Journal of Expert Systems (JES)
Vol. 2, No. 1, 2013, ISSN 2169-3064

www.worldsciencepublisher.org

Implementation and Analysis of Turbo Codes Using MATLAB

[1]Nabeel Arshad, [2] Abdul Basit

Department of Electrical Engineering Institute of Space Technology, Islamabad

[1]nabeel.cse07@gmail.com
[2]basitccp@gmail.com

Abstract: Information transmitted through a noisy channel gets deteriorated and becomes unreliable. To cater for such data altering effects, Forward Error Correction (FEC) techniques are used. A FEC technique; Turbo Code is very effective in such cases. In this paper, we have used different specifications such as coded and un-coded data, different coding schemes, varying the length of inter-leaver and the effect of presence or absence of inter-leaver in order to assess the performance of turbo codes. Simulations are conducted on MATLAB using BPSK modulation scheme in the presence of Additive White Gaussian Noise (AWGN) channel.

Key Words -- Forward Error Correction; Turbo Codes; Inter-leaver; AWGN, MATLAB; BPSK

1. INTRODUCTION

When digital data is transmitted through a communication channel, errors are embedded into it. Error Detection Coding is used to detect the errors in the received information and these errors can be corrected by the use of Error Correction Coding. Error Detection Coding and Error Correction Coding when used collectively, is known as Error Control Coding [1]. Error Control Coding has different types of coding schemes such as Linear Block Codes, Convolution Codes, and Turbo Codes etc. Linear Block Codes are useful if there is one to one correspondence between message and its code word.

Convolution Codes on the other hand, have additional structure like block codes in generator matrix [2]. This causes the encoding operation to be like a convolution operation. Convolution codes provide much better performance than block codes. When the block length increases, decoding complexity is also increased. In 1993, a technique was introduced which provides modest decoding complexity as block length increases [3].

This became known as Turbo Codes. Turbo Codes are also known as Parallel Concatenated codes and the decoding complexity is small for the dimension of code, whereas the code length possible is very long. The bounds of Shannon's Limit also become achievable for all practical purposes because the decoding complexity becomes small [4].

This paper explains configuration of turbo encoder and decoder and how this coding scheme provides better performance by varying the different parameters such as inter-leaver depth etc.

2. SYSTEM MODEL

In this section system model is explained in detail. In figure 1, system model is shown. Input sequence is first passed through turbo encoder. Then after modulation it is transferred through a communication channel. At receiver side data is demodulated and passed through turbo decoder.

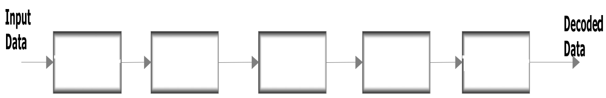

Figure 1: System Model

2.1 Turbo Encoder

The turbo encoder consists of two RSC (Recursive Systematic Coders) with an inter-leaver separating them [5]. The block diagram for turbo encoder is shown in figure 2. It is comprised of two transfer functions which represent the systematic components of RSC encoders and an inter-leaver. The input symbols are simply permuted in a random fashion and forwarded to the second RSC encoder. RSC encoders work best due to their IIR response. The transfer function of each encoder is given by:

$$Z(x) = \frac{h(x)}{g(x)}$$

It is important for both encoders to have the same transfer function. [7]

Suppose we have an input sequence $X = b = \{b_1, b_2, \dots \dots, b_{N-1}\}$. In encoder, this input sequence is presented in three ways. It is clichéd as it is to the output to produce the systematic output sequence $u_t^{(0)} = X_t$, t=0, 1, 2……, N-1.Second, the input sequence is passed through first RSC encoder with transfer function $Z(x)$, producing a parity sequence $\{u_0{}^{(1)}, u_1{}^{(1)}, \dots\dots, u_{N-1}{}^{(1)}\}$ at output. On combining $u_t{}^{(0)}$ and $u_t{}^{(1)}$ we get code rate R=1/2.

Third, input sequence is passed through inter-leaver of depth N. it produces interleaved output sequence say$\acute{X} = \prod(X)$. This interleaved sequence is fed towards the second RSC encoder with same transfer function as that of the first RSC encoder, resulting sequence $u^2 = \{u_0{}^{(2)}, u_1{}^{(2)}, \dots\dots, u_{N-1}{}^{(2)}\}$. Multiplexing these three output sequences gives the following output sequence:

$$u = \begin{Bmatrix} (u_0{}^{(0)}, u_0{}^{(1)}, u_0{}^{(2)}), (u_1{}^{(0)}, u_1{}^{(1)}, u_1{}^{(2)}), \dots\dots, \\ (u_{N-1}{}^{(0)}, u_{N-1}{}^{(1)}, u_{N-1}{}^{(2)}) \end{Bmatrix}$$

This Results in code rate of 1/3. The code has two parity sequences u^1 and u^2 which are independent of each other due to the presence of the interleaver.

