ISSN: 2455 - 9172

EIJO Journal of Engineering, Technology And Innovative Research (EIJO–JETIR)

Einstein International Journal Organization (EIJO)

Available Online at: www.eijo.in

Volume – 1, Issue – 2, May - June 2016, Page No. : 36 - 42

Review of Turbo Decoder Using Log-MAP Based Iterative Decoding Algorithm

Indu[1], Vikram Singh[2]

[1] M.Tech. Scholar, [2] Assistant Professor

Jayoti Vidyapeeth Women's University, Jaipur, India

E-Mail:[1]indu7778@gmail.com, [2]vchoudhary030@gmil.com

Abstract

In digital communication the error correction capability is better as compared to analog communication. When in digital transmission system bits get corrupted, then error occurs. For improving the error correction capability, turbo codes are used. Turbo codes are the parallel concatenation of two convolutional codes. For that purpose an interleaver is used which separates the code. In digital communication system, the use of turbo codes enhances the data transmission efficiency. When turbo codes are used then Bit Error Rate (BER) performance reaches the Shannon's channel capacity limit. The most popular iterative decoding algorithm requires an exponential increase in hardware complexity to achieve higher decoding accuracy. In this paper Log-MAP Iterative decoding algorithm is used for describing turbo decoder.

Keywords: Convolutional Codes, Iterative Decoding, Turbo Codes, SISO Decoder.

1. Introduction

To achieve possible data with a few errors turbo decoders are used. The technique used for the error control during data transmission is known as Forward Error Correction (FEC) technique. Turbo codes are used in 3-Generation / 4-Generation mobile applications. The use of turbo codes provides higher data rates for low order modulation schemes such as BPSK or QPSK [1]. By using parallel concatenated encoder structure and soft input soft output (SISO) iterative decoders [1] [2] turbo codes are designed. For generating soft output the iterative decoder uses the Most a Posteriori (MAP) decoding algorithm. The Most a Priori (MAP) algorithm is also known as BCJR. The MAP decoding algorithm is not valid for real time systems and it is computationally complex. MAP algorithm is used for calculating A Posteriori Probability (APP) for each message bit [3]. The MAP algorithm is sensitive to SNR mismatch [4]. In the MAP algorithm the computation of the probability is obtained using non–linear functions. Practically decoding is possible through logarithmic versions of MAP algorithm [5] [6] [7] and Soft Output Viterbi Algorithm (SOVA) [8] [9]. The logarithmic versions of MAP algorithm and SOVA are less sensitive to SNR mismatch. The computational versions of MAP algorithm has best bit error rate (BER). The Log-MAP algorithm has highest computational complexity and thus Turbo decoder has low bit error.

2. Concatenated Coding

A single error correction code is not enough to remove error with reasonable complexity. The solution of this problem is "**concatenated coding**". Concatenated coding provides much more powerful code.

There are two types of concatenated codes

- Serial Concatenation Codes
- Parallel Concatenation Codes

Serial Concatenation Codes

Serial concatenation codes are introduced by Forney in 1966. Fig. (1) Shows the block diagram for Serial Concatenation Coding.

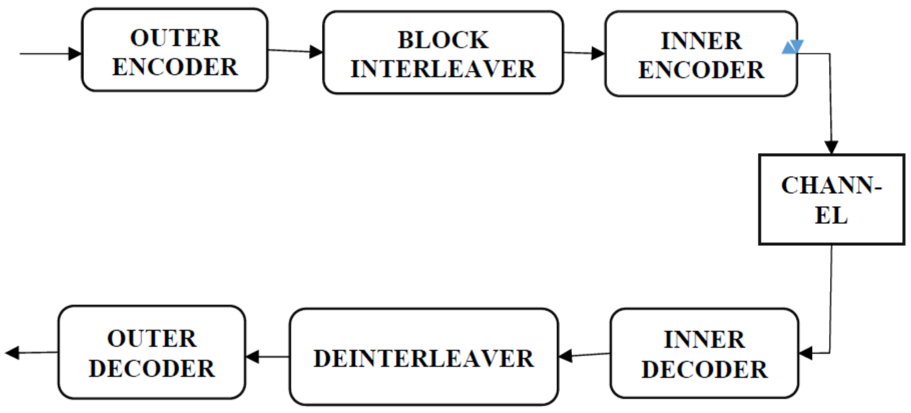

Figure 1: Block Diagram for Serial Concatenation Coding.

