

University of Nebraska - Lincoln

## DigitalCommons@University of Nebraska - Lincoln

---

Faculty Publications in Computer & Electronics Engineering (to 2015)    Electrical & Computer Engineering, Department of Engineering (to 2015)    of

---

2005

### Performance of convolutional interleavers with different spacing parameters in turbo codes

Sina Vafi

*University of Wollongong*

Tadeusz Wysocki

*University of Nebraska-Lincoln, [wysocki@uow.edu.au](mailto:wysocki@uow.edu.au)*

Follow this and additional works at: <https://digitalcommons.unl.edu/computerelectronicfacpub>



Part of the [Computer Engineering Commons](#)

---

Vafi, Sina and Wysocki, Tadeusz, "Performance of convolutional interleavers with different spacing parameters in turbo codes" (2005). *Faculty Publications in Computer & Electronics Engineering (to 2015)*. 23.

<https://digitalcommons.unl.edu/computerelectronicfacpub/23>

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.

# Performance of convolutional interleavers with different spacing parameters in turbo codes

Sina Vafi and Tadeusz Wysocki

**Abstract**—This paper considers application of a convolutional interleaver and the issue of influence of the space parameter in the turbo code performance. Similarly to previously considered convolutional interleavers, the new interleavers are designed as block interleavers and their performance in different code structures is compared with the interleaver having higher periods and space value of 1. In each comparison, the number of inserted stuff bits at the end of each data block is considered to be of similar order. Finally, suitable modification to the new interleavers is proposed improving performance for the codes with lower number of stuff bits.

**Index Terms**—Convolutional interleavers, turbo codes, period and space parameters.

## I. INTRODUCTION

Optimized convolutional interleavers can be introduced as a class of efficient block interleavers family. Their block-wise operation is forced by inserting the number of zero stuff bits equal to the number of the interleaver memories thus returning its memories to zero state at the end of each data block [1]. Conducted analysis on different turbo codes with this interleaver shows that the codes have relatively low free distance value with low multiplicities [2]. When interleaver with higher periods is applied, the distance between two adjacent bits at the original bit stream increases and consequently, higher free distance value for the code is expected. This should improve its performance in the error floor region.

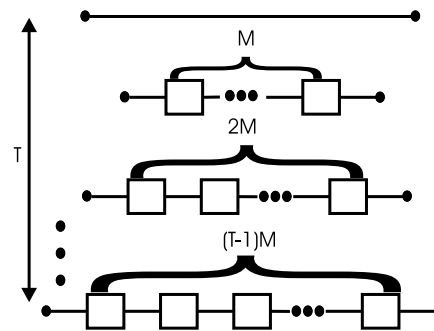
On the other hand, applying higher interleaver periods involves insertion of more stuff bits, which degrades the code efficiency. Hence, in [3], a modification to this interleaver has been presented, which improves the code reliability with a lower period and less stuff bits.

In this paper, we consider another interleaver characteristic that affects the turbo codes performance i.e., space parameter, which specifies the difference between the numbers of the interleaver memories in its two consecutive lines. We designed different optimized convolutional interleavers with higher values of space and similar number of stuff bits to the interleavers with periods and space value 1. Then, based on the utilized turbo code, we modify them to improve the code performance with lower number of stuff bits. The organization of the paper is as follows: Section 2 introduces structure of optimized convolutional interleavers with higher value of space. In Section 3, the analysis of different turbo codes with these

interleavers is performed and the obtained results are compared with conducted simulations. Finally, Section 4 concludes the paper.

## II. CONVOLUTIONAL INTERLEAVER STRUCTURE WITH HIGH SPACE VALUE

Period and space are known as two major convolutional interleaver parameters [4]. The input data stream is distributed into  $T$  parallel lines of the interleavers where  $T$  represents the interleaver period. Depending on the space value  $M$ , each interleaver line has  $M$  more delay elements than the previous line. Hence, the interleaved data appear in different time slots at the interleaver output. Fig. 1 shows the general form of the convolutional interleaver with period  $T$  and space  $M$ .



