International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Analysis of Convolutional Codes with Viterbi Algorithm and Turbo Codes for Reliable Space Telemetry

Ch.Ravi kumar[1], B.Likhith Babu[2], B.Naga Sai Mounika[3], G.VenkataRamu[4], D.Sarika[5], G.N.B.S.Prasanna[6]

[1]Sr.Assistant Professor, Department of ECE, SIR C.R.Reddy College of Engineering, Eluru, AndraPradesh, India
[2,3,4,5,6]UGScholar, Department of ECE, SIR C.R.Reddy College of Engineering, Eluru, Andra Pradesh, India

ABSTRACT
Reliable data transmission is a key requirement in modern space communication systems, where telemetry and telecomm and data must be delivered accurately over long distances and noise-dominated channels. Convolutional codes, recommended by the Consultative Committee for Space Data Systems (CCSDS), are widely used in space communication systems for reliable telemetry transmission, and Turbo codes are also adopted in advanced space missions to achieve higher coding gain. This project mainly focuses on the performance analysis of convolutional codes with Viterbi algorithm and Turbo codes to highlight advanced error correction performance for space communication links. In this project, both convolutional coding with Viterbi decoding and Turbo coding techniques are implemented and analyzed using MATLAB. Random binary data is generated, encoded using convolutional and Turbo encoders, modulated using BPSK, and transmitted through an Additive White Gaussian Noise(AWGN) channel. At the receiver, the encoded data is decoded using the Viterbi algorithm for convolutional codes and iterative decoding for Turbo codes. The Bit Error Rate (BER) is calculated by comparing transmitted and received sequences, and BER versus Eb/No performance curves are plotted to evaluate the coding gain. The simulation results demonstrate that both coding techniques significantly improve transmission reliability compared to un coded systems.

Keywords: Viterbi algorithm, convolutional codes, BER, telemetry, turbo codes.

I. INTRODUCTION

The main function of a communication system is to transmit information from the source to the destination with sufficient reliability. In the last two decades, there has been an explosion of interest in the transmission of digital information mainly due to its low cost, simplicity, higher reliability and possibility of transmission of many services in digital forms.

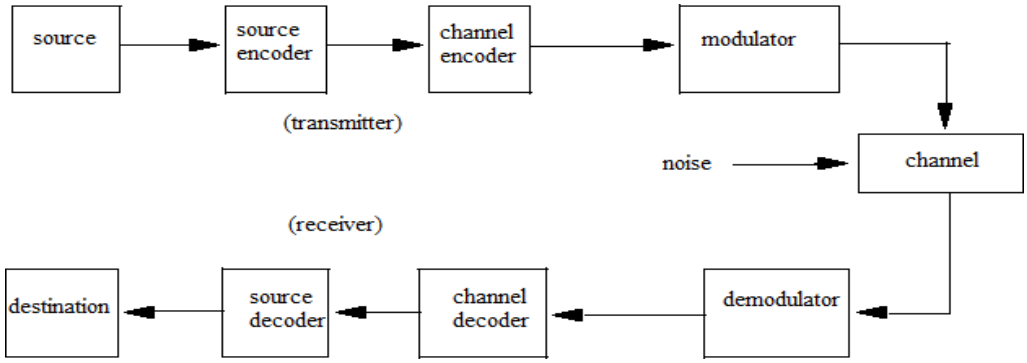


Fig1.1: A typical digital communication system

Figure 1.1 shows a simple block diagram of a digital transmission system. At the source, information suitable for transmission is produced. The input to this block is either analog or discrete. In the case of an analog input, appropriate processes, i.e. quantization, sampling and coding are performed so as to form a discrete signal. Discrete information obtained from the source block a certain sampling grate is then input to the source encoder block. In this block, symbol sequences are converted to binary sequences by assigning code words to the input symbols according to a specified rule and based on reducing the redundancy of the encoded data. Since redundancy has been removed from the information source, encoded information is sensitive to noise in transmission media. Hence, a channel encoder inserts redundancies into the source encoded data so as to protect the required signal against channel errors. Then, the modulator converts the input binary stream to a waveform compatible with the channel characteristics and provides suitable conditions for transmission of the signal. The remaining blocks of Figure1.1 perform in verse operations of their corresponding blocks at the transmitter to finalize extraction of the required signal at the destination.

