www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 12 Dec. 2016, Page No. 19666-19669

On theory and Performance of high throughput parallel block and S-random interleavers for Turbo Codes

Rutuja A. Deshmukh, Dr. Ashish R. Panat
Research Scholar
Email: rutujadeshmukh88@gmail.com
Research Guide
Email: ashishpanat@gmail.com

***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

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.

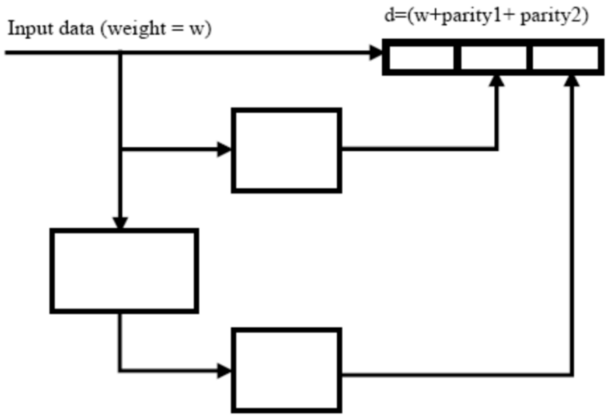


Fig. 1. Parallel concatenated turbo codes.

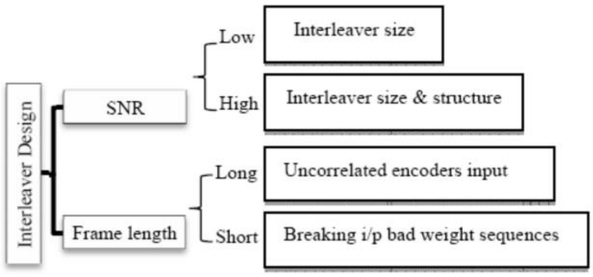


Fig. 2. Interleaver design dependence.

A random interleaver is simply a random permutation 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 configured as 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] adoptedhis 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.

II. CONSTITUENT ENCODER DESIGN

The use of a symbol interleaver implies that the constituent encoders should be optimized for “symbol-wise effective free distance.” This term refers to the minimum output distance when the input symbol sequence has exactly two symbols different from zero. The usual notion of effective free distance refers to the minimum output distance for a binary input Hamming distance of two. we use several variations of effective free distance. The superscript refers to the output distance, Hamming or Euclidean . The number in the subscript denotes the input weight, whether bit-wise or symbol-wise .We always imply squared Euclidean distance. For example, stands for the output squared Euclidean distance when the symbol-wise input weight is two.

Range of Encoders to Search

Without concatenation, searching for good trellis codes that maximize free distance requires examining only one code within each group of range-equivalent encoders. Two encoders are called *range-equivalent* if they have the same set of possible output sequences [10] (Forney’s notion of equivalence). So, it is sufficient to restrict attention within a set of canonical encoders, which are identified by Forney [11]. For turbo codes, the mapping from input to output sequences plays an important role. Range-equivalent codes can have quite different performance. For example, feedback encoders always have a range-equivalent feedforward encoder which would perform poorly with parallel concatenation.

Define as *input-Hamming-weight equivalent* encoders that map the same input weight error events to the same output distance. If two encoders are not input-Hamming-weight equivalent, we call them input-Hamming-weight distinct. When searching for constituent encoders that maximize effective distance, it is sufficient to examine all codes that are input-Hamming-weight distinct to each other.

III. ENHANCED BLOCK S-RANDOM (EBSR) INTERLEAVER

A. Designed Algorithm

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

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

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 Enhanced 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 EBSR 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:

$$r(i,j) = 1 + ik + j$$

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

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,..,*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 $y \in \{1\ldots l\}$ check the next condition:

$$\left|\pi_{S_1}(x) - \pi_{S_1}(y)\right| > S_1 \ , \ \ with\ |x-y| \le S_1$$

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 $S_2 < (k/2)^{1/2}$ and with column new positions satisfying:

$$\left|\pi_{S_2}(x) - \pi_{S_2}(y)\right| > S_2 \ , \ \ with\ |x-y| < S_2$$

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

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. 4. 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. 4 (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.

