September 2004

Computation of the free distance and low weight distribution of turbo codes with convolutional interleavers

Sina Vafi
University of Wollongong, sina@uow.edu.au

Tadeusz A. Wysocki
University of Wollongong, wysocki@uow.edu.au

Computation of the free distance and low weight distribution of turbo codes with convolutional interleavers

Abstract

This work presents an algorithm for computation of the free distance parameter of turbo codes with a convolutional interleaver implemented to act as a block interleaver. Based on the properties of the interleaver and the algorithm applied, we can also determine the other low weights of turbo codes and this is useful in determining the performance of the error floor of turbo codes. For different turbo code structures, the relevant parameters have been computed. The error rate simulations confirm the algorithm results.

Keywords

convolutional codes, error statistics, interleaved codes, turbo codes

Disciplines

Physical Sciences and Mathematics

Publication Details

This paper originally appeared as: Vafi, S & Wysocki, TA, PIMRC 2004 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 5-8 September 2004, 2, 1356-1359. Copyright IEEE 2004.

COMPUTATION OF THE FREE DISTANCE AND LOW WEIGHT DISTRIBUTION OF TURBO CODES WITH CONVOLUTIONAL INTERLEAVERS

Sina Vafi, Tadeusz Wysocki

University of Wollongong,Northfields Ave,2500,Wollongong, Australia
{sv39,wysocki}@uow.edu.au

Abstract - This paper presents an algorithm for computation of the free distance parameter of turbo codes with a convolutional interleaver implemented to act as a block interleaver. Based on the properties of the interleaver and the algorithm applied, we can also determine the other low weights of turbo codes and this is useful in determining the performance of the error floor of turbo codes. For different turbo code structures, the relevant parameters have been computed. The error rate simulations confirm the algorithm results.

Keywords - Turbo codes,convolutional interleaver,free distance,low weight distribution.

I. INTRODUCTION

Turbo code performance for moderate to high signal to noise ratios, where error floor occurs, is determined by the free distance (d_{free}) parameter, which is defined as the minimum Hamming distance between any possible codewords. Considering all input data streams in the determination of d_{free} is impossible, especially for medium to large input block lengths. Finding a suitable algorithm to compute the d_{free} for turbo codes has been considered in previous works, e.g.[1][2][3], which focus on those input data streams that possibly cause the free distance. The method presented in [1] considers unconstrained input block length and traces codewords with the minimum weight from low weight input data streams that return the Recursive Systematic Convolutional (RSC) encoders to the zero state. When increasing the interleaver length, a higher d_{free} value for turbo codes is expected. Therefore,input data streams with higher weights should be involved in the computation, which increases the complexity and lessens the reliability of the method. Garello, et.al [2] have presented an algorithm that removes this problem by computing the minimum distance of a constrained Sub-Code (CSC). Ref[3] outlines an improvement on this latter algorithm useful for high rate turbo codes with non-punctured constituent codes. The proposed algorithms are implemented using block interleavers in a turbo encoder. Application of such an interleaver creates d_{free} with high multiplicities, which lessens the turbo code performance [4]. In [5], we suggest a new structure for a block interleaver implemented as a convolutional interleaver by inserting the number of stuff bits equal to the number of interleaver memories. In comparison with the block interleaver, the convolutional interleaver uses less memory and this simplifies its hardware design with more compatibility for processing of different data lengths. This paper presents an algorithm for the d_{free} computation of turbo codes with convolutional interleavers. This algorithm is applicable also to determining other low codeword weights,which is helpful in order to evaluate precisely the error floor performance. The organization of this paper is as follows: Section 2 presents briefly the new convolutional interleaver structure. In Section 3, the proposed algorithm for computing the free distance and other low weights is explained and finally Section 4 concludes the paper.

II. CONVOLUTIONAL INTERLEAVER STRUCTURE

A convolutional interleaver consists of T parallel lines that represent the interleaver period. In general, each successive interleaver line has M more delay elements than the previous line [6]. In a turbo encoder structure, the interleaved data block is obtained by inserting zero bits into the convolutional interleaver memories to clear their contents. Depending on the length of the input data stream, the interleaved data will be terminated at one of the interleaver lines. This can be determined by the result of Rem(L,T), where L and Rem(L,T) are the interleaver length and the reminder of L/T operation,respectively. For the interleaver with T=3, M=1 and the input data stream $\{x_0, x_1, x_2, ..., x_{L-1}\}$,the interleaved data blocks corresponding to the different Rem(L,T)values would be :

Rem(L,T)=1:

$\{x_0, 0, 0, x_3, x_1, 0, x_6, x_4, ..., x_{L-3}, x_{L-5}, 0, 0, x_{L-2}\}$

Rem(L,T)=2:

