ESJ Natural/Life/Medical Sciences

Implementation of Convolutional Codes with Viterbi Decoding in Satellite Communication Link using Matlab Computational Software

Ayibapreye K. Benjamin (PhD)
Collins E. Ouserigha (PhD)
Niger Delta University, Nigeria

Doi:10.19044/esj.2021.v17n34p1

Submitted: 08 June 2020
Accepted: 22 September 2021
Published: 30 September 2021

Cite As:
Benjamin A.K. & Ouserigha E.C. (2021). *Implementation of Convolutional Codes with Viterbi Decoding in Satellite Communication Link using Matlab Computational Software.* European Scientific Journal, ESJ, 17(34), 1. https://doi.org/10.19044/esj.2021.v17n34p1

Abstract

This paper describes the design of convolutional encoders encoding broadband data with coding rates (r) 3/4 and 5/6 at the transmitter, and the Viterbi decoders for the decoding of the encoded data with same coding rates at the receiver. This design uses BPSK (Binary Phase Shift Keying) modulating scheme in an AWGN (Additive White Gaussian Noise) channel in analysing the BER (Bit-error rate) performance of next generation broadband wireless access systems. The constraint length of the convolutional encoder is K=7, with number of states n=2k-1, and the design was implemented using Matlab Computational Software. The higher coding rates of the convolutional encoders were achieved using puncturing codes and de-puncturing codes for the corresponding Viterbi decoders. In satellite network design, it is vital to compliment the system design with the BER requirement of the network. Plots of BER against SNR ( ) were made for the various coding rates and for data rates of 50000 bits and 12000 bits. The results were analysed based on comparison of coding gain obtained for the different coding rates, theoretical BER and calculated BER. In this paper, Matlab simulation of the convolutional encoder and Viterbi decoder shows improved coding gain for higher data rates and data sequences.

Keywords : Convolutional Encoder, Viterbi decoder, BER, SNR, BPSK

1.0. Introduction

Information transmission over a satellite communication system is prone to degradations in the quality of the information. In analogue links, the degradation is a measure of decrease in signal-to-noise ratio (SNR), while in digital links, this is measured in terms of the bit error rate (Dholakia, 2012).

Over the years, communication systems have provided tremendous changes in data transmission and reception over noisy communication channels such as Rayleigh channels, AWGN (Additive White Gaussian Noise) channels, and other channel models. Digital transmission of voice and data across a communication channel is obviously carried out through digital modulation, mainly Binary Phase Shift Keying and Quadrature Phase Shift Keying (Wong, 2009; Shannon, 1948). Error detection and correction schemes are very reliable in the reduction of noise and impairments at the receiver of digital satellite links. In Nigeria, there is recent emergence of voice and data services by GSM (Global System for Mobile Communication) service providers such as Globalcom, MTN, Etisalat, and Starcom. The transmission of voice and data across communication channels comes with some challenges such as the modulation and demodulation schemes utilized for these applications, the multiplexing techniques, error correction and detection schemes, and performance analysis of the communication link design. The aforementioned challenges are consequences of bandwidth limitation. Bandwidth is expensive and limited. This makes it very important to utilize bandwidth in terms of the number of users that can accommodate specific amount of it (Hamming, 1950; Muller, 1954).

In the transmission of digital data, error checking codes are very significant and the complexity and accuracy of these codes has to be balanced against the corresponding available bandwidth of the channel (Berlekamp, 1987; Huffman, 2010). Next Generation Broadband Wireless Access Systems (IEEE STD 802.16-2009) is a mobile communication system that promises efficient transmission of voice and data (internet). Hence, there is need to design error detection and correction schemes to enhance performance monitoring of the receiver for efficient reconstruction of the original transmitted data. Some communication channels are unreliable which makes a channel to be prone to noise and errors, but in this design, we will be looking at AWGN (Additive White Gaussian Noise) channel. Good error control performance can be met if the chosen error correction technique is based on the communication channel. Additive White Gaussian Noise (AWGN) is inherent in all communication channels. Forward error correction (FEC) schemes are implemented in the transmitter to encode the data stream and receiver to decode and correct the data steam in error, thereby minimizing the bit error rate (BER) to enhance system performance (Wozencraft, 1957; Massey, 1963). Reed-Solomon (RS) can be used in hardware at very high data rate because its decoding algorithm is relatively low. Hence, it seems to be an ideal scheme for any application, but it performs poorly in AWGN channel. Common channel models comprise of memory-less models where there is certain probability of occurrence of random error, and dynamic models where errors are prone to burst errors (Viterbi,1971; Proakis, 2008).

