

The Design of New Modified Block and Matched S-Random Interleavers for Turbo Codes

Tamer H. M. Soliman, Fengfan Yang, and S. Ejaz

Abstract—It is known that one of the essential building blocks of turbo codes is the interleaver and its design using random, semi-random (S-Random) and deterministic permutations. In this paper, two new types of turbo code interleavers, Modified Block S-Random (MBSR) interleaver and Modified Matched S-Random (MMSR) interleaver are proposed. The design algorithm for the new interleavers is described in depth, and the simulation results are compared to the two new interleavers with different existing interleavers based on the BER (Bit Error Rate) performances of the turbo codes. Through the simulation, we find a better performance of the MBSR interleaver than random and practical interleavers. In addition, the performance of MMSR interleaver is close to the code matched interleaver at different frame sizes and with less complex design.

Index Terms—Interleaver, semi random, turbo codes, weight distribution.

I. INTRODUCTION

A typical turbo code (TC) was first introduced in 1993 by Berrou *et al.*, [1] as a class of near channel capacity achieving codes. This turbo code is constructed by concatenating two parallel convolutional codes via an interleaver as shown in Fig. 1. The interleaver is an indexing function given by a permutation of bits index in the information frames with N frame length that plays a crucial role in the turbo codes architecture. Interleaver has three main functions: a) it constructs a long code from small memory convolutional codes by permuting the input bits such that the two constituent encoders are operating on different order input bits. b) Provides “scrambled” information data to the second constituent encoder to decorrelate the inputs of the two decoders, so that an iterative suboptimum decoding algorithm based on “uncorrelated” information exchange between the two constituent decoders can be applied. c) It changes the weight distribution of turbo codes such that, the overall weight for the generated codeword depends on how the outputs from the two constituent encoders are teamed together.

The main two properties characterize any interleaver are the interleaver spreading property which is the distance between adjacent bits before interleaving, and the randomness property that provides a non-fixed indexing function which is a good factor for correction in the iterative decoding. Turbo code interleaver types have been

extensively studied in different ways, and they fall into two main classes: Random interleavers and Deterministic interleavers.

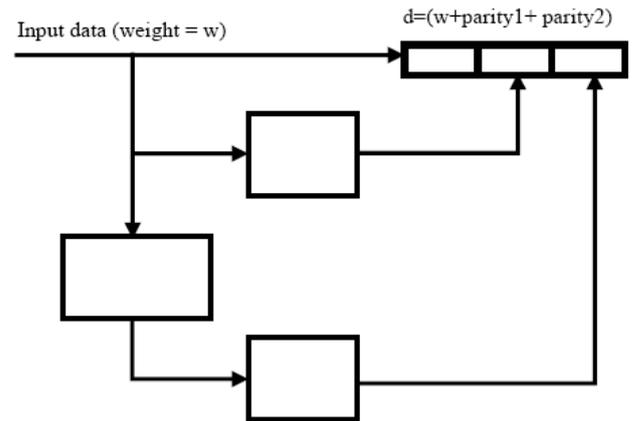


Fig. 1. Parallel concatenated turbo codes.

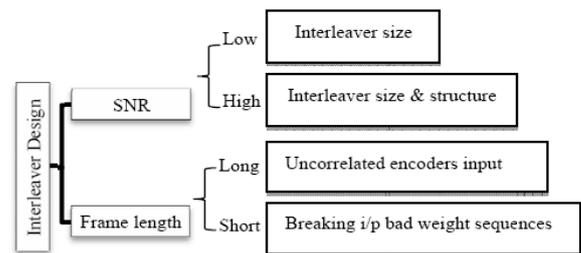


Fig. 2. Interleaver design dependence.

A random interleaver is simply a random permutation π with new positions produced by an indexing function based on the uniform probability distribution. While in major deterministic interleavers these new positions are linearly interleaved with a designed index function. Interleaver design of turbo codes has been recognized as the key factor in many publications. This design depends on different factors such as the signal-to-noise ratio (SNR) and the frame size (N) to be used; Fig. 2 summarizes this interleaver design dependence criteria. A Random interleaver in [1] utilized in turbo codes performs reasonable for long block sizes. However, for short block size, the performance of turbo codes with a random interleaver degrades substantially even it performs worse than that of convolutional codes with similar computational complexity. Furthermore, if we take memory and throughput requirements into consideration, it is preferable to employ the deterministic interleavers with lower interleaving and deinterleaving complexity for a convenient implementation.

