ELSEVIER
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Technology 6 (2012) 666 – 673
Procedia Technology

2nd International Conference on Communication, Computing & Security [ICCCS-2012]

Soft Output Decoding Algorithm for Turbo Codes Implementation in Mobile Wi-Max Environment.

Jagdish D. Kene[a], Kishor D. Kulat[b]

[a]Research Scholar, ECE Department, Visvesvaraya National Institute of Technology, Nagpur, India.
[b]Professor, ECE Department, Visvesvaraya National Institute of Technology,Nagpur, India.

Abstract

In the area of wireless communication system, researchers are concentrating on powerful forward error correction (FEC) coding techniques. A literature survey has shown that in the field of mobile Wi-Max environment Turbo code provides a coding gain close to Shannon limit. Since the performance of turbo codes depend on iterative decoding algorithms. Among various soft outputs decoding algorithms, Maximum A Posteriori (MAP) algorithm is used, which works on the principle of symbol estimation of the received signal. This paper presents logarithmic version of MAP decoding algorithm (Log-MAP). The author has taken efforts to modify the available Log MAP decoding algorithm in order to reduce the computational complexity. The results are presented for this new developed Max-Log-MAP decoding algorithm. Bit Error Rate (BER) performance and computational complexity of these algorithms are compared with reference to the standard defined by IEEE802.16e for mobile Wi-Max system.

Keywords- IEEE802.16e; Turbo codes; Coding gain; Iterative Decoding; Map Decoding.

1. Introduction

In the field of wireless communication system, error correction performance of turbo codes is close to Shannon limit [1], which is not practical for convolutional codes at low SNR [2]. Many researchers have shown their interest to find soft output decoding algorithms for implementation of turbo codes in real system applications. A very basic BCJR algorithm [3] is an algorithm for Maximum A Posteriori (MAP) decoding of error correcting codes. BCJR MAP decoding algorithm is computationally complex, sensitive to SNR mismatch and inaccurate estimation of noise variance.

* Jagdish D. kene. Tel.: +0-942-340-7300;
E-mail address: jagdish_kene@ece.vnit.ac.in, kdkulat@ece.vnit.ac.in.

doi:10.1016/j.protcy.2012.10.080

In the turbo decoding algorithm, MAP algorithm offers the best performance with iterative decoding, but cannot decode until the decoder receives the entire bit sequence [3]. Due to this BCJR algorithm leads to large decoding delay, power consumption and also required more memory size for iterations [5]. The iterative nature of turbo decoding algorithms leads to increase the complexity compare to conventional FEC decoding algorithms. Two basic iterative decoding algorithms, namely Soft Output Viterbi Algorithm (SOVA) and Maximum A Posteriori Probability (MAP) algorithm demands complex decoding operations over several iteration cycles [6]. Among these algorithms, SOVA has the least computational complexity but the worse BER performance while MAP algorithm provides significantly better BER performance with highest computational complexity. So for implementation of turbo code in real time system, the decoder complexity has to be reduced while preserving BER performance of the system [6].

In this paper we describe all version of MAP decoding algorithm and modify a new logarithmic version of MAP decoding algorithm. The new algorithm called Max-Log-MAP algorithm. Max-Log-MAP algorithm is less complex than Log-MAP algorithm but its BER performance is closed to Log-MAP algorithm [3]. These algorithms are less sensitive to SNR mismatch and provide accurate estimation of noise variance. The BER performance of these algorithms is tested with Binary Phase Shift Keying (BPSK) modulation scheme. The channel is considered as an Additive White Gaussian Noise (AWGN) channel [2].

The paper is organized as follows: Section 2 describes the structure of Turbo Encoder and Decoder. Section 3 briefly describes the MAP algorithm (modified version of BCJR) and Section 4 presents logarithmic version of MAP decoding algorithm. Section 5 presents the modified Max-Log-MAP decoding algorithm. Section 6 presents the BER performance of these algorithms are compared and verified for different parameters such as Number of Frame and Number of Iterations.

