University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Faculty Publications in Computer & Electronics Engineering (to 2015)

Electrical & Computer Engineering, Department of

2004

Modified Convolutional Interleavers and Their Performance in Turbo Codes

Sina Vafi
University of Wollongong

Tadeusz Wysocki
University of Nebraska-Lincoln, wysocki@uow.edu.au

Follow this and additional works at: http://digitalcommons.unl.edu/computerelectronicfacpub

 Part of the Computer Engineering Commons

Vafi, Sina and Wysocki, Tadeusz, "Modified Convolutional Interleavers and Their Performance in Turbo Codes" (2004). *Faculty Publications in Computer & Electronics Engineering (to 2015)*. 20.
http://digitalcommons.unl.edu/computerelectronicfacpub/20

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications in Computer & Electronics Engineering (to 2015) by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Published in Joint IST Workshop on the Mobile Future and Symposium on Trends in Communications, 2004. SympoTIC '04. 24-26 Oct. 2004; pages 54-57

Modified Convolutional Interleavers and Their Performance in Turbo Codes

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

Abstract

In previous work,we have presented application of a model of the convolutional interleaver in turbo codes acting as a block interleaver through inserting a number of stuff bits equal to the number of interleaver memories at the end of each data block.In order to get better turbo codes performance, the interleaver with larger period, which increases the number of stuff bits and reduces effective channel bandwidth usage, has been suggested. In this paper, we introduce a modification to this interleaver improving turbo codes performance without increasing the interleaver period. This is carried out by increasing distance of adjacent bits that are positioned in original input bit stream in the interleaving procedure. Application of the modified interleaver in different turbo codes structures have been verified and results have been compared with those for the previously suggested interleaver.

1 Introduction

Convolutional interleaver is known as one of non-block interleavers that due to existence of synchronization with deinterleaver and employment of less delay in its structure has been preferred to block interleavers in some applications [1]. In [2]and [3],the performance of turbo codes with convolutional interleavers using continuous decoding methods have been verified. In comparison with usual iterative turbo decoding method, continuous methods produce better performance with increase in complexity, which is directly related to the interleaver length and its strucutre [4]. In addition, results show that continuous decoding is more reliable in turbo codes with higher number of states, while in codes with lower number of states it is not better than iterative decoding methods [5]. We have presented application of convolutional interleaver operating as a block interleaver through the inserting zero stuff bits at the end of each block to return the interleaver memories to the zero state and allowing the use of iterative turbo decoding methods [6]. The obtained simulation results indicate that application of convolutional interleavers in turbo codes with an acceptable stuff bits number relative to the overall bits number leads to better performance than when block interleavers are used. In addition,comparison of the two presented different convolutional interleaver structures shows that the optimised convolutional interleaver, where the zero stuff bits at the end parts of the interleaved data are deleted, has performance close to the non-zero bit deletion convolutional interleaver in error floor region of turbo codes[7]. However, with the optimised interleaver the obtained free distance remains relatively low in value, which degrades performance of turbo codes. As one solution, increasing of the interleaver period has been suggested which, on the other hand increases percentage usage of stuff bits. Hence it is necessary to minimize the number of stuff bits to be utilized in improvement. As pointed out in [8],depending on the applied interleaver characteristics, the minimum distance between two adjacent bits after interleaving is equal to the interleaver period, except at the end parts of the interleaver where due to the deletion of stuff bits this distance is decreased. Therefore, improvement can be achieved by increasing the distance between two adjacent bits particularly those bits that have been located at the end parts of the interleaver. In this paper, we propose such a modification to the interleaver structure that, without increasing the interleaver period and consequently the stuff bit number, interleaver with better performance is obtained.Simulated results of the applied modification have been compared with the previously obtained results for zero bit deletion convolutional interleaver, which confirms improvement in turbo codes performance with the new interleaver, especially in the error floor region.In the next section,we explain the proposed modification to the interleaver. The simulation results confirming the performance improvements achieved through the proposed modification are presented in section 3, and section 4 concludes the paper.

2 Convolutional Interleaver Structure

A convolutional interleaver consists of T parallel lines of delay elements. In general, each successive line has M more delay elements than the previous line. Therefore at the given time, some input data will remain in the interleaver memories and appear at the interleaver output later. In order to make an interleaved block of data, it is necessary to return the memory states of the interleaver to the zero state, which is carried out by insertion of zero stuff bits at the end of each block. Figure 1-a shows interleaved data block with input length L=64 and the interleaver (T=8,M=1) in 15 rows and 8

0-7803-8556-X/04/$20.00 © 2004 IEEE

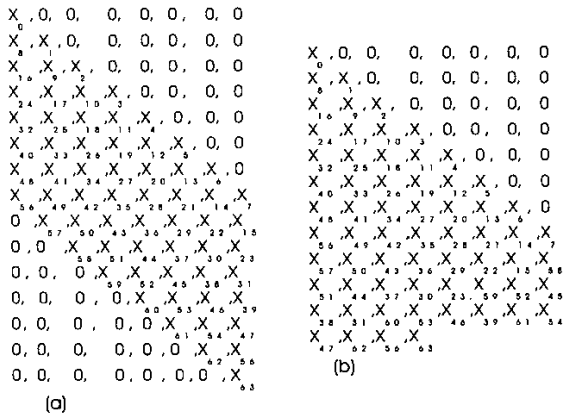

Figure 1: Convolutional interleaved data stream a) non-zero bit deletion b)zero bit deletion.

