Optimum Wordlength Search Using Sensitivity Information

Kyungtae Han and Brian L. Evans

Embedded Signal Processing Laboratory, Wireless Networking and Communications Group,
The University of Texas at Austin, Austin, TX 78712, USA

Many digital signal processing algorithms are first developed in floating point and later converted into fixed point for digital hardware implementation. During this conversion, more than 50% of the design time may be spent for complex designs, and optimum wordlengths are searched by trading off hardware complexity for arithmetic precision at system outputs. We propose a fast algorithm for searching for an optimum wordlength. This algorithm uses sensitivity information of hardware complexity and system output error with respect to the signal wordlengths, while other approaches use only one of the two sensitivities. This paper presents various optimization methods, and compares sensitivity search methods. Wordlength design case studies for a wireless demodulator show that the proposed method can find an optimum solution in one fourth of the time that the local search method takes. In addition, the optimum wordlength searched by the proposed method yields 30% lower hardware implementation costs than the sequential search method in wireless demodulators. Case studies demonstrate the proposed method is robust for searching for the optimum wordlength in a nonconvex space.

1. INTRODUCTION

Digital signal processing algorithms often rely on long wordlengths for high precision, whereas digital hardware implementations of these algorithms need short wordlengths to reduce total hardware costs. Determining the optimum wordlength can be time-consuming if assignments of wordlengths are performed by trial and error. In a complex system, 50% of the design time may be spent on wordlength determination [1].

Optimum wordlength choices can be made by solving equations when propagated quantized errors [2] are expressed in an analytical form. However, an analytical form is difficult to obtain in complicated systems. Searching the entire space by simulation guarantees to find optimum wordlength. Computation time, however, increases exponentially as the number of wordlength variables increases. For these reasons, many simulation-based wordlength optimization methods have explored a subset of the entire space [3–7].

Choi and Burleson [3] showed how a general search-based wordlength optimization can produce optimal or near-optimal solutions for different objective-constraint formulations. Sung and Kum [4] proposed simulation-based wordlength optimization for fixed-point digital signal processing systems. These search algorithms try to find the cost-optimal solution by using either "exhaustive" search or heuristics.

Han et al. [5] proposed search algorithms that can find the performance-optimal solution by using "sequential" or "preplanned" search. Those algorithms utilize the distortion sensitivity information with respect to the signal wordlengths at the system output such as propagated quantized error. Those algorithms assume that the hardware cost in each wordlength is the same. However, complicated digital systems such as a digital transceiver possess different cost or complexity in digital blocks.

A new algorithm that considers different hardware costs is proposed in [7]. The new algorithm utilizes the measure of the distortion sensitivity as well as complexity sensitivity. The new algorithm speeds up the search time to find an optimum wordlength by considering performance and cost as the objective function and the update direction.

This paper is organized as follows. In Section 2, related work for floating-point to fixed-point conversion is presented. Section 3 gives the background for wordlength optimization. Various search methods to find optimum wordlength are reviewed in Section 4. New sensitivity measures to

TABLE 1: Fixed-point conversion approaches for integer wordlength (IWL) and for fractional wordlength (FWL) determination.

Analytical approach		Statistical approach	
Range model for IWL	Error model for FWL	Range statistic for IWL	Error statistic for FWL
Wadekar [8]	Constantinides [9]	Cmar [10]	Cmar [10]
Stephenson [11]	Shi [12]	Kim [13]	Kum [14]
Nayak [15]	—	—	Shi [12]

TABLE 2: Optimum wordlength search methods.

Cost sensitivity	Error sensitivity	Nonsensitivity
Local search [3]	Sequential search [5]	Exhaustive search [4]
Evolutive search [6]	Max-1 search [16]	Branch and bound [3]
—	Preplanned search [5]	
Complexity-and-distortion measure search—proposed		

update search directions are generalized in Section 5. Case studies of the optimum wordlength design are presented in Section 6. In Section 7, simulation results are discussed. Section 8 concludes the paper.

2. RELATED WORK

During the floating-point to fixed-point conversion process, fixed-point wordlengths composed of the integer wordlength (IWL) part and the fractional wordlength (FWL) part are determined by different approaches as shown in Table 1. Some published approaches for floating-point to fixed-point conversion use an analytic approach for range and error estimation [8, 9, 11, 12, 15], and others use a statistical approach [10, 12–14]. An analytic approach has a range and error model for integer wordlength and fractional wordlength design. Some use a worst-case error model for range estimation [8, 15], and some use forward and backward propagation for IWL design [11]. Still, others use an error model for FWL [9, 12]. The advantages of analytic techniques are that they do not require simulation stimulus and can be faster. However, they tend to produce more conservative wordlength results.

A statistical approach has been used for IWL and FWL determination. Some use range monitoring for IWL estimation [10, 13], and some use error monitoring for FWL [10, 12, 14]. The work in [12] also uses an error model that has coefficients obtained through simulation. The advantage of statistical techniques is that they do not require a range or error model. However, they often need long simulation time and tend to be less accurate in determining wordlengths.

After obtaining models or statistics of range and error by analytic or statistical approaches, respectively, search algorithms can find an optimum wordlength. Some published methods search optimum wordlength without sensitivity information [3, 4], and with sensitivity information [3, 5, 16] as shown in Table 2. "Exhaustive" search [4] and "branch-and-bound" procedure [3] can find an optimum wordlength without any sensitivity information. However, nonsensitivity methods have an unrealistic search space as the number of wordlengths increases.

Some use sensitivity information to search an optimum wordlength. "Local" search [3] and "evolutive" search in [16] use cost sensitivity information. The advantage of cost sensitivity methods is that they can find an optimum wordlength in terms of cost. "Sequential" search and "preplanned" search in [5] and "Max-1" search in [16] use error sensitivity information. The advantage of employing error sensitivity is that they find the optimum wordlength in terms of error faster than the cost sensitivity method. However, both sensitivity methods do not always reach global optimum wordlength.

Cantin et al. provide a useful survey of search algorithms for wordlength determination. In this work, search algorithms are compared, and the "preplanned search" shows the smallest number of iterations to find a solution. However, the heuristic procedures do not necessarily capture the optimum solution to the wordlength determination problem, due to nonconvexity in the constraint space [9]. Thus, the distance between a global optimum wordlength and a local optimum wordlength searched by algorithms is considered. The proposed method is robust to search a near optimum wordlength. This paper discusses the distance and robustness of the proposed algorithm in Section 7.

3. BACKGROUND

3.1. Fixed-point data format

When designers model at a high-level, floating-point numbers are useful to model arithmetic operations. Floating-point numbers can handle a very large range of values and are easily scaled. In hardware, floating-point data types are typically converted or built as fixed-point data types to reduce the amount of hardware needed to implement the functionality. To model the behavior of fixed-point arithmetic hardware, designers need bit-accurate fixed-point data types.

Fixed-point data consists of an integer part and a fractional part. The number of bits assigned to the integer representation is called the integer wordlength (IWL), and the number of bits assigned to the fraction is the factional wordlength (FWL) [17]. Fixed-point wordlength (WL)

corresponds to the following equation:

$$WL = IWL + FWL. \tag{1}$$

The wordlength must be greater than 0. Given IWL and FWL, fixed-point data represents a value in the range R, with the quantization step Δ as

$$-2^{IWL} \leq R < 2^{IWL}, \quad \text{for signed,}$$
$$0 \leq R < 2^{IWL}, \quad \text{for unsigned}, \tag{2}$$
$$\Delta = 2^{-FWL}.$$

IWL and FWL are determined to prevent unwanted overflow and underflow. IWL can be determined by the following relation:

$$IWL \geq \lceil \log_2 R \rceil. \tag{3}$$

Here, $\lceil x \rceil$ is the smallest integer that is greater than or equal to x. The range, R, can be estimated by monitoring the maximum and minimum value or mean and standard derivation of a signal [13, 18]. FWL can be determined by wordlength optimization or tradeoffs in the design parameters during fixed-point conversion.

