Paper

Design of a Modified Interleaving Algorithm Based on Golden Section Theory Enhancing the Performance of Turbo Codes

Essedik Iftene[1,2], Adda Ali-Pacha[1], and Lahcène Hadj Abderrahmane[2]

[1] *Laboratoire de codage et de la sécurité de l'information, LACOSI, Département d'Electronique, Université des Sciences et de la Technologie d'Oran Mohamed-Boudiaf, USTO-MB, Oran, Algeria*
[2] *Algerian Space Agency (ASAL), Centre of Satellites Development, Oran, Algeria*

https://doi.org/10.26636/jtit.2022.158721

Abstract—This paper investigates the design of a modified matrix interleaving algorithm as a way to improve the performance of turbo codes. This proposed solution, known as the matrix-dithered golden (MDG) interleaver, utilizes the characteristics of a matrix interleaver combined with the golden section theory. The performance of the proposed interleaving method is compared with that of matrix (M), random (R), and dithered golden (DG) interleavers. The comparison is made in terms of bit error rate (BER), frame error rate (FER), computational complexity, and storage memory requirement. The turbo coded system is implemented and simulated using Matlab/Simulink software. Results of simulations performed both in the additive white Gaussian noise (AWGN) channel and the Rayleigh fading channel demonstrate the effectiveness of the proposed interleaver. The MDG interleaver is an effective replacement for random interleavers, as it improves BER and FER performance of the turbo code and is also capable of reducing the storage memory requirement without increasing the system's complexity.

Keywords*—*AWGN channel, golden section theory, interleaver, iterative decoding, Rayleigh fading channel, turbo code.

1. Introduction

Since their introduction in 1993 [1], turbo codes have received considerable attention and are currently the subject of extensive research [2], [3]. This is not only because of their powerful error correcting capability, but also because of their flexibility in terms of providing different block sizes and code rates [4]. A turbo code encoder consists of a parallel concatenation of two recursive systematic convolutional (RSC) encoders separated by an interleaver [5], [6]. The interleaver is a device that takes a given sequence of symbols at the input and produces identical symbols at the output, but in a different temporal order. The binary data sequence entering the turbo code's internal interleaver is denoted by d_N, where N is the length the data sequence. The binary data sequence at the output of the turbo code's internal interleaver is denoted by $d_{N-\Delta}$. The corresponding coded data is the binary output X_{3N}. A turbo code decoder employs two cascaded decoding blocks. An iterative

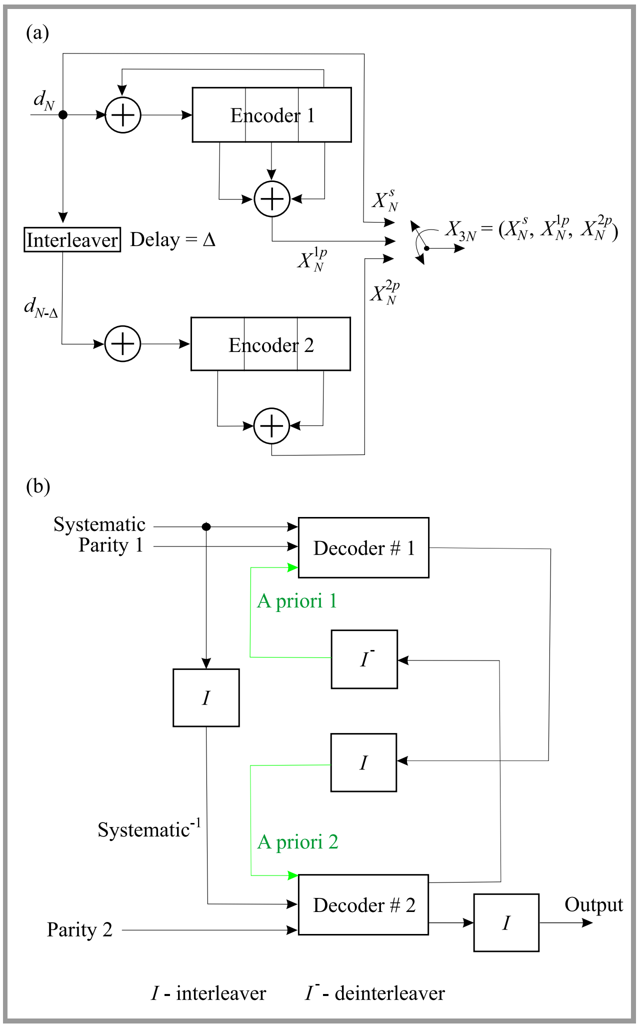

Fig. 1. (a) Turbo encoder structure, (b) turbo decoder structure.

scheme is used for decoding the turbo codes such that the overall performance can be improved [2], [3]. The structures of the turbo encoder and decoder are shown in Fig. 1.

It has been found that one way to improve the performance of a turbo coded system is to use a good interleaver structure [5], [7], [8]. Therefore, interleaver design is the subject of numerous research projects and a number of algorithms have been developed [9]–[20]. It is asserted for turbo codes that interleavers with some randomness tend to perform better than their fully structured counterparts, especially for large block sizes. However, a turbo coded system with a built-in random interleaver suffers from the problem of insufficient flexibility. A change in the length of interleaving requires another search of the interleaving pattern, which implies a more complex implementation. Furthermore, the generated interleavers, characterized by different lengths, should be stored separately in the memory [9], [10]. This causes a serious storage-related concern in a scenario in which many interleaving lengths need to be supported.

This paper introduces a modified architecture for a matrix interleaver, referred to as the matrix-dithered golden (MDG) interleaver which can resolve the problems of the existing random interleavers. The matrix-dithered golden interleaver aims to improve the BER and FER of turbo code, and to minimize the memory requirement by avoiding the need for generating and storing individual interleaving patterns for different interleaving lengths.

