



**HAL**  
open science

# A very low complexity block turbo decoder for product codes

Ramesh Pyndiah, P. Combelles, Patrick Adde

► **To cite this version:**

Ramesh Pyndiah, P. Combelles, Patrick Adde. A very low complexity block turbo decoder for product codes. GLOBECOM'96: IEEE Global telecommunications conference, Nov 1996, London, United Kingdom. pp.101 - 105, 10.1109/GLOCOM.1996.594341 . hal-02489017

**HAL Id: hal-02489017**

**<https://hal.science/hal-02489017>**

Submitted on 24 Feb 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A very low complexity block turbo decoder for product codes

Ramesh PYNDIAH\*, Pierre COMBELLES\*\* and Patrick ADDE\*

\*Télécom Bretagne, Technopôle Brest Iroise, BP-832, 29285 BREST, FRANCE.  
(Tel : (33) 98 00 13 92, Fax : (33) 98 00 10 98, email : ramesh.pyndiah@enst-bretagne.fr)

\*\* CCETT-DHN, 4 rue du Clos-Courtel, BP-59, 35512 CESSON-SEVIGNE, FRANCE.  
(Tel : (33) 99 12 41 11, Fax : (33) 99 12 40 98)

**ABSTRACT** - This paper presents a low complexity block turbo decoder for product codes. This new decoder, which has been derived from the near-optimum block turbo decoder described in [1], is a very good compromise between complexity and performance. For performance evaluation, we have considered the  $[BCH(64,57,4)]^2$  product code transmitted over a Gaussian channel using a QPSK modulation. The complexity of the new block turbo decoder is about ten times less than that of the near-optimum block turbo decoder for a coding gain degradation of only 0.7 dB.

## I - Introduction

Today, concatenated coding schemes are considered as being the best solutions for the powerful protection of digital information against errors at the receiver end due to linear perturbations of the received signal. The motivation for using concatenated coding schemes is to achieve the same performance as that of a single and powerful error correcting code but with a lower decoding complexity by associating two (or more than two) less powerful error correcting codes for data coding. In practical applications, the number of codes in a concatenated coding scheme is limited to two and concatenated coding schemes can be divided into two categories. The first one consists of the serially concatenated error correcting codes while the second one concerns error correcting codes concatenated in parallel. In the classical series concatenation scheme, the information bits are first encoded by an external code, for example a Reed-Solomon code. Then the encoded data is interleaved before being encoded by an internal code which is usually a convolutional code or a trellis coded modulation. At the receiver end, the Viterbi algorithm is used to perform the soft decoding of the internal convolutional code. The decoded data is then de-interleaved before being decoded by the external decoder. In practice, the external decoder is a hard (algebraic) decoder because a very powerful Reed-Solomon code is required to correct the residual errors at the output of the Viterbi decoder. Although hard decoding of the Reed-Solomon code is sub-optimal from the theoretical point of view, the serial concatenation of a convolutional code and a powerful Reed-Solomon code is today a standard for numerous applications. This results from the fact that, down to the early 1990's, series concatenated coding schemes gave the best results in terms of coding gain for a reasonable complexity of the decoder.

The concept of product code, which is a series concatenated coding scheme, was introduced, as early as 1954, by Elias [2]. The information bits are placed in a matrix. Then, the rows of the matrix are encoded by a linear block code and the columns of the matrix are encoded by a second block code. At the receiver end, the received matrix is decoded sequentially along the rows and columns of the matrix and this decoding process can be iterated. Although the product code is also a very attractive candidate for the construction of powerful error

