Efficient Distance Measurement Method for Turbo Codes that use Structured Interleavers

Youssouf Ould-Cheikh-Mouhamedou, Stewart Crozier, and Peter Kabal

***Abstract*— This letter presents an efficient and accurate distance measurement method for tail-biting turbo codes that use structured interleavers. This method takes advantage of the structure in the interleaver as well as the circular property of tail-biting. As such, it significantly reduces the computational complexity, which allows the accurate determination of high minimum distance ($d_{\min}$) in reasonable time. The efficiency of this method is demonstrated by its ability to determine the true $d_{\min}$ of 51 and the corresponding true multiplicities for a rate-1/3 turbo code that uses the UMTS 8-state polynomial generators and an MPEG-sized interleaver (1504 information bits) in reasonable time.**

***Index Terms*— Turbo codes, tail-biting, minimum distance, structured interleavers, DRP interleaver, DVB-RCS, UMTS.**

I. Introduction

INTERLEAVERS that yield high distances are important for lowering the "error floor" or flare of turbo codes [1], allowing them to achieve very low error rates at low to moderate signal-to-noise ratios (SNRs) [2]. A significant challenge is to determine their distance spectra or at least their minimum distances ($d_{\min}$) and corresponding multiplicities. Recently, two efficient distance measurement methods that use iterative decoding were presented in [3][4]. However, their accuracy is poor for high $d_{\min}$ interleavers, as shown in [5][6]. More accurate iterative methods were presented in [5]. As shown in [5][6], these methods find the correct $d_{\min}$ most of the time. However, the accuracy of these methods remains uncertain, especially for long interleavers that yield high $d_{\min}$ values. Even if they find the true $d_{\min}$, they cannot be guaranteed to find the correct multiplicities.

A novel and accurate distance measurement method was introduced by Garello *et al.* in [7] for single-binary turbo codes. It has been improved significantly by Rosnes in [8] and extended to tail-biting and double-binary turbo codes in [9][10]. This method tests all possible non-zero input data sequences $\boldsymbol{u}^{K-1} = (0, \cdots, 0, \chi)$, $\boldsymbol{u}^{K-2} = (0, \cdots, 0, \chi, \times)$, $\cdots$, $\boldsymbol{u}^{1} = (0, \chi, \times, \cdots, \times)$, $\boldsymbol{u}^{0} = (\chi, \times, \cdots, \times)$. Here, K is the interleaver length in symbols and 0 represents the zero-symbol (i.e., $\{0\}$ for single-binary turbo codes and $\{00\}$ for double-binary turbo codes). The variable χ is either $\{1\}$ for single-binary turbo codes or an element of $\{01, 10, 11\}$ for double-binary turbo codes. The variable $\times$ is an element of $\{0, 1\}$ or $\{00, 01, 10, 11\}$ for single- or double-binary turbo codes, respectively. For more details, see [7][10]. This method provides the true $d_{\min}$ and the true multiplicities. However, for interleavers that yield high $d_{\min}$ values, the complexity increases rapidly with $d_{\min}$, making the test impractical. This complexity can be reduced significantly for tail-biting turbo codes [11][12] that use highly structured interleavers. This is because the distance properties repeat every few data symbols. However, one must be careful when computing the multiplicities. It is not as simple as just testing a small number of indices. Examples showing this problem are discussed in the next section and a solution is also presented.

Manuscript received October 25, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Jing Li.

Y. Ould-Cheikh-Mouhamedou and S. Crozier are with the Communication Research Centre (CRC), Ottawa, Ontario, Canada (e-mail: {ymouhame, stewart.crozier}@crc.ca).

P. Kabal is with the Dept. of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada (e-mail: kabal@ece.McGill.ca).

Digital Object Identifier 10.1109/LCOMM.2006.06024.

II. Complexity Reduction

The new method is based on Garello's true distance measurement method [7][10]. In fact, the core of the algorithm remains the same as Garello's algorithm for each symbol index tested. The new method efficiently determines the true $d_{\min}$ and the true multiplicities for tail-biting turbo codes that use structured interleavers. Structured interleavers, such as dithered relative prime (DRP) interleavers [2], standard digital video broadcast with return channel via satellite (DVB-RCS) interleavers [13] and almost regular permutation (ARP) interleavers [14] have the following property:

$$\pi\left([i+M]_K\right) = \left[\pi(i) + Mp\right]_K, i = 0, \ldots, K-1 \quad (1)$$

where $[x]_K$ is x modulo K, and M is the number of repeating index increments required to implement the interleaver π. K must be a multiple of M and the integer values p and K must be relative primes to ensure that the interleaver references all symbol indices.