The rest of the paper is organized as follows. In Section 2, matrix, random and dithered golden interleavers are briefly explained. In Section 3, we present the designing method of the proposed interleaving algorithm. Simulation results and performance evaluation of these interleavers are provided and discussed in Section 4. The conclusion is presented in Section 5.

2. Interleavers

The interleaving process is a useful technique to enhance the error correcting capability of a turbo code [19]. Thanks to the interleaver, turbo codes can deal with burst errors by converting error patterns that contain long sequences of serial erroneous data into a more random error pattern, thus distributing errors among many code vectors [21]. Turbo codes work much better when errors in the received sequence are spread far apart [2]. An interleaver is used to randomize the error locations by taking a given sequence of symbols, and permutes their positions in a different temporal order [22], [23]. The inverse of the interleaving process is called deinterleaving and restores the interleaved sequence.

In general, we can classify interleavers into two broad categories [5], [7], [8]: random and deterministic interleavers. For deterministic interleavers, the position of every data bit is known according to an algorithm, while for random interleavers the position of each data bit is random. Some useful interleavers used in turbo code are discussed below.

The matrix interleaver [7], [11], [22] is one of the simplest types that is most commonly used in communication systems. This type of interleaver is easy to implement in practice and is characterized by a process in which data is permuted by being written row-wise and read column-wise. The matrix interleaver may have a good minimum distance, but the high multiplicity of low-weight code words makes this interleaver unsuitable [23], [24].

The interleaver with random properties is one of the essential building blocks of turbo codes [7], [18], [23]. Such an interleaver generates a random mapping between the input and output positions. Once the symbols are introduced into a random interleaver, the output symbols are chosen randomly, so that the same symbol that has already been selected is not repeated [21]. As the selection is random, it will be impossible to know the symbol positions at the interleaver output. Therefore, it would be necessary to maintain a correspondence table showing the dependence between the old and the new positions of the interleaved symbols, so that they can be deinterleaved [18], [24]. The random interleaver requires N indexes to be stored in order to implement an interleaver of a length of N. The fundamental concept of a random interleaver is simple, but its practical realization is more complex than that of a matrix interleaver [17], [23].

The golden section has applications in many mathematical problems [25]. It has been used for designing interleavers in turbo codes and is characterized by good proprieties [26]. Golden interleavers are based on sorting real-valued numbers derived from the golden section. Figure 2 illustrates the golden section principle.

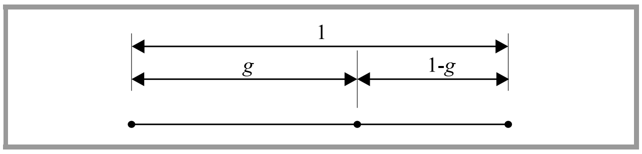

Fig. 2. Golden section principle.

For a given line segment of length 1, the problem is to divide it into a long segment of length g, and a shorter segment of length $1-g$, such that:

$$\frac{g}{1}=\frac{1-g}{g} .$$

Using this principle, the golden section value is calculated as $g\approx 0.618$ [8], [12], [22]. The first step in calculating the interleaver indexes is to compute the golden section value g. The second step is to compute the real increment value C, as:

$$C=\frac{N(g^{m}+j)}{r} , \tag{1}$$

where N is the length of the data sequence, m is a preselected non-zero positive integer preferably set to 1 or 2, r is a preselected non-zero integer defining a spacing between any pair of input elements that are to be maximally spread, and j is a preselected integer modulo r. In a typical implementation, j is set to 0, and r is set to 1 [8], [22], [26]. The third step is to generate a real-valued dithered golden vector v. The elements of v are calculated as:

$$v(n)=[s+n\times C+\mathbf{d}(n)] \mod N, \quad \text{for } n=1 \text{ to } N . \tag{2}$$

In Eq. (2), s is any real starting value and d is a dither vector. The starting value s is usually set to 0, but other real values can be selected. The dither vector is uniformly distributed between 0 and $N \times D$, where D is the normalized width of the dither distribution $\mathbf{d}(n)$ and is set to 0.01, according to [8], [14], [26]. The next step is to sort the dithered golden vector v and find the index vector $\mathbf{Z}$ that defines this sort, i.e. to find a sort vector $\mathbf{Z}$ such that $a(n) = v[\mathbf{Z}(n)]$ for $n = 1$ to N, where $\mathbf{a} = \mathrm{sort}(\mathbf{v})$. The dithered golden interleaver indexes are then given by $\alpha[\mathbf{Z}(n)] = n$, for $n = 1$ to N. In fact, vector $\mathbf{Z}$ is the inverse interleaver for α. The dithered golden interleaver requires the use of index memory for storing precomputed indexes. If the full indexes are stored, then the index memory can be excessive.

3. The Matrix-Dithered Golden Interleaving Algorithm

The process of designing a matrix-dithered golden (MDG) interleaver is performed according to the flowchart shown in Fig. 3. The interleaving design comprises four steps:

Step 1

Prepare the golden section model with its control parameters: D, s, m, r, and j. The values of the control parameters of the golden section model adopted in this paper are: $s = 0$, $m = 1$, $D = 0.1$, $j = 9$, and $r = 15$.

