

Performance Study of Turbo Code with Interleaver Design

Mojaiana Synthia, Md. Shipon Ali

Abstract— This paper begins with an investigation of the optimization of binary turbo code through good interleaver design. For this purpose, different types of interleaver have designed here and have evaluated their performance. Basically the performances of the turbo code using block interleaver, helical interleaver, random interleaver and odd-even interleaver have evaluated. From this investigation it has been seen that for small code length the performance of the block interleaver is superior in non-puncturing case and the performance of the odd-even interleaver is superior in puncturing case, but for large code length the performance of the random interleaver is better in both of puncturing and non-puncturing condition. In this paper, it has investigated that the performance of the odd-even interleaver (block interleaver with odd number of rows and columns) significantly increased in puncturing conditions.

Index terms— Turbo encoder; Turbo decoder; Interleaver; code rate; Puncturing; Nonpuncturing.

1. Introduction

Due to increasing demand for information exchange in modern civilization, the transfer of information from the source to its destination has to be done in such a way that the quality of the received information should be as close as possible to the quality of the transmitted information. Channel codes are used as an invaluable tool for this purpose. The channel encoder adds code bits to the transmission bit stream, based on the data bits at its input. These extra bits are reused by the channel decoder at the receiver to correct errors introduced into the transmission stream by a noisy or fading channel. But, there are two main disadvantages of error correcting codes. Firstly, the injection of extra bits into the transmission stream, thus increasing the bandwidth needed to transmit the signal. A second disadvantage is that they add complexity to the design of a communications system. In this paper some technique has introduced to overcome these problems.

2. Turbo Encoder

A general Turbo encoder is shown in Figure 2.1. It employs two identical systematic recursive Convolutional encoders connected in parallel with an interleaver preceding the second recursive convolutional encoder. The two recursive convolutional encoders are called the constituent encoders of the Turbo encoder. The information bits are encoded by both RSC encoders. The data frame, length of size N , inserts directly into the first encoder and after interleaving of length N , it feeds the second encoder. Therefore, N systematic bits can generate $2N$ parity bits and its gives a code rate of $1/3$. However, it can be removed some of the parity bits in order to increase the coding rate by means of puncturing

process. Furthermore, it is also possible to employ more than two component codes. However, in this paper it concentrates entirely on the standard turbo encoder structure using two RSC codes.

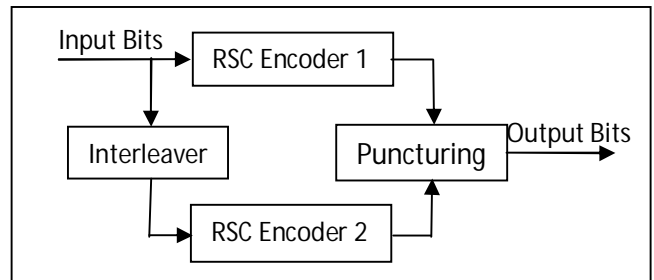


Fig (2.1): General Turbo Encoder.

3. Interleaver

The interleaver design is a key factor which determines the good performance of a turbo code. The output sequences of RSC encoder usually have high weight. However, some input sequences still cause the output sequence of RSC encoder to generate low weight codeword. Therefore, the interleaver scrambles the input sequence and generates randomness to the input sequence as a result high codeword can obtain. Some of the useful inter-leaver used in turbo code is discussed in the following sections.

A. Block Interleaver

Block interleaver is easy to implement in practice. This simplest interleaver is a memory in which data is written row-wise and read column-wise. It is also known as the "row-column" interleaver.

X1	X2	X3	X4
----	----	----	----

X5	X6	X7	X8
X9	X10	X11	X12
X13	X14	X15	X16
X17	X18	X19	X20

Table (3.A.1): Writing data row-wise in memory.

X1	X5	X9	X13	X17	X2	X6
----	----	----	-----	-----	----	----	-------

Table (3.A.2): Reading data column-wise from memory.

B. Helical Interleaver

A helical interleaver writes data into row-wise but reads data diagonal-wise. An example of helical interleaver is shown below:

X1	X2	X3
X4	X5	X6
X7	X8	X9
X10	X11	X12
X13	X14	X15

Table (3.B.1): Writing data row-wise in memory.

X13	X11	X9	X4	X2	X15	X10	...
-----	-----	----	----	----	-----	-----	-----

Table (3.B.2): Reading data diagonal-wise from memory.

C. Odd-Even Interleaver

An odd-even interleaver is a block interleaver in which the number of rows and columns must be odd numbers.