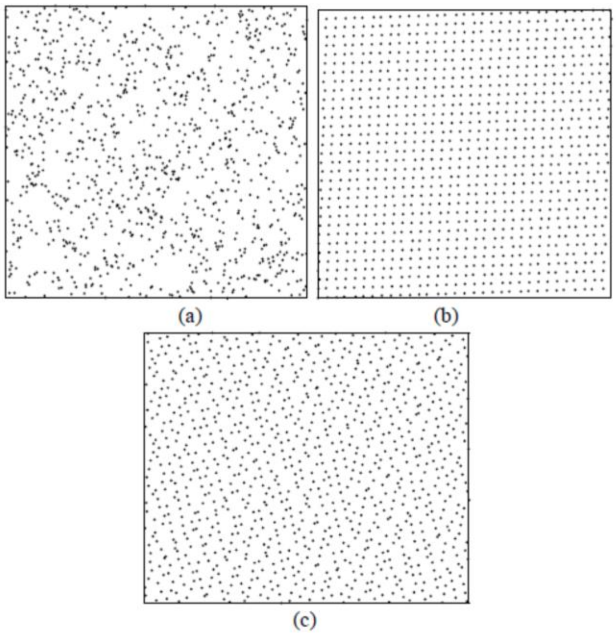


Fig. 4. Graphical representation of different interleavers of length N=1024 bits. (a) Uniform Random Interleaver, (b) Practical Size Interleaver and (c) EBSR Interleaver.

From Fig. 4 (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 bitsand the log-map decoder with 8-iterations algorithm are used. We compared the EBSR interleaver performance with different types of turbo code interleavers. The first interleaver is a Practical size
interleaver based on the algorithm in [6].

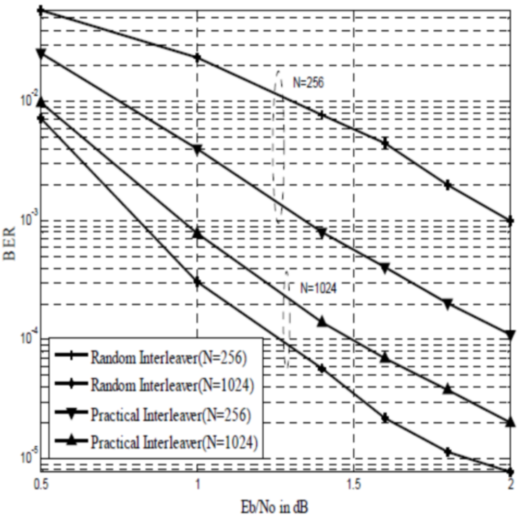
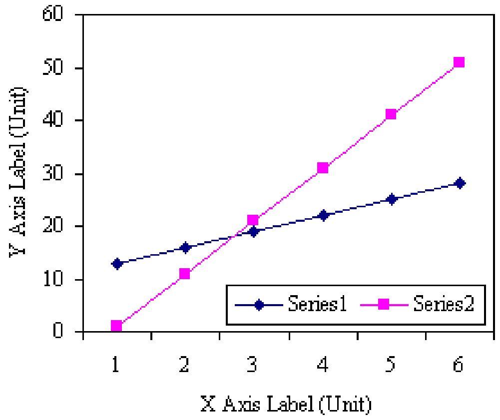


Fig. 5. BER performance comparison between 4-states rate 1/3 turbo code with Random and Practical interleavers at *N*= (256 and 1024) bits.

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, gives a good performance with very simple design, especially for long frames. For the simulated cases of these interleavers, a good performances are obtained at different block lengths
(*N*= 256, 400, 1024 and 2048) bits.

A. Figures and Tabl



Φιγυρε 1. Note how the caption is centered in the column.

REFERENCES

[1] Laddomada, M.; Scanavino, B., “A cost-function based technique for design of good prunable interleavers for turbo codes”, IEEE Transactions on Wireless Communications, Vol. 5, No. 8, pp. 1953 - 1958, 2006

[2] Snasel, V.; Platos, J.; Kromer, P.; Ouddane, N.; Abraham, A., “Interleaver optimization by population based metaheuristics”, 2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 672 - 677, 2010

[3] He, S.; Wu, Q.H. ; Saunders, J.R., “Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior”, IEEE Transactions on Evolutionary Computation, Vol: 13, No: 5, Page(s): 973 – 990, 2009

[4] Ajith Abraham, Pavel Krömer, Václav Snášel, Nabil Ouddane, "Evolving Turbo Code Interleavers by Genetic Algorithms ", in proceedings of International Conference on Complex, Intelligent and Software Intensive Systems, 2008.

[5] K.Prabhakar, Ch.Ravi Kumar, K.Padmaraju, "Interleaver optimization by particle swarm Algorithm ", International Journal for Technological Research in Engineering, Vol. 2, no.1, September-2014.

[6] Hokfelt J, Edfors O, Maseng T. Interleaver design for turbo codes based on the performance of iterative decoding. In: Proc IEEE ICC 1993. p. 93–7.

[7] Divsalar D, Pollara F. Turbo codes for deep-space communications. Jet propulsion lab, TDA progress report feb. 1995. p. 42–120.

[8] Perez C, Seghers J, Costello D. A distance spectrum interpretation of turbo codes. IEEE Trans Inf Theory 1996;42(2):1698–703.

[9] Dolinar S, Divsallar D. Weight distribution of turbo codes using random and nonrandom permutations. Jet propulsion lab, TDA progress report aug. 1995. p. 42–122.

