Turbo codes for telemedicine applications

K B Santhosh Kumar and B R Sujatha

Department of Electronics and Communication Engineering, Malnad College of Engineering, Hassan-573201, Karnataka, India.

E-mail: 1 kbs@mcehassan.ac.in, 2 brs@mcehassan.ac.in

Abstract. Telemedicine is an emerging technology in the area of remote medical services. It involves real-time communication between patients and healthcare providers through electronic equipment for diagnosing diseases. Biomedical signal processing is the process of measurement and analysis to deliver required information for doctors to make decisions. The identification of diseases is made upon collecting the biomedical signals from the patient’s body. Telemedicine concepts can be employed for this method. For remote diagnosis purposes, as the biomedical signal needs to transmitted through wireless media there is a need for efficiently coding them. In this paper, turbo coding is used and is transmitted through the AWGN channel. The performance of BER for different turbo decoding algorithms are compared to obtain error free transmission.

1. Introduction

Telemedicine has been a boon to patients in rural areas as they have access to the efficient medical expertise available in the urban India. For error free diagnosis, doctors need to get accurate physiological information of the patients. In this regard, EEG signals is one of the important physiological signals that defines the working of human brain. The electrical activity in the brain gives information about its complex behaviour along with its dynamic and non-linear properties. The EEG signal is generated by cortical nerve cell inhibitory and excitatory postsynaptic potentials [1]. The frequency range of EEG signal is 1 Hz to 100 Hz and 10 µV to 100 µV is the approximately amplitude range[2]. Parkinson’s disease (PD) is one of the progressive neurodegenerative disorder which can be detected using EEG signal, affecting about 1% of the world population over 60 years of age. For efficient transmission of these signals, detection and processing is very much necessary. Turbo coding is one of the error correction coding which is used in our approach. Turbo codes have made the first big leap towards reaching the Shannon limit, the highest transmission rate that can be achieved over a noisy channel without errors[3], [4].

In this paper, the turbo code is implemented in a Simulink platform. This paper is organized as follows, encoder and decoder of turbo code structure are explained in Section 2. In Section 3, the processing blocks required for the implementation of turbo codes, modulation schemes and various decoding algorithms used are described. Section 4 shows the results along with the discussion. Finally, Section 5 presents the conclusion.
2. Turbo Codes

2.1 Turbo Encoder
Two or more convolutional encoders are connected in parallel to design turbo encoders. An interleaver is placed before the second encoder to make sure that the input is statistically independent before applying it to the second encoder [5]. Both the convolutional encoders receive the same input bitstream as input with one of them interleaved using a random interleaver. The encoders scramble the input data before sending it through the physical channel. The systematic and recursive bits are extracted from the encoded data by making use of puncturing technique. Both multiplexing and puncturing techniques are employed prior to transmission of data over the channel. Figure 1 shows the schematic structure of a turbo encoder. The BER performance of the turbo code is limited at higher values of signal to noise ratio due to relatively minimum distance of the code. Hence, low weight codeword multiplicity can be reduced from the design of turbo codes [6].

The first transmitted sequence by the first encoder is,
\[ y(1) = (y_1^{(1)}, y_2^{(1)}, \ldots, y_{k-1}^{(1)}) \]  
(1)

The second encoder generates parity sequence given by,
\[ y(2) = (y_1^{(2)}, y_2^{(2)}, \ldots, y_{k-1}^{(2)}) \]  
(2)

The final transmitted code word is given by,
\[ y = (y_1^{(1)}, y_1^{(2)}, y_2^{(1)}, y_2^{(2)}, \ldots, y_{k-1}^{(1)}, y_{k-1}^{(2)}) \]  
(3)

![Figure 1. Structure of Turbo Encoder.](image)

2.2 Turbo Decoder
Figure 2 shows the turbo decoder structure. Decoder 1 and 2 are connected iteratively through interleavers and de-interleavers [7]. This connection allows the advantage of probability of the received bits of the second decoder to be taken by first decoder. This process is repeated until the BER becomes minimum. A hard decision is made on the soft output of decoder 2 at the end of the decoding process [8]. The systematic bitstream from the channel, applied as input to the decoder 1 is given by,
\[ y^{(s)} = \{y_1^{(1)}, y_2^{(1)}, \ldots, y_{k-1}^{(1)}\} \]  
(4)