Without puncturing, the distance properties of tail-biting turbo codes repeat every M indices. With puncturing, they repeat every L indices if K is a multiple of L, where L is the least common multiple of M and the various mask lengths used to puncture the data and parity symbols. Thus, the $d_{\min}$ is guaranteed to be found if the first L indices are tested for all $\Delta_1 \cdot \Delta_2$ state combinations, where Δ_1 and Δ_2 are the number of starting (and ending) states in the first encoder (ENC1) and the second encoder (ENC2), respectively. However, the indices to be tested need not be the first L indices if there are at least L zero symbols between some error events.

An error event refers to the input symbols associated with a path in the trellis that departs from the all-zero state and returns to the all-zero state without passing through the all-zero state. Each input sequence $\boldsymbol{u}_{\min}$ that causes $d_{\min}$ has at least Z consecutive zero symbols that are not a part of any error events. Note that Z can be as small as 0 for very short interleavers. This Z determines the number of state combinations to be considered and the locations of the L indices to be tested. If $Z < (L-1)$, the first L indices $\{L-1, \cdots, 0\}$ must be tested considering all $\Delta_1 \cdot \Delta_2$ state

combinations. If $Z \geq (L-1)$, which is usually the case even for fairly short interleavers, only the state combinations where ENC1 starts and ends in the all-zero state need to be considered (i.e., Δ_2 state combinations). This leads to a reduction in complexity, especially if puncturing is involved. It is also enough to test the L indices $\{Z, \cdots, Z-L+1\}$. This reduces the complexity even further, especially for large Z, because the search space is reduced as more leading zero symbols are placed in front of the indices to be tested.

As mentioned above, care must be taken when determining the multiplicities. A '*shift*' of an input sequence refers to a circular shift of the input sequence by a multiple of L positions. Any input sequence that causes $d_{\min}$ can be used to *represent* all shifts of that input sequence that also cause $d_{\min}$. The multiple shifts of this input sequence will be counted later by multiplying by K/L. The goal now is to count only one representative from each unique set of shifted input sequences. The following two examples demonstrate the details associated with the determination of such representative input sequences for two cases, namely, $Z < (L-1)$ and $Z \geq (L-1)$. For the examples considered below, let L be 4.

Case ($Z < L-1$): Assume that $d_{\min}$ is caused by the representative input sequence $\boldsymbol{u}_{\min}$. Recall that all state combinations must be considered. Each non-zero symbol $\overline{\chi}$ in $\boldsymbol{u}_{\min}$, with enough zero symbols preceding it, will cause a shift of $\boldsymbol{u}_{\min}$ to be found. More precisely, H shifts of $\boldsymbol{u}_{\min}$ will be found where H is the number of $\overline{\chi}$ symbols in $\boldsymbol{u}_{\min}$ that are immediately preceded by at least b consecutive zero symbols that satisfy $b \geq [i]_L$, where i is the position of a $\overline{\chi}$ in $\boldsymbol{u}_{\min}$. This is demonstrated using the universal mobile telecommunications system (UMTS) 8-state polynomial generators [15]. Assume that $d_{\min}$ is caused by the representative single-binary input sequence $\boldsymbol{u}_{\min} = (1_0, 0, 0, 1, 0, 0, 1_2, 1, 0, 1_1, 0, 1)$, where subscripts are used for reference purposes. When testing the first $L = 4$ indices, three shifts of $\boldsymbol{u}_{\min}$ will be found (i.e., $H = 3$):

- $\boldsymbol{u}^2_{\min} = (0, 0, 1_2, 1, 0, 1_1, 0, 1, 1_0, 0, 0, 1)$
- $\boldsymbol{u}^1_{\min} = (0, 1_1, 0, 1, 1_0, 0, 0, 1, 0, 0, 1_2, 1)$
- $\boldsymbol{u}^0_{\min} = (1_0, 0, 0, 1, 0, 0, 1_2, 1, 0, 1_1, 0, 1)$