correcting schemes, it did not yield very good performance until the early 1990's. The deceiving performance of product codes was due mainly to the use of sub-optimal hard decoders for decoding the rows and columns of the matrix. In 1994, R. Pyndiah *et al* [1] proposed a near-optimum iterative algorithm for decoding product codes. This algorithm is based on soft decoding of the component codes and soft decision at the output of the elementary decoders. Thus, at each iteration, the elementary decoders can operate under near-optimum conditions, that is Maximum Likelihood Sequence Decoding. Coding gains close to the theoretical coding gain expected from product codes when using Maximum Likelihood Sequence Decoding were obtained. For a BER (Bit Error Rate) of  $10^{-5}$ , the signal to noise ratio ( $E_b/N_0$ ) was at  $2.5 \pm 0.2$  dB of Shannon's theoretical limit for different BCH product codes. These performance are comparable to those obtained with series concatenated coding schemes and also to those of convolutional turbo codes, proposed by C. Berrou [2] in 1993, which are parallel concatenated coding schemes. In this paper we shall refer to the iterative decoding algorithm of product codes as the block turbo decoder.

Series and parallel concatenated coding schemes are today comparable in terms of performance. The dominating factor in the choice of one solution among these different candidates today is the complexity of the decoders. The near-optimum iterative decoding algorithm (block turbo-decoder) proposed in [1] has a much higher complexity and performance than previous iterative decoding algorithms[3][4]. From there on, we tried to reduce the complexity of the block turbo decoder and we have developed a very low-complexity block turbo decoder which exhibits only a 0.7 dB degradation of the coding gain.

This paper describes the modifications we have brought to the near-optimum block turbo decoder to develop a low-complexity block turbo decoder. In section II, we recall the minimum requirements concerning product codes and the near optimum block turbo decoder for a better understanding of the modifications we propose. In section III, we analyze the complexity of the block turbo decoder and we identify the procedures which have the highest computation complexity. Section IV is dedicated to the modifications in the block turbo decoder to obtain the low complexity block turbo decoder. The performance of the low complexity block turbo decoder are given in section V and in section VI, we draw some conclusions.

## II - Block turbo code

Let us consider two systematic linear block codes  $\mathcal{C}^1$  having parameters  $(n_1, k_1, \delta_1)$  and  $\mathcal{C}^2$  having parameters  $(n_2, k_2, \delta_2)$  where  $n_i, k_i$  and  $\delta_i$  ( $i=1,2$ ) stand for code length, number of information bits and minimum Hamming distance

of code  $\mathcal{C}^i$ . The product code  $\mathcal{P} = \mathcal{C}^1 \otimes \mathcal{C}^2$  is obtained by placing  $(k_1 \times k_2)$  information bits in a matrix of  $k_1$  rows by  $k_2$  columns. The  $k_1$  rows of the matrix are coded using code  $\mathcal{C}^2$  and the  $k_2$  columns of the resulting  $(k_1 \times k_2)$  matrix are coded using code  $\mathcal{C}^1$ . All the  $n_1$  rows of the product code are code words of  $\mathcal{C}^2$  exactly as the  $n_2$  columns are code words of  $\mathcal{C}^1$  by construction. Furthermore, the parameters of the product code  $\mathcal{P}$  are given by  $n = n_1 \times n_2$ ,  $k = k_1 \times k_2$  and  $\delta = \delta_1 \times \delta_2$  and the code rate  $R$  is given by  $R = R_1 \times R_2$  where  $R_i$  is the code rate of code  $\mathcal{C}^i$ .

Let us now consider the transmission of the  $n$  binary symbols  $\{-1, +1\}$  of  $\mathcal{P}$  using a QPSK signaling over a Gaussian channel. Suppose that the coded matrix  $[\mathbf{E}]$  is transmitted column by column and let  $\mathbf{E} = (e_1, \dots, e_l, \dots, e_{n_1})$  be one of the columns of the matrix. The vector  $\mathbf{R} = (r_1, \dots, r_l, \dots, r_{n_1})$  at the output of the coherent demodulator, for a transmitted column  $\mathbf{E}$ , can be written as :

$$\mathbf{R} = \mathbf{E} + \mathbf{G} \quad (1)$$

where  $\mathbf{G} = (g_1, \dots, g_l, \dots, g_{n_1})$  are Additive White Gaussian Noise (AWGN) samples of standard deviation  $\sigma$ . By using Maximum Likelihood Sequence Decoding, one can show that optimum decision  $\mathbf{D}$  corresponding to a transmitted column  $\mathbf{E}$  of the matrix is given by :