Need for coding:

Backward error correction (BEC) requires only error detection: if an error is detected, the sender is requested to retransmit the message. While this method is simple and sets lower requirements on the code's error-correcting properties, it on the other hand requires duplex communication and causes undesirable delays in transmission.

Forward error correction (FEC) 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 errors occurred. Since FEC codes require only simplex communication, they are especially attractive in wireless communication systems, helping to improve the energy efficiency of the system.

II. TURBO CODES AND LLR ALGEBRA

A. Turbo Codes

Coding theory is the study of the properties of codes and their fitness for a specific application. Codes are used for data compression, cryptography, error-correction and more recently also for network coding. Codes are studied by various scientific disciplines—such as information theory, electrical engineering, mathematics, and computer science—for the purpose of designing efficient and reliable data transmission methods. This typically involves the removal of redundancy and the correction (or detection) of errors in the transmitted data.

There are essentially two aspects to Coding theory:

1. Data compression (or, source coding)
2. Error correction (or, channel coding).

These two aspects may be studied in combination. Source encoding attempts to compress the data from a source in order to transmit it more efficiently. This practice is found every day on the Internet where the common Zip data compression is used to reduce the network load and make files smaller. The second, channel encoding, adds extra data bits to make the transmission of data more robust to disturbances present on the transmission channel. The ordinary user may not be aware of many applications using channel coding. A typical music CD uses the Reed-Solomon code to correct for scratches and dust. In this application the transmission channel is the CD itself. Cell phones also use coding techniques to correct for the fading and noise of high frequency radio transmission. Data modems, telephone transmissions, and NASA all employ channel coding techniques to get the bits through, for example the Turbo code and LDPC codes.

Linear block codes:

Linear block codes have the property of linearity, i.e. the sum of any two code words is also a code word, and they are applied to the source bits in blocks, hence the name linear blocks codes. There are block codes that are not linear, but it is difficult to prove that a code is a good one without this property.

In coding theory, block codes refers to the large and important family of error- correcting codes that encoded at a in blocks. There is a vast number of examples for block codes, many of which have a wide range of practical applications. The main reason why the concept of block codes is so useful is that it allows coding theorists, mathematicians, and computer scientists to study the limitations of all block codes in a unified way. Such limitations often take the form of bounds that related iffe rent parameters of the block code to each other, such as its rate and its ability to detect and correct errors. Examples of block codes are Reed–Solomon codes, Hamming codes, Golay codes, and Reed–Muller codes. These examples also belong to the class of linear codes, and hence they are called linear block codes.

The block code and its parameters:

Error-correcting codes are used to reliably transmit digital data over unreliable communication channels subject to channel noise. When a sender wants to transmit a possibly very long data stream using a block code, the sender breaks the stream up into pieces of some fixed size. Each such piece is called message and the procedure given by the block code encodes each message individually in to a codeword, also called a block in the context of block codes. The sender then transmits all blocks to the receiver, who can in turn use some decoding mechanism to (hopefully) recover the original messages from the possibly corrupted received blocks. The performance and success of the overall transmission depends on the parameters of the channel and the block code.

Formally, block code is an injective mapping

$C: \sum k \rightarrow \sum n.$

Here, Σ is a finite and non empty set and k and n are integers. The meaning and significance of these three parameters and other parameters related to the code are described below.

The alphabet Σ:

The data stream that has to be encoded is modeled as a string over some **alphabet** Σ. The size |Σ | of the alphabet is often written as q. If q=2, then the block code is called a binary block code.

The message length k:

Messages are elements m of Σ^k, that is, strings of length k. Hence the number k is called the **Message length** or **dimension** of a block code. The block length n

The **block length** nofa block code is the number of symbols in a block. Hence, the elements c of Σ^n are strings of length n and corresponds to blocks that may be received by the receiver. Hence they are also called received words. If c =C(m) for some message m, then c is called the codeword of m.

The rate R:

The **rate** of a block code is defined as the ratio between its message length and its block length: R=k/n.

