International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2226-2231 ISSN: 2249-6645

# The Reliability in Decoding of Turbo Codes for Wireless Communications

T. Krishna Kanth[1], D. Rajendra Prasad[2]
[1]M. Tech student, Dept of ECE, St.Ann's college of Engineering and Technology, Chirala, AP, India
[2]Assoc. Professor, Dept of ECE, St.Ann's college of Engineering and Technology, Chirala, AP, India

***ABSTRACT:*** *Turbo codes are one of the most powerful types of error control codes and high performance forward error correction codes currently available. They will be used later in the thesis as powerful building blocks in our search for better bandwidth efficient code schemes. Turbo codes emerged in 1993 and have since become a popular area of communications research. This paper provides a description of three turbo codes algorithms. Soft-output Viterbi algorithm, logarithmic-maximum a posteriori turbo algorithm and maximum- logarithmic-maximum a posteriori turbo decoding algorithms are the three candidates for decoding turbo codes. Soft-input soft-output (SISO) turbo decoder based on soft-output Viterbi algorithm (SOVA) and the logarithmic versions of the MAP algorithm, namely, Log-MAP decoding algorithm. The bit error rate (BER) performances of these algorithms are compared.*

***KEYWORDS:*** *Turbo codes, Channel coding, Iterative decoding.*

## I. INTRODUCTION

In information theory and coding theory with applications in computer science and telecommunication, error detection and correction or error control are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data. The near Shannon limit error correction performance of Turbo codes and parallel concatenated convolutional codes have raised a lot of interest in the research community to find practical decoding algorithms for implementation of these codes. The demand of turbo codes for wireless communication systems has been increasing since they were first introduced by Berrou et. al. in the early 1990's.Various systems such as 3GPP, HSDPA and WiMAX have already adopted turbo codes in their standards due to their large coding gain. In it has also been shown that turbo codes can be applied to other wireless communication systems used for satellite and deep space applications.

The MAP decoding also known as BCJR algorithm is not a practical algorithm for implementation in real systems. The MAP algorithm is computationally complex and sensitive to SNR mismatch and inaccurate estimation of the noise variance . MAP algorithm is not practical to implement in a chip. The logarithmic version of the MAP algorithm and the Soft Output Viterbi Algorithm (SOVA) are the practical decoding algorithms for implementation in this system.

## II. SHANNON–HARTLEY THEOREM

The field of Information Theory, of which Error Control Coding is a part, is founded upon a paper by Claude Shannon in 1948. Shannon calculated a theoretical maximum rate at which data could be transmitted over a channel perturbed by additive white Gaussian noise (AWGN) with an arbitrarily low bit error rate. This maximum data rate, the *capacity* of the channel, was shown to be a function of the average received signal power, W, the average noise power N, and the bandwidth of the system. This function, known as the *Shannon-Hartley Capacity Theorem*, can be stated as:

$$C=W \log_2 (1+S/N) \text{ bits/sec}$$

If W is in Hz, then the capacity, C is in bits/s. Shannon stated that it is theoretically possible to transmit data over such a channel at any rate $R \le C$ with an arbitrarily small error probability

## III. CODING IN WIRELESS COMMUNICATIONS

Coding theory is the study of the properties of codes and their fitness for a specific application and used for data compression, 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. Most digital communication techniques rely on error correcting coding to achieve an acceptable performance under poor carrier to noise conditions. Basically coding in wireless communications are of two types:

***III.1. Source coding:*** In computer science and information theory, 'data compression', 'source coding', or 'bit-rate reduction' involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. The lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it. The process of reducing the size of a data file is popularly referred to as data compression, although its formal name is source coding (coding done at the source of the data before it is stored or transmitted).

***III.2. Channel coding:*** The channel coding also called as forward corrections codes (FEC).The purpose of channel coding is to find codes which transmit quickly, contain many valid code words and can correct or at least detect many errors. Channel coding is referred to the processes done in both transmitter and receiver of a digital communications system .While not mutually exclusive, performance in these areas is a trade off. So, different codes are optimal for different applications. The needed properties of this code mainly depend on the probability of errors happening during transmission. Channel coding is distinguished from source coding, i.e., digitizing of analog message signals and data compression.