3.2. Formulation of the optimum wordlength

The wordlength is an integer value, and a set of n wordlengths in a system is defined to be a wordlength vector, that is, $\mathbf{w} \in \mathbb{I}^n$ such as $\{w_1, w_2, \ldots, w_n\}$. We assume that the objective function f is defined by the sum of every wordlength implementation cost function c as

$$f(\mathbf{w}) = \sum_{k=1}^{n} c_k(w_k), \tag{4}$$

where c_k has a real value so that $c_k : \mathbb{I} \to \mathbb{R}$. The quantized performance function p indicates propagated precision or quantized error and is constrained as follows:

$$p(\mathbf{w}) \geq P_{\text{req}}, \tag{5}$$

where p has a real value so that $p : \mathbb{I} \to \mathbb{R}$, and P_{req} is a constant for a required performance. We also consider the lower bound wordlength $\underline{w}$ and upper bound wordlength $\overline{w}$ as constraints for each wordlength variable:

$$\underline{w}_k \leq w_k \leq \overline{w}_k, \quad \forall k = 1, \ldots, n. \tag{6}$$

The complete wordlength optimization problem can then be stated as

$$\min_{\mathbf{w} \in \mathbf{I}^n} f(\mathbf{w}) \quad \text{subject to } p(\mathbf{w}) \geq P_{\text{req}}, \; \underline{\mathbf{w}} \leq \mathbf{w} \leq \overline{\mathbf{w}}. \tag{7}$$

The goal of the wordlength optimization is hence to search for the optimizer $\mathbf{w}^*$ that minimizes the objective function $f(\mathbf{w})$ in (7).

3.3. Finding the optimum wordlength

One of the algorithms for searching the "optimum" wordlength starts with an initial feasible solution $\mathbf{w}^{(0)}$ and performs an update via

$$\mathbf{w}^{(h+1)} = \mathbf{w}^{(h)} + s\boldsymbol{\xi}^{(h)}. \tag{8}$$

Here, h is an iteration index, s is the integer step size, and ξ is an integer update direction. A sound initial guess, a well-chosen step size, and a well-chosen update direction can reduce the number of iterations to find optimum wordlengths.

Optimum wordlengths can be found by solving equations when the performance function p is expressed in an analytical form. If there is no analytical form to express the performance, then simulation-based search methods can be used to search for optimum wordlengths by measuring the performance function. Typical approaches involve assigning wordlength vector $\mathbf{w}^{(0)}$ to a lower bound, an upper bound, or a vector between the lower and upper bound. Step size can be fixed or adapted. The update direction is adapted according to the search algorithms in Section 4.

During iteration, the stopping criteria are dependent on the search algorithm. The algorithm that starts from the lower bound stops when the performance P reaches the required performance P_{req}. The algorithm that starts from the upper bound stops when P falls below P_{req}. Other algorithms stop when the performance P or cost c converges within a neighborhood.

4. REVIEW OF SIMULATION-BASED SEARCH METHODS

Optimum wordlengths can be found by solving equations when the performance function P is expressed in an analytical form. If there is no analytical form to express the performance, then simulation-based search methods can be used to search for optimum wordlengths by measuring the performance at the system output.

4.1. Complete search

Complete search (CS) tests every possible combination of wordlengths between the lower bound and upper bound and measures the performance of each combination by simulation. Then optimum wordlengths can be selected from the simulations results.

For example, assuming that the number of independent variables to find optimum wordlength is two, and the lower bound and upper bound are $\{2, 2\}$ and $\{8, 7\}$, respectively, the possible wordlength combinations are shown in Figure 1. The number of trial tests or trials is 42. The optimum wordlength can be selected from the given simulation results after simulation completes.

The total number of tests in N wordlength variables is

$$E_{\text{CS}}^N = \prod_{k=1}^{N} (\overline{w}_k - \underline{w}_k + 1). \tag{9}$$
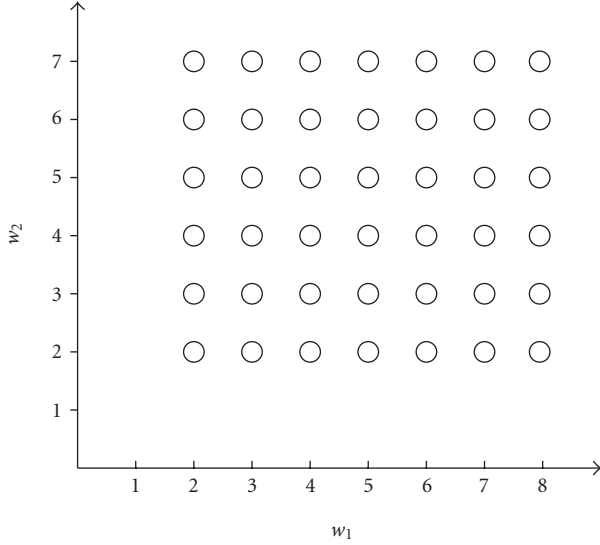
FIGURE 1: The possible wordlength combinations searching the entire space in complete search ($\underline{\mathbf{w}} = \{2, 2\}$; $\overline{\mathbf{w}} = \{8, 7\}$; trials $= 42$).
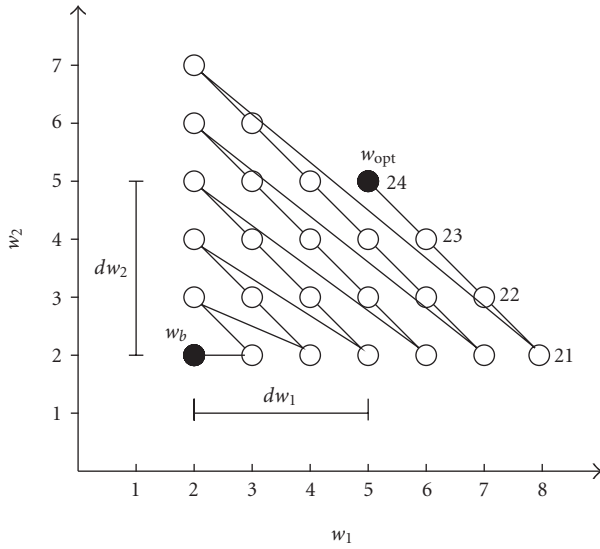

FIGURE 2: The direction of exhaustive search ($\underline{\mathbf{w}} = \{2, 2\}$; optimum point $= \{5, 5\}$; distance d in (10) is 6; trials $= 24$).

Complete search is guaranteed to find a global optimum point, but computational time and the number of tests increase exponentially as the number of wordlength variables increases.

4.2. Exhaustive search

Sung and Kum [4] search for the first feasible solution. They search for a wordlength with the minimum wordlength as the initial guess and increment the wordlength by one until the propagated error meets the minimum error. For example, assuming that we are trying to find the optimum wordlength for two variables, the minimum wordlengths are $\{2, 2\}$, and

each of wordlength cost is similar, the search path is shown in Figure 2. An optimized point $\{5, 5\}$ is given for a comparison between search methods. The minimum number of trials is 24.

We have generalized the total number of experiments of the exhaustive search in N dimensions with the sum of the distance. The sum of the distance, d, is defined as

$$d = dw_1 + dw_2 + \cdots + dw_N, \tag{10}$$

where dw_i is the distance between the minimum wordlength and the optimum wordlength in ith dimension. The expected number of experiments of the exhaustive search is calculated by using the summation of combination-with-replacement in [19] as

$$
\begin{aligned}
E_{ES}^N(d) &= \sum_{r=0}^{d-1} C^R(N, r) \\
&= C^R(N + 1, d - 1) = \binom{N + d - 1}{d - 1} \\
&= \frac{(N + d - 1)!}{\{(N + d - 1) - (d - 1)\}!(d - 1)!} \\
&= \frac{(d + N - 1) \cdots (d + 2)(d + 1)d}{N!}.
\end{aligned}
\tag{11}
$$

The trials may be bounded as

$$E_{ES}^N(d) \leq E_{ES}^{N,d} < E_{ES}^N(d + 1). \tag{12}$$

The number of experiments is always less than that of complete search if at least two feasible solutions exist. However, the exhaustive search method is not always guaranteed in finding the global optimum.

