Effect of Various CODEC Parameters on the Performance of Modified Max-Log-MAP Turbo Decoding Algorithm

R.KRISHNAMOORTHY[1], N.S.PRADEEP[2] and V.AARTHI[3]
[1]Department of CSE, Anna University-Chennai, BIT Campus,
Trichy-24, Tamilnadu, India.
[2]Department of ECE, Anna University-Chennai, BIT Campus,
Trichy-24, Tamilnadu, India.
[3]Department of ECE, Velalar College of Engineering and Technology,
Erode, Tamilnadu, India.

rkrish26@hotmail.com, pine_deep@yahoo.com, aarthivellingiri@gmail.com

Abstract: - Turbo decoder uses any one of the decoding algorithms, Maximum A posteriori Probability (MAP), or Soft Output Viterbi Algorithm(SOVA) because it produces error correction near to Shannon's limit .The Max-Log-MAP is a Soft Input Soft Output (SISO) algorithm, which determines the probability of most likely path through the trellis and hence it gives sub optimal performance compared to Log-MAP algorithm. A simple but effective technique to improve the performance of Max-Log-MAP (MLMAP) algorithm is to scale the extrinsic information exchanged between two decoders using appropriate Scaling Factors (SF). Modified Max-Log-MAP (M-MLMAP) algorithm is achieved by fixing an arbitrary SF for inner decoder S_2 and an optimized SF for the outer decoder S_1. This paper presents the performance analysis for the Modified Max-Log-MAP decoding algorithm by optimizing the scaling factor S_1 to achieve low Bit Error Rate (BER). The performance of various scaling factors is compared and optimized scaling factor is obtained, which is an empirical value. Appropriate mathematical relationship between scaling factor and E_b/N_0 is also proposed. The effect of the proposed algorithm for a range of CODEC parameters is investigated in a systematic fashion, in order to gauge their performance ramifications. The use of an emphatically determined optimal scaling factor improved the performance of MLMAP decoding algorithm in terms of BER. The proposed M-MLMAP algorithm showed a gain of 0.75dB over MLMAP algorithm at BER of $2x10^{-5}$ for fading channel.

Key-Words:- BER, Log-MAP, Max-Log-MAP, Scaling Factor, Turbo codes.

1 Introduction

A major advancement in the channel coding area was introduced by Berrou et al. in 1993 by the advent of turbo codes [3]. Turbo codes have shown the best Forward Error Correction (FEC) performance known up to now. Turbo codes are revolutionary in the sense that they allow reliable data transmission within a half decibel of the Shannon's Limit. A massive amount of research effort has been performed to facilitate the energy efficiency of turbo codes. As a result, turbo codes have been incorporated into many standards used by the NASA Consultative Committee for Space Data Systems (CCSDS) [4], Digital Video Broadcasting (DVB) [6], both Third Generation Partnership which requires throughputs from 2 Mb/s to several 100 Mb/s, in Project (3GPP) [18] standards for IMT-2000, Wideband CDMA, 4G and WIMAX. The iterative nature of Turbo decoding algorithms increase their complexity compared to conventional FEC decoding algorithms. Two iterative decoding algorithms, Soft Output Viterbi Algorithm [9]-[11] and Maximum A posteriori Probability [14], [15] algorithm require complex decoding operations over several iteration cycles. So, for real-time implementation of turbo codes, reducing the decoder complexity while preserving BER performance is an important design consideration. In this paper, a modification to the Max-Log-MAP algorithm is investigated and its performance is analyzed. Section II gives an overview of the turbo decoding process, the MAP algorithm and its simplified versions, the Log-MAP and Max-Log-MAP algorithms. The extrinsic information scaling is introduced in Section III. Section IV presents simulation results and performance analysis of the proposed algorithm for various CODEC parameters. It is also obtained the mathematical relationship to select the best scaling factor for a given E_b/N_0 at low BER with reduced complexity.

2 Turbo Decoder

Turbo decoder uses any one of the decoding algorithm, MAP or SOVA [16] because it produces error correction near to Shannon's limit [17]. In a typical Turbo decoding system shown in Fig. 1, two decoders operate iteratively and pass their decisions to each other after each iteration. These decoders produce soft-outputs to improve the decoding performance. Such a decoder is called a SISO decoder [9]. 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".

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

$$L(d_k) = \log\left[\frac{P(d_k = 1|Y)}{P(d_k = 0|Y)}\right] \qquad (1)$$

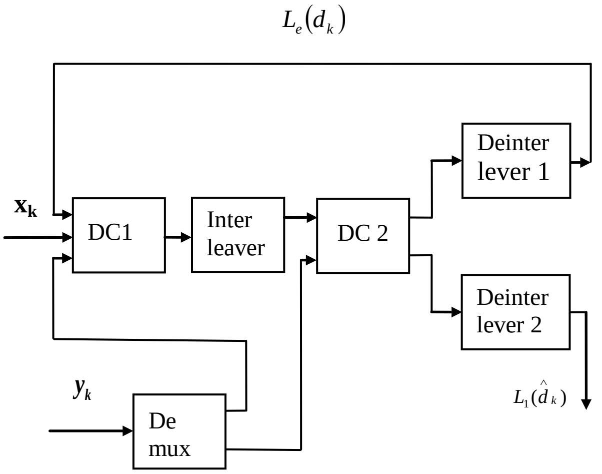

Fig.1 Turbo Decoder

LLR can be decomposed into 3 independent terms, as

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

Where $L_{apri}(d_k)$ is the a-priori information of d_k, $L_c(d_k)$ is the channel measurement, $L_e(d_k)$ is the extrinsic information. Extrinsic information from one decoder becomes the a-priori information for the other decoder at the next decoding stage. LLRs can be calculated by two different SISO algorithms SOVA and MAP Algorithm. In this paper Max-Log-MAP algorithm is investigated.