Decoder 1 also receives parity bits given by,
\[ y^{(p)} = \{y_1^{(2)}, y_2^{(2)}, \ldots, y_{k-1}^{(2)}\} \]  
(5)

![Figure 2. Turbo decoder.](image)
3. Design of Turbo Codes in Simulink

Simulation of turbo code generation is carried out in MATLAB-R2018a using Simulink Version 9.1 (R2018a) [8]. The turbo coding model is made up of MATLAB blocks, user-defined MATLAB system blocks as well as user-defined functions, all of which are necessary for its functioning as shown in Figure 3. The primary toolboxes used are the communication system and discrete system processing toolbox. The model has the following specifications:

- Code rate: 1/4 and 1/3.
- Encoder polynomials: 13, 15.
- Interleavers type: Random interleavers.
- Modulation: 16 QAM.
- Data : EEG signal from database of a Parkinson’s patient.

![Figure 3. Turbo coder system developed in Simulink.](image)

The various subcomponents of designed system are as follows,

3.1 EEG signal

In this work, the input EEG signal is taken from the EEG database of a patient suffering from Parkinson’s disease [9]. The extracted EEG signal has power frequency noise that is filtered using the FIR Notch filter with a centre frequency $f_0=50\text{Hz}$.

3.2 Turbo Encoder

Matlab environment is used to generate input from the EEG database. The input is loaded into the Simulink platform from the workspace of the MATLAB with the definite frame length. The input signal is applied to encoder 1 as well as to the interleaver and interleaver output is applied as input to encoder 2 so that data is statistically independent[10]. Encoder 1 is followed by two puncture blocks. The output of both the encoders are multiplexed, concatenated and is transmitted over the Additive White Gaussian Noise (AWGN) channel, as shown in Figure 4.

![Figure 4. Turbo encoder system developed in Simulink.](image)
3.3 Quadrature Amplitude Modulation
Both amplitude and phase are modulated to determine the modulation state in the Quadrature Amplitude Modulation (QAM). 16 QAM uses four bps and to encode 12 bits, it is broken down into four bits per group making a total of three groups, thereby using any 3 symbols [11], [12].

3.4 Concatenation
This is a user-defined block that concatenates the systematic bits and parity bits from the encoders. Its parameters are obtained from the function which retrieves the parameters of the trellis from the polytrellis object in the encoder and makes them available to all the blocks within the turbo encoder [13]. The block takes the two convolutional codewords, removes the interleaved systematic bits, and concatenates its parity bits with the codeword of the upper encoder. The two sets of termination bits are then added at the end of the new (2, 1) codeword.

3.5 Channel
An AWGN channel is used as the communication channel. Its SNR (E_b/N_0) values are set by the function ‘berSimulation’, while the symbol period is updated automatically by the model itself each time the simulation is run.

3.6 Turbo decoder
The communications system toolbox block accepts the channel information i.e., systematic bits and parity bits and previous information, and outputs an updated version of the apriori information which, in this model is the extrinsic information. It uses the decoding algorithms True APP, Log MAP, and Log Max algorithm[14]. In the turbo decoder, an APP Decoder Simulink block processes both the inputs employing poly2trellis function to generate trellis using the constraint length and code generator (octal) as shown in Figure 5. Same trellis structure is used in both encoder and decoder where, trellis = poly2trellis(5, [37 21],37) where, the value 5 indicates encoder constraint length, the values [37 21] is the generator matrix polynomial and the value 37 is the feedback connection polynomial. APP decoder block has two inputs, L(u) representing the sequence of log-likelihoods of encoder input bits and L(c) representing the sequence of loglikelihoods of code bits. L(c) vectors have length of Q·n, where Q is a positive integer, if convolution code uses 2·n possible symbols per alphabet. Similarly, L(u) vectors have length of Q·k, if the decoder data uses 2·k possible output symbols per alphabet. The number of frames processed by block at each step is represented by the integer Q. The L(u) input to the APP encoder is the output of the Random de-interleaver block delayed by n samples and multiplied by a train of periodic pulses [15].

