Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(6): 29-36

Research Article ISSN: 2394 - 658X

Simulative Analysis of Turbo Codes over AWGN Channel for BPSK Modulation Scheme

Moina Arora and Satbir Singh

Department of Electronics Technology, GNDU, Regional Campus, Gurdaspur, Punjab, India
moina_arora@yahoo.com

ABSTRACT

Turbo codes are widely used in satellite communication and 3G services. These are the major area of research in these days. Turbo codes are the forward error correcting codes that has made the near Shannon limit performance possible when suboptimal iterative decoding algorithms are used. The classical turbo encoder is built by using two recursive convolution encoders separated by an interleaver. In this paper we have implemented a system in which data is first encoded using turbo encoder and then using AWGN channel it is transferred to decoder where it is decoded using iterative soft input soft output decoder and the performance of the decoder with respect to the number of iterations has been discussed in the result section of this paper. Also, how the bit error rate changes with respect to the number of iteration is been discussed.

Key words Turbo Codes, AWGN, Bit Error Rate (BER)

INTRODUCTION

The fundamental task of a communication system is to transmit and receive information, whether it is in voice or data form. In theory, a communication system requires that information should be transferred from transmitter to receiver in such a way that the information received is of same quality as that transmitted. In practice, the presence of distortion in various form means that achieving this is not always possible, especially where wireless Medias are used for transmission. To achieve something close to the transmitted information at the receiver, a communication system must guard against the changes that this distortion can cause in the transmitted data. Error control coding (channel coding) is one method available to the communication system designer, arranging and supplementing the data in such a way that errors can be corrected at the receiver.

In 1948, Shannon introduces the concept of channel capacity, describing the limit to the amount of data that could be transmitted across any given channel. Since then, attaining this maximum theoretical channel capacity has been the goal of many mobile communications researchers. Hence, reliable transmission (i.e. transmission with bit error probability less than any given value) is still possible over noisy channels as long as the transmission rate is less than a number called the channel capacity which is characterize by the channel to accomplish reliable communication over noisy channels [5]. The task of channel coding is to encode the information sent over a communication channel in such a way that in the presence of channel noise, errors can be detected and/or corrected. For this we have two types of error correcting codes as mentioned below

Backward Error Correction (BEC)

It requires only error detection after the transmission. If an error is detected, the sender is requested to retransmit the message. While this method is simple and easy to implement it also sets less requirements on the code's error-correcting properties, whereas on the other hand this type of system requires duplex communication for sending information back to the transmitter whenever error is detected which causes undesirable delays in transmissions. So this is only used in small systems where distance and transmission time are not of much concern.

Forward Error Correction (FEC)

It requires that the decoder should also be capable of correcting a certain number of errors, i.e. it should be capable of locating the positions where the error has been occurred. Since FEC codes require only simplex communication, they are especially attractive in wireless communication systems.

Shannon proposed forward error correcting (FEC) codes as a viable solution. Although, Shannon' s work did suggest a way to solve the problem, it did not propose a mechanism to design good FEC codes which could provide performance close to the channel capacity with reasonable capacity. In a digital communication system, the purpose of the channel code is to add redundancy to the binary data stream to combat the effect of signal degradation in the channel. Ideally, channel codes should meet the following requirements

- Channel codes should have high bit rate to maximize data throughput.
- Channel codes should have good bit error rate (BER) performance at the desired signal to noise ratio (SNR) to minimize the energy needed for transmission
- Channel codes should have low encoder/ decoder complexity to limit the size and cost of the transceivers.
- Channel codes should introduce only minimal delays, especially in voice transmission, so that no degradation in signal quality is detective.

These requirements are very difficult to obtain simultaneously; excellent performance in one requirement usually comes at the price of reduced performance in another [5]. However, for cellular voice and data communication, it is desirable that all these requirements should be met, which makes cellular communication system very difficult to design. Designing a channel code is always a trade-off between energy efficiency and bandwidth efficiency.

