Design of a Deterministic Interleaver for Turbo Codes and Proof of its Completeness

Xingcheng Liu
Department of Electronics and Communications Engineering
Sun Yat-Sen University, Guangzhou 510275, China
e-mail: xcl02v@ecs.soton.ac.uk

Abstract - A new turbo interleaver is proposed and the proof of its mathematical completeness is provided. Our simulation results demonstrate that the BER performance of the associated turbo codec is similar to that of the turbo codec employing a classic random interleaver, despite the simple implementation of the proposed scheme. Even though the interleaver proposed here is a random one, the permuted results are deterministic, when the system parameter h is set to a fixed integer. This characteristic is especially useful for reducing the hardware requirements imposed by the interleaving and deinterleaving operations.

I. Introduction

Since the invention of turbo codes [1] in 1993, their importance has grown substantially, finding numerous applications in practical systems. Since the choice of the turbo interleaver determines the turbo codec's achievable performance, interleaving schemes have been the subject of intensive study in the literature [2–10]. However, in real-time interactive applications, such as voice communications, the time delay arising from interleaving has to be minimized. The main objective of this paper is to design short interleavers relying on modulo operations. Numerous authors have suggested interleaver designs that are suitable for short block sizes [2–5]. Generally speaking, it is difficult to guarantee a unique, unambiguous one-to-one mapping between the input and output operands, when using modulo operations. In this contribution we provide a solution to this ambiguity problem and provide the proof of the uniqueness of the algebric transformation used by the proposed interleaver. As a further application we also found this interleaving scheme useful in generating self-similar traffic streams, when the interleaving length is in excess of 2^{15}.

There are two major criteria predetermining the design of an interleaver [5–8, 11–13]. Specifically, the weight distribution or distance spectrum properties of the code, and the correlation between the specific soft output values of each of the constituent decoder, in particular, between the parity bits and the original bits.

This work was partially carried out in the Department of Electronics and Computer Science, University of Southampton, UK during a sabbatical leave from Sun Yat-Sen University.

It is widely recognised that an error floor will occur in the BER curve of turbo codes at high signal-to-noise ratios [1]. Increasing the interleaver length N increases the achievable minimum distance of the code [11–13], which plays an important role in determining the performance of turbo codes at low BERs. Furthermore, when the information provided by one of the decoders for the other decoder is less correlated with the original input information sequence, the performance of iterative decoding also improves. Based on these correlation-related characteristics, an interleaver design criterion relying on the weight distribution was proposed in [7]. Finally, the trellis termination of turbo codes is critical concerning their minimum distance d_{min} and the achievable performance [14].

The paper is organized as follows. In Section II, the proposed interleaving scheme is introduced and the proof for its completeness is given. The performance of turbo codes when using the proposed interleaver is characterized in Section III. Finally, our conclusions are drawn in Section IV.

II. Design of an Interleaver for Turbo Codes and the Proof of its Completeness

In [15], a design method was proposed for generating turbo interleavers. The focus of discussions was on a series of search operations invoked for acquiring important error patterns first, then initializing the interleaver generation process, and finally iteratively carrying out the interleaver generation process, until the required interleaving length was achieved. In contrast to this method [15], we propose a non-iterative interleaver design procedure relying on a number of mapping operations or transformations. The proposed procedure is formulated as follows.

Let us denote the position of the information bits by the index k, commencing from 0. The interleaver length N meets the constraint of $N = 2^m$ for $m \geqslant 2$. First we define a set of mapping steps as follows:

(i) mapping f: $k \mapsto k(k+1)/2 \bmod N$;

(ii) mapping g: $f(k) \mapsto f((k+h) \bmod N)$ for $0 < h < N$, which corresponds to a simple shift operation.

Based on the mapping operations defined above, we can formulate a theorem as follows.

Theorem

Assume N and m are positive integers and $N = 2^m$, while k is an integer obeying $k \in [0,N\text{-}1]$. If we use the

mapping functions f and g, then the mapping from k to g is an unambiguous one-to-one full mapping.

Before providing the proof of this theorem, we provide its interpretation. During the design of the interleavers, no matter how the positions of the information bits have changed, we always have to guarantee that there exists one and only one new position after the mapping corresponding to the original information bit's position. This unique, one-to-one mapping ensures that the original bits can be unambiguously mapped back to their original positions after deinterleaving.

Proof

The proof of the above theorem will be provided in two steps with respect to the mapping operations (i) and (ii).

