International Journal of Latest Trends in Engineering and Technology
Vol.(7)Issue(3), pp. 124-130
DOI: http://dx.doi.org/10.21172/1.73.517
e-ISSN:2278-621X

Turbo Code using a Novel Hybrid Interleaver

Jaya Krishna Sunkara[1], S. Sri Manu Prasad[2] and Dr. T. Venkateswarlu[3]

Abstract- Turbo code in an attempt to realize a technique that approaches the theoretic limit utilizes a crucial component, interleaver, which converts burst errors caused by impulsive noise to simple errors. But the processing of data through the interleaver incurs time as well as memory. Both time and memory complexities are directly related to the total number of information bits. The time required for interleaver processing is so high because of which some applications omits the complete interleavers. But the consequent schemes suffer from high bit error rate. In this paper a novel hybrid two stage interleaving scheme was proposed which reduces the time required for interleaving processing while maintaining the bit error rate criteria up to the levels obtained with block or 3G Partnership Project interleavers.

Keywords – Interleaver, turbo code, block interleaver, 3GPP interleaver.

I. INTRODUCTION

The main function of a communication system is to transmit information from the source to the destination with sufficient reliability. In the last two decades, there has been an explosion of interest in the transmission of digital information mainly due to its low cost, simplicity, higher reliability and possibility of transmission of many services in digital forms [1]. Theoretically, Shannon stated that the maximum rate of transmitted signal or capacity of a channel over Additive White Gaussian Noise (AWGN), with an arbitrarily low bit error rate depends on the SNR and the bandwidth of the system (W), according to [2]:

$$C = W \log_2\left(1+\frac{S}{N}\right) \quad (1)$$

here C is the channel capacity, S/N signal power to noise power ratio, respectively. Based on this theory, it would be possible to transmit information with any rate (R) less than or at best equal to the channel capacity ($R \leq C$), when suitable coding is applied. Instead of S/N, the channel capacity can be represented based on the signal to noise ratio per information bit (E_b/N_0). Considering the relationship between SNR and E_b/N_0, and the channel capacity (with value R), equation (1) can be rewritten as follows:

$$\frac{S}{N} = \frac{E_b}{N_o} X \frac{R}{W} \quad (2)$$

$$\frac{C}{W} = \log_2\left(1+\frac{E_b}{N_o}.\frac{R}{W}\right) \quad (3)$$

In the case of an infinite channel bandwidth ($W \rightarrow \infty$, $C/W \rightarrow 0$) the Shannon bound is defined by:

$$\frac{E_b}{N_o} = \frac{1}{\log_2 e} = 0.693 \quad (4)$$

[1] PG Scholar, Dept. of ECE, SVU, Tirupati, INDIA
[2] Project Manager, ATD, Samsung R&D Institute India, Banglore, INDIA
[3] Professor, Dept. of ECE, SVU, Tirupati, INDIA

In order to achieve this bound, i.e. E_b/N_0 = -1.59 dB value, it would be necessary to use a code with such long length that encoding and decoding would be practically impossible. However, the most significant step in obtaining this target, was by Forney, who found that a long code length could be achieved by concatenation of two simple component codes with short lengths linked by an interleaver [3]. Conventionally, a turbo code is analyzed as a block code by using a block interleaver and terminating RSC encoders to a known state at the end of each data block.

Codes with this structure are generally decoded using an iterative decoding technique. Depending on the number of iterations, the decoding procedure can approach the optimum decoding. Similarly in block codes, the error probability for turbo codes in the case of additive white Gaussian channel is upper bounded and is given below [4]:

$$P_b \leq \sum_{d=d_{free}}^{\infty} \sum_{w=1}^{L} a(w,d) \frac{w}{2L} erfc\left(\sqrt{d \frac{2RE_b}{N_o}}\right) \quad (5)$$

where R, w, d, L, $a(w,d)$ and d_{free} denote the code rate, the information weight, the codeword weight, the information length, total number of codewords with weight w and minimum as the free distance value, respectively.