2. Turbo Codes

The turbo codes contain the structure that approaches the Shannon limit by using recursive encoders and Iterative network of soft output decoders [8]. The coder modifies the convolutional codes with short constraint length as block codes for large block length sequence and Iterative soft output decoder improves the estimation of the received message signal [9].

2.1. Turbo Encoder

The turbo encoders are designed from two or more convolutional encoders connected in concatenated parallel form with an Interleaver between them to ensure that the data the data received by the second encoder is statistically independent [10]. The block diagram of the encoder is shown in figure 1.

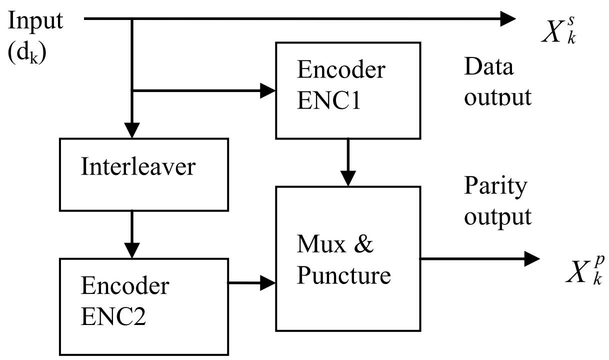

Figure 1: The structure of Turbo Encoder.

The binary input data sequence is represented by $d_k = [d_1, d_2, \text{--------}\ d_n]$. The input sequence is passes through the convolutional encoder ENC_1 and generates the coded bit stream X_{k1}. The data is then interleaved i.e. the data bit are loaded row-wise and read out column-wise. The bits are often readout in a pseudo random manner. The interleaved data sequence is passed to the second convolutional encoder ENC_2 and coded bit stream X_{k2} is generated. The coded data sequence (X_k) is multiplexed and punctured before it is to be sent across physical channel consisting of systematic code bits (X_k^s) and parity bits from the first encoder ($X_{k_1}^p$) and second encoder ($X_{k_2}^p$) [9]. The puncture unit is employed to extract the systematic bits and recursive bits from the received information. These will be used by the decoder to ensure the data is error free when it arrives at the end user terminal [11].Turbo codes can perform effectively at low signal to noise ratio (SNR) with small number of low weight code wards. This small minimum distance code limits the performance of turbo codes at higher SNR. Therefore turbo codes can employ to reduce the multiplicity of low weight code wards.

2.2. Turbo Decoder

Turbo decoder extracts the systematic bits and recursive bits from the received information. A block diagram of turbo decoder is shown in figure 2. The input to the turbo decoder is the received sequence represented as R_k = [Y_k^s, Y_k^p] It consists of two decoders DEC1and DEC2. DEC1 decodes sequence from ENC1 while DEC2 decodes sequence from ENC2. Each of the decoder acting as Maximum A Posteriori (MAP) decoder. The received sequence systematic value and received parity sequence are the input to the DEC1. The DEC1 generates the sequence are of soft estimate EXT1 is called extrinsic data which does not contains any information. The output sequence of DEC1 is interleaved and then passed to second decoder DEC2. The property of interleaver is same as in case of encoder [8]. DEC2 takes as its input systematic received bits Y_k^s and parity bits Y_k^p along with the interleaved form the extrinsic information EXT1provided by the first decoder DEC1.