4.3. Sequential search

The basic notion of sequential search is that each trial eliminates a portion of the region being searched [5]. This procedure is also called a "Min+b search" in [16]. The sequential search method decides where the most promising areas are located, and continues in the most favorable region after each set of experiments [20]. The sequential search algorithm can be summarized by the following four steps.

(1) Select a set of values for the independent variables, which satisfy the desired system performance during the one-variable simulations.

(2) Evaluate the system performance.

(3) Choose feasible locations at which system performance is evaluated.

(4) If the system performance of one point is better than others, then move to the better point, and repeat the search, until the point has been located within the desired accuracy.

The base point is the minimum wordlength as an initial wordlength $\mathbf{w}^{(0)}$ in (8). In step (3), the direction of search, ξ as in (8) is chosen in accordance with maximum derivative
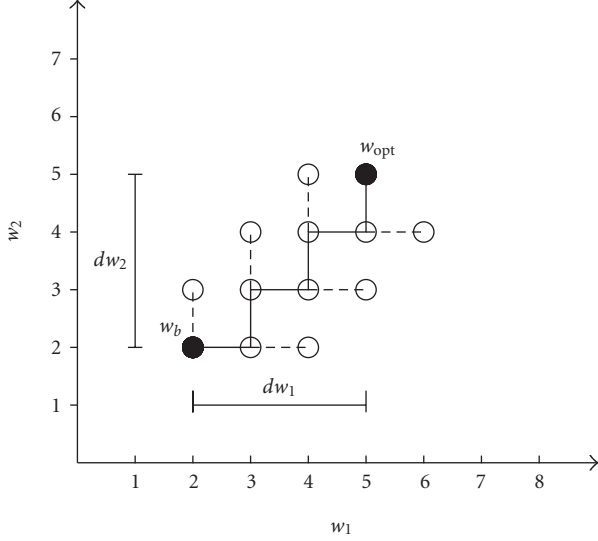
FIGURE 3: The direction of sequential search ($\underline{w}$ = {2, 2}; optimum point = {5, 5}; distance d in (10) is 6; trials = 12).
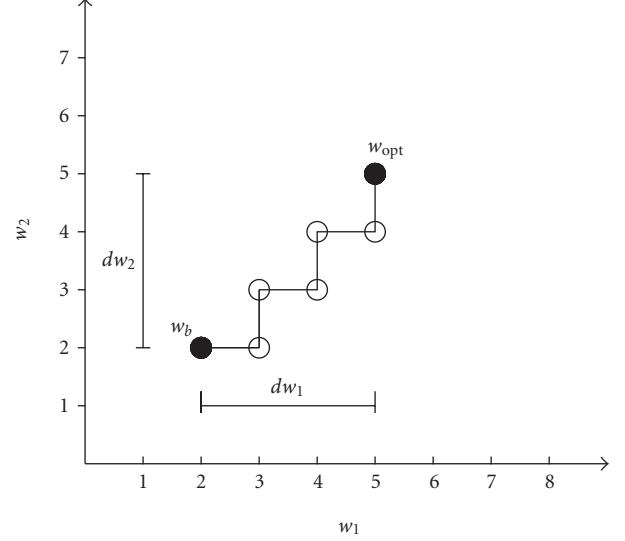

FIGURE 4: The direction of preplanned search ($\underline{w}$ = {2, 2}; optimum point = {5, 5}; distance d in (10) is 6; trials = 6).

of their performance

$$
\xi_j = \begin{cases} \{1, 0, 0, \dots, 0\}, & \text{if } m_j = \nabla \dfrac{p}{w_1}, \\ \{0, 1, 0, \dots, 0\}, & \text{if } m_j = \nabla \dfrac{p}{w_2}, \\ \dots \\ \{0, 0, 1, \dots, 0\}, & \text{if } m_j = \nabla \dfrac{p}{w_N}, \end{cases} \tag{13}
$$
$$
m_j = \max\left(\nabla \frac{p}{w_1}, \nabla \frac{p}{w_2}, \dots, \nabla \frac{p}{w_N} \right),
$$

where ∇ is the gradient operator.

In Figure 3, starting from wordlength base point {2, 2}, we measure performance of {2, 3} and {3, 2} from the direction of sequential search in step (3). If the performance of {3, 2} is better than that of {2, 3}, then a new wordlength vector moves into {3, 2}. Simulations are repeated until satisfying the desired performance.

We have generalized the trials of the sequential search in N dimensions as

$$
E_{SS}^N = N \cdot (dw_1 + dw_2 + \dots + dw_N). \tag{14}
$$

In this example, the numbers of trials are 12 from (14) and also 12 from Figure 3. The number of trials is reduced by using sensitivity information. However, an optimum wordlength can be a local optima.

Local search [3] uses sensitivity information with the above procedure, but it uses cost sensitivity instead of performance sensitivity.

4.4. Preplanned search

A preplanned search [5] is one in which all the experiments are completely scheduled in advance. The directions

are obtained from the sensitivity of performance of an independent variable. The optimum point is found by employing the steepest descent among local neighbor points.

The preplanned search algorithm in N dimensions is summarized by the following steps.

(1) Select a set of values for the independent variables, which satisfy the desired performance during the one-variable simulations.

(2) Make a performance sensitivity list from the one-variable simulations.

(3) Make a test schedule with the sensitivity list to follow the higher sensitivity points from base point.

(4) Evaluate the performance at those points.

(5) Move to the points, until the point has been located within the desired accuracy.

In step (3), the direction of preplanned search is chosen in accordance with maximum derivative of an independent performance

$$
\xi_j = \begin{cases} \{1, 0, 0, \dots, 0\}, & \text{if } m_j = \nabla \dfrac{p_1}{w_1}, \\ \{0, 1, 0, \dots, 0\}, & \text{if } m_j = \nabla \dfrac{p_2}{w_2}, \\ \dots \\ \{0, 0, 1, \dots, 0\}, & \text{if } m_j = \nabla \dfrac{p_N}{w_N}, \end{cases} \tag{15}
$$

where

$$
m_j = \max\left(\nabla \frac{p_1}{w_1}, \nabla \frac{p_2}{w_2}, \dots, \nabla \frac{p_N}{w_N} \right). \tag{16}
$$

In Figure 4, starting from the base point {2, 2}, the preplanned search makes a list of the directions of the steepest ascent by comparing the gradients of the independent performances in one dimension from the one-variable simulations. If the gradient, which is calculated from the one-variable

(a) Analog demodulator.
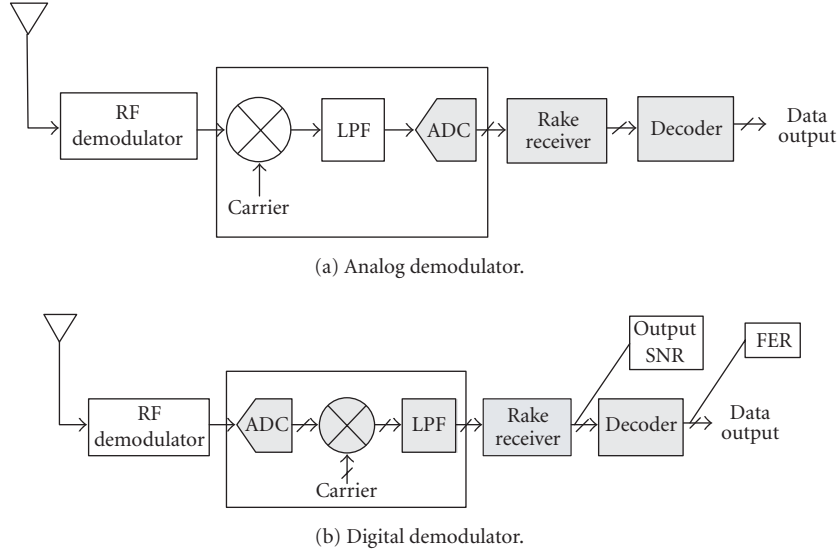

(b) Digital demodulator.