Codes with lower rate can usually correct more errors. If more errors can be corrected, the communication system can operate with a lower transmit power, transmit over longer distances, tolerate more interference, use smaller antennas and transmit at a higher data rate. These properties make the code energy efficient. On the other hand, low-rate codes have a large overhead and are hence they require large bandwidth. Also, decoding complexity grows exponentially with code length, as the code length increases complexity also gets increased. Also long (low-rate) codes set high computational requirements to conventional decoders. This is the one of the major problems of channel coding ie encoding is easy but decoding is tough. For every combination of bandwidth (W), channel type, signal power (S) and received noise power (N), there is a theoretical upper limit on the data transmission rate R, for which error-free data transmission is possible. This limit is called channel capacity or also Shannon capacity (after Claude Shannon, who introduced the notion in 1948. For additive white Gaussian noise channels, the formula is

$$\mathrm{R} < W\log_2(S|\mathrm{N})\ \text{bits/sec}$$

In practical settings, there is no such thing as an ideal error-free channel. Instead, error-free data transmission is interpreted in such a way that the bit error probability can be minimized to an arbitrarily small constant. Now, if the transmission rate, the bandwidth and the noise power are fixed, we get a lower bound on the amount of energy that must be expended to convey one bit of information. Hence, Shannon capacity sets a limit to the energy efficiency of a code. Although Shannon developed his theory already in the 1940s, several decades later the code designs were unable to come close to the theoretical bound. Even in the beginning of the 1990s, the gap between these theoretical bound and practical implementations was still at best about 3dB. This means that practical codes required about twice energy as the theoretical predicted minimum.

As shown in the Fig. 1only the turbo and LDPC codes are approaching the theoretical Shannon limit [2].

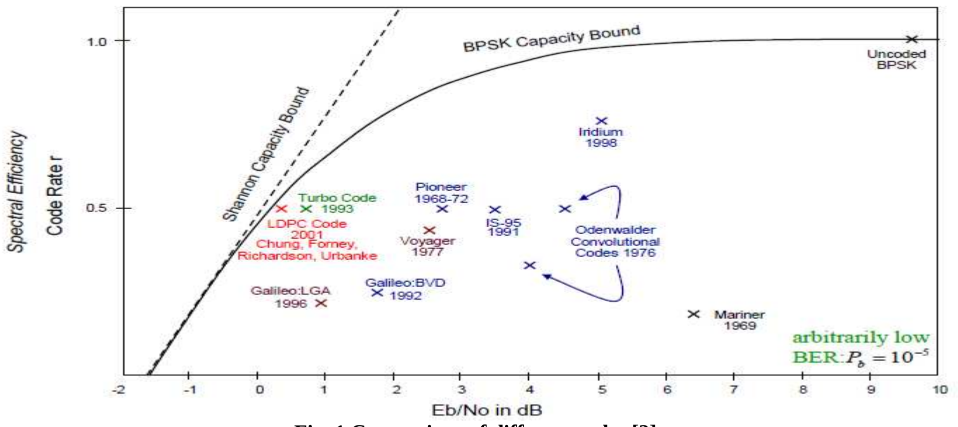


Fig. 1 Comparison of different codes [2]

DEVELOPMENT OF CODES

In channel coding, redundancies are introduced in the information sequence in order to increase its reliability. The channel coding theorem states that even at relatively low E_b/N_0, reliable communication can still be maintained. However, the theorem tells us nothing about how to design the codes that achieve such performance. All what it say is that the code should appear random. Unfortunately random codes are very difficult to decode. There should be some structure in the code to make the decoding feasible.

The proposal of parallel concatenated convolution codes (PCCC) [13], called turbo codes, has solved the problem of structure and randomness by allowing structure through concatenation and randomness with interleaving. The introduction of turbo codes has increased the interest in the coding area since these codes fulfil most of the requirements given by the channel coding theorem.

DEVELOPMENT OF SYSTEM MODEL

Binary Signal Generator

It generates the binary symbols of random pattern that are the message bits which are encoded by the encoder in the following stages.