One method for the design of deterministic permutations is based on block interleavers [2], [3]. In this interleaver, the input data are written along the rows of the memory

Manuscript received May 31, 2014; revised July 17, 2014. This work was supported in part by College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

The authors are with the Nanjing university of Aeronautics and Astronautics, College of Electronic and Information Engineering, Nanjing (e-mail: thms78@gmail.com, saqibejaz@gmail.com).

configured as $N = k \times l$ matrix of k -rows and l -columns and then read out along the columns. In the case of the SNR factor, for low SNR values, any interleaver works conveniently as long as it guarantees that the two inputs of the Recursive Systematic Coder (RSC) encoders are sufficiently uncorrelated. On another hand, numbers of interleaver structures have been designed at moderate to high SNR, where the code performance depends on both the interleaver structure and size [4].

The designed algorithm in [5] is based on eliminating of the low weight paths in the code trellis that give large contributions to the error probability (with long frames). In addition, the author in [6] adopted his design on suppressing the interleaver correlation and breaking up the self-terminating bad weight-2 input sequences.

Although a lot of work was done in the turbo coding interleaver design, the goal of this paper is to build two interleaver algorithms that have the ability to give the turbo coding system better performance in both short and long frames. We first design an interleaver that has deterministic characteristics with some randomness behavior, and then compare it with some other popular turbo interleavers using the bit error rate performance criteria. In the second design we derive a fast algorithm for a turbo matched interleaver based on reducing the algorithm conversion constrains specially for large frame length.

The rest of this paper is organized as follows. Section II introduces the terminology used throughout the interleaver design and its criteria. In Section III, we present the designed method of the MBSR interleaver; then compare its performance with some other turbo code interleavers. In Section IV the designed algorithm of MMSR interleaver is presented, then simulation results of its BER performance with turbo codes compared with random, practical size and matched interleavers are introduced. Finally, the conclusions are given in Section V.

II. THE OPTIMUM INTERLEAVING DESIGN

There are two major algorithms in the design of an interleaver: 1) the correlation between parity and the information input data sequences and 2) the distance spectrum properties (weight distribution) of the code. The first criterion is a measure of the effectiveness of the iterative decoding algorithm. The designed algorithm in [7] depends on the fact that if the two data sequences are less correlated, then the performance of the iterative decoding algorithm improves. For Criterion 2 the weight distribution of turbo codes can be defined as how the codewords from one of the simple component encoders are teamed with codewords from the other encoder. The turbo codes construction depends mainly on the infinite impulse response (IIR) characteristics of its recursive systematic convolutional component encoders, which has infinite weight (for a never-ending information stream). This IIR property is important for building turbo codes, because it avoids low-weight encodings that are impervious to the action of the permuters. The best interleaver design has as its objective to include the two designed algorithms, matching low-weight parity check sequences of first RSC constituent encoder with high-weight parity check

sequences of the other encoder (i.e. the ability of breaking the low weight input sequence patterns) and keeping of its data less correlative (suitability to be iteratively decoded).

A semi-random (S-random) interleaver in [8] satisfies in its design two important characteristics: limited deterministic design mixed with some degree of randomness. This S-random interleaver can map low weight input patterns in the first component encoder to high weight patterns in the second encoder. Depending on this S-Random algorithm, our main task is to construct new interleaver structures that can suppress the bad input sequences and have the ability of preserving the randomness of its bits distribution. These new interleavers can have the ability of giving good performance of turbo codes on low/high SNR and with long/short frames.

III. MODIFIED BLOCK S-RANDOM (MBSR) INTERLEAVER

A. Designed Algorithm

The block interleaver function defined by a matrix with k rows and l columns with $N = k \times l$ is:

$$\pi(i + j \times k + 1) = i \times l + j + 1 \quad (1)$$

where, $i \in I = \{0, 1, \dots, k - 1\}$ and $j \in J = \{0, 1, \dots, l - 1\}$. This interleaver can break the low-weight input sequence, as it is limited with one row. Nevertheless, it fails to break many combined lower-weight sequences that appears in several consecutive rows [9]. To solve this problem, we design a Modified Block S-random interleaver depending on columns and rows reordering of the block interleaver, which can spread low-weight sequences as much as possible. In [10] an algorithm for columns reordering is applied when the maximum length burst of errors is greater than the k (row length), but in our algorithm we use both columns and rows reordering technique. This new algorithm increases the interleaver ability to break bad sequences. We can consider MBSR interleaver is an improved version of the block interleaver as it can combine the characteristics of block interleaver with that of S-random interleaver.

