Effect of Constraint Length and Code Rate on the Performance of Enhanced Turbo Codes in AWGN and Rayleigh Fading Channel

T.GNANASEKARAN and V.AARTHI
Department of Electronics and Communication Engineering
Bannari Amman Institute of Technology
Sathyamangalam, Erode District, TamilNadu,
India.
t.gnanasekaran@gmail.com, aarthivellingiri@gmail.com

Abstract: - Turbo coding (TC) has been adopted as a channel coding scheme for several 3G mobile systems, in particular 3GPP (Third Generation Partnership Project) and in upcoming 4G standards for high data rates**.** Turbo decoder uses Maximum A posteriori Probability (MAP), or Soft Output Viterbi Algorithm (SOVA) because it produces error correction near to Shannon's limit. A simple but effective technique to improve the performance of the decoding algorithms is to scale the extrinsic information exchanged between two decoders. Modified Log MAP (MMAP) and Modified SOVA (MSOVA) algorithms are achieved by fixing an arbitrary value of scaling factor for inner decoder (S_2) and an optimized value for the outer decoder (S_1). We proposed to enhance the performance of MMAP and MSOVA by optimizing both the scaling factors S_1 and S_2, thus achieving low bit error rate (BER). This paper investigates the effects of constraint length and code rate on the performance of the enhanced Turbo codes. A comprehensive analysis of the algorithms considering different channel conditions and iterations are also presented.

Key-Words: - Constraint Length, Extrinsic information, MAP, Scaling Factor, SOVA, 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 [1]. TC has shown the best Forward Error Correction (FEC) performance till date. They are revolutionary in the sense that they allow reliable data transmission within a half decibel of the Shannon Limit. A massive amount of research effort has been performed to facilitate the efficiency of TC. Thus TC have been incorporated into many standards used by the NASA Consultative Committee for Space Data Systems (CCSDS) [2], Digital Video Broadcasting (DVB) [3], both Third Generation Partnership Project (3GPP) [4] standards for IMT-2000, Wideband CDMA which requires throughputs from 2 Mb/s to several 100 Mb/s, in 4G and WIMAX.

Two iterative decoding algorithms, Soft Output Viterbi Algorithm [5], [6], [7] and Maximum A posteriori Probability [6], [8] algorithm require complex decoding operations over several iteration cycles. The relative complexity of the decoding algorithms depends on the constraint length. Hence for real time implementation of TC, reducing the decoder complexity while preserving BER performance is an important design consideration.

To overcome the above drawbacks, we present an analysis on the effect of constraint length and code rate on the performance of the proposed decoding algorithms.

2 Turbo Encoder

A basic turbo encoder is a recursive systematic encoder that employs two convolutional encoders in parallel, where the second encoder is preceded by an interleaver and is shown in Fig.1. The interleaver is usually selected to be a pseudo random interleaver that reorders the bits in the information sequence before being fed to the second encoder. The use of

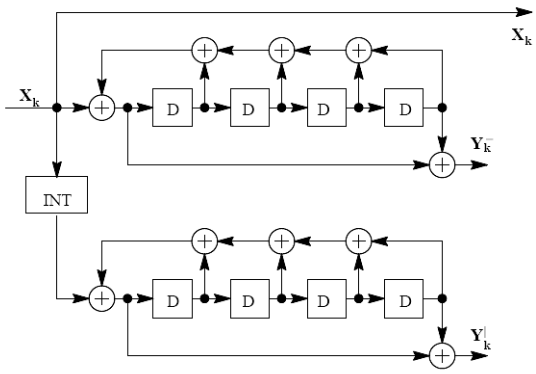

Fig.1 Turbo Encoder

interleaver in conjunction with two encoders result in code words that have relatively few nearest neighbors. This makes the code word relatively sparse. Hence the coding gain achieved by a turbo code is due to the reduction in the number of nearest neighboring code words that result from interleaving. It is observed that the nominal rate at the output of the turbo encoder is 1/3.This increases the redundant bits and hence the error probability decreases.

The RSC component codes shown in the Fig.1 are k=5 (constraint length) code with generator polynomials G_0=31 and G_1=17. These generator polynomials are optimum in terms of maximizing the minimum free distance of the component codes.

3 Turbo Decoder

In a typical Turbo decoding system shown in Fig.2, two decoders (DC 1 and DC 2) 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 [12]. 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 the 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

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

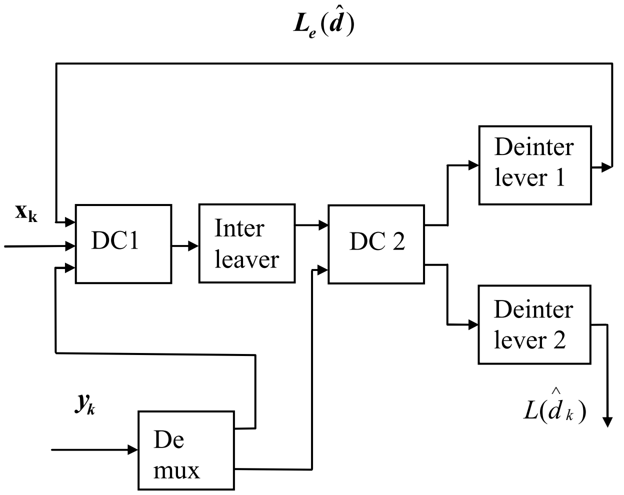

Fig.2 Turbo Decoder

LLR can be decomposed into 3 independent terms,

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

where $L_{apri}(d_k)$ is the a-priori information of d_k, $L_c(d_k)$ is the channel measurement, $L_e(d_k)$ is the extrinsic information. Extrinsic information [13] 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 MAP and SOVA.

