International Journal of Engineering Research and General Science Volume 3, Issue 4, July-August, 2015
ISSN 2091-2730

Implementation and Performance Analysis of Convolutional Encoder and Viterbi Decoder Using Matlab

Manika Pandey[#1], Vimal Kant Pandey [*2]
[#1] M.Tech scholar, ECE, DIT University, Dehradun, India

[*2]Assistant Professor/ECE, DIT University, Dehradun, India
[1]manika.mann@gmail.com

Abstract— During the transmission of data over channel, the data gets affected by the noise, which leads in reducing the effectiveness of system. To obtain the effective output, error detection and correction technique is achieved using error control codes providing Forward Error Correction technique which plays an important role in wireless networking facilitating the effective communication, thereby limiting the noise. The paper illustrates the convolution encoding with the decoding based on Viterbi algorithm. The performance of the code is analysed in terms of Bit Error Rate in comparison with the un-coded data. The complexity of the decoder increases with the increase in constraint length. The simulation was performed using BPSK modulation scheme in symmetric Additive White Gaussian Noise channel in MATLAB.

Keywords— Convolution Encoder, Viterbi Decoder, Forward Error Correction (FEC), AWGN and BPSK, Bit Error Rate (BER).

INTRODUCTION

The communication plays an important role in the phenomenal growth of World Wide Web [1]. During the transmission of data from one end to another, the data gets corrupted due to the occurrence of noise or interferences in the transmission channel. Hence, the error control becomes necessary for the effective output at the receiver end. Channel coding is a technique involving detection and correction of errors, resulting in reduction of noise in the channel. Forward Error Correction (FEC) is the error correcting scheme which automatically detects and corrects errors by adding redundant bits to the message signal, thereby increasing the bandwidth of signal.

The two main forms of FEC techniques are, Block Codes and Convolutional Codes [2]. The input in form of binary are fed into the modulator which maps it into the desired digital modulation waveform. Further, the waveform passes through the channel where the Additive White Gaussian Noise (AWGN) is added to it. The presence of AWGN corrupts the data which results in noisy signal. The decoding scheme detects and corrects the errors in it.

The block codes are applied only to the blocks of data which corrects burst of errors, whereas the convolutional codes can be applied to continuous data streams as well as to the blocks of data which corrects random errors [3]. The output of convolutional encoder depends on present and the previous output. With the increase in number of memories of convolutional encoder, the complexity of the device increases leading to the increase in computation [4].

The performance criterion during the digital data transmission is determined by Bit Error Rate (BER), i.e. Number of error bits/ total transmitted bits. The relation between the signal and noise is explained by Signal-to-noise ratio (SNR), i.e. signal power/ noise power, which is inversely proportional to BER.

This brief is organized as follows. Section 2 describes the theoretical background of the considered code. Based on the proposed method section 3 demonstrate the simulation and result synthesis based on un-coded and coded soft decision decoding. Finally, we conclude the brief in section 4.

THEORETICAL BACKGROUND

Convolutional codes

The convolution codes deals with the random errors. It is a forward error correction technique which involves the addition of redundant bits which determines the generated error due to the presence of noise in the channel. The three parameters defined in convolutional codes are, n, k, and m. The 'n' is the coded sequence bits obtained after encoding, 'k' is the information bits, and m is the memory elements used. The information bit 'k' is the continuous stream of data [5]. The code can also be specified by (n, k, L) parameter, where L (constraint length) is, $L= k(m-1)$. The decoding in convolution codes is done in two methods, hard decision decoding as well as soft decision decoding.

1) Convolution Encoder: The length of constraint in the convolutional encoder is fixed. Each of the input bit enter the shift register. At each input, the bit is shifted into the left most stage and the bits existing previously, are shifted one position towards right. The process continues until the data arrives at the input of the encoder. The outputs are obtained by module-2 adders which are used with the shift registers. The bandwidth efficiency of the code is measured in terms of code rate, i.e. the no. of input bits (k) / no. of output bits (n). The value of k and n, ranges between 1 to 8, and m from 2 to 10. For resetting the registers, zero bits are appended in the message. The code rate may fall below *k/n*, as the added bits do not contain any information into it. The most essential parameter of the encoder is generator polynomial. The generator polynomial for convolutional codes is determined by making the selection of bits in memory registers which are to be added, so as to generate the output bits for the output bit.

The convolution's encoding circuit is sequential; therefore its operation can be described with the help of state diagram as the state of encoder itself is defined as the shift register content of itself. The three alternative methods used for describing convolutional code are Tree diagram, Trellis diagram and State diagram.