2.1 The MAP Algorithm

The MAP algorithm [15] 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

$$L(d_k) = \log\left[\frac{\sum_{S_k}\sum_{S_{k-1}} \gamma_1(S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k(S_k)}{\sum_{S_k}\sum_{S_{k-1}} \gamma_0(S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k(S_k)}\right] \qquad (3)$$

where α is the forward state metric, β is the backward state metric, γ is the branch metric, and S_k 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

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

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

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

Branch metrics are calculated for each possible trellis transition as

$$\gamma_i(S_{k-1}, S_k) = A_k P(S_k|S_{k-1}) \exp\left[\frac{2}{N_0}\left(y_k^s x_k^s(i) + y_k^p x_k^p(i, S_{k-1}, S_k)\right)\right] \qquad (6)$$

where i=(0,1), A_k is a constant, x_k^s and x_k^p are the encoded systematic data bit and parity bit, y_k^s and y_k^p are the received noisy systematic data bit and parity bit respectively.

2.2 The Log-MAP Algorithm

To avoid complex mathematical calculations of MAP decoding, computations can be performed in the logarithmic domain [15]. Furthermore, logarithm and exponential computations can be eliminated by the following approximation

$$\max{}^*(x,y) \triangleq \ln\left(e^x + e^y\right) = \max(x,y) + \ln\left(1 + e^{-|y-x|}\right) \quad (7)$$

The last term in max*(.) operation can easily be calculated by using a look-up table (LUT). So equations (3)-(6) become

$$L(d_k) = \max_{(S_{k-1},S_k)}{}^*\left(\bar{\gamma}_1(S_{k-1},S_k) + \bar{\alpha}_{k-1}(S_{k-1}) + \bar{\beta}_k(S_k)\right) - \max_{(S_{k-1},S_k,0)}\left(\bar{\gamma}_0(S_{k-1},S_k) + \bar{\alpha}_{k-1}(S_{k-1}) + \bar{\beta}_k(S_k)\right) \quad (8)$$

$$\bar{\alpha}_k(S_k) = \max_{S_{k-1},i}{}^*\left(\bar{\alpha}_{k-1}(S_{k-1}) + \bar{\gamma}_i(S_{k-1},S_k)\right) \quad (9)$$

$$\bar{\beta}_k(S_k) = \max_{S_k,i}{}^*\left(\bar{\beta}_{k+1}(S_{k+1}) + \bar{\gamma}_i(S_k,S_{k+1})\right) \quad (10)$$

$$\bar{\gamma}_i(S_{k-1},S_k) = \frac{2}{N_0}\left(y_k^s x_k^s(i) + y_k^p x_k^p(i,S_{k-1},S_k)\right) + \log\left(P\left(S_k|S_{k-1}\right)\right) + \log\left(P\left(S_k|S_{k-1}\right)\right) + K \quad (11)$$

where K is a constant.

2.3 The Max-Log-MAP Algorithm

The correction function $f_c = log\left(1 + e^{-|y-x|}\right)$ in the max*(.) operation can be implemented in different ways. The Max-Log-MAP algorithm [12]-[14] simply neglects the correction term and approximates the max*(.) operator as

$$ln(e^x + e^y) \approx \max(x,y) \quad (12)$$

at the expense of some performance degradation. Because of this approximation Max-Log-MAP algorithm gives sub-optimal performance.

3 Proposed Modified Max-Log-MAP (M-MLMAP) Decoding algorithm

It has been proposed to scale the extrinsic information exchanged between the constituent decoders [5], [7], and [8]. With this modification equation (11) for branch metric calculations can be rewritten as

$$\bar{\gamma}_i(S_{k-1},S_k) = \frac{2}{N_0}\left(y_k^s x_k^s(i) + y_k^p x_k^p(i,S_{k-1},S_k)\right) + \log\left(P\left(S_k|S_{k-1}\right)\right) + s_i \log\left(P\left(S_k|S_{k-1}\right)\right) + K \quad (13)$$

The only modification is the scaling factor s_i where i = 1, 2 for decoder1 and decoder2 respectively.