$\{x_0, 0, 0, x_3, x_1, 0, x_6, ..., x_{L-4}, x_{L-6}, 0, x_{L-1}, x_{L-3}\}$

Rem(L,T)=0:

$\{x_0, 0, 0, x_3, x_1, 0, x_6, ..., x_{L-7}, 0, x_{L-2}, x_{L-4}, 0, 0, x_{L-1}\}$

Since insertion of stuff bits reduces the usage of channel bandwidth, an optimisation should be performed on the interleaver to ensure that the number of stuff bits is equal to the number of interleaver memories. This can be achieved by adding a zero deletion block after interleaving, which deletes extra zero stuff bits that are inserted at the end of each block. In this case, the memory contents at the end

0-7803-8523-3/04/$20.00 ©2004 IEEE.

of each block have a zero value which remains until the beginning of the next block [5]. Considering the interleaver in the above example, the optimised interleaved data block would be :

Rem(L,T)=1:

$\{x_0, 0, 0, x_3, x_1, 0, x_6, x_4, ..., x_{L-3}, x_{L-5}, x_{L-2}\}$

Rem(L,T)=2:

$\{x_0, 0, 0, x_3, x_1, 0, x_6, ..., x_{L-4}, x_{L-6}, x_{L-1}, x_{L-3}\}$

Rem(L,T)=0:

$\{x_0, 0, 0, x_3, x_1, 0, x_6, ..., x_{L-2}, x_{L-4}, x_{L-1}\}$

In designed turbo codes, trellis termination and truncation are applied for the first and the second RSC encoders, respectively. Therefore, when stuff bits are inserted into the convolutional interleaver after the trellis termination, they have no effect on the systematic and the first parity data. Hence, these bits can be eliminated from the end part of this data. Based on the outlined characteristics, an algorithm is presented here to compute free distance and other low weights of the full rate turbo codes using the convolutional interleaver with M=1 and an arbitrary period. This procedure will be explained in the next section.

III. Algorithm for Computing the Free Distance and Low Weight Distribution

The applied algorithm, similar to the one presented in [1], considers low weight input data streams that return the first RSC encoder to the zero state. In the optimised convolutional interleaver, the distance between two adjacent bits, before and after interleaving, is equal to the interleaver period, except for the end part of the interleaved data block, where zero bit deletion is conducted to reduce the overall stuff bits number. In this part, the minimum distance would be less than the interleaver period and we expect that the data interleaved in this part will create lower weight than the data interleaved by other parts in the second RSC encoder [7]. Regarding this interleaver structure and considering the effect of the tail bits on the overall weight of the turbo code, we can present an algorithm to estimate d_{free} as follows:

Firstly, among all input data block streams with the minimum weight(i.e.1) those which return the first RSC encoder to the zero state are selected. Then, their bit 1 positions will be compared with the bit positions that have been located at the end part of the interleaved data. If this condition is established, the overall codeword weight is computed and stored as a d_{free} value. In order to identify the other input data streams with the mentioned property the same comparison will be performed and the minimum obtained d_{free} value considered as d_{free} at the end of the first step. A similar procedure is followed for higher input data weights until the computed d_{free} is lower than or equal to the weight of the input data stream. The final d_{free} is allocated as d_{free} of the turbo code.

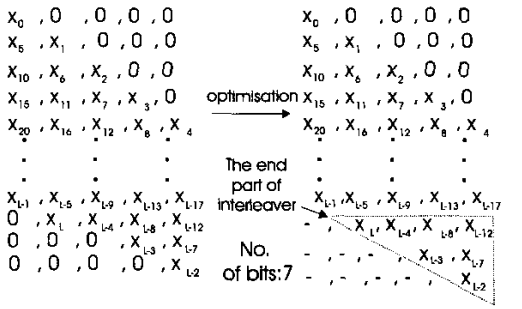

Fig. 1
Interleaved data for interleaver (T=5,M=1),Rem(L,T)=2.

Since bit 1 positions should be located at the end part of the interleaver, the pattern length consisting of all the 1s inside the data stream, should not exceed the bit number of the end part of the interleaver. For this purpose, we can encode all the low weight patterns with a length equal to the bit number of the end part of the interleaver and then, the ones that return the RSC encoder to the zero state are selected. In this case, the algorithm considers all input data streams which contain the patterns selected. This procedure is helpful for determining the free distance value of turbo codes with medium to large interleaver lengths, where the process of examining all the low weight data streams is long and even impossible to perform.In simulations,which have been conducted , input data weights of up to 4 using short period interleavers have been considered in order to simplify deinterleaver synchronization.

For example,as shown in Figure 1, the number of bits at the end of the interleaver with (M=1,T=5),Rem(L,T)=5 is equal to 7 and the patterns with length 7 that return the RSC encoder (1,7/5) to the zero state have been presented in Table 1. It is obvious that the algorithm covers all the patterns shifted cyclically that satisfy the above condition.