A larger ate means that the amount of actual message per transmitted block is high. In this sense, the rate measures the transmission speed and the quantity 1 –R measures the overhead that occurs due to the encoding with the block code. It is a simple information theoretical fact that the rate cannot exceed 1 since data cannot be compressed in general. Formally, this follows from the fact that the code C is an injective map.

The distanced:

Now, we define the distance or minimum (Hamming) distance ofa linear block code as the minimum of Hamming distances between two different code words:

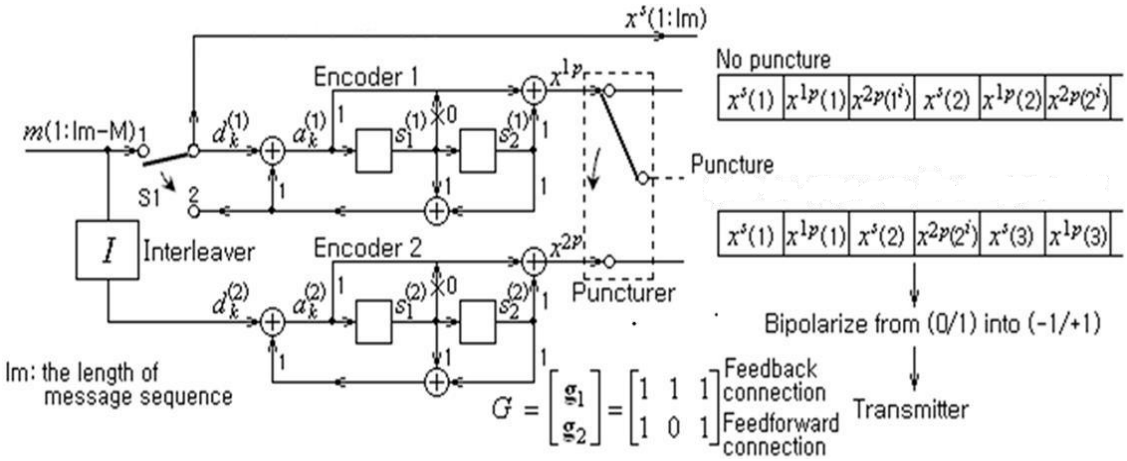
$d_{min}(c) = Min_{i\neq j}\, d_H(c_i, c_j)$

Here, Hamming distance between two different code words c_i and c_j is the number of bits having different values and can be found as the weight of the modulo-2 sum/difference of the two code words:

B. LLR Algebra

First let us introduce the basis of LLR algebra.

AprioriL-value,which is a softvalue measuring how high the probability of a binary random variable being +1 is in comparison with that of u being-1:



$$L_u(u) = \ln \frac{P_u(u=+1)}{P_u(u=-1)}$$

This is a priori information known before here sulty caused by u becomes available.While the sign of LLR

$$\hat{u} = \text{sign}\{L_{\mathbf{u}}(u)\} = \begin{cases} +1 & P_{\mathbf{u}}(u=+1) > P_{\mathbf{u}}(u=-1) \\ -1 & P_{\mathbf{u}}(u=+1) < P_{\mathbf{u}}(u=-1) \end{cases}$$

III. BLIND IDENTIFICATION FOR TURBO CODES

In this paper, the received encoded bit stream is divided in to three branches: the information bits X, the parity check bits of RSC1 Y and the paritycheck bits of RSC2 Z.Ourblind identification approach for Turbo codes hastwostages. First, we identify RSC1 by LLR for syndromeprior probability algorithm. RSC2 is then identified by decoding based on zero insertion and LLR accumulation.

A. Identification fo rRSC1 Based on LLR Accumulation

The component code RSC1 is recursive systematic convolutional code.

ForRSC1,given an encoderθ_1 and its parity-check matrix $H_\theta F$,we can obtain:

Where c_v stands for the code-word. Note that the elements of $H_\theta F$ are"0"or"1",thus we can rewrite 5 as

$c^\theta \oplus c^\theta \oplus c^\theta \oplus \ldots \oplus c^\theta = 0$

v,1

v,2

v,3

v,k

If and only if 0=0′.