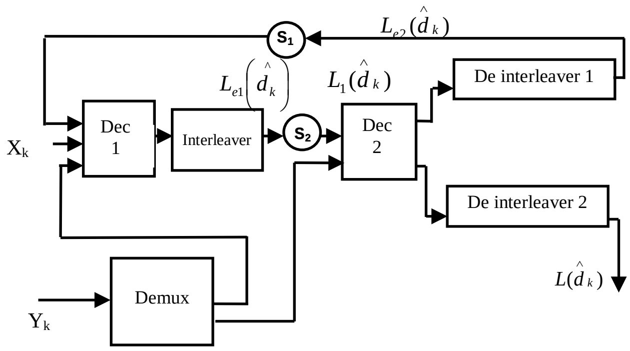

Fig.2 Turbo Decoder with Scaling Factors

Max-Log-MAP algorithm suffers from two distortions [2]: over-optimistic soft outputs and correlation between the intrinsic and extrinsic information. The performance is degraded substantially due to first of these distortions and mildly due to the second. The first type of distortion, which depends on E_b/N_0, is considered. The compensation co-efficient is calculated. The compensation of $L_e(d_k)$ is possible with a scaling factor. Algorithms are modified by multiplying extrinsic information $L_e(\hat{d}_k)$ with the chosen scaling factor before it is being fed back to the input. The scaling factor must be chosen in such a way that it gives substantial improvement in the reliability of output from the decoder and decreases the number of iterations involved in attaining the Shannon's capacity limit of error performance. M-MLMAP algorithm is achieved by fixing an arbitrary value for inner decoder (S_2) and an optimized value for the outer decoder (S_1), which gives lowest BER. Scaling factor S_1 depends on E_b/N_0 to give low BER and better performance than MLMAP algorithm. The proposed Turbo decoder with optimized scaling factors is shown in Fig. 2. The algorithms are enhanced by multiplying the extrinsic information $L_e(\hat{d}_k)$ with the optimized scaling factor S_1 before it is being fed back to the input and decoder 2 respectively and is given by

$$z_k = \left[L_{e2}(\hat{d}_k)\right] \times S_1^* \quad (14)$$

$$L_1(\hat{d}_k) = \left[\frac{2}{\sigma^2}x_k + L_{e1}(\hat{d}_k)\right] \times S_2 \quad (15)$$

Where * in equation (14) indicates that the scaling factor S_1 is optimized to get least BER at a given E_b/N_0.

4 Simulation Results and Discussion

4.1 Simulation Profile

The proposed scaling factors for the turbo coded system are simulated in AWGN channel. Transmission of 500 frames with a constant frame length of 2048 bits and random interleaver [1] is taken to show the effect of the scaling factors on the performance of error correction. Simulation results have been gathered with different combinations of scaling factors at different E_b/N_0 to view the least BER at the decoder side. The simulation parameters are,

- Channel: AWGN
- Modulation: Quadrature Phase shift Keying (QPSK)
- Component Encoders : Two identical Récursive Convolution codes (RSCs)
- Rate=1/2 (punctured)
- Interleaver: 2048 bit random interleaver
- Iteration: 8
- Frame limit: 500

The scaling factors considered range from 0.05 to 0.95 for M-MLMAP algorithm and is shown in Fig. 3. A wide range of scaling factors, the E_b/N_0 and the corresponding BER has been showed. The scaling factor having the lowest BER for a particular E_b/N_0 is considered to be optimized SF.

Table I shows the optimized scaling factor (S_1), having the lowest BER against E_b/N_0. It is observed from Table 1 that S_1 is found to vary with E_b/N_0 and hence it is not only optimal but also adaptive with respect to E_b/N_0. Unlike in [13] we have used adaptive SF, rather than fixed SF.

Table 1: Optimized Scaling Factor (S_1) and BER for varying E_b/N_0

E_b/N_0 (dB)	Optimized Scaling Factor (S_1)	Corresponding BER
0	0.95	$1.0720\text{x}10^{-1}$
0.5	0.90	$6.2454\text{x}10^{-2}$
1	0.90	$1.3669\text{x}10^{-2}$
1.5	0.90	$2.1744\text{x}10^{-4}$
2	0.85	$1.9767\text{x}10^{-5}$
2.5	0.85	$1.9767\text{x}10^{-5}$
3	0.90	$9.6767\text{x}10^{-6}$
3.5	0.95	$8.2034\text{x}10^{-6}$
4	0.90	$1.1836\text{x}10^{-6}$

Fig. 4 shows the performance of Modified Max-Log-MAP algorithm with the scaling factors S_1=0.85(optimal) and S_2=0.75(arbitrary) is giving better results comparing with the Max-Log-MAP algorithm without scaling factor. The MLMAP and M-MLMAP algorithms are also compared with the standard Log-MAP decoding algorithm, at E_b/N_0 of 2dB in AWGN channel. It is noted from the Fig.4 that Log-MAP which is an optimal algorithm gives better performance than sub-optimal Max-Log-MAP algorithm. But with the introduction of appropriate scaling factor, the performance of Max-Log-MAP algorithm is improved and is found that the proposed M-MLMAP algorithm gave optimal performance with BER of $5\text{x}10^{-6}$ for iteration 5. The BER of Log-MAP and MLMAP algorithms are 1×10^{-5} and 2×10^{-5} respectively for iteration 5.

