A COMPARATIVE STUDY ON THE MODIFIED MAX-LOG-MAP TURBO DECODING BY EXTRINSIC INFORMATION SCALING

Mustafa Taskaldiran[*], Richard C.S. Morling[*], and Izzet Kale[*+]
[*]*Applied DSP and VLSI Research Group, Department of Electronic Systems, University of Westminster, London, W1W 6UW, United Kingdom*
[+]*Applied DSP and VLSI Research Centre, Eastern Mediterranean University, Gazimagusa, N. Cyprus*
{m.taskaldiran, morling, kalei}@wmin.ac.uk

Abstract

A simple but effective technique to improve the performance of the Max-Log-MAP algorithm is to scale the extrinsic information exchanged between two MAP decoders. A comprehensive analysis of the selection of the scaling factors according to channel conditions and decoding iterations is presented in this paper. Choosing a constant scaling factor for all SNRs and iterations is compared with the best scaling factor selection for changing channel conditions and decoding iterations. It is observed that a constant scaling factor for all channel conditions and decoding iterations is the best solution and provides a 0.2-0.4 dB gain over the standard Max-Log-MAP algorithm. Therefore, a constant scaling factor should be chosen for the best compromise.

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]. 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 Limit. At first, the extraordinary performance of turbo codes encountered some doubts by the communication community. However, their performance has been verified by many researchers in a short time after the emergence of turbo codes [2, 3]. 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), Digital Video Broadcasting (DVB), both Third Generation Partnership Project (3GPP) standards for IMT-2000, and Wideband CDMA which requires throughputs from 2 Mb/s to several 100 Mb/s.

The iterative nature of turbo-decoding algorithms increases their complexity compare to conventional FEC decoding algorithms. Two iterative decoding algorithms, Soft-Output-Viterbi Algorithm (SOVA) and Maximum A posteriori Probability (MAP) Algorithm require complex decoding operations over several iteration cycles. So, for real-time implementation of turbo codes, reducing the decoder complexity while preserving bit-error-rate (BER) performance is an important design consideration.

In this paper, a modification to the Max-Log-MAP algorithm is investigated. This modification is to scale the extrinsic information exchange between the constituent decoders. Section 2 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 3. Section 4 presents simulation results and compares the performance of different methods to choose the best scaling factor.

2. Turbo Decoder

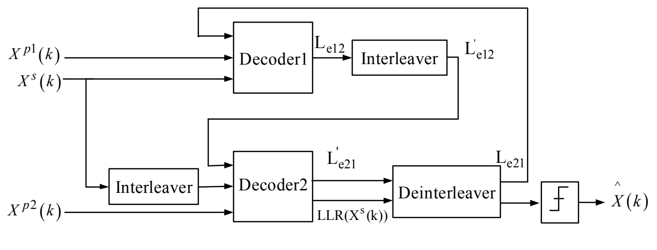

Figure 1. Iterative Turbo Decoding

In a typical turbo decoding system (see Figure 1), 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-

1-4244-0697-8/07/.00 ©2007 IEEE.

Output (SISO) decoder [4]. 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" [5].

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

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

LLR can be decomposed into 3 independent terms, as

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

where $L_{apri}(d_k)$ is the a-priori information of d_k, $L_c(x_k)$ is the channel-measurement, and $L_e(d_k)$ is the extrinsic information. Extrinsic information from one decoder becomes the a-priori information for the other decoder at the next decoding stage. L_{e12} and L_{e21} in Figure 1 represent the extrinsic information from decoder1 to decoder2 and decoder2 to decoder1 respectively. LLRs can be calculated by two different SISO algorithms SOVA and MAP Algorithm.

2.1. The MAP 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

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

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

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

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

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

Branch metrics are calculated for each possible trellis transition as

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

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.

2.2. The 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

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

The last term in $\max^*(.)$ operation can easily be calculated by using a look-up table (LUT).

So equations (3)-(6) become

$$\begin{aligned} L(d_k) = {} & \max^*_{(S_{k-1},S_k,1)}\left(\bar{\gamma}_1(S_{k-1}, S_k) + \bar{\alpha}_{k-1}(S_{k-1}) + \bar{\beta}_k(S_k)\right) \\ & - \max^*_{(S_{k-1},S_k,0)}\left(\bar{\gamma}_0(S_{k-1}, S_k) + \bar{\alpha}_{k-1}(S_{k-1}) + \bar{\beta}_k(S_k)\right) \end{aligned} \tag{8}$$

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

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

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

where K is a constant.