For the construction of convolutional encoders, m boxes representing memory elements are drawn and then connected with the module-2 adders, which represents *n* output bits. The memory registers and module-2 adder connections are made using the bits which specify the generator polynomial [6].

2) ***Convolutional Decoder:*** This algorithm was devised and analysed by Viterbi [7]. In this algorithm maximum likelihood decoding is performed which is defined as process which decreases the computational load. It does so by taking the average of a particular structure in the code trellis. This advantage of this algorithm over brute-force decoding is that the complexity is not defined in terms of the number of symbols in the encoded sequence [8].

The resemblance between the received symbol and transmitted symbol is measured by hamming distance and the paths which are not suitable for maximum likelihood are rejected by this algorithm. If there is more than one path that emerges from the one particular state, then the state having the lowest path metric is selected and this path is called the surviving path [9]. Thus, for every state this process of selecting the surviving path is done. By this way, the decoder proceeds deeper into the trellis, assembling results by rejecting the paths having high metric. This early elimination of the paths with high metrics minimizes the complexity of decoder.

The decoding algorithm uses two metrics, the branch metric (BM) and path metric (PM). Via branch metric we measure the "distance" between transmitted and received data, and in trellis it is defined for each arc [10]. In hard decision decoding, decoding of received bit sequence is performed. Whereas, the process in which the voltage samples are decoded directly before they are digitized, is known as soft decision decoding. In hard decision decoding, the Hamming distance between expected parity bits and received ones is the branch metric.

The path metric value is a value which is associated with a state in trellis, means a value associated with each node. In case of hard decision decoding, it corresponds in trellis, hamming distance over the most likely path from initial state to the current state. Whereas the 'Most Likely' corresponds to the path which has smallest Hamming distance between initial and current state which between the two states, is measured over all the paths which are possible. The path which has smallest Hamming distance minimizes total number of bit errors, and is most likely when bit error rate is low. In Viterbi algorithm, key insight is that the receiver is able to compute the path metric for a (state, time) pair which is incrementally using path metrics of the previously computed states and branch metrics. When whole of the input sequence is processed, the decoder selects a path which has the minimum metric and then output it as the result of decoding.

SIMULATION AND RESULT SYNTHESIS

The paper presents the Convolutional coding and Viterbi decoding, along with binary phase-shift keying (BPSK) modulation. A Convolution encoder with (2, 1, 5) parameter is used. Where 2 is the length of message, 1 is the information bit, and 5 is the Length of Constraint. The performance of the code is analysed in terms of Bit Error Rate (BER). The performance of convolutional code is evaluated with respect to the un-coded data. The coding rate for convolutional code is considered as ½.

Figure 1 shows the BER plot for the un-coded data when the number of bits were 1000. The figure shows that the un-coded data provides the BER of 0.08 for 0 dB SNR value. Figure 2 shows the BER plot for the convolutional codes for the same value of SNR, the BER was found to is 0.03. The performance of un-coded data is poor in comparison to the convolutionally coded data. The error correcting capability of the code improves with the increase in the number of bits, keeping the constraint length same. Figure 3 shows the BER performance for un-coded data, when the number of bits was increased to 10,000 and figure 4 shows the convolutionally decoded result for the same number of bits. As shown in figure 4, the performance of convolutionally decoded data improves when the bits were increased from 1000 to 10,000. More will be the generator polynomials in the code, lesser will be the Bit Error Rate, resulting in improvement in error correction.