The hard decision processes the output of second random de-interleaver. The hard decision block produces the output by applying mask based on the likelihood of binary transformation. The first decoder receives the systematic channel observation, given by,
The observations of the first encoder’s parity bits is also received by first decoder, given by

\[ y^{(p)} = \{ y_1^{(2)}, y_2^{(2)}, \ldots, y_{k-1}^{(2)} \} \]  

(7)

3.6.1 True MAP Algorithm
One of the most advantageous decoding algorithms is the MAP algorithm. It is very difficult for the computation process of soft input and output algorithms. The Log-MAP and the Max-Log-MAP are the two simplified versions of the True MAP algorithm. LLRs are computed in the MAP algorithm. It stores the information in the format of the bit \( L(d_k) \) as presented [16].

\[
L(d_k) = \ln \left[ \sum \sum \gamma_{S_k} \alpha_{S_k-1} \beta_{S_k} \right] 
\]

(8)

Where,
- \( S_k \) = trellis state at trellis time \( k \)
- \( \alpha \) = forward state metric
- \( \beta \) = backward state metric
- \( \gamma \) = branch metric

3.6.2 Log-MAP algorithm
The algorithms complexity is reduced by approximating the exponential and the logarithmic operations using a relation represented by,

\[
\max^*(x,y) = \ln(e^x + e^y) = \max (x,y) + \log (1+e^{y-x})
\]

(9)

Where \( f_c = \log (1+e^{y-x}) \) is the correction function.

3.6.3 Max Log-MAP algorithm
Max-Log-MAP algorithm is used, as a MAP algorithm is complex and cannot perform in real-time [12]. The correction function can be implemented in different ways and hence is neglected by the Max-log-MAP algorithm. The approximated \( \max^*(.) \) operator is given by,

\[
\ln (e^x + e^y) = \max (x,y)
\]

(10)

Thus, reducing the complexity.

3.7 Performance testing without coder
Figure 6 shows the Error Rate Correction block which is a communications system toolbox block that compares the transmitted and received signal and calculates the bit error rate (BER) without turbo coding. The model in the figure 6 is used to transmit the signal without turbo coding though AWGN channel. The obtained output is uncoded data and the BER performance is plotted for the same.

4. Results and Discussions
In this section, the result developed in Simulink version 2018a obtained from the turbo codes is presented. For the turbo codes the parameters of interest are the interleaver length, number of iterations, and decoding algorithm as listed in table 1. The performance of the BER is investigated
with code rates 1/4 and 1/3 and for different decoding algorithms. The BER performance for the rates $R=\frac{1}{3}$ and $R=\frac{1}{4}$, with block size 6144 with turbo encoder is shown in Figures 7 and 8. As shown in simulation results, the BER performance is improved for the $R=\frac{1}{3}$ compared to the encoder rate $R=\frac{1}{4}$ as shown in figure 9. The value of BER is almost constant up to SNR of 2 dB and it decreases for higher values of SNR for rate $R=\frac{1}{3}$ and $\frac{1}{4}$ turbo code. We observed more than 5 dB variation for the code rate $R=\frac{1}{3}$ over $R=\frac{1}{4}$.

| Parameters          | Type/ Value               |
|---------------------|---------------------------|
| Generator matrix    | [13 11]                   |
| Code Rate           | 1/4 and 1/3               |
| Decoder Algorithm   | True APP, Log MAP, MAX Log MAP |
| Frame size, N       | 6144 bits                 |
| Number of iterations| 4                         |

Table 1. Turbo Decoder Parameters

![Figure 7. BER performance of True APP turbo coder for Rate $\frac{1}{4}$](image)

![Figure 8. BER performance of True APP turbo coder for Rate $\frac{1}{3}$](image)
5. Conclusion

In this paper, we have established, investigated, and described the turbo codes in detail and demonstrated its performance in the AWGN channel using the SIMULINK platform. The turbo codes performance is measured by considering several parameters including different decoding algorithms. The BER of turbo codes with rate 1/3 performs better compare to rate 1/4 at E_b/N_0 of 5dB. The performance approaches to near Shannon limit capacity on maximum attainable data transfer rate over an AWGN channel. We investigated that, the True APP and Log MAP decoding algorithms are the best-suited algorithm with a lower bit error rate compared to Max-Log MAP algorithm. As further work, we have planned to carry out performance comparison for different types of interleaver, changed coding rates and modulation methods.