Parallel Concatenation Codes

Parallel concatenated coding is better as compared to serial concatenated coding. The parallel concatenation of two recursive systematic convolutional (RSC) codes generates the original turbo code.

3. Turbo Encoder

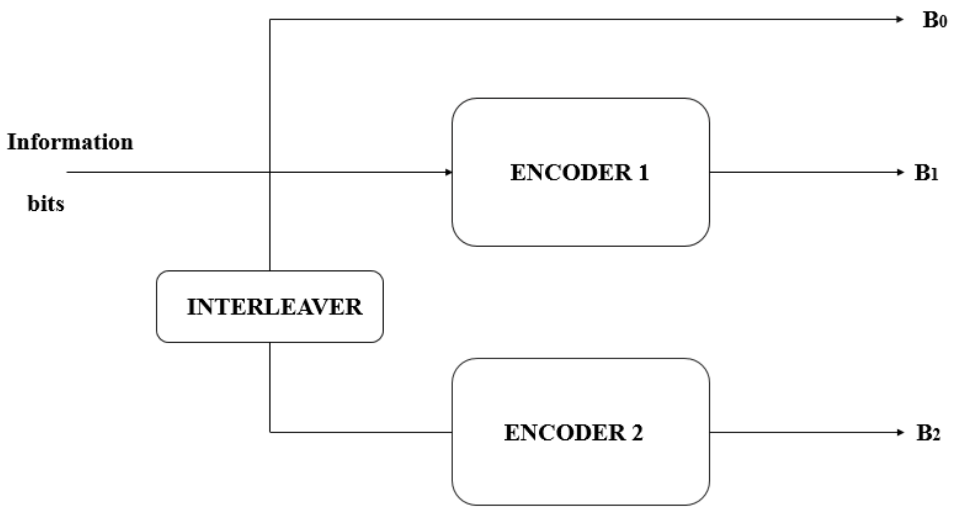

Figure 2: Block Diagram for Turbo Encoder [10].

Fig. (2) Shows the block diagram of a turbo encoder. A turbo encoder can be designed using two recursive systematic convolutional (RSC) codes; this is also known as parallel concatenation. An interleaver is used for separating the encoders. An Interleaver is a device which is used for permuting the data sequence in some predetermined manner. One of the two encoder's output is used to form a code word since one of the output is permuted form of the systematic output.

The first encoder generates the systematic output B0 and recursive convolutional B1 sequences while second encoder discards its systematic sequence and generates only one output i.e. recursive convolutional B2 sequence.

4. Decoding Algorithms

There are some decoding algorithms; names are Maximum a Posteriori (MAP) algorithm, Log-Map algorithm, Max-Log-Map algorithm and Soft Output Viterbi Algorithm (SOVA).

The MAP algorithm is known as a Maximum Likelihood (ML) algorithm. The MAP is used to find most probable information bit that was transmitted. MAP algorithm minimizes the bit or symbol error probability. In the MAP algorithm for returning information bits need not form a connected path through the trellis.

The Soft Output Viterbi Algorithm (SOVA) is asymptotically maximum likelihood algorithm at moderate and high SNR. SOVA is used to find the most probable information sequence to have been transmitted given the code sequence. The use of SOVA minimizes the word error probability. In this algorithm for returning information bits need a connected path.

Implementation in the MAP algorithm is not easy because of its computational complexity and it is sensitive to round-off errors that occur while representing with finite precision. Since the MAP algorithm does not performs practical decoding, therefore its logarithmic versions i.e. Log-Map and Max-Log-Map are used for practical decoding.