3.1 MAP Algorithm

The MAP algorithm [8], [14] 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

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

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

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

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

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

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, and, y_k^s and y_k^p are the received noisy systematic data bit and parity bit respectively.

3.2 Log MAP Algorithm

To avoid complex mathematical calculations of MAP decoding, computations can be performed in the logarithmic domain [8]. 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 (3)-(6) become

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

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

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

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

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

$$+ \log\left(P(S_k|S_{k-1})\right) + K \quad (11)$$

where K is a constant.

3.3 SOVA Algorithm

In this section [10], we explain a variation of the Viterbi algorithm, referred to as the SOVA. SOVA has two modifications over the Viterbi algorithms. The path metrics used are modified to take account of a-priori information when selecting the maximal likelihood path through the trellis. Another modification is made so that it provides a soft output in the form of the a-posteriori LLR $L(u_k|y)$ for each decoded bit. S_k^s gives the states along the surviving path at state $S_k = s$ in the trellis. The probability that this is the correct path through the trellis is given by

$$P\left(s_k^s \middle| y_{j\le k}\right) = \frac{P\left(s_k^s \wedge y_{j\le k}\right)}{P\left(y_{j\le k}\right)} \quad (12)$$

The iterative decoding of Turbo codes uses the a-priori information from a component decoder. It is independent of the channel outputs used by that decoder. The extrinsic LLR $L_e(u_k)$ for the bit u_k uses all the available received parity bits and all the received systematic bits except the received values y_k^s associated with u_k. The systematic bits are also used by the other component decoder, which is the interleaved or deinterleaved version of $L_e(u_k)$ as its a-priori LLRs. The a-priori LLRs $L_e(u_k)$ are not truly independent from the channel outputs. The extrinsic LLR $L_e(u_k)$ is affected by the received systematic bit relatively close to the bit u_k. When LLR $L_e(u_k)$ is used as the a-priori LLR by the other component decoder, the iterative decoding provides good results. When calculating the LLR of the bit u_k , SOVA must take into account of the probability that the paths merging with the ML path from stage k to stage k+δ in the trellis were incorrectly discarded. This is done by considering the values of the metric difference $\Delta_i^{s_i}$ for all states s_i along the ML path from trellis stage *i=k* to *i*= $k+\delta$ The LLR can be approximated by

$$L(u_k|y) \approx u_k \min_{\substack{i=k\ldots k+\delta \\ u_k \ne u_k^i}} \Delta_i^{s_i} \quad (13)$$

where u_k is the value of the bit given by the ML path, and u_k^i is the value of this bit for the path which merged with the ML path and was discarded at trellis stage *i*. Thus the minimization in [8] is carried out only for those paths merging with the ML path which would have given a different value for the bit u_k if they had been selected as the survivor path. The path which merges with the ML path, but would have given the same value for u_k as the ML path, obviously do not affect the reliability of the decision u_k .

4 Enhanced Turbo Decoder

The SOVA and Log MAP algorithms suffer from two distortions: over optimistic soft outputs and correlation between the intrinsic and extrinsic information [15]. 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 $L_e(u_k)$ is possible with a common 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 [10]. 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 [11]. MMAP and MSOVA algorithms [9], [16] are achieved by fixing an arbitrary value for inner decoder (S_2) and an optimized value for the outer decoder (S_1). For Enhanced MAP (EMAP) and Enhanced SOVA (ESOVA), both S_1 and S_2 are optimized. Scaling factor S_2 depends on E_b/N_0 to give low BER and better performance than modified decoding algorithms. The proposed Turbo decoder with optimized scaling factors is shown in Fig.3. The algorithms are enhanced by multiplying the extrinsic information $L_e(\hat{d}_k)$ with the optimized scaling factors S_1 and S_2 before it is being fed back to the input and decoder 2 respectively, and are given by

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

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

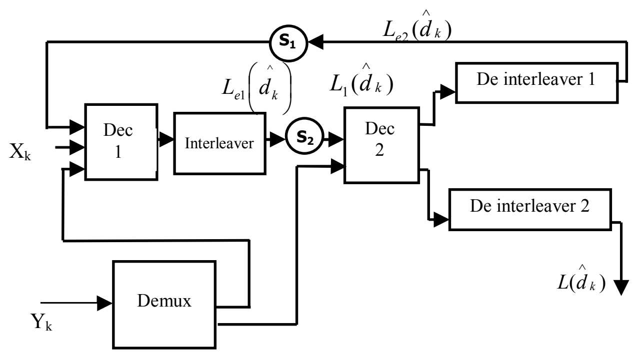

Fig.3 Turbo Decoder with Optimized Scaling Factors

5 Simulation Results and Discussion

Transmission of 1140 frames with a frame length of 2048 bits and random interleaver [17] is taken to show the effect of the scaling factors on the performance of TC and to analyze the outcome of various constraint lengths and code rates. The simulation parameters for finding the optimized scaling factors are,