Turbo Encoder

A turbo code is the parallel concatenation [13] of a number of RSC codes. Usually the number of codes is kept low, typically two, as the added performance of more codes is not justified by the added complexity and increased overhead. The input to the second encoder is an interleaved version of the systematic x, thus the outputs of coder 1 and coder 2 are time displaced codes generated from the same input sequence.

Thus the turbo encoder consists of two blocks of recursive convolutional encoder and an interleaver. The two encoders used are normally identical. There are different types of interleaver present which can be use depending upon the output requirements. The input sequence is only presented once at the output. The outputs of the two coders are given to the parallel to serial converter to transmit the required code.
The generic design of a turbo code is depicted in Fig. 3.

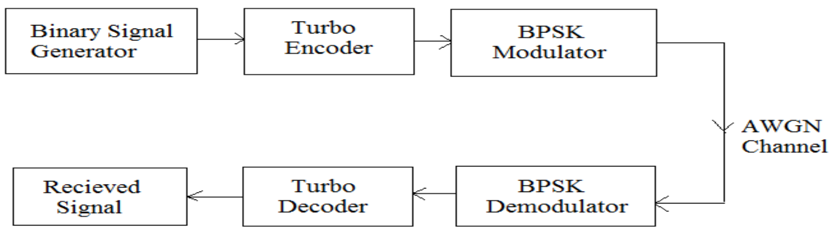


Fig. 2 System model

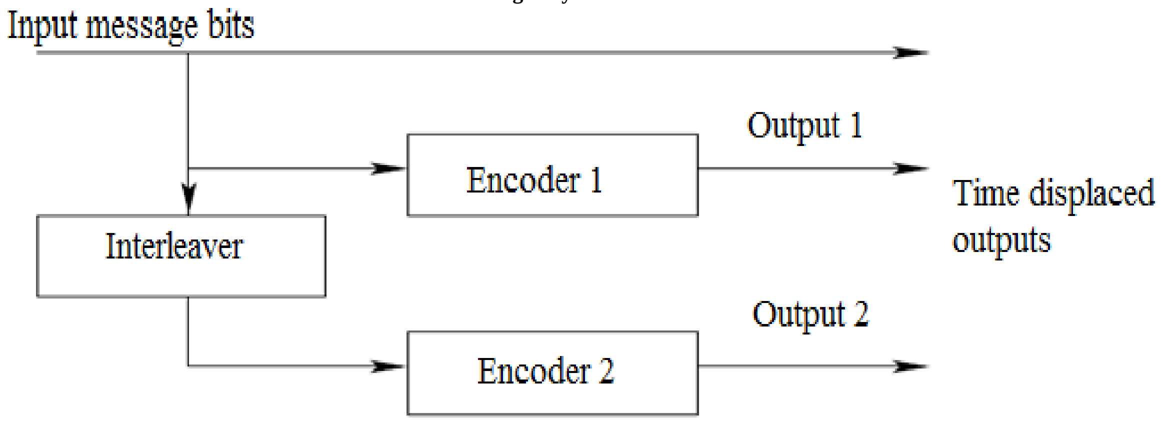


Fig. 3 Turbo encoder

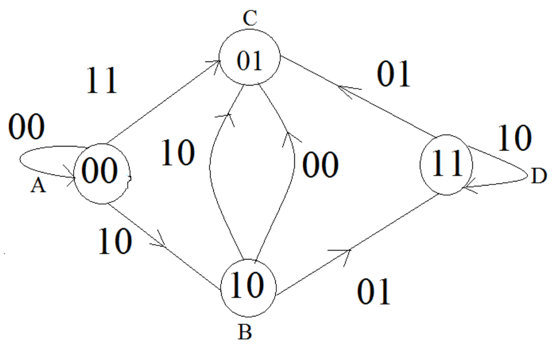


Fig. 4 State diagram

The choice of the interleaver [1] is a crucial part in the turbo code design. The task of the interleaver is to 'scramble' bits in a random fashion. This 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 means that even if one of the output code words has low weight, the other usually does not, and there is a smaller chance of producing an output with very low weight [19]. State diagram of the basic turbo encoder is given as below in Fig. 4 which is explaining us the state transition that encoder will go while coding the message bits.