when indices 2, 1 and 0 are tested, respectively. This is because 1_2, 1_1 and 1_0 at positions 6, 9 and 0 in $\boldsymbol{u}_{\min}$ are immediately preceded by at least $[6]_4 = 2$, $[9]_4 = 1$ and $[0]_4 = 0$ zeros, respectively. Since $\boldsymbol{u}^2_{\min}$, $\boldsymbol{u}^1_{\min}$ and $\boldsymbol{u}^0_{\min}$ are shifts of $\boldsymbol{u}_{\min}$ by 4, 8 and 0 positions to the left, respectively, three shifts of $\boldsymbol{u}_{\min}$ are found. However, the goal is to count only one representative of $\boldsymbol{u}_{\min}$. One efficient solution is to recognize that when $\boldsymbol{u}^2_{\min}$ is found, $\boldsymbol{u}^1_{\min}$ and $\boldsymbol{u}^0_{\min}$ will also be found. Similarly, when $\boldsymbol{u}^1_{\min}$ is found, $\boldsymbol{u}^2_{\min}$ and $\boldsymbol{u}^0_{\min}$ will also be found. As well, when $\boldsymbol{u}^0_{\min}$ is found, $\boldsymbol{u}^2_{\min}$ and $\boldsymbol{u}^1_{\min}$ will also be found. To count $\boldsymbol{u}_{\min}$ only once, when each shift of $\boldsymbol{u}_{\min}$ is found it is counted $1/H$ times, where H is the (predicted) total number of shifts found. In this example, $H = 3$ and $\boldsymbol{u}_{\min}$ is counted only once by counting it 1/3 of the time each of the three times a shift of it is found.

Case ($Z \geq L-1$): Since ENC1 starts and ends in the all-zero state, only a $\overline{\chi}$ at the beginning of an error event could cause a shift of $\boldsymbol{u}_{\min}$ to be found. Again, this is demonstrated using the UMTS 8-state polynomial generators. Assume that $d_{\min}$ is caused by the representative single-binary input sequence $\boldsymbol{u}_{\min} = (0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1) = (0, e_1, 0, 0, 0, 0, e_2, e)$, where $e_1 = (1, 1, 1, 0, 1)$, $e_2 = (1, 1, 0, 0, 0, 1)$ and $e = (1, 0, 1, 1)$ are distinct error events. In this example, Z is 4 and the indices to be tested are $\{4, 3, 2, 1\}$. Only two shifts of $\boldsymbol{u}_{\min}$ will be found:

- $\boldsymbol{u}^2_{\min} = (0, 0, e_2, e, 0, e_1, 0, 0)$
- $\boldsymbol{u}^1_{\min} = (0, e_1, 0, 0, 0, 0, e_2, e)$

when indices 2 and 1 are tested, respectively. Since $\boldsymbol{u}^2_{\min}$ and $\boldsymbol{u}^1_{\min}$ are shifts of $\boldsymbol{u}_{\min}$ by 8 and 0 to the left, respectively, two shifts of $\boldsymbol{u}_{\min}$ are found. However, the goal is to count only one representative of $\boldsymbol{u}_{\min}$. As before, one solution is to recognize that when $\boldsymbol{u}^2_{\min}$ is found, $\boldsymbol{u}^1_{\min}$ will also be found and vice versa. In this example, $H = 2$ and $\boldsymbol{u}_{\min}$ is counted only once by counting it 1/2 of the time each of the two times a shift of it is found.

Given that an arbitrary $\boldsymbol{u}^j_{\min}$ was found while testing index j, the question now is how to recognize the other shifts of $\boldsymbol{u}^j_{\min}$ that will also be found. The answer is as follows for the two cases.

Case ($Z < L-1$): Let $\ell(i)$ be the number of consecutive zero symbols immediately preceding a $\overline{\chi}$ at position i in $\boldsymbol{u}^j_{\min}$, where $i = j+1, \cdots, K-1$ are tested for $\overline{\chi}$. From the first example given above, it follows that a shift of $\boldsymbol{u}^j_{\min}$ is guaranteed to be found during the test of index $[i]_L$ if $\ell(i) \geq [i]_L$.

Case ($Z \geq L-1$): Let $\ell_e(i)$ be the number of consecutive zero symbols immediately preceding an error event that starts at position i in $\boldsymbol{u}^j_{\min}$, where $i = j+1, \cdots, K-1$ are tested for the start of an error event. A circular shift of position i must result in a new position $i' \in \{Z, \cdots, Z-L+1\}$. From the second example given above, it follows that a shift of $\boldsymbol{u}^j_{\min}$ is guaranteed to be found during the test of index i' if $\ell_e(i) \geq i'$. It can be shown that $i' = i - L \cdot \lfloor (i - Z + L - 1)/L \rfloor$, where $\lfloor x \rfloor$ is the largest integer less than or equal to x.