2.3. The Max-Log-MAP Algorithm

The correction function $f_c = \ln(1 + e^{-|y-x|})$ 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

$$\ln(e^x + e^y) \approx \max(x,y) \tag{12}$$

Table 1. Scaling factors for different SNRs and iterations for decoder 1 (D1) and decoder 2 (D2) (R=1/3, interleaver length=1024, generator polynomial (13, 15) $_{oct}$)

Iterations	1		2		3		4		5		6	
E_b/N_o	D1	D2	D1	D2	D1	D2	D1	D2	D1	D2	D1	D2
0	0*	0.5	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.7
0.25	0	0.6	0.6	0.6	0.6	0.6	0.6	0.7	0.6	0.7	0.7	0.7
0.50	0	0.6	0.6	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.7
0.75	0	0.6	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.7	0.7	0.8
1	0	0.6	0.7	0.8	0.8	0.7	0.8	0.6	0.5	0.7	0.4	0.8
1.25	0	0.7	0.9	0.7	0.5	0.9	0.5	0.8	0.4	0.8	0.3	1
1.5	0	0.7	0.7	0.8	0.5	1	0.5	0.9	0.4	0.9	0.4	0.8

*No extrinsic information from decoder2 to decoder1 for the 1st iteration.

at the expense of some performance degradation.

This simplification eliminates the need for a LUT required to find the corresponding correction factor in the $\max^*(.)$ operation. The performance degradation due to this simplification is about 0.5dB compared to the Log-MAP algorithm [6].

3. Extrinsic Information Scaling

It has been proposed to scale the extrinsic information exchanged between the constituent decoders [7-9]. With this modification equation (11) for branch metric calculations can be rewritten as

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

The only modification is the scaling factor s_d where $d=1,2$ for decoder1 and decoder2 respectively.

Extrinsic information scaling has been proposed to compensate for the optimistic LLR calculations of SOVA [9]. A gain of 0.4 dB has been reported for a code of memory length 4 at BER of 10^{-4} [9]. Scaling factor modification has also been applied and tested on the Max-Log-MAP algorithm. Authors of [7] has reported 0.2-0.4dB gain over the standard algorithm for 3GPP standards. They used a constant scaling factor of 0.7. In [8], scaling factor optimization for Max-Log-MAP decoding is explained as mutual information combining which is the evolution of the information exchange between the two MAP decoders. The best scaling factors for each iteration were calculated for different SNRs by off-line computation. The performance difference between the modified Max-Log-MAP and Log-MAP was reported as 0.05 dB for UMTS-based turbo coding [8]. The performance improvement introduced by the scaling factor modification is explained as the correction of the accumulated bias due to maximum (max) operation in the Max-Log-MAP algorithm [8].

Table 2. Scaling factors for different SNRs (R=1/3, interleaver length=5114, generator polynomial (13, 15) $_{oct}$)

E_b/N_o(dB)	0	0.25	0.5	0.75	1	1.25	1.5
Dec1	0.6	0.6	0.7	0.7	0.9	0.5	0.4
Dec2	0.6	0.7	0.7	0.8	0.8	0.8	0.8

4. Simulation Results

A turbo code of rate R = 1/3, memory length $m=3$, generator polynomial (13, 15)$_{oct}$, and interleaver lengths of 1024 and 5114 have been simulated in Matlab to obtain the best scaling factors for different SNRs and decoding iterations. These scaling factors are obtained via simulations by choosing the scaling factors corresponding to the minimum BER. AWGN channel is assumed during simulations. Table 1 shows the best scaling factors for iterations 1 to 6 and SNR values of 0 dB to 1.5 dB. Table 2 shows the best scaling factors after 6 iterations only for different SNRs assuming a constant scaling factor for both decoders. The performance of the modified algorithm is compared with the standard Max-Log-MAP and Log-MAP algorithms. Figure 2 and 3 show the BER performances of the Log-MAP, the Max-Log-MAP, and the modified Max-Log-MAP with scaling factor 0.7 after 6 decoding iterations for interleaver lengths 5114 and 1024 respectively. A constant scaling factor (0.7) provides approximately 0.4 dB improvement over the standard Max-Log-MAP algorithm at a BER of 10^{-4}. The same result was presented in [7] in the case of IMT-2000 parameters for all interleaver lengths. Table 3 shows BER values at E_b/N_o=1dB, for an interleaver length of 1024.

From the simulation results, it is observed that changing scaling factors for different SNRs/iterations or just for SNRs doesn't improve the decoding performance. As it's shown in Table 3, BER values for 3 different methods of scaling factor modification are almost

Table 3. BER values for each iteration at E_b/N_o=1dB

(R=1/3, interleaver length=1024, generator polynomial $(13, 15)_{oct}$) *Sf: Scaling Factor

	BER				
Iteration	Log-MAP	Max- Log-MAP	Constant Sf*	Sf for SNRs	Sf for SNRs and Iterations
1	0.0396	0.0509	0.0467	0.0482	0.0462
2	0.0125	0.0301	0.0186	0.0239	0.0180
3	0.0034	0.0188	0.0067	0.0119	0.0064
4	0.0012	0.0132	0.0028	0.0068	0.0028
5	0.0006	0.0101	0.0016	0.0044	0.0016
6	0.0004	0.0087	0.0011	0.0034	0.0010