Based on the above design criteria, our new interleaver structure is constructed by the following procedure:

- 1) Forming the conventional data matrix ($N = k \times l$) as:

$$r(i, j) = 1 + ik + j \quad (2)$$

where $i \in I = \{0, 1, \dots, k - 1\}$ and $j \in J = \{0, 1, \dots, l - 1\}$.

- 2) Select the index factor S_1 such that $S_1 < \sqrt{l/2}$, then we apply the S-Random algorithm to the first row $r(0, j)$ of the array. First we randomly select the first position from the finite set $\{1, 2, \dots, l\}$, then randomly select next possible future positions (order) and arrange them one by one to form the interleaved sequence by comparing each position with the last S_1 positions already selected and for x and $y \in \{1, \dots, l\}$ check the next condition:

$$|\pi_{S_1}(x) - \pi_{S_1}(y)| > S_1, \text{ with } |x - y| \leq S_1 \quad (3)$$

- 3) If the condition is satisfied, then we go to the next possible position and if not we must select another

position until the condition satisfaction. Finally, columns permutations are done depending on these new positions.

- 4) Columns rearrangement is done for each column by applying the same criteria with index factor S_2 , where $S_2 < \sqrt{k/2}$ and with column new positions satisfying:

$$|\pi_{S_2}(x) - \pi_{S_2}(y)| > S_2, \text{ with } |x - y| < S_2 \quad (4)$$

where, x and $y \in \{1, \dots, k\}$.

- 5) Finally, we read the output of the data in columns.

B. Simulation Analysis

This new MBSR interleaver design aims to combine the advantages of the block interleaver and of the S-Random interleaver. The randomness and the bit distribution of uniform random, practical size [6] and MBSR interleavers with $N=1024$ bits are shown in Fig. 3. In this figure, the X-axis is the input bit positions of each interleaver and the Y-axis is the interleaved (permuted) bit positions. Fig. 3 (a) and (b) show the comparison between the uniform random and practical size interleavers, it can be observed that the points in practical size interleaver are distributed more uniformly in the plane. This property can help to avoid short error events in one component code to be interleaved to short error events in the other component code.

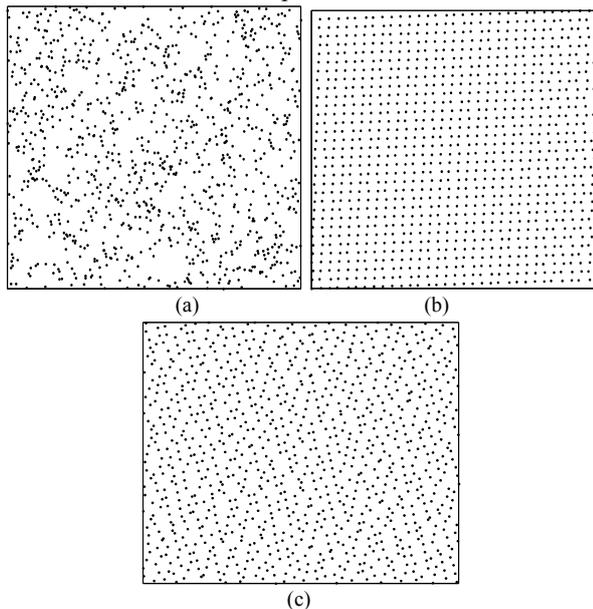


Fig. 3. Graphical representation of different interleavers of length $N=1024$ bits. (a) Uniform Random Interleaver, (b) Practical Size Interleaver and (c) MBSR Interleaver.

From Fig. 3 (c) we can observe that the MBSR interleaver combines the two characteristics of Random and Practical interleavers. As for this plot, we can observe irregularity in the density of points in the plane; we can also observe that there are some periodic patterns. For the AWGN channel simulation, we have used a rate 1/3 turbo code consists of two identical (1, 5/7) RSC with code rate $R = 1/2$. The interleaver sizes of $N = 256, 1024$ bits and the log-map decoder with 8-iterations algorithm are used. We compared the MBSR interleaver performance with different types of turbo code interleavers. The first interleaver is a Practical size interleaver based on the algorithm in [6].