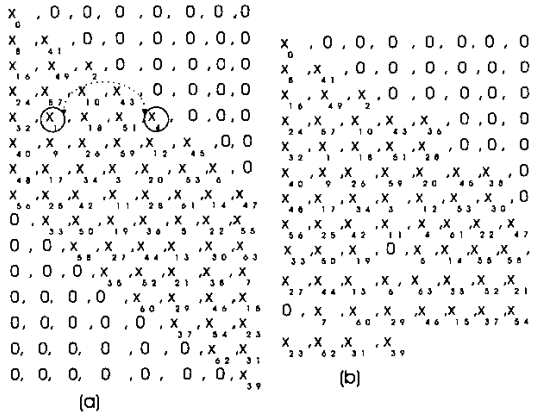

Figure 2: Modified Convolutional interleaved data a) just even column shifting b)even and odd column shifting and zero bit deletion.

columns. Optimised interleaver is obtained by deletion of stuff bits that has been located at the end parts of interleaved data to reduce the number of stuff bits.This has been illustrated in Figure 1-b [6]. In [7], an application of this interleaver in turbo codes has been verified and shows that with less stuff bits it has performance close to non-zero bit deletion interleaver, particularly in the error floor region due to lower multiplicities of the free distance and other low weights values. However,as it has been shown in Figure 1-b, deletion of stuff bits reduces distance between adjacent bits and causes codeword with lower weight being generated, which degrades turbo codes performance. For example in Figure 1-b, after optimisation, the distance between 62 and 63 bits,i.e.X_{62} and X_{63}, is decreased from 8 to 2.

In order to remove this drawback without increasing the interleaver period, which increases the number of stuff bits, we propose replacement of some bits located at the end parts of the interleaver with other bits positioned in higher interleaver parts such that new bits located at the end parts of the interleaver have sufficient distance with the bits adjacent to them before interleaving. This procedure should also prohibit generation of low weight patterns that return the second Recursive Systematic Convolutional (RSC) encoder to the zero state. We apply a modification to the non-zero bit deletion of convolutional interleaver where bits are distributed regularly in the interleaver columns. Due to the existence of different delay elements in each line, distribution of data time in each column of the interleaver output is different. Therefore, suitable shift of bits located in one interleaver column increasing existence distance between adjacent bits, which have been located in different columns is performed. Of course, if similar shift was done for all the columns, the distance would not change. Thus, different shift patterns should be used for different columns. For more simplicity, we consider distinct shifts only for odd and even columns. Finally, zero stuff bits located at the end parts of the interleaver are deleted to optimise conducted modification to the interleaver.

For even columns, each bit is shifted (2M+1)*T units.Those bits, which position after shift exceeds position of the last valid bit, are transferred to the valid bit position on the top of each column. For each column, the number of shifted bits is assumed to be even. If the overall number is odd, the first stuff bit data before the first data bit in each column is also shifted to maintain the even bit number. The even number is selected in order to achieve the acceptable distance between adjacent bits.

However,due to the even columns bits shift, it is possible that the new interleaved data block is characterized by lower weights than the former interleaver and can generate more low weight patterns that return the second RSC encoder to the zero state. In the considered example, as shown in Figure 2-a, for turbo codes (1,7/5)with input data weight 2, if bit 1 and bit 4 in the fifth row of the interleaver have value of 1 the second RSC encoder will be returned to zero state with weight 4. In addition to the other column 2 bits, similar condition can be observed between bits of other even columns and the corresponding odd columns bits. Hence it is necessary to shift these odd column bits in a way compatible with the applied RSC structure to omit low weight patterns from the interleaver and increase weight of turbo codes.We have found that reverse sorting of odd column bits,except column 1 and 3, can provide sufficient distance for RSC encoders with different states. Similarly as with even columns, the number of shifted bits is considered to be even. Figure 2-b shows the interleaved data block in the presented example after modification and deletion of zero stuff bits.In comparison with the previous example,the distance between bits 62 and 63 has increased to 12. As it has been shown in Figure 2-b, in one row of the interleaver bits distance of column 1 and 6 before and after interleaving has not changed. This is due to the low input data length,L=64, compared with the interleaver period,T=8, in this example. In practical designs, the