Using the defined golden section model, generate two dither vectors sequences $\mathbf{d}_{row}$ and $\mathbf{d}_{column}$ of real numbers with their length equal to the length of the largest frame N:

$$\mathbf{d}_{row}(n) = \{d_{r1}, d_{r2}, \ldots, d_{rN}\}, \quad \mathbf{d}_{row}(n) \in [1, N \times D],\ n \in [1, N]\ , \tag{3}$$

$$\mathbf{d}_{column}(n) = \{d_{c1}, d_{c2}, \ldots, d_{cN}\}, \quad \mathbf{d}_{column}(n) \in [1, N \times D],\ n \in [1, N]\ . \tag{4}$$

Step 2

Start with the conventional matrix interleaver M. Find an appropriate number of rows $\mathbf{N}_r$ and determine the number of columns N_c for a particular frame size N. The range of the input frame size is divided into two sub-blocks and each sub-block has a different row number and a different column number given by:

$$\begin{aligned} N_r &= \mathrm{floor}(\sqrt{N}), \quad \text{if} \quad N < 512\ , \\ N_r &= \mathrm{floor}(\sqrt[3]{N}), \quad \text{if} \quad N \geq 512\ , \\ N_c &= \mathrm{ceil}\frac{N}{N_r}\ . \end{aligned} \tag{5}$$

Write, in a row-wise fashion, left to right, and starting with the top row, the input data $\mathbf{D}_{in}$ into a matrix $\mathbf{M}$ with N_r rows and N_c columns.

$$\mathbf{M}(i,:) = \mathbf{D}_{in}\left[(i-1) \times N_c + 1 : i \times N_c\right]\ , \ \text{for } i = 1 \text{ to } N\ . \tag{6}$$

Write the vector $\boldsymbol{d_{row}}$ inside a matrix $\boldsymbol{M_{d_{row}}}$ having N_r rows and N_c columns to obtain N_r different $\boldsymbol{d_{row}}$ vectors, each with length N_c, and write the vector $\boldsymbol{d_{column}}$ inside a matrix $\boldsymbol{M_{d_{column}}}$, having N_r rows and N_c columns to obtain N_c different $\boldsymbol{d_{column}}$ vectors, each with length N_r.

Step 3

Using Eq. (7), generate the dithered golden matrix $\mathbf{v}_{row}$, and order each row according to its magnitude, to form the intra-row permutation matrix $\mathbf{Z}_{row}$. Indexes matrix $\mathbf{Z}_{row}$ and matrix $\mathbf{a}_{row}$, which is the sorted version of matrix $\mathbf{v}_{row}$, are related as Eq. (8):

$$\mathbf{v}_{row}(i,:) = \left[s + i \times C_{row} + \mathbf{M}_{\mathbf{d}_{row}}(i,:)\right] \bmod N_r, \ \text{for } i = 1 \text{ to } N_r\ , \tag{7}$$

$$\mathbf{a}_{row}(i,:) = \mathbf{v}_{row}\left[\mathbf{Z}_{row}(i,:)\right], \ \text{for } i = 1 \text{ to } N_r\ . \tag{8}$$

Perform the intra-row permutations of matrix $\mathbf{M}$, based on the constructed intra-row permutation pattern $\mathbf{Z}_{row}$.

$$\mathbf{M}_{row}(i,j) = \mathbf{M}\left[i, \mathbf{Z}_{row}(i,j)\right], \ \text{for } i = 1 \text{ to } N_r,\ j = 1 \text{ to } N_c\ . \tag{9}$$

Step 4

Similarly, to the step 3 and using Eq. (10), generate another dithered golden matrix $\boldsymbol{v_{column}}$, and order each column of this matrix according to their magnitude, to form the intra-column permutation matrix $\mathbf{Z_{column}}$. Indexes matrix $\mathbf{Z_{column}}$ and the matrix $\boldsymbol{\alpha_{column}}$, which is the sorted version of the matrix $\boldsymbol{v_{column}}$, are related as Eq. (11):

$$\boldsymbol{v_{column}}(:,j) = \left[s + j \times C_{column} + \boldsymbol{M_{d_{column}}}(:,j)\right] \bmod N_c, \quad \text{for } j = 1 \text{ to } N_c\ , \tag{10}$$

$$\boldsymbol{\alpha_{column}}(:,j) = \boldsymbol{v_{column}}\left[\mathbf{Z_{column}}(:,j)\right], \ \text{for } j = 1 \text{ to } N_C\ . \tag{11}$$

Perform the intra-column permutations of matrix $\mathbf{M}_{row}$ obtained in step 3, based on the constructed intra-column permutation pattern $\mathbf{Z}_{column}$:

$$\mathbf{M}_{column}(i,j) = \mathbf{M}_{row}\left[\mathbf{Z}_{column}(i,j), j\right], \quad \text{for } i = 1 \text{ to } N_r,\ j = 1 \text{ to } N_c\ . \tag{12}$$

Finally, the entire data block is read from the permuted matrix $\mathbf{M}_{column}$, column-wise, top to bottom, starting with the left column:

$$\mathbf{D}_{out}\left[(j-1) \times N_r + 1 : j \times N_r\right] = \mathbf{M}_{column}(:,j), \ \text{for } j = 1 \text{ to } N_c\ . \tag{13}$$

4. Comparative Performance Analysis of Interleavers

To verify the effectiveness of the proposed interleaving approach, comparisons to matrix (M), random (R) and dithered golden (DG) interleavers have been made based on such parameters as complexity, BER, FER, and memory usage. These interleavers were introduced into an unpunctured turbo code at the rate of 1/3, in which two identi-