The trauma of removal of noise, interference, and impairments in the channels of communication networks still remains a major challenge to network designers. This is a major concern as the efficiency and reliability of a communication system is to transmit and receive digital signals error-free. The transmission of analogue signal such as telephone channels and television may be put into digital form and then converted back to analogue form for routing to the end users. This process may be costly in terms of bandwidth, but it offers improved noise performance and increased immunity to interference. In the transmission of information mainly over a satellite link, there is always some degradation in the quality of information. In analogue systems, the degradation takes the form of a decrease in signal-to-noise ratio (SNR), but in digital systems we measure degradation of the information content of a signal in terms of the bit error rate (BER) (Ristenbatt, 1973; Pratt, 1989).

BER is the figure of merit for a digital radio link, and it is also called the bit error probability (PB). This is the probability that a bit sent over the link will be received incorrectly (i.e., that a 1 will be read as 0 or vice versa) or, alternatively, the fraction of a large number of transmitted bits will be received incorrectly. Like a probability, it is usually stated as a single number ---- for example 1×10^{-4} or $.0001$. By using phase shift keying, usually coherent QPSK, we can again trade bandwidth for signal power and achieve good bit error rates (BER) with high signal-to-noise ratio $(\frac{E_b}{N_o})$ (Van Trees, 1979; Pratt, 1989). In the process of reducing error in communication channels, error detection and correction encoding also known as forward error correction (FEC) plays a significant role. In digital communication systems, we can add extra redundant bits to our data stream to detect an occurrence of error in the data, and also where the particular bit or bits have been corrupted.

2.0. System Overview Of The Viterbi Decoder

Figure 2.1 illustrates an overview of the model of the convolutional encoder with Viterbi decoder.

ISSN: 1857-7881 (Print) e - ISSN 1857-7431

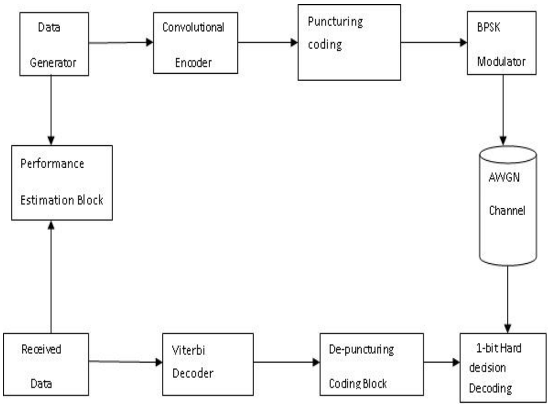

Figure 2.1. Overview of the Viterbi Decoding Model

Binary Random Data Generator

This is the first step involved in modelling of a convolutional encoder with viterbi decoder for next generation broadband wireless access system using Matlab computational software. A random generator was used to generate a uniformly distributed binary data bits of N length using the Matlab random generator rand(), and data rounded off to avoid decimal values with minimum values of zero and maximum values of 1. Prior to the generation of the binary random data, the number of states of the encoder was defined and the constraint length and the number of shift registers was specified. The encoder was set at an initial state of zero using zeros().