D. Random Interleaver

The most used turbo code interleaver is the random interleaver. It basically generates a mapping between the input and the output positions. It can be designed for arbitrary length of input bits. If the length of the input is N , the mapping set will have $N!$ combinations and chooses one combination to permute the input bits. A simple random interleaver to permute data is shown below:

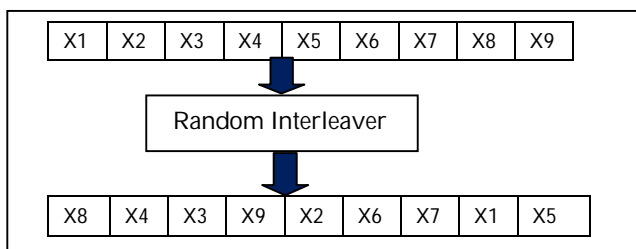


Fig (3.D.1): The random interleaver with length 10.

4. Turbo Decoder

In a turbo encoder with T constituent encoders, the encoder output contains a single systematic output and T parity outputs from the RSC encoders (assuming no puncturing), $T - 1$ of which operate on an interleaved version of original data block. Thus, the output of the turbo encoder can be viewed as the output of T

independent RSC encoders, except the systematic information only need be transmitted for one of the encoders. The decoder can reconstruct the systematic bits for the other encoders because it knows the interleaving patterns that were used. Thus, the decoder can be decomposed into T convolutional decoders with each one operating on the output of a single constituent encoder. In order to get the best possible estimate of the original message, these separate decoders must be able to share the results of their calculations. To accomplish this, turbo decoders use iterative feedback decoding. Figure 4.1 shows a schematic of a turbo decoder for the classical turbo code with $T=2$.

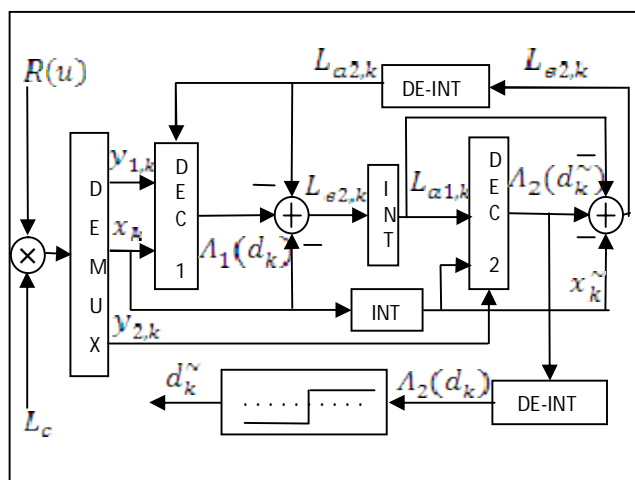


Fig (4.1): General Turbo Decoder.

This turbo decoder corresponds to the turbo encoder of figure 2.1. The first decoder uses the systematic information x_k , the output from the first constituent encoder $y_{1,k}$, and a priori information from the second decoder $L_{a2,k}$, to calculate soft estimates of the original data in the block known as a log-likelihood ratio (LLRs), Λ_1 . The systematic and a priori information are subtracted from Λ_1 in order to prevent positive feedback. What is left over is the new information calculated by the first decoder $L_{e1,k}$, known as the extrinsic information. This extrinsic information will be used as a priori information by the second decoder. The second decoder uses this a priori information along with the systematic information, and the output of the second constituent encoder $y_{2,k}$. However, $y_{2,k}$ was calculated from an interleaved version of x_k , so both the systematic information and extrinsic information from the first decoder must be interleaved (forming x_k^- and $L_{a1,k}$, respectively) before being used in the second decoder. The second decoder produces the extrinsic information $L_{e2,k}$ that is de-interleaved and then fed back to the first encoder to be used as a priori information $L_{a2,k}$. After the first decoding cycle has completed, the decoder is not taking in new inputs. Instead, it is iterating toward a

best estimate of the transmitted data using the received values that it now has stored in memory. After l iterations, the feedback loop is broken and the LLRs produced by the second decoder are de-interleaved one final time to put them in the same order as the original data block and a hard limiter makes the final bit decisions to produce the decoded block.

5. Simulation Result

In this paper, interleavers effects have investigated into two different conditions depend on transmission data rate. One is non-puncturing condition where all of parity bits are transmitted along systematic bits. And other is puncturing condition where some of parity bits are transmitted along systematic bits.

A. Effect in Non-puncturing Case

Figure 5.A.1, 5.A.2 and 5.A.3 shows the BER performance of the turbo code using random interleaver, helical interleaver and block interleaver; within non-puncturing situations. From figure 5.A.1, it is observed that the performance of both of the block interleaver and helical interleaver are better than the random interleaver for small code length input data. Again, it is also shown that for large code length input data the performance of random interleaver is better than others as figure 5.A.3. The performance of block interleaver and helical interleaver are almost same for any code length of input data.