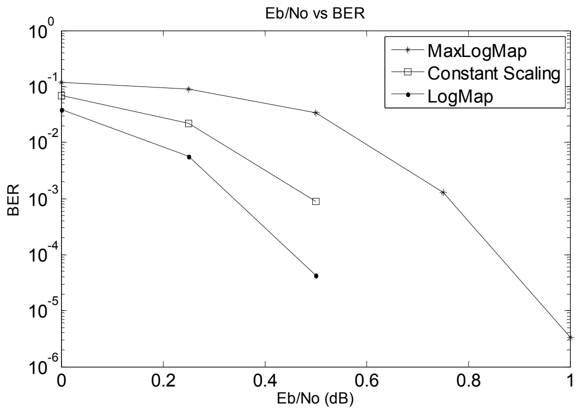

Figure 2. BER versus E_b/N_o for Log-MAP, Max-Log-MAP and modified Max-Log-MAP with scaling factor = 0.7 (interleaver length 5114, 6 iterations).

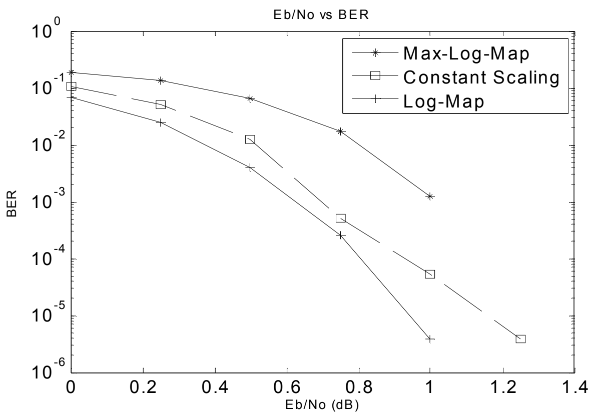

Figure 3. BER versus E_b/N_o for Log-MAP, Max-Log-MAP and modified Max-Log-MAP with scaling factor = 0.7 (interleaver length 1024, 6 iterations).

identical. These results have also been verified for different interleaver lengths. Although, changing scaling factors for SNRs (and decoding iterations) provides an improvement over the standard Max-Log-MAP algorithm, this improvement is observed to be equal to the improvement obtained by a constant choice of the scaling factor. The performance difference between the Log-MAP and the modified Max-Log-MAP is around 0.1 dB as observed from simulations.

5. Conclusion

The performance gap between the Max-Log-MAP and Log-MAP algorithms can be compensated by scaling the extrinsic information exchange between two constituent MAP decoders. This modification in the Max-Log-MAP algorithm can be implemented simply by multiplying the extrinsic information by a scaling factor. The modified Max-Log-MAP algorithm has been simulated by choosing this scaling factor as a constant as well as choosing the best scaling factors for different SNRs and decoding iterations. Simulation results show that there's almost no performance gain when we adaptively change the scaling factor with different channel conditions and for different decoding iterations against keeping the scaling factor constant. On the other hand, a proper choice of the scaling factor provides 0.4 dB improvement for the Max-Log-MAP algorithm. From our simulations, this optimum constant scaling factor is found to be 0.7.

6. References

[1] Berrou, C., A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC'93*, pp. 1064 - 1070.
[2] Benedetto, S. and G. Montorsi, "Performance evaluation of turbo-codes," *Electronics Letters*, pp. 163 – 165, 1995.
[3] Divsalar, D., S. Dolinar, R.J. McEliece, and F. Pollara, "Performance analysis of turbo codes," *MILCOM '95*, pp.91-96.
[4] Sklar, B., *Digital Communications: Fundamentals and Applications,2nd edition*, Prentice Hall, Englewood Cliffs, 2001.
[5] Heegard, C. and S.B. Wicker, *Turbo Coding, 1st edition,* Kluwer Academic Publisher, Boston, 1999.

[6] Valenti, M.C. and J. Sun, “The UMTS Turbo code and an Efficient Decoder Implementation Suitable for Software-Defined Radios,” *International Journal of Wireless Information Networks*, vol.8, no.4, pp. 203-214, 2001.
[7] Vogt, J., and A. Finger, “Improving the Max-Log-MAP turbo decoder,” *Electronics Letters*, pp. 1937 – 1939, 2000.
[8] Claussen, H., H.R. Karimi, and B. Mulgrew, “Improved Max-Log-MAP turbo-decoding using maximum mutual information combining,” *PIMRC 2003*, pp. 424 - 428.
[9] Papke,L., P. Robertson, and E. Villebrun, “Improved decoding with the SOVA in a parallel concatenated (Turbo-code) scheme,” *ICC’96*, pp. 102 - 106.