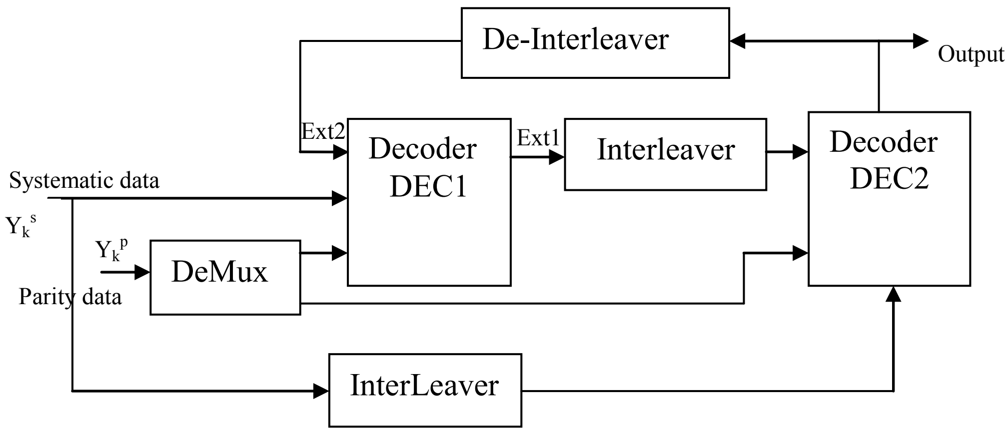

Figure 2: The structure of Turbo Decoder.

The DEC2 produces the output which when de-interleaved using inverse form of interleaver (load in column and read out in row). This consists of soft estimates EXT2 of the transmitted data sequence (d_k) is feedback to DEC1. This procedure is repeated in a iterative manner and continues until the bit error rate is zero (converges).

At the end of decoding process simple threshold operation is performed to carry out hard decision on the soft output of the second decoder DEC2 [9].

3. MAP Algorithm

In turbo decoding process, encoded information sequence (X_k) is transmitted over an AWGN channel and a noisy signal (Y_k) is received at the destination. In general, each decoder computes the Log Likelihood Ratio (LLR) to extract the information data bit from received signal (Y_k). The LLR is calculated for each bit (d_k) of data block length N is defined as [1]

$$L(d_k) = Log\left[\frac{\Pr\left(d_k = 1\right)|Y}{\Pr\left(d_k = 0\right)|Y}\right] \quad (1)$$

Where Pr ((d_k=1)|Y) is A Posteriori Probability (APP) of the information input data at time k (d_k). When it is equal to 1 then decoder gives entire received data [5]. The map decoding algorithm is a recursive technique. The Map algorithm calculates the LLR for each data bit as

$$L(d_k) = \ln\left[\frac{\sum_{S_k}\sum_{S_{k-1}} \gamma_1\left(S_{k-1}, S_k\right)\alpha\left(S_{k-1}\right)\beta\left(S_k\right)}{\sum_{S_k}\sum_{S_{k-1}} \gamma_0\left(S_{k-1}, S_k\right)\alpha\left(S_{k-1}\right)\beta\left(S_k\right)}\right] \quad (2)$$

Where α is the forward state metric, β is the backward state metric, γ is the branch metric and S_k is the encoder trellis state at trellis time k.
S_k is represented by considering v- tuple.

$$S_k = \left(a_k, a_{k-1}, a_{k-2}, \ldots\ldots\ldots\ldots a_{k-v+1}\right)$$

Where a_k is output of first shift register in the recursive encoder. Forward state metrics are calculated by forward recursion from trellis time k=1 to k=N. where N is the number of information bits in one data frame. The recursive calculation of forward state metric is performed as

$$\alpha_k\left(s_k\right) = \sum_{j=0}^{1}\alpha_{k-1}\left(s_{k-1}\right)\gamma_j\left(s_{k-1}, s_k\right) \quad (3)$$

Similarly, the backward state metrics are calculated by a backward recursion from trellis time k=N to k=1 as

$$\beta_k\left(s_k\right) = \sum_{j=0}^{1}\beta_{k+1}\left(s_{k+1}\right)\gamma_j\left(s_k, s_{k+1}\right) \quad (4)$$

The branch metrics are calculated for each possible trellis transition as

$$\gamma_i\left(S_{k-1}, S_k\right) = A_k \, P_r\left(S_k|S_{k-1}\right) \exp\left[\frac{2}{No}\left(Y_k^s X_k^s(i)\right) + Y_k^p X_k^p\left(i, S_{k-1}, S_k\right)\right] \quad (5)$$

Where i = (0, 1), A_k is a constant.
X_k^s, X_k^p are encoded systematic data bits and parity bits.
Y_k^s, Y_k^p are received noisy systematic data bits and parity bits respectively [6].

