International Journal of Engineering & Technology, 7 (4) (2018) 2949-2952

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.15370
Research paper

A Survey of interleaving techniques in turbo codes

M. Rajani Devi [1,2] *, K. Ramanjaneyulu [3], B. T. Krishna [4]

[1] *Research Scholar, JNTU Kakinada, Kakinada – 533003*
[2] *Associate Professor, Vardhaman College of Engineering, Shamshabad-501218*
[3] *Professor, PVP Siddhartha Institute of Technology, Kanuru, Vijayawada – 520007*
[4] *Professor, JNTU Kakinada, Kakinada – 533003*
**Corresponding author E-mail: rajimerigala@gmail.com*

Abstract

This paper focuses on the interleaving techniques used in Turbo codes in wireless communications. The performance of turbo codes is measured in terms of Bit Error Rate (BER) versus Signal to Noise Ratio (SNR) and the number of iterations. It tends to increase with better interleaving, better encoding and decoding algorithms. This paper presents the concepts of turbo coding, different interleaving methods and decoding techniques.

Keywords: *Bit Error Rate; Interleaving; Signal to Noise Ratio; Turbo Coding; Turbo Decoding.*

1. Introduction

Turbo codes are among the latest state of the art techniques for Forward Error Correction (FCC). It was with the help of these codes that the channel capacity advanced to the next heights [1] [2]. 3G and 4G mobile communications use turbo codes. They are also used in deep space satellite communications and are also used in applications where reliable information is transferred over bandwidth or latency constrained communication in the existence of data corrupting noise [3].

From 1948, researchers were trying to design codes close to channel capacity. The problem was if there was a very random like code which does not have any decoding structure, then the decoding complexity is very large. So the challenge is to design a code which has enough randomness into it along with enough structure, which can be exploited for decoding the code. The turbo codes are class of codes which are random, because of an inherent interleaver structure, fitted in the parallel concatenation structure. There is enough structure in the code which allows one to do efficient decoding of these error correcting codes.

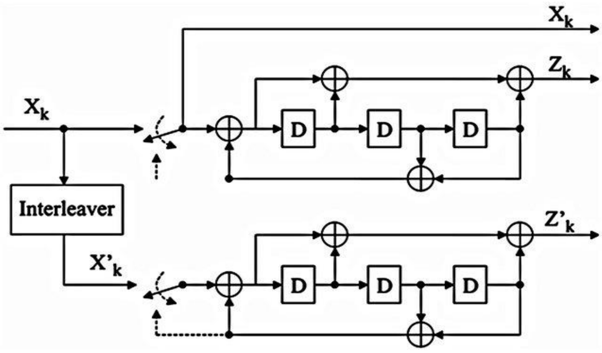

Fig. 1: Turbo Encoder.

An encoder of a turbo code consists of parallel concatenation. The Turbo encoder is better explained with the help of conventional block diagram depicted in figure 1. The turbo codes have a rate half-convolutional code with the input as the message denoted by u. The input is also fed to the interleaver block, the output of which is fed to rate half convolutional code. The first block keeps both the systematic version and the parity, whereas the lower block keeps on the parity. The terminations are generated at the outputs in the form of parity bits [4]. Nominally, this becomes a rate one third turbo code. If the termination is accounted, there will be a slight reduction of one third in the rate. Usually k will have to be large, at least 50, 100 for obtaining meaningful results [5]. To get higher rate codes like rate half, puncturing would be used. There are puncturing patterns that are possible depending on target rate. For rate half, there is a standard puncturing pattern in which all the odd-numbered bits from one parity are kept together and all the even numbered bits are kept in the other parity. The crucial point is both these blocks have to recursive systematic convolutional encoders.

Initially proposed turbo codes use systematic convolutional encoder. Subsequently researchers have also worked on designing design of non-systematic turbo codes but the ones, which were initially proposed in 1993 by Berrou et al, used systematic convolutional encoder. The third thing to note very crucial is the use of recursive encoders. The recursive encoders are feedback encoders. So there is a feedback from the output to the input. There is a feedback from the output going to the input there is a feedback from the output going to the input. Larger the minimum distance better is the error correcting capability of the code.