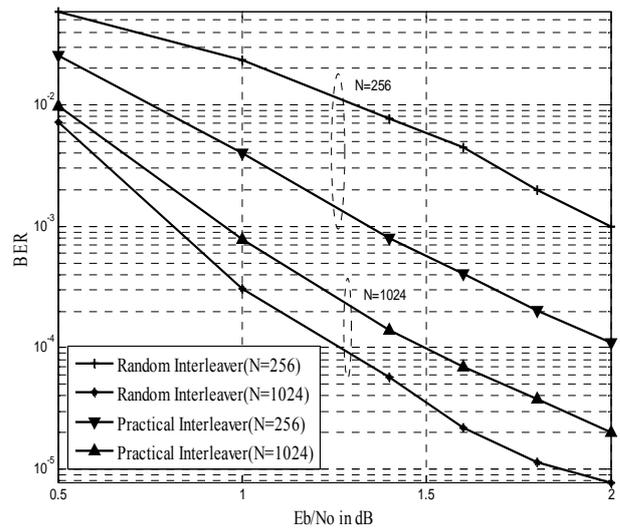


Fig. 4. BER performance comparison between 4-states rate 1/3 turbo code with Random and Practical interleavers at $N= (256$ and $1024)$ bits.

The other two types are the random interleaver that follows the law of uniform probability distribution and the 1-step block S-Random interleaver that depends in his design on columns permutation only neglecting the rows permutations. From Fig. 4 and Fig. 5 we can observe that for the short frames ($N = 256$ bits), the practical interleaver gives a better performance than that of random interleaver. However, for long frames (1024 bits) the performance of the random interleaver is better than that of 1-step and practical interleavers. Fig. 5 indicates that for the lower values of E_b/N_0 , the performances of the random and MBSR interleavers are almost identical. However, for the higher values one notices a difference in performance between them, as in case of the MBSR interleavers it gives better performance than that of random one.

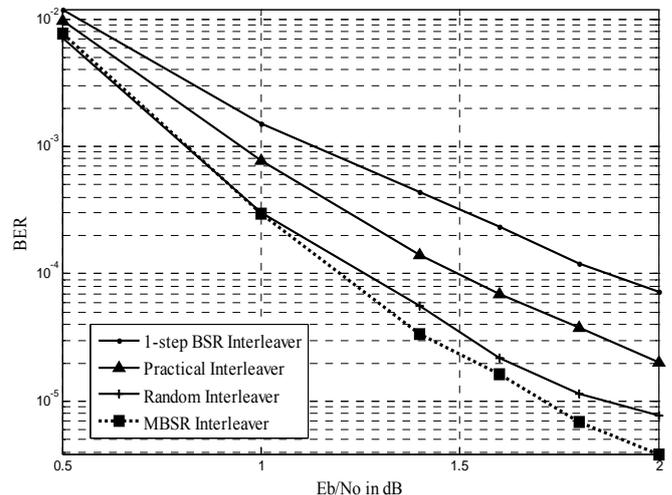


Fig. 5. BER performance comparison between 4-states rate 1/3 turbo code with MBSR, 1-step BSR, Random and Practical interleavers at $N=1024$ bits.

IV. MODIFIED MATCHED S-RANDOM (MMSR) INTERLEAVER

An interleaver can be designed to break low weight input sequence patterns, which produce low weight parity-check sequences at the output of one of the constituent encoders, so that the input sequences to the other constituent encoder will generate high weight parity-check sequences. The weight

distribution of error correcting codes can be used to compute the error performance bounds for the performance evaluation of any linear block codes. Turbo codes can be represented as an equivalent block code if its constituent convolutional encoders are terminated to the all-zero state. The simple calculation method of the weight distribution of terminated convolutional codes based on the state transition matrix of encoder is already shown [11].

A. Designed Algorithm

As in the previous section we combined the characteristics of Block and S-Random interleavers to design MBSR interleaver, here the MMSR interleaver design will depend on the combination between S-Random and matched interleavers. As we explained before that S-Random, constraint spreads the elements positions such that any two elements within a window of size S will not be located in a window of size S in the interleaved sequence. In our design of MMSR interleaver, we modify the designed algorithm in [5] in large frames depending mainly on removing bad low weights (two, three and 4) input sequences that have significant contribution to the error performance with less complexity as following.

1) Bad weight-2 input sequences

Bad input sequences are these that produce low weights at encoder's outputs. The bad weight-2 sequence (00 ... 00100100 ... 00) with the minimum distance between the two ones is ($\mu = 3$) generates a finite output codewords. This bad sequence forces the encoder back to the all-zero state without any trellis termination will be in the form of:

$$U_2(D) = (1 + D^{3k})D^\tau \tag{5}$$

where $k = 1,2,3, \dots$ and time delay $\tau = 1,2, \dots$. From Fig. 1 the overall weight of any generated codeword is the summation of the input weight, weight of first parity and weight of second parity. For breaking bad weight-2 inputs, we need that $d > d_{max}^{w=2}$, where $d_{max}^{w=2}$ is the maximum weight generated by the bad weight-2 input sequence that should be eliminated by our interleaver design.

