PERFORMANCE COMPARISION OF TURBO CODES AND MODIFIED TURBO CODES WITH DIFFERENT RATE

Sumit Kumar, Hemant Dalal

Abstract— **In wireless networks, Turbo Convolutional Codes are widely used to reduce Bit Error Rate (BER). However, Turbo codes have require large decoding complexity. But encoding complexity is larger in Turbo codes. Modified Turbo codes are low complexity turbo codes require 2-dimensional interleavers with large spreading factor. However, since interleavers used in turbo codes interleave bit positions in one dimensional array. In this paper, it is shown that BER of turbo codes is improved using interleavers. Moreover, MTC decoders require less computations than those of Turbo codes and improves the performance of the system in terms of BER.**

Index Terms— Decoding complexity, Modified Turbo Codes, Bit error rate.

I. INTRODUCTION

In wireless communication systems Forward Error Correcting codes (FEC) play an important role. in wireless communication networks, it is desirable to have FEC with low BER and low decoder complexity for reliable data transmission. Block codes, Convolutional codes [1] and TCC [2] are included in standards for 2G wireless networks. For the given implementation complexity, convolutional codes and TCC outperform block codes. Recently, TCC and LDPC codes have been introduced in standards for 3G wireless networks. Turbo convolutional code (TCC) are excellent errorcorrecting codes. Rate-1/2 TCC with interleaver length of 65 536, achieve near Shannon capacity performance [3]. However, TCC decoders are computationally complex [4]. It has been shown that modified turbo codes (MTC) using concatenations of relatively simple constituent convolutional and block codes achieve bit error rate (BER) performance close to that of TCC at computationally reduced complexity. Turbo codes play a major role in the channel coding scheme used in communication system like wireless communication. It is due to their exceptional performance that turbo codes are being accepted as 3GPP standard in personal communications. In next era of wireless communications, mainly the 4G applications, there is a need to provide the best QOS (Quality of Service) provisioning. For certain type of transmission like text transmission, the packet loss is intolerable while delay is acceptable. But for real time video, there can be an acceptable degradation in the video, but delay in the system cannot be accepted. Turbo codes are the most adaptable error coding scheme used to adapt to the varying QOS requirement. Modified turbo codes (MTC) relatively simple constituent convolutional and block codes achieve bit error rate (BER) performance close to that of TCC at reduced complexity [5]. Recently, low complexity hybrid turbo codes (LCHTC) a class of MTC is comparable with that of TCC BER performance and at significantly reduced decoding complexity. The modified version of LCHTC, called as improved low complexity hybrid turbo code (ILCHTC) has better error convergence than LCHTC with comparable decoding complexity [6]. But the decoding complexity of LCHTC and ILCHTC is about 50% less than that of TCC. Turbo codes can be achieved by serial and parallel concatenation of two or more codes called the constituent codes using interleaver between them so that data sequence for the two encoders is different [7]. The constituent codes can be either block codes or convolutional codes. Simply,

A turbo code is formed from the parallel concatenation of two codes separated by an interleaver.

But, currently most of the work on turbo codes has essentially focused on Convolutional turbo codes (CTC). A classical turbo code consists of parallel concatenation of two binary recursive systematic convolutional (RSC) codes and separated by a permutation interleaver as shown in Fig.1 [8]. Serial concatenation is also possible. RSC codes are a key component of Turbo Codes. They are based on Linear Feedback Shift-Registers (LFRS) and act as pseudorandom scramblers.

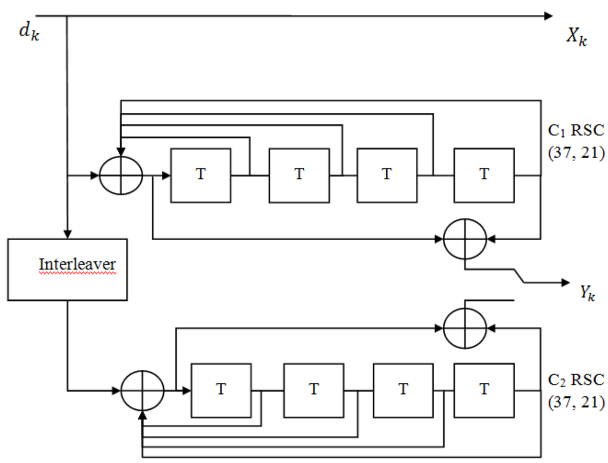

Fig. 1 The classical turbo codes

Manuscript received April, 2014.
Sumit Dalal, *Electronics and communication department, CBS Group of institutions, Jhajjar, Haryana, INDIA.*
Hemant Dalal, *Electronics and communication department, CBS Group of institutions, Jhajjar, Haryana, INDIA.,*