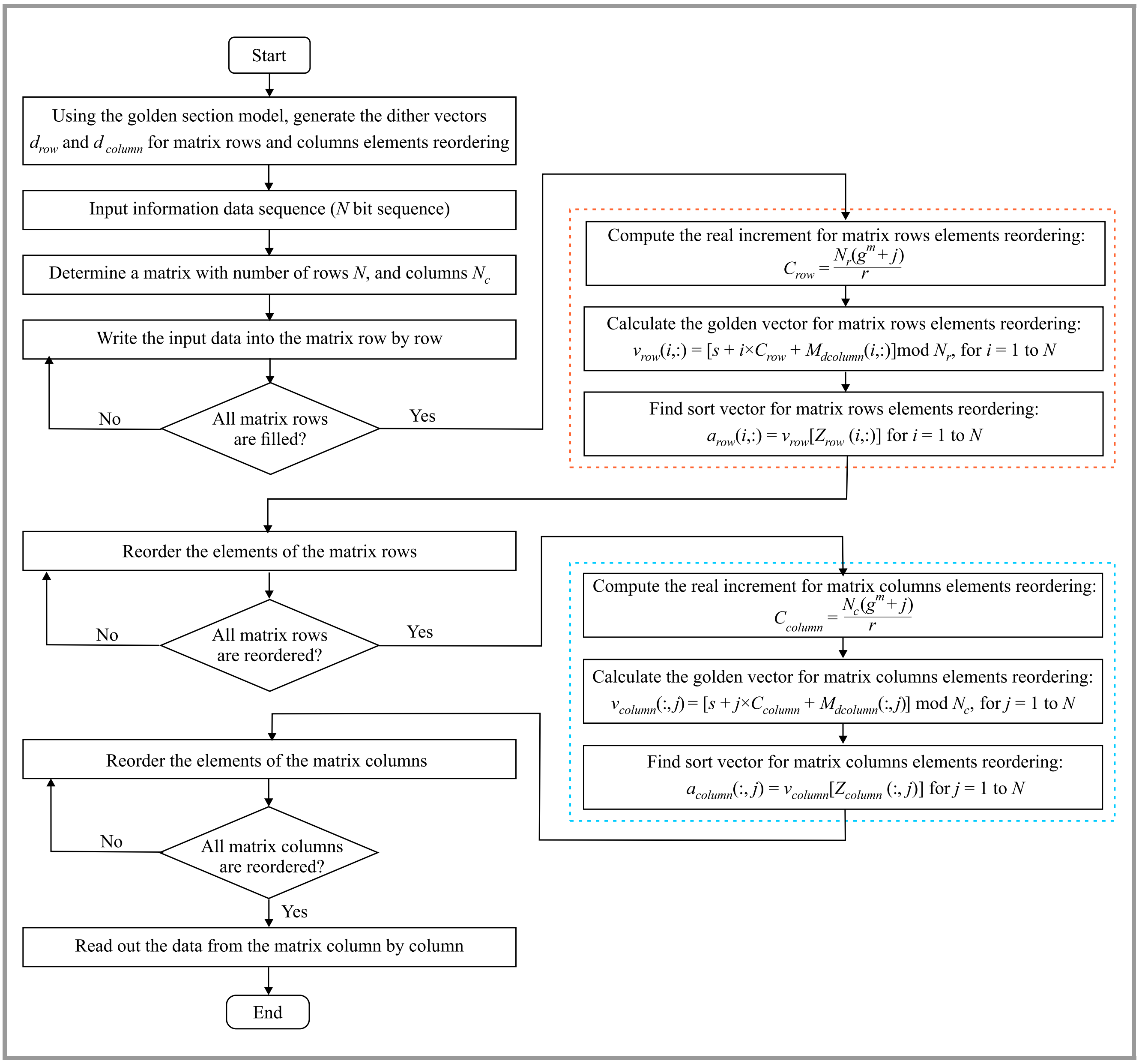

Fig. 3. Flowchart of the MDG interleaver algorithm.

cal recursive systematic convolutional encoders of generator polynomials $(7,5)_{oct}$, having the constraint length, $K = 3$, are connected in parallel [6], [7], [14], [26]. The turbo coded system is implemented and simulated using Matlab/Simulink software. All simulation results presented are based on iterative decoding using the maximum a posteriori (MAP) algorithm [2]. Turbo decoders suffer from high decoding latency due to the iterative decoding process. Latency can be lowered by reducing the number of required decoding iterations. Hence, the number of iterations in the decoder is selected to equal 7. Two noise models were considered: AWGN channel and Rayleigh fading channel. The data length is taken as 400 and 1024 bits. For each SNR value, the simulation stops after having counted at least 60 error frames. The trellis termination is applied to both RSC component encoders.

The normalized width of the dither distribution D and other parameters, such as m, r, and j is an important design parameter for generating the dithered golden vector. Hence, in this paper, we perform a search for the best values of these design parameters, by using the BER and FER as a measure of quality.

Figure 4 shows the influence of design parameters D, j, r, m, and the number of matrix row N_r on the performance of the matrix-dithered golden interleaver in a Rayleigh fading channel, for two interleaving sizes $N = 400$ and $N = 1024$.

Performance of the matrix-dithered golden interleaver depends on the choice of the design parameters. The result shows that for a small frame length of approximately 400 bits, the best results are obtained by selecting the number of rows of the interleaving matrix to be $N_r = \text{floor}(\sqrt{N})$. For large frame lengths, i.e. of 1024 bits, the