Channel: AWGN
Modulation: Quadrature Phase shift Keying (QPSK)
Component encoder: Récursive convolution codes (RSC)
Interleaver: 2048 bit random interleaver
Iteration: 8
Frame limit: 1140

Though the scaling factors considered range from 0.1 to 0.99, in EMAP algorithm the range from 0.7 to 0.9 gives reduced BER and its performance is shown in Fig.4. Similarly for ESOVA the suitable range giving reduced BER is 0.7 to 0.99 as shown in Fig.5. A wide range of scaling factors, E_b/N_0 and the corresponding BER has been showed. The scaling factor giving the least BER for a particular E_b/N_0 is considered to be optimum.

Table 1 shows the optimized scaling factor (S_2) (giving the least BER), E_b/N_0 and the corresponding BER for EMAP and ESOVA algorithms. It is found that for E_b/N_0 greater than 1.0dB the optimized scaling factor for enhanced Log MAP algorithm is constant and is found to be 0.85. For ESOVA, S_2 is found to vary with E_b/N_0 and is adaptive with respect to E_b/N_0. The BER for EMAP algorithm at 3dB is 1.6211×10^{-6}. Similarly at 3dB, the BER for enhanced SOVA is comparatively reduced and its value is 9.8144×10^{-7}. Thus ESOVA gives reduced BER and better performance than EMAP algorithm.

Table 1
Optimized Scaling Factor (S_2) and BER for varying E_b/N_0

Eb/No	EMAP		ESOVA	
dB	Optimized Scaling Factor (S_2)	BER	Optimized Scaling Factor (S_2)	BER
0	0.89	1.0800×10^{-1}	0.71	1.2528×10^{-1}
0.5	0.89	5.9358×10^{-2}	0.71	7.5956×10^{-2}
1	0.88	7.4698×10^{-3}	0.71	2.0261×10^{-2}
1.5	0.85	7.5571×10^{-5}	0.99	9.0685×10^{-4}
2	0.85	5.8887×10^{-6}	0.85	3.2388×10^{-5}
2.5	0.85	1.9629×10^{-6}	0.92	3.9258×10^{-6}
3	0.85	1.9629×10^{-6}	0.86	9.8144×10^{-7}
3.5	0.85	1.6211×10^{-6}	0.99	9.8144×10^{-7}
4	0.85	1.6211×10^{-6}	0.95	9.8144×10^{-7}

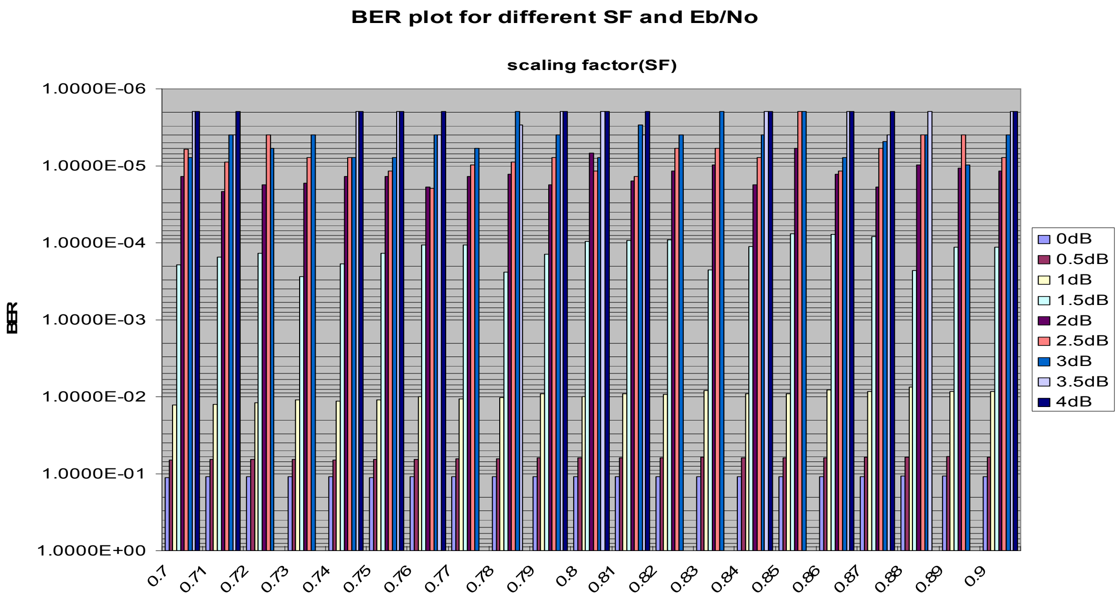

Fig.4 BER plot of various Scaling Factors and E_b/N_0 with code generator (7,5), punctured for Log MAP algorithm