Design of Convolutional Encoder

The convolutional encoder is implemented using shift registers and modulo-2- addition of the inputs of the shift registers. The randomly generated data are used as input to the shift registers of the convolutional encoders. The shift registers are set at a zero initial state. A Matlab code is written to insert one bit of the randomly generated data at a particular instant to the shift registers, and for every input data inserted, there is a shift of content in the shift registers from left to right, which result in a change of state of the shift register. For a rate 1/2 encoder and for every input bit inserted into the shift registers, we obtain two (2) output bits. The output follows the path of the generator polynomial of the convolutional encoder (Shamsi et al., 2013). In this project design, with a constraint length of 7, the generator polynomials in octal form are $G_1 = 171, G_2 = 133$. The generator polynomials are written in binary form as 1111001 and 1011011 respectively. The number of states of the encoder is 64 with m = 6, where m is the number of shift registers. The encoder rates required for this design are rate 3/4 and 5/6, and they are implemented using punctured codes. The essence of the use of the puncturing code is to obtain higher data rates. This block deletes some predetermined convolutional codes bits in the encoded sequences. These discarded bits are neither transmitted nor considered in the decoding procedure. Two rates are required for this design; the first is rate 3/4, puncturing is required in this rate where the third and the fifth bits of each 6bits are discarded. The last rate is 5/6, where the seventh and eighth bits of each 8bits are deleted (Suganya et al., 2013).

Mapping of Channel Symbols to Signal

Mapping the one/zero output of the convolutional encoder onto an antipodal baseband signalling scheme is simply a matter of translating zeros to +1s and ones to -1s. This can be accomplished by performing the operation b = 1-2a on each convolutional encoder output symbol.

BPSK Modulation Scheme

BPSK is the simplest form of PSK. It employs two phase which are separated by 180° and so it can also be termed 2-PSK. For example, the two constellation points are positioned on the real axis at 0° and 180°.

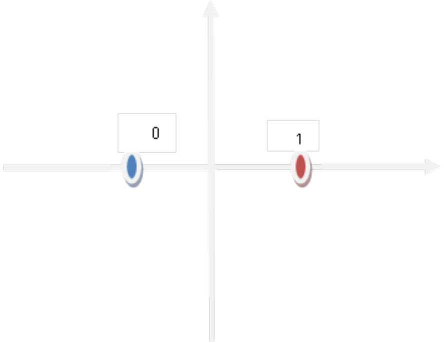

Figure 2.2. Constellation Plot for BPSK

We can transmit binary data with the signals below:
For binary ‘0’;

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) \tag{1}$$

For binary ‘1’;

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) \tag{2}$$

Addition of AWGN with the Transmitted Channel Symbols

It is of necessity that AWGN is added to the signal to be transmitted, as noise is inherent in communication channels. Adding noise to the transmitted channel symbols produced by the convolutional encoder involves generating Gaussian random numbers. It also entails scaling the numbers according to the desired energy per symbol to noise density ration. E_s/N_0 adds the scaled Gaussian random numbers to the channel symbol values (Katta, 2014).

For the uncoded channels, $E_s/N_0 = E_b/N_0$, since there is one channel per bit. However, for the coded channel, $\frac{E_s}{N_0} = \frac{E_b}{N_0} + 10log_{10}(\frac{k}{n})$. For example, for rate 1/2 coding, $\frac{E_s}{N_0} = \frac{E_b}{N_0} + 10log_{10}\left(\frac{1}{2}\right) = \frac{E_b}{N_0}\frac{E_b}{N_0} - 3.01dB$. Similarly, for rate 2/3 coding, $\frac{E_s}{N_0} = \frac{E_b}{N_0} + 10log_{10}\left(\frac{2}{3}\right) = \frac{E_b}{N_0} - 1.76dB$.

The Gaussian random number generator is the only interesting part of this task. C only provides a uniform random number generator using the function, rand(). In order to obtain Gaussian random numbers, we take advantage of relationships between uniform, Rayleigh, and Gaussian distributions.

