Applied Mathematical Sciences, Vol. 4, 2010, no. 41, 2021 - 2038

Iterative Decoding of Generalized Parallel Concatenated OSMLD Codes

Fouad Ayoub

UFR : ACSYS, MohammedV-Agdal
University, Faculty of Sciences, 4 Avenue Ibn
Battouta B.P. 1014 Agdal, Rabat, Morocco
ayoubfouadnf@gmail.com

Mostafa Belkasmi

MohammedV-Souisi University
SI2M Labo, ENSIAS, Rabat, Morocco
belkasmi@ensias.ma

Idriss Chana

MohammedV-Souisi University
SI2M Labo, ENSIAS, Rabat, Morroco
id_chana@yahoo.fr

Abstract

In this paper we well introduce a new decoding algorithm for generalized parallel concatenated block codes(GPCB). We are interested in decoding generalized parallel concatenated block codes based on two systematic one step majority logic decoding (OSMLD) codes using a soft output version of threshold algorithm with Lucas's connexion scheme. The effects of various component codes, interleaver size (Number of sub-blocks), interleaver pattern, and the number of iterations are investigated. The simulation results show that the slope of curves and coding gain are improved by increasing the number of iterations and/or the interleaver size. Proposed decoding scheme provides a performance near the Shannon limit as it is evident from simulation results.

Keywords: Parallel concatenated block codes, OSMLD codes, iterative decoding

1 Introduction

Turbo codes were introduced in 1993 by Berrou, Glavieux and Thitimajshima [1] as a class of concatenated codes. To construct the turbo codes they used concatenated recursive convolutional codes with an interleaver inserted between the two recursive convolutional encoders. In 1994, R.Pyandiah [5] adopted this process for the product block codes using as component decoder a soft output version of Chase decoder, in 1998, Lucas applied the same process for several families of block codes(e g OSMLD) using the soft output version of Harthman/Rudolph as a component decoder.
Belkasmi et al [2] have presented an iterative decoding algorithm based on a SISO extension form of Massey algorithm [4]. In this work they use an iterative decoding process applied to decode parallel concatenated OSMLD codes and using Pyndiah's connexion scheme. In this perspective we present a generalisation of Belkasmi's [2] work using Lucas's connexion scheme.
In this paper, section 2 describes the encoder structure and definition of the generalized parallel concatenated block codes. Section 3 presents the component decoder. Section 4 describes the iterative decoding of the GPCB codes. The simulation results are given in section 5. Section 6 concludes this paper.

2 Generalized Parallel concatenated block codes

The structure of parallel concatenated block codes is shown in figure 1, two linear block encoders are linked through an interleaver of length k, so that every block of k information bits entering the second encoder is just a permuted version of the block that entered the first encoder. The PCB codeword is then formed by adding to the input information bits the parity-check bits generated by the first and the second encoder.

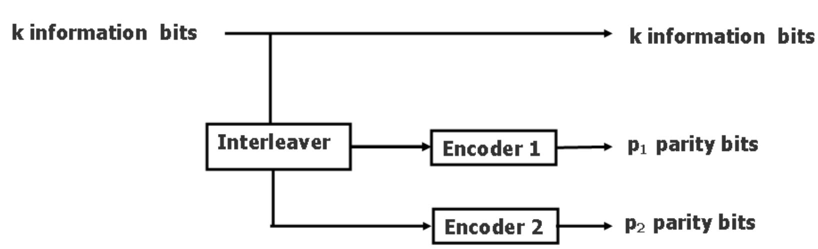
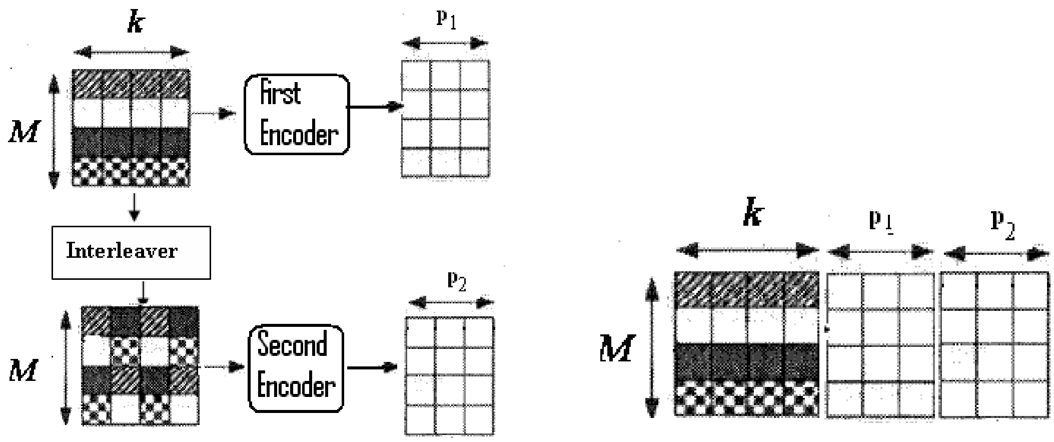