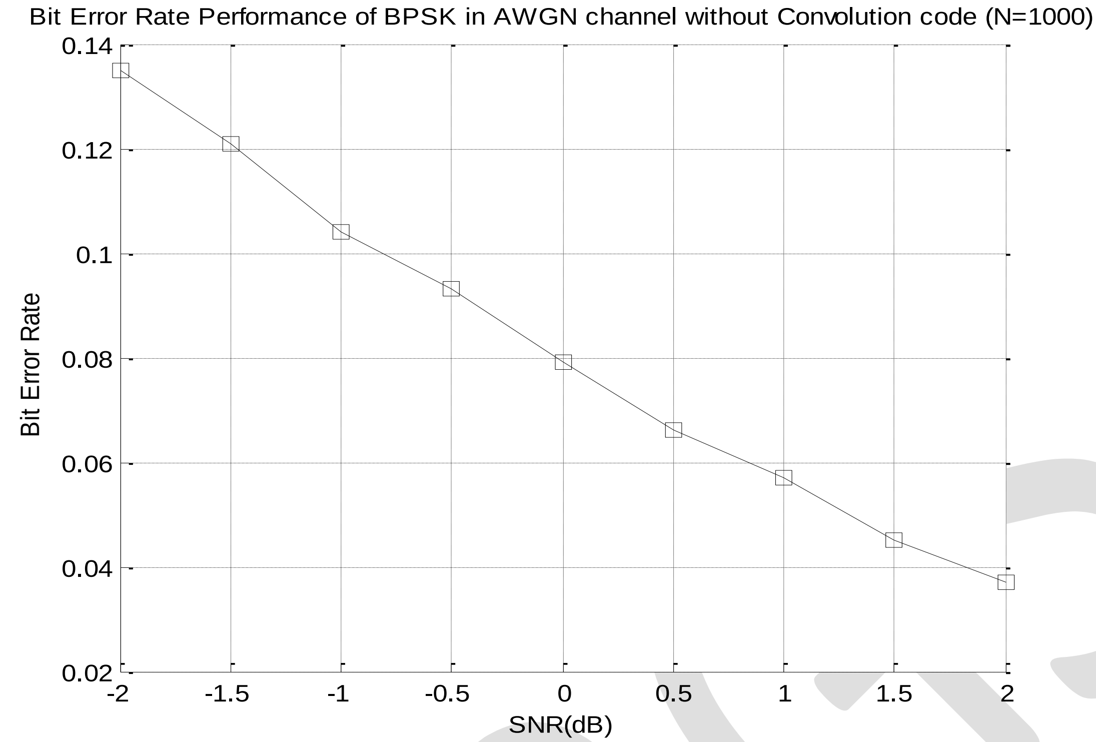

Fig.1. Bit Error Rate Performance for Un-coded Data (N=1000).

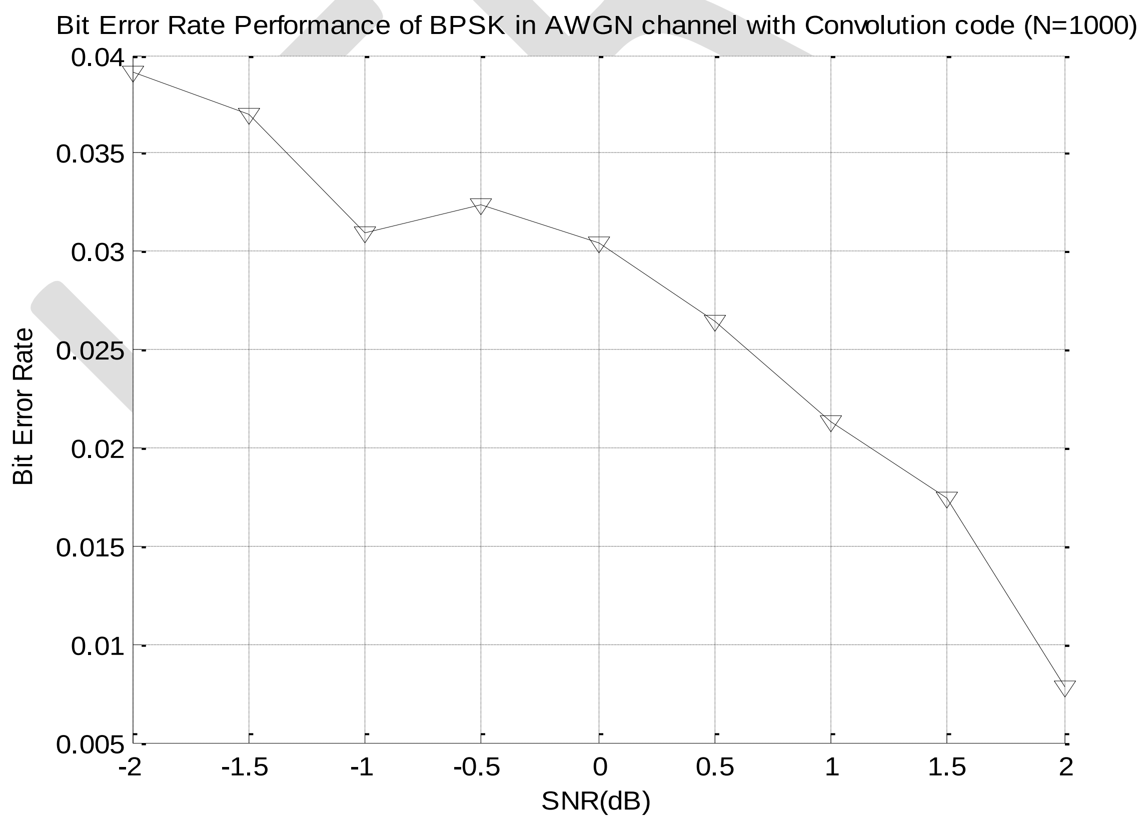

Fig.2. Bit Error Rate Performance for Convolutionally Coded Data (N=1000).