The prime components of the turbo encoder are, first thing is the concatenation of two or more encoders in a parallel fashion as opposed to serial concatenation. In serial concatenation what happens is, there are bits coming to the output of first encoder, which are fed as input to the second encoder. The interleaver just permutes the bit. So some bits which were there in, let's say bit location 1, maybe put in bit location 56 and it just shuffles bits here. Now the design of interleaver and the role of interleaver is very crucial for turbo codes.

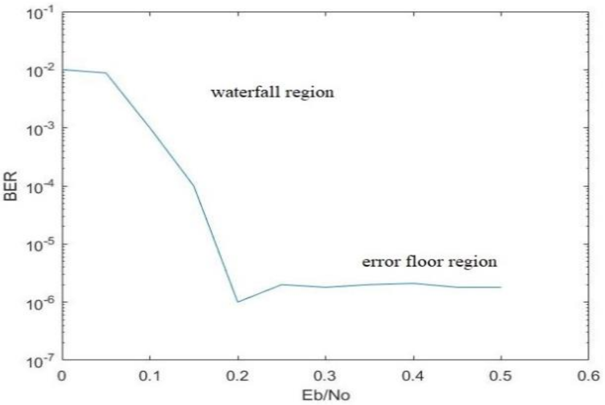

Fig. 2: Bit Error Rate Performance versus Signal to Noise Ratio.

In figure 2, the graph represents the bit error rate performance versus signal to noise ratio for a rate one third turbo code and for a block size more than 65000 is plotted. So there is a region where there is a steep fall in bit error rate performance. And this region is called waterfall region. And then there is a region where there is hardly any improvement in bit error rate in spite of increasing the signal to noise ratio is known as error floor region. Now performance in the waterfall region is governed by the convergence behavior of the constituent encoders under iterative decoding algorithm. And performance in the error floor region is governed by the distance spectrum of the turbo codes.

$$\mathrm{BER} = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{E_b/N_0}\right) \tag{1}$$

So typically, the constituent encoders the convolution constituent encoders that gives good performance are the ones which have short constraint length typically 3 4 like memory 3 memory 4. These are the ones that give good performance. If we use a more stronger code with memory 5 6 then typically the performance in the waterfall region is not good. Now what is being shown in figure 1 is a parallel concatenation of [2] encoders. These [2] encoders can be the same encoders or they could be different. So if the constituent encoders are same we call it a symmetric turbo code and if these constituent encoders are different we call it asymmetric turbo code.
Choice of recursive and systematic encoders is very crucial because for a recursive encoder a weight 1 pattern cannot terminate an encoder, whereas for a feed forward encoder a weight 1 pattern can terminate the encoder. That's why we should use a recursive encoder and systematic encoders help in initial performance in waterfall region and that's why they are preferred over non-systematic encoders. Especially if performance in the waterfall region is of interest. The process of removing some information bits and not sending them is called as puncturing. So in order to get higher rates, one can do puncturing to get higher rate codes. Bits can be punctured from the parity sequence to produce higher code rates. Puncturing of the information bits can also be done.
The encoder diagram in figure 1 uses 2 encoders and 1 interleaver. This structure can be extended for multiple encoders and multiple interleavers.

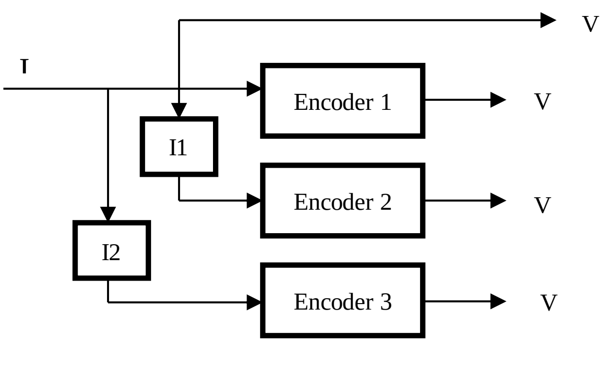

Fig. 3: Multiple Turbo Code.