ISSN: 2278 – 7798

The input information bits feed the first encoder and, after having been interleaved by the interleaver, enter the second encoder. The codeword of the parallel concatenated code consists of the input bits to the first encoder followed by the parity check bits of both encoders. The trellis structure is identical for the RSC code and the NSC code and these two codes have the same free distance d_f However, the two output sequences (X_k and Y_k) do not correspond to the same input sequence (d_k) for RSC and NSC codes. This is the main difference between the two codes. When punctured code is considered, some output bits (X_k or Y_k) are deleted according to a chosen puncturing pattern defined by a matrix.

Interleaving is a process of rearranging or reordering of a data sequence in a one to one deterministic format. The interleaver is a one to one mapping function. The inverse of this process is called deinterleaving which restores the received sequence to its original order. Interleaving is a practical technique to enhance the error correcting capability of coding.

The interleaver can be represented in many ways by a vector π, suppose the information bit sequence can be represented by the equation as:

$$d = (d_0, d_1, \ldots, d_{k-1}) \quad \ldots(1)$$

Then the interleaved bit sequence can be represented as:

$$d_\pi = (d_{\pi 0}, d_{\pi 1}, \ldots, d_{\pi(k-1)}) \quad \ldots(2)$$

Here K represents the number of symbols in the information sequence d. This also means that the symbol at position i in information sequence d is interleaved to position $\pi-1\ (i)$ in d_π, where $\pi-1$ is the de-interleaver that acts on the interleaved information sequence and restores the sequence to its original order.

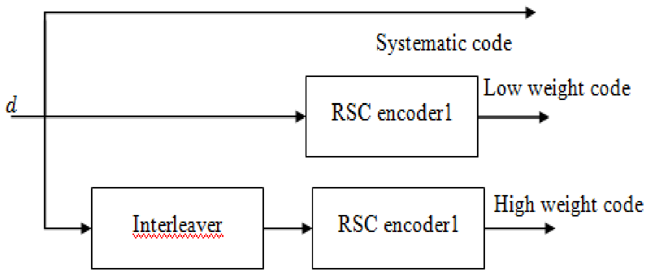

Fig. 2 Interleaver for Turbo Codes.

Interleaver serves two purposes. Firstly, if the input to the second encoder is interleaved, its output is usually quite different from the output of the first encoder. This reduces the chances of producing output with low weight. Higher weight means better decoder performance. From Figure 2, the input sequence d produces a low-weight recursive convolutional code sequence for RSC Encoder 1. To avoid having RSC Encoder 2 produce another low-weight recursive output sequence, the interleaver permutes the input sequence d to obtain a different sequence that produces a high-weight recursive convolutional code sequence. Thus, the turbo code's code weight is moderate. Second the code is a parallel concatenation of two codes, iterative encoding can be employed. Second encoder input is interleaved so it will produce different output sequence than the first encoder. This is the main trick for achieving near theoretical limit performance for the turbo codes.

II. Modified Turbo Codes

Turbo code achieves near Shannon's bound performance. CTC is relatively powerful in terms of error correction, one must be very careful when decoding concatenated codes to achieve near optimum decoding performance. To implement such SISO decoder one would need to implement a decoder with infinite complexity or close. Decoder complexity of TCC does not reduce even if puncturing is used to adjust the code rate. Also, the decoder complexity increases for wireless channels as Turbo codes require stronger constituent codes and low code rates. These things encourage us to construct low complexity block turbo codes.

It has been shown that concatenated coding schemes, using relatively simple constituent Convolutional and Block codes, can achieve performance close to the theoretical limits and requires low decoding complexity and are termed as *Modified Turbo Codes* (MTC). The MTC solution is more attractive for a wide range of applications. MTC can be implemented using three basic ideas given as follows:

1. The utilization of block codes instead of commonly used non-systematic or systematic convolutional codes.
2. The utilization of soft input soft output decoding. Instead of using hard decisions, the decoder uses the probabilities of the received data to generate soft output which also contain information about the degree of certainty of the output bits.
3. Encoders and decoders works on permuted versions of the same information. This is achieved by using an interleaver.