4. Log MAP Algorithm

In order to avoid the complex mathematical calculations of MAP decoding algorithm, the computation can be performed in the logarithmic domain [6]. The logarithm and exponential computations can be eliminated by the following approximation.

$$Max^*(x,y)=\ln\left(e^x+e^y\right)=Max(x,y)+\ln\left(1+e^{-|y-x|}\right) \tag{6}$$

The last term in above equation can easily be calculated by using a Look Up Table (LUT) therefore equation (2) and (5) becomes,

$$L(d_k)=\underset{(S_{k-1},S_k)/d_k=1}{Max^*}\left(\bar{\gamma}_1(S_{k-1},S_k)\right)+\bar{\alpha}_{k-1}(S_{k-1})+\bar{\beta}_k(S_k)-\underset{(S_{k-1},S_k)/d_k=0}{Max}\left(\bar{\gamma}_0(S_{k-1},S_k)\right)+\bar{\alpha}_{k-1}(S_{k-1})+\bar{\beta}_k(S_k) \tag{7}$$

$$\bar{\alpha}_k(s_k)=\underset{S_{k-1},\,i}{Max^*}\left(\bar{\alpha}_{k-1}(s_{k-1})+\bar{\gamma}_i(s_{k-1},s_k)\right) \tag{8}$$

$$\bar{\beta}_k(s_k)=\underset{S_k,\,i}{Max^*}\left(\bar{\beta}_{k+1}(s_{k+1})+\bar{\gamma}_i(s_k,s_{k+1})\right) \tag{9}$$

Branch metric

$$\bar{\gamma}_i(s_{k-1},s_k)=\frac{2}{N0}\left(Y_k^s\,X_k^s(i)+Y_k^p\,X_k^p(i,s_{k-1},s_k)+Log\left(P(s_k|s_{k-1})\right)\right)+K \tag{10}$$

Where K = Constant.

5. Max-Log-MAP Algorithm

In order to consider low decoding delay, small memory size and less complexity in hardware, we modify the algorithm called Max-Log-MAP algorithm [6]. The correction function in equation (6) is
fc $=\ln\left(1+e^{-|y-x|}\right)$ is consider as negligible term in Max-Log-MAP algorithm. Then Jacobi algorithm is loosely approximated as in equation (10) at the expense of some performance degradation.

$$Max^*(x,y)=ln\left(e^x+e^y\right)=Max(x,y)$$

The overall result of this simplification eliminates the need of LUT required to find the corresponding correction factor in max operation. Due to this simplification the BER performance at low SNR is degrades about 0.1 dB compare to Log-MAP algorithm as shown in figure 3. But from hardware point of view, the algorithm is less complex than the existing algorithm.

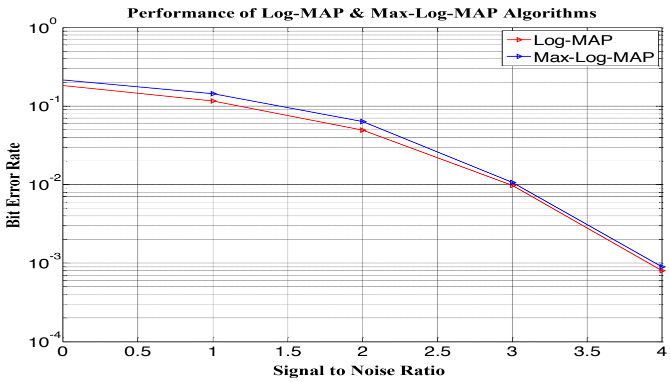

Figure 3: Performance comparison of Max-Log-MAP and Log-MAP.

6. Simulation Analysis and Results

Since turbo code offers flexibility for various decoding algorithms such as 1) Log-MAP 2) Max-Log-MAP and 3) SOVA. Simulation is carried out using turbo code. Simulation platform was same for using all the three algorithms. Parameters selected for the comparative study of these algorithms are as follows.

- Modulation technique--BPSK,

- Channel—AWGN,
- Coding Rate—1/2,
- Generator Polynomial—101,010,
- Frame size –204 symbols,
- Parameter for performance measurement—SNR Vs. BER,
- Range of SNR—0-5dB,
- Steps increment for SNR—0.5dB,
- BER requirement for reliable communication is $\leq 10^{-3}$.

