Interleaver Optimization of Convolutional Turbo Code for 802.16 Systems

Sung-Joon Park, *Member, IEEE,* and Jun-Ho Jeon

***Abstract*—Convolutional turbo code (CTC) has been adopted as an optional channel coding scheme for wireless metropolitan area networks (WMAN) and digital video broadcasting (DVB) owing to its powerful error correction capability and flexibility. In spite of the fact that the internal interleaver of CTC mainly affects the characteristics of the code, interleaving parameters, especially for small and moderate block sizes, have not been fully optimized in terms of performance. In this paper, we investigate structural behavior of interleaving parameters and suggest several optimization methodologies for the CTC interleaver of 802.16 systems. Simulation shows that power gains up to 0.7 dB can be acquired at a target frame error rate of 10^{-5} by only substituting the newly optimized parameters for the current ones.**

***Index Terms*—Convolutional turbo code, interleaver, 802.16, wireless metropolitan area network.**

I. Introduction

Wireless metropolitan area networks (WMAN) whose IEEE standard is called 802.16 are communication systems designed to support fixed and mobile broadband wireless access in licensed bands. Two main subjects of the standard currently spotlighted are WiMAX (worldwide interoperability for microwave access) and mobile WiMAX based on OFDM and OFDMA, respectively. In both systems, convolutional turbo code (CTC) has been commonly adopted as an optional channel coding to mitigate data corruption that occurs during transmission through a wireless channel [1], [2]. The CTC in 802.16 is a double-binary turbo code which was originally devised by Douillard and Berrou [3]. It is different from classical turbo code [4] in that it allows two information bits to be encoded and decoded at the same time, promising higher throughput and reduced latency.

One of the major research topics influencing the performance of a turbo code is efficient design of internal interleaver. Among numerous permutation models, a dithered relatively prime (DRP) interleaver was proposed in [5] and as succeeding work an almost regular permutation (ARP) model was suggested in [6]. The basic framework of ARP has been adopted in 802.16 and DVB-RCS/RCT (digital video broadcasting standards for return channel via satellite and terrestrial distribution system [7], [8]). In the interleaver case of 802.16, interleaving addresses are generated by regular increment based on an parameter P_0 which is relatively prime with an interleaver size and by irregular increment coming from offset values due to parameters P_1, P_2, P_3. Therein, many kinds of block sizes are defined to support various data rates and each block size has four interleaving parameters P_0, P_1, P_2 and P_3 involving the generation of interleaving addresses. For large block sizes, the parameters were optimized in [9] and embodied in the specification [2], however, the parameters for small and moderate block sizes such as $N \leq 240$ have not been optimized in the sense of decoding performance. From these observations, in this paper we try to find the optimum interleaving parameters of CTC for 802.16 systems. We divide the optimization problem into two parts incurring regular and irregular increments. That is, we first set up some of offset vectors as candidates and then optimize P_0 in terms of spread and minimum distance. We present simulation results in an additive white Gaussian noise (AWGN) channel and show the power gains achieved by applying the proposed parameters.

The remainder of this paper is organized as follows. In Section II, we briefly review the CTC encoder and its internal interleaver described in 802.16 specification. The optimization methodologies for determining interleaving parameters are investigated in Section III and simulation results are discussed in Section IV. Finally, in Section V, we conclude with a brief summary of the paper.

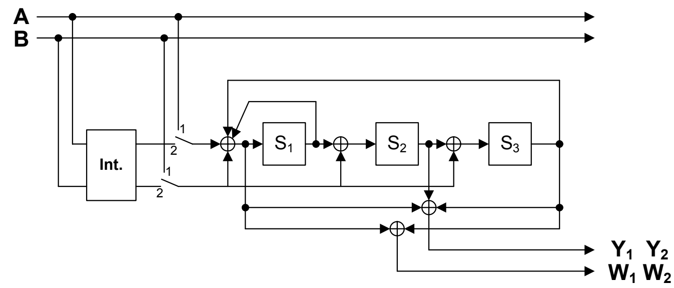

Fig. 1. CTC encoder for 802.16 systems.

II. CTC Interleaver of 802.16 Systems

A. Turbo Encoder

Fig. 1 illustrates the structure of the CTC encoder in 802.16 specification. The encoder is fed by each block (encoder packet) consisting of N_{ep} bits and the block is divided into two subblocks ($\mathbf{A}$, $\mathbf{B}$) consisting of N bits (N=$N_{ep}/2$). The encoder generates the subblocks ($\mathbf{A}$, $\mathbf{B}$) encoded systematically, the subblocks ($\mathbf{Y}_1$, $\mathbf{W}_1$) encoded by $\mathbf{A}$ and $\mathbf{B}$, and the subblocks ($\mathbf{Y}_2$, $\mathbf{W}_2$) encoded by the interleaved versions of $\mathbf{A}$ and $\mathbf{B}$.