FIGURE 5: Analog and digital demodulators in CDMA receiver and performance measurement position.
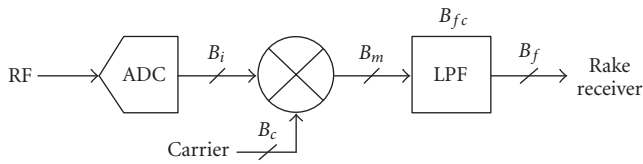

FIGURE 6: A digital demodulator block.

simulations at w_1 of 2 bits, is larger than that at w_2 for 2 bits, then next feasible location is $\{3, 2\}$. Then, if the gradient at w_1 of 3 is smaller than that at w_2 of 2, the next feasible location is $\{3, 3\}$. The simulation path would be $\{2, 2\}$, $\{3, 2\}$, $\{3, 3\}$, and so forth. After scheduling the feasible points, the performance of these points is evaluated until the value of the performance meets the desired accuracy.

We generalized the trials of the preplanned search in N dimensions as

$$E_{PS}^N = dw_1 + dw_2 + \cdots + dw_N. \qquad (17)$$

In this example, the trials are 6 from (17) and from Figure 4. The number of trials is the least among the search methods reported so far. However, finding the global optimum wordlength is not guaranteed.

4.5. Search example in CDMA demodulator wordlength design

Typical demodulators are implemented with an analog block in front of an analog-to-digital converter (ADC) block as shown in Figure 5(a). As the speed of the ADC increases, analog parts are replaced with digital parts in communication systems [21]. We replaced the analog demodulator with the digital demodulator as shown in Figure 5(b).

The demodulator converts modulated signals into baseband signals. In the digital demodulator block of Figure 6, the sampled data values output by the ADC are multiplied by a carrier signal to shift the spectrum down to the baseband. The out-of-band signal is removed by the lowpass filter (LPF). The variables in the digital demodulator are given below [22, 23]:

  (i) (B_i): input wordlength;
 (ii) (B_c): carrier wordlength;
(iii) (B_m): multiplier output wordlength;
 (iv) (B_f): filter output wordlength;
  (v) (B_{fc}): filter coefficient wordlength.

The output SNR is used for performance measurement instead of frame error rate (FER), which is a general measurement to evaluate CDMA systems because direct measurement of FER requires at least 10^5 frames during the simulation [24]. The required output SNR in this system is over 0.8 dB, or FER is under 0.03 [23].

For the initial point, minimum wordlength is selected by the independent one-variable simulations in which one variable changes while other variables keep high precision. Satisfying the output SNR of 0.8 dB, the minimum wordlength of $\{B_i, B_c, B_m, B_f, B_{fc}\}$ is $\{4, 3, 4, 5, 7\}$, which is acquired from the one-variable simulations shown as Figure 7. For a simplified example, we assume that the cost-per-bit is one. In the exhaustive search, the next points are searched: $\{5, 3, 4, 5, 7\}$, $\{4, 4, 4, 5, 7\}$, $\{4, 3, 5, 5, 7\}$, $\{4, 3, 4, 6, 7\}$, $\{4, 3, 4, 5, 8\}$, $\{5, 4, 4, 5, 7\}$, and so forth. The search is continued until the communications performance meets the specific desired requirement. In the sequential search, the next point is one of the following: $\{5, 3, 4, 5, 7\}$, $\{4, 4, 4, 5, 7\}$, $\{4, 3, 5, 5, 7\}$, $\{4, 3, 4, 6, 7\}$, and $\{4, 3, 4, 5, 8\}$. The next point would have the largest communication performance among them. From Table 3, $\{4, 3, 4, 6, 7\}$ is the next

TABLE 3: Sequence of the sequential search for CDMA demodulator (traffic channel rate set 1 in additive white Gaussian noise, input SNR = −17.3 dB, Eb/Nt = 3.8, rate = 9600 bps, and desired performance: output SNR > 0.8 dB, FER < 0.03).

Step	$\{B_i, B_c, B_m, B_f, B_{fc}\}$	Output SNR	FER	Result
1, 2	$\{4, 3, 4, 5, 7\}$	0.711	0.038	Fail
3	$\{5, 3, 4, 5, 7\}$	0.735	—	—
3	$\{4, 4, 4, 5, 7\}$	0.694	—	—
3	$\{4, 3, 5, 5, 7\}$	0.712	—	—
3	$\{4, 3, 4, 6, 7\}$	0.759	—	Max
3	$\{4, 3, 4, 5, 8\}$	0.704	—	—
4	$\{4, 3, 4, 6, 7\}$	0.759	0.035	Fail
3	$\{5, 3, 4, 6, 7\}$	0.763	—	—
3	$\{4, 4, 4, 6, 7\}$	0.722	—	—
3	$\{4, 3, 5, 6, 7\}$	0.773	—	Max
3	$\{4, 3, 4, 7, 7\}$	0.751	—	—
3	$\{4, 3, 4, 6, 8\}$	0.749	—	—
4	$\{4, 3, 5, 6, 7\}$	0.773	0.034	Fail
⋮	⋮	⋮	⋮	⋮
3	$\{6, 3, 5, 6, 7\}$	0.798	—	—
3	$\{5, 4, 5, 6, 7\}$	0.802	—	—
3	$\{5, 3, 6, 6, 7\}$	0.805	—	Max
3	$\{5, 3, 5, 7, 7\}$	0.803	—	—
3	$\{5, 3, 5, 6, 8\}$	0.798	—	—
4	$\{5, 3, 6, 6, 7\}$	0.805	0.029	Pass
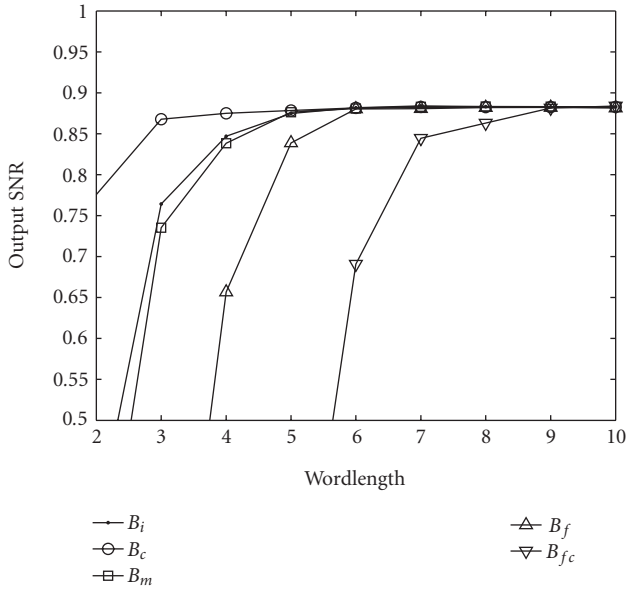
FIGURE 7: Result of the independent one-variable simulations on a CDMA demodulator.

location because it has the largest communication performance. The simulation moves the current point to the new point and continues to search until the performance exceeds the specific desired requirement, which is an output SNR of 0.8 dB in this case. The final point is $\{5, 3, 6, 6, 7\}$, as shown in Table 3. The distance between the base and the optimum

point is 4 by using (10). The number of trials for the sequential search to find an optimum wordlength is 20 by using (14).