Modified turbo code consists of concatenation of convolutional code and zigzag code. Zigzag codes are modified form of SPC codes. Since Zigzag codes show better performance than SPC code with slightly more complexity, SPC codes are replaced by Zigzag codes in Modified Turbo Codes. First few component codes of MTC are series combinations of Zigzag code and Convolutional code. Zigzag codes are used in remaining component codes. Use of good codes, like Convolutional Codes, in first few component codes improves error performance of remaining component codes, resulting in overall improvement in error performance. Since Convolutional codes are not used in all component codes, MTC has lower decoding complexity than that of standard Turbo code. BER performance of MTC is close to Turbo code.

To construct MTC a sequence of N information bits is arranged in an array of size J x K, here J represents number of rows and K represents number of columns in the rearranged data matrix.

$$[D]_{1xN} = [d]_{JxK} \quad \ldots(3)$$

$$N = J \text{ x } K \quad \ldots(4)$$

Fig. 3 shows a constituent encoder for general MTC. Encoder 1 encodes each column of information bits. Encoder 2 encodes each row of information bits and encoder 3 is used to encode parity bits computed by either encoder 1 or encode 2.

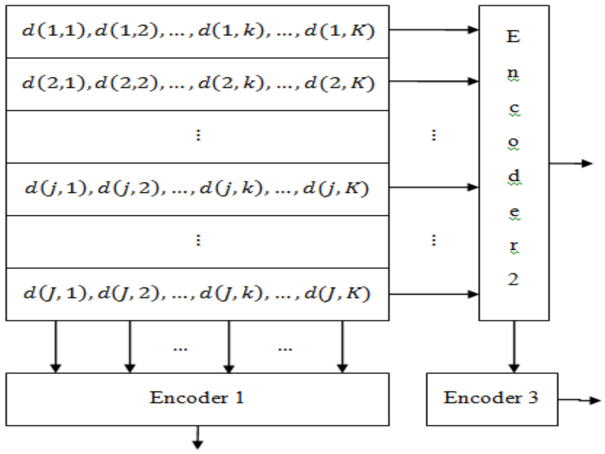

Fig. 3 Constituent Encoder for MTC

A general encoder for MTC consists of parallel concatenation of ***M*** constituent encoders is shown in Fig. 4. Random Interleavers are used to interleaver information bit sequence before each constituent encoder except the first constituent encoder. Each constituent encoder encodes information bit sequence in different manner i.e. different combination of rows and columns are encoded by encoder 1 and encoder 2 of the constituent encoder. Interleaver can be represented by π_m where $m = 1,2, \ldots, M-1$.

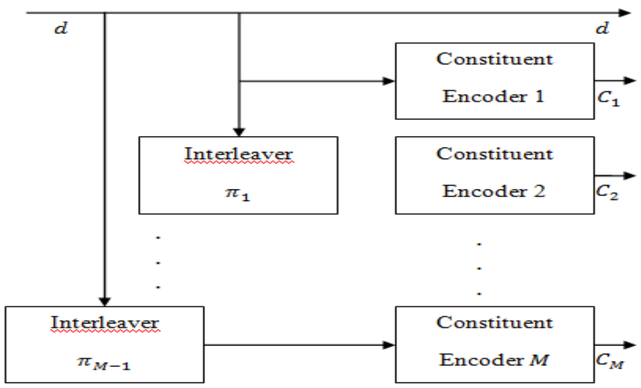

Fig. 4 Overall Encoder Structure of a General MTC

III. Simulation Results

Turbo code encoder consists of parallel concatenation of two rate **R =1/2** RSC encoders using random Interleaver. In Fig. 5 component used to design this turbo encoder block is shown. Parameter trellis structure defines number of state, constrained length, code generator and feedback connection for convolutional encoder. Trellis structure is given by generator polynomial. Random interleaver interleaves the information bit sequence using random permutations.

Deinterlacer separates the elements of the input signal to generate the output signals. The odd-numbered elements of the input signal become the first output signal, while the even-numbered elements of the input signal become the second output signal. To adjust code rate from **R =1/4** to **R =1/3** the odd bit sequence of second convolutional encoder is terminated. Parameters used for different component of turbo code encoder are shown below in the table 1.