Manuscript received January 21, 2009. The associate editor coordinating the review of this letter and approving it for publication was V. Stankovic.

This research was supported in part by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Tund) (KRF-2008-331-D00341, and in part by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

The authors are with the Department of Electrical Engineering, Kangnung National University, Gangneung, Gangwon 210-702, Republic of Korea (e-mail: psj@ieee.org; jjh@ultra30.kangnung.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2009.090143

TABLE I
INTERLEAVING PARAMETERS FOR 802.16 SYSTEMS

	Current				Optimized				Gain
N	P_0	P_1	P_2	P_3	P_0	P_1	P_2	P_3	[dB]
24	5	0	0	0	5	12	0	12	-
36	11	18	0	18	11	18	0	18	-
48	13	24	0	24	11	24	0	24	0.7
72	11	6	0	6	11	36	0	36	-
96	7	48	24	72	11	48	0	48	0.4
108	11	54	56	2	17	90	0	90	0.3
120	13	60	0	60	49	100	0	100	-
144	17	74	72	2	53	108	0	108	-
180	11	90	0	90	73	150	0	150	0.4
216	13	108	0	108	7	180	0	180	0.5
240	13	120	60	180	41	120	60	180	0.2

B. Internal Interleaver

The CTC encoder has an internal interleaver to shuffle an input data sequence and to provide the constituent encoder with a renewed sequence as shown in Fig. 1. The operation of the interleaver is explained by the following procedure.

[Step 1: Switching of alternate couples]

For $j = 0, 1, \cdots, N-1$,

$$\text{if } (j \ mod \ 2 = 0), (B_j, A_j) = (A_j, B_j), \tag{1}$$

where N is the length of a subblock, A_j and B_j represent the j-th bits of **A** and **B** respectively, and $x \ mod \ 2$ means the modulo-2 of x.

[Step 2: Interleaving address generation]

For $j = 0, 1, \cdots, N-1$,

$$\begin{array}{lll} switch & j \ mod \ 4; & \\ case\ 0: & i = (P_0 \cdot j + 1) & mod\ N \\ case\ 1: & i = (P_0 \cdot j + 1 + N/2 + P_1) & mod\ N \\ case\ 2: & i = (P_0 \cdot j + 1 + P_2) & mod\ N \\ case\ 3: & i = (P_0 \cdot j + 1 + N/2 + P_3) & mod\ N, \end{array} \tag{2}$$

where i means the interleaving address of j, and P_0, P_1, P_2 and P_3 are interleaving parameters depending on the subblock length N. Note that in (2) each interleaving address is generated by two factors, the regular factor occurred from the term $P_0 \cdot j$ and the irregular factor originated from P_1, P_2, P_3. We will call $(0, (N/2+P_1) \ mod \ N, P_2, (N/2+P_3) \ mod \ N)$ offset vector $\overline{O}$ hereafter. Due to the existence of the offset vector, the interleaving addresses become *almost regular*. They are divided by four groups and the interleaving addresses in a group have the same offset. Also, for the special case having all-zero offset vector, it is not ARP but *regular* permutation (RP) where adjacent interleaving addresses always differ by P_0. Note that P_0 should be relatively prime with N in order to generate valid interleaving addresses continuously. Table I shows the current interleaving parameters of 802.16 systems. We can observe that P_0's have no regularity except that they are prime with respect to N, and P_1, P_2, P_3 are set to $N/2, 0, N/2$ for many N and $N/2, N/4, 3N/4$ for some N.

III. INTERLEAVING PARAMETERS OPTIMIZATION

A. Optimization Methodology

Four parameters P_0, P_1, P_2, P_3 of the CTC interleaver govern the error performance of the code and thus the values need to be selected carefully. Since P_0 and P_1, P_2, P_3 have different effects on the generation of interleaving addresses as described in Section II-B and it is nearly impossible to optimize four parameters simultaneously, we divide the optimization problem of interleaving parameters into two parts, offset vector and P_0 optimization.