In this equation, $a(w,d)$ determines multiplicity of the calculated weights [4]. From the above the inference is that high free distance codes with less multiplicity will have the maximum error bound at less range and hence the performance will be good. Study on turbo codes specifies that block interleaver based turbo codes have low free distance as well as with high multiplicity. This affects the performance of overall turbo codes to a great extent particularly average to higher SNRs. The scenario where the error performance is poor at middle to high SNRs is referred as 'Error Floor' in the literature.

One of the effective solutions to lessen the error floor effect is utilization of a suitable interleaver compatible with the structure of constituent RSC encoders. In this case, input bit streams, which produce low weights for the first RSC code are permuted by an interleaver in a way that prohibits generation of low weights for the second RSC code to increase free distance value of the turbo code. It has been accepted that the best performance is achieved by random interleavers [5]. But the implementation of such an interleaver where the interleaving takes place on a random basis and the de-interleaving should also takes place in the same random basis is very difficult. In addition to implement and use same random structure in interleaving and de-interleaving these have to be synchronized to some extent for which the input stream of bits may be required to be stored. This is really not anticipated in some cases, where the length of the bit stream is so large [6].

The block interleavers incur more memory so as to maintain the shifting order. But non-block interleavers are designed with less memory and also have self-synchronization property with their de-interleavers so that the complexity of non-block interleavers is less. To utilize the above features of non-block interleavers in turbo codes, these interleavers need to operate as block interleavers. This can be simply accomplished if few bits are injected at the end of every data block driving the interleaver memories to the known state. In addition to utilizing advantages of non-block interleavers, this process eases the coding and decoding analysis.

The problem with this technique is that the injected bits consume some useful bandwidth which doesn't convey any information. To reduce the number of the bits, few optimizations may be done to the interleaver arrangement. A number of studies were undertaken on the importance of interleaver on the turbo code performance [7]-[9] and a number of interleavers are presented in the literature with diverse features and complexities [10] – [13].

The interleaver which is basically introduced to scramble the order of information bits takes enormous amount of time in cases. When the complexity of the interleaver is very high, this high complexity should result in less error probability and usually takes more time for execution and when

the complexity is less then may be the time for execution is less and error probability will be high. The otherwise cases will be either worst or ultimate. In the case of block interleaver also when the execution time is less then the capacity of converting burst errors into simple errors is less and consequently the error probability is high. In this paper a novel scheme was presented which aims to reduce the time complexity of the interleaving process to great extent while maintaining a similar error performance.

The rest of the paper is organized as follows. Block and 3GPP interleavers are explained in section II. The proposed hybrid interleaver was presented in section III. Experimental results are presented in section IV. Concluding remarks are given in section V.

II. INTERLEAVERS

A. *Block Interleaver*

Block interleaver as mentioned earlier rearranges the input bit stream. The block interleaver doesn't omit any bit in the sequence as well as it doesn't repeat any of the bits. Block interleavers can be classified into mainly three categories. They are Matrix interleaver, Random interleaver and Algebraic interleaver. The time complexity associated with block interleaver is (2MN – 2M +2), when the input bits are arranged as a matrix of order MXN. From the expression it is obvious that if the number of rows is large than that of columns the time complexity is very less. But in this case the capability of the interleaver in converting the burst error to simple error is limited to few bits. When the number of rows is less than that of columns then the time complexity is high and at the same time the length of the burst that can converted to simple error is high.