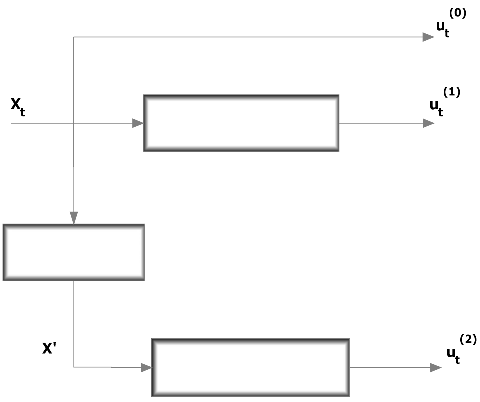

Figure 2: Turbo Encoder

2.2 Inter-leaver:

Coding techniques such as convolutional codes are suitable for channels with random errors like binary symmetric channel or AWGN channel. [8] But there are many channels where errors occur continuously. A burst of errors produces large errors in code words. So, there appears a need of strong correction capability. In order deal with these bursty channels, inter-leaver is introduced. Inter-leaver works by taking an input sequence and permuting it randomly or according to a prescribed method.

Inter-leaver proves to be very effective in dealing with bursty channels by permuting the data at the receiver side. There are various types of inter-leavers like block inter-leaver and convolutional inter-leaver. In turbo codes inter-leaver that permutes the data randomly is preferred. In figure 3, below working of random inter-leaver is shown. Suppose our input is [1,2,3,4,5] and after passing through inter-leaver we obtain [4,3,1,5,2].

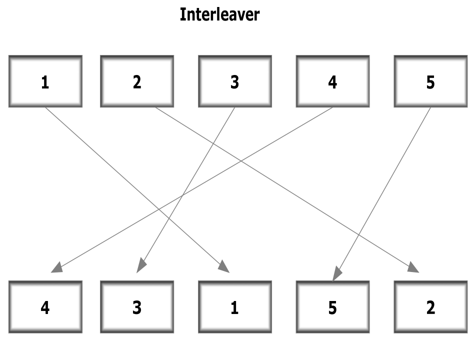

Figure 3: Inter-leaver

2.3 Turbo Decoder:

The block diagram of turbo decoder is shown in figure 4. After modulation, the encoded data is transmitted through a communication channel and the output of the channel is a vector *v*. The vector *v* is first de-multiplexed into the vectors $V^{(0)}$ (corresponding to $u^{(0)}$), $V^{(1)}$ (corresponding to $u^{(1)}$) and $V^{(2)}$ (corresponding to $u^{(2)}$). The working of the decoder is as follows; the data $(V^{(0)}, V^{(1)})$related to first encoder are fed to Decoder 1. The algorithm running inside decoder 1 is Viterbi algorithm explained in next section.

The output of decoder 1 is interleaved and then fed to decoder 2. As the data from RSC encoder 2 was the interleaved version of the input, this restricts the vector $V^{(0)}$to be passed through an interleaver before passing through decoder 2. Then, the output from decoder 2 is fed to decoder 1 and this process of transferring the information back and forth continues until some maximum number of iterations is achieved.

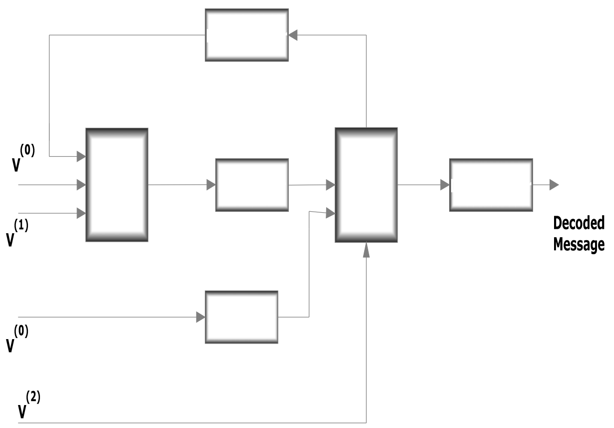

Figure 4: Turbo Decoder

3. Viterbi Algorithm

Given the received data, maximum likelihood code sequence can be calculated with the help of Viterbi algorithm. This algorithm is the shortest path algorithm; it computes the shortest path associated with the code through trellis. The decoder accepts the input information $u = [u_0, u_1, \dots\dots]$ and makes a guess about the transmitted information and from that guess estimates the sequence of input information $\{X_0, X_1, \dots\}$.