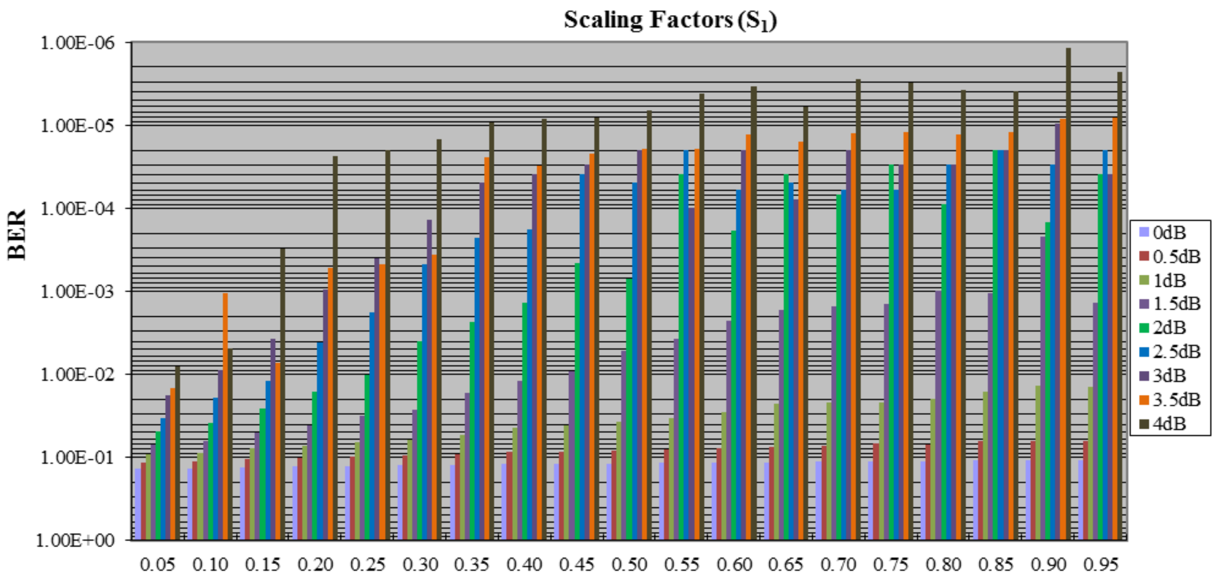

Fig. 3 BER plot of various Scaling Factors and E_b/N_0 with code generator (7,5), punctured for Max-Log-MAP algorithm.

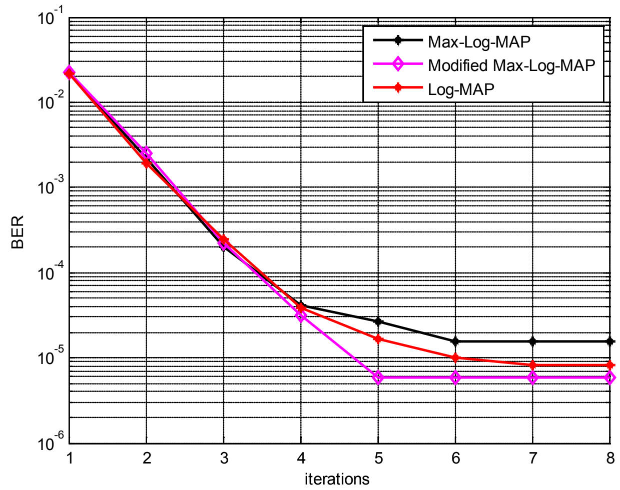

Fig. 4 BER values of Log-MAP, Max-Log-MAP and Modified Max-Log-MAP decoding algorithms for different iterations. Code generator (7,5), punctured, frame length=2048, frame limit=500, for 2dB in AWGN channel.

It is also observed from Fig. 4 that the BER performance of M-MLMAP algorithm remains constant from iteration 5. It is revealed for M-MLMAP algorithm, the efficient BER has been achieved by 5 iterations. Thus in the proposed M-MLMAP algorithm, complexity has been reduced by 37.5% compared to standard MLMAP algorithm and the BER has been reduced by the order of 10^{-1} compared to MLMAP algorithm. The main design criterion for any decoding algorithm is to reduce the BER and complexity, which is achieved by the proposed M-MLMAP algorithm.

Table II gives the summary of the number of iterations, BER and the percentage of reduction in complexity for each decoding algorithms. Compared to Log-MAP algorithm, the complexity of Max-Log-MAP algorithm is reduced but at the cost of BER. But the proposed M-MLMAP algorithm gives improved performance with least complexity.

Analyses are carried out to show the performance of decoding algorithms in AWGN and Fading channels, with QPSK modulation.

Table 2: Number of Iterations Required For Each Decoding Algorithm

Decoding Algorithms	**Iteration from which BER is constant**	**Complexity reduced in %**	**Corresponding BER**
Log-MAP	7	12.5	$8.1460\text{x}10^{-6}$
MLMAP	6	25	$1.5740\text{x}10^{-5}$
M-MLMAP	**5**	**37.5**	$\mathbf{5.8887x10^{-6}}$

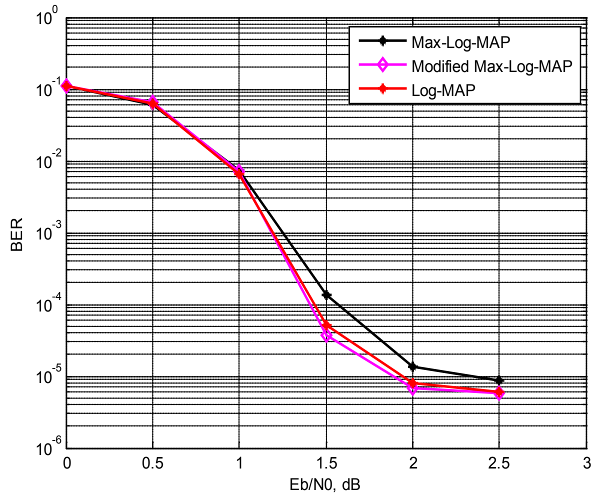

Fig. 5 Performance comparison of Log-MAP, Max-Log-MAP and Modified Max-Log-MAP in AWGN channel. Code generator (7,5), punctured, frame length=2048, frame limit=500.