***Types of FEC Codes:***

1. Linear block codes.
2. Convolutional codes.

***1. Linear Block Codes:*** With Block Codes a block of data has error detecting and correcting bits added to it. One of the simplest error correcting block code is the Hamming Code, where parity bits are added to the data. By adding the error correcting bits to data, transmission errors can be corrected. However since more data has to be squeezed into the same channel bandwidth the more errors will occur. 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 block 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. Linear block codes are summarized by their symbol alphabets (e.g., binary or ternary) and parameters ($n$, $m$, $d_{min}$) where n is the length of the codeword, in symbols, m is the number of source symbols that will be used for encoding at once, $d_{min}$ is the minimum hamming distance for the code. Block codes submit k bits in their inputs and forwards n bits in their output. These codes are frequently known as (n,k) codes. Apparently, whatever coding scheme is, it has added n-k bits to the coded block .Block codes are used primarily to correct or detect errors in data transmission. Commonly used block codes are Reed–Solomon codes, BCH codes, Golay codes and Hamming codes.

***2. Convolutional Codes:*** Despite of block codes which are memory less, convolutional codes are coding algorithms with memory. Since, their coding rate (R) is higher than its counterpart in block codes they are more frequently used coding method in practice. Every convolutional code uses m units of memory, therefore a convolutional code represents with (n,k,m).In Convolutional coding the input bits are passed through a shift register of length K.N output bits are generated by modulo 2 adding selected bits held in different stages of the shift register. For each new data bit N output bits are produced. The output bits are influenced by K data bits, so that the information is spread in time. The channel code is used to protect data sent over it for storage or retrieval even in the presence of noise (errors).In practical communication systems, convolutional codes tend to be one of the more widely used channel codes. These codes are used primarily for real-time error correction and can convert an entire data stream into one single codeword. The Viterbi algorithm provided the basis for the main decoding strategy of convolutional codes. The encoded bits depend not only on the current informational $k$ input bits but also on past input bits.

## IV. TURBO CODES

Turbo codes are one of the most powerful types of error control codes (ECC) currently available and a class of high performance forward error correction (FEC) codes. They will be used later in the thesis as powerful building blocks in our search for better bandwidth efficient code schemes. Turbo codes emerged in 1993 and have since become a popular area of communications research. It is a combination of both block and convolutional codes. The encoder for a turbo code consists of two convolutional codes in parallel, with their inputs separated by a pseudo-random interleaver. The decoder consists of two Maximum A Posteriori (MAP) decoders connected in series via interleavers, with a feedback loop from the output of the second to the input of the first.

***IV.1. Turbo Codes Encoding:*** The encoder for a turbo code is a parallel concatenated convolutional code. Figure 1 shows a block diagram of the encoder first presented by Berrou et al. The input sequence is passed into the input of a convolutional encoder, and a coded bit stream is generated. The data sequence is then interleaved. That is, the bits are loaded into a matrix and read out in a way so as to spread the positions of the input bits. The bits are often read out in a pseudo-random manner. The interleaved data sequence is passed to a second convolutional encoder, and a second coded bit stream is generated. The code sequence that is passed to the modulator for transmission is a multiplexed (and possibly punctured) stream consisting of systematic code bits and parity bits from both the first encoder and the second encoder.

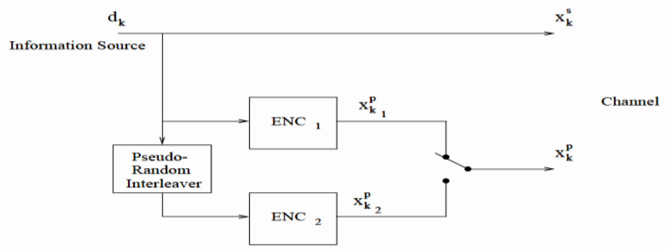


*fig 1: Structure of Turbo Encoder*

***Interleaving:*** It is a device for reordering a sequence of bits or symbols. A familiar role of interleavers in communications is that of the symbol interleaver which is used after error control coding and signal mapping to ensure that fading bursts affecting blocks of symbols transmitted over the channel are broken up at the receiver by a de-interleaver, prior to decoding.