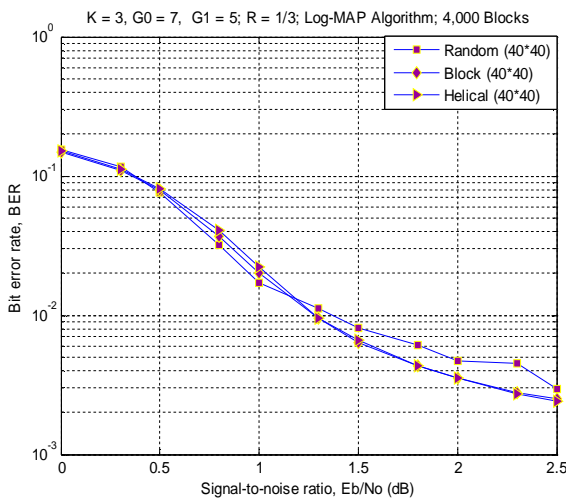


Fig (5.A.1): BER performance with different interleavers.

B. Effect in Puncturing Case

Bit puncturing is required in order to assure the efficient use of available bandwidth. Turbo code performance using random interleaver, helical interleaver and block interleaver within puncturing situations are shown in figure 5.B.1, 5.B.2 and 5.B.3.

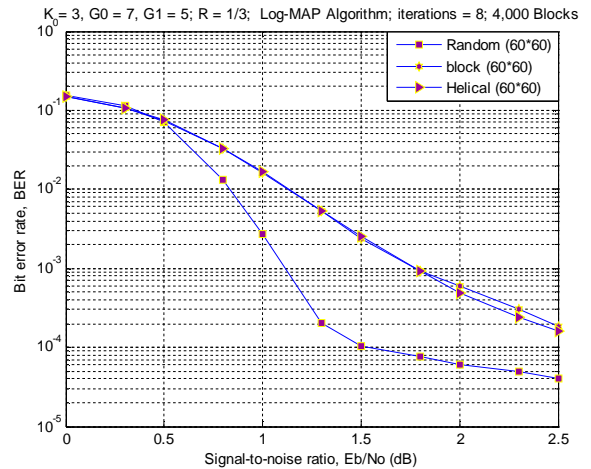


Fig (5.B.2): BER performance with different interleavers.

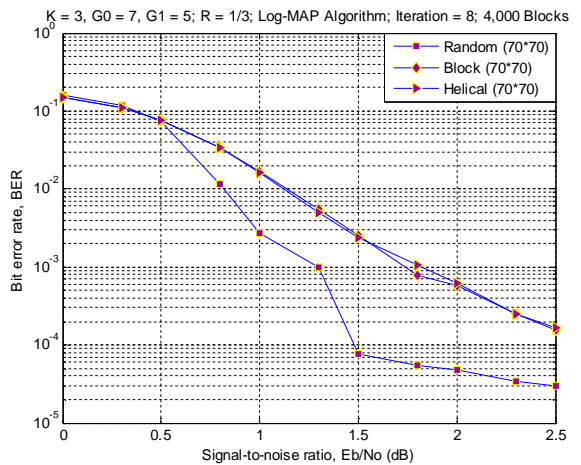


Fig (5.A.3): BER performance with different interleavers.

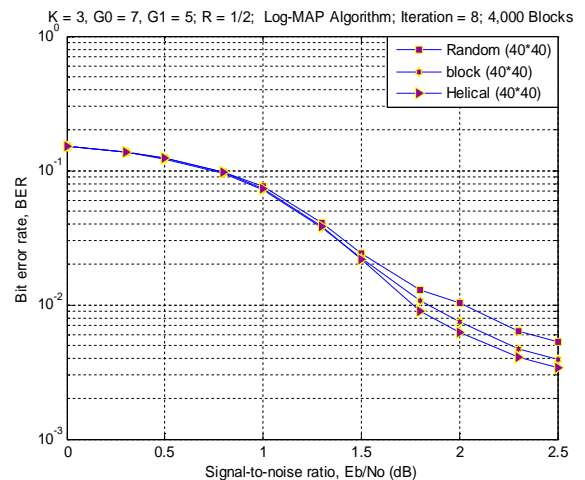


Fig (5.B.1): BER performance with different interleavers.