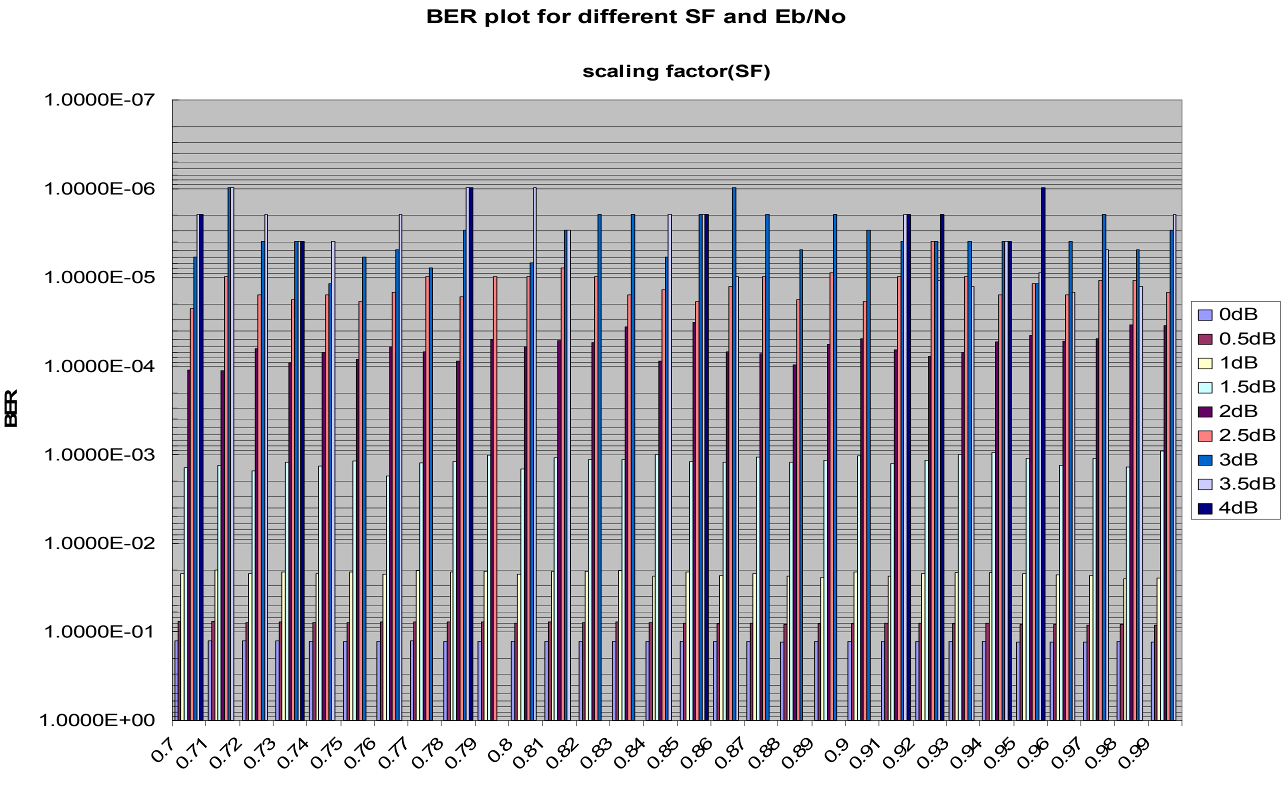

Fig.5 BER plot of various Scaling Factors and E_b/N_0 with code generator (7,5), punctured for SOVA algorithm

Fig.6 shows the performance of Enhanced Log MAP algorithm with the scaling factors S_1=0.9 and S_2=0.85 is giving better results than the Modified Log MAP algorithm with scaling factors S_1=0.9 and S_2=0.755. The MMAP and EMAP algorithms are also compared with the standard algorithm without any scaling factor, at E_b/N_0 of 2.5dB. This graph gives evidence on the improved performance of EMAP algorithm in terms of BER. It is noted that for iteration 4, the BER of MMAP [16] and EMAP algorithms are 0.5×10^{-5} and 1×10^{-6} respectively.

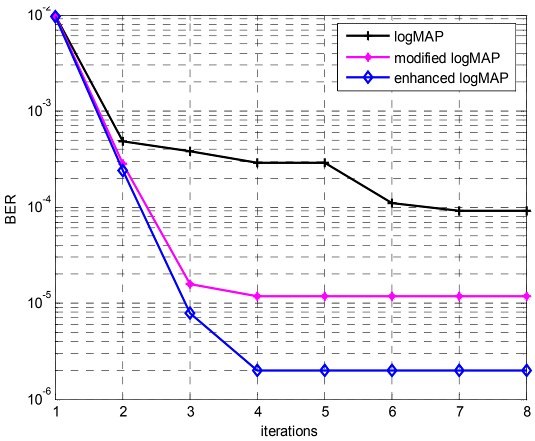

Fig.6 BER plot of Log MAP, MMAP and EMAP decoding algorithms for different iterations, for 2.5dB in AWGN channel.