Figure 1: Encoder structure of parallel concatenated block (PCB) codes

Figure 2 shows the structure of generalized parallel concatenated block codes. This construction was introduced independently by Nilson et al [9] and Benedetto et al [10]. A block of N data bits at the input of the GPCB encoder is subdivided into M sub-blocks. Each sub-block of length k is encoded using a component encoder in order to produce parity check bits. The input block is scrambled by the interleaver, denoted by Π, before entering the second encoder. The codeword of GPCB code consists of the input block followed by the parity check bits of both encoders.
A systematic GPCB code is based on two systematic block component codes, C_1 with parameters (n_1, k), and C_2 with parameters (n_2, k). The length of the information word to be encoded by the GPCB code is given by the size of the interleaver $N = M \times k$. The first encoder produces $P_1 = M \times (n_1 - k) = M \times p_1$ parity check bits. The second encoder produces $P_2 = M \times (n_2 - k) = M \times p_2$ parity check bits. Thus the total number of parity bits generated by the GPCB encoder is $P = P_1 + P_2 = M \times (n_1 + n_2 - 2 \times k)$. The length of the GPCB codeword is given by $L = N + P = M \times (n_1 + n_2 - k)$. Consequently, the code rate of the GPCB codes can be computed by :

$$\frac{N}{L} = \frac{M \times k}{M \times (n_1 + n_2 - k)} = \frac{k}{n_1 + n_2 - k}$$

This implies that the GPCB code rate is independent of the number of sub-blocks M. Table 1 gives some examples of codes based on this construction where the two components codes are the same, and for different values for M. In this contribution several interleaving techniques were invoked such as random, block, diagonal, helical, and cyclic interleaver.



(a): Structure of encoder (b): Transmitted codeword

Figure 2: Encoder structure of generalized parallel concatenated block codes

3 Component decoder

3.1 Threshold decoding

3.1.1 One step majority logic codes

Consider an (n, k) linear code C with parity-check matrix H. The row space of H is an $(n, n-k)$ linear code, denoted by $C^{\perp}$, which is the dual code of C, or the null space of C. For any vector U in C and any vector V in $C^{\perp}$, the inner product of U and V is zero. Now suppose that a code vector U in C is BPSK modulated and transmitted over an AWGN channel. Let $E(e_1, e_2, \ldots, e_n)$ and $R(r_1, r_2, \ldots, r_n)$ be the error vector and the received binary vector respectively. Then $R = U + E$. For any vector V in the dual code $C^{\perp}$, we can form the following linear sum of the received digits:

$$A = \sum_{i=0}^{n} r_i \cdot v_i$$

This is called a parity-check sum. Using the fact that $\langle U, V \rangle = 0$, we obtain the following relationship between the check sum A and error digits in E:

$$A = \sum_{i=1}^{n} e_i \cdot v_i$$

Suppose that there exist J vectors in the dual code $C^{\perp}$, which have the following properties :