1: General structure of convolutional interleaver with period of  $T$  and Space  $M$ .

It has been shown that increasing the period of the optimized interleaver with the space value of 1 can improve the turbo code performance in the error floor region [2]. However, for short to medium interleaver lengths the interleaver with higher period may be needed, which degrades the code performance in terms of increasing the number of stuff bits relative to the number of input data.

This minimum distance between two adjacent bits without considering the effect of zero bit deletion at the end part of the interleaver, is generally governed by the product of  $T$  and  $M$ . Therefore, increasing the interleaver space value instead of its period can be considered as another way to provide a sufficient minimum distance number between the adjacent bits.

In order to evaluate performance of such interleavers, we construct them in a way giving similar numbers of stuff bits to these interleavers with space value of 1.

The authors are with the University of Wollongong, Northfields Ave., NSW, 2522, Australia. email: {sv39, wysocki}@uow.edu.au.

**I:** Generated Minimum distance values between adjacent bits of input bit stream from different interleavers.

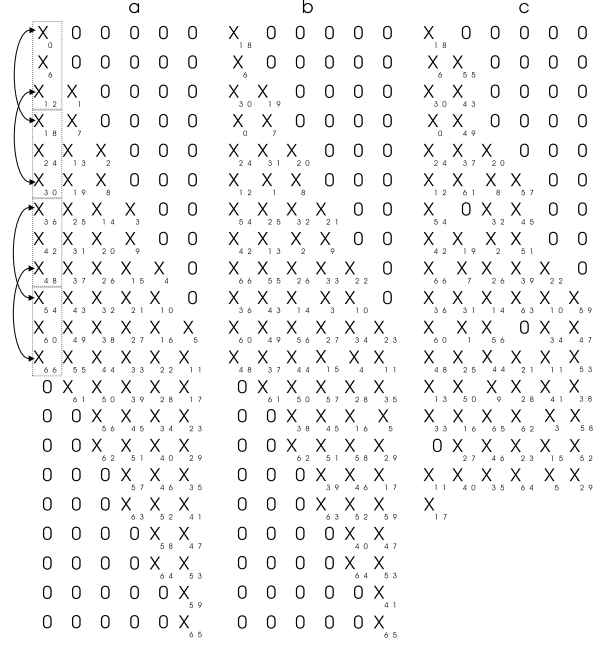
Space (M)	Period (T)	Minimum distance value	No. of stuff bits $T(T-1)M/2$
2	6	12	30
3	5	15	30
1	8	8	28
1	19	19	171
1	20	20	190
2	14	28	182
3	11	33	165
4	10	40	180
5	9	45	180
6	8	48	168
7	7	49	147

Therefore, it is necessary to design the new interleavers with lower periods. Based on arithmetic sequence, the overall number of stuff bits for the illustrated convolutional interleaver in Fig. 1 is given by:

$$S = \sum_{i=1}^T s_i = M + 2M + \dots + (T-1)M = \frac{T(T-1)M}{2} \quad (1)$$

where  $s_i$  represents number of applied memories in the  $i$ th interleaver line.