***B. Turbo codes Decoding***

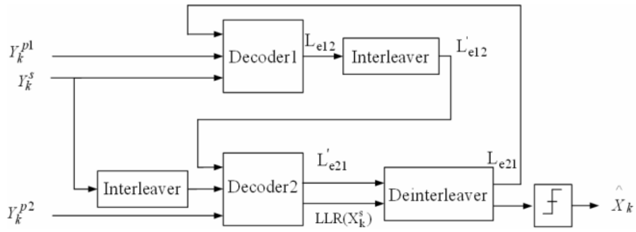


*fig 2. Iterative Turbo Decoding*

In a typical turbo decoding system (see Fig. 2), two decoders operate iteratively and pass their decisions to each other after each iteration. These decoders should produce soft-outputs to improve the decoding performance. Such a decoder is called a Soft-Input Soft- Output (SISO) decoder . Each decoder operates not only on its own input but also on the other decoder's incompletely decoded output which resembles the operation principle of turbo engines. This analogy between
the operation of the turbo decoder and the turbo engine gives

This coding technique its name, "turbo codes" .Turbo decoding process can be explained as follows: Encoded information sequence $X_k$ is transmitted over an Additive White Gaussian Noise (AWGN) channel, and a noisy received sequence $Y_k$ is obtained. Each decoder calculates the Log-Likelihood Ratio (LLR) for the k-th data bit $d_k$, as

$$L(d_k) = \log [P(d_k=1|y / P(d_k=0|y)] \quad (1)$$

LLR can be decomposed into 3 independent terms, as

$$L(d_k) = L_{apri}(d_k) + L_c(d_k) + L_e(d_k) \quad (2)$$

Where $L_{apri}(d_k)$ is the a-priori information of $(d_k)$, $L_c(d_k)$ is the channel measurement, and $L_e(d_k)$ is the extrinsic information exchanged between the constituent decoders. Extrinsic information from one decoder becomes the a-priori information for the other decoder at the next decoding stage. $L_{e12}$ and $L_{e21}$ in Figure 1 represent the extrinsic information from decoder1 to decoder2 and decoder2 to decoder1 respectively.

LLR computations can be performed by using one of the two main turbo decoding algorithms SOVA and MAP algorithms. The MAP algorithm seeks for the most likely data sequence whereas SOVA, which is a modified version of the Viterbi algorithm, seeks for the most likely connected

path through the encoder trellis. The MAP algorithm is a more complex algorithm compared to SOVA. At high SNR, the performance of SOVA and MAP are almost the same. However, at low Signal-to-Noise Ratios (SNRs) MAP algorithm is superior to SOVA by 0.5 dB or more. The following sections explain the MAP algorithm and its simplified versions Log-MAP and Max-Log-MAP algorithms.

## V. DECODING ALGORITHMS TURBO CODES

We review now the decoding algorithms used within $DEC_1$ and $DEC_2$ to implement the soft input, soft-output processing needed for iterative decoding. We begin with the Maximum A Posteriori (MAP), algorithm. Decoding of convolutional codes is most frequently achieved using the Viterbi algorithm, which makes use of a decoding *trellis* to record the estimated states of the encoder at a set of time instants. The Viterbi algorithm works by rejecting the least likely path through the trellis at each node, and keeping the most likely one. The removal of unlikely paths leaves us, usually, with a single source path further back in the trellis. This path selection represents a 'hard' decision; on the transmitted sequence.