[10] Sadjadpour HR, Sloane NJA, Salehi M, Nebe G. Interleaver design for turbo codes. IEEE J Sel Areas Commun 2001;19(5):831–7.

[11] Crozier N. New high-spread high-distance interleavers for turbo codes. In: Proc 20th biennial symp commum 2000. p. 3–7.

[12] Simona V. Halunga, Nicolae Vizireanu, Octavian Fratu, "Imperfect cross-correlation and amplitude balance effects on conventional multiuser decoder with turbo encoding", Digital Signal Processing, vol. 20, pp. 191–200, 2010.

[13] Muhammad Arif, Noor Muhammad Sheikh, Asrar U.H. Sheikh, "Design of two step deterministic interleaver for turbo codes", Computers and Electrical Engineering, vol. 34, pp. 368–377, 2008.

[14] Lucian Trifina & Daniela Tarniceriu, "Improved method for searching interleavers from a certain set using Garello’s method with applications for the LTE standard", annals of telecommunications - annales des télécommunications, Vol. 69, no. 5-6, pp 251-272, 2014.

[15] Horia Balta, Catherine Douillard and Radu Lucaciu, "Multi-non-binary turbo codes", EURASIP Journal on Wireless Communications and Networking, 2013.

[16] Mohd Azri Mohd Izhar, Norsheila Fisal, Xiaobo Zhou, Khoirul Anwar and Tad Matsumoto, "Exploitation of 2D binary source correlation using turbo block codes with fine-tuning", EURASIP Journal on Wireless Communications and Networking, 2013.

[17] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," IEEE Trans. Commun., vol. 44, no. 10, pp. 1261-1271, 1996.

[18] C. Berrou, R. Pyndiah, P. Adde, C. Pouillard, and R.L. Bidan, "An overview of Turbo codes and their applications," in The European Conference on Wireless Technology, Paris, Oct. 2005, pp. 1-9.

[19] F. Peper, "Nanocomputers," in Computational Complexity, New York: Springer, 2012, pp. 1998-2028.

[20] Design and Simulation of Turbo Encoder in Quantum-Dot Cellular Automata Zhang, M.; Cai, L.; Yang, X.; Cui, H.; Feng, C. Nanotechnology, IEEE Transactions on Year: 2015, Volume: 14, Issue: 5 Pages: 820 - 828, DOI: 10.1109/TNANO.2015.2449663

[21] 4. P Robertson, E Villebrun, P Hoeher, A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, in Proc. IEEE International Conference on Communications (ICC) (Seattle, USA, 1995), pp. 1009–1013

[22] 5. P Robertson, P Hoeher, E Villebrun, Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding. ETT 8(2), 119–125 (1997)

[23] 3. LR Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimizing symbol error rate. IEEE Trans. Inform. Theory IT-20(2), 284–287 (1974)

[24] AJ Viterbi, An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes. IEEE J. Sel. Areas Commun. 16(2), 260–264 (1998)

[25] Michal Sybis and Piotr Tyczka, “Reduced complexity Log-MAP algorithm with Jensen inequality based non-recursive max∗ operator for turbo TCM decoding”, EURASIP Journal on Wireless Communications and Networking 2013, 2013:238

[26] He, S.; Wu, Q.H. ; Saunders, J.R., “Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior”, IEEE Transactions on Evolutionary Computation, Vol: 13, No: 5, Page(s): 973 – 990, 2009

[27] Goldberg, David (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley Professional. ISBN 978-0201157673.

[28] Holland, John (1992). Adaptation in Natural and Artificial Systems. Cambridge, MA: MIT Press. ISBN 978-0262581110.

[29] Aloysius George, B. R. Rajakumar, D. Binu,(2012) "Genetic algorithm based airlines booking terminal open/close decision system". ICACCI '12 Proceedings of the International Conference on Advances in Computing, Communications and Informatics Pages 174-179

[30] Zhang, J., Lo, W.L., and Chung, H., "Pseudocoevolutionary Genetic Algorithms for Power Electronic Circuits Optimization", IEEE Trans Systems, Man, and Cybernetics, Part C., Vol.36, No.4, July 2006, pp. 590–598.

[31] W. J. O’Brien, B. I. Evans, and G. L. Howick, “A new view of the predation cycle of a planktivorous fish, white crappie (pomoxis annularis),” Canadian J. Fisheries Aquatic Sci., vol. 43, pp. 1894–1899, 1986

[32] Won Y. Yang, Yong S. Cho, Won G. Jeon, Jeong W. Lee, Jong H. Paik, Jae K. Kim, Mi-Hyun Lee, Kyu I. Lee, Kyung W. Park, Kyung S. Woo, MATLAB®/Simulink® for Digital Communication, A-Jin Publishing Co., Korea, 2009

[33] B.R. Rajakumar, A George, “A new adaptive mutation technique for genetic algorithm”, 2012 IEEE Computational Intelligence & Computing Research (ICCIC), 2012