$$\mathbf{D} = \mathbf{C}^i \text{ if } |\mathbf{R} - \mathbf{C}^i|^2 < |\mathbf{R} - \mathbf{C}^l|^2 \quad \forall l \in [1, 2^{k_1}], l \neq i \quad (2)$$

where  $\mathbf{C}^i = (c_1^i, \dots, c_l^i, \dots, c_{n_1}^i)$  is the  $i^{\text{th}}$  code word of  $\mathcal{C}^1$  and :

$$|\mathbf{R} - \mathbf{C}^i|^2 = \sum_{j=1}^{n_1} (r_j - c_j^i)^2 \quad (3)$$

is the squared Euclidean distance between  $\mathbf{R}$  and  $\mathbf{C}^i$ . As the number of code words increases exponentially with the number of information bits, maximum likelihood sequence decoding becomes too complex for block codes with more than ten information bits. Long block codes are in fact the most interesting for practical applications since they are the ones having the highest code rate. To reduce the complexity of the soft block decoder, Chase proposed in 1972 an algorithm [5] which approximates maximum likelihood sequence decoding of block codes with a low computation complexity and a small performance degradation. Instead of reviewing all the code words  $\mathbf{C}^l$  for  $1 \leq l \leq 2^{k_1}$ , the Chase algorithm searches for the code words at Hamming distance within a sphere of radius  $(\delta_1 - 1)$  centered on  $\mathbf{Y} = (y_1, \dots, y_l, \dots, y_{n_1})$  where  $y_j = 0.5(1 + \text{sgn}(r_j))$  and  $y_j \in \{0, 1\}$ . To further reduce the number of reviewed code words, only the most probable code words within the sphere are selected by using channel information  $\mathbf{R}$ . This search procedure can be decomposed into three steps :

- 1) determine the position of the  $[\delta_1/2]$  least reliable binary elements of  $\mathbf{Y}$  using  $\mathbf{R}$ ,

- 2) form test patterns  $\mathbf{T}^q$  defined as  $n_1$ -dimensional binary vectors with a "1" in the least reliable positions and "0" in the other positions,
- 3) decode  $\mathbf{Z}^q = \mathbf{Y} \oplus \mathbf{T}^q$  using a hard decoder and memorize the code words  $\mathbf{C}^q$ .

The maximum likelihood transmitted sequence  $\mathbf{D}$  is then given by decision rule (2) with the reviewed code words restricted to those found at step 3) above. Once we have determined the optimum decision  $\mathbf{D}$ , we have to compute the reliability of each of the components of vector  $\mathbf{D}$  in order to obtain the soft decisions at the output of the decoder. The reliability of decision  $d_j$  can be obtained from the Log Likelihood Ratio (LLR) of transmitted symbol  $e_j$  which is defined by :

$$(LLR)_j = \ln \left( \frac{\Pr\{e_j = +1 / \mathbf{R}\}}{\Pr\{e_j = -1 / \mathbf{R}\}} \right) \quad (4)$$

By considering the different authorized code words, the numerator of (4) can be expressed as :

$$\Pr\{e_j = +1 / \mathbf{R}\} = \sum_{\mathbf{C}^i \in \mathcal{S}_j^{+1}} \Pr\{\mathbf{E} = \mathbf{C}^i / \mathbf{R}\} \quad (5)$$

where  $\mathcal{S}_j^{+1}$  is the set of code words  $\{\mathbf{C}^i\}$  such that  $c_j^i = +1$ . For the denominator we obtain a similar expression where  $\mathcal{S}_j^{+1}$  is replaced by  $\mathcal{S}_j^{-1}$  which is the set of code words  $\{\mathbf{C}^i\}$  such that  $c_j^i = -1$ . By applying BAYES rule and assuming that the different code words are uniformly distributed, we obtain for  $(LLR)_j$  the following expression :