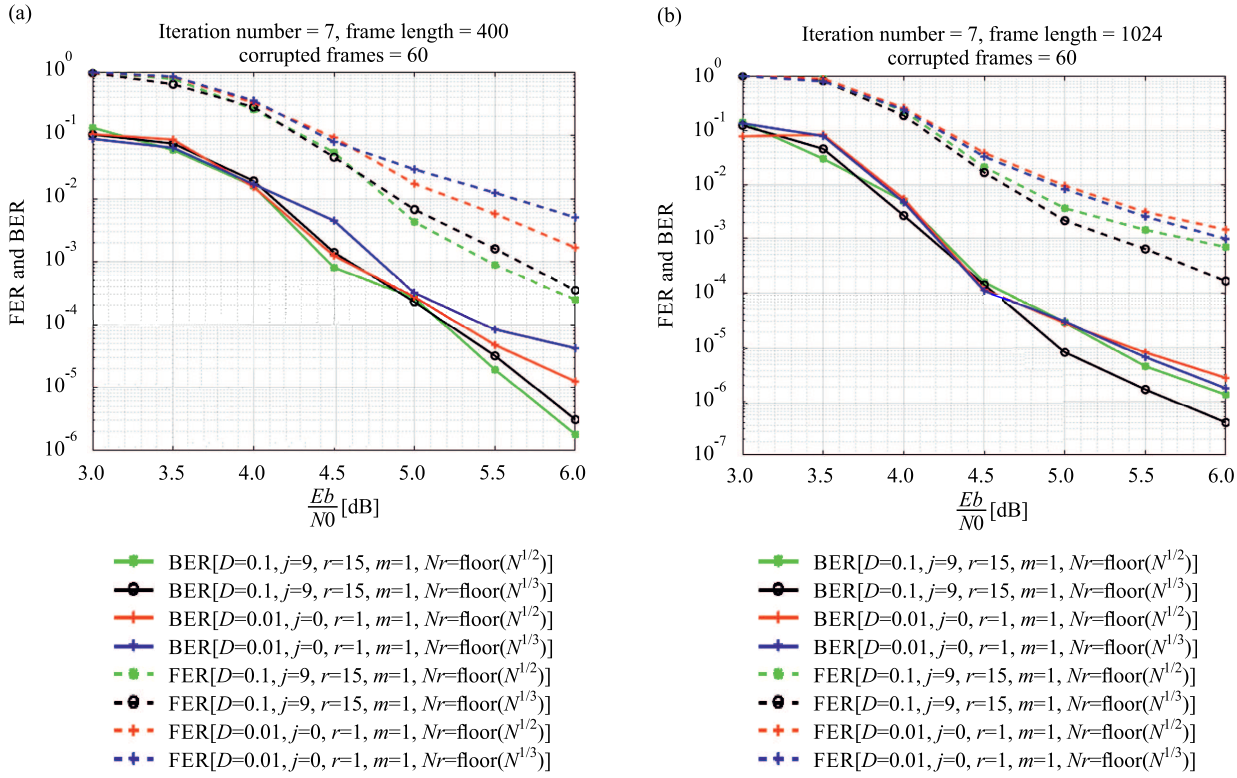

Fig. 4. Influence of the MDG interleaver design parameters on the BER and FER performance in the Rayleigh fading channel: (a) frame length $N = 400$ and (b) frame length $N = 1024$.

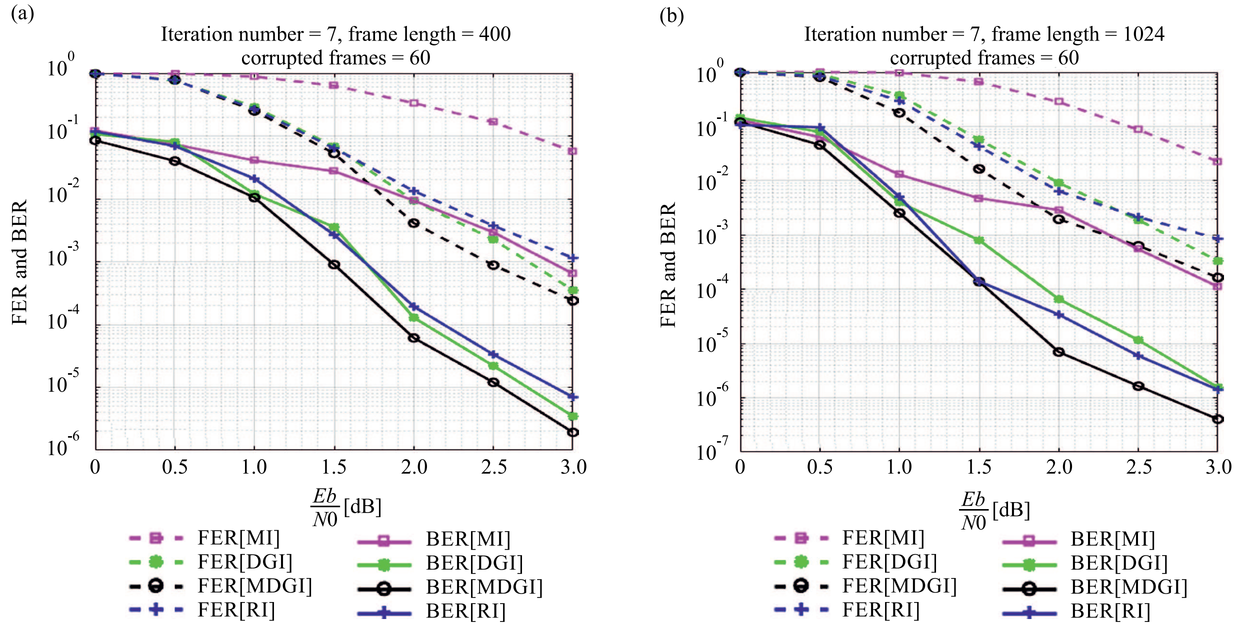

Fig. 5. BER and FER performance comparison between interleavers in the AWGN channel: (a) frame length $N = 400$ and (b) frame length $N = 1024$.

selection of the number of rows as $N_r = \text{floor}(\sqrt[3]{N})$, improves the interleaver's performance. The result also shows, that for any frame size, the best BER and FER performances is obtained for matrix-dithered golden interleaver with design parameters are set as the normalized width of the dither distribution $D = 0.1$, $j = 15$, $r = 9$, and $m = 1$.