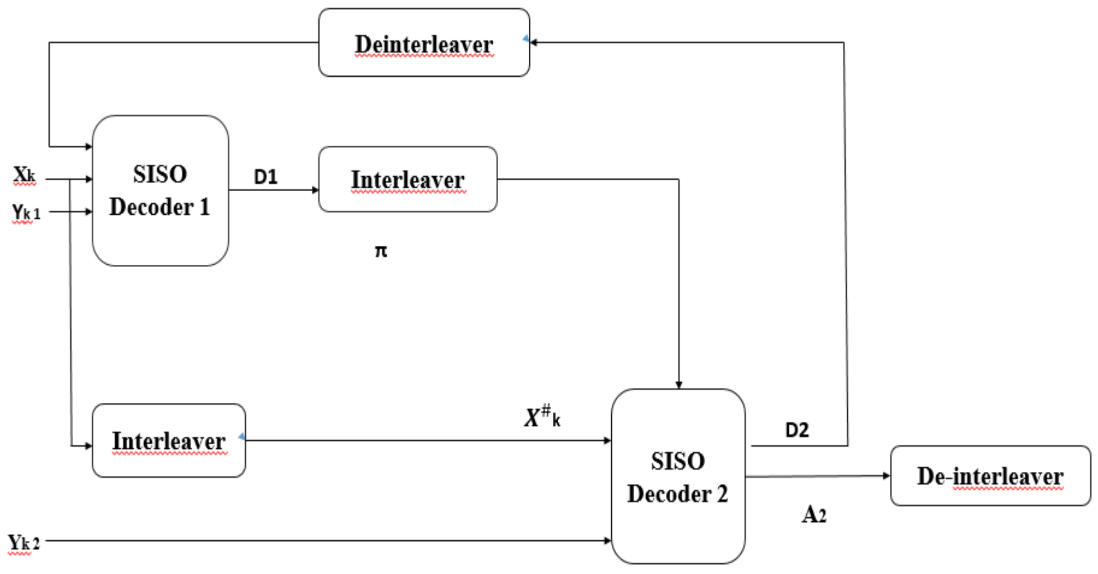

Figure 3: Block Diagram of an Iterative Turbo Decoder [10].

Fig. (3) Shows the block diagram of an iterative turbo decoder. In the turbo decoding process two SISO decoders are used. The first SISO decoder generates two outputs i.e. soft output and subsequently extrinsic information (EI). The extrinsic information [D1] is interleaved and it is used by the second SISO decoder as the estimate of the A priori probability (APP). The second SISO decoder also produces the extrinsic information [D2] and passes it after de-interleaving to the first SISO decoder to be used during the subsequent decoding operation.

5. Log-Map Algorithm

Multiplication in the log domain becomes addition, which reduces the hardware complexity. The Jacobian algorithm [11] is used instead of addition because the addition in log domain is not straight forward. The Jacobian algorithm approximates the addition of two integers according to the equation (1).

$$\ln\left(e^{x} + e^{y}\right) = \max(x,y) + \ln(1+\exp-|y-x|)$$

$$= \max(x, y) + f_c(|y-x|) \text{..........} (1).$$

The above equation shows that addition in the log domain reduces a maximization operation followed by a correction function $f_c()$ and the correction function becomes almost zero when the values of x and y are not same.

Therefore, the approximation of the above equation is

$$\ln\left(e^{x} + e^{y}\right) \approx \max(x,y) \text{ ....} (2)$$

In the log domain MAP algorithm work in two ways:-

A. When addition is performed like a maximization function alone as per equation (2) then it is called the Max-Log-Map algorithm.

B. When addition is performed like a maximization function, followed by a correction term as for equation (1), then it is called Log-Map algorithm.