Recall that H is the (predicted) total number of shifts found. Each time an input sequence $\boldsymbol{u}_{\min}$ that causes $d_{\min}$ is found, H is determined and the codeword multiplicity is increased by $1/H$. Also, the information bit multiplicity is increased by $w(\boldsymbol{u}_{\min})/H$, where $w(\boldsymbol{u}_{\min})$ is the Hamming weight of $\boldsymbol{u}_{\min}$. The overall true codeword multiplicity ($A_{d_{\min}}$) and the true information bit multiplicity ($W_{d_{\min}}$) are obtained by multiplying the multiplicities determined above by K/L.

III. Example Distance and Complexity Results

A double-binary turbo code that uses the DVB-RCS 8-state polynomial generators [13] and a single-binary turbo code that uses the UMTS 8-state polynomial generators [15] were used. The reported CPU times were obtained with a 2.4 GHz Pentium 4 (Xeon) processor. MPEG-sized (1504 information bits) interleavers were used. T_{OLD} and $T_{\text{NEW}}(Z)$ refer to CPU times (in minutes) required with the old and new methods, respectively. Results are presented for various code rates, R_c, and several Z values for the new method.

Table I shows the results for the double-binary DVB-RCS 8-state turbo encoder with the MPEG-sized standard interleaver. With this standard interleaver, $L = 4$ symbols is sufficient for all the code rates in Table I. The $T_{\text{NEW}}(Z)$ results are for

TABLE I

MINIMUM DISTANCES, MULTIPLICITIES AND CPU TIMES IN MINUTES FOR THE DVB-RCS ENCODER WITH MPEG-SIZED STANDARD INTERLEAVER.

R_c	1/3	2/5	1/2	2/3	4/5
$d_{\min}$	33	27	19	12	9
$A_{d_{\min}}$	376	376	376	188	3572
$W_{d_{\min}}$	3384	3384	3384	1316	20680
T_{OLD}	351	353	120	52	240
$T_{\text{NEW}}(Z=3)$	6.95	7.18	2.10	1.66	3.65
$T_{\text{NEW}}(Z=150)$	3.36	3.00	0.91	0.38	2.25

TABLE II

MINIMUM DISTANCES, MULTIPLICITIES AND CPU TIMES IN MINUTES FOR THE DVB-RCS ENCODER WITH NEW MPEG-SIZED DRP INTERLEAVERS.

R_c	1/3	2/5	1/2	2/3	4/5
$d_{\min}$	40	30	22	14	10
$A_{d_{\min}}$	1128	1504	3760	188	7332
$W_{d_{\min}}$	7332	9024	28388	1692	41924
T_{OLD}	10153	751	482	854	1215
$T_{\text{NEW}}(Z=3)$	482	29	14	17	22
$T_{\text{NEW}}(Z=150)$	270	1.80	2.91	7.20	10.88

$Z = L - 1 = 3$ symbols (6 bits) and $Z = 150$ symbols (300 bits).

Table II shows the results obtained with new MPEG-sized DRP interleavers for the DVB-RCS encoder. With these new interleavers, $L = 4$ symbols is sufficient for the code rates in Table II, except for rate 1/3 where $L = 8$. As an example, for rate 2/5, the use of $Z = 3$ and $Z = 150$ symbols reduced the execution times by factors of 25 and 400, respectively, compared to the old method. Note that for rate 1/3, the new DRP interleaver gives a $d_{\min}$ of 40, whereas the standard interleaver gives a $d_{\min}$ of 33.

Table III shows the results for the single-binary UMTS 8-state turbo encoder with new MPEG-sized DRP interleavers. With these new interleavers, $L = 8$ bits is sufficient for all the code rates in Table III. The accurate determination of $d_{\min} = 51$ would not be possible in reasonable time without the use of the new method. The reported T_{OLD} values for code rates 1/3, 2/5 and 1/2 are optimistic estimates obtained by testing only a subset of indices.

The $T_{\text{NEW}}(Z \geq 150)$ results in Tables I, II and III show a typical reduction in execution time by a factor of 40 to 400. The Z values of 150 symbols, 150 symbols and 200 bits used in Tables I, II and III, respectively, were obtained using safe lower bounds on the maximum number of zero symbols that are sure to occur between error events. These bounds depend on the constituent encoders, the number of error events, and the structure of the interleavers. The maximum Z values that could be used are likely much higher than those used above. Future work includes finding tighter lower bounds on the maximum Z values, so complexity can be reduced further.