B. Offset Vector

Based on the observation that the most and the second most appearing offset vectors in the current specification are $(0, 0, 0, 0)$ and $(0, 0, N/4, N/4)$, we consider the following four types of offset vector.

$$\text{[Type A]} \qquad \overline{O} = (0, 0, 0, 0) \tag{3}$$

$$\text{[Type B]} \qquad \overline{O} = (0, 0, \alpha, \alpha) \tag{4}$$

$$\text{[Type C]} \qquad \overline{O} = (0, \alpha, 0, \alpha) \tag{5}$$

$$\text{[Type D]} \qquad \overline{O} = (0, 0, 0, \alpha) \tag{6}$$

Type A means all-zero offset vector in which P_1, P_2, P_3 are set to $N/2, 0, N/2$. Type B, Type C and Type D take $(0, 0, \alpha, \alpha)$, $(0, \alpha, 0, \alpha)$ and $(0, 0, 0, \alpha)$ as offset vector respectively, where $N/4$ is selected as α if N is a multiple of 16 and $N/3$ is selected if N is a multiple of 12 in order to produce valid addresses consecutively.

C. P_0 Optimization

After P_1, P_2, P_3 are fixed according to one of the above four offset vectors, all we have to do is select P_0 with which the interleaver outputs the best performance. In this paper, we consider two optimization schemes for P_0 as follows.

[Optimization 1]

In the first scheme we use the spread defined in [10], which is rewritten as

$$S(j_1, j_2) = f(j_1, j_2) + f(i_1, i_2), \tag{7}$$

where $f(u, v)$ is a circular distance defined by

$$f(u, v) = \min\{|u - v|, N - |u - v|\}. \tag{8}$$

With N and P_1, P_2, P_3 fixed, for a candidate of P_0 which is relatively prime with N, we calculate the minimum spread of the code which is given by

$$S_1 = \min_{j_1, j_2, j_1 \neq j_2} \{S(j_1, j_2)\}. \tag{9}$$

By searching for the above minimum spread for all candidates of P_0, we can find P_0 which maximizes S_1.

[Optimization 2]

The error rate curve of a turbo code is generally divided into two regions: waterfall and error-floor regions. The former is mainly affected by spread and the latter by minimum distance d_{min} of a code. Many attempts to find d_{min} of a turbo code, including the work by Perez *et al.* [11], have been done, but there was difficulty in finding it due to computational complexity. Recently, an iterative algorithm based on an impulse was suggested [12], and it was reported that the approach is quite exact in case of deploying multiple impulses [13]. In Optimization 2, we find P_0 maximizing d_{min} for given N and P_1, P_2, P_3. To calculate d_{min}, we use the double impulse method suggested in [13], when considering accuracy and complexity.

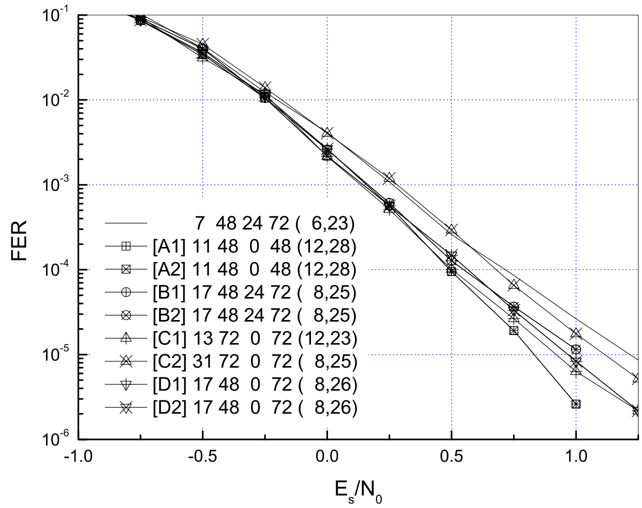

Fig. 2. FER versus SNR: $N = 96$.

IV. Simulation Results

In this section, we present simulation results for the proposed schemes and compare them with the conventional one. The CTC encoder with a code rate of 1/3 conforming to the 802.16 specification [1], [2] is applied, and quadrature phase shift keying (QPSK) and AWGN are assumed for simulation. As a decoding method, we used the Max-Log-MAP algorithm which is widely used due to its simplicity and performance. Eight iterations are executed to decode each frame and 0.75 is selected as a weighting factor of extrinsic information to enhance decoding performance. We simulated over all of the block sizes given in Table I and collected at least 100 frame errors to acquire each frame error rate.