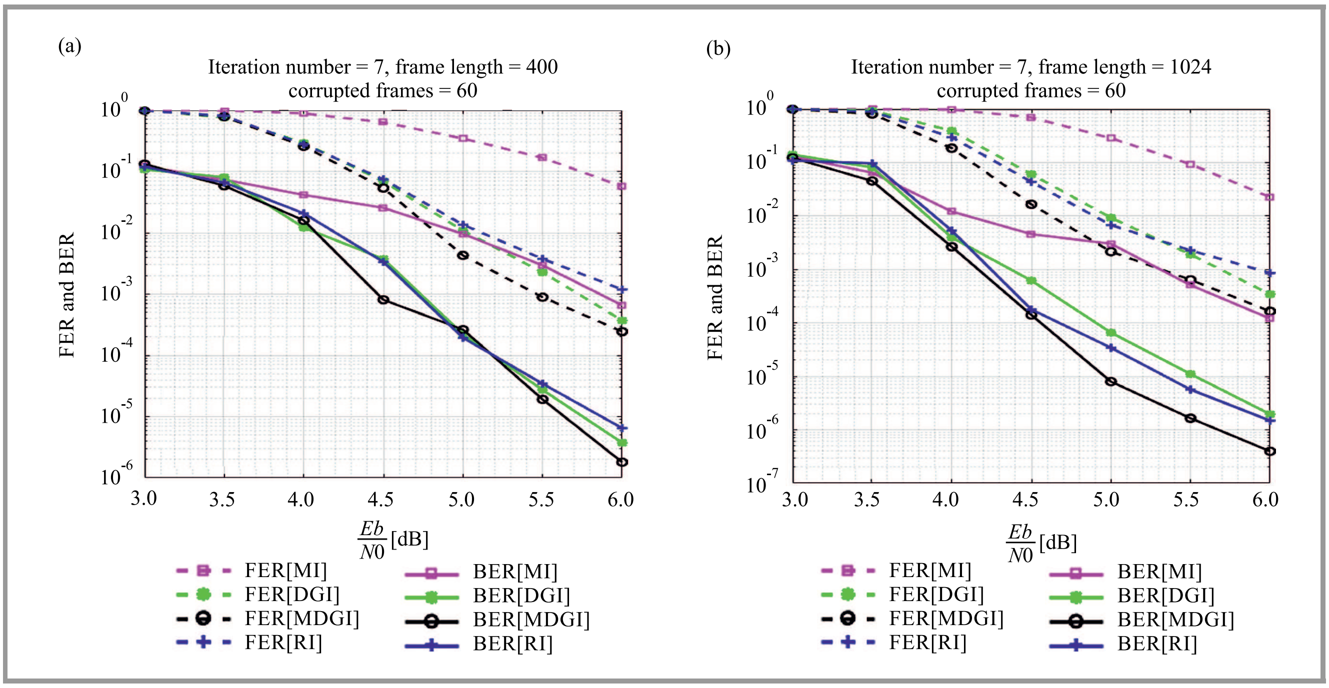

Fig. 6. BER and FER performance comparison between interleavers in the Rayleigh channel: (a) frame length $N = 400$ and (b) frame length $N = 1024$.

Table 1

Comparison of the computational complexity in terms of the number of cycles required to obtain the interleaving pattern

Interleaver length	Random interleaver	Dithered golden interleaver	Matrix interleaver	Matrix-dithered golden interleaver
$N = N_r \times N_c$	N	N	$2 \times (N_r \times N_c)$	$2 \times (N_r + N_c)$

Figures 5 and 6 show the comparisons of BER and FER performance in AWGN channel and Rayleigh fading channels, respectively. A turbo code with the interleaving length of $N = 400$ and 1024, and decoding iterations of 7 is tested in this comparison. The following algorithm design parameters are used: $D = 0.1$, $j = 9$, $r = 15$, $m = 1$, with the frame lengths equaling $N = 400$ and $N = 1024$.

Figures 5–6 show that almost the same turbo code behavior is recorded both in AWGN and Rayleigh fading channels. However, there is a fall in performance recorded in the Rayleigh fading channel, compared to AWGN. This loss equals approximately 3 dB. It is also observed that the performance of the turbo code improves significantly as the interleaving length increases, and that the MDG interleaver exhibits better BER and FER performance than other interleaver schemes for all SNRs. Considering that an interleaving length of 1024 bits and the AWGN channel are used, the results show that performance of the matrix-dithered golden interleaver is approximately 0.4 dB better than that of the random interleaver, and 0.6 dB better than that of the dithered golden interleaver at BER of 10^{-5}. A more significant gain is obtained relative to the performance of the matrix interleaver. The results show that, at an SNR value of 3 dB, the matrix interleaver has a BER of 10^{-4}, whereas the proposed interleaver has a BER of only 4×10^{-7}.

In Table 1, computational complexity of the different interleaving algorithms discussed in this work is presented. Here, complexity means the number of cycles required for the generation of interleaving patterns. The MDG interleaving scheme is extremely efficient in reducing computational complexity, compared to random, matrix, and dithered-golden interleaving schemes. By using the MDG interleaver, one may interleave a block of N_r rows and N_c columns in $2 \times (N_r + N_c)$ cycles, since only one cycle per row or column is needed. Performance is significantly improved compared to the traditional implementation which needs $2 \times (N_r \times N_c)$ cycles.

The memory requirement for different interleavers is shown in Table 2. The values are calculated based on the number of interleaving patterns to be stored, as a function of number of interleaving lengths n that need to be supported by the interleaver. The frame length is represented as N.

The results show that in the case of the random interleaver and the dithered golden interleaver, the memory size required for storing interleaving patterns depends on the number of interleaving lengths. Therefore, storage memory becomes large if multiple frame lengths have to be supported by the interleaving algorithm. However, the memory requirement of the matrix-dithered golden interleaver is independent of the number of interleaving lengths, as in this case, only one interleaving pattern, generated for the largest interleaving length, is to be stored instead of storing all interleaver patterns generated for different interleaving lengths. The slightly increased memory requirement of the

Table 2
Comparison of memory requirements of different interleaving algorithms

Interleaver	Memory requirement
Random interleaver	$n \times N$
Dithered golden interleaver	$n \times N$
Matrix interleaver	N
Matrix-dithered golden interleaver	$3 \times N$

MDG interleaver, compared with the matrix interleaver, is related to the calculation and storage of the dithered golden matrices for intra-row and intra-column permutations. Because the proposed approach offers better BER and FER performance than the matrix interleaver, such a slight additional memory requirement is acceptable.