Table 1 shows the minimum distance for different interleavers with similar number of stuff bits. The Table indicates that interleavers with higher space exhibit greater minimum distance values than interleavers with space value of 1. Although the distance between two adjacent bits increases, due to the shorter periods, the distance between distributed bits in each interleaver line decreases. This distance can be specified by the value of  $T - 1$ . As a known method, in order to verify performance of these interleavers in turbo codes, weight distribution of the code with input data weight of 2 must be calculated [5]–[7]. When both bit 1 positions in the interleaved data with weight 2 are located in one line of the interleaver, and the generated interleaved data return the second Recursive Systematic Convolutional (RSC) encoder to the zero state, the low codeword weight for this encoder is achieved. Simultaneously, similar condition occurs for the first RSC encoder because the distance of distributed bits in one of the interleaver line is identical to their distance in the input data stream. Because of this and the low value of the interleaver period, the number of resulting low weight encoded data increases, leading to higher multiplicities of patterns with weight close to the free distance value. As a result, the performance of such codes dramatically degrades. For example, as shown in Fig. 2-a, in the interleaver ( $M = 2$ ,  $T = 6$ ), the distance between distributed bits in each interleaved data line is equal to 5. When this interleaver is used for turbo codes(1,7/5) the existence of pattern  $(00 \dots 00100 \dots 0100 \dots 00)_L$  with length  $L$  including  $n = 3k + 2$  zeros( $k = 0, 1, \dots$ ) between two bit 1s will return both RSC encoders to zero state and generate low weight codewords. Depending on the interleaver length, the number of patterns that provide similar conditions



**2:** Conducted modification on the interleaver ( $T = 6$ ,  $M = 2$ ) a) original bit stream b)increasing column bits distance procedure c)even column bits shifts equal to  $5T$  and zero bit deletion from the end part of the interleaver.

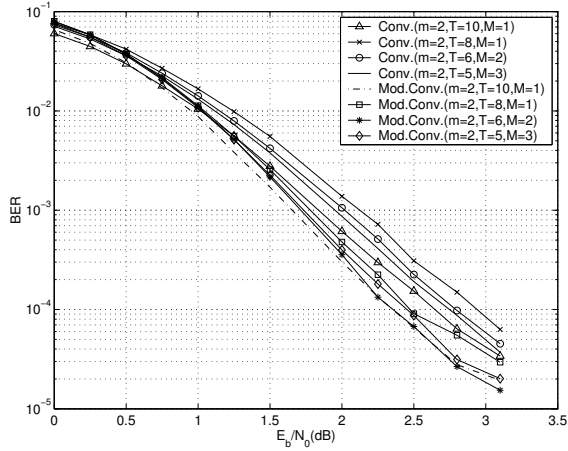
can increase. Hence, it is necessary to perform suitable modification to this interleaver to increase the distance between two adjacent bits that have been located in one line of the interleaver. The modification is as follows:

Three consecutive bits in one column are considered as one group. The first and third bit of each group are replaced with the adjacent bits of the next group to increase minimum distance of the interleaved bits in one column to be twice of the original distance. Then, even column bits shifting similar to the method presented in [3] is performed to provide sufficient distance between adjacent bits that have been located in different columns. Finally, optimization is conducted by deleting stuff bits located at the end part of the interleaver. Fig. 2-b and 2-c show modification procedures for the prepared interleaver.

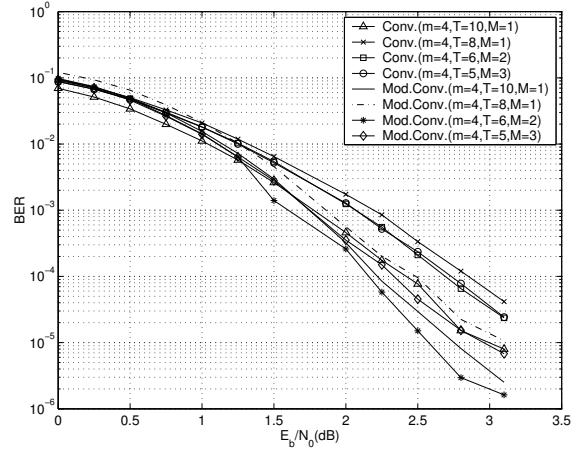
The applied modification will not remove patterns that return the second RSC encoder to the zero state. In fact, this method generates patterns that return relevant encoder to the zero state with higher codeword weights. One can consider further optimizing the number of bits and bits replacement procedure to increase the distance between the adjacent data bits after interleaving.