Given a uniform random variables U, a Rayleigh random variable R can be obtained by:

$$R = \sqrt{2.\sigma^2}\ln(\frac{1}{1-U}) = \sigma\sqrt{2}.\ln(1/(1-U) \tag{3}$$

Where σ^2 is the variance of the Rayleigh random variable. Given R and a second uniform random variable V, two Gaussian random variables G and H can be obtained by:

$$G = R\cos V \tag{4}$$

$$H = R\sin V \tag{5}$$

In the AWGN channel, the signal is corrupted by additive noise, n(t), which has the power spectrum $N_0/2$ watts/Hz. The variance σ^2 of this noise is equal to $N_0/2$. If we set the energy per symbol, Es equal to 1, then

$$\frac{E_s}{N_0} = \frac{1}{2\sigma^2} \tag{6}$$

Where;

$$\sigma = \sqrt{1/(2\cdot\left(\frac{E_s}{N_0}\right))} \tag{7}$$

Quantising the Received Channel Symbol

An ideal viterbi decoder would work with infinite precision, or at least with floating-point numbers. In practical systems, there is need to quantize the received channel symbols with one or a few bits of precision in order to reduce the complexity of the viterbi decoder, and not to mention the circuits that precede it. If the received channels symbols are quantized to one-bit precision

(<0V =1, ≥ 0V = 0), the results is called hard-decision data. If the received channels symbols are quantized with more than one bit of precision, the results is called soft-decision data. A viterbi decoder with soft decision data inputs quantized to three or four bits of precision can perform about 2dB better than the one working with hard-decision inputs (Sunanda et al., 2014).

It is important to note that the usual quantization precision is three bits, where more than three bits provides little additional improvement.

The selection of the quantizing levels is an important design decision because it can have a significant effect on the performance of the link. The following is a very brief explanation of one way to set those levels.

Lets assumes our received signals levels in the absence of noise are – 1V = 1, + 1V = 0. With noise, our received signals has mean +/-1 and standard deviation of:

$$\sqrt{\frac{1}{2\cdot\left(\frac{E_s}{N_o}\right)}}$$

Demodulation

BPSK demodulation of the binary data is done by simply translating the -1V and +1V representing 1’s (ones) and 0’s respectively back to its binary form. This can be achieved by setting a threshold at which values less than the threshold equals zero, or else it equals 1.

De-puncturing

It is of great necessity that we de-puncture the received signal in order to obtain the same bit sequence sent by the transmitter. According to the puncturing pattern, for those bits deleted, the de-puncturing block inserts a dummy ‘zero’ bit to make the data sequence as same as the convolutional coding sequence.

Performing Viterbi Decoding

A viterbi decoder uses the viterbi algorithm for decoding a bit stream that has been encoded using FEC based on a convolutional code. The viterbi decoder is used in many FEC applications and systems where data are transmitted and subject to errors before reception. The viterbi algorithm is commonly used in wide range of communications and data storage applications (Proakis, 2012). It is used for decoding convolutional codes, in the base band detection for wireless systems, and also for detection of recorded data in magnetic disk drives. The requirements for the viterbi decoder or viterbi detector, which is a processor that implements the viterbi algorithm, depend on the applications where they are used. The block diagram consists of viterbi decoder shown in Figure 2.3. The block diagram has the following modules: Branch Metrics, Add-Compact-Select (ACS), Register Exchange, Maximum Path Metric Selection, and Output Register Selection (Proakis, 2008).