5. Conclusion

Based on the golden section theory, a modified architecture for a matrix interleaving scheme referred to as the matrix-dithered golden (MDG) interleaver, has been suggested in this paper to further improve the performance of a turbo-coded system.

It was concluded that the proposed interleaving method improves BER and FER performance of the turbo codes. Compared with the random interleaver and the dithered golden interleaver, the MDG interleaver reduces computational complexity and storage memory requirements, as only one interleaving pattern needs to be generated and stored. The increased memory requirement of the MDG interleaver, compared with the matrix interleaver, is related to the calculation and storage of the dithered golden matrices for intra-row and intra-column permutations. Because the proposed approach is characterized by lower computational complexity, as well as by better BER and FER performance compared with the matrix interleaver, the slight additional memory requirement is acceptable.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1", in *Proc. of IEEE Int. Conf. on Commun. ICC'93*, Geneva, Switzerland, vol. 2, pp. 1064–1070, 1993 (DOI: 10.1109/ICC.1993.397441).

[2] D. Sah, "Iterative decoding of turbo codes", *J. of Adv. College of Engin. and Manag.*, vol. 3, pp. 15–30, 2017 (DOI: 10.3126/jacem.v3i0.18810).

[3] S. Jasim and A. Abbas, "Performance of turbo code with different parameters", *J. of Univer. of Babylon*, vol. 25, no. 5, pp. 1684–1692, 2017 [Online]. Available: https://iasj.net/iasj/pdf/42c4640e163c6f20

[4] R. M. Deshmukh and S. A. Ladhake, "Analysis of various puncturing patterns and code rates: Turbo code", *Int. J. Electron. Engin. Res.*, vol. 1, no. 2, pp. 79–88, 2009 [Online]. Available: https://www.idc-online.com/technical_references/pdfs/data_communications/Analysis%20of%20Various%20Puncturing%20Patterns.pdf

[5] Y. J. Harbi, "Effect of the interleaver types on the performance of the parallel concatenation convolutional codes", *Int. J. of Elec. and Comp. Sci. IJECSIJENS*, vol. 12, no. 3, pp. 25–31, 2012 [Online]. Available: http://ijens.org/Vol_12_I_03/128703-5858-IJECS-IJENS.pdf

[6] M. K. Gupta and V. Sharma, "To improve bit error rate of turbo coded OFDM transmission over noisy channel", *J. of Theoret. and Appl. Inform. Technol.*, vol. 8, no. 2, pp. 162–168, 2009 [Online]. Available: http://www.jatit.org/volumes/research-papers/Vol8No2/9Vol8No2.pdf

[7] M. Synthia and M. S. Ali, "Performance study of turbo code with interleaver design", *Int. J. of Scient. & Engin. Res.*, vol. 2, no. 7, pp. 1–5, 2011 [Online]. Available: https://www.ijser.org/onlineResearchPaperViewer.aspx?Performance_Study_of_Turbo_Code_with_Interleaver_Design.pdf

[8] A. S. Hadi, "Performance of turbo-codes with some proposed interleaver schemes", *Al-Khwarizmi Engin. J.*, vol. 2, no. 2, pp. 1–4, 2006 [Online]. Available: https://iasj.net/iasj/download/ee3c4595b8142501

[9] M. Salim, R. P. Yadav, K. Narwal, and A. Sharma, "A new block S-random interleaver for shorter length frames for turbo codes", *Bull. of Elec. Engin. and Inform.*, vol. 2, no. 4, pp. 293–298, 2013 (DOI: 10.11591/eei.v2i4.196).

[10] M. A. Khan, V. Jeoti, and R. S. Manzoor, "Performance evaluation of seed based random (SBR) interleaver in Rayleigh fading channel", in *Proc. of IEEE 4th Int. Conf. on Intell. and Adv. Syst. ICIAS 2012*, Kuala Lumpur, Malaysia, pp. 311–313, 2012 (DOI: 10.1109/ICIAS.2012.6306208).

[11] A. E. Hassan, M. Shokair, A. A. Elazm, D. Truhachev, and C. Schlegel, "Proposed deterministic interleavers for CCSDS turbo code standard", *J. of Theoret. & Appl. Inform. Technol.*, vol. 16, pp. 29–33, 2010 [Online]. Available: http://www.jatit.org/volumes/research-papers/Vol16No1/4Vol16No1.pdf

[12] X. Zou, M. Wang, and G. Feng, "A new interleaver design for iteratively decoded bit-interleaved coded modulation", *Int. J. of Soft Comput.*, vol. 3, pp. 338–343, 2008 [Online]. Available: http://docsdrive.com/pdfs/medwelljournals/ijscomp/2008/338-343.pdf

[13] O. Y. Takeshita and D. J. Costello, "New deterministic interleaver designs for turbo codes", *IEEE Trans. on Inform. Theory*, vol. 46, no. 6, pp. 1988–2006, 2006 (DOI: 10.1109/18.868474).

[14] L. H. Abderrahmane, M. Bacha, and A. Mebrek, "A new optimised interleaver structure for turbo coding", in *Proc. of 27th Canadian Conf. on Elec. and Comp. Engin. CCECE 2014*, Toronto, ON, Canada, 2014 (DOI: 10.1109/CCECE.2014.6900945).

[15] D. Wang and H. Kobayashi, "On design of interleavers with practical size for turbo codes", in *Proc. of IEEE Int. Conf. on Commun. ICC 2000. Global Convergence Through Commun.*, New Orleans, LA, USA, vol. 2, pp. 618–622, 2000 (DOI: 10.1109/ICC.2000.853570).

[16] H. R. Sadjadpour, N. J. Sloane, M. Salehi, and G. Nebe, "Interleaver design for turbo codes", *IEEE J. on Selec. Areas in Commun.*, vol. 19, no. 5, pp. 831–837, 2001 (DOI: 10.1109/49.924867).