### III. SIMULATION RESULTS

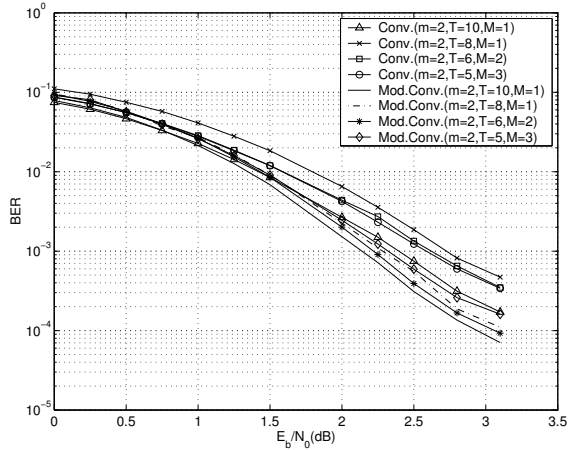
In the simulations, we have used 4- and 16- state turbo codes ( $m = 2, 1, 7/5$ ) and ( $m = 4, 1, 35/23$ ), where  $m$  represents number of RSC encoder memories that have been used. Among techniques providing turbo encoded data block, trellis termination and truncation



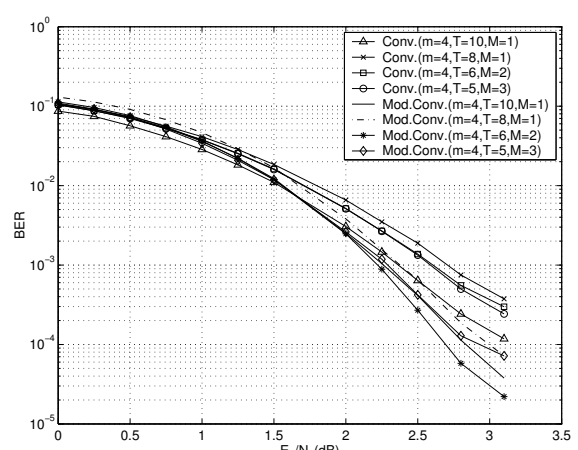
3: Simulation results for 4 state full rate turbo codes with interleavers length  $L = 169$ .



5: Simulation results for 16 state full rate turbo codes with interleavers length  $L = 169$ .



4: Simulation results for 4 state half rate turbo codes with interleavers length  $L = 169$ .



6: Simulation results for 16 state half rate turbo codes with interleavers length  $L = 169$ .

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 stuff bits number equal to  $\frac{T(T-1)M}{2}$  value. The optimized convolutional interleaver structure has been examined for different values of the space parameter. Performance of the resulting turbo encoders have been assessed in the presence of Additive White Gaussian Noise (AWGN). The implemented decoder has realized 8 iterations of the Soft Output Viterbi Algorithm (SOVA) algorithm [8]. In all cases, we have considered input data with weights no greater than 4.

Fig. s 3 and 4 show simulated results of the 4- state full and half rate turbo code with the interleaver length  $L = 169$ . For the full rate turbo code, the interleavers ( $T = 5, M = 3$ ) and ( $T = 6, M = 2$ ) give 0.2 dB better performance than ( $T = 8, M = 1$ ) for all signal to noise ratios. In addition, they have very close performance to

the interleaver ( $T = 10, M = 1$ ) while the number of stuff bits has been reduced by 35%. For the half rate turbo codes these interleavers will slightly improve the code performance in comparison with the interleaver ( $T = 8, M = 1$ ).

Fig. s 5 and 6 show results of the conducted simulations for the full and half rate 16- state turbo code, respectively. Similarly as for the half rate 4- state turbo code, interleavers with higher space parameters have exhibited slightly better performance than the interleaver ( $T = 8, M = 1$ ).