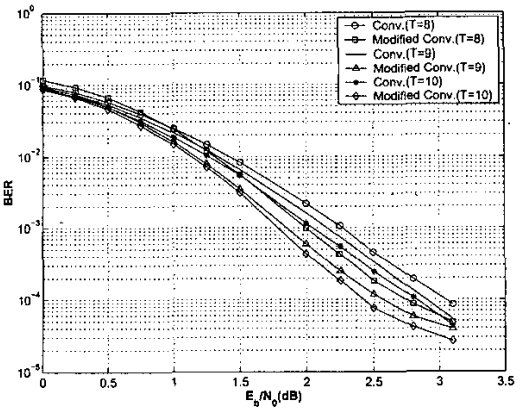

Figure 3: Performance of the 4 state full rate turbo code with interleaver periods T=8 to T=10 and length L=169.

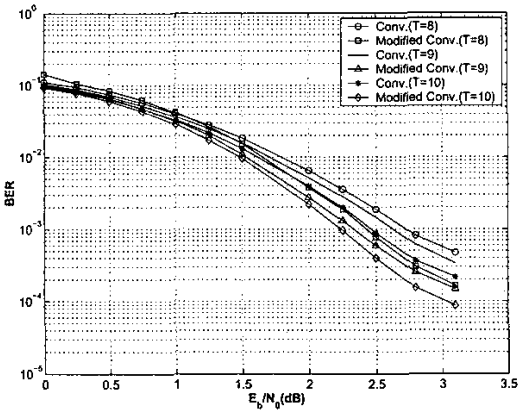

Figure 4: Performance of the 4 state half rate turbo code with interleaver periods T=8 to T=10 and length L=169.

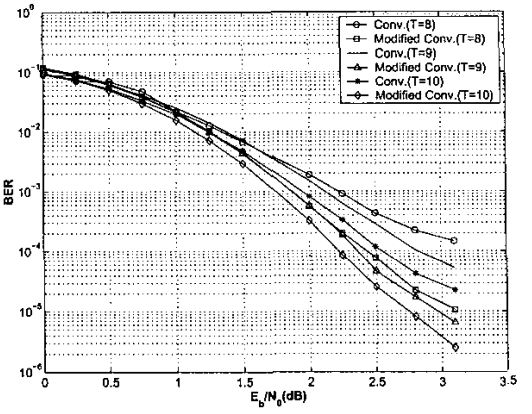

Figure 5: Performance of the 16 state full rate turbo code with interleaver periods T=8 to T=10 and length L=169.

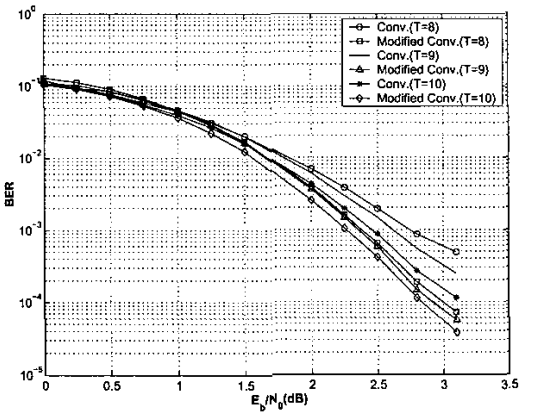

Figure 6: Performance of the 16 state half rate turbo code with interleaver periods T=8 to T=10 and length L=169.

interleaver period should be properly selected relative to the interleaver length in order to generate acceptable number of stuff bits. In the case of longer interleaver lengths, when applying the presented modification, the bits distance between column 1 and 6 before and after interleaving will differ because column 1 bits would remain constant while column 6 bits resorted from 5th or 6th row of the interleaver, depending on the number of bits in column 6, such that the new column 6 bits positions have reasonable distance with bits position of column 1 in similar rows.Conducted simulations verify performance improvement after application of the new interleaver in turbo codes.

3 Simulation Results

In the simulations, 4 and 16 state turbo codes with specifications of (1,7/5,m=2)(1,35/23,m=4) have been considered, where m represents number of memories for the RSC encoders. Among techniques providing turbo encoded data block, trellis termination and truncation have been selected for the first and second RSC encoders, respectively. Zero stuff bits are inserted to the interleaver after trellis termination and since they do not have any effect on the systematic and the first parity data, they will be removed from the mentioned data parts to reduce overall stuff bits number equal to $\frac{T(T-1)}{2}$ value. In the decoder, the iterative decoding method using Soft Output Viterbi Algorithm (SOVA) is used [9]. 8 iterations and Additive White Gaussian Noise (AWGN) have been considered in each simulation. In the conducted simulations, performance of zero bit deletion and applied modification properties on the convolutional interleavers have been compared. For this purpose, input data stream with weight no larger than 3 have been considered.