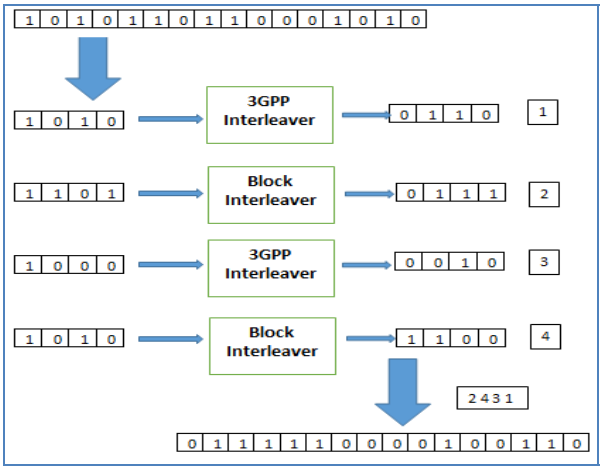

Fig. 1 *Block diagram of Hybrid interleaving scheme*

B. *3GPP Interleaver*

The Turbo code internal interleaver involves bits-input to a rectangular matrix, intra-row and inter-row permutations of the rectangular matrix and bits-output from the rectangular matrix. The bits input to the interleaver are denoted by $i_1, i_2, i_3, \ldots i_K$, where K is the integer number of the bits and takes one value of $K \leq 5114$. The relation between the bits input to the Turbo code internal interleaver and the bits input to the channel coding is defined by $i_k = o_{irk}$ and $K = K_i$. The bit sequence $i_1, i_2, i_3, \ldots i_K$ input to the Turbo code internal interleaver is inscribed into the rectangular matrix. After the bits-input to the R×C rectangular matrix, the intra-row and inter-row permutations for the R×C rectangular matrix are done.

The output of the interleaver is the bit sequence read out column by column from the intra-row and inter-row permuted R × C rectangular matrix starting with bit y'_1 in row 0 of column 0 and ending with bit y'_{CR} in row R - 1 of column C - 1. The output is pruned by deleting dummy bits that were padded to the input of the rectangular matrix before intra-row and inter row permutations, i.e. bits y'_k that corresponds to bits y_k with k > K are removed from the output. The bits output from Turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_K$, where x'_1 corresponds to the bit y'_k with smallest index k after

pruning, x'_2 to the bit y'_k with second smallest index k after pruning, and so on. The number of bits output from Turbo code internal interleaver is K and the total number of pruned bits is: $R \times C - K$.

III. HYBRID INTERLEAVER

In the previous sections block and 3GPP interleavers are described. Now let us take another look at block interleavers. Consider matrix interleaver for the time and memory complexity analysis. Assume the incoming flow of data bits are framed as MXN array. The values of M and N depend on the expected size of burst error. The memory requirement at the interleaver stage will be MN bits (assuming bits as reference). But overriding of bits on filling and emptying can't be allowed, hence a memory of 2MN bits is required; MN bits of memory when writing the bits on MN array and separate MN bits which will be filled by reading from previous MN array.

Similar memory requirement is needed at the de-interleaver as well. Hence a total of 4MN bits of memory is required for MN bits of data. The time complexity is crucial thing in fast communications. Some of the practical installations omit the interleaving and de-interleaving in order to reduce the latency introduced by the interleaver pair. The minimum end-to-end delay due to the block interleaver is (2MN-2M+2). The other interleaver considered is 3GPP interleaver which is basically the improved matrix based block interleaver suggested by Third Generation Partnership Project. The time and memory constraints are close to that of standard block interleaver. When the number input frames is high then the complexity of both the interleaving schemes becomes similar because in 3GPP interleaver in addition to standard matrix writing and reading there are some additional operations which are somehow scalar operations. Now if the total bits in input frame are split into subgroups and if the individual groups are processed using the normal interleavers the time complexity can be reduced. Let the input frame is split into 4 groups. So the total number of input bits to individual interleaver becomes MN/4. They can be organized using M/2 x N/2 matrix. Hence the time complexity of individual interleaver becomes (MN/2 – M +2).