3.0. Discussion Of Results

Performance of BER with Code Rate ¾

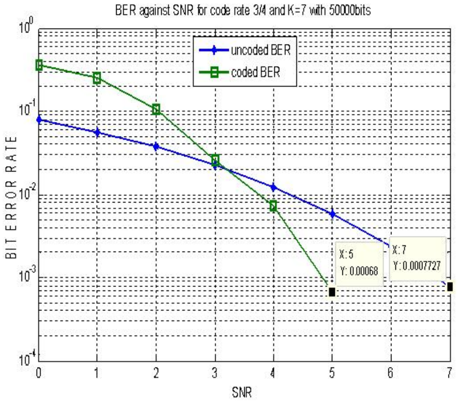

Figure 3.1. *A Plot of BER against SNR for K=7 and r =3/4 for 50000 bits*

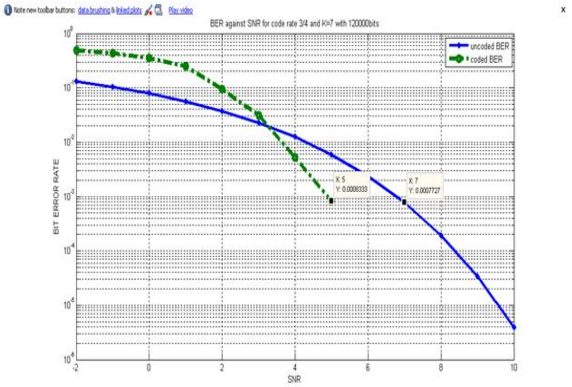

Figure 3.2. A Plot of BER against SNR for K=7 and r=3/4 for 12000 bits

The figures above represent the BER performance of rate 3/4 decoder at different data sequences of 10 kbits and 12 kbits respectively. The BER curve for rate 3/4 with data sequence of 10 kbits was plotted with SNR range of 0:1:7. The coding gain of Figure 3.2 is approximately 1 dB. There was a flat exponential decrease in BER at 0 dB, but with increase in SNR, there was a slight deviation in the path of the BER curve for the coded data. This illustrates that the coded BER curve decays faster at the initial point of the curve and suddenly becomes slower with increasing SNR (E_b/N_0). Figure 3.2 has a coding gain of 2 dB for SNR values of 5 dB to 7 dB. The BER range on the y-axis of the plot decreased from 10^{-4} to 10^{-6} which once again proves that increasing the SNR results to a reduction in the BER.

The figure below shows the curve of BER against SNR (E_b/N_0) for both uncoded and coded data. The SNR range was set between 1 dB and 7 dB, while the input data sequence of 50 kbits was used for the implementation of this design. The coding gain of this design is 2 dB which is very good considering the SNR range. For a code rate of 3/4 with data sequence of 50 kbits, we achieved a coding gain of 2 dB which is very close to the coding gain of 2.3 dB obtained by a higher rate of 5/6 with higher SNR range of -2 to 10 dB and data sequence of 12 kbits. The BER performance of the decoded data is better than the uncoded data as we can clearly see from the plot. The BER curve for the decoded data falls sharply above $10^{-3,}$ which is slightly above the BER value of the uncoded data. Thus, we are expected to obtain better BER for higher data sequence, higher SNR, and higher code rate.

Performance of BER with Code Rate 5/6

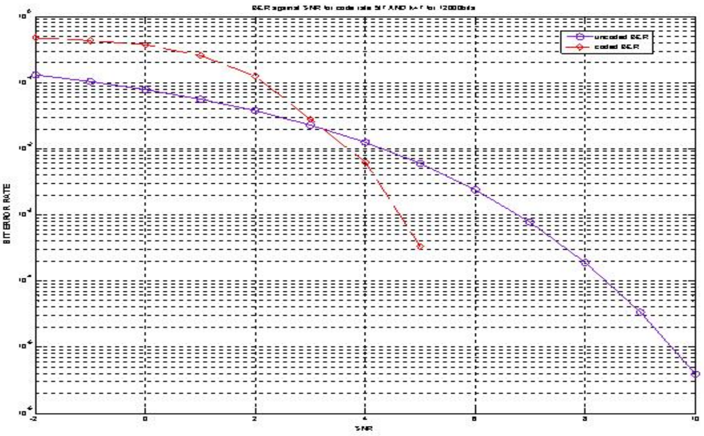

Figure 3.3. A Plot of BER against SNR for K=7 and r=5/6 for 12000 bits

Figure 3.3 shows the BER against SNR curve for 12000 bits data sequence. The SNR values ranges from -2dB to 10dB with single line spacing. Increasing the data sequence helps to improve the BER performance as we intend to reduce the speed of data transmission which results in lowering of the BER. It is obvious from Figure 4.1 and Figure 4.7 that increasing the data sequence improves the BER performance both for lower and higher code rates. The coding gain of the rate 5/6 encoder is about 2.3 dB. For low SNRs, the performance of the uncoded data is better than the coded data as the SNR (E_b/N_0) for uncoded data is greater than the SNR of a coded data. The SNR increases for higher coding rate. For instance, for rate 1/2 $E_s/N_0 = E_b/N_0 - 3.01$

ISSN: 1857-7881 (Print) e - ISSN 1857-7431

dB, while that of rate 2/3 is $E_s/N_0 = E_b/N_0 – 1.76$ dB. For 12000 bits input data sequence, the BER reduces to 10^{-6} with SNR range of -2:1:10.