Fig. 5 shows the performance of Log-MAP, Max-Log-MAP and Modified Max-Log-MAP in AWGN channel. At E_b/N_0 of 1.5dB and above, M-MLMAP algorithm is better than MLMAP with BER of 5×10^{-6} at E_b/N_0 of 2.5dB. The proposed M-MLMAP algorithm achieves performance closer to optimal Log-MAP algorithm. M-MLMAP gives better performance than MLMAP with a gain of 0.3dB at BER of $3\text{x}10^{-5}$ on the curve.

Similar analysis is being done for the Rayleigh Fading channel and is shown in Fig. 6.

The performance in fading channel is almost identical to the AWGN channel with M-MLMAP and Log-MAP algorithms giving almost identical performances. On comparing MLMAP and proposed M-MLMAP algorithms, later showed a gain of 0.75dB at BER of $2x10^{-5}$ on the curve, which validates the robustness of the proposed algorithm.

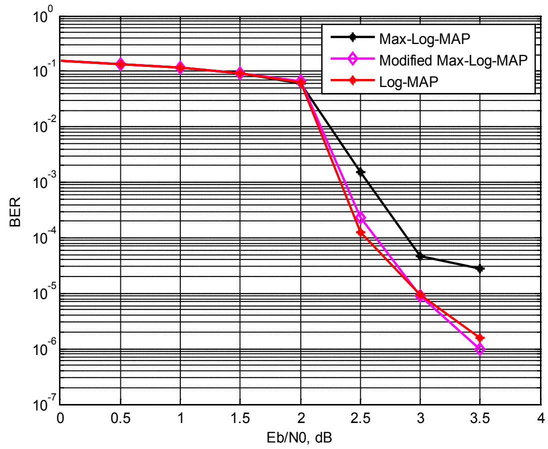

Fig. 6 Performance comparison of Log-MAP, Max-Log-MAP and Modified Max-Log-MAP in Rayleigh Fading channel. Code generator (7,5), punctured, frame length=2048, frame limit=500.

The following has been observed from above graphs: Log-MAP algorithm is optimal in enactment but complex; MLMAP algorithm is simple but gives non-optimal performance; the proposed M-MLMAP is both simple and optimal. Hence M-MLMAP algorithm is best suited for practical applications.

Fig. 7 is a plot between scaling factor (S_1) and E_b/N_0 for M-MLMAP algorithm.

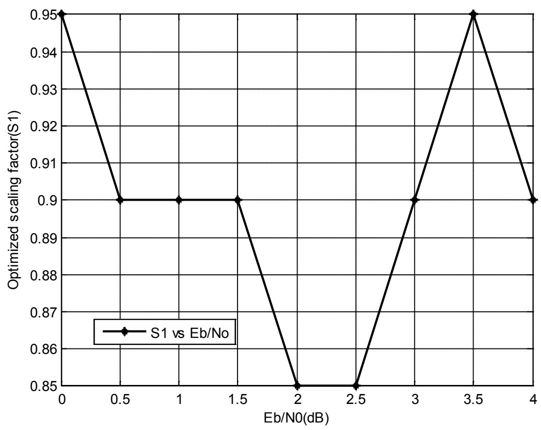

Fig. 7 Plot for Modified Max-Log-MAP between E_b/N_0 and optimum scaling factors (S_1) for AWGN channel, code generator (7,5), punctured, frame length=2048, frame limit=500, and scaling factor S_2=0.75.

It shows the variation of magnitude of scaling factor S_1 with respect to the E_b/N_0 where scaling factor S_2 is kept constant. It can be inferred from the plot that there is a relational dependence between scaling factor S_1 and E_b/N_0. The graph also shows variation in the scaling factors for E_b/N_0. The variation between these two parameters is considered to make the Turbo decoder as adaptive. The adaptive decoder, by itself, will set the scaling factor of the decoder corresponding to the received E_b/N_0. To make it adaptive, we have obtained an expression by the curve fitting method.

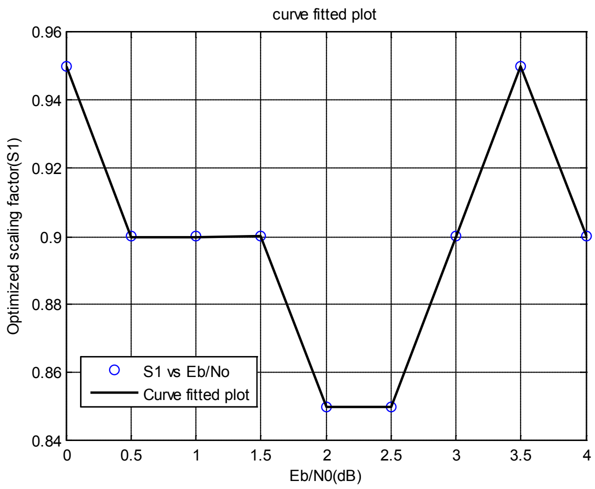

Fig. 8 Curve fitted plot between E_b/N_0 and optimum scaling factor for M-MLMAP.