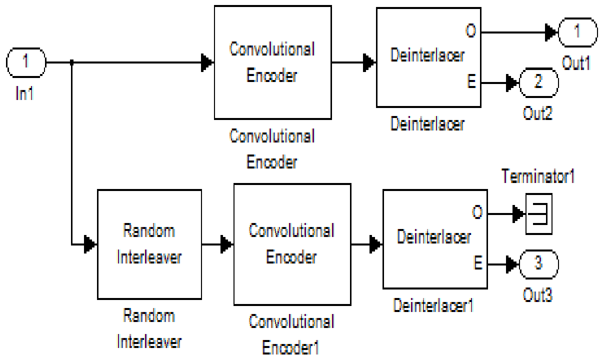

Fig. 5 TCC Encoder R=1⁄3

Table 1. Parameters for Turbo Code encoder

RSC Encoder	Parameters	Values
	Trellis	Poly2trellis (5, [37 21],37)
	Output	Truncated (reset every frame)
Random Interleaver	No. of Element	1024*128
	Initial seed	54123

BER performance for rate **R=1/3 and R=1/2** Turbo code coded data in accordance with simulation parameters is shown in Fig. 6 and 7 respectively.

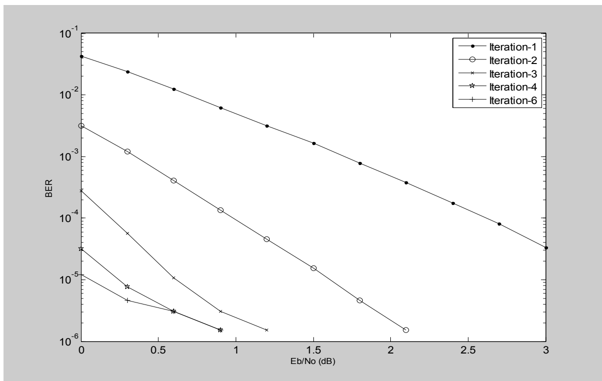

Fig. 6 BER Performance of Rate R=1⁄3 Turbo codes for Different Iteration

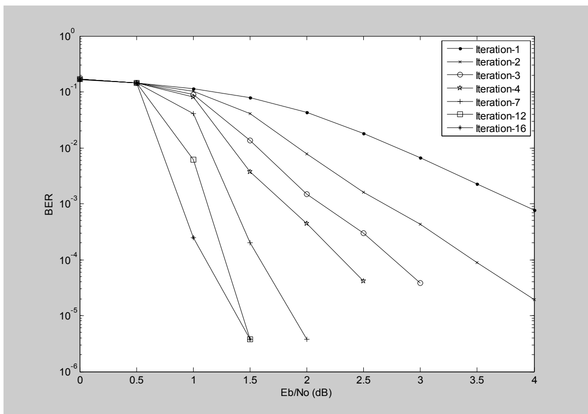

Fig. 7 BER Performance of Rate R=1⁄2 Turbo codes for Different Iteration

ISSN: 2278 – 7798

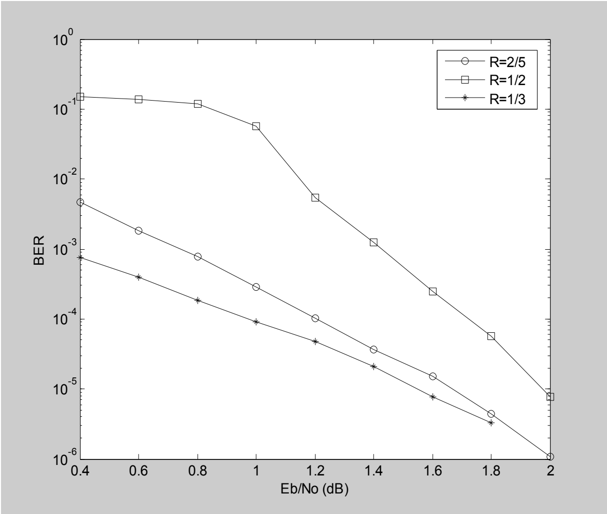

Fig. 8 BER Comparison for rate R=1⁄2, 2⁄5 and 1⁄3 CTC

Simulation result shows that BER performance of rate ***R* = 1⁄3** Turbo code is best and BER performance of rate ***R* = 2⁄5** Turbo code is better than rate ***R* = 1⁄2** Turbo code for low signal to noise ratio. There is a big difference in the BER performance of rate ***R* = 1⁄2** and rate ***R* = 1⁄3** Turbo code up to ***R* = 1⁄3** . For higher value of E_b/N_o **= 1** BER performance is nearly same for rate ***R* = 1⁄2** and ***R* = 1⁄3** Turbo code.