$$(LLR)_j = \ln \left( \frac{\sum_{\mathbf{C}^i \in \mathcal{S}_j^{+1}} P\{\mathbf{R} / \mathbf{E} = \mathbf{C}^i\}}{\sum_{\mathbf{C}^i \in \mathcal{S}_j^{-1}} P\{\mathbf{R} / \mathbf{E} = \mathbf{C}^i\}} \right) \quad (6)$$

where  $P\{\}$  is the probability distribution function of  $\mathbf{R}$  conditionally to  $\mathbf{E}$ . As shown in [1], relation (6) can be approximated by :

$$(LLR)_j \approx \frac{1}{2\sigma^2} \left( |\mathbf{R} - \mathbf{C}^{-1(j)}|^2 - |\mathbf{R} - \mathbf{C}^{+1(j)}|^2 \right) \quad (7)$$

where  $\mathbf{C}^{+1(j)}$  is the code word in  $\mathcal{S}_j^{+1}$  at minimum distance from  $\mathbf{R}$  and  $\mathbf{C}^{-1(j)}$  is the code word in  $\mathcal{S}_j^{-1}$  at minimum distance from  $\mathbf{R}$ . By expanding relation (7) using (3) we finally obtain the following expression :

$$(LLR)_j \approx \frac{2}{\sigma^2} \left( r_j + \sum_{l=1, l \neq j}^n \eta_l c_l^{+1(j)} p_l \right) \quad (8)$$

where :

$$p_l = \begin{cases} 0 & \text{if } c_l^{+1(j)} = c_l^{-1(j)} \\ 1 & \text{if } c_l^{+1(j)} \neq c_l^{-1(j)} \end{cases} \quad (9)$$

By normalizing the approximated  $(LLR)_j$  with respect to  $2/\sigma^2$  we obtain :

$$r'_j = r_j + w_j \quad (10)$$

where :

$$w_j = \sum_{l=1, l \neq j}^n \eta c_l^{+1(j)} p_l \quad (11)$$

The estimated normalized  $(LLR)_j$ ,  $r'_j$  (relation 10), is given by  $r_j$  plus  $w_j$  which is the additional information given by the decoder to the reliability of the decoded bit. The approximated normalized LLR of  $e_j$  ( $r'_j$ ) is an estimation of the soft decision at the output of the soft block decoder. It has the same sign as  $d_j$  and its absolute value indicates the reliability of the decision. To compute  $(LLR)_j$  using (7) we must identify the two code words  $\mathbf{C}^{+1(j)}$  and  $\mathbf{C}^{-1(j)}$ . This is done by increasing the number of least reliable bits and test patterns in the Chase algorithm. For each code word  $\mathbf{C}^q$  found at step 3), we compute its Euclidean distance from  $\mathbf{R}$  :

$$M^q = |\mathbf{R} - \mathbf{C}^q|^2 \quad (12)$$

As in the Chase algorithm we select the code word  $\mathbf{D}$  at minimum Euclidean distance from  $\mathbf{R}$ . Then we search for the code word  $\mathbf{C}$  at minimum Euclidean distance from  $\mathbf{R}$  such that  $c_j \neq d_j$ . If we find such a code word then  $r'_j$  is given by :

$$r'_j = \left( \frac{(M^{\mathbf{C}} - M^{\mathbf{D}})}{4} \right) d_j \quad (13)$$

else we use the following equation :

$$r'_j = r_j + (\beta \times d_j) \quad (14)$$

where  $\beta$  is a constant which increases with the iteration number and is optimized by simulation. Equation (14) can be justified by the fact that  $w_j$  has the same sign as  $d_j$  but we have not been able to compute the amplitude of the reliability since we have not found code word  $\mathbf{C}$ .