Table 1: Some Examples of GPCB codes

Component code	**M**	**GPCB-OSMLD Code parameters**	**Code rate**
DSC(7,3)	1	(11,3)	0.27
DSC(7,3)	100	1100,300)	0.27
BCH(15,7)	1	(23,7)	0.30
BCH(15,7)	10	(230,70)	0.30
DSC(21,11)	10	(310,110)	0.35
DSC(21,11)	200	(6200,2200)	0.35
DSC(73,45)	1	(101,45)	0.44
DSC(73,45)	10	(1010,450)	0.44
DSC(73,45)	100	(10100,4500)	0.44
DSC(73,45)	200	(20200,9000)	0.44
DSC(273,191)	1	(355,191)	0.53
DSC(273,191)	10	(3550,1910)	0.53
DSC(273,191)	100	(35500,19100)	0.53
DSC(273,191)	300	(106500,57300)	0.53

1. The j^{jh} component of each vector is a 1
2. For $i \neq j$ there is at most one vector whose i^{th} component is a 1

These J vectors are said to be orthogonal on the j^{th} digit position. We call them orthogonal vectors. Now, let us form J parity-check sums from these J orthogonal vectors. For each i in $\{1, \ldots, J\}$

$$A_i = \sum_{p \neq j} e_p + e_j$$

We see that the error digit e_j is checked by all the check sums above. Because of the second property of the orthogonal vectors, any error digit other than e_j is checked by at most one check sum.
If all the error digits in the sum A_i are zero, the value of A_i is equal to e_j. Based on this fact, the parity-check sums orthogonal on e_j can be used to estimate e_j, or to decode the received digit r_j.
Table 2 shows some examples of (OSMLD) codes. In this table we used the abbreviation DSC for Difference Set Cyclic codes, EG for Euclidean Geometry codes and BCH for Bose Chaudhuri and Hocquenghem codes. The EG codes used in this study are 0-order and according to [12], they are OSMLD codes.

Table 2: Set of OSMLD Codes

n	**k**	**J**	**Minimal distance**	**Rate**	**Code family**
7	3	3	4	0.42	DSC
15	7	4	5	0.46	BCH
21	11	5	6	0.52	DSC
63	37	8	9	0.58	EG
73	45	9	10	0.61	DSC
255	175	16	17	0.68	EG
273	191	17	18	0.69	DSC
1023	781	32	33	0.76	EG
1057	813	33	34	0.76	DSC
4161	3431	65	66	0.82	DSC

3.1.2 Majority logic decoding principle

Majority logic decoding is a binary input binary output decoding algorithm [12]. It is introduced in this section with an example.
The error digit e_j is decoded as 1 if at least one-half of the check sums orthogonal on e_j, are equal to 1; otherwise, e_j is decoded as 0 like majority rule. If $C(n,k)$ is a cyclic code, each e_i can be decoded simply by cyclically permuting the received word R into the buffer store.
As an example : Let us consider the (7,3) code, which is the short code in difference set codes (DSC) family (see Table 1). This code is specified by the perfect difference set $P = \{0,2,3\}$ of order 2^1. From this prefect set, we can form the following three check sums orthogonal on e_6 :

$$\begin{cases} A_1 = e_3 & + & e_4 & + & e_6 \\ A_2 = e_1 & + & e_5 & + & e_6 \\ A_3 = e_0 & + & e_2 & + & e_6 \end{cases}$$

If a simple error $e = (000001)$ occur, than we have $A_1 = A_2 = A_3 = 1$. If a double error occur, as an example $e_6 = 1$ and one value of $e_0, \ldots, e_5$ is equal to 1, then two values of A_i are 1. A simple decoding rule can be expressed as :

$$\begin{cases} e_6 = 1 & \text{if at least 2 values of } A_i \text{ are 1} \\ e_6 = 0 & \text{otherwise} \end{cases}$$