The total time complexity becomes 4*(MN/2 – M +2). But when the total input bits are split into four groups and applied to separate interleavers, the four groups of input bits are concentrated in that region only. That means the first quarter of bits is placed again in the first quarter only, similarly the remaining quarters. Hence a second stage of interleaving is proposed in which these quarters are interleaved based on a direct assignment without using another interleaver. Hence this scheme can be regarded as a two stage interleaving scheme. In this paper block interleavers are used and any other interleavers and any other combinations can also be considered. The structure of the proposed interleaver is shown in figure 1. From the above discussion it is clear that by splitting the input frame into four parts the execution time will be reduced greatly. The table 1 gives the comparison between the above interleavers theoretically.

Table 1: Theoretical Time Calculations for MN=1024

MN	M	N	Block Interleaver	Hybrid Interleaver
1024	2	512	2046	2048
	4	256	2042	2040
	8	128	2034	2024
	16	64	2018	1992
	32	32	1986	1928
	64	16	1922	1800
	128	8	1794	1544
	256	4	1538	1032
	512	2	1026	8

IV. Simulation Results

In this section the simulation results of the turbo codes with four varieties of interleavers are presented. The four varieties are block interleaver, 3GPP interleaver, no interleaver and the proposed hybrid interleaver. The input frame sizes are varied from 4 to 1024 and in this section input frames sizes 16, 64, 256 and 1024 bits are considered. First of all consider input frame size of 16. Note that the decoding algorithm used throughout this work is Max-log-MAP algorithm [14-17]. The BER vs E_b/N_o performance for input frame size of 16 bits with different interleavers is shown below in figures 2 and 3.