Table 1
Patterns returning the RSC encoder (1,7/5)to the zero state for the convolutional interleaver (T=5,M=1),Rem(L,T)=2.

x_j	x_{j+1}	x_{j+2}	x_{j+3}	x_{j+4}	x_{j+5}	x_{j+6}
1	0	0	0	0	0	1
1	0	0	0	1	1	0
1	1	0	0	0	1	0
1	1	0	0	1	0	1
1	0	1	0	0	1	1
1	1	0	1	1	0	0
1	0	1	1	0	1	0
1	0	0	0	1	1	0
1	0	0	1	0	0	0
1	1	1	0	0	0	0

The computed d_{free} values of turbo codes (1,7/5)(1,35/23)

Table 2
Free distance parameters for turbo codes (1,7/5) with interleaver T=3.

L	d_{free}	N_{free}	$\tilde{w}_{free}$
165	7	1	2
166	6	1	3
167	9	1	2

Table 3
Free distance parameters for turbo codes (1,35/23) with interleaver T=3.

L	d_{free}	N_{free}	$\tilde{w}_{free}$
165	10	3	3
166	9	1	3
167	10	2	3

Table 4
Free distance parameters for turbo codes (1,7/5) with interleaver T=5.

L	d_{free}	N_{free}	$\tilde{w}_{free}$
1020	8	1	3
1021	8	1	3
1022	8	1	3
1023	8	1	3
1024	8	2	3

Table 5
Free distance parameters for turbo codes (1,35/23) with interleaver T=5.

L	d_{free}	N_{free}	$\tilde{w}_{free}$
1020	11	2	4
1021	11	2	4
1022	11	2	3.5
1023	11	2	3
1024	10	1	3

for the convolutional interleavers (M=1,T=3),(M=1,T=5) with different lengths have been presented in Tables 2 to 5, where N_{free} and $\tilde{w}_{free}$ represent the total number or multiplicities of the codewords with weight d_{free} and the average input data weight related to d_{free}, respectively. The results confirm that free distance value will be increased by increasing the interleaver period or the constraint length of the RSC encoders.

In comparison with the block interleaver, the convolutional interleaver generates d_{free} with fewer multiplicities and this can be considered as an advantage. However, a turbo code with acceptable performance is achieved by the relatively high interleaver period, and this increases the stuff bits number. Therefore, we need to compromise the turbo codes performance and the stuff bits number.

Depending on the interleaver period, the bit numbers which are located at the end part of the interleaver alter. In order for the algorithm to be usable for different interleaver periods we can decrease or increase the area of the end part of the interleaver relative to the interleaver period. For example, in the free distance calculating for turbo codes (1,7/5) and (1,35/23) with the interlaver period T=3 in addition to the end part of the interleaver, two rows of the interleaved data on top of this part have been considered,too.

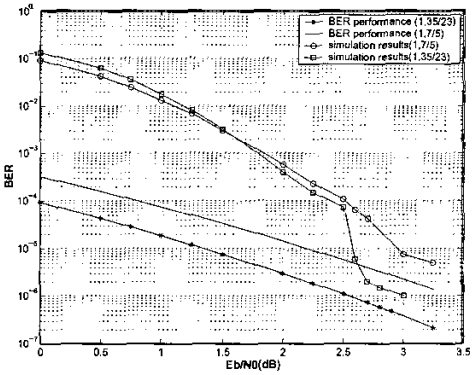

Fig. 2
Simulation result for turbo codes (1,7/5) and (1,35/23) with interleaver (T=10,M=1),L=1024.

Based on this definition it is possible to find out the other low weights of a turbo code, which weights are obtained from the considered end part of the interleaver. It has been seen that increasing the input data stream length has no effect on the multiplicities of d_{free} or some of other low weights.This has been confirmed by considering different short input data lengths such that the free distance and the other low weights can be determined by definition of the weight calculation [1] [5].Tables 6 to 9 give specifications of weights higher than d_{free} for the above mentioned turbo code structures. In these tables,N_d and $\tilde{w}_d$ are the number of codewords and the average weight of the input data, corresponding to codewords with weight d,respectively. Based on the low weight distributions listed above, we can evaluate the full rate turbo codes' performance in the error floor region. This is carried out by calculating Bit Error Rate(BER) versus signal to noise ratio in the Additive White Gaussian Noise (AWGN) channel with the following equation [4]:

$$BER \cong \sum_{d=d_{free}}^{3(L+m)} \frac{N_d \tilde{w}_d}{L} Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right) \qquad (1)$$