Table1: Comparative performance of various algorithms.

Decoding algorithms	SNR at 10^{-3}	Iterations required
Log-MAP	3.8 dB	14
Max-Log-MAP	3.9 dB	10
SOVA	5.0 dB	10

The BER performance of Max-Log-MAP algorithm is compared to that of Log-MAP algorithm has been shown in figure 3. From the table1, It is clear that for desired BER response, Log-MAP algorithm performs better but the decoding delay is large which is obvious from large number of iterations. Working with SOVA algorithm can reduce this decoding delay at the cost of BER performance (Higher SNR requirement).

Author suggest that use of Max-Log-MAP algorithm for achieving optimum BER performance with relatively less decoding delay for a particular application, performs better compare to other two. The BER Vs SNR performances of these two algorithms have also been tested for the parameters such as (1) various numbers of frames, (2) various numbers of Iterations. The effect of variation of decoding iterations on Log-MAP algorithm is shown in figure 4 for 1000 number of frames. Figure 5 shows the performance of Max-Log-MAP for the same number of iterations. The performance of Log-MAP and Max-Log-MAP decoding algorithms for various numbers of frames for 5 iterations are shown in figure 6 and figure 7 respectively. From the graphs, it is clear that with increasing number of Iterations and Frame sizes the BER response is further improved in Max-Log-MAP algorithm.

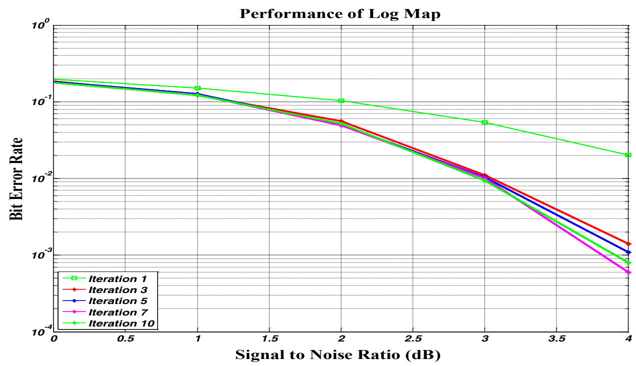

Figure 4: Effect of various Iterations on Log-MAP based Decoder.

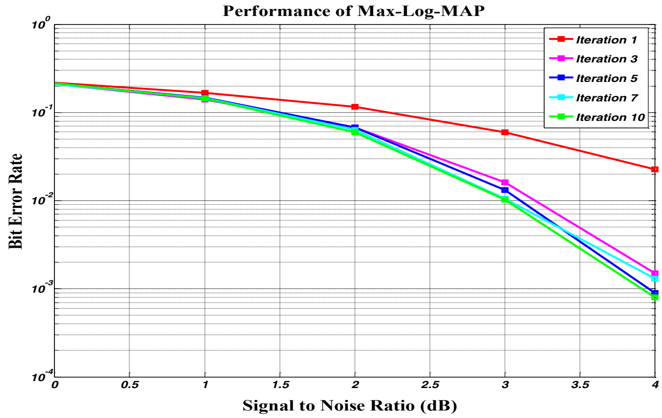

Figure 5: Effect of various Iterations on Max-Log-MAP based Decoder.

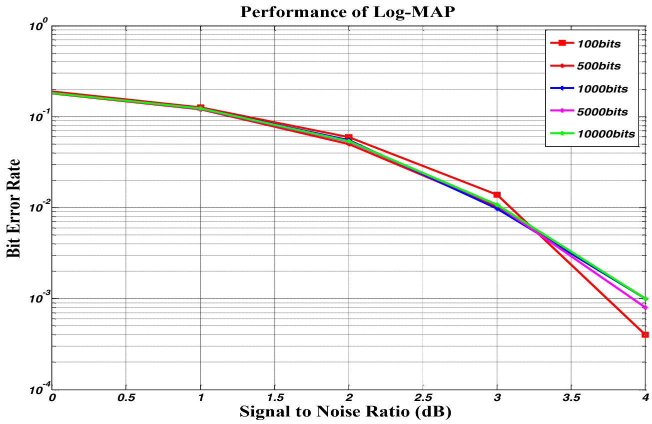

Figure 6: Effect of various Frames on Log-MAP based Decoder.