It is also observed from Fig.6 that the performance remains constant from iteration 4. It is revealed for Log MAP algorithm, the efficient BER has been achieved by 4 iterations. Thus in the proposed EMAP algorithm, complexity has been reduced by 50% compared to Log MAP algorithm and the BER has been reduced by the order of 10^{-1} compared to MMAP algorithm. The main design criterion for any decoding algorithm is to reduce the BER and complexity, which is achieved by the proposed EMAP algorithm.

Fig.7 shows the performance of Enhanced SOVA algorithm with the scaling factors S_1=0.56 and S_2=0.92 is giving better results comparing with the Modified SOVA [10] and SOVA algorithm, at E_b/N_0 of 2.5dB. At the end of 8^{th} iteration in the decoding part the difference in BER for MSOVA and ESOVA is about 0.8×10^{-1}. It is noted that as the iteration increases, the performance of ESOVA improves. Since SOVA is less complex compared to Log MAP, the number iterations can be increased to 6 without degradation in performance.

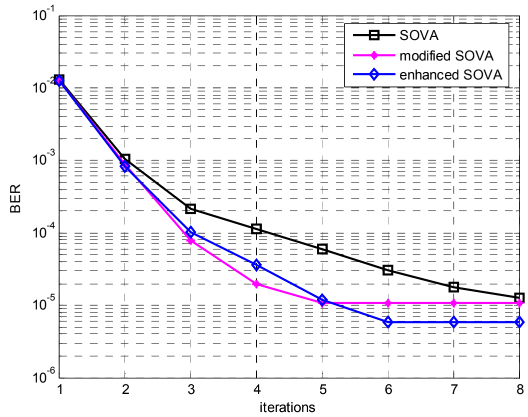

Fig.7 BER plot of SOVA, MSOVA and ESOVA decoding algorithms for different iterations, for 2.5dB in AWGN channel.

Table 2 gives the summary of the number of iterations required, BER and the percentage of reduction in complexity for each decoding algorithms. Compared to Log MAP algorithm, the complexity of MMAP and EMAP algorithms are reduced. EMAP also shows BER improvement than MMAP but with the similar complexity. Though the complexity reduction of ESOVA is higher than MSOVA, the BER of ESOVA is greatly reduced.

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	$9.0897\text{x}10^{-5}$
MMAP	4	50	$1.7777\text{x}10^{-5}$
EMAP	4	50	$1.9629\text{x}10^{-6}$
SOVA	8	0	$1.2759\text{x}10^{-5}$
MSOVA	5	37.5	$1.0796\text{x}10^{-5}$
ESOVA	6	25	$5.8887\text{x}10^{-6}$

The summary of scaling factors for various decoding algorithms is shown in Table 3.

Table 3
Scaling Factors for Various Decoding Algorithms

Decoding Algorithms	Scaling factor	
	Decoder1(S_1)	Decoder2(S_2)
MMAP	0.9 *	0.755
EMAP	0.9 *	0.85 *
MSOVA	0.56 *	0.98
ESOVA	0.56 *	Adaptive *

* - Optimized Scaling Factors

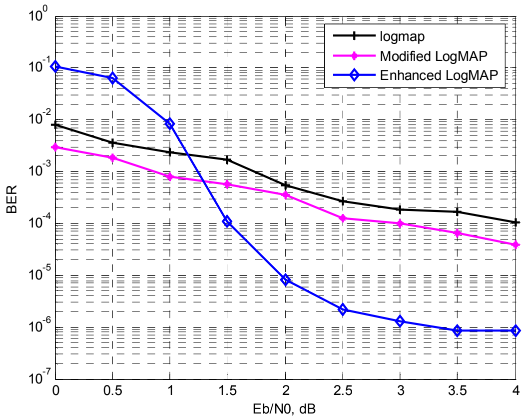

Fig.8 Performance of Log MAP, MMAP and EMAP in AWGN channel.

Analyses are carried out to show the performance of the decoding algorithms in AWGN and Rayleigh fading channels, with QPSK modulation. Fig.8 shows the performance of Log MAP, MMAP and EMAP in AWGN channel. It is found that the BER of EMAP algorithm is 8×10^{-7}.

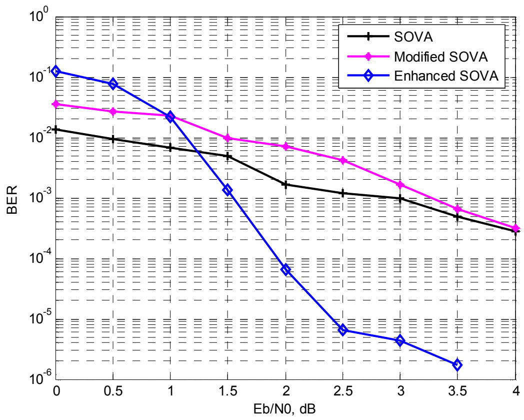

Fig.9 Performance of SOVA, MSOVA and ESOVA in AWGN channel.