[17] C. J. Corrada-Bravo and I. Rubio, "Deterministic interleavers for turbo codes with random-like performance and simple implementation", in *Proc. of 3rd Int. Symp. on Turbo Codes*, Brest, France, 2003 [Online]. Available: http://ccom.uprrp.edu/~labemmy/Wordpress/wp-content/uploads/2010/11/Deterministic-Interleavers-for-Turbo-Codes-with-Random-like-Performance-and-Simple-Implementation.pdf

[18] M. Kovaci, H. G. Balta, and M. M. Nafornita, "The performances of interleavers used in turbo codes", in *Proc. of Int. Symp. on Signals, Circuits and Syst. ISSCS 2005*, Iasi, Romania, vol. 1, pp. 363–366, 2005 (DOI: 10.1109/ISSCS.2005.1509929).

[19] C. C. Bravo and I. Rubio, "Algebraic construction of interleavers using permutation monomials", in *Proc. of IEEE Int. Conf. on Commun.*, Paris, France, vol. 2, pp. 911–915, 2004 (DOI: 10.1109/ICC.2004.1312634).

[20] D. S. Kim, H. Y. Oh, and H. Y. Song, "Collision-free interleaver composed of a latin square for parallel-architecture turbo codes", *IEEE Commun. Lett.*, vol. 12, no. 3, pp. 203–205, 2008 (DOI: 10.1109/LCOMM.2008.071423).

[21] G. A. A. Kareem, "Efficient selection of circular and random interleaver for turbo codes multiuser CDMA system with Hadamard spreading code in AWGN and Rayleigh fading channels", *J. of Engin. and Sustainable Develop.*, vol. 16, no. 2, pp. 117–132, 2012 [Online]. Available: https://www.iasj.net/iasj/pdf/d6335a53589140fe

[22] G. B. Purushottama and B. R. Sujatha, "Turbo codes with golden section interleaver", *Int. Res. J. of Engin. and Technol.*, vol. 2, no. 3, pp. 1069–1073, 2015 [Online]. Available: https://www.irjet.net/archives/V2/i3/Irjet-v2i3158.pdf

[23] P. D. Bahirgonde and S. K. Dixit, "BER analysis of turbo code interleaver", *Int. J. of Comp. Appl.*, vol. 126, no. 14, pp. 1–4, 2015 (DOI: 10.5120/ijca2015906278).

[24] B. Raje and K. Markam, "Review paper on study of various Interleavers and their significance", *Int. Res. J. of Engin. and Technol.*, vol. 5, no. 11, pp. 430–434, 2018 [Online]. Available: https://www.irjet.net/archives/V5/i10/IRJET-V5I1082.pdf

[25] S. Crozier, J. Lodge, P. Guinand, and A. Hunt, "Performance of turbo-codes with relative prime and golden interleaving strategies", in *Proc. of the 6th Int. Mobile Satellite Conf. IMSC'99*, Ottawa, Canada, 1999 pp. 268–275.

[26] L. H. Abderrahmane and S. Chellali, "Performance comparison between Gaussian interleaver, Rayleigh interleaver, and dithered golden interleaver", *Ann. of Telecommun.*, vol. 63, no. 7–8, pp. 449–452, 2008 (DOI: 10.1007/s12243-008-0037-2).

Essedik Iftene received his B.Sc. degree in Electronics from University of Sciences and Technologies, Algeria, in 2008, and M.Sc. in Space Communication Systems from High Institute of Aeronautics and Space, Toulouse, France, in 2010. He also earned an M.Sc. in Space Technologies and Applications from University of Sciences and Technologies, Algeria. Currently, he is a research assistant at the Department of Space Instrument Research at the Satellite Development Center (CDS), Algeria, and is pursuing a Ph.D. in Space Instrument Design at University of Sciences and Technologies, Algeria. His research interests include wireless networks, channel coding and modeling, and their application in space communication systems.

https://orcid.org/0000-0003-0995-5850
E-mail: essedik.iftene@univ-usto.dz
Laboratoire de codage et de la sécurité
de l'information (LACOSI)
Département d'Electronique
Université des Sciences et de la Technologie
d'Oran Mohamed-Boudiaf USTO-MB
BP 1505, El M'naouer
Bir El Djir 31000 Oran, Algeria
Algerian Space Agency (ASAL)
Centre of Satellites Development
Ibn Rochd USTO Oran BP 4065, Algeria

Adda Ali-Pacha majored in Telecommunications Engineering in January 1986. He received a B.Sc. in Mathematics in 1986 and an M.Sc. in Signal Processing in 1993. He also obtained a Ph.D. in Data Security in 2004. He worked for a telecommunications authority (PTT Oran), holding the position of the Head of Telephone Traffic for two years (1986-1988). He has been working for University of Sciences and Technology of Oran (USTO) Algeria since 1989, serving as an academic teacher/researcher in the Electronics Institute. His research focuses on telecommunication domains.

E-mail: adda.alipacha@univ-usto.dz
Laboratoire de codage et de la sécurité
de l'information (LACOSI)
Département d'Electronique
Université des Sciences et de la Technologie
d'Oran Mohamed-Boudiaf USTO-MB
BP 1505, El M'naouer
Bir El Djir 31000 Oran, Algeria

Lahcène Hadj Abderrahmane is a researcher at the Satellite Development Center. Currently, he works on RF systems and channel coding, focusing primarily on designing satellite communication systems. He is an RF team leader at the Department of Space Instrument Research and an Assistant Professor teaching telecommunications to Master's degree students.

E-mail: hadjabderrahmanel@yahoo.fr
Algerian Space Agency (ASAL)
Centre of Satellites Development
Ibn Rochd USTO Oran BP 4065, Algeria