On receiving matrix  $[\mathbf{R}]$  corresponding to a transmitted matrix  $[\mathbf{E}]$ , the first decoder performs the soft decoding of the rows (or columns) of the matrix, estimates the matrix  $[\mathbf{R}']$  using relations (13) or (14) and gives as output  $[\mathbf{W}(2)]$ . Then the next decoder performs the same operations on the columns (or rows) using as input :

$$[\mathbf{R}(2)] = [\mathbf{R}] + \alpha(2)[\mathbf{W}(2)] \quad (15)$$

The decoding procedure described above is then iterated by cascading elementary decoders. The constant  $\alpha(m)$  is used to reduce the influence of  $[\mathbf{W}(m)]$  in the first iterations when the BER is relatively high.

### III - Complexity of block turbo decoder

In 1994, R. Pyndiah *et al* showed [1] that the block turbo decoder, described in section II, provides near optimum decoding performance for binary product codes when transmitted using QPSK signaling over a Gaussian channel. As stated in our introduction, the series or parallel concatenated coding schemes exhibit, today, comparable performance in terms of

coding gain and the dominating factor in the choice between one solution or the other will be decoder complexity. This explains our motivation to analyze the complexity of the block turbo decoder and to reduce its complexity while maintaining a small coding gain degradation.

Let us estimate the complexity of the block turbo decoder described in section II and identify the procedures which have the highest computation complexity. We shall estimate the average complexity for a given column (or a given row) of the coded matrix so that we can directly compare this complexity to that of a simple block decoder. If we consider the decoding of a column of the matrix, the first source of complexity in the block turbo decoder is the number of times the hard decoding of the block code is performed. In reference [1], the number of test patterns used was seventeen ( $q=17$ ) and thus, for this procedure (procedure-1), the decoding complexity of one column is  $q$  times the complexity of the hard decoder for code  $\mathcal{E}^1$ . If we use the same code for the rows and columns of the matrix that is  $\mathcal{E}^1 = \mathcal{E}^2$ , then the average complexity of procedure-1 for decoding one column is :

$$Q_1 = (q \times IT \times 2) \quad (16)$$

times the complexity of the hard decoder where  $IT$  is the number of iterations. Thus if we consider a three iterations turbo decoder, the average complexity of procedure-1 for decoding one column is  $Q_1 = (17 \times 3 \times 2) = 102$  times that of the hard decoder.

The second procedure (procedure-2) which requires a high computation complexity is the computation of the extrinsic information  $\mathbf{W}$  associated to a decision  $\mathbf{D}$ . For each component  $w_j$ , procedure-2 has to search among the  $q$  code words given by procedure-1 if there is a code word  $\mathbf{C}$  at smallest distance from  $\mathbf{R}$  such that  $c_j \neq d_j$ . If we find such a code word  $\mathbf{C}$  then we use relation (13) else we use relation (14) to compute  $w_j$ .

Although this procedure involves mainly low complexity compare and save operations, the global complexity of procedure-2 is relatively high because these operations are performed a large number of times. If we call  $P$  the complexity of an elementary compare and save operation, the average complexity of procedure-2 for the decoding of a column is given by :

$$Q_2 = (q \times n_1 \times IT \times 2) \times P \quad (17)$$

for product codes with  $\mathcal{E}^1 = \mathcal{E}^2$ . If we consider a block turbo decoder with parameters  $q = 17$ ,  $n_1 = 64$  and  $IT = 3$ , then the average complexity of procedure-2 for decoding one column of the matrix is  $Q_2 = (17 \times 64 \times 3 \times 2) \times P = 6528 \times P$ . To the complexity of procedures 1 and 2 we must add that of the computation of the metrics (12) and a few compare and save operations which can be considered as negligible when compared to procedures 1 and 2.

Thus it is clear that the block turbo decoder, described in section II, is a relatively complex algorithm when compared to a hard block decoder. The next section of this paper describes the modifications we have brought to the block turbo decoder to reduce its complexity.