1

For RSC1,its input bitstreamX andits parity check bitstream Y are already known.Because of its systematic property, the encoded bits of RSC1 can be built up with X and Y.

LetX=($x_1, x_2, \ldots, x_N$)denote th einputbits

Y=($y_1, y_2, \ldots, y_N$)denote the parity checkbits.Then the encoded bits of RSC1can be shown as

$out1=(x_1, y_1, x_2, y_2, \ldots, x_N, y_N)$

Accordingto[11][12], for the encoded bit stream,if the parity check relations satisfy, the syndromeposterior Probability illustrated as follows would most probably take high values.

$$\hat{y} \triangleq \boxplus_{j=1}^{p} LLR(c_{v_j})$$

Thus we can identify the encoder ofRSC1.The generator matrix of RSC1 is chosen from candidatesetΘ_1.

We Calculate$_{()} \triangleq \sum^{k} |y|$ for each$\theta^{(i)} \epsilon \Theta$. if$\theta^{(i)}$ is taken as the encoder,its γθ(K)should be the maximal

In all the accumulaions.i=1

B. Identification for RSC2 and Recovery of Interleaver

The key idea of LLR accumulation is the parity check relations in the code-words. However, for RSC2, the information bit stream X is permuted bythe interleaver Π resulting the absence of the information bits for the code-words. Thus the LLR algorithm is no longer applicable. Figure.2 illustrates the structure of the interleaver Π and RSC2

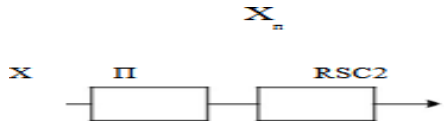


Figure2:Interleaver and RSC2

The informationbits denotedbyX=(x_1,x_2 ,’’’,xn)arepermuted bythe interleaver withoutput denoted byXΠ($x\pi 1$, $x\pi 2$, , $x\pi N$). ThenX Πare encoded by RSC2 as information bits to get the parity check bits.

Z=(z_1,z_2,’ ‘,zN)

Toidentifythe encoder, we need to recover the missinginformationbits. Note that the intermediate variable XΠ is not only the output of the interleaver, but also the input for the encoder.If we could get XΠ,we can easily recover the interleaver and then identify the encoder.Therefore we firstly get the intermediate variableXΠ.

The encoder of RSC2 is chosen from a candidate set denoted byΘ_2 and the length of the interleaver is N. The identification of RSC2 has three steps.
Step1 Decoding based on zero insertion.

Step2 Recovery interleaver based on Bit-Exclusive-Or.

Step3RSC2 identification based on LLR algorithm.

Step1:Decoding Based on Zero Insertion

As mentioned before, in the context of RSC2, the information bits are out-of-order as a result of the interleaver, which makes it unable to generate the complete code-words sequence directly. Thus we need to recover the input information bits of RSC2 XΠ to identify the encoder by the
LLR algorithm. As the output of the interleaver Π , XΠ is a permutation of X. The only prior knowledge for the interleaver is its length N. Until now, there is no efficient way to recover XΠ under this condition. Therefore we have to recover XΠ through its another status which is the input Information

bits for RSC2. Fortunately, as for the encoder, the correspondence between the input information bits and the encoded output bits are one-to-one. Thus we can recover X Πbased on decoding the code-words of RSC2.However,the output of RSC2 is incomplete with its information bits missing. Thus the parity check bit stream Z is preprocessed by zero insertion to complete the code-words then decoded with BCJR algorithm to obtain X_{11}.

The parity check bit stream Z is preprocessed by zero insertion, which means we insert zeros instead of the information bits. Let Z[z1,z2, …,zN] denote the parity check bits. We insert 0 to the corresponding information positions. For RSC2, the code-word sequence should be shown as

out2 [0, z_1,0,z_2,,0, z_N].(9)

Obviously the encoded bit stream in (9) is of many different bits compared with the actual code-words. Thus the efficient decoding algorithm is required. Here we use BCJR algorithm [16]. It is a soft-input soft -output (SISO) algorithm based on posteriori probability of the information bits. It aims to find the maximal posterior probability of the information bits, and performs well at high BER as it takes use of the soft information of the demodulated data rather than the hard decision.