Working of this algorithm is as follows; we know that a coded sequence $\{u_{0,} u_1, \dots\dots\}$ represents a path through trellis on the encoder side. The received sequence becomes corrupted when the encoded data is passed through a channel and hence does not represent the same path through the trellis. The decoder estimates a path closest to the received information through trellis. The likelihood function appropriate for the channel determines the measure of “closest” and this gives us the decoded output. For a Binary Symmetric Channel (BSC) the likelihood path is closest in hamming distance however for the AWGN channel, the likelihood path is closest in Euclidean distance. [9], [10]

3.1 AWGN Channel:

For this channel, output from demodulator is a continuous alphabet. We cannot describe these alphabets as correct or incorrect. It is seen that maximizing the $p(z|u^m)$ is equal to maximizing the correlation between the codeword u^m and the value received from channel z given below:

$$\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} z_{ij}\, u_{ij}{}^{m}$$

So decoder will select a code-word that provides closest Euclidean distance to z. in order to apply the above equation, decoder needs to be able to handle analog values.[5] But this is impractical because decoders are implemented in digital domain, so we need to quantize our data. This quantized Gaussian channel referred to as soft decision channel.

4. SIMULATION RESULTS

In this section MATLAB simulations are shown. The performance of turbo codes is evaluated in terms of BER. These simulations are conducted in the presence of AWGN channel and the modulation scheme used is BPSK. The BER plots for coded and un-coded data, Convolutional codes Vs. Turbo codes are shown in this section. Similarly, performance of turbo codes in the presence of inter-leaver and without inter-leaver and for various lengths of inter-leaver is shown.

In figure 5 a comparison is made between coded and un-coded data over AWGN channel. It is clear from the graph that with the use of coding scheme we can achieve desired BER at less value of Eb/No as compared with the un-coded data. In the figure shown below coded data is providing a gain of 3dB as compared with the un-coded data.

In figure 6 a comparison is made between different types of coding schemes. It is known that turbo codes provide better performance as compared with other coding schemes. It is quite clear from the graph that turbo codes provide better performance and a gain of 1dB for BER 10^{-1}.

In figure 7 a comparison is made when we use inter-leaver in turbo codes and when we did not use it. Since it is known that inter-leaver is used to change the burst of errors into random errors and error performance improves in this way. Same thing is shown in figure 7, when we use an inter-leaver error performance improves.

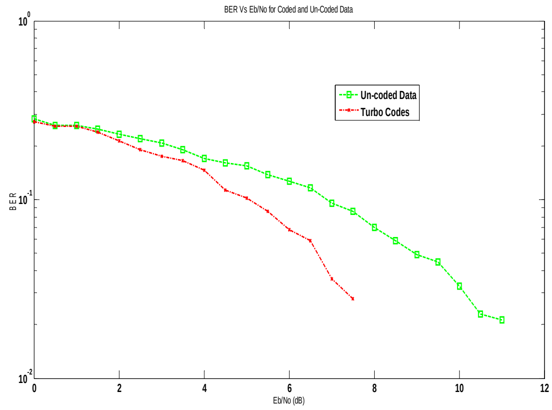

Figure 5: Coded Vs Un-Coded data

In figure 8, a comparison is made for varying length of the inter-leaver. Larger the length of inter-leaver more able it will to randomize the errors and error performance will improve. It is shown the figure that inter-leaver with length 300 provides better performance as compared with the other ones.