### IV - Low complexity block turbo decoder

To reduce the complexity of the block turbo decoder, we must first reduce the number of test patterns  $q$  and then

simplify procedure-2 so as to bring down the number of compare and save operations to a more reasonable value. In a first attempt, we tried to reduce  $q$  and we observed that the degradation in coding gain was relatively small for  $q \geq 8$ . On the other hand, for  $q < 8$  we observed an important degradation of the coding gain as we reduced  $q$ . Thus we are limited to  $q=8$  for coding gain considerations and the average decoding complexity for one column is only divided by a factor 2. Next, we tried to reduce the complexity of procedure-2. For this we tried to find an easy way to identify code word  $\mathbf{C}$  without going through the search procedure. We found that we could considerably reduce  $Q_2$ , with a small coding gain degradation, by replacing code word  $\mathbf{C}$  by decision  $\mathbf{D}(m-1)$  when computing the extrinsic information  $\mathbf{W}(m+1)$ . This is the main breakthrough which led to the new low complexity block turbo decoder.

Let us consider the  $m^{\text{th}}$  decoding of the product code and suppose that we start by decoding the columns of the product code. For odd values of  $m$ , the decoder processes the columns of the product code while for even values of  $m$ , the decoder processes the rows of the product code. Suppose now that the decoder is performing the  $m^{\text{th}}$  decoding of the product code and that it has gone through procedure-1 and that it has selected the optimum code word  $\mathbf{D}(m)$  which is at minimum Euclidean distance from the input data  $\mathbf{R}(m)$ . To compute the new extrinsic information  $\mathbf{W}(m+1)$  we propose to use the following equation :

$$w_j(m+1) = \left| \frac{|\mathbf{R}(m) - \mathbf{D}(m-1)|^2 - |\mathbf{R}(m) - \mathbf{D}(m)|^2}{4} \right| \times d_j(m) - r_j(m) \quad (18)$$

when  $d_j(m) \neq d_j(m-1)$  and otherwise we use :

$$w_j(m+1) = \beta \times d_j(m) \quad (19)$$

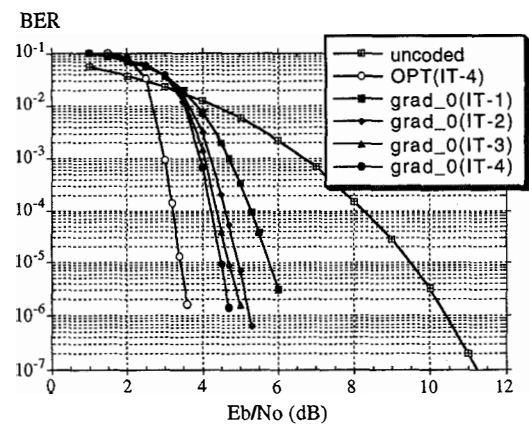
By using this algorithm, the complexity of procedure-2 is now equal to  $Q_2 = (n_1 \times IT \times 2) \times P'$  where  $P'$  consists only of a comparison between two binary elements and thus has a much lower complexity than  $P$ . Furthermore, we noticed that this new iterative decoding algorithm was less sensitive to the number of test patterns. For product codes based on BCH codes with a minimum distance of 4, we managed to reduce the number of test patterns down to four ( $q=4$ ) while maintaining a small degradation of the coding gain. Thus the complexity of procedure-1 is now equal to  $Q_1 = (4 \times 3 \times 2) = 24$  times that of the hard decoder while the complexity of procedure-2 is equal to  $Q_2 = (64 \times 3 \times 2) \times P' = 384 \times P'$ . The complexity of procedure-1 has been divided by four and that of procedure-2 has been divided by more than seventeen.