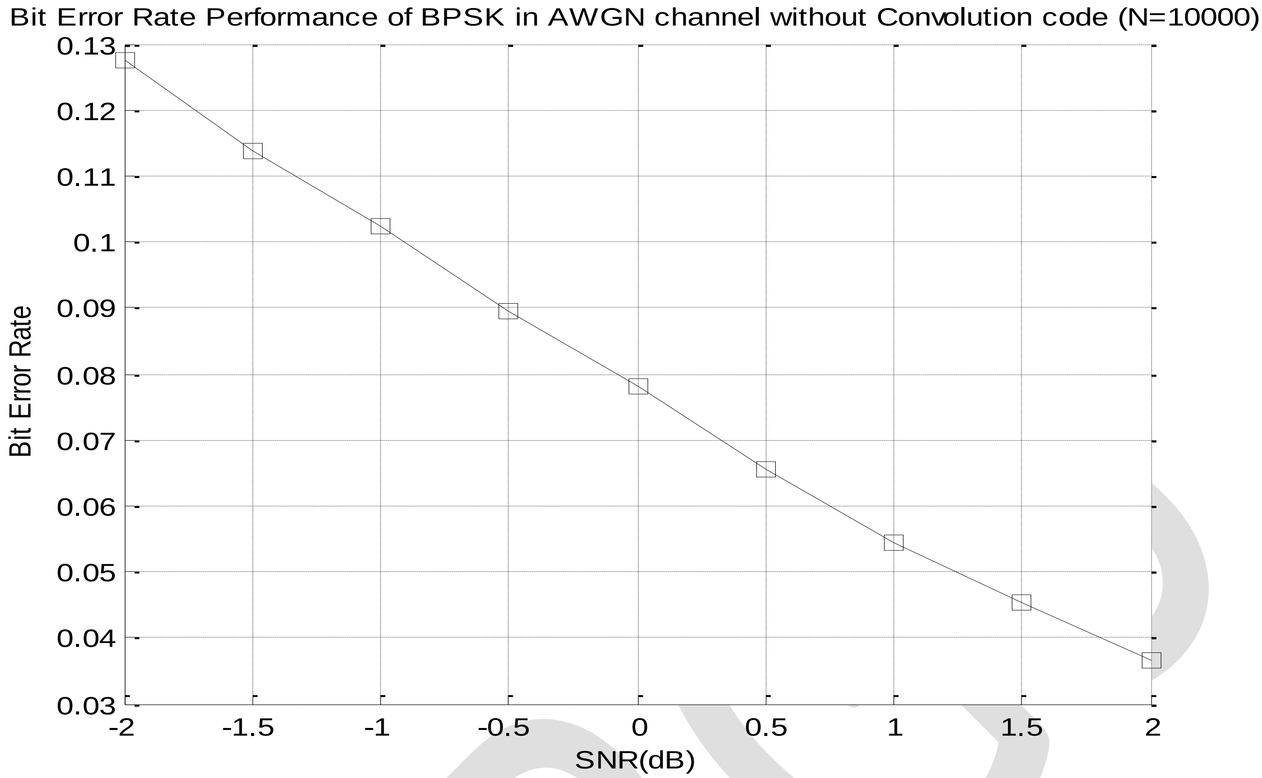

Fig.3. Bit Error Rate Performance for Un-Coded Data (N=10,000).