(i) Consider the integers s and t, both residing in the range of $[0, N-1]$, while we have $s \neq t$. Without loss of generality, we may assume that $s > t$. Then the goal of the proof is to show that $f(s) \neq f(t)$. We adopt the method of contradiction for providing the proof.

Assume that $f(s) = f(t)$. Then there is an integer $n > 0$, which obeys the following formula:

$$s(s+1)/2 = t(t+1)/2 + 2^m n, \tag{1}$$

consequently yielding:

$$(s-t)(s+t+1) = 2^m n. \tag{2}$$

Let $s = t + x$, where we have $x \geqslant 1$. From Equation (2), we have

$$x(2t+x+1)/n = 2^{m+1}. \tag{3}$$

Let furthermore $n = n_1 n_2$, obviously n_1 and $n_2 \geqslant 1$. Insert n into Eq. (3), yielding:

$$(x/n_1)((2t+x+1)/n_2) = 2^{m+1}. \tag{4}$$

Eq. (4) can be decomposed into two formulae, namely into:

$$x/n_1 = 2^{m_1} \tag{5}$$

$$(2t+x+1)/n_2 = 2^{m_2}, \tag{6}$$

where we have m_1 and $m_2 \geqslant 0$, and $m_1 + m_2 = m + 1$. From Eqs. (5) and (6), we get

$$2t + 1 = 2^{m_2} n_2 - 2^{m_1} n_1. \tag{7}$$

When none of m_1 and m_2 is equal to 0, the right-hand side in Eq. (7) is even, but the left-hand side is odd, which yields a contradiction. Hence, the original assumption is false. Consequently, we have $f(s) \neq f(t)$.

Since m_1 and m_2 cannot be 0 at the same time according to Eqs. (4)-(6), there are two circumstances left for our discussion, in which either we have $m_1 = 0$ or $m_2 = 0$.

(a) If $m_1 = 0$, then $m_2 = m + 1$. According to Eqs. (5) and (6), we have

$$s - t = n_1, \tag{8}$$

$$s + t + 1 = 2^{m+1} n_2. \tag{9}$$

Because both t and $s \leqslant 2^m - 1$ are integers, consequently we have:

$$t + s + 1 \leqslant 2^{m+1} - 1. \tag{10}$$

Again, because $n_2 \geqslant 1$, from Eq. (9) we have:

$$s + t + 1 \geqslant 2^{m+1}. \tag{11}$$

Obviously, Eqs. (10) and (11) are in contradiction.

(b) If $m_2 = 0$, then we have $m_1 = m + 1$, hence we can rewrite Eqs. (5) and (6) as:

$$(s-t)/n_1 = 2^{m+1}, \tag{12}$$

$$(s+t+1)/n_2 = 1. \tag{13}$$

From Eq. (12) we have:

$$s - t = 2^{m+1} n_1 \geqslant 2^{m+1}. \tag{14}$$

However, since $s \leqslant (N-1) = 2^m - 1$ and $t > 0$, consequently we have

$$s - t < 2^{m+1}. \tag{15}$$

Therefore, a contradiction occurs between Eqs. (14) and (15).

The above proof confirmed the first part of the theorem.

(ii) For the mapping operation of :

$$f(k) \mapsto f((k+h) \mod N) \quad for \quad 0 < h < N \tag{16}$$

the proof is more straightforward than that of part (i) above. To proceed further, we can divide the region into two parts, where the index of one part satisfies $k + h \geqslant N$ and the other $k + h < N$.

When $k+h \geqslant N$, i.e. when $k \geqslant N-h$, all the elements of the right-hand of Eq.(16) that satisfy the relation $f(k) \mapsto f((k+h) \mod N$ become a set, denoted as S_1. Obviously, the set is given by $S_1(k) = \{0, 1, 2, ..., h-1\}$.

On the other hand, all the right-hand-side elements of Eq.(16) that satisfy the same relation as the above become another set, when $k + h < N$, i.e. $k < N - h$, denoted as S_2. Again, the set is specified as $S_2(k) = \{h, h+1, h+2, ..., N-1\}$. Accordingly, we obtain

$S_1(k) \bigcap S_2(k) = \Phi$.

Because there are h different elements in the set S_1 and $(N-h)$ seperate elements in the set S_2, this mapping is a unique one.

To summarize according to (i) and (ii), we can draw the following conclusion: the mapping from k to g is a unique one-to-one mapping. □

Taking $N = 16$ and $h = 1$ as an example, we can exemplify the mapping process as follows:

$(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$

$\mapsto (0, 1, 3, 6, 10, 15, 5, 12, 4, 13, 7, 2, 14, 11, 9, 8)$

$\mapsto (1, 3, 14, 6, 13, 12, 10, 2, 0, 8, 15, 9, 4, 7, 11, 5)$

The implementation of this interleaver can be described as follows.