Higher weight, as we saw above, is beneficial for the performance of the decoder. Secondly, since the code is a parallel concatenation of two codes, the divide-and-conquer strategy can be employed for decoding. If the input to the second decoder is scrambled, also its output will be different or 'uncorrelated from the output of the first encoder. This means that the corresponding two decoders will gain more from information exchange. The interleaver design has a significant effect on code performance. A low weight code can produce poor error performance, so it is important that one or both of the coders produce codes with good weight. If an input sequence x produces a low weight output from coder 1, then the interleaved version of x needs to produce a code of good weight from coder. There are different types of interleavers available as given below [11].

A 'row-column' Interleaver
In this type the data is written row-wise and is read column wise. This inteleaver design is very simple and also it provides little randomness.
A 'helical' Interleaver
In this data is written row-wise and is read diagonally.
An 'odd-even' Interleaver
In this first, the bits are left uninterleaved and encoded, then scrambled and encoded, then now only the even-positioned coded bits are stored. Odd-even encoders can be used, when the second encoder produces one output bit per one input bit.
A pseudo-random Interleaver
This type of interleaver is defined by a pseudo-random number generator or a look-up table.
The design of these interleavers is given as below

Input				
X_1	X_2	X_3	X_4	X_5
X_6	X_7	X_8	X_9	X_{10}
X_{11}	X_{12}	X_{13}	X_{14}	X_{15}

Row-column interleaver output														
X_1	X_6	X_{11}	X_2	X_7	X_{12}	X_3	X_8	X_{13}	X_4	X_9	X_{14}	X_5	X_{10}	X_{15}
Row-column interleaver output														
X_1	X_6	X_{11}	X_2	X_7	X_{12}	X_3	X_8	X_{13}	X_4	X_9	X_{14}	X_5	X_{10}	X_{15}
Odd –even interleaver output														
Encoder output without interleaving														
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
Y_1	-	Y_3	-	Y_5	-	Y_7	-	Y_9	-	Y_{11}	-	Y_{13}	-	Y_{15}
Encoder output with row column interleaving														
X_1	X_6	X_{11}	X_2	X_7	X_{12}	X_3	X_8	X_{13}	X_4	X_9	X_{14}	X_5	X_{10}	X_{15}
-	Z_6	-	Z_2	-	Z_{12}	-	Z_8	-	Z_4	-	Z_{14}	-	Z_{10}	-
Final output of the encoder														
Y_1	Z_6	Y_3	Z_2	Y_5	Z_{12}	Y_7	Z_8	Y_9	Z_4	Y_{11}	Z_{14}	Y_{13}	Z_{10}	Y_{15}

Fig. 5 Interleaver designs

Modulator
Binary phase shift keying ie BPSK modulator is used in this system that will modulate the encoded data produced by the turbo encoder with a carrier signal so that it can be transmitted on another frequency of the signal.

$$V_{bpsk}(t) = b(t) \cos w_0(t)$$

AWGN Channel
After modulating the coded data the additive white Gaussian noise is added into the encoded data which will get added to the required information and degrade the signal while transmitting through the channel. As in the practical model whenever signal is transmitted using wireless media noise get added in the required data and degrade the

signal quality so we use AWGN ie our model to make it practical realization. BER rat foe BPSK over AWGN channel is expressed as

$$P_B = \frac{1}{2} erfc \sqrt{\frac{E_b}{N_0}}$$

Demodulator
Binary phase shift keying is used in the modulation so same type of demodulator is used in the receiver circuit which will demodulate the signal from the carrier signal and extract the original information from the signal.