At E_b/N_0 of 1.5dB and above EMAP algorithm is better. But for lower E_b/N_0 values (<1.5dB), MMAP algorithm is better. So the proposed algorithm yields the lowest BER.

Fig.9 gives the performance of SOVA, MSOVA and ESOVA in AWGN channel. It is found that MSOVA fails to improve in AWGN channel, whereas ESOVA does. At E_b/N_0 of 1.5dB and above ESOVA algorithm is better. But for lower E_b/N_0 values (<1.5dB), SOVA algorithm is better. At E_b/N_0 of 3.5dB, BER of ESOVA is 1.7×10^{-6} which shows two fold improvement in performance compared to SOVA and MSOVA.

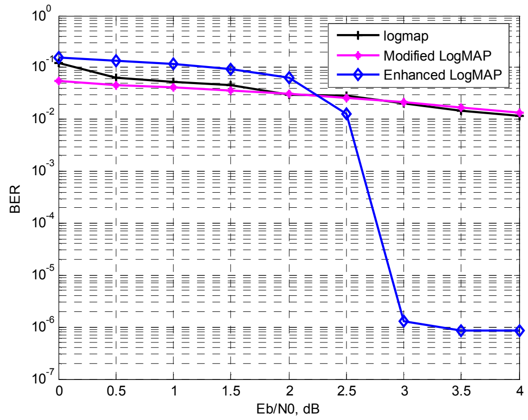

Fig.10 Performance of Log MAP, MMAP and EMAP in Rayleigh fading channel.

Similar analysis is done for the Rayleigh Fading channel and is shown in Fig.10 and Fig.11. The performance of EMAP algorithm in fading channel is almost identical to that in AWGN channel for E_b/N_0 greater than 2.5dB, which validates the robustness of the EMAP algorithm and is shown in Fig.10.

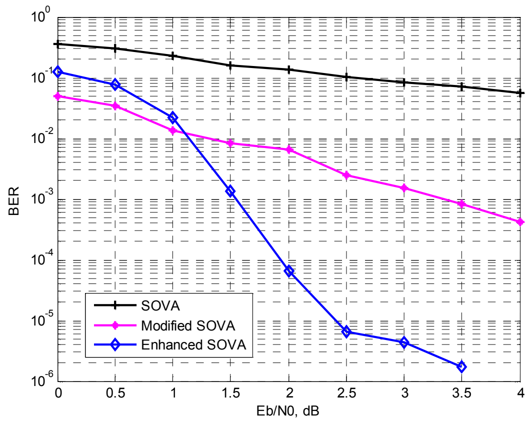

Fig.11 Performance of SOVA, MSOVA and ESOVA in Rayleigh fading channel.

The performance of the proposed ESOVA algorithm in fading channel is shown in Fig.11. It is almost identical to the performance of AWGN channel for all values of E_b/N_0, which validates the robustness of the ESOVA algorithm. On scaling the extrinsic information with optimized scaling factors S_1 and S_2, the SOVA algorithm is optimized. Thus it is observed that no further enhancement to the algorithm is required. So in both the channel conditions, the proposed Turbo decoding algorithms gave improved performance.

The performance of enhanced decoding algorithms are analyzed considering three code generators (7,5), (15,13) and (31,17) with constraint length k=3, 4 and 5 respectively and two code rates.

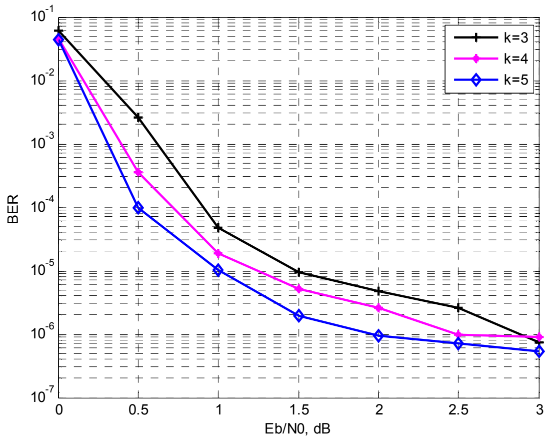

Fig.12 Effect of constraint length on the EMAP algorithm in AWGN channel

The effect of constraint length on the performance of EMAP algorithm in AWGN and fading channels are shown in Fig.12 and Fig.13 respectively. For EMAP algorithm in AWGN channel, on increasing the constraint length from k=3 to k=4, the performance of Turbo code improves by 0.5dB at BER of $4\text{x}10^{-6}$ over the curve.

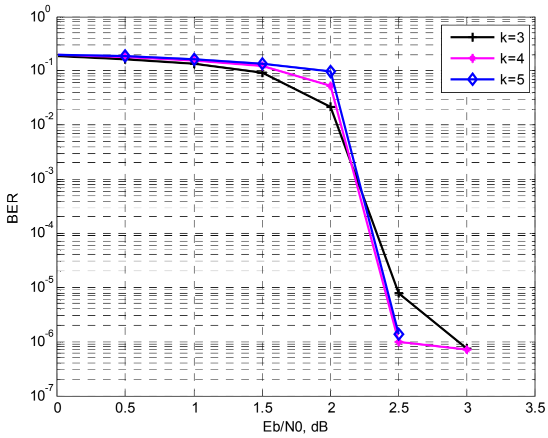

Fig.13 Effect of constraint length on the EMAP algorithm in fading channel

Similarly k=5 code gives further improvement of 0.5dB at BER of $9\text{x}10^{-7}$ than k=4 code and is shown in Fig.12. But Rayleigh fading channel shows no performance improvement on increasing the constraint length as shown in Fig.13.