MTC consists of parallel concatenation of zigzag code with convolutional code little modification in the encoder structure and decoder structure. Zigzag encoder and decoder are designed using Matlab script code. In this section we compute BER performance for LCHTC and ILCHTC.

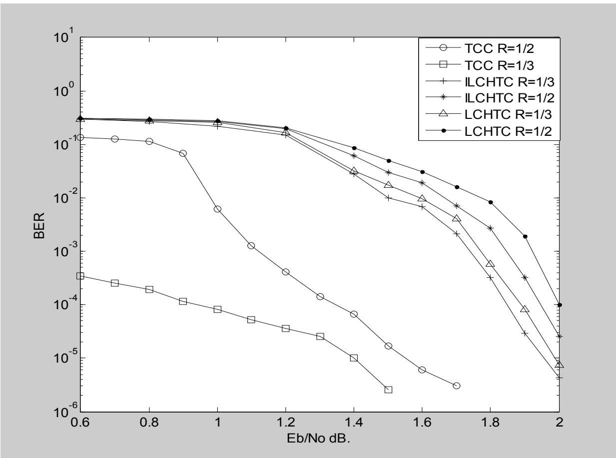

Fig. 9 BER Comparison for MTC and Turbo Code

According to simulation results shows that BER is nearly constant up to E_b/N_o **= 1 dB** for MTC and there is a difference of nearly 10^{-3} in BER for rate ***R* = 1⁄3** Turbo Code and MTC. Rate ***R* = 1⁄3** Turbo Code shows much better performance for low signal to noise ratio. But at higher signal to noise ratio BER performance for Turbo Code and MTC is nearly same. Simulation results show that at **BER** $= 8\times10^{-6}$ at E_b/N_o **= 1 dB** is achieved for ***R* = 1⁄3** LCHTC which is 0.5 dB away from E_b/N_o for the same BER for ***R* = 1⁄2** Turbo Code. Simulation result shows that there is not much difference in the BER performance for different rate MTC. **BER** $= 4\times10^{-6}$ is achieved at E_b/N_o = 2dB for rate **R=1/3** ILCHTC which is only 0.4 dB away from E_b/N_o for the same BER for R= ½ CTC.

IV. CONCLUSION

It has been investigated from this research work that MTC shows much less decoder complexity with negligible loss in BER performance. ILCHTC shows BER slightly improved than LCHTC and error convergence is also better than LCHTC due to the use of convolutional encoder for encoding few rows of information bit sequence for ILCHTC. While for LCHTC zigzag encoder encode all rows of information bit sequence and zigzag parity bits are encoded by convolutional codes.

REFERENCES

[1] Forney G. D., “Convolutional Codes: I. Algebraic Structure,” *IEEE Transaction Information Theory*, vol. 16, no. 6, November 1970, pp. 720-738.

[2] Berrou C., Glavieux A. and Thitimajshima P. “Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes,” *IEEE Proceedings of the International Conference on Communications (ICC93)*, Geneva, Switzerland, May 1993, pp. 1064-1070.

[3] Johannesson, R., Zigangirov, K.Sh.: “Fundamentals of convolutional coding” (*University Press India Limited, 2001, 1st edn*.).

[4] Papaharalabos, S., Sweeney, P., Evans, B.G. “SISO algorithms based on Max-Log-MAP and Log-MAP turbo decoding”, *IET Commun*., 2007, 1, (1), pp. 49–54.

[5] Ping, L., Huang, X., Phamdo, N. “Zigzag codes and concatenated zigzag codes”, IEEE Trans. Inf. Theory, 2001, 47, (2), pp. 800–807.

[6] Bhise, A., Vyavahare, P. “Improved low complexity hybrid turbo codes and their performance analysis”, *IEEE Trans. Commun*., 2010, 58, (6), pp. 1620–1622.

[7] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.

[8] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in proc. *IEEE ICC’93,* vol. 2, pp.1064-1070, May 1993.