In the preplanned search, the search path is estimated from the sensitivity of each one-variable simulation shown in Figure 7. Starting from the minimum wordlength or base point, $\{4, 3, 4, 5, 7\}$, the first expected point is $\{4, 3, 4, 6, 7\}$ because B_f has the greatest derivative among each wordlength at the base point from Figure 7. The sequence of the preplanned search points is $\{4, 3, 4, 5, 7\}$, $\{4, 3, 4, 6, 7\}$, $\{4, 3, 4, 6, 8\}$, $\{4, 3, 5, 6, 8\}$, $\{4, 4, 5, 6, 8\}$, and so forth. Simulations move the current point to the next point until the performance exceeds the specific desired requirement. The optimum point is $\{5, 4, 5, 6, 8\}$ and distance is 5 by using (10). The number of trials of the preplanned search to find an optimum wordlength is 5 by using (17).

4.6. Comparison

The four search methods are compared with the trials from (9), (11), (14), and (17), as shown in Table 4. The numbers of trials are calculated besides the one-variable simulations which all of the search methods use. The complete search needs 283920 trials to find optimum wordlength from (9) with $\overline{w}_k = \{16, 16, 16, 16, 16\}$ and $\underline{w}_k = \{4, 3, 4, 5, 7\}$ assuming that the maximum wordlength is 16 bits. If the computer simulation to calculate frame error rate per trial in CDMA system takes about 10 minutes, the complete search to find an optimum wordlength would require 5 years, which is unrealistic design time.

TABLE 4: Comparison of complete, exhaustive, sequential, and preplanned search ($N = 5$, $\overline{w}_k = \{16, 16, 16, 16, 16\}$, $\underline{w}_k = \{4, 3, 4, 5, 7\}$, and the term d is defined in (10)).

Search	Distance (d)	Equation for number of experiments from (9), (11), (14), (17)	Trials
Complete	—	$\prod_{k=1}^{N}(\overline{w}_k - \underline{w}_k + 1)$	283920
Exhaustive	4	$(d+4)(d+3)\cdots(d)/5!$	56
Sequential	4	$5 \cdot d$	20
Preplanned	5	d	5

The exhaustive search needs 56 trials by using (11), which is less than the complete search. The exhaustive search is, however, inefficient to find the optimum wordlength when the wordlength variables for optimization are numerous and the distance between base and optimum point is longer.

The sequential search and preplanned search requires 20 and 5 trials, respectively, which are less than the other search methods. The preplanned search has the lowest number of experiments among search methods, but its distance using (10) is larger than that for sequential search. It implies that the wordlength of the sequential search method is closer to a global optimum with respect to hardware cost.

The sequential search and preplanned search have a loss of direction problem encountered by techniques based on the gradient projection method. This problem can be solved by adapting the step size.

The sequential search and the preplanned search reduce the trials by rates of 64% and 91%, respectively, when compared to the exhaustive search for wordlength optimization in the CDMA demodulator design. However, preplanned search seldom converges to the same optimum point, and the distance is longer than that of the other search methods.

5. SENSITIVITY MEASUREMENTS

The sensitivity information used for update directions in (8) can help reduce the search space dramatically. The sensitivity information can be obtained by measuring hardware complexity and distortion or propagated quantized precision loss. Complexity measure is used for hardware cost function in [3]. Distortion measure in [5] utilizes the sensitivity information of a propagated quantization error. Complexity-and-distortion measure in [7] combines two measures to update the search direction.

5.1. Complexity measure

The complexity measure method considers hardware complexity function as the cost function in (4) and uses the sensitivity information of the complexity as the direction to search for the optimum wordlengths. The local search in [3] uses complexity measure. The sensitivity information is calculated by gradient of the complexity function. For steepest descent direction, the update direction is

$$\xi_{CM} = -\nabla f_c(\mathbf{w}),   \tag{18}$$

where ∇ is gradient of function.

Complexity measure method updates wordlengths from the direction of the lowest sensitive complexity until a system meets a required performance such as P_{req} in (5). The complexity measure method searches the wordlengths that minimize hardware complexity. However, it demands a large number of iterations since it does not use any distortion sensitivity information that can speed up to find the optimum wordlengths. For example, in a system composed of adders and multipliers, the complexity sensitivity of a multiplier is larger than that of an adder. The complexity measure method increases the wordlength in the adder with the right of priority during an increase procedure even if the wordlength in the multiplier affects the propagated quantized performance more. It would waste computer simulation time if the complexity sensitivity of an adder is much smaller than that of a multiplier.

5.2. Distortion measure

The distortion measure method considers distortion function as the objective function in (4) and uses the sensitivity information of the distortion for the direction to search for the optimum wordlengths. Sequential search uses distortion measure. This method assumes that every cost or complexity function would be the same or equal to 1, and selects wordlengths with the update direction according to the distortion sensitivity information.

The complexity objective function is replaced with the distortion objective function $d(\mathbf{w})$ as

$$f_d(\mathbf{w}) = d(\mathbf{w}),   \tag{19}$$

and the complexity minimization problem is changed into a distortion minimizing problem by

$$\min_{\mathbf{w} \in \mathbf{I}^n} f_d(\mathbf{w}), \quad \text{subject to } d(\mathbf{w}) \leq D_{req},$$
$$c(\mathbf{w}) \leq C_{req}, \quad \underline{w} \leq \mathbf{w} \leq \overline{w},   \tag{20}$$

where D_{req} is required distortion, and C_{req} is a complexity constant.

The sensitivity information is also calculated by gradient of the distortion function. For the steepest descent direction, the update direction is

$$\xi_{DM} = -\nabla f_d(\mathbf{w}).   \tag{21}$$

For the distortion, Fiore and Lee [25] computed an error variance, and Han et al. [5] measured output SNR.

The distortion measure method reduces the number of iterations for searching the optimum wordlengths, since

FIGURE 8: Wordlength model for a fixed broadband wireless access demodulator.

the search direction depends on the distortion by changing the wordlengths. This method rapidly finds the optimum wordlength satisfying the required performance by a fewer number of iterations compared to complexity measure method. However, the wordlengths do not guarantee the optimum wordlengths in terms of the complexity.

5.3. Complexity-and-distortion measure

The complexity-and-distortion measure combines the complexity measure with the distortion measure by a weighting factor. In the objective function, both complexity and distortion are simultaneously considered. We normalize the complexity and the distortion function and multiply them with complexity and distortion weighting factors, α_c and α_d, respectively. The new objective function is

$$f_{cd}(\mathbf{w}) = \alpha_c \cdot c_n(\mathbf{w}) + \alpha_d \cdot d_n(\mathbf{w}), \qquad (22)$$

where $c_n(\mathbf{w})$ and $d_n(\mathbf{w})$ are normalized complexity function and distortion function, respectively. The relation between the weighting factors is

$$\alpha_c + \alpha_d = 1, \qquad (23)$$

where

$$0 \le \alpha_c \le 1, \qquad 0 \le \alpha_d \le 1. \qquad (24)$$

Using (22), the objective function gives a new optimization problem

$$\min_{\mathbf{w} \in \mathbf{I}^n} f_{cd}(\mathbf{w}), \quad \text{subject to } d(\mathbf{w}) \le D_{\text{req}},$$
$$c(\mathbf{w}) \le C_{\text{req}}, \quad \underline{w} \le \mathbf{w} \le \overline{w}, \qquad (25)$$

where D_{req} and C_{req} are the required distortion and a complexity constant, respectively. This optimization problem is to find wordlengths that minimize complexity and distortion simultaneously according to the weighting factors.