The figure 3 depicts [2] encoders and 1 interleaver can be extended for multiple encoders and multiple interleavers. So this will result in a rate 1 by 4 multiple turbo code. And there are class of interleavers like S random interleavers which imposes some additional constraints which will allow that two adjacent bits are at least spread by some amount.
Interleave coding is an essential way to hide the transmission errors or to fix the errors when they appear. Transmission errors appear during bursts [6]. During the transmission, if any word of a sentence consists of bursts of errors, then instead of removing the entire word, a couple of letters would be taken out from each word. The decoder would recompile the sentence the same way it was sent [7].

2. Types of interleave

2.1. Block interleave

In this interleaving technique, the incoming message bits are organized into a 2D array with m rows and n columns with the data bits occupying the rows first [6]. The de-interleaving process comprises of unravelling the columns first to get the original information. Equation 1 depicts the indexing function of the block interleaver.

$$d_{IB}(j) = nj + \left(\left\lfloor \frac{j}{m} \right\rfloor\right) \bmod A \tag{2}$$

Where [y] is the largest integer <y, A = m x n (row size multiplied by column size)

2.2. Linearized interleave

The interleaver finds it difficult to use the floor function which makes it difficult to analyze the floor function [8]. The linearized interleaver was introduced having the index function,

$$d_{IN}(j) = pj + q \bmod LM \tag{3}$$

Where p, q: fixed integers in such a way that p is relatively prime to N

HCF (p,L) =1.

Index k is the angular coefficient of the linear interleaver. The number of solutions for the interleaving functions is only one and therefore it described the proper permutation which has one to one mapping of elements. For any τ, unique value of t exists. Large values of dmin are generated using these interleavers but they fail to produce low multiplicities. The block inteleavers produce good values of dfree, but it fails to produce the posterior part / thin spectrum.

2.3. S-random interleaves

The performance of the turbo codes is improved by the introduction of the S-random interleavers [9]. This interleaver especially affects the error floor region. The input and output indices that are close are neglected in this interleaving technique. The test condition for the random interleaver is depicted below dI(i) is the randomly selected index.

$$|dI(j) - dI(l)| \geq T, \text{ for all } j - S < l < I \tag{4}$$

If this condition of choosing index fails anywhere then that is castoff and a different one is selected. This process continues until all the N indices are chosen. By aggressively choosing the parameters the algorithm is not complete successfully as it results in the increase of the search time with S and T. However values of S, T < √ (L/2) complete within the reasonable time.

2.4. Quadratic interleaves

The quadratic interleavers differ in the way of implementation but are exactly similar to that of the random interleavers [9]. The quadratic indexing is depicted below:

$$d_{I_N}(i) = \frac{ki(i+1)}{2} \bmod L \quad (5)$$

Where k represents odd constant. Like the random interleavers, the quadratic interleavers also work well in the error floor region. The advantage they possess over the random interleavers is the fast algorithm formulation. The access of look-up tables are also redundant in these techniques.

2.5. Dithered relatively prime (DRP) interleaves

Dithered Relatively Prime Interleavers are a special designed interleavers. These were invented by Crozier and Guinard. Combining simpler component interleavers, DRP interleavers form a prevalent and operational interleaving method [10]. This interleaver is generated by a global permutation operation and two local dithering operations. This makes the interleaver represent as a combination of short trajectories. The overall calculation of the interleavers takes less memory than the counterparts even though the central interleaver is calculated recursively.

2.6. Convolutional interleave

In the convolutional interleaver, the input data is fed to a series of shift registers [11] [12]. The internal schematic is represented in the figure below:

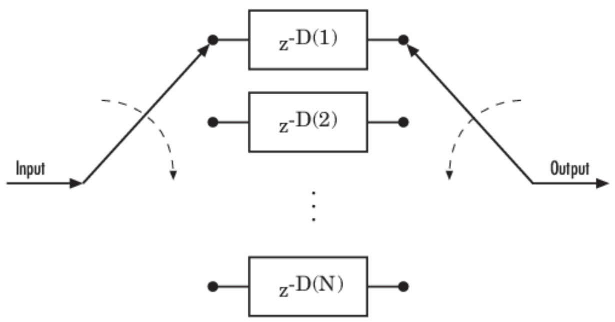

Fig. 4: Convolutional Interleave.