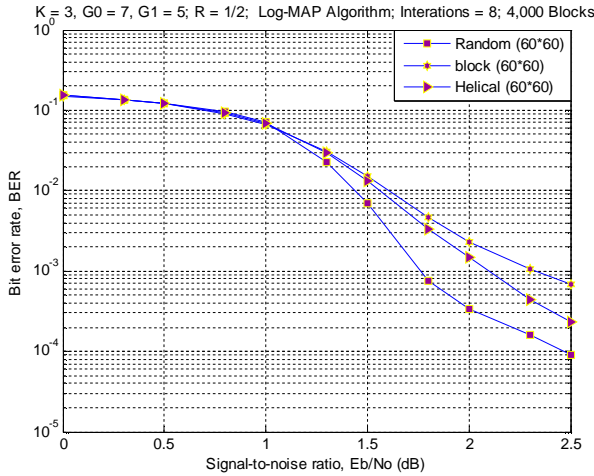


Fig (5.B.2): BER performance with different interleavers.

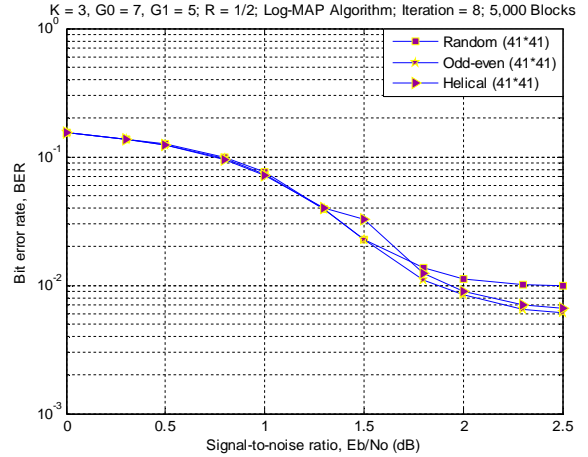


Fig (5.B.4): BER performance with different interleavers.

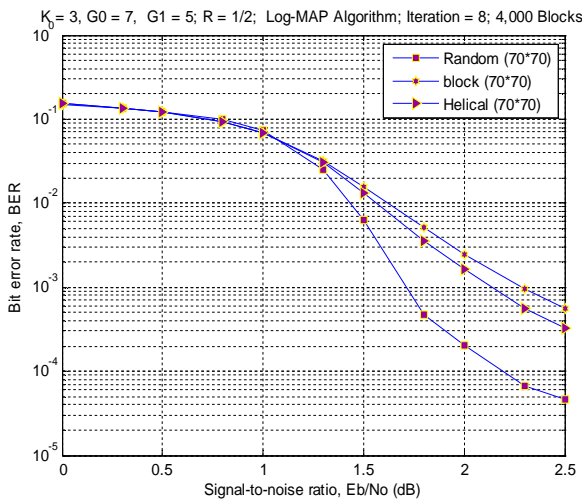


Fig (5.B.3): BER performance with different interleavers.

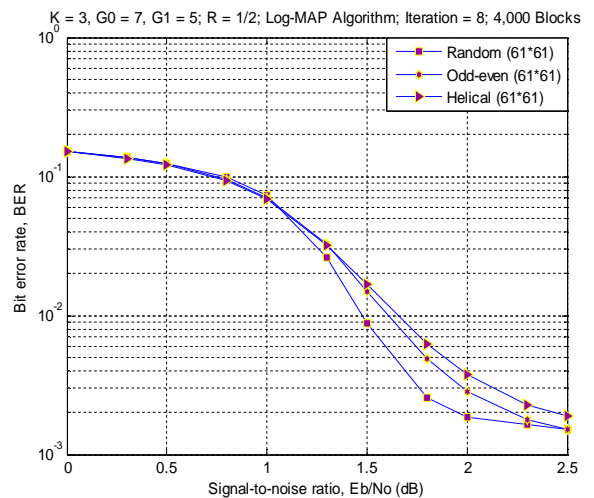


Fig (5.B.5): BER performance with different interleavers.

From these figures it is observed that the resultant performances are almost same as that obtained in non-puncturing conditions except that the performance of helical interleaver is better than the block interleaver.

In puncturing case, the performance of block interleaver significantly increased when its number of rows and columns are odd (odd-even interleaver). The performances of turbo code using odd-even interleaver comparison with the helical and random interleaver are shown in figure 5.B.4, 5.B.5 and 5.B.6. Also, it is observed that the performance of odd-even interleaver is better than the helical interleaver for any code length of input data. Again, for below of (60x60) code length the odd-even interleaver's performances are better than random interleaver. Also for large code length of input data the performance of the random interleaver is still better than others.