3.2 Soft-input soft-output threshold decoding

Threshold decoding is simply the logical extension to soft decisions of majority decoding described above [7]. In the original work of Massey [4], it considered

two alternatives of the decoding algorithm. We consider here the method which use the B_i equations that are obtained from A_i by eliminating the e_j term.
Let us consider a transmission of block coded binary symbols $\{0,1\}$ using a BPSK modulation over AWGN channel, the decoder soft output for the j^{th} bit position of a given soft input $R(r_1,\ldots,r_n)$ is defined as :

$$LLR_j = \ln \frac{P(c_j = 1/R)}{P(c_j = 0/R)} \tag{1}$$

where $C(c_1,\ldots,c_n)$ is the transmitted codeword. Expression (1) is a log likelyhood ratio for the symbol c_j. The hard decision vector corresponding to the received vector R is denoted by $H(h_1,\ldots,h_n)$. For a code with J orthogonal parity check equations, (1) can be expressed as :

$$LLR_j \simeq \ln \frac{P(c_j = 1/\{B_i\})}{P(c_j = 0/\{B_i\})} \tag{2}$$

where B_i, $i \in \{0,\ldots,J\}$ are obtained from the orthogonal parity check equations on the j^{th} bit as follows :
The term B_0 is defined to be $B_0 = h_j$. For each index i in $\{1,\ldots,J\}$ the term B_i is computed by using the i^{th} orthogonal parity equation. By applying BAYES rule, (2) becomes

$$LLR_j \simeq \ln \frac{P(\{B_i\}/c_j = 1) \times P(c_j = 1)}{P(\{B_i\}/c_j = 0) \times P(c_j = 0)} \tag{3}$$

Since the parity check equations are orthogonal on the jth symbol the individual probabilities are all independent and (3) can be rewritten as :

$$LLR_j \simeq \sum_{i=0}^{J} \ln \frac{P(\{B_i\}/c_j = 1)}{P(\{B_i\}/c_j = 0)} + \ln \frac{P(c_j = 1)}{P(c_j = 0)} \tag{4}$$

Assume that the transmitted symbols are equally likely to be 0 or 1, and thus the last term in (4) is null. As a result, we obtain

$$LLR_j \simeq \sum_{i=1}^{J} \ln \frac{P(\{B_i\}/c_j = 1)}{P(\{B_i\}/c_j = 0)} + \ln \frac{P(\{B_0\}/c_j = 1)}{P(\{B_0\}/c_j = 0)} \tag{5}$$

According to [7], (5) can be expressed as

$$LLR_j \simeq (1 - 2B_0) \cdot w_0 + \sum_{i=1}^{J} (1 - 2B_i) \cdot w_i \tag{6}$$

where the value of $(1-2B_0)$ is equal to +1 or -1 and w_i is a weighting term proportional to the reliability of the i^{th} parity check. It is easy to show that:

$$(1-2B_0)\cdot w_0 = \frac{4\cdot E_s}{N_0}\cdot r_j \tag{7}$$

Where E_s is the energy per symbol and N_0 is the noise spectral density.

$$w_i = ln[\frac{1+\prod_{k=1}^{n_i}\tanh(\frac{L_{ik}}{2})}{1-\prod_{k=1}^{n_i}\tanh(\frac{L_{ik}}{2})}] \tag{8}$$

where n_i is the total number of terms in the i^{th} orthogonal parity equation without c_j,ik represents the k^{th} element of the i^{th} parity equation and

$$L_{ik} = \frac{4E_s}{N_0}\cdot \mid r_{ik} \mid \tag{9}$$

Thus the soft output can be split into two terms, namely into a normalized version of the soft input r_j and an extrinsic information W_j representing an estimates made by the orthogonal bits on the current bit c_j. Hence (6) is rewriting in

$$LLR_j = \frac{4E_s}{N_0}\cdot r_j + W_j \tag{10}$$

The algorithmic structure of the SISO threshold decoding can be summarized as follows :