Fig.8 shows the graph fitted with best accuracy so as to generate a polynomial expression which helps to determine the appropriate optimized scaling factor for the given E_b/N_0.

Equation 16 shows a polynomial expression of eighth degree with nine coefficients for proposed M-MLMAP algorithm. The expression is given by,

$$f(x) = -0.0067x^8 + 0.1111x^7 - 0.7589x^6 + 2.7178x^5 - 5.4093x^4 + 5.8236x^3 - 3.0001x^2 + 0.4725x + 0.95 \quad (16)$$

Authors of [13] have reported 0.2-0.4dB gain over the standard Max-Log-MAP algorithm for 3GPP standards. They used a constant scaling factor of 0.7. But in our paper, the optimized scaling factor S_1 is adaptive with respect to E_b/N_0 as shown in Fig. 7 with a gain of 0.75dB in fading channel.

4.2 The Effect of Various CODEC Parameters on the performance of M-MLMAP algorithm

In this section we present simulation results for turbo codes with Modified Max-Log-MAP decoding algorithm using Quadrature Phase Shift Keying (QPSK) over Additive White Gaussian Noise (AWGN) channels. We show that there are many parameters, some of which are interlinked, which affect the performance of M-MLMAP. Some of these parameters are:

- The number of decoding iterations used.
- The frame length or latency of the input data.
- The generator polynomials of the component codes.
- The constraint lengths of the component codes.
- The effect of code rates.

The standard parameters that we have used in our simulations are shown in Section 4.1. The turbo encoder uses two component RSCs in parallel. The RSC component codes are codes with generator polynomials in octal or decimal representation. These generator polynomials are optimum in terms of maximizing the minimum free distance of the component codes. The effects of changing these parameters are examined in Section 4.2.3. The standard interleaver used between the two component RSC codes is a 2048–bit random interleaver. Unless otherwise stated, the results in this section are for half-rate codes, where half the parity bits generated by each of the two component RSC codes are punctured. Usually 8 iterations of the component decoders are used, but in the next section we consider the effect of the number of iterations.

4.2.1 The Effect of the Number of Iterations Used

Fig. 9 shows the performance of a turbo decoder using the proposed M-MLMAP algorithm versus the number of decoding iterations which were used. As the number of iterations used by the proposed turbo decoder increases, the turbo decoder performs significantly better. However, after eight iterations there is little improvement achieved by using further iterations. For example, it can be seen from Fig. 9 that using 16 iterations rather than eight gives an improvement of only about 0.2dB.

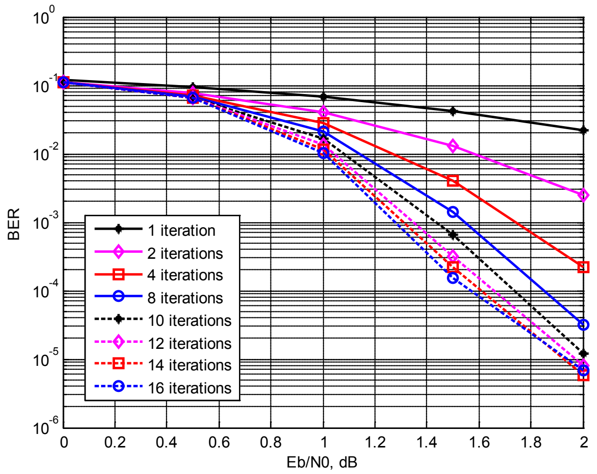

Fig. 9 BER performance using different numbers of iterations for the M-MLMAP Algorithm.

Hence, for complexity reasons usually only about eight iterations are used in all our simulations.

4.2.2 The Effect of the Frame Length of the Code

In the original paper on turbo coding by Berrou et al. [3], and many of the subsequent papers, impressive results have been presented for coding with very large frame lengths. For many applications, such as speech transmission systems, the large delays inherent in using high frame lengths are unacceptable. Therefore, an important area of turbo coding research is achieving as impressive results with short frame lengths as have been demonstrated for long frame length systems.

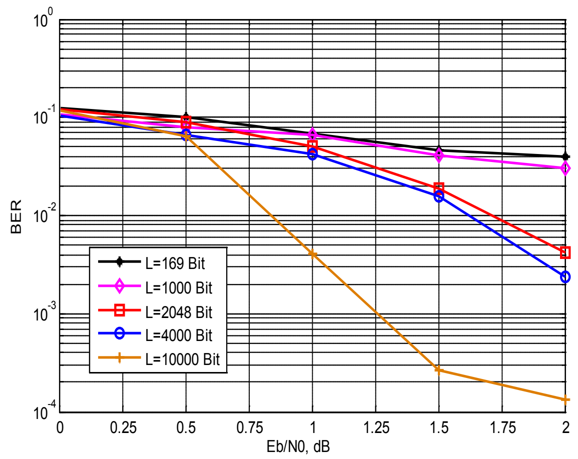

Fig.10 Effect of frame length on the BER performance of M-MLMAP Turbo decoding algorithm.

Fig. 10 shows how dramatically the performance of Turbo coder with proposed algorithm depends on the frame length used in the encoder. The 169-bit code would be suitable for use in a speech transmission system at approximately 8 kb/s with a 20-ms frame length, while the 1000 bit code would be suitable for video transmission. The larger frame length systems would be useful in data or non-real time transmission systems.