For the minimum parity check weight generated by weight-2 input (z_{min}), the parity-check sequences generated by the input sequence in (5) can be expressed as:

$$w(p_j) = k_j(z_{min} - 2) + 2 \tag{6}$$

where, $k_j=1, 2, 3 \dots$ and $j=1, 2$. So for breaking bad low weight-2 input sequences we need:

$$d = 2 + (k_1(Z_{min} - 2)) + (k_2(Z_{min} - 2)) > d_{max}^{w=2} \tag{7a}$$

$$6 + (k_1 + k_2)(Z_{min} - 2) > d_{max}^{w=2} \tag{7b}$$

$$k_1 + k_2 > \frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} \tag{7c}$$

As ($k_1=1, 2, 3 \dots$), so for $k_1 = 1$ (for the worst case).

$$k_2 > \frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} - 1 \tag{8}$$

Multiplying both sides by μ , then:

$$\mu k_2 > \mu \left[\frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} - 1 \right] \tag{9}$$

As, by applying the S-random constraint we can get:

$$|\pi(i_1) - \pi(i_2)| \geq (S + 1), \text{ whenever } |i_1 - i_2| \leq S \tag{10}$$

$$\mu k_2 \geq S + 1, \text{ for } \mu k_1 \leq S \tag{11}$$

From (9), (11) we have that:

$$(S + 1) > \mu \left[\frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} - 1 \right] \tag{12}$$

$$S > \mu \left[\frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} - 1 \right] - 1 \tag{13}$$

Therefore, the minimum value of S for S-Random constraints to allow breaking of bad weight-2 sequences will be as:

$$S_{min}^{w=2} = \mu \left[\frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} - 1 \right] - 1 \tag{14}$$

2) Bad weight-3 input sequences

For bad weight-3 input sequences that generate low weight codewords, Table I indicates the weight-3 input sequences that can generate codewords with most significant contribution to the error performance of 4-state ($1, 1 + D^2/1 + D + D^2$) turbo codes.

TABLE I: BAD WEIGHT-3 INPUT SEQUENCES THAT GENERATE SIGNIFICANT LOW WEIGHT CODEWORDS

Input weight (w)	Output weight (d)	Parity weights	Input sequences
3	7	2	111
	9	2, 4	111 10101 - 110001
	11	2, 4, 6	111 10101 - 110001 100000011 - 110000001
	13	2, 4, 6, 8	111 10101 - 110001 100000011 - 110000001 11000000001

The mapping from one of these weight-3 input sequences at the first constituent encoder to another bad weight-3 input sequence for the other constituent encoder is very easy to be prevented. As an example for having the output weight ($d = 9$) codewords, first it can be produced by an input weight $w = 3$ and parity check weights as $P_1 = 2$ when $P_2 = 4$ or $P_1 = 4$ for $P_2 = 2$. For the first case, the input sequence of the first encoder is (111); its parity sequence (101) with $P_1 = 2$ and with output sequence (111011). The

input sequence of the second encoder will be (10101 or 110001) to make $P_2 = 4$. This mapping can be easily prevented by a very simple S-Random constraint. Also for the second case when the input sequence of the first encoder is (10101 or 110001) giving output sequence (1101100111 or 111000010111) with $P_1 = 4$. Also the mapping from these sequences to (111) sequence at the input of the second encoder can be broken easily by S-Random constraint with $S_{min}^{w=3} \geq 10$, where $S_{min}^{w=3}$ is the minimum value of S to allow breaking of bad weight-3 input sequences.

3) *Bad weight-4 input sequences*

The same as with weight-2 input sequences with overall output codewords weight ($d = 4 + P1 + P2$) so we have:

$$d = 12 + (k_3 + k_4 + k'_3 + k'_4) \cdot (Z_{min} - 2) \quad (15)$$

where k_3, k_4, k'_3 and $k'_4 = 1, 2, 3, \dots$ For the output codeword weight to be greater than the maximum weight generated by the bad weight-4 input sequences that should be eliminated ($d_{max}^{w=4}$), the condition is: $d > (d_{max}^{w=4})$, so:

$$k_3 + k_4 + k'_3 + k'_4 > \frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} \quad (16)$$

As ($k_3, k_4=1, 2, 3, \dots$), so for $k_3 = k_4 = 1$.

$$k'_3 + k'_4 > \frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} - 2 \quad (17)$$

For (17) by multiplying both sides by μ , then:

$$\mu(k'_3 + k'_4) > \mu \left[\frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} - 2 \right] \quad (18)$$