For each j=1,.....,n
Compute the terms B_i *and* $w_i, i \in \{1,......,J\}$
Calculate B_0 *and* w_0
Compute the extrinsic information :

$$W_j = \sum_{i=1}^{J}(1-2B_i)\cdot w_i$$

The Soft-output is obtained by:

$$LLR_j = \frac{4E_s}{N_0}\cdot r_j + W_j$$

4 The proposed iterative decoding of GPCB-OSMLD codes

In this sectoin we describe the proposed decoding algorithm for generalized parallel concatenated block codes, it's designed to decode the generalized parallel concatenated block codes based on two systematic OSMLD codes using a soft output version of threshold algorithm with Lucas's connexion scheme.
This algorithm is iterative as it is shown in figure 3. At each iteration, two component decoders are used. The first one uses the systematic information and the first parity check symbols in order to generate extrinsic information by the SISO Threshold algorithm. This extrinsic information concern as well as : Parts check bits and systematic bits. The first part is used to update the reliabilities of the parity check bits which will be used only by the fist decoder, and the second part is used to update the reliabilities of the systematic bits which will be interleaved and feed into the second decoder. The second decoder also generates the extrinsic information using threshold decoder, and then updates the reliabilities of the parity check bits and systematic bits for the second time in one iteration. The updated reliabilities for systematic bits will be desinterleaved and added to the systematic information used in the last iteration and feed again into first decoder, for the next iteration. The process resume until a maximum number of iterations is reached.

In figure 3, we use some notions which will be defined in the following :
R: Recieved word, it consist of three parts, the systematic information called Y, and (Z_1, Z_2) defined below :
Z_1 : The parity check information generated by first encoder.
Z_2 : The parity check information generated by second encoder.

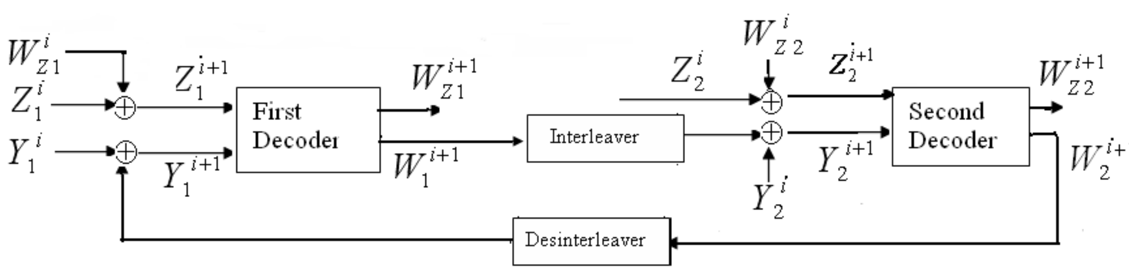


Figure 3: Proposed iterative decoding scheme

We adopt also the notation below:
$Y_1 = Y$: The systematic information present at the entry of first decoder.
$Y_2 = \Pi(Y)$: The systematic information present at the entry of second decoder.
W_1 : The extrinsic information generated by the first decoder for Y_1.

W_2 : The extrinsic information generated by the second decoder for Y_2.
W_{Z_1} : The extrinsic information generated by the first decoder for Z_1.
W_{Z_2} : The extrinsic information generated by the second decoder for Z_2.
D_j : The j^{th} bit of hard decision(decoded word).
i_{max} : Number of iterations
i, j : Counters.

Step 1: Initialising :
$i = 0$
$Y_1^{(0)} = Y$
$Y_2^{(0)} = \Pi(Y)$
$Z_1^{(0)} = Z_1$
$Z_2^{(0)} = Z_2$
Step 2:
While $(i < i_{max})$
{
$Y_1^{(i+1)} = Y_1^{(i)} + \Pi^{-1}(W_2^{(i)})$
$Z_1^{(i+1)} = Z_1^{(i)} + W_{Z_1}^{(i)}$
$Y_2^{(i+1)} = Y_2^{(i)} + \Pi(W_1^{(i)})$
$Z_2^{(i+1)} = Z_2^{(i)} + W_{Z_2}^{(i)}$
i=i+1
}
Step 3: Decision
for$(j = 0; j < k \times M; j++)$
$LLR_j = \frac{4E_s}{N_0} \cdot Y_{2_j}^{i_{max}} + W_{2_j}^{i_{max}}$
if$(LLR_j > 0)$
$D_j = 1$
else
$D_j = 0$

5 The simulation results

In this section, we present the simulation results and analysis for some GPCB-OSMLD codes. Transmission over the additive white Gaussian noise (AWGN) channel and binary antipodal modulation is used. We are interested in the information bit error rate (BER) for different signal to noise ratios per information bit (Eb/N0) in dB. We show that there are many parameters which affect the performance of GPCB-OSMLD codes. These parameters are : The number of decoding iterations, the parameters of component codes and interleaver size (see table 3).