Acknowledgement

Waveform EMG database for this study was accessed from Physionet, managed by members from the Laboratory of Computational Physiology and Margret and H.A. Rey Institute for Nonlinear Dynamics, Effect of Deep Brain Stimulation on Parkinsonian Tremor, Published on 27th August 2001, Version:1.0.0.
References

[1] Rangayyan 2002, Biomedical Signal Analysis, Wiley, New York, 2nd edition, 2002.
[2] D.C. Reddy 2005, Biomedical Signal Processing: Principles and techniques, Tata McGraw Hill, New Delhi, 2005.
[3] Shu Lin and Daniel J. Costello 2004, Error control coding, Pearson, Prentice Hall, 2nd edition, 2004.
[4] Branka Vucetic, Yonghui Li and Lance C. Perez, Fan Jiang 2007, Recent Advances in Turbo Code Design and Theory, Proceedings of the IEEE, Volume: 95, Issue: 6, June 2007, pp1323 - 1344.
[5] Ken Gracie and Marie Hamon 2007, Turbo and Turbo-Like codes: principles and applications in telecommunications, Proceedings of the IEEE, Volume: 95, Issue: 6, June 2007, pp1228 - 1254.
[6] Axel Huebner; Kamil Sh. Zigangirov; Daniel J. Costello 2008, Laminated turbo codes: A new class of block-convolutional codes, IEEE Transactions on Information Theory, Volume: 54, Issue: 7, July 2008, pp3024 - 3034.
[7] Sumit Kumar, Hemant Dalal 2014, Performance Comparison of Turbo Codes and Modified Turbo Codes with Different Rate, International Journal of Science, Engineering and Technology Research (ISETRT), Volume 3, Issue 5, May 2014, pp 1520-1523.
[8] Raad, I. and Yakan 2005, M., Implementation of a turbo codes test bed in the Simulink environment. The Eighth International Symposium on Signal Processing and Its Applications, 2005 Piscataway IEEE, 2005 IEEE, pp 847-850.
[9] D. PuthankattilSubha, Paul K. Joseph, Rajendra Acharya U and Choo Min Lim 2008, EEG Signal Analysis: A Survey, Journal of Medical Systems Published online: 6 December 2008, @Springer Science + Business Media, LLC 2008, pp195-212.
[10] Md Abdul Halim, Satya Prasad Majumder 2018, Performance Analysis of a SIMO Wireless Link with Turbo Coding, 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 13-15 Sept. 2018, pp 636-641.
[11] Khaizuran Abdullah, Ahmad Fadzil Ismail, Wahidah Hashim and Zahir M. Hussain 2010, An optimal circular 16-QAM modulation technique for wavelet and follower based OFDM, International Conference on Computer and Communication Engineering (ICCCE’10), 11-12 May 2010.
[12] Purushottama G B and B R Sujatha 2015, Turbo Codes with Golden Section Interleaver, International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 03-June-2015, pp1069-1073.
[13] Ghassan M. Kraidy 2016, On Progressive Edge-Growth Interleavers for Turbo Codes, IEEE Communications Letters, VOL. 20, NO. 2, February 2016.
[14] MohanadAbdulhamid and Mbogua Thairu 2019, Performance Analysis of Turbo Codes Over AWGN Channel, Scientific Bulletin of the Electrical Engineering Faculty, ISSN 2286-2455, DOI: 10.1515/sbeef-2019-0009, April 2019, pp43-48.
[15] Bhupendra Singh and Ishtendra Pratap Singh 2015, Performance Enhancement of LOG MAPturbo decoder for mobile applications, 2015 International Conference on Recent Developments in Control, Automation and Power Engineering (RDAPE), Noida, India, 12-13 March 2015, pp 259-264.
[16] Santhosh Kumar K B, B R Sujatha 2020, Reliable Transmission of EEG Signal In Tele-Medicine. International Journal of Advanced Science and Technology, Vol. 29 No. 10s (2020), 1st June 2020, pp8687-8694.