D (1), D(2) .. and so on are the shift registers. The delay produced by each of the shift registers is fixed. The outputs of the shift registers are fed to the outputs one after the other.

3. Turbo decoder

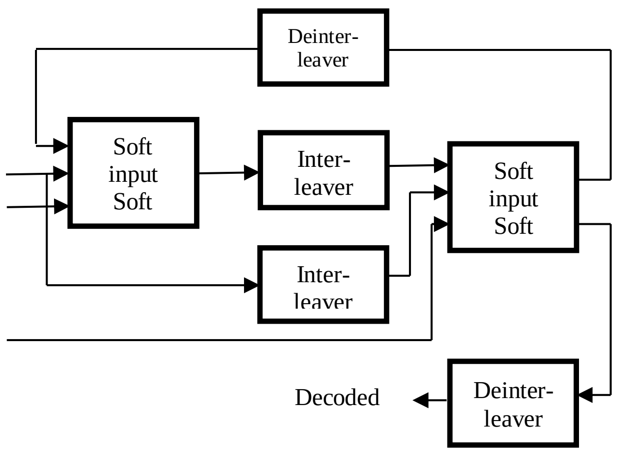

Fig. 5: Turbo Decoder.

The figure shows the block diagram of turbo decoder. In the architecture, the main blocks are the [2] decoders which correspond to the 2 encoders depicted in the encoder diagram. These two decoders come under the category of soft inputs soft output decoder. The term soft inputs soft output decoder means that this decoder receive real values, in contrast to that of the encoded binary values. The values are not quantized to 0s or 1s.the received signal is a noisy sequence directly from the channel. Here, the decoder produces not only the decision about whether the bit it thinks is 1 or 0 as output, but also provides information about the probability of the bit to be 0 or 1. That is the reason why the input and the output are labelled as soft, as opposed to hard decoder where the output would have been just 0s and 1s.

In the encoder discussed in section I, the information sequence was permuted using an interleaver, and the interleaved signal was sent to the second encoder. A simple encoder produced 3 outputs, first the information bits, second the parity bit coming out from the first encoder and the third, the parity bit coming out of the second encoder. After these bits pass through the channel, noisy version of the information sequence reaches the receiver.

In the decoder structure presented above, each decoder decodes the bits encoded by the two encoders encoded at the transmitter part. The decoder 1 takes the received information bits as input and the received parity bits.

- xn is the received information bit
- y1n is the received parity bit corresponding to the encoder 1
- y2n is the received parity bit corresponding to the encoder 2

The second decoder get the parity bits of encoder 2 ad the direct input, the interleaved version of the parity bits corresponding to the encoder 1 and finally the interleaved version of the output of the first decoder [13]. This is done to reciprocate the process that the signal underwent during the time of encoding.

Y1n and y2n are the two inputs to the decoder from the channel. The third input to the decoder is the output of the second decoder. This is a prior knowledge about the information sequence. Now initially when the decoding starts, no prior knowledge about the information bit is present. So it is assumed that the information bit is equally likely to be zero or [1] and the likelihood in this case is equal [14]. Considering the outputs of decoder 1, one of the output is called extrinsic information as the decoder based on the Trellis structure of the convolutional code computes this information. This output is fed as a prior knowledge to the second decoder. So, there are 2 decoders which are working independently but one decoder, once it decodes information sequence, passes some information to the other decoder, which is the probability likelihood of the bit being 0 or 1. So the other decoder will take the information as an input and then re-computes its probability and then it again re-computes some new probabilities of bit being 0 and 1 and this information is passed back to the first decoder. This information exchange repeats until both the decoders converge to particular decision. The inclusion of the interleavers in the decoder is to ensure that the order of the bits to the decoder blocks stay in sync with the encoder blocks. Finally, the input data is obtained by de-interleaving the output of the second decoder.