BCJR algorithm needs two inputs: the complete encoded sequence arranged at intervals of information bits and parity check bits which we already obtained in (9), and the priori knowledge of the information bits. On the condition of BPSK modulation, 0/1 are mappingonto -1/+1. Ifthe information bits are zeros, the priori probability will be set as 0.5 for BCJR. Thus the preliminary XΠ can be decoded based on (9) with BCJR algorithm.

Step2:Recovery interleaver based on Bit-Elusive-Or

In step 1, the output of the interleaver $X_\pi$is obtained. Thus for the interleaver π,its input sequence X, output sequence$X_\pi$and its length n are already known. Thus the interleavercan be recovered based on bit-exclusive- oralgoritham.This algorithm can be illustrated as follows:

Let matrix A and matrix B with N columns respectively denote the input and the output of theinterleaver, where

$$A = [x_1 \;\; x_{12} \cdots x_{1N}$$
$$x_{21} \; x_{22} \cdots x_{2N}$$
$$x_{m1} \; x_{m2} \cdots x_{mN}]$$
$$B = [x_{\pi 11} x_{\pi 12} \cdots x_{\pi 1N}$$
$$x_{\pi 21} x_{\pi 22} \cdots x_{\pi 2N}$$
$$x_{\pi m1} x_{\pi m2} \cdots x_{\pi mN}]$$

Take the column vector $(1 \le i \le N)$from A,then calculate the bit-exclusive-or with each vector$x_{\pi j}(1 \le j \le N)$ofB.Theresultsaresavedintoan N*NmatrixC,whose elements$c_{i,j}$ represent the different numbers of the i-th column of A and the j-th column of B.

The index of the minimum value ofthe i-throwof Cstandsfor the mapping position for i.Then the interleaverΠ can be reconstructed.

Step3: RSC2 Identification Based on LLR Algorithm

After step 2, the interleaver Π is recovered. Since its input information bit stream X is already known, its outputbit streamXΠ canbe updated,whichis also the information bits of RSC2.Thus the code-words of RSC2canbefilledupbyXΠand Z. Compared with the code-words in step 1 obtained from zero insertion, the updated code-words are much more accurate. Based on this code-words, the RSC2 can be identified by the LLR accumulation.

IV. SIMULATION RESULTS

randomly generated and its length N is 1784. Rate of the Turbo codes is 1/3. The coded bit stream is modulated by BPSK, then transmitted over an additive white Gaussian noise (AWGN) channel. As illustrated in Section II, the received code- words stream is split into three parts including the information bit stream X, parity check bit stream of RSC1 Y and parity check bit stream of RSC2 Z.

A. Identification Result of RSC1

TheencodedbitstreamofRSC1is madeup withinformationbitstreamXandparitybitstreamY.Thusthe LLRs for syndrome a posteriori probabilities are directly calculated for each θΘ by (8).

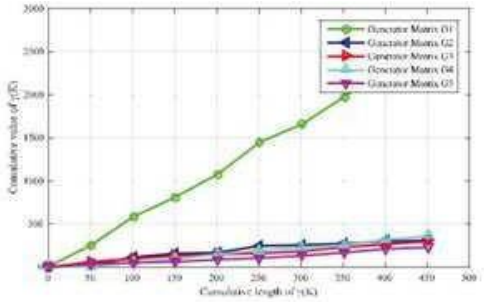


Figure3:Comparison of Cumulative Metrics for RSC1 With All the Five Generator Matrices for SNR = 3.5 Db

Note that the accumulation differences between the true generator matrix and the other matrices are smaller for SNR=3.5dB than the one in identification of RSC2 introduces a lot of uncertain bits which leads to the inaccuracy ofthe interleaver recovery. At last,theAccumulation of LLR is based on code-words with higherBER.As a result, the differences between the right matrix and the other candidates are less.