(1) Select a suitable interleaving length N in order to meet $N = 2^m$;

(2) Map the set T_1 consisting of N integers between 0 and $(N - 1)$ to the set T_2 according to the rule f;

(3) Carry out the shift mapping from the set T_2 to T_3, end the interleaving process.

III. Performance of the Interleaver

The performance of the proposed interleaver was evaluated using the following parameters. The octally represented generator polynomials of the constraint-length 5 constituent codes are (23,35). The interleavers' length N is 128, 256, and 1024, respectively. The SOVA algorithm was used. The corresponding simulation results were shown in Figures 1-3. Fig. 1 shows the corresponding BER versus SNR curves in conjunction with different interleaver lengths. As shown in the figure, when we have N=1024 bits, the BER is below 10^{-3}, if the SNR is higher than 1.2dB.

To elaborate a little further, when the interleaving length is 1024 bits, the corresponding performance curves are provided in Fig. 2 for comparison between the interleaver proposed here and that of Liu et al [16]. The figure shows that the required SNR is decreased by 1dB in comparison to using the interleaver of Liu et al at $BER = 10^{-3}$.

By contrast, in Fig. 3 the performance curves were obtained using different interleaving schemes. The length of the interleavers used is similar, having 128 or 169 bits. As shown in the figure, the performance at this short interleaver length is marginally better, than that of Liu et al [16], but 1dB worse than that of [15]. This is because the interleaver of [15] was obtained using step-by-step iterative search, which improved the distance spectrum of the codewords and also because the interleaving length was slightly longer. However, the implementation of the latter scheme was significantly more complex.

Although the simulations were carried out by using the proposed interleaver having a short length of 128 bits, we found it useful in generating self-similar traffic streams, where the implemented length of the interleaver was up to $N = 2^{15}$, which will be the topic of another paper.

IV. Conclusion

Although there are numerous interleavers proposed in the literature [2–7, 9, 15], their mathematical completeness has not been shown so far to our knowledge. Here,

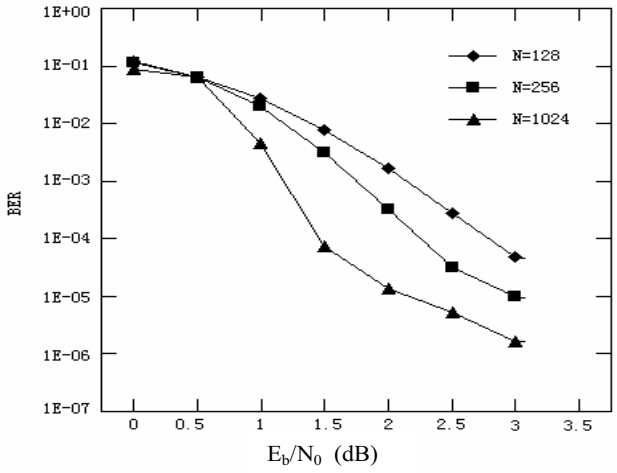

Fig. 1 Simulation results of BER vs SNR with different interleaving length for turbo codes consisted of component codes (23, 35, 1/2)