The effect of increasing the constraint length of the component codes used in ESOVA is shown in Fig.14 and Fig.15. For the constraint length four Turbo code we used the optimum minimum free distance generator polynomials [8] for the component codes 15 and 13. The resulting turbo code gives an improvement of about 0.5 dB at a BER of 10^{-7} over the curve. For the constraint length five Turbo code we used the generator polynomials 31 and 17, which were the polynomials used by Berrou *et al.* [1] in the original paper on TC.

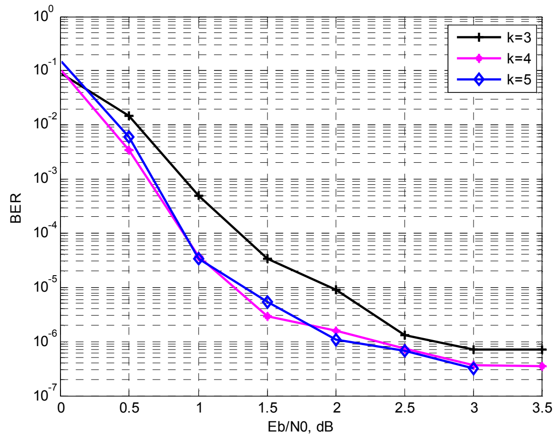

Fig.14 Effect of constraint length on the ESOVA algorithm in AWGN channel

It can be seen from Fig.14 and Fig.15 that increasing the constraint length of the turbo code does improve its performance, with the k=4 code performing about 0.5 dB better than the k=3 code at a BER of 10^{-6}, and the k=5 code giving a further improvement of about 0.2 dB.

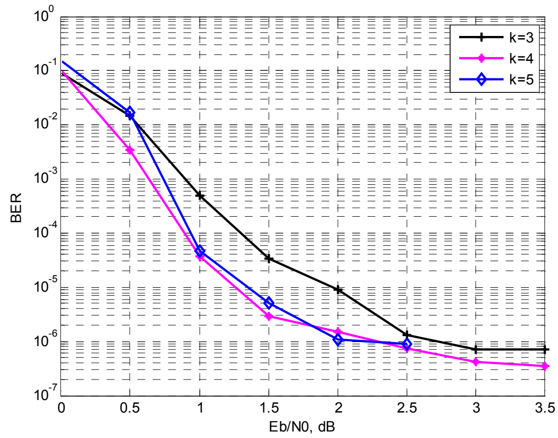

Fig.15 Effect of constraint length on the ESOVA algorithm in fading channel

However, these improvements are provided at the cost of approximately doubling or quadrupling the decoding complexity. So constraint length 4 Turbo code is considered as the suitable choice giving reduced BER at comparatively reduced complexity. It is also found from Fig.14 and Fig.15 that the proposed ESOVA algorithm in Rayleigh fading channel performs equally well as that of AWGN channel for various constraint lengths.

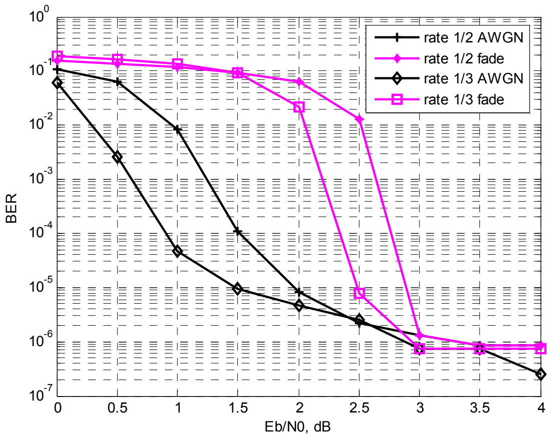

Fig.16 Code rate and channel comparison for EMAP with k=4

The Fig.16 and Fig.17 shows the effect of code rates 1/2(punctured) and 1/3(unpunctured) on the performance of EMAP and ESOVA algorithm, in AWGN and Rayleigh fading channels for constraint length four TC.

The performance of AWGN channel for EMAP algorithm is superior to the fading channel as shown in Fig.16. Rate 1/3 Turbo code performs about 0.3dB better than rate 1/2 Turbo code at a BER of $8x10^{-6}$ over the curve, for both the channel conditions. This is due to the increased redundancy of code rate 1/3 which gives improved reliability and hence reduces the BER.