Because the decoder is comprised of two constituent modules, it is twice as complex as a conventional decoder when performing a single iteration. Two iterations will require twice the computation, rendering it four times as complex as a conventional decoder. Performing 10 – 20 iterations will impose substantial latency. For many applications, the delay is an acceptable price to pay for the bit error rate turbo codes deliver. But for others, such as voice or real-time applications, the delay of output data is prohibitive.

4. Conclusion

With the advent of 3G and 4G wireless communications, a data rate of about a few Giga bits per second is feasible and provided to the wireless user. When the data rate is in this scale, the channel capacity is crucial. To sustain the channel noise the power used in the check bits directly affects the usage of wireless equipment and also the success of any wireless application. When this parameter contradicts with bit error rate, things become more difficult to a designer. This is why about twice the power that was specified theoretically was utilized to attain theoretical channel capacity. Turbo codes overcome this limitation to some extent by using recursive coders and iterative soft decoders. The limitations of turbo codes include relatively high decoding complexity and high latency. Different interleavers that can be incorporated to get even better performance are presented. The decoding procedure was elaborated. Latest improvements and possible extensions are presented.

References

[1] M. Viterbi, "Shannon theory, concatenated codes and turbo coding", http://occs.donvblack.com/viterbi/index.htm, 1998.

[2] Shannon CE. "Communication in the presence of Noise". Proceedings IRE, Jan 1949, reprinted Proceedings IEEE, Vol. 86, and no. 2 Feb 1998.

[3] J. B. Cain, G. C. Clark, Jr., and J. M. Geist, "Punctured Convolutional Codes of Rate (n¡1)=n and Simplified Maximum Likelihood Decoding," IEEE Transactions on Information Theory, vol. IT-25, pp. 97–100, Jan 1979. https://doi.org/10.1109/TIT.1979.1055999.

[4] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," IEEE Transactions on Communications, vol. 44, pp. 1261–1271, Oct. 1996. https://doi.org/10.1109/26.539767.

[5] Oscar Y. Takeshita and Daniel J. Costello Jr,"New deterministic interleaver designs for turbo codes", IEEE Transactions on Information Theory, Vol. 46, Issue 6, pages 1988–2006, Sept. 2000. https://doi.org/10.1109/18.868474.

[6] Arya Mazumdar, A. K. Chaturvedi, and Adrish Banerjee, "Construction of turbo code interleavers from 3-regular Hamiltonian graphs", arXiv:cs.IT/0512093 v1, December 2005.

[7] Libero Dinoi and Sergio Benedetto, "Design of fast-prunable s-random interleavers", IEEE Transactions on Wireless Communications, Vol. 4m No. 5, pages 2540–2548, September 2005.

[8] D. Divsalar and F. Pollara, "On the design of turbo codes", TDA Progress Report 42-123, November 1995.

[9] Jonghoon Ryo and Oscar Y. Takeshita, "On quadratic inverses for quadratic permutation polynomials over integer rings", IEEE Transactions on Information Theory, Vol. 52, No. 3, pages 1254–1260, March 2006. https://doi.org/10.1109/TIT.2005.864442.

[10] Stephane Y. Le Goff and Faisal Obaid Al-Ayyan, "On the design of bit-interleaved turbo-coded modulation", IEEE Proceedings Information Theory Workshop, pages 73 – 75, September 2001.

[11] EirikRosnes and OyvindYtrehus, "On the design of bit interleaved turbo-coded modulation with low error floors", IEEE Transactions on Communications, Vol. 54, No. 9, pages 1563–1573, September 2006. https://doi.org/10.1109/TCOMM.2006.881215.

[12] Stephane Y. Le Goff and Faisal Obaid Al-Ayyan, "Design of bit interleaved turbo-coded modulations", IEEE Electronics Letters, Vol. 37, No. 16, pages 1030–1031, August 2001. https://doi.org/10.1049/el:20010702.

[13] Adrish Banerjee, Francesca Vatta, BartoloScanavino, and Daniel J. Costello Jr., "Nonsystematic turbo codes", IEEE Transactions on Communications. Vol. 53, No. 11, pages 1841–1849, November 2005. https://doi.org/10.1109/TCOMM.2005.858672.