of noise in the system. Thus, the improvement factor in terms of performance evaluation time which is a function of the channel SNR and the quantization noise is plotted on the Figure 5 for three different configurations of the data word-length. It is seen that the improvement increases with reduction in channel and quantization noise. The increasing trend on the log scale as seen in Figure 5 is an indicator of the improvement that can be obtained for low BER simulations.

SNR	$m = [1,2,2,4]$			$m = [1,1,2,4]$			$m = [1,1,1,4]$		
	FP	Hy	IF	FP	Hy	IF	FP	Hy	IF
19 dB	128	110e-3	**1.1e+3**	113	72e-3	**1.5e+3**	98	54e-3	**1.8e+3**
21 dB	128	96e-3	**1.3e+3**	113	72e-3	**1.5e+3**	98	39.e-3	**2.5e+3**
23 dB	129	98e-3	**1.3e+3**	113	67e-3	**1.6e+3**	102	31e-3	**3.2e+3**
25 dB	128	97e-3	**1.3e+3**	113	67e-3	**1.6e+3**	98	32e-3	**3.0e+3**

Table 1: Comparative study of execution times for different SSFE configurations

With increasing channel SNR, the received symbol moves closer to the transmitted symbol and the chances that the quantization noise perturbation would cause a decision error decreases. Thus the Improvement Factor improves with increasing channel SNR from one to several orders of magnitude. When the quantization noise power decreases (with increasing data word-lengths), the number of decision errors in comparison to the floating-point simulation naturally decreases, leading to reduction in simulation effort and hence improving the Improvement Factor. This can be seen across all SSFE configurations in Table 1. The decision operators used in the SSFE algorithm are used to generate proba-

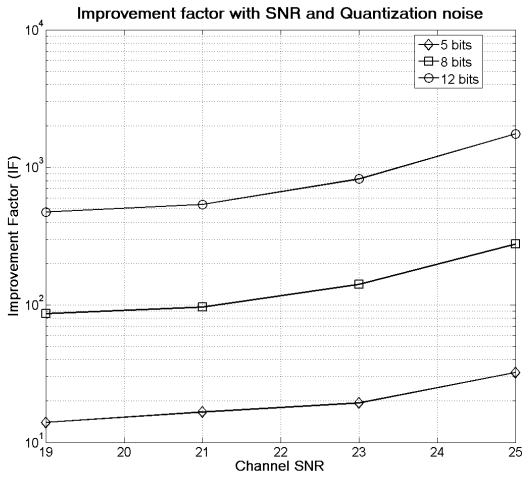


Figure 5: Improvement with hybrid technique in comparison to pure simulation

ble symbol candidates depending upon the proximity of the calculated signal with the constellation and are are further tested upon down the path in the tree. In cases where there are more than one probable symbols (for e.g. the 4^{th} antenna in $m = [1, 1, 2, 4]$ case generates 4 symbols) a small quantization noise perturbs the signal and totally different constellations could be chosen. This is interesting in the case of the proposed hybrid simulation because the choice of probable symbols at the output of decision operator in the hybrid technique can occur in a permuted fashion. Such permutations when considered as decision errors lead to more simulation leading to longer time for the hybrid approach. Instead, a quick optimization can be performed to align such permutations during hybrid simulation to match with the order of the decision operator output obtained from infinite precision simulation. This improves the result obtained further as it now reduces the number of unwanted simulations.

These phenomena can play an important role while choosing the algorithm for word-length optimization. An optimization algorithm which attempts to reduce the number of bits while starting with the maximum number of bits stands to gain by this approach. In contrast, an algorithm which starts with minimum number of bits and tries to improve the performance by adding more bits to word-lengths suffer from more decision errors due to high quantization noise.

6. CONCLUSION AND PROSPECTS

In this paper, the problem of performance estimation of fixed-point systems in the presence of un-smooth operators is considered. We propose to accelerate the process of simulation by using analytical techniques that have been developed for systems with smooth operators. This technique can be considered a hybrid technique as it puts together the best of both simulation and analytical techniques.

The principle idea behind this technique is to be able to perform simulation selectively only when un-smooth errors occur but use analytical simulation otherwise. The proposed algorithm is shown to maintain sufficient accuracy while improving the performance evaluation time. Three different configurations of the chosen MIMO decoding algorithm were explored to study the performance of the hybrid technique. The experimental results show an improvement of orders of magnitude in the time taken for completing the performance evaluation process over pure simulation. The trends in improvement indicate exponential increase in the improvement factor(IF) for low BER simulation.

The improved performance evaluation time by adopting the proposed hybrid technique can either improve the optimization time or allow the designer to expand the available design space that can be explored during optimization.

7. REFERENCES

[1] G. Caffarena, J. A. Lopez, G. Leyva, C. Carreras, and O. Nieto-Taladriz. Optimized architectural synthesis of fixed-point datapaths. *Reconfigurable Computing and FPGAs, International Conference on*, 0:85–90, 2008.

[2] G. Constantinides and G. Woeginger. The complexity of multiple word-length assignment. 15(2):137–140, 2002.

[3] M. Clark, M. Mulligan, D. Jackson, and D. Linebarger. Accelerating Fixed-Point Design for MB-OFDM UWB Systems. *Comms Design (an EE times Community)*, January 2005.

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

[5] M. Li, B. Bougard, W. Xu, D. Novo, L. Van Der Perre, and F. Catthoor. Optimizing near-ML MIMO detector for SDR baseband on parallel programmable architectures. In *Proc. of DATE*, pages 444–449. ACM, 2008.

[6] K. Parashar, R. Rocher, D. Menard, and O. Sentieys. Analytical approach for analyzing quantization noise effects on decision operators. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'10).*, 10:V–1554–7 vol.10, 14-19 March 2010.

[7] R. Rocher, D. Menard, P. Scalart, and O. Sentieys. Analytical accuracy evaluation of Fixed-Point Systems. In *12th European Signal Processing Conference (EUSIPCO 2007)*, Poznan, Poland, September 2007.

[8] C. Shi and R. Brodersen. Floating-point to fixed-point conversion with decision errors due to quantization. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04).*, 5:V–41–4 vol.5, 17-21 May 2004.

[9] B. Widrow, I. Kollár, and M.-C. Liu. Statistical Theory of Quantization. *IEEE Trans. on Instrumentation and Measurement*, 45(2):353–61, Apr. 1996.

[10] M. Willems, H. Keding, T. Grotker, and H. Meyr. FRIDGE: An Interactive Fixed-Point Code Generation Environment for HW/SW-CoDesign. In *IEEE International Conference on Acoustics, Speech and Signal Processing 1997 (ICASSP 97)*, pages 297–290, Munich, Germany, April 1997.

[11] MEXX compiler. http://www.mathworks.com/.