It can be seen from Fig. 10 that the performance of turbo codes is very impressive for systems with long frame lengths (for 10,000 bit code). In the simulation of M-MLMAP algorithm, frame length of 2048 bit is used.

4.2.3 The effect of the generator polynomials of the component codes

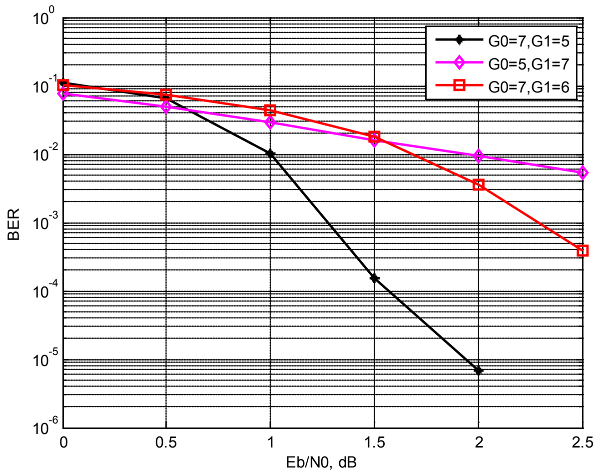

Fig.11 Effect of generator polynomials on BER performance of M-MLMAP Turbo decoding algorithm.

Both the constraint length and the generator polynomials used in the component codes of turbo codes are important parameters.

Fig.11 shows the huge difference in performance that can result from different generator polynomials being used in the component codes. The other parameters used in these simulations were the same as detailed above in Section 4.1. All the results given in this paper were obtained using constraint length three component codes. For these codes we have used the optimum generator polynomials in terms of maximizing the minimum free distance of the component convolutional codes, i.e., 7 and 5 in octal representation. These generator polynomials were also used for constraint length three turbo coding by Hagenauer et al. in [11]. It can be seen from Fig. 11 that the order of these generator polynomials is important—the octal value 7 should be used for the feedback generator polynomial of the encoder (denoted here by G0). If G0 and G1 are swapped round, turbo codes with M-MLMAP decoding algorithm gives a significant degradation in performance.

4.2.4 The effect of the constraint lengths of the component codes.

The effect of increasing the constraint length of the component codes used in proposed turbo coder is shown in Fig.12. For the constraint length four turbo code we again used the optimum minimum free distance generator polynomials for the component codes (13 and 15 in decimal representations). The resulting K=4 turbo code gives an improvement of about 0.2dB at a BER of $2x10^{-5}$ over the K=3 curve.

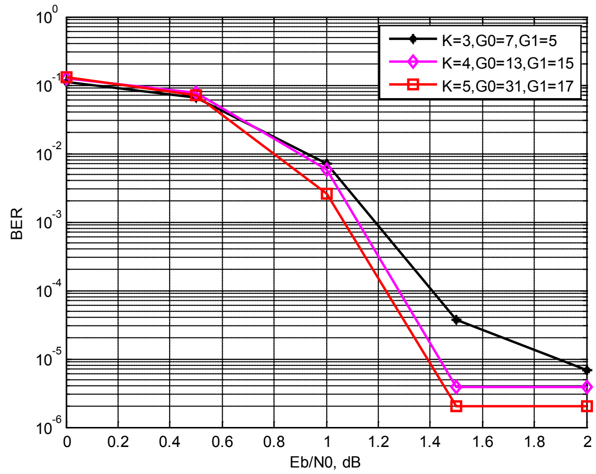

Fig.12 Effect of constraint length on BER performance of M-MLMAP Turbo decoding algorithm.

For the constraint length 5 turbo code we used the decimal generator polynomials 31 and 17, which were the polynomials used by Berrou et al.[3] in the original paper on turbo coding. It can be seen from Fig. 12 that increasing the constraint length of the M-MLMAP based turbo decoder does improve its performance, with the K=4 code performing about 0.2dB better than the K=3 code at a BER of $2x10^{-5}$, and the K=5 code giving a further improvement of about 0.05dB. However, these improvements are provided at the cost of approximately doubling or quadrupling the decoding complexity. Therefore, we have used component codes with a constraint length of 3 (G0=7, G1=5 code) in this paper.

4.2.5 The effect of the code rates

Again, in a turbo encoder two or more component encoders are used to generate parity information from an input data sequence. We have used two RSC component encoders, and this is the arrangement most commonly used for turbo codes with rates below two-thirds. Typically, in order to give a half-rate (rate 1/2) code, half the parity bits from each component encoder are punctured. This was the arrangement used in their original paper by Berrou et al. on turbo codes [1]. However, it is of course possible to omit the puncturing and transmit all the parity information from both component encoders, which gives a one-third rate (rate 1/3) code.

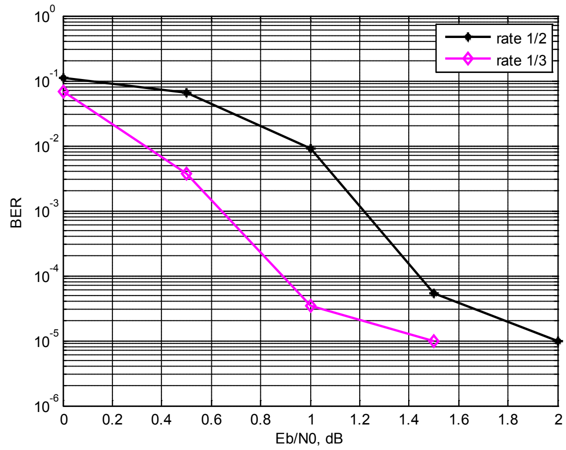

Fig.13 Effect of code rates on BER performance of M-MLMAP Turbo decoding algorithm.