The Viterbi decoder estimates a maximum likelihood *sequence*. Making hard decisions in this way, at an early point in the decoding process, represents a loss of valuable information. It is frequently advantageous to retain finely-graded probabilities, 'soft decisions', until all possible information has been extracted from the received signal values. The turbo decoding relies on passing information about individual transmitted bits from one decoding stage to the next. The interleaving of the received information sequence between decoders limits the usefulness of estimating maximum likelihood *sequences*. So, an algorithm is required that can output soft-decision maximum likelihood estimates on a *bit-by-bit* basis. The decoder should also be able to accept soft decision inputs from the previous iteration of the decoding process. Such a decoder is termed a Soft Input-Soft Output (SISO). Berrou and Glavieux used two such decoders in each stage of their turbo decoder. They implemented the decoders using a modified version of an SISO algorithm proposed by Bahl, Cocke, Jelinek and Raviv [31]. Their *modified Bahl algorithm* is commonly referred to as the *Maximum A Posteriori* or MAP algorithm, and achieves soft decision decoding on a bit-by-bit basis by making two passes of a decoding trellis, as opposed to one in the case of the Viterbi algorithm. The MAP algorithm is an optimal but computationally complex SISO algorithm. The Log-MAP and Max-Log-MAP algorithms are simplified versions of the MAP algorithm. MAP algorithm calculates LLRs for each information bit as

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

Where α is the forward state metric, β is the backward state metric,γ is the branch metric, and k S is the trellis state at trellis time k . Forward state metrics are calculated by a forward recursion from trellis time k = 1 to, k = N where N is the number of information bits in one data frame. Recursive
calculation of forward state metrics is performed as

$$\alpha_k (S_k) = \sum_{j=0}^{1} \alpha_{k-1}(S_{k-1})\gamma_j(S_{k-1}.S_k) \quad (4)$$

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

$$\beta_k (S_k) = \sum_{j=0}^{1} \beta_{k-1}(S_{k+1})\gamma_j(S_k.S_{k+1}) \quad (5)$$

Branch metrics are calculated for each possible trellis transition as

$$\gamma_i(S_{k-1}, S_k) = A_k P(S_k \mid S_{k-1}) \exp\left[\frac{2}{N_o}\left(y_k^s x_k^s(i) + y_k^p x_k^p(i, S_{k-1}, S_k)\right)\right] \quad (6)$$

Where i = (0,1) , k A is a constant, $x^s_k$ and $x^p_k$ are the encoded systematic data bit and parity bit, and, $y^s_k$ and $y^p_k$ are the received noisy systematic data bit and parity bit respectively.

***LOG-MAP ALGORITHM:*** To avoid complex mathematical calculations of MAP decoding, computations can be performed in the logarithmic domain. Furthermore, logarithm and exponential computations can be eliminated by the following approximation.

$$\max^*(x,y) \triangleq \ln(e^x + e^y) = \max(x,y) + \log(1 + e^{-|y-x|})$$

So equations (3)-(6) become

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

$$\bar{\alpha}_k(S_k) = \max^*_{(S_{k-1}, i)}\left(\bar{\alpha}_{k-1}(S_{k-1}) + \bar{\gamma}_i(S_{k-1}, S_k)\right)$$

$$\bar{\beta}_k(S_k) = \max^*_{(S_k, i)}\left(\bar{\beta}_{k+1}(S_{k+1}) + \bar{\gamma}_i(S_k, S_{k+1})\right)$$

$$\bar{\gamma}_i(S_{k-1}, S_k) = \frac{2}{N_o}\left(y_k^s x_k^s(i) + y_k^p x_k^p(i, S_{k-1}, S_k)\right) + \ln(P(S_k \mid S_{k-1})) + K$$

Where K is a constant.

The Log-MAP parameters are very close approximations of the MAP parameters and therefore, the Log-MAP BER performance is close to that of the MAP algorithm.

***MAX-LOG-MAP ALGORITHM:*** The correction function $f_c = \log(1 + e^{|y-x|})$ in the * max (.) operation can be implemented in different ways. The Max-log-MAP algorithm simply neglects the correction term and approximates the * max (.) operator as at the expense of some performance degradation. This simplification eliminates the need for an LUT required to find the corresponding correction factor in the * max (.) operation.