Turbo Decoder
The two main types of decoder are Maximum A Posteriori (MAP) and the Soft Output Viterbi Algorithm (SOVA). MAP looks for the most likely symbol received whereas SOVA looks for the most likely sequence[3]. Both MAP and SOVA perform similarly at high Eb/No. At low Eb/No MAP has a distinct advantage, gained at the cost of added complexity. We have used soft input soft output decoder in the communication system ie iterative decoder is being used in the system.

SOVA Decoder
SOVA is very similar to the standard Viterbi algorithm used in hard demodulators. It uses a trellis to establish a surviving path but, unlike its hard counterpart, compares this with the sequences that were used to establish the non-surviving paths. Where surviving and non-surviving paths overlap the likelihood of that section being on the correct path is reinforced. At the output of each decoding stage the values of the bit sequence are scaled by a channel reliability factor, calculated from the likely output sequence, to reduce the probability of over-optimistic soft outputs [3]. The sequence and its associated confidence factors are then presented to the interleaver for further iterations. After the prescribed number of iterations, the SOVA decoder will output the sequence with the maximum likelihood.

Another strategy involves combining simple codes in a parallel fashion, so that each part of the code can be decoded separately with less complex decoders and each decoder can gain from information exchange with others. This is called the divide-and-conquer strategy. In a typical communications receiver, a demodulator is often designed to produce soft decisions, which are then transferred to a decoder. The error-performance improvements of systems utilizing such soft decisions compared to hard decisions are typically approximated as 2 dB in AWGN[17]. Such a decoder could be called a soft input/hard output decoder, because the final decoding process out of the decoder must terminate in bits (hard decisions). With turbo codes, where two or more component codes are used, and decoding involves feeding outputs from one decoder to the inputs of other decoders in an iterative fashion, a hard-output decoder would not be suitable. That is due to hard decisions into a decoder degrade system performance (compared to soft decisions).

Hence, what is needed for the decoding of turbo codes is a soft input/soft output decoder. For the first decoding iteration of such a soft input/soft output decoder, illustrated in Fig.6 we generally assume the binary data to be equally likely, yielding an initial a priori LLR value of L(d) = 0 for the third term in Equation [6]. The channel LLR value Lc(x), is measured by forming the logarithm of the ratio of the values of l1 and l2 for a particular observation of x, which appears as the second term in Equation. The output L(dˆ) of the decoder in is made up of the LLR from the detector, L′(dˆ) , and the extrinsic LLR output, Le(dˆ), representing knowledge gleaned from the decoding process. As illustrated in Fig., for iterative decoding, the extrinsic likelihood is fed back to the decoder input, to serve as a refinement of the a priori probability of the data for the next iteration.

Map decoder MAP was first proposed by Bahl5 et al and was selected by Berrou et al as the optimal decoder for turbo codes. MAP looks for the most probable value for each received bit by calculating the conditional probability of the transition from the previous bit, given the probability of the received bit. The focus on transitions, or state changes within the trellis, makes LLR a very suitable probability measure for use in MAP [10].

The MAP algorithm is unlike the Viterbi algorithm (VA), where the APP for each data bit is not available. Instead, the VA finds the most likely sequence to have been transmitted. However, there are similarities in the implementation of the two algorithms. When the decoded bit-error probability, PB, is small, there is very little performance difference between the MAP and Viterbi algorithms. However, at low values of bit-energy to noise-power spectral density, Eb/N0, and high values of PB, the MAP algorithm can outperform decoding with a soft-output Viterbi algorithm called SOVA [18] by 0.5 dB or more. For turbo codes, this can be very important, since the first decoding iterations can yield poor error performance. The implementation of the MAP algorithm proceeds somewhat like performing a Viterbi algorithm in two directions over a block of code bits. Once this bidirectional computation yields state and branch metrics for the block, the APPs and the MAP can be obtained for each data bit represented within the block. We describe here a derivation of the MAP decoding algorithm for systematic convolutional codes assuming an AWGN channel model, as presented by Pietrobon [18]. We start with the ratio of the APPs, known as the likelihood ratio Λ(dk) , or its logarithm, called the LLR, as shown below.