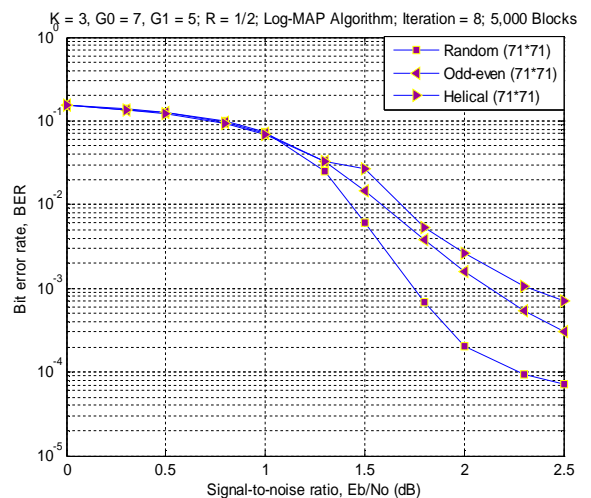


Fig (5.B.6): BER performance with different interleavers.

6. Result and Conclusion

In this paper, effects of interleavers have studied in details with both of the puncturing and non-puncturing case. From there it has seen that for non-puncturing case, the performance of the block interleaver and helical interleaver are almost same and better than the random interleaver at small code length data transmission. But, for large code length data transmission the random interleaver perform much better than both of the block interleaver and helical interleaver. In puncturing case, the performances of these interleaver's are almost same as that obtained in non-puncturing conditions except that the performance of helical interleaver is better than the block interleaver. But in puncturing case, that the performances of odd-even interleaver is better than the helical interleaver. This improvement is occurred because it can only possible to transmit at least one parity bit for each systematic bit by using odd-even interleaver. However, for below of (60×60) code length the odd-even interleaver's performances are better than the performance of random interleaver.

REFERENCES

- [1] Robert H. Morelos-Zaragoza, "The Art of Error Correcting Coding," SONY Computer Science Laboratories Inc, Japan, 2002.
- [2] Jason R. Hess, "Implementation of a Turbo Decoder on a Configurable Computing Platform," Virginia Polytechnic Institute and State University, USA, 1999.
- [3] Alain Glavieux, "Channel Coding in Communication networks From Theory to Turbo codes," ISTE Ltd, USA, 2007.
- [4] T. E. Hunter, "Coded Cooperation: A New Framework for User Cooperation in Wireless Networks," Ph.D. dissertation, The University of Texas at Dallas, May, 2004.
- [5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-correcting Coding and Decoding: Turbo-Codes," the International Conference on Communications, (Geneva, Switzerland), pp.1064 in1070, May, 1993.
- [6] S. Le Go, A. Glavieux, and C. Berrou, "Turbo-Codes and High Spectral Efficiency Modulation," Proceedings of IEEE international conference on communications, New Orleans, LA. pp. 645-649, May 1-5, 1994.
- [7] Joachim Hagenauer, "Iterative Decoding of Binary Block and Convolutional Codes," IEEE Transactions on Information Theory, Vol. 42, No. 2, March 1996.
- [8] Mark Bingham, "Symbol-Based Turbo Codes for Wireless Communications," Electrical and Computer Engineering Discipline, Waterloo, Ontario, Canada, 2002.
- [9] K. V. Ravi and Tam Soh Khum, "Performance of Turbo TCM in Wideband CDMA Applications," Department of Electrical Engineering, National University of Singapore, 2000.
- [10] D. Gnaedig, E. Boutillon and M. Jezequel "Design of Three-Dimensional Multiple Slice Turbo Codes," EURASIP Journal on Applied Signal Processing, 2005.
- [11] Fu-hua Huang, "Iterative Turbo Code. Decoder" John Wiley & Sons, New York, 2000.
- [12] Bernard Sklar, "Digital Communications: Fundamentals and Applications," Second Edition, 2001.

- [13] Matthew C. Valenti and Jian Sun, "Handbook of RF and Wireless Technologies," John Wiley & Sons, New York, 2003.
- [14] Salma Ben Jamaa, Michel Kieffer and Pierre Duhamel, "Exact MAP Decoding of Cabac Encoded Data," University of Paris, France.
- [15] George White "Optimised Turbo Codes for Wireless Channels," Ph.D. Thesis, Department of Electronics, University of York, UK, October, 2001.
- [16] Jinghu Chen, Marc P. C. Fossorier and Shu Lin, "Bi-directional SOVA Decoding for Turbo-codes," Department of Electrical Engineering, University of Hawaii, USA.

- Author1: Mojaiana Synthia, Electronics & Communication Engineering Discipline, Khulna University, Bangladesh.
- Author2: Md. Shipon Ali, Grameenphone Limited, Dhaka, Bangladesh