By using the weight-4, the S-Random constraint is:

$$|\pi(i_1) - \pi(i_3)| \text{ and } |\pi(i_2) - \pi(i_4)| \geq (S + 1) \quad (19)$$

where, $|i_1 - i_2|$ and $|i_3 - i_4| \leq S$.

$$\mu k'_3 \geq (S + 1) \quad (20a)$$

$$\mu k'_4 \geq (S + 1) \quad (20b)$$

From Eq. (20a) and (20b) we have:

$$\mu k'_3 + \mu k'_4 \geq 2(S + 1) \quad (21)$$

$$\mu(k'_3 + k'_4) \geq 2(S + 1) \quad (22)$$

From Eqns. (18), (22),

$$2(S + 1) > \mu \left[\frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} - 2 \right] \quad (23)$$

$$S > \frac{\mu}{2} \left[\frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} - 2 \right] - 1 \quad (24)$$

So, the minimum value of S for S-Random constraints to allow breaking of bad weight-4 sequences will be:

$$S_{min}^{w=4} = \frac{\mu}{2} \left[\frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} - 2 \right] - 1 \quad (25)$$

By modifying the value of S in the MMSR interleaver designed algorithm with $S_{min}^{w=2}$, $S_{min}^{w=3}$ and $S_{min}^{w=4}$, the interleaver has the ability to eliminate bad low weights (2, 3 and 4) input sequences giving better performance especially with long frames. The designed algorithm can be simplified as shown in the following:

a) *First step: S-random interleaver constraint*

First, we select the starting value of S at $S = \sqrt{(N/2)}$ and a specific number of iterations (iter) (how many main loops can occur before failure).

Calculate $S_{min}^{w=2}$ and $S_{min}^{w=4}$ from Eq. (14) and (25) respectively.

Pick the first position randomly from the set $\varphi = \{1, 2, 3, \dots, N\}$ as ($pos_1 = rand(1:N)$) and erase this selection (pos_1) from the set φ .

Each randomly selection of the next position is compared to the S previously selected. If the current selection is equal to any of the S previous selections within a distance of $\pm S$, then the current selection is rejected and selects another one. If not then pick the current selection, erase it from the remaining set ($\varphi - selections$) and then go to the next selection and so on. If no possible selection converges with the requirements under the defined iterations number, then the value of S is reduced by 1 and starts the new search again until having a satisfied selection at a certain value of S (Sgen).

b) *Second step: code matched constraint*

Check whether the designed S-Random constraint satisfies breaking of the bad low weights (2, 3 and 4) input sequences by checking the following condition:

$$Sgen \geq \max(S_{min}^{w=2}, S_{min}^{w=4}) \quad (26)$$

If the condition is satisfied, then end the design and save the current mapping as an interleaver output.

Otherwise, go to step (d) with extra conditions for each selected position. These conditions needs to be satisfied for both weight-2 and weight-4 as following:

Weight-2 input sequences:

$$k_2 > \frac{d_{max}^{w=2} - 6}{(Z_{min} - 2)} - k_1 \quad (27)$$

Weight-4 input sequences:

$$k'_3 + k'_4 > \frac{d_{max}^{w=4} - 12}{(Z_{min} - 2)} - (k_3 + k_4) \quad (28)$$

B. *Simulation Analysis*

In this section, the performance of the MMSR interleaver is represented by analyzing the results from BER curve where the simulations were run for different interleavers lengths (256, 400, 1024 and 2048 bits).

Fig. 6 indicates that the MMSR interleaver has good random distributed characteristics compared with the uniform random interleaver in addition to its matching designed distribution. For simulation, we use the rate 1/3 turbo code consists of two RSC encoders each with code rate $R = 1/2$. The coded bits are transmitted over AWGN channel; and eight log-MAP decoding iterations are performed at the decoder. The BER curves are shown in Fig. 7, 8 and 9 which depict the proposed MMSR interleaver against the random, practical [8] and code matched [4]

interleavers.

The results of the simulations yield to a better performance of the designed MMSR interleaver than those performances of practical and random interleavers. Although we can observe the same performance of MMSR and code matched interleavers for all simulated frame lengths (256, 400, 1024 and 2048 bits) as shown in Fig. 7, 8 and 9, but the algorithm needs less time for conversion with fewer constraints compared with that of the matched interleaver algorithm specially for long frame lengths (1024 and 2048 bits) as the probability that $S_{gen} \geq \max(S_{min}^{w=2}, S_{min}^{w=4})$ increases and only the S-Random constrain is sufficient for the algorithm to build the needed interleaving sequence.