Unlike when the modification is equipped to the interleavers with space value of 1 [3], for interleavers with higher values of space parameter there is no need for reverse bit sorting of odd columns during the modification process. Hence, finding an optimum shift value improving the code performance especially for long interleaver lengths or higher spaces is achieved through even column bits shifting and several simulations with

**II:** Shifting unit values for modified turbo codes with different interleavers.

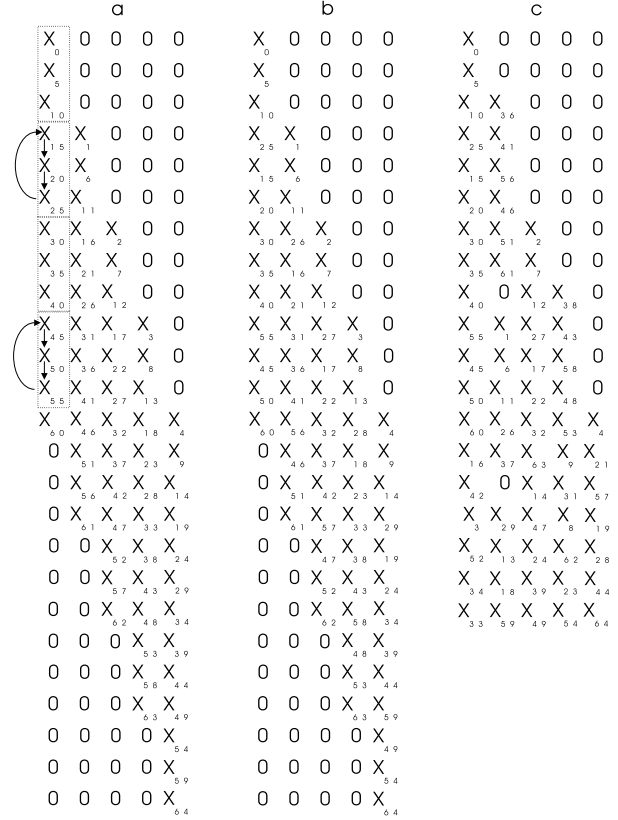
State Code	Interleaver specifications	Interleaver length	Shift Unit value
4	$T = 6, M = 2$	167	$13 \times T$
4	$T = 5, M = 3$	167	$12 \times T$
16	$T = 6, M = 2$	167	$13 \times T$
16	$T = 5, M = 3$	167	$10 \times T$
4	$T = 14, M = 2$	1024	$15 \times T$
4	$T = 11, M = 3$	1024	$7 \times T$
16	$T = 14, M = 2$	1024	$10 \times T$
16	$T = 11, M = 3$	1024	$16 \times T$

different shift unit values. Table 2 gives even columns bits shifting values of each utilized interleaver. For the interleaver ( $T = 6, M = 2$ ) the procedure mentioned in the previous section is conducted before shifting to sufficiently improve the code performance. The simulation results presented in Fig. 3 confirm similar performance of the interleavers with higher values of space parameter and that of the interleaver ( $T = 10, M = 1$ ) with lower number of stuff bits by 35%, while, in Fig. 4, the modified interleaver ( $T = 6, M = 2$ ) has closer performance to the interleaver ( $T = 10, M = 1$ ) than other modified interleavers with similar number of stuff bits.

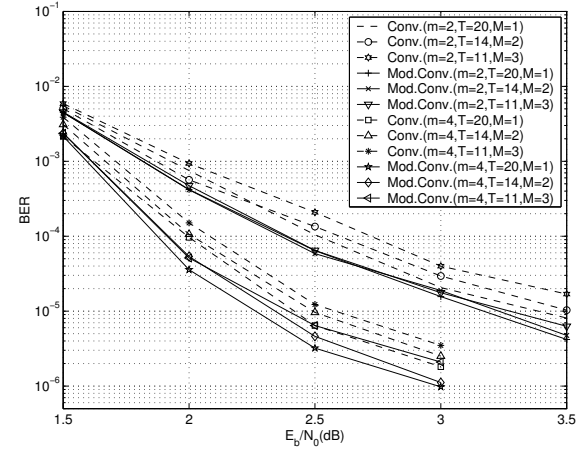
For the full and half rate 16- state turbo code, the modified interleaver ( $T = 6, M = 2$ ) has even better performance than the interleaver ( $T = 10, M = 1$ ) by 0.25 dB. By comparing modified interleaver results with unmodified interleavers results, it is easy to notice that the proposed modifications can improve the full and half rate code performance with the lower number of stuff bits. Since the interleaved data from the interleaver ( $T = 5, M = 3$ ) returns the second RSC encoder of the turbo code (1,35/23) to zero state, a special modification prior to the even column bit shifting is required. Similarly to the previously explained case for the interleaver ( $T = 6, M = 2$ ), each three consecutively located bits in one column are considered as one group. Then, each bit is replaced with its adjacent bit. Finally, even column bits shifting is applied, in order to complete the modification procedure. Fig. 7 illustrates the modification process.