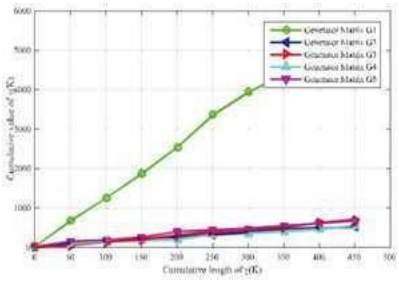


Figure4.Comparison of Cumulative Metrics for RSC1 with All the Five Generator Matrices for SNR = 5 Db

Figure. 3 and Figure. 4 demonstrate the accumulationsof LLRfor five matricesfor SNR=3.5dB and 5dB. We can easily find out that in both figures, the accumulation of generate matrix $G_1$is much higher than the other four generator matrices. It indicates that $G_1$is identified as the encoder of RSC1.

B. Identification Result of Rsc2

For identification of RSC2, the bit stream is firstly dealt with zero insertion and decoded with each $\theta \in \Theta$. Then based on X and all the $X\Pi$, the interleaver Π can be recovered. After this, all the $X\Pi$ can be updated. At last, the LLRs are accumulated to identify the encoder.

Figure. 5 and Figure. 6 show the accumulations of LLR of all the five generator matrices for SNR=3.5dB and SNR=5dB. In both figures, the accumulation curves of $\gamma\theta(K)$ for matrix G2 are larger than the other four generator matrices.It indicates that the encoder of RSC2 is G2

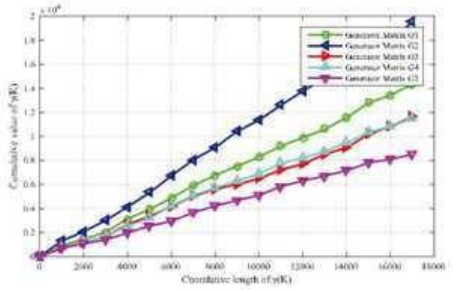


Figure5:Comparison of Cumulative Metrics for RSC2 With All the Five Generate Matrices for SNR = 3.5 Db

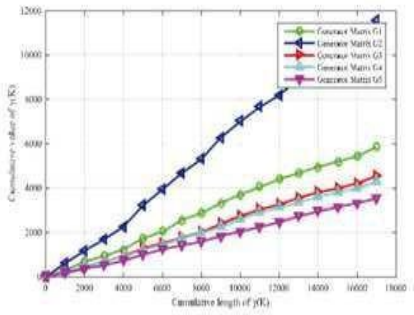


Figure6:Comparison of Cumulative Metrics for RSC2 with All the Five Generate Matrices for SNR = 5 Db

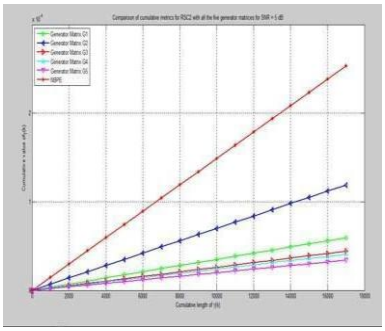
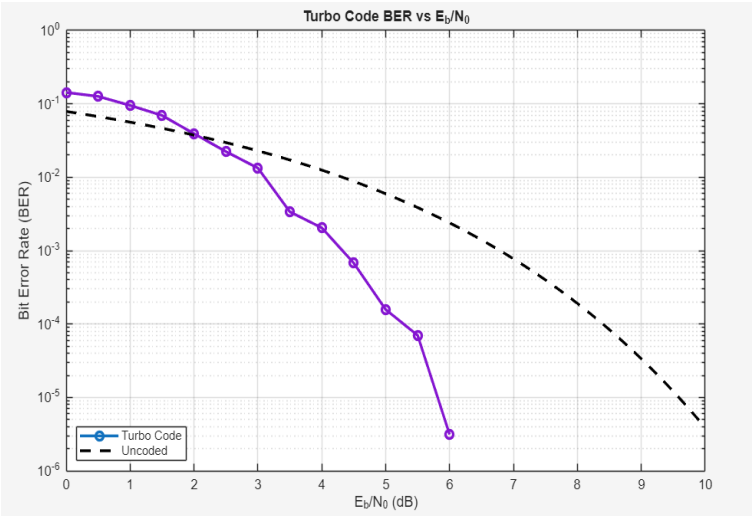


Figure7:Novel blind parameter-estimation

The improvement for identification of RSC2 and the application for Turbo codes with various rates. we design a novel blind encoder parameter estimator for turbo codes. We propose to separate the feedback components from the forward path in a recursive convolution encoder so as to blindly estimate the parameters.