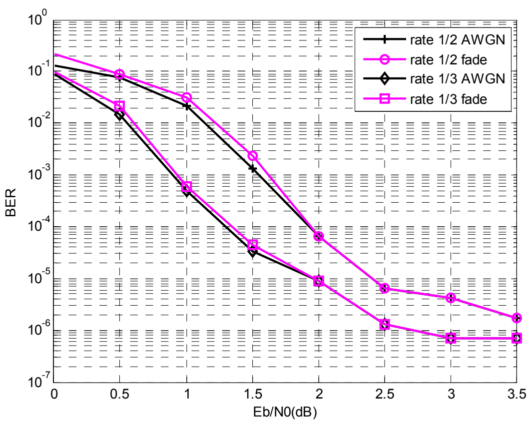

Fig.17 Code rate and channel comparison for ESOVA with k=4

Fig.17 shows the code rate and channel comparison for ESOVA with constraint length four TC. Rate 1/3 TC shows improvement of about 0.4dB than rate 1/2 at a BER of $9x10^{-6}$ over the curve. It is also found that the performance of ESOVA algorithm is almost similar in both AWGN and Rayleigh fading channels.

So, on using the ESOVA algorithm optimized performance for fading channel is achieved. The proposed ESOVA is highly robust for practical channel conditions giving lowest possible BER identical to that of the theoretical AWGN channel.

6 Conclusion

Thus optimizing both the scaling factors in Log MAP and SOVA algorithm lead to the improvement in performance of the decoding algorithms in AWGN and Rayleigh fading channels. On increasing the constraint length, the performance of proposed Turbo code improved for EMAP algorithm in AWGN channel. But Rayleigh fading channel shows no performance improvement. It is found that for ESOVA algorithm, increasing the constraint length of the TC does improve its performance, with the k=4 code performing better than the k=3 code at a BER of 10^{-6}, and the k=5 code giving further improvement in performance, in both AWGN and fading channels. Rate 1/3 TC outperforms rate 1/2 TC for both EMAP and ESOVA algorithms giving improved reliability and reduced BER.

References:

[1] C.Berrou, A.Glaviex and P.Thitimajshima, Near Shannon limit error-correction coding-Turbo codes, *Proceedings of International Conference on Communication*, May 1993, pp. 1064-1070.

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

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

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

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

[6] Hagenauer J, Robertson P and Papke L, Iterative ("Turbo") Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms, *Proceeding of ITG*, October 1994, pp. 21-29.
[7] Papke L, Robertson P and Villebrun E, Improved decoding with SOVA in parallel concatenated (Turbo-code) scheme, *Proc. IEEE Int. Conf. on Commun. (ICC), Dallas, USA,* July 1996, pp. 102–106.
[8] Jonson P.Woodard and Lajos Hanzo, Comparative Study of Turbo Decoding Techniques: An Overview, *IEEE Transactions on vehicular technology*, Vol.49, No.6, November 2000, pp. 2208-2233.
[9] Gnanasekaran T and 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.
[10] T.Gnanasekaran and K. Duraiswamy, Modified Soft Output Viterbi Error Correcting Algorithm, *International Journal of Computer Science and Network Security*, Vol.8, No.9, September 2008, pp. 161-165.
[11] Shannon C. E, A Mathematical Theory of Communication, *The Bell System Technical Journal*, Vol. 27, October 1948, pp. 379–423.
[12] Hagenauer J and Hoeher P, A Viterbi Algorithm with Soft-Decision Outputs and its Applications, *Proceedings of GLOBECOM'89*, 1989, pp. 1680-1686.
[13] Colavolpe G, Ferrari G and Raheli R, Extrinsic information in iterative decoding: a unified view, *IEEE Trans. Commun.,* Vol. 49, 2001, pp. 2088–2094.
[14] P Robertson, E Villebrun and P Hoher, A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, *Proc. IEEE ICC '95, Seattle, USA,* June 1995, pp. 1009-1013.
[15] C Chaikalis, M Salimi-Khaligh, N Panayotopoulos and J M Noras, Reconfiguration between soft output viterbi and log maximum a posteriori decoding algorithms, *Proc. IEE 3G2002, London, UK*, March 2000, pp.316-320.
[16] Gnanasekaran T, Duraiswamy K, Aarthi V and Brindha C, Performance of Turbo Codes Using Code Generator [15, 13] and [31, 17] in AWGN and Fading Channel, *International Journal of Recent Trends in Engineering (Electrical & Electronics),* Vol. 2, No. 6, November 2009, pp 20-23.
[17] Barbulescu A. S and Pietrobon S. S, Interleaver Design for Turbo Codes. *Electronics Letters*, Vol. 30, No. 25, December 1994, pp. 2107-2108.

Authors Biography:

Dr.T.Gnanasekaran, is graduated from University of Madras in 1989, completed his post graduate and PhD from Anna University Chennai, India. He is presently with Bannari Amman Institute of Technology, Sathyamangalam, TamilNadu, India, as Associate Professor. He has published 12 national and international journals. His areas of interests are Error Control Coding, Advanced Digital Communication, WiMAX and LTE.

V.Aarthi completed her under graduation from Anna University Chennai and post graduation from Anna University Coimbatore, India. She is currently with Velalar College of Engineering and Technology, Erode as Assistant Professor. She has presented three papers in national level conferences and three international conferences. She has also published two international journals.