Figures 3 and 4 show results obtained from the full and half rate turbo codes with input data length L=169

and different interleaver periods, respectively. In both interleavers, with increasing of the period performance of the interleaver has improved. The results confirm that application of new convolutional interleaver with period T=8 creates better performance than convolutional interleaver with zero bit deletion and period T=10 characteristics. It means that the new interleaver has a number of stuff bits reduced from 45 to 28, which is equal to 45 percent.

Obtained simulation results in Figures 6 and 7 related to 16 state turbo codes with length L=169 confirm the above results. Again, the new interleaver with lower period,i.e. T=8, has similar performance as the previous convolutional interleaver with higher period (T=10). The interleaver with shorter period simplifies synchronization between applied interleaver and deinterleaver, of course.

Considering two interleavers with similar period, new interleaver with T=10 improves turbo codes(1,7/5) performance in error floor region by 0.4 dB for full and half rates. For the half and full rate turbo codes (1,35/23) the improvement is equal to 0.3dB and over 0.5 dB, respectively.More verifications have been presented in Figure 7 for convolutional interleaver periods T=15, T=20 and input data length L=1024. The results are similar to the former part.With increasing of the interleaver period, performance of both turbo codes structures has improved by 0.3 dB. In addition, the obtained results from modified convolutional interleaver with T=15 indicate better reliability than zero bit deletion convolutional interleaver T=20 for both turbo codes, which again approximately reduces stuff bits number by 45 percent.Regarding conducted simulations with different interleaver periods,we can conclude that associated modification will increase distance of adjacent bits to generate better randomization and improve turbo codes performance. These results have been obtained for convolutional interleavers with M=1.Since employment of an interleaver with higher M will increase adjacent bits distance in comparison with the distance for M=1, we expect that applying such interleaver with equal stuff bit number in its structure could create more improvement to turbo codes performance. As a result interleavers with shorter periods and better quality can be produced. The application of these interleavers in turbo codes and finding a suitable modifications similar to the algorithm presented in this paper will be followed in future works.

4 Conclusions

In this paper, we have presented efficient and simple modification to convolutional interleavers by increasing the distance of adjacent bits,resulting in possibilities of applying interleavers with shorter period and lower number of stuff bits. The new interleaver has been designed utilizing properties of the RSC encoder to prohibit generation of patterns that create low weight codewords. The simulation results confirm that the new interleaver has better performance than the previously presented interleavers. Further research will be conducted for higher values of M.

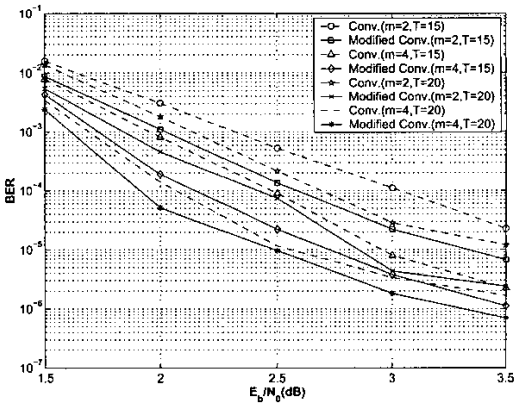

Figure 7: Performance of the 4 and 16 state half rate turbo code with interleaver periods T=15,T=20 and length L=1024.

References

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

[2] E.K.Hall and G.Wilson, "Stream-oriented turbo codes," *IEEE Trans.Inform.Theory.*, vol. 47, no. 5, pp. 1813–1831, July. 2001.

[3] B.Bai, X.Ma, and X.Wang, "Novel algorithm for continuous decoding of turbo codes," *IEE proc. Commun.*, vol. 146, no. 5, pp. 271–274, Oct. 1999.

[4] M. Breiling and L. Hanzo, "The super-trellis strucure of turbo codes," *IEEE. Trans. on. Inform. theory*, pp. 2212–2228, Sep. 2000.

[5] S. Benedetto and G. Montorsi, "Performance of continuous and blockwise decoded turbo codes," *IEEE. Communications Letters*, pp. 77–79, May 1997.

[6] 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.

[7] S. Vafi and T. Wysocki, "Iterative turbo decoder design with convolutional interleavers," *4th international symposium on CSNDSP,Newcastle,UK.*, pp. 124–127, July. 2004.

[8] S.Vafi and T.Wysocki, "Computation of the free distance and low weight distribution of turbo codes with convolutional interleavers," *accepted for 15th PIMRC*, 2004.

[9] J. Hagenauer and Hoher, "A viterbi algorithm with soft-decision outputs and its applications," *Globecom*, pp. 47.1.1–47.1.7, Nov. 1989.