Table 3: Simulation parameters

Parameter	**Value**
Component decoder	Threshold
Iterations	1 to 20
Modulation	BPSK
Channel	AWGN
Interleaver pattern	Random interleaver (default value) Diagonal interleaver Cyclic interleaver Block interleaver helical interlever
Interleaver size(N=Mxk)	1xk, 10xk, 100xk, 300xk

5.1 The Turbo effect

In this study, we want to emphasize the effect of number of iterations on the system performance using an Random interleaver for GPCB 0SMLD (30300, 13500) which is obtained from DSC(73,45) code with M=300, and GPCB OSMLD (35500, 19100) which is obtained from DSC(273,191) with M=100. The figures 4 and 5 show that the performances increase with number of iterations. According to these figures we obtain respectively about 2.5dB and 3dB coding gain at 10^{-5} from iteration 1 to 20. So we note that the turbo effect is established for this family of codes.

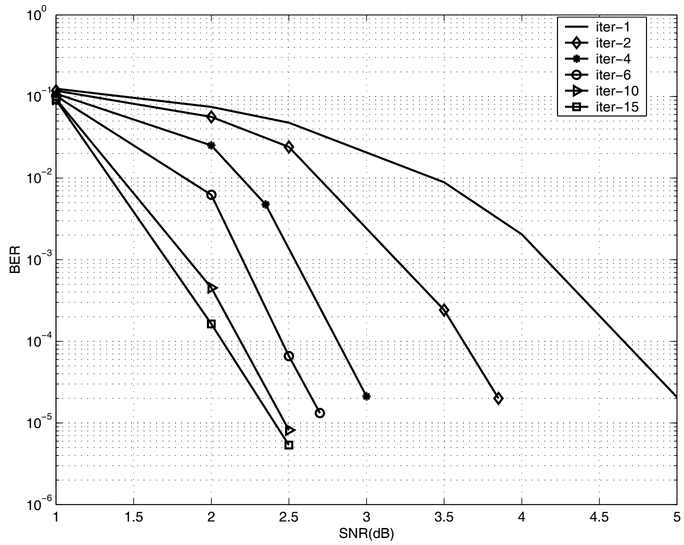


Figure 4: Performance of GPCB-OSMLD(30300,13500) code on AWGN channel

5.2 The effect of block size (parameter M)

Figure 6 shows the BER versus SNR results of the GPCB OSMLD (355,191) code for different values of M (1, 10,100 and 300). By increasing M we obtain about a gain of 2.2dB, the amelioration becomes negligeable when M is greater than 300. Through the figure 7 we observe that there is a compromise between parameter M and the number of iterations, for example for the number iteration equal to 1 or 2, we observe that even if we increase M we don't improve the BER, however when the number of iterations is greater than 3 increasing M improves the performance of our decoder.