Before proceeding to the simulations, we shall make a few comments on this new algorithm. First of all,  $\mathbf{D}(m-1)$  is not always a code word. Let us consider  $m$  odd and suppose that the decoder is processing the columns of the product code. Then  $\mathbf{D}(m)$  is a code word. However in matrix  $[\mathbf{D}(m-1)]$ , the decoder guarantees that we have code words along the rows of the matrix but not along the columns. Thus, there is no guarantee that column  $\mathbf{D}(m-1)$  is a code word and thus  $\mathbf{W}(m+1)$  does not have the same interpretation in the new algorithm as in the previous one. Although the equations in the new algorithm are quite similar to those in the near-optimum

algorithm, the theoretical justification of this new algorithm is quite different from that of the previous one. This algorithm is in fact closer to a gradient algorithm and so we shall refer to it as the gradient algorithm.

## V - Performance of the low complexity decoder

We have evaluated the performance of the low complexity block turbo decoder on an additive white gaussian channel when using QPSK signaling. For our evaluations, we have considered the product code with parameters  $\mathcal{C}^1 = \mathcal{C}^2 = \text{BCH}(64,57,4)$ . We used four test patterns in our simulations which are based on the three least reliable bits in position  $(I_1, I_2, I_3)$ . As we shall be considering different versions of this low complexity algorithm or gradient algorithm we shall call this algorithm grad\_0. The Bit Error Rate (BER) function of the signal to noise ratio (SNR) obtained when decoding product code  $[\text{BCH}(64,57,4)]^2$  using the grad\_0 algorithm is given in figure 1 after each iteration (IT) up to IT-4.



**Figure 1** : Bit Error Rate versus SNR ( $E_b/N_0$ ) after each iteration when decoding product code  $[\text{BCH}(64,57,4)]^2$  using grad\_0 algorithm.

From figure 1, we clearly observe that there is an amelioration of the BER at each additional iteration for any SNR greater or equal to 3.5 dB. However, there is a substantial degradation of the BER for a given SNR when we compare these curves with those (OPT) obtained with the near-optimum algorithm [1]. We shall now consider the near-optimum algorithm as a reference for our comparisons. For a BER of  $10^{-5}$ , the coding gain obtained with the grad\_0 algorithm exhibits a degradation of 1.1 dB at iteration-4. This is a reasonable penalty for such an important reduction in decoder complexity. Furthermore, the degradation in coding gain (1.1 dB) at iteration-4 is nearly constant for a BER less or equal to  $10^{-3}$  (see fig. 1).

From there on, we tried to reduce the coding gain degradation of the gradient algorithm. The first amelioration was obtained by allowing the selection of a code word only if :

$$\left\{ |r_j| \text{ such that } d_j(m) \neq d_j(m-1) \right\} < \gamma(m) \quad (20)$$

where  $\gamma(m)$  is a constant which increases with  $m$ . This modified version of the gradient algorithm will be referred to as grad\_1. From figure 2, we observe that grad\_1 exhibits a coding gain degradation of only 0.9 dB.

The last amelioration was obtained by weighting the constant  $\beta(m)$  by the following function :

$$\chi(m) = \left( \frac{2}{2 + d_H(\mathbf{D}(m), \mathbf{D}(m-1))} \right) \quad (21)$$

where  $d_H$  is the Hamming distance. This algorithm which is referred to as grad\_2 exhibits a degradation of only 0.7 dB (see fig. 2).

These two ameliorations represent only a small increase in the complexity of the gradient algorithm and they have reduced the coding gain degradation of the gradient algorithm down to only 0.7 dB. This new block turbo decoder offers a very good compromise between complexity and performance.

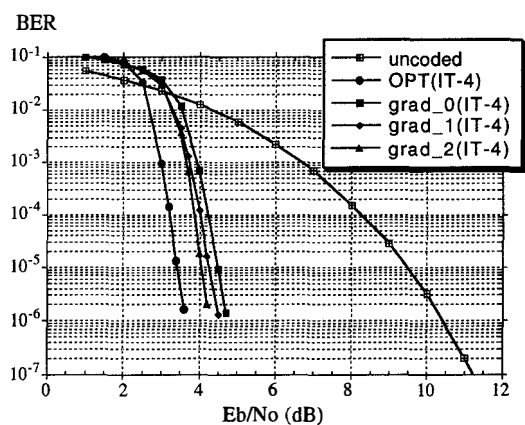


Figure 2 : Bit Error Rate versus SNR at iteration-4 when decoding product code  $[BCH(64,57,4)]^2$  with different block turbo decoders.