More testing has been performed with higher interleaver lengths. Fig. 8 shows simulation results of the half rate 4 and 16 state turbo codes with the interleaver length  $L = 1024$ . In each case, the results show that interleavers ( $T = 14, M = 2$ ) and ( $T = 11, M = 3$ ) have close performance to the interleaver ( $T = 20, M = 1$ ), while the number of stuff bits have been reduced by 4 and 13%, respectively.

In the case of 4% reduction of number of stuff bits, the modified interleaver ( $T = 14, M = 2$ ) creates close and 0.2 dB better performance than the interleaver ( $T = 20, M = 1$ ) for the 4 and 16 state turbo code, respectively. In addition, modified interleaver ( $T = 11, M = 3$ ) gives similar performance to interleaver ( $T = 20, M = 1$ ) in the error floor region, while number of stuff bits has



**7:** Conducted modification on the interleaver ( $T = 5, M = 3$ ) a) original bit stream b) increasing column bits distance procedure c) even column bits shifts equal to  $6T$  and zero bit deletion from the end part of the interleaver.



**8:** Simulation results for 4 and 16 state half rate turbo codes with interleavers length  $L = 1024$ .

been decreased by 13%. However, the relevant graphs of 4 state code shows 0.2 dB worse performance for this modified interleaver in comparison with the interleaver ( $T = 20, M = 1$ ).

The obtained results confirm that selecting the sufficient interleaver with higher values of space parameter

and applying suitable modification compatible with the RSC encoder structure can create better performance for the utilized code. This is particularly visible for the high state turbo code with short interleaver lengths. Although for the higher modified interleaver lengths with relatively low number of stuff bits, similar performance to the interleaver with space value of 1 is achieved.

#### IV. CONCLUSIONS

In this paper, performance of turbo codes with the optimized convolutional interleaver having higher space values than 1 has been examined. The obtained simulation results for different codes show that applying the suitable structure for the new interleaver can improve the code performance to surpass performance of the previously suggested interleavers with space value 1. Some modifications related to these interleavers have been proposed to improve the code performance with reduced number of stuff bits.

#### REFERENCES

- [1] S. Vafi and T. Wysocki, "Iterative turbo decoder design with convolutional interleavers," *4th international symposium on CSNDSP, Newcastle, UK.*, pp. 124–127, July. 2004.
- [2] S.Vafi and T.Wysocki, "Computation of the free distance and low weight distribution of turbo codes with convolutional interleavers," *IEEE 15th PIMRC, CD ROM*, 2004.
- [3] —, "Modified convolutional interleavers and their performance in turbo codes," *accepted for the 2nd SympoTIC 04.*, 2004.
- [4] G.D.Forney, "Burst-correcting codes for the classic bursty channel," *IEEE Trans. Commun.*, vol. COM-19, pp. 772–781, Oct. 1971.
- [5] 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.
- [6] 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.
- [7] M.Z.Wang, A.Sheikh, and F.Qi, "Interleavers design for short turbo codes," *Globecom*, pp. 894–898, 1999.
- [8] J. Hagenauer and Hoher, "A viterbi algorithm with soft-decision outputs and its applications," *Globecom*, pp. 47.1.1–47.1.7, Nov. 1989.