$$\wedge(\hat{d_k}) = \sum_m \lambda_k^{1,m} / \sum_m \lambda_k^{0,m}$$

$$L(\hat{d_k}) = \log\left[\sum_m \lambda_k^{1,m} / \sum_m \lambda_k^{0,m}\right]$$

where $\lambda_k^{i,m}$ the joint probability that data dk = i and state Sk = m conditioned on the received binary sequence R_1^N, observed from time k = 1 through some time N, is

$$\lambda_k^{i,m} = P\,(dk = i,\, Sk = m \mid R^N{}_1)$$

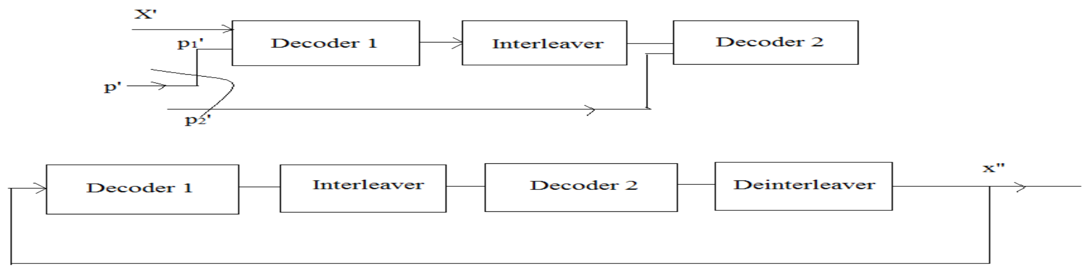
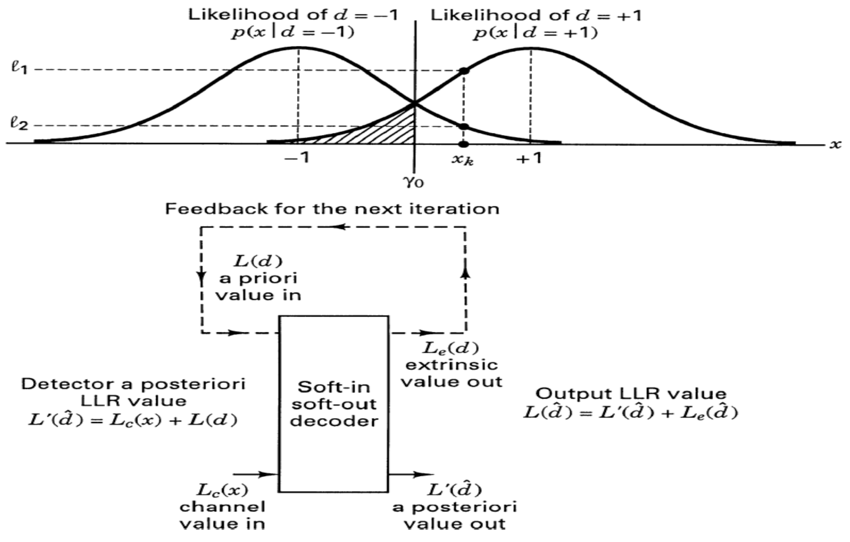


Fig. 6 Turbo decoder



Likelihood of $d = -1$
$p(x \mid d = -1)$
Likelihood of $d = +1$
$p(x \mid d = +1)$
ℓ_1
ℓ_2
-1
x_k
$+1$
x
γ_0
Feedback for the next iteration
$L(d)$ a priori value in
$L_e(d)$ extrinsic value out
Detector a posteriori LLR value $L'(\hat{d}) = L_c(x) + L(d)$
Soft-in soft-out decoder
Output LLR value $L(\hat{d}) = L'(\hat{d}) + L_e(\hat{d})$
$L_c(x)$ channel value in
$L'(\hat{d})$ a posteriori value out

Fig. 7 Soft in Soft Out decoder [20]