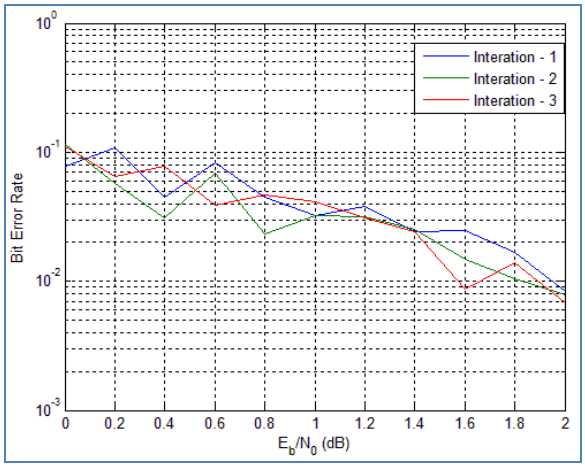

Fig. 2 BER Vs. E_b/N_o, 3GPP Interleaver, Input Frame size = 16 bits

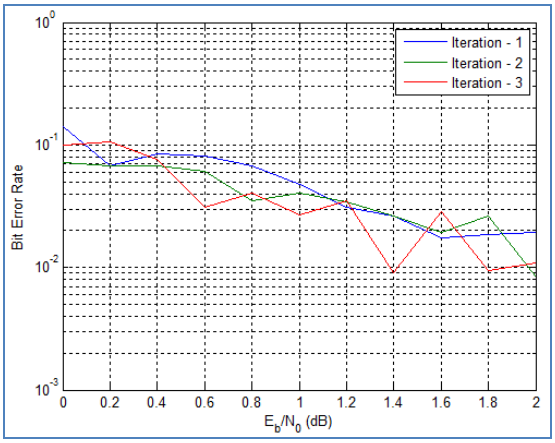

Fig. 3 BER Vs. E_b/N_o, Hybrid Interleaver, Input Frame size = 16 bits

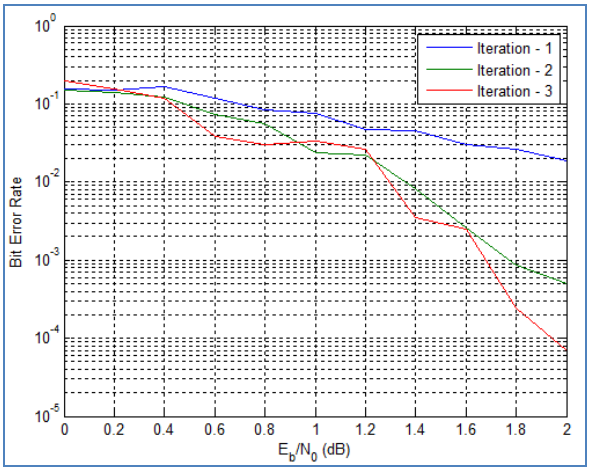

Fig. 4 BER Vs. E_b/N_o, 3GPP Interleaver, Input Frame size = 1024 bits

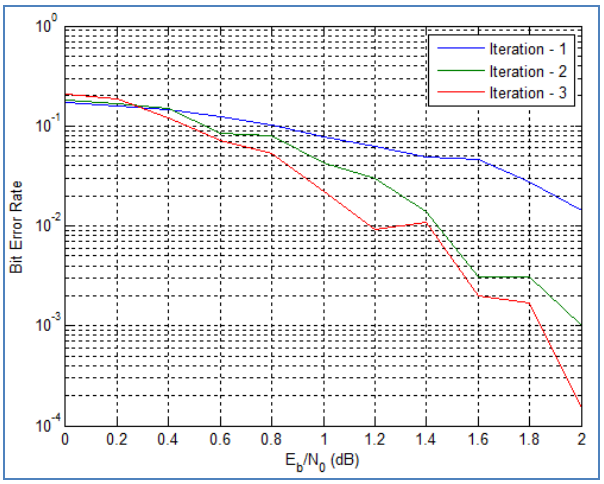

Fig. 5 BER Vs. E_b/N_o, Hybrid Interleaver, Input Frame size = 1024 bits

The performance of turbo code with different interleavers seems to be almost similar; the simulation time is also in the same lines ranging from 8.28 seconds in the case of no interleaver to 14.71 seconds in the case of hybrid interleaver. The purpose of interleaver is to reorder the information bits so that the signal to noise ratio is high as the interleaver converts the burst error caused by impulsive noise to simple errors. Hence use of interleaver is mandatory. But to check the performance the no interleaver case is considered.

In the above case where the frame length is 16 bits, the BER performance of turbo code with no interleaver is similar to that of other cases as the chance of occurrence of impulsive noise increases as the length increases, hence the effect of impulsive noise in no interleaver case prevails when the frame length is high. Now consider the frame length to be 1024 bits. The BER vs E_b/N_o performance was given in the figures 4 and 5. Now, as thought in the last paragraph the BER with no interleaver was very high compared to all other cases, i.e., with interleaver. The presence of interleaver has lowered the BER. Now the question is how much does this interleaving effect on the total time of execution? The interleaving has a serious effect on the total time required to transfer the data. In some applications in order to enable fast communication interleaver is dropped owing to high error rate. The interleaving scheme proposed in this paper lowers the time required for coding and decoding. Table 5 gives the time required for completion of interleaving and de-interleaving in turbo codes. From the table it is evident that the time required in case of hybrid interleaver is comparable to that of no interleaver case.

Table 2: Time required for interleaving and de-interleaving in seconds

	16-Bits	64-Bits	256-Bits	1024-Bits
No Interleaver	8.287	6.635	10.601	38.065
Block Interleaver	13.55	19.755	102.867	1105.41
3GPP Interleaver	11.885	14.576	753.50	4305.97
Hybrid Interleaver	14.715	11.424	24.171	148.216

V. CONCLUSIONS

In this paper an attempt has been made to reduce the time required for turbo coding and decoding in the presence of interleaving. Absence of interleaver, block interleaver and 3GPP interleavers are considered. A novel hybrid two stage interleaver was presented. The first stage of the interleaver deals with splitting of input frame and interleaving individual parts there off. The second stage deals interleaving complete parts of first stage as a single atomic block. The absence of interleaver presents a

serious escalation in BER particularly for long frames and incurs very low time. Block and 3GPP interleavers lower the BER because of the interleaving but take some significant amount of time for its operation to take place. At times, for a 1024 bit long frame the 3GPP interleaver takes about 4300 seconds and block interleaver spends more than 1100 seconds whereas the proposed hybrid interleaver takes about 150 seconds for the same length frame.