The performance of such a code, compared to the corresponding half-rate code, is shown in Fig. 13. In this figure, the encoders use the same parameters as were described above for Fig. 12. It can be seen that transmitting all the parity information gives a gain of about 0.6 dB, in terms of E_b/N_0, at a BER of $2x10^{-5}$. This corresponds to a gain of about 2.4 dB in terms of channel SNR.

5 Conclusion

Thus on optimizing the scaling factor S_1 in Max-Log-MAP algorithm, improvement in performance is achieved. The proposed Modified Max-Log-MAP algorithm not only reduces the BER but also the complexity, which is the main design criterion for Turbo codes. In AWGN channel, the proposed M-MLMAP algorithm achieves performance closer to optimal Log-MAP and better performance than MLMAP with a gain of 0.3dB at BER of $3x10^{-5}$. The performance in fading channel is almost identical to that in AWGN channel with M-MLMAP showing a gain of 0.75dB at BER of $2x10^{-5}$, which proves the robustness of the proposed algorithm. There exists a relational dependence between scaling factor S_1 and E_b/N_0. The analytical expression provides simplification of the selection of the best scaling factor for the received E_b/N_0.

Finally, to gauge the expected coding performance of the proposed Modified Max-Log-MAP algorithm a range of performance results using a variety of CODEC parameters like decoding iterations, frame lengths, constraint lengths, generator polynomials and code rates is also provided.

References:

[1] Barbulescu, A. S., Pietrobon, S. S., Interleaver Design for Turbo Codes, *Electronics Letters*, Vol. 30, No. 25, 1994, pp. 2107-2108.

[2] C Chaikalis, J M Noras, Implementation of an improved reconfigurable SOVA/log-MAP turbo decoder in 3GPP, *Proc. IEE 3G2002, London, UK*, 2002, pp.146-150.

[3] C.Berrou, A.Glaviex, P.Thitimajshima, Near Shannon limit error-correction coding Turbo codes, *Proc. Int. Conf. Communication*, 1993, pp. 1064-1070.

[4] CCSDS, Recommendation for space data system standards, TM synchronization and channel coding, 131.0-B-1, Blue Book, 2003.

[5] Colavolpe G., Ferrari G., Raheli R., Extrinsic information in iterative decoding: a unified view, *IEEE Trans. Commun.*, Vol. 49, 2001, pp. 2088–2094.

[6] DVB, Interaction channel for satellite distribution systems, ETSI EN 301 790, v. 1.2.2, 2000.

[7] Gnanasekaran, T; Aarthi, V and Pradeep, NS., Performance enhancement of modified turbo decoding algorithms for mobile WiMAX, *Australian Journal of Electrical & Electronics Engineering*, Vol. 9, No. 2, 2012, pp. 153-163.

[8] Gnanasekaran, T., Duraiswamy, K., Application of Scaling factors for MAP and SOVA for Robust Performance in Forward Error Correction, *International*

Journal of Recent Trends in Engineering, Vol. 1, No. 3, 2009, pp. 47-51.

[9] Hagenauer, J. Hoeher, P., A Viterbi Algorithm with Soft-Decision Outputs and its Applications, *Proceedings of GLOBECOM'89*, 1989, pp. 1680-1686.

[10] Hagenauer, J., Papke, L., Decoding "Turbo" Codes with the Soft Output Viterbi Algorithm (SOVA), *Proceedings of International Symposium on Information Theory*, *Trondheim, Norway*, 1994, pp. 164-168.

[11] Hagenauer, J., Robertson, P., Papke, L., Iterative ("Turbo") Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms, *Proceedings of ITG*, 1994, pp. 21-29.

[12] H. Claussen, H.R. Karimi, and B. Mulgrew, Improved max-log map turbo decoding using maximum mutual information combining, *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, 2003, pp. 424 -428.

[13] J Vogt and A Finger, Improving the max-log-MAP turbo decoder, *IEE Electronics Letters*, Vol. 36, No. 23, 2000, pp. 1937-1939.

[14] Jonson P.Woodard, Lajos Hanzo, Comparative Study of Turbo Decoding Techniques: An Overview, *IEEE Transactions on vehicular technology*, Vol.49, No.6, 2000, pp. 2208-2233.

[15] P Robertson, E Villebrun, P Hoher, A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, *Proc. IEEE ICC '95, Seattle, USA*, 1995, pp. 1009-1013.

[16] Papke L., Robertson P., Villebrun E., Improved decoding with SOVA in parallel concatenated (Turbo-code) scheme, *Proc. IEEE Int. Conf. on Commun. (ICC), Dallas, USA*, 1996, pp. 102–106.

[17] Shannon, C. E., A Mathematical Theory of Communication, *The Bell System Technical Journal*, Vol. 27, 1948, pp. 379–423.

[18] Third Generation Partnership Project (3GPP), Technical Specification 25.212: Multiplexing and Channel Coding (Frequency Division Duplex Mode), ver. 3.0.0, 1999.