THE STATE METRICS AND THE BRANCH METRIC

$$\alpha_k^m = \sum_{j=0}^{1} \alpha_{k-1}^{b(j,m)} \delta_{k-1}^{j,b(j,m)}$$

$$\beta_k^m = \sum_{j=0}^{1} \delta_k^{j,m} \beta_{k+1}^{f(j,m)}$$

$$\delta_k^{i,m} = A_k \pi_k^i \exp\left[\frac{1}{\sigma^2}\left(x_k u_k^i + y_k v_k^{i,m}\right)\right]$$

And log likelihood ratio

$$\Delta(d_k) = \pi_k \exp\left(\frac{2x_k}{\sigma^2}\right) \frac{\sum \alpha_k^m exp \frac{(y_k y_k^{1m})}{\sigma^2} \beta_{k+1}^{f(1,m)}}{\sum a_k^m \; exp \frac{(y_k y_k^{0m})}{\sigma^2} \beta_{k+1}^{f(0,m)}} = \pi_k exp \frac{2x_k}{\sigma} \pi_k^{\alpha}$$

$$L(\hat{d_k}) = L(d_k) + L_e(x^k) + L_e(\hat{d_k})$$

RESULTS AND DISCUSSIONS

This whole system is simulated using MATLAB code with recursive convolutional encoder random interleaver and iterative viterbi decoder in which different parameters are used and the following results are simulated. These results are simulated using following parameters
No of iterations = 5
Eb/N0 ranges from 0 to 10
Signal to noise ratio is 0.7 dB with these the following results are obtained which are showing the BER ie bit error rate vs Eb/N0.

As from the Fig. 8, it is depicted that as the number of iterations are increasing the BER is decreasing. In the theoretical bound for Eb/N0 4 the BER is 10^{-2} whereas for the fourth iteration at Eb/N0 4 the BER reduced to 10^{-4}. Thus it is showing as the no of iterations are increasing the BER is decreasing. Also, the curve gets smoother with increase in number of iterations. The following screen shot shows the error matrix which obtained after the simulation of the whole program.
Error matrix

$$\begin{pmatrix}
531 & 354 & 331 & 339 & 347 \\
298 & 156 & 138 & 123 & 127 \\
138 & 43 & 39 & 36 & 35 \\
59 & 10 & 5 & 5 & 4 \\
17 & 1 & 1 & 1 & 1 \\
2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

This error matrix also showing as the program is getting iterated more times the error is decreasing.

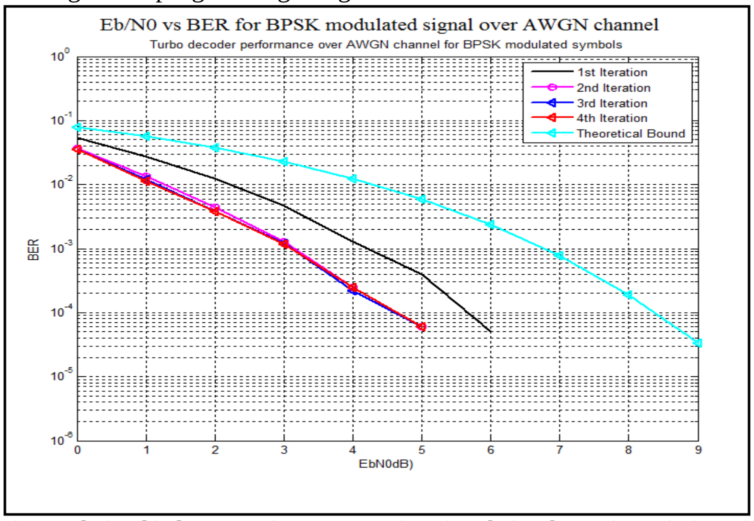


Fig. 8 Relationship between Bit Error Rate (BER) and Signal to Noise ratio (E_b/N_o)

CONCLUSION

Despite of other technologies being used today turbo codes have been emerged as one of the most promising technology in wireless and satellite communication systems. A practical communication system is being designed and simulated using MATLAB codes and the results are discussed in which BER curve and error matrix is obtained.