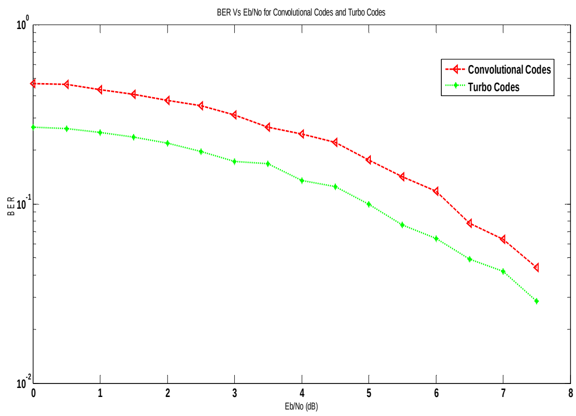

Figure 6: Convolutional Vs Turbo Codes

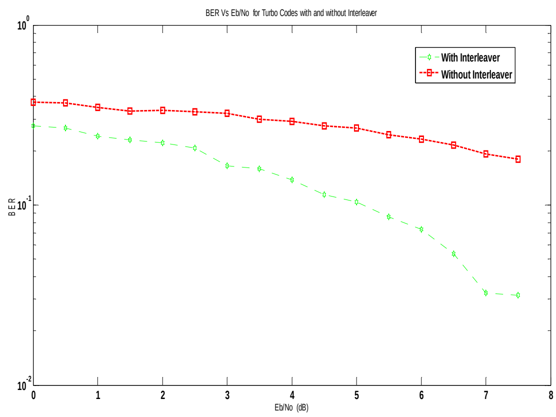

Figure 7: Turbo Codes with and Without Inter-leaver

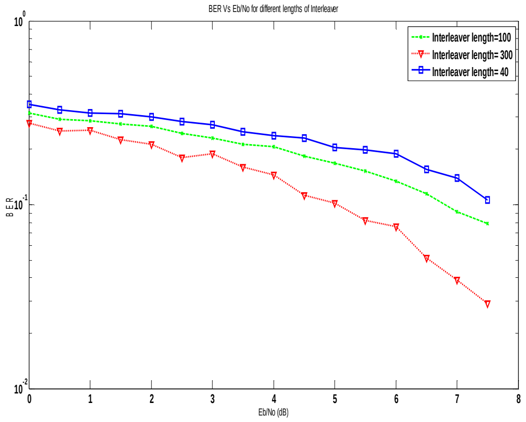

Figure 8: Turbo Codes with varying length of Inter-leaver

5. CONCLUSION

In this paper, performance of turbo codes is evaluated. We have used coded and un-coded data to compare the performance, different coding schemes. In order to see the importance of inter-leaver, we have shown performance of turbo codes in the presence of inter-leaver and in the absence of inter-leaver. Similarly performance of turbo codes by varying the length of inter-leaver is also shown.

It is concluded that turbo codes provide better error performance and sufficient coding gain as compared with other schemes. It is also deduced that presence of inter-leaver is much important in order to improve the error performance.

REFERENCES

[1]. Todd K. Moon, “Error Correction Coding Mathematical Methods and Algorithms”.

[2]. Bernard Sklar, “Digital Communication Fundamentals and Applications”, communication engineering services Tarzana, California.

[3]. Claude Berrou and Alan Glavieux, “Near Optimum Error Correcting Coding and Decoding: Turbo Codes”, IEEE Transactions on Communications, Vol. 44, No. 10, October 1996.

[4]. Hui Jin and Robert J. McEliece, “Coding Theorems for Turbo Codes Ensembles”, IEEE Transactions on Information Theory, Vol.48, No.6, June 2002.

[5]. SHANNON, C.E.: “Probability of Error for Optimal Codes in a Gaussian Channel”, Bell Sys. Tech. J., May 1959, 38, P.P 611-656.

[6]. Christoph S Tuder, Christian Benkeser and Quiting Huang, “Design and Implementation of a Parallel Turbo Decoder ASIC for 3GPP LTE”, IEEE Transactions of Solid State Circuits, Vol. 46, No. 1, January 2011.

[7]. Gregory poltyrev, “On coding without restrictions on AWGN channel”, IEEE Transactions on Information Theory. Vol. 40, No. 2, March 1994.

[8]. G. X. Zhu, B. Feng and W. Y. liu, “ A BER model for turbo codes for AWGN channel”, IEEE Intl. workshop on VLSI Design and Video Tech, Suzhou, China, May 28-30,2005

[9]. Bertrand M. Hochwald, “Source and Channel Coding tradeoff over Gaussian Channel”, ISIT 1998, Cambridge, U.S.A, August 16-21.

[10]. Richard V. Cox, Fellow IEEE and Carl-Erik W. Sundberg, Fellow IEEE, “An Efficient Adaptive Circular Viterbi algorithm for Decoding generalized Tailbiting Convolutional codes”, IEEE Transactions on Vehicular Technology, Vol. 43, No. 1, February 1994.