A comparison between $d_{\min}$ and the CPU times reported in Tables I, II and III shows that an increasing $d_{\min}$ value results in a significant increase in execution time. This demonstrates the importance of efficient distance measurement methods.

Combining this new method with the significant improvement achieved recently by Rosnes [8] will enable the execution times to be reduced even further.

TABLE III

MINIMUM DISTANCES, MULTIPLICITIES AND CPU TIMES IN MINUTES FOR THE UMTS ENCODER WITH NEW MPEG-SIZED DRP INTERLEAVERS.

R_c	1/3	2/5	1/2	2/3	4/5
$d_{\min}$	51	38	28	14	9
$A_{d_{\min}}$	940	376	1692	376	2068
$W_{d_{\min}}$	7708	2256	9588	1692	10152
T_{OLD}	302400	129600	34560	1421	504
$T_{\text{NEW}}(Z=7)$	12108	6468	1353	17	6.3
$T_{\text{NEW}}(Z=200)$	5578	908	651	10.35	3.53

IV. CONCLUSION

A very efficient distance measurement method for tail-biting turbo codes that use structured interleavers was presented. The efficiency of this method was demonstrated for both single- and double-binary turbo codes, using structured interleavers that have high minimum distances for various code rates. Taking advantage of the interleaver structure and the circular property of tail-biting, the execution times were reduced by a factor of 40 to 400. This means much larger interleavers with higher distances can be tested using this true $d_{\min}$ measurement method.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC'93)*, pp. 1064–1070.

[2] S. Crozier and P. Guinand, "Distance upper bounds and true minimum distance results for turbo-codes designed with DRP interleavers," in *Proc. 3^{rd} Int. Symp. Turbo Codes 2003*, pp. 169–172.

[3] C. Berrou, S. Vaton, M. Jézéquel, and C. Douillard, "Computing the minimum distance of linear codes by the error impulse method," in *Proc. IEEE Globecom 2002*, pp. 10–14.

[4] R. Garello and A. Vila, "The all-zero iterative decoding algorithm for turbo code minimum distance computation," in *Proc. IEEE Int. Conf. Commun. (ICC'04)*, pp. 361–364.

[5] S. Crozier, P. Guinand, and A. Hunt, "Computing the minimum distance of turbo-codes using iterative decoding techniques," in *Proc. 22^{nd} Biennial Symposium Commun. 2004*, pp. 306–308.

[6] Y. Ould-Cheikh-Mouhamedou, S. Crozier, and P. Kabal, "Comparison of distance measurement methods for turbo codes," in *9^{th} Canadian Workshop on Information Theory (CWIT'05)*, pp. 36–39.

[7] R. Garello, P. Pierleoni, and S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 800–812, May 2001.

[8] E. Rosnes and O. Ytrehus, "Improved algorithms for the determination of turbo-code weight distributions," *IEEE Trans. Commun.*, vol. 53, pp. 20–26, Jan. 2005.

[9] E. Rosnes and O. Ytrehus, "An efficient algorithm for tailbiting turbo code weight distribution calculation," in *Proc. 3^{rd} Int. Symp. Turbo Codes*, pp. 439–442.

[10] Y. Ould-Cheikh-Mouhamedou, S. Crozier, and P. Kabal, "Distance measurement method for double binary turbo codes and a new interleaver design for DVB-RCS," in *Proc. IEEE Globecom 2004*, pp. 172-178.

[11] S. Crozier, P. Guinand, J. Lodge, and A. Hunt, "Construction and performance of new tail-biting turbo codes," in *Proc. of the 6^{th} Int. Workshop on Digital Signal Processing Techniques for Space Applications (DSP'98)*.

[12] C. Berrou, C. Douillard, and M. Jézéquel, "Multiple parallel concatenation of circular recursive convolutional (CRSC) codes," *Annals Telecommun.*, vol. 54, pp. 166–172, Mar.-Apr. 1999.

[13] European Telecommunications Standards Institute, "Interaction channel for satellite distribution systems." ETSI EN 301 790, V1.3.1, Mar. 2003.

[14] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan, and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model," in *Proc. IEEE Int. Conf. Commun. (ICC'04)*, pp. 341–345.

[15] "3^{rd} generation partnership project (3GPP) technical specification group: Universal mobile telecommunications system (UMTS); multiplexing and channel coding (FDD), TS 25.212 v3.4.0," Sept. 2000.