Acknowledgement
Authors are thankful to the department of Electronics Technology for providing necessary research facilities.

REFERENCES

[1] LH Abderrahmane, Design of a New Interleaver using Cross Entropy Method for Turbo Coding, *The Institution of Engineering and Technology*, **2013**, 7, 828–835.
[2] C Shannon, Communication in the Presence of Noise, *Proceedings IRE reprinted Proceedings IEEE*, **1949,** 86.
[3] A Glavieux and C Berrou, Turbo-Codes and High Spectral Efficiency Modulation, *Proceedings of IEEE International Conference on Communications*, New Orleans, USA, **1994**, 645-649.

[4] S Kerouédan, Three Dimensional Turbo Codes and the Performance Enhancement, *Springer, EURASIP Journal on Wireless Communications and Networking* **2013**, 32, 1678-1685.
[5] L Perez, J Seghers and DJ Costello, A Distance Spectrum Interpretation of Turbo Codes, *IEEE Transactions on Information Theory*, **1996,** *42*, 1698–1708.
[6] D Spielman, Linear-Time Encodeable and Decodable Error-Correcting Codes, *IEEE Trans Inform Theory*, **1996,** 42, 1723–1731.
[7] J Kim and J McLaughlin, Rate-Compatible Puncturing of Low-Density Parity-Check Codes, *IEEE Transactions on Information Theory*, **2011,** 50, 2824–2836.
[8] S Benedetto, D Divsalar, G Montorsi and F Pollara, Parallel Concatenated Trellis Coded Modulation, *Proceedings of IEEE International Conference on Communications*, **1996,** 974-978.
[9] J Hagenauer and P Hoeher, A Viterbi Algorithm with Soft-Decision Outputs and its Applications, *Proceeding of GLOBECOM*, **1989**, 1680-1686.
[10] S Pietrobon, Implementation and Performance of a Turbo/MAP Decoder, *International J Satellite Communications*, **1998**, 15**,** 23-46
[11] OY Takeshita and DJ Costello, New Deterministic Interleaver Designs for Turbo Codes, *IEEE Trans Information Theory*, **2000,** 46, 1988–2006.
[12] R Pyndiah, A Picart and A Glavieux, Performance of Block Turbo Coded 16-QAM and 64-QAM Modulations, *Proceedings of GLOBECOM'95*, **1998,** 1039-1043.
[13] J Sun and OY Takeshita, Interleavers for Turbo Codes using Permutation Polynomials Over Integer Rings, *IEEE Transactions on Information Theory*, **2005,** 51, 1567-1572.
[14] L Perez, J Seghers and D Costello, A Distance Spectrum Interpretation of Turbo Codes, *IEEE Trans Information Theory*, **1996,** 42, 1698–1709.
[15] S Benedetto and G Montorsi, Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes, *IEEE Trans Information Theory*, **1996,** 42, 409-428.
[16] K Kwon, Design of Rate-Compatible Block Turbo Code with a Low-Degree Generating Polynomial, *Springer Wireless Pers Commun*, **2014**, 77, 1615–1628.
[17] C Roth, Efficient Parallel Turbo-Decoding for High-Throughput Wireless Systems, *IEEE Trans Inf Theory*,**2000** 23,1505-1511
[18] D Divsalar and F Pollara, Multiple Turbo Codes for Deep-Space Communications, *The Telecommunications and Data Acquisition Progress Report 42-121*, **1995,** 66-77.
[19] L Perez, J Seghers and D Costello, Performance Enhancement of Modified Turbo Codes with Two-Stage Interleavers, *IEEE Trans Information Theory*, **1996,** 42, 1698–1709.
[20] B Seklar, Fundamental of Turbo Codes, web: http://ptgmedia.pearsoncmg.com/images/art_sklar3_turbocodes/elementLinks/art_sklar3_turbocodes.pdf