## VI. PRINCIPLES OF ITERATIVE DECODING

In a typical communications receiver, a demodulator is often designed to produce soft decisions, which are then transferred to a decoder. The improvement in error performance of systems utilizing such soft decisions is typically approximated as 2 dB, as compared to hard decisions in AWGN. Such a decoder could be called a soft input/ hard output decoder, because the final decoding process out of the decoder must terminate in bits (hard decisions). With turbo codes, where two or more component codes are used, and decoding involves feeding outputs from one decoder to the inputs of other decoders in an iterative fashion, a hard-output decoder would not be suitable. That is because a hard decision into a decoder degrades system performance (compared to soft decisions).

Hence, what is needed for the decoding of turbo codes is a soft input/ soft output decoder. For the first decoding iteration of such a soft input/soft output decoder, we generally assume the binary data to be equally likely, yielding an initial a priori LLR value of L(d)=0. The channel LLR value, Lc(x), is measured by forming the logarithm of the ratio of the values. The output L(d) of the decoder in Figure 3 is made up of the LLR from the detector, L'(d) , and the extrinsic LLR output, Le(d) ,representing knowledge gleaned from the decoding process. As illustrated in Figure 3, for iterative decoding, the extrinsic likelihood is fed back to the decoder input, to serve as a refinement of the a priori probability of the data for the next iteration.

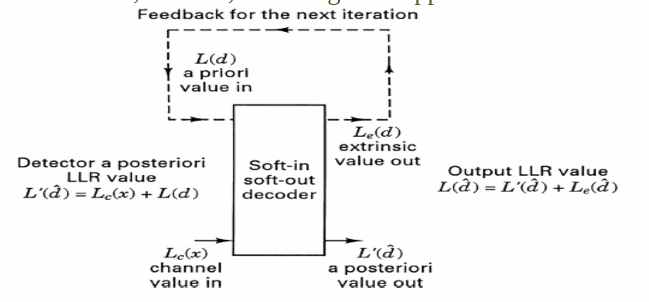


*fig3:soft input soft output decoder*

## VII. SIMULATION RESULTS

The simulation curves presented shows the influence of iteration number, Block length, code rate and code generator. Rate ½ codes are obtained from their rate 1/3 counterparts by alternately puncturing the parity bits of the constituent encoders. In figures (4-5) BER for SOVA and LOG MAP as a function of Eb/No curves are shown for constituent codes of constraint length three and code rate ½. Eight decoding iterations were performed for Block length of 1024 . Also the improvement achieved when the block length is increased from 1024 to 4096 for both algorithms. For figure 6, LOG MAP shows better performance than SOVA for constraint length of three and for block length of 1024.And from the figure 7,we can observe the BER performances of LOG MAP and MAX-LOG MAP algorithms. The MAX-LOG MAP algorithm gives better BER performance.