REFERENCES

[1] Benedetto, S., Bieglieri, E.: 'Principles of digital transmission with wireless applications', *Kluwer academic/Plenum publishers*, 1999.

[2] Shannon. C.E.: 'A mathematical theory of communication', *Bell System Technical Journal*, vol. 27, pp. 379–423,623–656, 1948.

[3] Forney, G.D.: 'Concatenated codes', *MIT Press*, 1966.

[4] Seghers, J.: 'On the free distance of turbo codes and related product codes', *Swiss Federal Institute of Technology, Zurich, Switzerland, Diploma project 6613*, vol. 42, August 1995.

[5] Dolinar, S., Divsalar, D.: 'Weight distributions for turbo codes using random and nonrandom permutations', *TDA Progress Report*, pp. 56–65, 15, August 1995.

[6] Sun, J., Takeshita, O.Y.: 'Interleavers for turbo codes using permutation polynomials over integer rings', *IEEE Transactions on Information Theory*, no. 1, pp. 101 – 119, January 2005.

[7] Atousa, Mohammadi, H.S., Zhuang, W.: 'Variance of the Turbo Code Performance Bound Over the Interleavers', *IEEE Transactions on Information Theory*, Vol. 48, No. 7, July 2002.

[8] Prabhavati, Bahirgonde, D., Shantanu, Dixit, K.: 'BER Analysis of Turbo Code Interleaver', *International Journal of Computer Applications*, vol. 126, no.14, September 2015.

[9] Maan, Kousa, A.: 'Performance of Turbo codes with Matrix Interleavers', *The Arabian Journal for Science and Engineering*, vol. 28, no. 2B, 2003.

[10] Ching-Lung Chi.: 'Quadratic Permutation Polynomial Interleavers for LTE Turbo Coding', *International Journal of Future Computer and Communication*, vol. 2, no. 4, August 2013.

[11] Mazumdar, A., Chaturvedi, A.K., Banerjee, A.: 'Construction of Turbo Code Interleavers from 3-regular Hamiltonian Graphs', *IEEE Communications Letters*, vol. 10, no. 4, April 2006.

[12] Jing Sun, Takeshita, O.Y.: 'Interleavers for Turbo Codes Using Permutation Polynomials Over Integer Rings', *IEEE Transactions On Information Theory*, vol. 51, no. 1, January 2005.

[13] Yuan, J., Vucetic, B., Feng, W.: 'Combined Turbo Codes and Interleaver Design', *IEEE Trans. on Commns*, vol. 47, no. 4, 1999.

[14] Maniraj Kumar, P., Sutha, S., Krishna Mourthy, V.: 'Mathematical Implementation of MAX LOG MAP Algorithm for Low Power Applications in Turbo Decoders', *IJAREEIE*, vol. 3, iss. 5, May 2014.

[15] Deborah Roseline, P., Thavaseelan, G., Rani Hemamalini, R.: 'Design of Hybrid Log-MAP Algorithm for Turbo Decoding Using QPP Interleaver', *Middle-East Journal of Scientific Research (Sensing, Signal Processing and Security),* 23, 274-279, 2015.

[16] Ashwini, S., Maniraj Kumar, P., Umayal, S., Sutha, S.: 'Area Efficient of Max Log Map Algorithm Using SB/DB Methods for Turbo Decoder', *International Journal of Scientific Engineering and Technology,* vol.3, iss. 6, pp : 775-780, June 2014.

[17] Sadjadpour, H.R.: 'Maximum A Posteriori Decoding Algorithms For Turbo Codes', *Proceedings of SPIE*, vol. 4045, 2000.