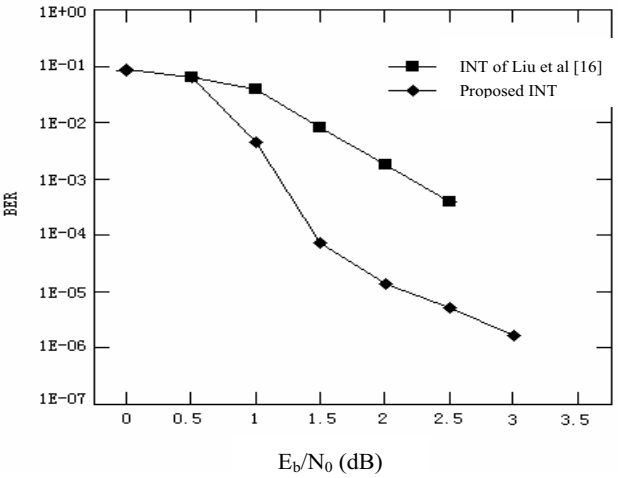

Fig. 2 Performance comparison between different interleaving schemes with N=1024

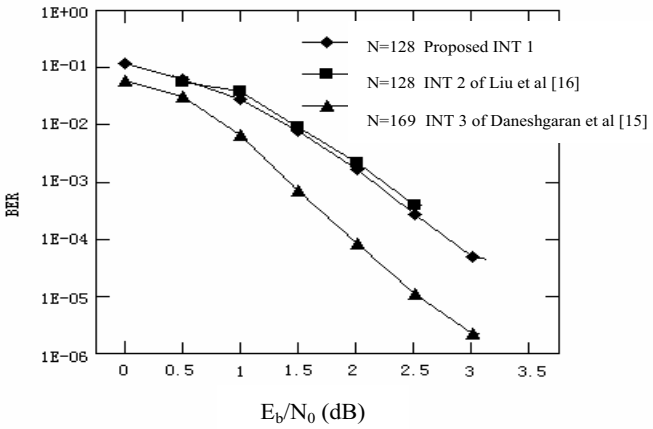

Fig. 3 Performance comparison between different interleaving schemes with short blocks

we provided the proof for the first time. Our simulations showed that the BER performance is similar to that of other authors [15, 16], but the design of our scheme is simpler. On the other hand, although the interleaver proposed here is a random one, the permuted results are deterministic, when the parameter h is set to a fixed integer. This characteristic is especially useful for reducing the hardware requirments of the interleaving and deinterleaving operations.

V. Acknowledgement

The author gratefully acknowledges the support from the Royal Society KC Wong Felowship of the UK. Heartfelt thanks go to Professor Lajos Hanzo for his careful review, modifications and helpful comments on improving the paper. The author also would like to express his gratitude to Professor Guangzhao Zhang for his helpful comments.

References

[1] C. Berrou, A. Glavieux, and P. Thitimasjshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proceedings, IEEE International Conference on Communications*, pp. 1064–1070, May 1993.

[2] O. Y. Takeshita and D. Costello, "New classes of algebraic interleavers for turbo-codes," in *Proceedings, 1998 IEEE International Symposium on Information Theory*, p. 419, 1998.

[3] O. Takeshita and D. Costello, "New deterministic interleaver designs for turbo codes," *IEEE Transactions on Information Theory*, vol. 46, pp. 1988 –2006, Sept. 2000.

[4] H. Herzberg, "Multilevel turbo coding with short interleavers," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 303 –309, Feb. 1998.

[5] H. R. Sadjadpour, N. J. A. Sloane, M. Salehi, and G. Nebe, "Interleaver design for turbo codes," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 831 –837, 2001.

[6] J. Yuan, B. Vucetic, and W. Feng, "Combined turbo codes and interleaver design," *IEEE Transactions on Communications*, vol. 47, pp. 484 –487, April 1999.

[7] J. Hokfelt and T. Maseng, "Methodical interleaver design for turbo codes," in *Proceedings, International Symposium on Turbo Codes and Related Topics, (Brest, France)*, pp. 212–215, Sept. 1997.

[8] J. D. Andersen, "Selection of code and interleaver for turbo coding," in *Proc. First ESA Workshop on Tracking, Telemetry and Command Systems, ESTEC, The Netherlands*, June 1998.

[9] S. A. Barbulescu and S. S. Pietrobon, "Interleaver design for three dimensional turbo codes," in *IEEE International Symposium Information Theory, Whistler, British Columbia, Canada*, p. 37, Sep. 1995.

[10] S. Maric, "Class of algebraically constructed permutations for use in pseudorandom interleavers," *Electronics Letters*, vol. 30, pp. 1378–1379, Aug. 1994.

[11] L. Perez, J. Seghers, and D. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Transactions on Information Theory*, vol. 42, pp. 1698 – 1709, Nov. 1996.

[12] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Transactions on Communications*, vol. 44, pp. 591 –600, May 1996.

[13] D. Divsalar and R. J. McEliece, "Effective free distance of turbo codes," *Electronics Letters*, vol. 32, no. 5, p. 445, 1996.

[14] A. S. Barbulescu and S. S. Pietrobon, "Terminating the trellis of turbo-codes in the same state," *Electronics Letters*, vol. 31, no. 1, pp. 22 –23, 1995.

[15] F. Daneshgaran and M. Mondin, "Design of interleavers for turbo codes: Iterative interleaver growth algorithms of polynomial complexity," *IEEE Transactions on Information Theory*, vol. 45, pp. 1845–1859, Sept. 1999.

[16] D. Liu, C. Tang, and Q. Zhang, "Improvement on the design of pseudo-random interleavers(in Chinese)," *Radio Engineering*, vol. 29, no. 6, pp. 52–55, 1999.