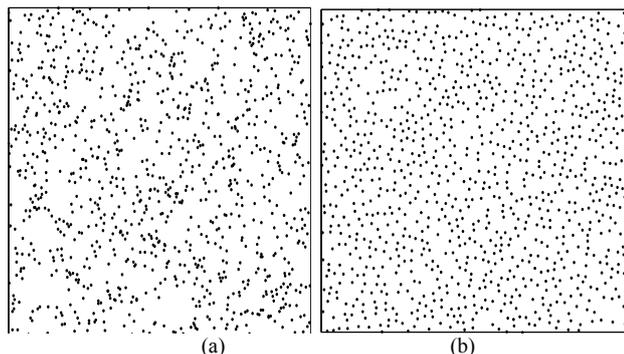


Fig. 6. Graphical representation of (a) Uniform Random and (b) MMSR interleavers with length $N=1024$ bits.

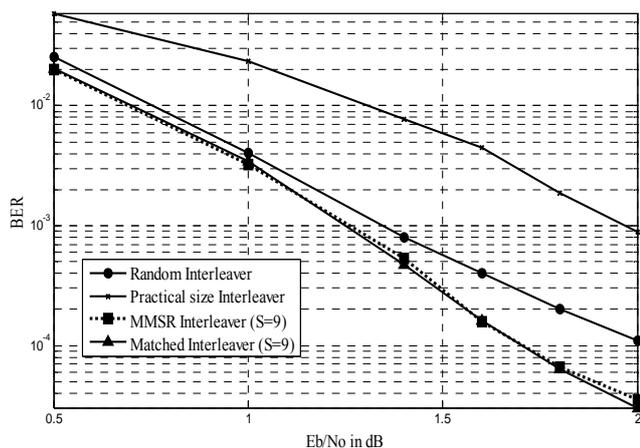


Fig. 7. BER performance comparison between 4-states rate 1/3 turbo code with MMSR, Matched, Random and Practical interleavers at $N=256$ bits.

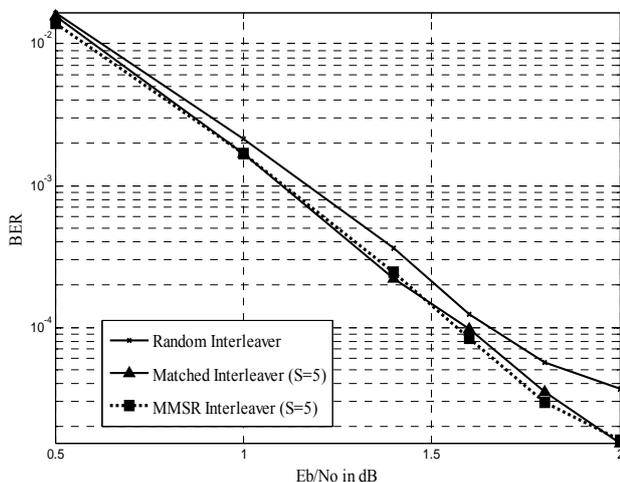


Fig. 8. BER performance comparison between 4-states rate 1/3 turbo code with MMSR, Matched and Random interleavers at $N=400$ bits.

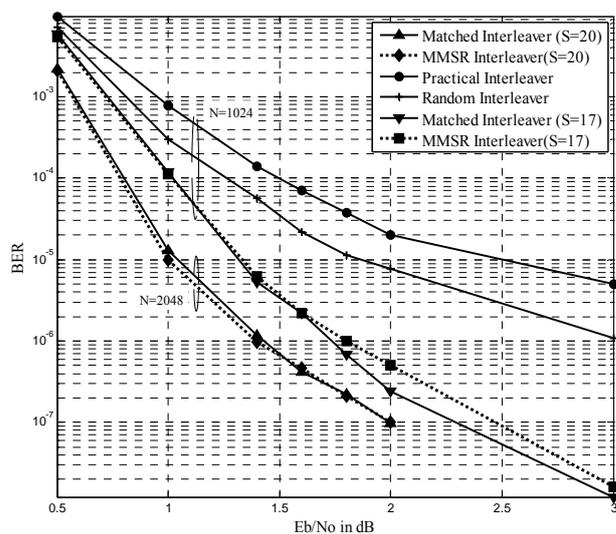


Fig. 9. BER performance comparison between 4-states rate 1/3 turbo code with MMSR, Practical, Matched and Random interleavers at $N= (1024$ and $2048)$ bits.