Conclusion

This paper presents BER performance of a convolutional encoder with coding rates 3/4 and 5/6 on a wireless AWGN channel using BPSK modulation technique. The proposed convolutional encoder and viterbi decoder designs were simulated in MATLAB computational software. Results obtained show improved BER performance for code rates 3/4 and 5/6 with higher data sequences than uncoded data sequences. For higher data sequences, better coding gains were obtained as shown by the simulated results. Low BER was reported for higher SNR (E_b/N_0), which provides improved system performance and higher coding gains. Future works should explore hard and soft decision viterbi algorithm, which can be distinguished by hamming distance metric as soft decision algorithm cannot use hamming distance metric because of its limited resolution. Understanding of simple convolutional codes will help to expand further into high performance schemes of error control coding like Turbo codes.

References:

1. Berlekamp, E., Peile, R. & Pope, S. (1987). The application of error control to communications. IEEE Communications Magazine, 25(4), 44-57.
2. Dholakia, A. (2012). Introduction to convolutional codes with applications (Vol. 275). Springer Science & Business Media.
3. Hamming, R. W. (1950). Error detecting and error correcting codes. The Bell system technical journal, 29(2), 147-160.
4. Huffman, W. C. & Pless, V. (2010). Fundamentals of error-correcting codes. Cambridge university press.
5. Katta, K. (2014). Design of convolutional encoder and viterbi decoder using MATLAB. International Journal for Research in Emerging Science and Technology, 7.
6. Massey, J. L. (1963). Threshold decoding. MIT Press, Cambridge.
7. Muller, D. E. (1954). Application of boolean algebra to switching circuit design and to error detection. Transactions of the IRE professional group on electronic computers, (3), 6-12.
8. Pratt, T. & Bostian, C. W. (1989). Satellite Communications, John Wiley and Sons.
9. Proakis, J. (2008). Masoud Salehi Digital Communications.
10. Proakis, J. G., Salehi, M. & Bauch, G. (2012). Contemporary communication systems using MATLAB. Nelson Education.

11. Ristenbatt, M. P. (1973). Alternatives in digital communications. Proceedings of the IEEE, 61(6), 703-721.
12. Shamsi, M., Lolive, D., Barbot, N. & Chevelu, J. (September, 2019). Corpus Design Using Convolutional Auto-Encoder Embeddings for Audio-Book Synthesis in Interspeech (pp. 1531-1535).
13. Shannon, C. E. (1948). A mathematical theory of communication. Bell system technical journal, 27(3), 379-423.
14. Suganya, G. S. & Kavya, G. (April, 2013). RTL design and VLSI implementation of an efficient convolutional encoder and adaptive viterbi decoder. In 2013 International Conference on Communication and Signal Processing (pp. 494-498). IEEE.
15. Sunanda, A. & Remmanapudi, S. (2014). VLSI Implementation of Efficient Convolutional Encoder And Modified Viterbi Decoder. International Journal for Research in Applied Science & Engineering Technology (IJRASET), (3).
16. Van Trees, H. L. (1979). Ed., Satellite Communications, IEEE Press, New York.
17. Viterbi, A. (1971). Convolutional codes and their performance in communication systems. IEEE Transactions on Communication Technology, 19(5), 751-772.
18. Wong, Y. S., Ong, W. J., Chong, J. H., Ng, C. K. & Noordin, N. K. (November,2009). Implementation of convolutional encoder and viterbi decoder using VHDL. In 2009 IEEE Student Conference on Research and Development (SCOReD) (pp. 22-25). IEEE.
19. Wozencraft, J. M. (1957). Sequential decoding for reliable communication.