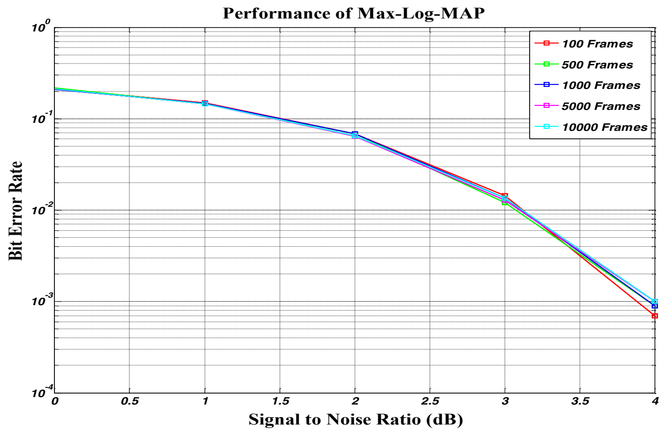

Figure 7: Effect of various Frames on Max-Log-MAP based Decoder.

7. Conclusion

The performance of turbo codes used in mobile Wi-Max system has been analyzed for Log-MAP and modified Max- Log-MAP decoding algorithm. The simulation is carried out on the same platform for both these algorithms. At low SNR, the BER performance of Max-Log-MAP decoding algorithm slightly degrades (0.1dB at Bit Error Rate 10^{-3}) compare to conventional Log-MAP algorithm. Max-Log-MAP still has a good BER response than SOVA decoding algorithm. For the desired signal power, Max-Log-MAP algorithm reduces the hardware complexity. In Max-Log-MAP algorithm, the bit error probability is further decreases with increasing number of decoding Iterations and Frames.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes, " Proceeding of IEEE ICC 93, pp. 1064-1070.
[2] Jagdish D. Kene, Kishor D. Kulat, “Performance Evaluation of IEEE 802.16e WiMax Physical Layer”, 2nd International Conference on Current Trends in Technology (NUiCONE-2011), Ahmedabad, India, Dec. 8-10, 2011.
[3] H.R. Sadjadpour and F. Park, “Maximum A Posteriori decoding algorithms for turbo codes”, In digital wireless communication II, Proc of SPIE, vol. 4045, pp. 73-83, 2000.
[4] Xiao-ming Wang, Hou-de Quan, Chi Zhang, “An improved Log-MAP decoding algorithm”, 6th IEEE Joint International Conference.
[5] P.Robertson, E.Villebrum, and P.Hoeher, “A Comparision of Optimal and Sub-Optimal MAP Decoding algorithms Operating in the Log-domain”, International Conference on Communications, PP.1009-1013, June 1995.
[6] Mustafa Taskaldiran, Richard C.S. Morling, and Izzet Kale, “The Modified Max-Log-MAP Turbo Decoding Algorithm by Extrinsic Information Scaling for Wireless Applications”, Applied DSP and VLSI Research Group, Department of Electronic Systems, University of Westminster, London, United Kingdom.
[7] T Gnanasekaran, V Aarthi, “Performance Enhancement of Modified Log MAP Decoding Algorithm for Turbo Codes” Proceedings of the International Conference on Communication and Computational Intelligence – 2010.
[8] IEEE 802.16-2006: "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems”.
[9] IEEE. Standard 802.16-2004. Part16: “Air interface for fixed broadband wireless access systems, October 2004”.
[10] M. K. Gupta, Vishwas Sharma. “To Improve Bit Error Rate of Turbo Coded OFDM Transmission over Noisy Channel”, Journal of Theoretical and Applied Information Technology, 2009.
[11] I. S. Raad, M. Yakan, “Implementation of a Turbo Codes Test Bed in the Simulink Environment”, Eighth International Symposium on Signal processing and its applications, 2005.