V. CONCLUSION

The interleaver plays a vital role in the performance improvement of turbo coding system. In this paper, we have presented two new efficient algorithms for turbo code interleavers design. In this work, our design depends mainly on the S-Random constraint to have a good interleaver pattern that ensures good spreading properties in breaking bad low weights input sequences. In MBSR interleaver designed criteria, we combined S-Random constraint with block interleaver, while it is combined with matched interleaver algorithm in the design of MMSR interleaver which gives a good performance with very simple design, especially for long frames. For the simulated cases of these two interleavers, a good performance is obtained at different block lengths ($N= 256, 400, 1024$ and 2048) bits.

ACKNOWLEDGMENT

I would like to thank the past and present Staff Members of the College of Electronic and Information Engineering of Nanjing University of Aeronautics and Astronautics for their great support and guidance especially my great respectable professor Fengfan Yang.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. Int. Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [2] G. C. Clark, Jr. and J. B. Cain, *Error Correction Coding for Digital Communications*, New York: Plenum, 1981.
- [3] I. Richer, "A sample interleaver for use with viterbi decoding," *IEEE Trans. Commun.*, vol. 26, pp. 406–408, Mar. 1978.
- [4] J. Yuan, B. Vucetic, and W. Feng, "Combined turbo codes and interleaver design," *IEEE Trans. Commun.*, vol. 47, pp. 484–487, Apr. 1999.
- [5] W. Feng, J. Yuan, and B. S. Vucetic, "A code-matched interleaver design for turbo codes," *IEEE Trans. Commun.*, vol. 50, pp. 926–937, June 2002.
- [6] D. Wang and H. Kobayashi, "On design of interleavers with practical size for turbo codes," in *Proc. 2000 IEEE International Conference Communications*, vol. 2, pp. 618 – 622, 2000.
- [7] J. Hokfelt, O. Edfors, and T. Maseng, "Turbo codes: correlated extrinsic information and its impact on iterative decoding performance,"

in *Proc. IEEE 49th Vehicular Technology Conference '99*, Houston, Texas, vol. 3, pp. 1871-1875.

- [8] S. Dolinar and D. Divsalar, "Weight distribution for turbo codes using random and nonrandom permutations," *JPL Progress*, pp. 56-65, August 15, 1995.
- [9] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc. ICC'95*, Seattle, WA, June 1995.
- [10] E. Dunscombe and F. C. Piper, "Optimal interleaving scheme for convolutional codes," *Electronics Letters*, vol. 25, no. 22, pp. October 26, 1989.
- [11] J. K. Wolf and A. J. Viterbi, "On the weight distribution of linear block codes formed from convolutional codes," *IEEE Trans. Commun.*, vol. 44, pp. 1049-1051, Sept. 1996.



Tamer H. M. Soliman was born in Alsharkia Egypt, 1978. He received his B.S. in communication and radar engineering from MTC, Cairo, Egypt in 2000. He received his M.S. in applications of error correction codes in satellite communication from faculty of engineering, Alexandria University, Alexandria, Egypt, in 2011. He is currently in Ph.D. degree in NUAA (Nanjing University of Aeronautics and Astronautics). His major research interests are information theory and channel coding specially coded-cooperative diversity, turbo coding and secured chaotic techniques.



Saqib Ejaz received the B.Sc. electrical engineering with honors in 2006 from University of Engineering and Technology, Taxila, Pakistan and he received his M. Sc. in electronic engineering from the GhulamIshaq Khan Institute of Engineering Sciences and Technology, KPK, Pakistan in 2008. At present, he is a Ph.D candidate at College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics. His main research interests are focused on the areas of channel coding (turbo, Reed Muller and LDPC codes), coded-cooperative diversity, signal space diversity, labelling diversity, and MIMO systems.



Fengfan Yang received the B.Sc., M.Sc., and Ph.D. degrees from Nanjing University of Aeronautics & Astronautics (NUAA), Northwestern Polytechnical University and Southeast University in 1990, 1993, and 1997 in P. R. China, respectively, all in electronic engineering. He has been with College of Information Science and Technology, Nanjing University of Aeronautics & Astronautics since May 1997. From October 1999 to May 2003, he was a research associate at Centre for Communication Systems Research, University of Surrey, UK, and Department of Electrical and Computer Engineering (ECE), McGill University, Canada. He currently spends one year sabbatical leave as a visiting scholar at ECE, McGill University. His major research interests are information theory and channel coding, especially at iteratively decodable codes, such as turbo codes and LDPC codes, and their applications for mobile and satellite communications.