The update direction for the steepest decent direction to find the optimum wordlength $\mathbf{w}$ is

$$\xi_{\mathbf{CDM}} = -\nabla f_{cd}(\mathbf{w}). \qquad (26)$$

From (22) and (26), the update direction is

$$\xi_{\mathbf{CDM}} = -(\alpha_c \cdot \nabla c_n(\mathbf{w}) + \alpha_d \cdot \nabla d_n(\mathbf{w})). \qquad (27)$$

Setting the complexity and distortion weighting factor, α_c or α_d from 0 to 1, the complexity-and-distortion method searches for an optimum wordlength with tradeoffs between complexity measure method and distortion measure method. The complexity-and-distortion measure becomes the complexity measure or the distortion measure when $\alpha_d = 0$ or $\alpha_c = 0$, respectively.

The complexity-and-distortion measure method can reduce the number of iterations for searching the optimum wordlengths, because the distortion sensitivity information is utilized. This method can more rapidly find the optimum wordlength that satisfies the required performance by using less iteration compared to the complexity measure method. However, the wordlengths are not guaranteed to be optimal in terms of the complexity.

6. CASE STUDY

6.1. OFDM demodulator design

Digital communication systems have digital blocks such as a demodulator that needs wordlength optimizations. Searching algorithms in Section 4 were applied to the wordlength optimization of CDMA demodulator design in Section 4.5. From the CDMA case study, the sequential search is one of the promising methods to find an optimum wordlength. In this section, complexity measure, distortion measure, and complexity-and-distortion measure in Section 5 are applied in the sequential search framework to determine wordlengths for a fixed broadband wireless demodulator.

Fixed broadband wireless access technology is intended for high-speed voice, video, and data services, which is presently dominated by cable and digital subscriber line technologies [26]. One of the designs for orthogonal frequency division multiplexing (OFDM) demodulators for fixed broadband wireless access is shown in Figure 8. For the wireless channel, we used Stanford University Interim models [27, 28].

The main blocks in the demodulator for finite wordlength determination are the fast fourier transform (FFT), equalizer, and estimator. For wordlength variables, we choose the wordlengths that have the most significant effect on complexity and distortion in the system. For the OFDM demodulator, we select wordlength variables w_0, w_1, w_2, and w_3 for the FFT input, equalizer right input, channel estimator input, and equalizer upper input, respectively, as shown in Figure 8.
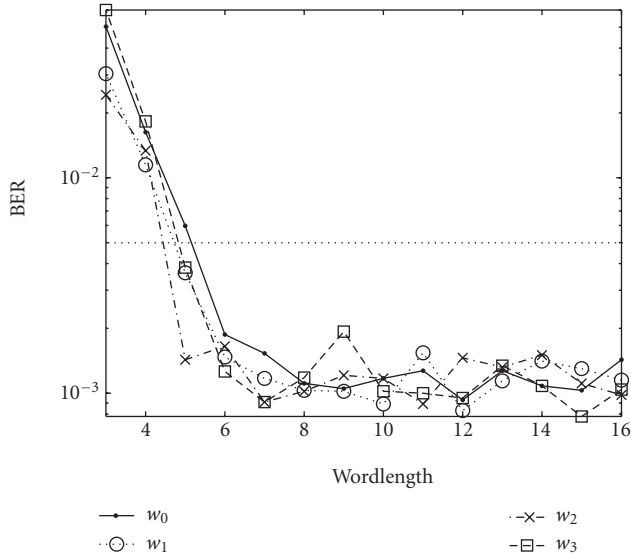
FIGURE 9: Wordlength effect for the demodulator in Figure 8, with Stanford University Interim wireless channel model number 3, SNR of 20 dB, FFT length of 256, and least-squares comb-type channel estimator without error-control coding.

We assume that the internal wordlengths of the given blocks have already been decided. In simulation, only the inputs to each block are constrained to be in fixed-point type, whereas the blocks themselves are simulated in floating-point type.

For the hardware complexity, the number of multiplications is measured assuming that processing units are not reused. The number of multiplications in a K-point FFT block is

$$\text{Cost}_{\text{FFT}} = \frac{K}{2} \log_2 K, \tag{28}$$

where K is the number of taps. The cost of the 256-point FFT in the fixed broadband wireless access is estimated to be 1024. Approximately, the simplified complexity vector $\mathbf{c}$ of the wordlength per bit is assumed to be $\{1024, 1, 128, 2\}$ from [4, 29].

We also assume the complexity increases linearly as wordlength increases to simplify demonstration. For the distortion measurement, bit error rate (BER) is measured. The minimum wordlength searched by changing one wordlength variable, while other variables have high precision (i.e., 16 bits), is used for the initial wordlength [4, 5]. The simulation for the minimum wordlength is shown in Figure 9.

Assuming the minimum performance of BER is 5×10^{-3}, the minimum wordlength is $\{5, 4, 4, 4\}$ from Figure 9. Starting from the minimum wordlength, wordlengths are increased according to the sensitivity information of different measures in Section 5. We measure the number of iterations until they find their own optimum wordlength satisfying the required performance such as BER $\leq 2 \times 10^{-3}$

without channel decoder. For the optimum wordlength, we follow the hybrid procedure [16] that combines a wordlength increase followed by a wordlength decrease. Simulation results are presented in Section 7.

6.2. IIR filter case

OFDM demodulation case requires a large number of long simulations. This becomes especially time-consuming when each simulation takes hours in ensemble average of BER estimation. For more general case, infinite impulse response (IIR) filter that has 7 wordlengths is simulated. There are various methods for getting error function and cost function as described in related work section. For simplifying the simulation, mean square error (MSE) is measured for the error function, and a linear cost function of wordlength is assumed. Required performance of the IIR filter is assumed MSE of 0.1. In the IIR filter case study, the wordlength vector has 7 elements, and the hardware complexity of the arithmetic block has less difference when compared to the OFDM case study. Results are presented in Section 7.

7. RESULTS

The wordlength optimization problem is a discrete optimization problem with a nonconvex constraint space [30]. This nonconvexity makes it harder to search for a global optimum solution [31]. Tables 5 and 6 show that there are several local optimum wordlengths that satisfy error specification and minimize hardware complexity in case studies. In this section, wordlength optimization methods used in case studies are compared in terms of number of iteration and hardware complexity, and future work is discussed.

7.1. Number of iterations

The number of iteration to search an optimum wordlength in OFDM demodulator design is shown in Figure 10. The initial wordlength does not satisfy the desired performance. After a number of trials by updating wordlength as in (8), the error at the system output decreases. The sequential search and the CDM search reach the feasible area after 15 trials. However, the local search takes 38 trials. After arriving at the feasible area, an optimum wordlength is searched again. In this case, the wordlengths, which are searched by the sequential search or the CDM search, already arrive at an optimum wordlength. However, the local search needs more iterations to find an optimum wordlength. The total number of trials to find an optimum wordlength in each method for OFDM case is shown in Table 5. The sequential search and the CDM method can find an optimum solution in one-fourth of the time that the local search method takes.

In IIR filter design, the number of iterations to search an optimum wordlength is shown in Figure 11. This figure demonstrates the number of trials in an infeasible area and a feasible area. After the search methods reach a feasible region, where MSE of IIR filter is under 0.1, the search methods continue searching an optimum wordlength. The sequential

TABLE 5: Simulation results of several search methods starting from the minimum wordlength for the demodulator arcs in Figure 8. $N = 4$, $\underline{w}_k = \{5, 4, 4, 4\}$, and $\overline{w}_k = \{16, 16, 16, 16\}$. CDM is the complexity-and-distortion measure. α_c is a weighting factor.