Fig. 2 is the plot of frame error rate (FER) versus signal to noise power ratio (SNR) for N of 96. The curve with no symbol represents the performance obtained by the current interleaving parameters and it is regarded as a reference for comparison. In the legend, alphanumeric characters wrapped by a square bracket imply an applied method, i.e., [A1] denotes that Type A and Optimization 1 are used for parameter optimization. The following four numbers mean the optimized interleaving parameters P_0, P_1, P_2, P_3, and the following two numbers embraced with a round bracket mean S_1 and d_{min} by the interleaving parameters in order. In case of [A1] and [A2], we can obtain a power gain of 0.4 dB over the conventional scheme at a target FER of 10^{-5}.

In Table I, the optimized interleaving parameters and the power gains provided by the parameters are listed in accordance with N, where a target FER for measuring the power gains is set to 10^{-5}. Regular permutations [A1] and [A2] have the best performance among all schemes at N smaller than 100, but they are getting worse as N increases. This can be explained as follows. By their *regular* nature, optimizations based on A-type offset vector can have large spread S_1 when compared to others, which helps convergence in the waterfall region. On the contrary, with the increase of N, the measure S_1 becomes relatively less important and minimum distance d_{min} arises meaningfully since the waterfall region also shrinks. For moderate N, from 108 to 216, a C-type optimization presents the best performance, and a B-type optimization achieves it for N of 240. From the results, we can obtain power gains for many N by using the optimized parameters. Note that the achievable gains up to 0.7 dB are quite large since what we have to do to acquire them is only to substitute the newly optimized parameters for the current ones. As indicated, for some N such as 24, 36, 72, 120 and 144, the difference between the performance by the conventional and the proposed parameters is nearly negligible.

V. Conclusions

We have investigated the optimization problem of the CTC interleaving parameters for 802.16 systems from the observation that they have not been optimized for small and medium block sizes such as $N \leq 240$. Four types of offset vector and two optimization techniques have been considered, and thus eight schemes were examined as a combination of those. We have found that we can obtain power gains for many block sizes and the gain amounts to 0.7 dB for N of 48. Our results also evidentially showed that spread is primarily important for small block sizes, whereas minimum distance is more important for large block sizes. The proposed optimizations could be utilized in real applications since they only require to replace the current interleaving parameters and do not involve any hardware alteration.

References

[1] IEEE Std 802.16-2004, Part 16: Air interface for fixed and mobile broadband wireless access systems, Oct. 2004.

[2] IEEE Std 802.16-2004/Cor 1-2005, Part 16: Air interface for fixed and mobile broadband wireless access systems, Feb. 2006.

[3] C. Douillard and C. Berrou, “Turbo code with rate-m/(m+1) constituent convolutional codes,” *IEEE Trans. Commun.,* vol. 53, no. 10, pp. 1630–1638, Oct. 2005.

[4] C. Berrou and A. Glavieux, “Near-optimum error-correcting coding and decoding: turbo codes,” *IEEE Trans. Commun.,* vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[5] S. Crozier and P. Guinand, “High-performance low-memory interleaver banks for turbo-codes,” in *Proc. IEEE VTC,* Atlantic City, NJ, USA, Oct. 2001, pp. 2394–2398.

[6] C. Berrou, Y. Saouter, C. Douillard, S. Kerouedan, and M. Jezequel, “Designing good permutations for turbo codes: toward a single model,” in *Proc. IEEE ICC,* Paris, France, June 2004, pp. 341–345.

[7] DVB, ETSI EN 301 790, Interaction channel for satellite distribution systems, vol. 1.2.2, 2000.

[8] DVB, ETSI EN 301 958, Interaction channel for digital terrestrial television, vol. 1.1.1, 2002.

[9] IEEE C802.16maint-05/014, Improved CTC performance, Jan. 2005.

[10] S. Crozier, “New high-spread high-distance interleavers for turbo-codes,” in *Proc. 20th Biennial Symposium on Communicaitons,* Kingston, Canada, May 2000, pp. 3–7.

[11] L. C. Perez, J. Seghers, and D. J. Costello Jr., “A distance spectrum interpretation of turbo codes,” *IEEE Trans. Inform. Theory,* vol. 42, no. 11, pp. 1698–1709, Nov. 1996.

[12] R. Garello and A. V. Casado, “The all-zero iterative decoding algorithm for turbo code minimum distance computation,” in *Proc. IEEE ICC,* Paris, France, June 2004, pp. 361–364.

[13] S. Crozier, P. Guinand, and A. Hunt, “Estimating the minimum distance of turbo-codes using double and triple impulse methods,” *IEEE Commun. Lett.,* vol. 9, no. 7, pp. 631–633, Sept. 2005.