where $\frac{E_b}{N_0}$,m and R are the signal to noise ratio, the number of tail bits and the code rate,respectively. Conducted simulation for turbo codes(1,7/5)and(1,35/23)with the interleaver length L=1024 and (M=1,T=10)gives ($d_{free} = 10/N_{free} = 3/\tilde{w}_{free} = 3$)($d_{free} = 11/N_{free} = 1/\tilde{w}_{free} = 3$) characteristics. These free distance values together with other

low weight values have been applied in the above equation and compared with the performance of turbo codes at the end of decoder using iterative decoding technique with the Soft Output Viterbi Algorithm (SOVA) and 10 iterations. The results indicate that the proposed algorithm is useful to access the turbo code's performance in the error floor region for such a convolutional interleaver. Table 10 gives specification of the weights applied in the simulations.

IV. Conclusions

In this paper, we have presented an efficient and simple algorithm to compute the minimum distance value of turbo codes using a convolutional interleaver as a block interleaver. The algorithm has been applied for different turbo code structures with different interleaver lengths. Then, the other low weights have been determined and used to evaluate the accurate error floor performance.The accuracy of the method has been confirmed by the simulation of the iterative turbo decoding performance for moderate to high signal to noise ratios region.

Table 6
Low weight parameters for turbo codes (1,7/5) with interleaver T=3.

Rem(L,T)	0	1	1	2	2
d	8	7	8	7	8
N_d	1	1	1	1	1
$\tilde{w}_d$	3	2	3	3	2

Table 7
Low weight parameters for turbo codes (1,35/23) with interleaver T=3.

Rem(L,T)=0			Rem(L,T)=1			Rem(L,T)=2		
d	N_d	$\tilde{w}_d$	d	N_d	$\tilde{w}_d$	d	N_d	$\tilde{w}_d$
11	2	3	10	2	3	11	3	3.3
12	5	3.8	11	6	4	12	7	4
13	9	4	12	5	4	13	10	4

Table 8
Low weights parameters for turbo codes (1,7/5) with interleaver T=5.

Rem (L,T)	0	0	1	1	2	2	3	3	4	4
d	9	10	9	10	9	10	9	10	9	10
N_d	3	4	3	5	3	5	1	3	3	3
$\tilde{w}_d$	1.6	2.2	2.3	2.8	2.6	2.4	3	2.3	2	2

Table 9
Low weights parameters for turbo codes (1,35/23) with interleaver T=5.

Rem (L,T)	0	0	1	1	2	2	3	3	4	4
d	12	13	12	13	12	13	12	13	11	12
N_d	1	8	2	6	3	5	2	5	1	5
$\tilde{w}_d$	3	4.1	3.5	3	3.6	4.4	4	4.2	3	4.2

Table 10
weight parameters for turbo codes (1,7/5)(1,35/23) with interleaver T=10 and length L=1024.

Weight	Turbo code(1,7/5)		Turbo code(1,35/23)	
d	N_d	$\tilde{w}_d$	N_d	$\tilde{w}_d$
10	3	3	0	0
11	1	2	1	3
12	1	4	0	0
13	3	2.3	0	0
14	11	3.3	0	0
15	7	3	9	3.3
16	20	3.5	1	3
17	17	3.2	5	4
18	25	3.1	11	3.7
19	34	4	9	3.4
20	43	3.75	6	3.6
21	57	3.54	16	3.375
22	119	4	29	4.48
23	29	3	33	4
24	111	4	35	3.75
25	-	-	38	3.74
26	-	-	60	4.1

References

[1] J. Seghers, "On the free distance of turbo codes and related product codes," *Swiss federal Institute of technology,Zurich,Switzerland,Diploma project 6613*, vol. 42, Aug. 1995.

[2] P.Pierleoni R.Garello and S.Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: algorithms and applications," *IEEE Journal on selec. areas in commun.*, vol. 19, pp. 800–812, May 2001.

[3] E.Rosnes and O.Ytrehus, "Improved algorithms for high rate turbo code weight distribution calculation," *ICT*, vol. 1, pp. 104–110, March 2003.

[4] L. C. Perez, J. Seghers, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans.Inform.Theory*, vol. 42, no. 1, pp. 1698–1709, Nov. 1996.

[5] S.Vafi, T.Wysocki, and I.Burnett, "Convolutional interleaver for unequal error protection of turbo codes," *7th international DSPCS and 2nd WITSP,Coolangatta, Australia.*, pp. 485–491, Dec. 2003.

[6] G.D.Forney, "Burst-correcting codes for the classic bursty channel," *IEEE Trans. Commun.*, vol. COM-19, pp. 772–781, Oct. 1971.

[7] S.Dolinar and D.Divsalar, "Weight distributions for turbo codes using random and non-random permutations," *TDA progress report*, , no. 42-122, pp. 56–65, Aug. 1995.