Search method	α_c	Number of trials	Wordlengths for variables	Complexity estimate
Sequential search [5]	0	16	$\{10, 9, 4, 10\}$	10781
CDM	0.5	15	$\{7, 10, 4, 6\}$	7702
Local search [3]	1	69	$\{7, 7, 4, 6\}$	7699

TABLE 6: Simulation results in IIR filter of several search methods. $N = 7$, $\underline{w}_k = \{1, 1, 1, 1, 1, 1, 1\}$, and $\overline{w}_k = \{16, 16, 16, 16, 16, 16, 16\}$. CDM is the complexity-and-distortion measure. α_c is a weighting factor. (Max-1 search starts from $\overline{w}_k$. Sequential search starts from $\underline{w}_k$).

Search method	α_c	Number of trials	Wordlengths for variables	Complexity estimate
Max-1 search [16]	0	94	$\{4, 5, 4, 5, 2, 2, 4\}$	378
Sequential search [5]	0	56	$\{4, 5, 4, 5, 2, 2, 4\}$	378
CDM	0.25	44	$\{4, 5, 4, 4, 2, 2, 5\}$	366
CDM	0.5	33	$\{6, 5, 5, 4, 1, 2, 4\}$	363
CDM	0.75	71	$\{6, 4, 4, 4, 2, 16, 13\}$	561
Local search [3]	1	126	$\{9, 5, 16, 4, 1, 16, 16\}$	723

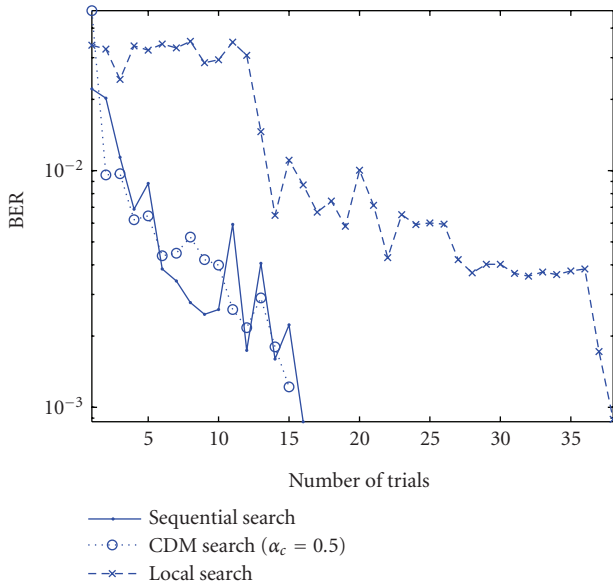FIGURE 10: Number of iterations for optimum wordlength with various search algorithms in OFDM demodulator wordlength design.

FIGURE 11: Number of iterations for optimum wordlength in IIR filter with various search algorithms.

search and the local search need a total of 56 and 126 iterations, respectively, including iterations in feasible and infeasible area as shown in Table 6. The "Max-1" search starting from the feasible area needs 96 iterations. The CDM methods with weighting factor of 0.25, 0.5, and 0.75 are used for comparison. When α_c is less than 0.5, the CDM methods have the property of the sequential search. When α_c is greater than 0.5, the CDM methods search as the local search does. In Figure 11, the CDM methods with weighting factor of 0.25 and 0.75 show similar shape as the sequential search and the local search, respectively. In the IIR filter case, the CDM method with α_c of 0.5 can find an optimum solution in one-fourth of the time that the local search method takes.
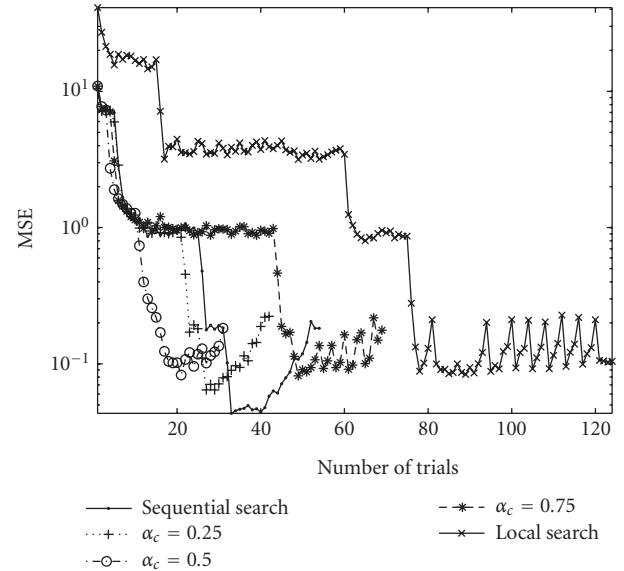
In general, if error sensitivity information for searching an optimum wordlength is used, the number of iterations can be reduced. The sequential search and the CDM method with less than α_c of 1 use the error sensitivity information. Thus, they converge quickly into an optimum wordlength that satisfies the required error performance.

7.2. Hardware complexity

Tables 5 and 6 show the hardware complexity according to the searched optimum wordlengths in various methods. The results show that the sequential search method, which only uses error sensitivity information for the update direction,

TABLE 7: Simulation results in noise cancellation with Wiener filter [32] of several search methods. $N = 5$, $\underline{w}_k = \{1, 1, 1, 1, 1\}$, and $\overline{w}_k = \{16, 16, 16, 16, 16\}$. CDM is the complexity-and-distortion measure. α_c is a weighting factor.

Search method	α_c	Number of trials	Wordlengths for variables	Complexity estimate
Sequential search [5]	0	21	$\{4, 5, 5, 3, 2\}$	1331
CDM	0.25	23	$\{4, 4, 5, 4, 2\}$	1200
CDM	0.5	24	$\{5, 4, 4, 5, 4\}$	1074
CDM	0.75	167	$\{4, 4, 4, 5, 4\}$	1073
Local search [3]	1	170	$\{5, 4, 4, 15, 3\}$	1082

finds an optimum wordlength that has higher complexity than the CDM method and the local search in the OFDM demodulation case study. However, an optimum wordlength searched by the local search method, which uses hardware complexity information, has higher complexity in the IIR filter case study. If the design space is convex and has only one optimum solution, then various search methods find the optimum solution. However, wordlength optimization problem has many local optimum solutions because of nonconvex space. As the number of wordlength variables increases and as the system becomes more complicated, the probability in being stuck in a local optimum solution increases. In the IIR filter case with 7 elements in wordlength vector, the wordlengths searched by the local search method are far from globally optimal.

The CDM search with the weighting factor of 0.5 finds an optimum wordlength that has the lowest hardware complexity in this IIR case study. The CDM search with the weighting factor of 0.75 tends to be the local search. The hardware complexity from the CDM method of 0.75 is between the CDM of 0.5 and the local search. Similarly, the complexity from CDM method of 0.25 is between the sequential search and the CDM of 0.5.

For more examples, additional optimum wordlength search results in a noise cancellation with Wiener filter [32] are shown in Table 7.

7.3. Discussion

The CDM method, which uses error and complexity sensitivity for optimum wordlength search, takes advantages from the sequential search and the local search. This method reduces the number of iterations because of the error sensitivity that helps to fast reach feasible boundary. At the same time, this method finds a near-optimum wordlength that has lower hardware complexity because of the sensitivity of hardware complexity. The proposed method is robust for search optimum wordlength in a nonconvex space because this method is not easily captured by local optimum solutions.

The complexity-and-distortion measure method has flexibility to search for an optimum wordlength by setting weighting factor. Designer can select the weighting factor, α_c, as in (23). The α_c of 0.5 means that the CDM method equally uses the sensitivity information of the error and the complexity. The α_c of 0.5 in CDM is reasonable for optimum wordlength search algorithms.

7.4. Future work

For an extension of work, various methods can be combined for wordlength optimization. Wordlength grouping [4] can be used to reduce a wordlength vector. Error model or error monitoring instead of error measuring can be used to reduce the simulation time. Actual cost model [12] can be used to get accurate result. For the searching method, different search methods such as binary search can be combined. Preplanned search, which is the fastest error sensitivity search method as compared in [16], can employ CDM methods to reach more quickly a near-optimum wordlength.