Let $\overline{\alpha}$ be the log of α, then

$$\overline{\alpha} = \ln \alpha(s_i)$$

$$= \ln \sum_{s_i} \exp[\overline{\alpha}(s_{i-1}) + \overline{\gamma}(s_{i-1} \rightarrow s_i)]$$

$$= max_{s_{i-1}\epsilon A}{}^{*}[\overline{\alpha}(s_{i-1}) + \overline{\gamma}(s_{i-1} \rightarrow s_i) \text{ ....} (3)$$

Where A= the set of all states s_{i-1} that are connected to state s_i

Max*(x, y) = max(x, y) $\rightarrow$ for the Max-Log-Map algorithm

Max (x, y) + f_c(|y-x|) $\rightarrow$ for the Log-Map algorithm

Similarly, let $\overline{\beta}(s_i)$ represent the logarithmic of $\beta(s_i)$. It follows that,

$$\bar{\beta} = \ln \beta(s_i)$$

$$= \ln \textstyle\sum_{s_{i+1\epsilon B}} \exp[\,\bar{\beta}(s_{i+1}) + \bar{\gamma}(s_i \rightarrow s_{i+1})]$$

$$= max_{s_{i+1\epsilon B}}{}^{*} \, [\,\bar{\alpha}\,(s_{i+1}) + \bar{\gamma}\,(s_i \;\rightarrow\; s_{i+1}] \text{ .... } (4)$$

Where B= the set of all states s_{i+1} that are connected to state s_i

The $\bar{\alpha}\,(s_i)$ and $\bar{\beta}\,(s_i)$ have been calculated for all the states in the trellis, the log likelihood ratio can be computed as follows:

$$\wedge \; = \ln \textstyle\sum_{S_1} \exp\,[\,\bar{\alpha}\,(s_i) + \bar{\gamma}\,(s_{i} \rightarrow s_{i+1}) + \bar{\beta}\,(s_{i+1})]$$

$$- \ln \textstyle\sum_{S_0} \exp\,[\,\bar{\alpha}\,(s_i) + \bar{\gamma}\,(s_{i} \rightarrow s_{i+1}) + \bar{\beta}\,(s_{i+1})]$$

$$= max_{S_1}{}^{*} \, [\,\bar{\alpha}\,(s_i) + \bar{\gamma}\,(s_{i} \rightarrow s_{i+1}) + \bar{\beta}\,(s_{i+1})] \text{ ………. } (5)$$

Consequently the Log-Map algorithm proceeds as follows:-

(A) Forward Recursion for Log-MAP algorithm:-

(1) First of all create an array

$$\bar{\alpha}\,(\text{j,i}),\ 0 \leq \text{j} \leq 2^{M}\text{-1},\ 0 \leq i \leq L$$

Where L = the length of the data input

The result of forward recursion is stored in this array. Initialization of this array is:-

$$\bar{\alpha}(\text{j}, 0) = \begin{Bmatrix} 0, if\ j = 0 \\ -\infty, if\ j \neq 0 \end{Bmatrix} \text{ ………. } (6)$$

(2) Time index is i and it begins with time index when i=1

(3) State index is j and it begins with state index when j=0

(4) When $s_{i=s_j}$ and α is updated according to the following equation:-

$$\bar{\alpha}\,(\text{j, i}) = max_{s_{i-1}=s_{j'\epsilon A}}{}^{*} \, \{\bar{\alpha}\ (\text{j}', \text{i-1}) + \bar{\gamma}\,(s_{i-1} \rightarrow s_i\} \text{ ………. } (7)$$

Where A= the set of all states s_{i-1} that are connected to states s_i.

(5) Increment in j

(6) When j=2^M-1, all possible states have been considered and continue to step (7), otherwise return to step (4).

(7) Increment in i

(8) When i=L, then the end of all trellis has been reached: continue to step (2), otherwise return to step (3).

(B) Backward Recursion For Log-MAP algorithm:-

(1) First of all create an array

$$\overline{\beta}\text{ (j, i), } 0 \leq j \leq 2^M\text{-1, } 0 \leq i \leq L$$

The result of backward recursion is stored in this array. Initialization of this array is:-

$$\overline{\beta}\text{ (j, 0)} = \begin{Bmatrix} 0, if\ j = 0 \\ -\infty, if\ j \neq 0 \end{Bmatrix} \text{.......... (8)}$$

(1) For backward recursion time index is i=L-1

(2) For backward recursion state index is j=0

(3) When $s_{i=s_j}$ and β is updated according to the following equation:-

$$\overline{\beta}\text{ (j, i)} = max_{s_{i-1}=s_{j'\in B}} * \{\beta\text{ (j}', \text{i+1)} + \gamma(s_{i \to} s_{i+1}\} \text{ .... (9)}$$

Where B= the set of all states s_{i+1} that are connected to state s_i.