V. CONCLUSION

The graph shows the performance of a Turbo Code system in terms of Bit Error Rate (BER) versus Eb/N0 (Signal-to-Noise Ratio per bit). The x-axis represents Eb/N0 in dB, indicating the quality of the communication channel, while they –axis (on a logarithmic scale) represents the BER, which measures how many bits are received in correctly. The purple solid curve with circular markers corresponds to the Turbo coded system, and the black dashed curve represents the un coded system. This comparison helps in understanding how much improvement Turbo coding provides over a basic transmission without error correction. Thus, the graph clearly demonstrates that Turbo coding significantly enhances the reliability of data transmission by achieving very low Bit Error Rates at relatively lower Eb/N0 values. As the SNR increases, the Turbo coded curve shows a rapid reduction in errors, eventually dropping below 10^{-5} and approaching 10^{-6}, indicating highly accurate communication. In contrast, the uncoded case maintains a higher error rate even at the same SNR levels. This highlights the strong coding gain and efficiency of Turbo codes, making them highly suitable for modern communication systems operating in noisy environments Conclusion



The simulation results clearly show that both convolutional and Turbo codes achieve an approximate coding gain of 4 dB. In the case of convolutional coding with Viterbi decoding, this4 dB gain is observe data Bit Error Rate (BER) of around 10^{-5}.In contrast, for Turbo coding, the same4 dB gain is achieved at a much lower BER, close to 10^{-6}, indicating its superior error correction performance. This improvement indicates that for the same error performance, coded systems require less signal power compared to un coded transmission. Such coding gain is highly beneficial in practical applications like satellite and deep-space communication, where power efficiency and reliable data transmission are critical.

REFERENCES

1. R. G. Gallager, Information Theory and Reliable Communication. Hoboken, NJ,USA: Wiley, 1968.

2. P J. K. Wolf and A. J. Viterbi, ‘‘On the weight distribution of linear block codes formedfromconvolutional codes,’’IEEETrans.Commun.,vol.44,no.9,pp.1049–1051,Sep.1996.

3. D.E.Knuth,TheArtofComputerProgramming,vol.3,2nded.Reading,MA,USA: Addison-Wesley, 1998.

4. S.M. P.C.Fossorier,S.Lin,andD.J.Costello,‘‘Ontheweight distributionofterminated convolutional codes,’’ IEEE Trans. Inf. Theory, vol. 45, no. 5, pp. 1646–1648, Jul. 1999.

5. R.Garello,P.Pierleoni,andS.Benedetto,‘‘Computingthefreedistance ofturbocodesand serially concatenated codes with interleavers: Algorithms and applications,’’ IEEE J. Sel. Areas Commun., vol. 19, no. 5, pp. 800–812, May 2001

6. Y.Polyanskiy,H.V.Poor,andS.Verdu,‘‘Channel codingrateinthefiniteblocklength regime,’’ IEEE Trans. Inf. Theory, vol. 56, no. 5, pp. 2307–2359, May 2010

7. Universal Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD), document TS 125 212, V10.1.0, May 2011

8. C.Lou,B.Daneshrad,andR.D.Wesel,‘‘Convolutional-code-specificCRCcoded design,’’ IEEETrans.Commun.,vol.63,no.10,pp. 3459–3470, Oct. 2015..

9. H. Yang, L. Wang, V. Lau, and R. D. Wesel, ‘‘An efficient algorithm for designing optimal CRCs for tail-biting convolutional codes,’’ in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2020, pp. 292–297.

10. L. Wang, D. Song, F. Areces, T. Wiegart, and R. D. Wesel, ‘‘Probabilistic shaping for trellis-codedmodulationwithCRC-aidedlist decoding,’’IEEETrans.Commun.,vol.71,no. 3, pp. 1271–1283, Mar. 2023.