8. CONCLUSION

This paper generalized wordlength optimization methods that use sensitivity measures. The proposed complexity-and-distortion measure equation can express the local search or sequential search by changing the weighting factor. The weighting factor can reduce the number of iterations and the hardware complexity compared to the local search and the sequential search, respectively. In our case studies, the complexity-and-distortion method is simulated and compared. The proposed method can find the optimum solution in one-fourth of the time that the local search takes. In addition, the optimum wordlength searched by the proposed method has 30% lower hardware implementation costs than sequential search in wireless demodulators. Case studies demonstrate that the proposed method is robust for searching optimum wordlength in a nonconvex space. Future extensions of this work include combination with analytic wordlength optimization and preplanned search.

REFERENCES

[1] H. Keding, M. Willems, M. Coors, and H. Meyr, "FRIDGE: A fixed-point design and simulation environment," in *Proceedings of IEEE Design, Automation and Test in Europe (DATE '98)*, pp. 429–435, Paris, France, February 1998.

[2] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 1998.

[3] H. Choi and W. P. Burleson, "Search-based wordlength optimization for VLSI/DSP synthesis," in *Proceedings of IEEE Workshop on VLSI Signal Processing, VII*, pp. 198–207, La Jolla, Calif, USA, October 1994.

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

[5] K. Han, I. Eo, K. Kim, and H. Cho, "Numerical word-length optimization for CDMA demodulator," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '01)*, vol. 4, pp. 290–293, Sydney, NSW, Australia, May 2001.

[6] M.-A. Cantin, Y. Savaria, D. Prodanos, and P. Lavoie, "An automatic word length determination method," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '01)*, vol. 5, pp. 53–56, Sydney, NSW, Australia, May 2001.

[7] K. Han and B. L. Evans, "Wordlength optimization with complexity-and-distortion measure and its application to broadband wireless demodulator design," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 5, pp. 37–40, Montreal, Quebec, Canada, May 2004.

[8] S. A. Wadekar and A. C. Parker, "Accuracy sensitive word-length selection for algorithm optimization," in *Proceedings of International Conference on Computer Design: VLSI in Computers and Processors (ICCD '98)*, pp. 54–61, Austin, Tex, USA, October 1998.

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

[10] R. Cmar, L. Rijnders, P. Schaumont, S. Vernalde, and I. Bolsens, "A methodology and design environment for DSP ASIC fixed point refinement," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 271–276, Munich, Germany, March 1999.

[11] M. Stephenson, J. Babb, and S. Amarasinghe, "Bidwidth analysis with application to silicon compilation," in *Proceedings of ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 108–120, Vancouver, BC, Canada, June 2000.

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

[13] 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 SystemsPart II: Analog and Digital Signal Processing*, vol. 45, no. 11, pp. 1455–1464, 1998.

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

[15] A. Nayak, M. Haldar, A. Choudhary, and P. Banerjee, "Precision and error analysis of MATLAB applications during automated hardware synthesis for FPGAs," in *Proceedings of Design, Automation and Test in Europe (DATE '01)*, pp. 722–728, Munich, Germany, March 2001.

[16] M.-A. Cantin, Y. Savaria, and P. Lavoie, "A comparison of automatic word length optimization procedures," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 2, pp. 612–615, Phoenix-Scottsdale, Ariz, USA, May 2002.

[17] *SystemC 2.0 User's Guide*, 2002, [online], available: www.systemc.org.

[18] S. Kim and W. Sung, "A floating-point to fixed-point assembly program translator for the TMS 320C25," *IEEE Transactions on Circuits and SystemsPart II: Analog and Digital Signal Processing*, vol. 41, no. 11, pp. 730–739, 1994.

[19] K. H. Rosen, *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, Boca Raton, Fla, USA, 2000.

[20] G. S. G. Beveridge and R. S. Schechter, *Optimization: Theory and Practice*, McGraw-Hill, New York, NY, USA, 1970.

[21] J. A. Wepman, "Analog-to-digital converters and their applications in radio receivers," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 39–45, 1995.

[22] S. Nahm, K. Han, and W. Sung, "A CORDIC-based digital quadrature mixer: comparison with a ROM-based architecture," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '98)*, vol. 4, pp. 385–388, Monterey, Calif, USA, May–June 1998.

[23] K. Han, I. Eo, K. Kim, and H. Cho, "Bit constraint parameter decision method for CDMA digital demodulator," in *Proceedings of 5th CDMA International Conference and Exhibition*, vol. 2, pp. 583–586, Seoul, Korea, November 2000.

[24] J.-S. Wu, M.-L. Liou, H.-P. Ma, and T.-D. Chiueh, "A 2.6-V, 44-MHz all-digital QPSK direct-sequence spread-spectrum transceiver IC [wireless LANs]," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 10, pp. 1499–1510, 1997.

[25] P. D. Fiore and L. Lee, "Closed-form and real-time wordlength adaptation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, vol. 4, pp. 1897–1900, Phoenix, Ariz, USA, March 1999.

[26] H. Bolcskei, A. J. Paulraj, K. V. S. Hari, R. U. Nabar, and W. W. Lu, "Fixed broadband wireless access: state of the art, challenges, and future directions," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 100–108, 2001.

[27] V. Erceg, K. V. S. Hari, M. S. Smith, et al., "Channel models for fixed wireless applications," in *IEEE 802.16. proposal 802.16.3c-01/29*, 2001.

[28] D. S. Baum, "Simulating the SUI channel models," Tech. Rep., Information Systems Laboratory, Stanford University, Stanford, Calif, USA, 2001.

[29] B. Shim and N. Shanbhag, "Complexity analysis of multicarrier and single-carrier systems for very high-speed digital subscriber line," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 282–292, 2003.

[30] G. A. Constantinides, "High level synthesis and word length optimization of digital signal processing systems," Ph.D. dissertation, Department of Electronic & Electrical Engineering, University College London, London, UK, 2001.

[31] R. Fletcher, *Practical Methods of Optimization, Vol. 2: Constrained Optimization*, John Wiley & Sons, New York, NY, USA, 1981.

[32] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, New York, NY, USA, 1996.

Kyungtae Han received the B.S. degree in information engineering from Korea University in 1996, and the M.S. degree in electrical engineering from Seoul National University in 1998. Since August 2002, he has been pursuing his Ph.D. degree in electrical engineering in The University of Texas at Austin. In 2000, he joined Electronics and Telecommunications Research Institute, Korea, as a Research Engineer. His research experiences include modem designs for CDMA and OFDM systems. His current research activities are on tradeoffs in signal quality versus implementation complexity to develop near-optimal, low-complexity, and low-power algorithms.

Brian L. Evans is Professor of electrical and computer engineering at The University of Texas at Austin, Austin, Tex, USA. His B.S. E.E. C.S. (1987) degree is from the Rose-Hulman Institute of Technology, and his M.S. E.E. (1988) and Ph.D. E.E. (1993) degrees are from the Georgia Institute of Technology. From 1993 to 1996, he was a Postdoctoral Researcher at UC Berkeley. His research efforts are in embedded real-time signal and image processing systems. In signal processing, his research group focuses on the design and real-time software implementation of high-speed ADSL transceivers and multiuser OFDM systems. In image processing, his group focuses on the design and real-time software implementation of high-quality halftoning for desktop printers and smart image acquisition for digital-still cameras. Dr. Evans has published over 140 refereed conference and journal papers. Dr. Evans is the Primary Architect of the Signals and Systems Pack for Mathematica. He was a key contributor to UC Berkeley's Ptolemy classic electronic design automation environment for embedded systems, which has been successfully commercialized by Agilent and Cadence. He is an Associate Editor for the IEEE Transactions on Signal Processing. He is the recipient of a 1997 US National Science Foundation Career Award.