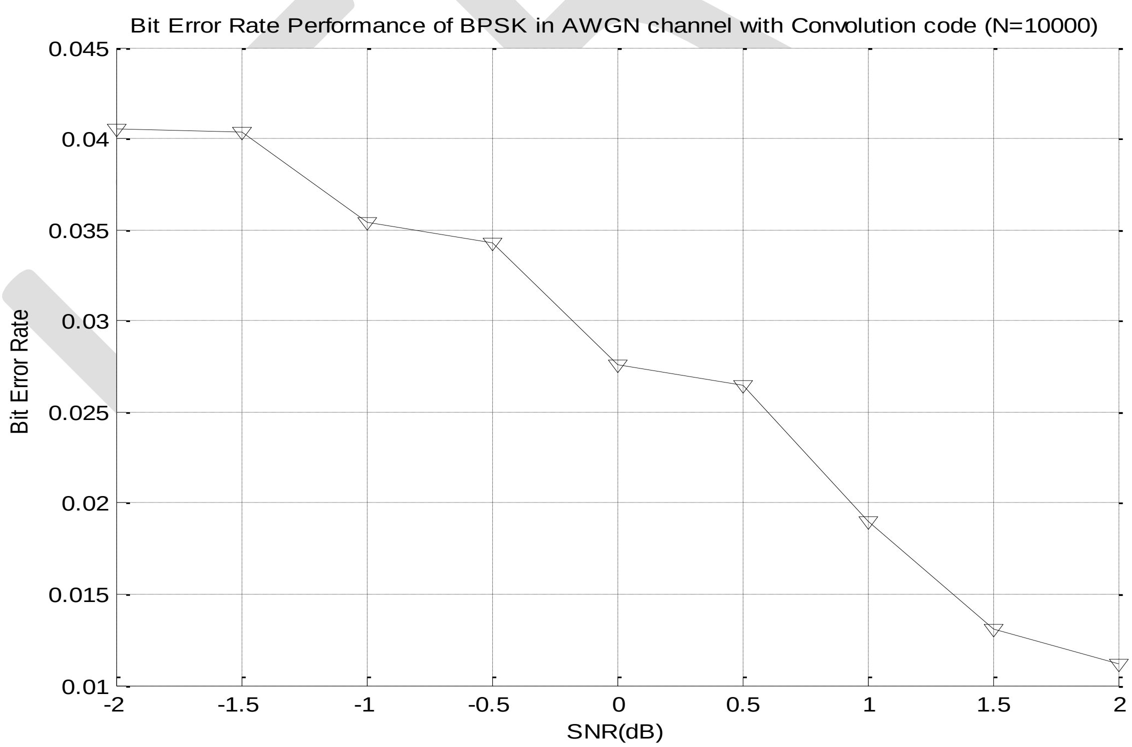

Fig.4. Bit Error Rate Performance for Convolutionally Coded Data (N=1000).

CONCLUSION

With the evolvement of hard decision decoding, the convolutional code with increased bits performs in a far better manner in comparison to un-coded data. Along with the increased data bits, code rate must be less, as the code rate is the ratio of information bits to the data bits.

Via this paper we present the deep and clear understanding of convolutional codes with hard decision decoding making them simpler and easier to implement. The simulation showed that the convolution code performs far better than un-coded data with increased number of bits. The BER performance degrades with the increases in code rate. So, it's better to consider coding rate as small as possible [12].

The present work can be further extended by finding results for hard as well as soft decision decoding and hardware implementation of them using FPGA.

REFERENCES:

[1]- J. Proakis, Digital communications, NY: McGraw Hill, 2001.

[2]- R.C. Bose, D.K. Ray-Chaudhuri,"On a class of error correcting binary group codes", Inf. Cntrol, 3, pp. 68-79, March 1960.

[3]- J.A.Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communications,"*IEEE Transactions on Communication Technology*, vol. com-19, pp. 835-848, October 1971.

[4]- G.C.Clark, Jr., and J. B. Cain,*Error-Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

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

[6]- A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,"*IEEE Transactions on Information Theory*, vol.IT-13, pp. 260-269, April, 1967.

[7]- J. B. Cain, G. C. Clark and J. M. Giest, "punctured convolutional codes and simplified maximum likelihood decoding", IEEE transactions on Information Theory, vol. IT 25, No. 1, PP 97-100, January 1979.

[8]- Sven Riedel, "symbol by symbol MAP decoding algorithm for high rate convolutional codes that use reciprocal dual codes", IEEE Journal on Selected Area in Communications, Vol.16, No. 2, February 1998.

[9]- J.A.Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communications, "*IEEE Transactions on Communication Technology*, vol. com-19, pp. 835-848, October 1971.

[10]- F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York, NY: Elsevier Science Publishing Company, Inc., 1977.

[11]- G. D. Farney, lr., "The Viterbi algorithm", *Proc. IEEE, vol. 61, pp. 268-278*, Mar. 1973.

[12]- Nabeel Arshad, Abdul Basit, "Implementation and Analysis of Convolutional Codes Using matlab", international journal of multidisciplinary sciences and engineering, vol. 3, no. 8, august 2012