(4) Increment in j

(5) When j=2^M-1, all possible states have been considered and continue to step (7), otherwise return to step (4).

(6) Decrement in i

(7) When i=0, then the end of trellis has been reached: continue to step (C), otherwise return to step 3.

(C) Log Likelihood Ratio (LLR):-

For i = (0, 1, 2, ….., L-1), the LLR is

$$\wedge_i = max_{S_1} * [\overline{\alpha}\text{ (j, 1)} + \overline{\gamma}\ (s_{i \to} s_{i+1}) + \beta\text{ (j}' \to \text{i+1)]}$$

$$-\ max_{S_0} * [\overline{\alpha}\text{ (j, 1)} + \overline{\gamma}\ (s_{i \to} s_{i+1}) + \beta\text{ (j}' \to \text{i+1)].......... (10)}$$

Where $S_1 = \{ (s_i{=}S_j) \to (s_{i+1} = S_{j'}): m_i = 1\}$ is the set of all state transitions with a message bit of 1.

$S_0 = \{ (s_i{=}S_j) \to (s_{i+1} = S_{j'}): m_i = 0\}$ is the set of all state transitions with a message bit of 0.

The above set of all equations shows that all complex multiplications are reduced to simple addition operation. The parameters of Log-MAP algorithm is very close to approximations of the MAP parameters. Therefore, the BER performance of the Log-MAP algorithm is close to the MAP algorithm.

6. Conclusion

In digital communication system, the use of turbo codes enhances the data transmission efficiency. When turbo codes are used then Bit Error Rate (BER) performance reaches the Shannon’s channel capacity limit. In the Log-MAP algorithm, all complex multiplications are reduced to simple addition operation. Therefore, the BER performance of the Log-MAP algorithm is close to the MAP algorithm.

7. References

[1]. Jagdish D. Kene, Kishor D. Kulat, “Performance Evaluation of IEEE 802.16e Wi-Max Physical Layer”, IEEE Conference Proceedings' NUiCONE, 8-10 December 2011, PP. 1-4.

[2]. Jagdish D. Kene, Kishor D. Kulat, “Performance Optimization of Physical Layer Using Turbo Codes: A Case Study of Wi-Max Mobile Environment”, IEEE Conference Proceedings, ET2ECN, 19-21 December 2012.

[3]. L. Bahl, J. Cocke, F. Jelinek, and J. Rajiv. Optimal decoding of linear codes for maximum symbol error rate. IEEE Transactions on Information Theory, Volume 20(2):Pages 284–287, March 1974.

[4]. T.A. Summers and S.G. Wilson, "SNR Mismatch and Online Estimation in Turbo Decoding, "IEEE Trans. on Comm. Vol. 46, no. 4, pp. 421-424, April 1998.

[5]. P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and Sub-Optimal Maximum A Posteriori Algorithms Suitable for Tarbo Decoding, " European Trans. on Telecomm. vol. 8, no. 2, pp. 1 19-126, March-April 1997.

[6]. P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-optimal MAP Decoding Algorithrrts Operating in the Log Domain" International Conference on Communications, pp. 1009-1013, June 1995.

[7]. S. Benedetto, G. Montorsi, D. Divsalr, and F. Pollara "Soft-Output Decoding Algorithm in Iterative Decoding of Turbo Codes," TDA Progress Report 42-124, pp. 63-87, February 15, 1996.

[8]. J. Hagenauer, and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and Its applications, "Proc. Of GLOBECOM, pp. 1680-1686, November 1989.

[9]. J. Hagenauer, Source-Controlled Channel Decoding, "IEEE Transaction on Communications, vol. 43, No. 9, pp. 2449-2457, September 1995.

[10]. Module 6 “Channel Coding” Lesson 37 “Turbo Coding” version 2 ECE IIT, Kharagpur.

[11]. M Valenti. Iterative Detection and Decoding of Wireless Communications. PhD thesis, Virginia Polytechnic and State University, July 1999.