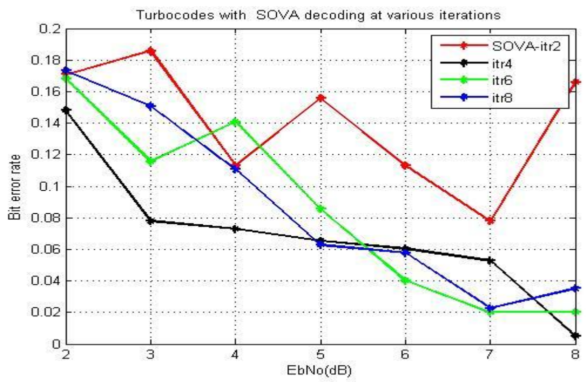


*fig4: Iterations performed by sova decoding algorithm*

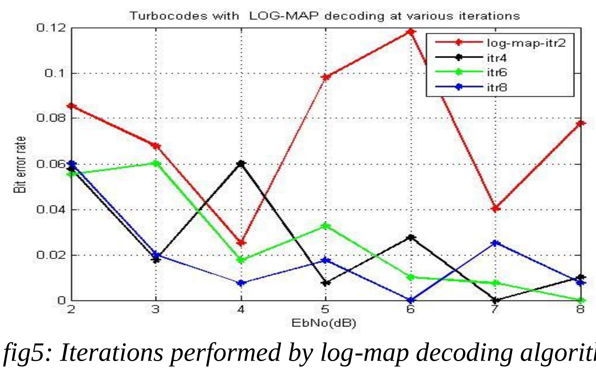


*fig5: Iterations performed by log-map decoding algorithm*

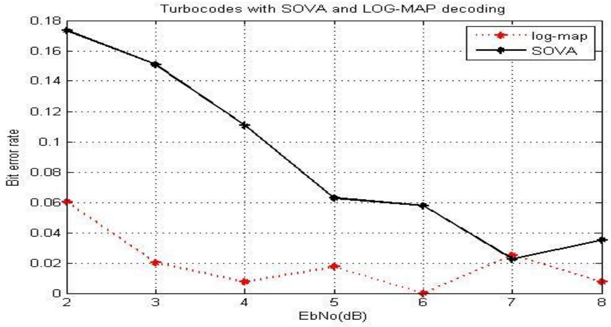


*fig6:BER performances by SOVA and Log-Map decoding algorithms*

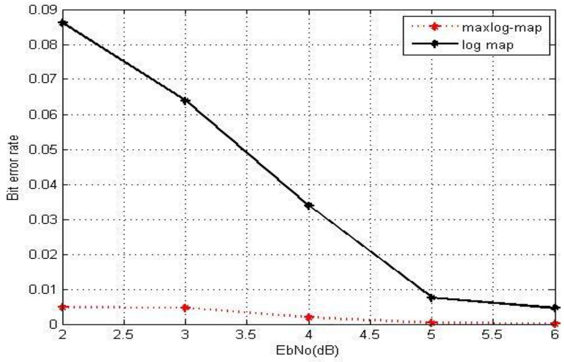


*fig7:BER performances by Max Log-Map and Log-Map decoding algorithm*

## VIII. CONCLUSION

Our Simulation results shows that the decoding algorithms of Max-Log MAP performs better in terms of block length compared to SOVA and Log-MAP, and thus it is more suitable for wireless communication.

## REFERENCES

[1]. Grace Oletu, Predrag Rapajic," The Performance of Turbo codes for Wireless Communication Systems,IEEE2011.

[2]. 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.

[3]. S. Benedetto, G. Montorsi, "Design of Parallel Concatenation Convolutional Codes: IEEE Trans. on communication, vol. 44,No.5, May 1996.

[4]. C. Berrou, "The Ten-Year-Old Turbo Codes are Entering into Service," IEEE Commun. Mag. vol. 41, no. 8, pp.110-116, Aug 2003.

[5]. L. Bahi, J. Cocke, F. Jelinek, and J. Raviv, "Optimum decoding of linear codes for minimizing symbol error rate," IEEE Trans.on Inf. Theory, vol. IT-20, pp. 284-287, Mar. 1974.

[6]. 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.

[7]. P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and Sub-Optimal Maximum A Posteriori Algorithms Suitable for Turbo Decoding,"European Trans. on Telecomm. vol. 8, no. 2, pp. 119-126, March-April 1997.

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

[9]. 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.

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

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

[12]. J.Hagenauer E.Offer, and L.Papke,"Iterative Decoding of Binary Block and Convolutional Codes," IEEE Trans.Inform. Theory, 42:429-45, March 1996.

[13]. W.J. Gross and P.G. Gulak, "Simplified MAP algorithm suitables for implementation of turbo decoders" Electronic Letters, vol. 34, no.16, August 6, 1998.

[14]. J. Hagenauer, L. Papke, "Decoding Turbo Codes With the Soft Output Viterbi Algorithm (SOVA)," in Proc. Int. Symp.On Information Theory, p164, Norway, June 1994.

[15]. J. Hagenauer, P. Robertson, L. Papke, "Iterative Decoding of Systematic Convolutional Codes With the MAP and SOVA Algorithms," ITG Conf., Frankfurt, Germany,pp 1-9, Oct. 1994

[16]. J. Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Bloc and Convolutional Codes," IEEE Trans. Infor. Theory, Vol. IT. 42, No.2, pp 429-445, March 1996.