## VI - Conclusion

The low complexity block turbo decoder presented in this paper offers a good compromise between complexity and performance. When compared with the near-optimum block turbo decoder, this low complexity turbo decoder exhibits a coding gain degradation of only 0.7 dB for a complexity which is divided by ten. The average decoding complexity of a row (or a column) of a product code is slightly greater than the complexity of a soft block decoder [5] multiplied by the number of times the product code is decoded. With this low complexity block turbo decoder, the time required to decode the columns (or rows) of a product code is much smaller than for the near-optimum algorithm. Thus, instead of cascading elementary decoders to iterate the decoding procedure [2], we can use of the same decoder to perform several iterations. The complexity of this new block turbo decoder does not depend on the number of iterations. This represents another significant reduction in terms of circuit complexity. If we consider a three iterations turbo decoder, the circuit complexity is divided by six when compared with a pipe-lined turbo decoder. Furthermore, by using the same turbo decoder to perform three iterations, we can reduce by a factor six the decoding delay which is also an important parameter for some applications. This feature gives an important advantage to block turbo codes over convolutional turbo codes [2]. Indeed, for convolutional turbo codes one must cascade the elementary decoders to perform the iterations because the data blocks are correlated. The data flow through each elementary decoder of the convolutional turbo code must

be continuous and thus it is not possible to use the same elementary decoder to perform several iterations on the same data block as for block turbo codes.

Although this new algorithm is very interesting for practical applications, we do not have a very clear theoretical justification for explaining its relatively good behaviour. We believe that the theoretical justification for this new algorithm is quite different from that of the near-optimum algorithm even though the first one has been derived from the second one. The low complexity block turbo decoder has a number of similarities with the gradient algorithm and we are actually investigating the theoretical justification of this new algorithm.

## Acknowledgement

The authors thank the Centre National d'Etudes des Télécommunications (CNET) and the Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT) for their financial support.

## References

- [1] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, "Near optimum decoding of products codes," in proc. of *IEEE GLOBECOM '94 Conference*, vol. 1/3, Nov.- Dec. 1994, San Francisco, pp. 339-343 .
- [2] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : Turbo-codes (1)," *IEEE Int. Conf. on Comm. ICC' 93*, vol 2/3, May 1993, pp. 1064-1071.
- [3] S. M. Reddy, "On decoding iterated codes," *IEEE Trans. Inform. Theory*, vol IT-16, Sept 1970, pp. 624-627.
- [4] S. M. Reddy and J. P. Robinson, "Random error and burst correction by iterated codes," *IEEE Trans. Inform. Theory*, vol IT-18, Jan. 1972, pp. 182-185.
- [5] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol IT-18, Jan. 1972, pp. 170-182.
- [6] G. Battail, "Pondération des symboles décodés par l'algorithme de Viterbi", *Ann. Télécommun.*, 42, n° 1-2, 1987, pp.31-38.
- [7] R. Pyndiah, A. Picart and A. Glavieux, "Performance of block turbo coded 16-QAM and 64-QAM modulations," in proc. of *IEEE GLOBECOM '95 Conference*, vol. 2/3, Nov. 1995, pp. 1039-1044 .
- [8] A. Picart and R. Pyndiah, "Performance of turbo-decoded product codes used in multilevel coding," in proc. of *IEEE ICC'96 Conference*, Dallas TX.
- [9] F.J. Macwilliams and N.J.A. Sloane, "The theory of error correcting codes," *North-Holland publishing company*, 1978, pp. 567-580.
- [10] P. Adde, R. Pyndiah and O. Raoul, "Performance and complexity of a block turbo coder circuit," submitted for publication in *ECECS'96 Conference*.