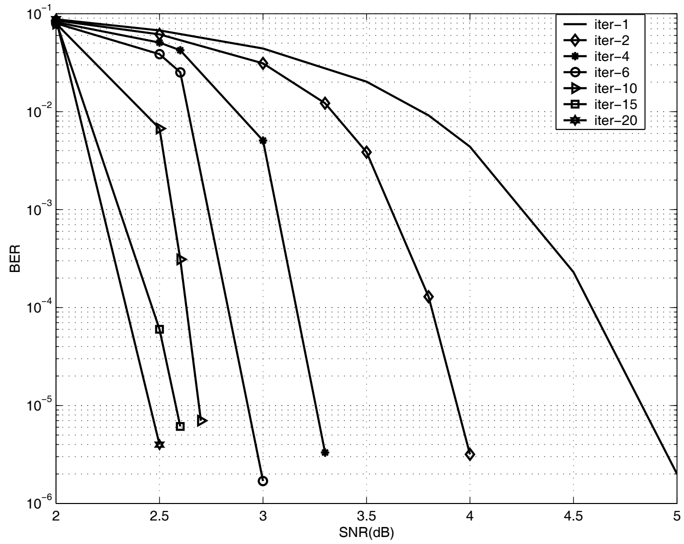


Figure 5: Performance of GPCB-OSMLD(35500,19100) code on AWGN channel

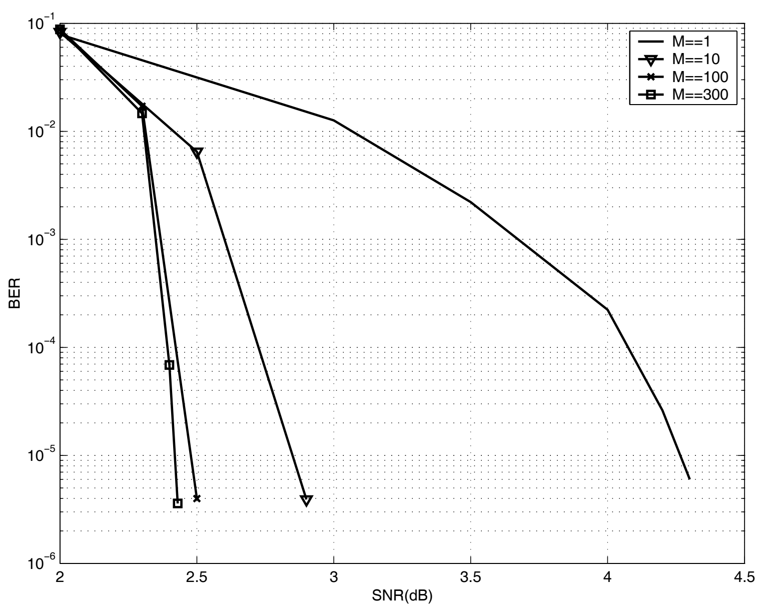


Figure 6: Performance of GPCB-OSMLD(355,191) code for different values of M on AWGN channel

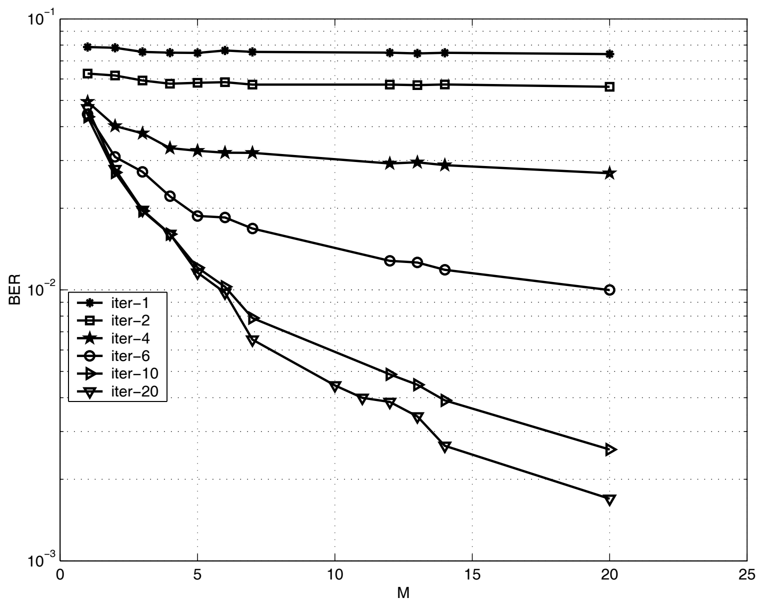


Figure 7: BER of GPCB-OSMLD(101,45) code for different number of iterations versus the parameter M for an SNR=2dB

5.3 The effect of interleaver

To study the influence of the interleaver pattern on the GPCB-OSMLD codes performance, we have evaluated the BER of the GPCB-OSMLD (1111,495) code using different interleavers such as diagonal, cyclic, block, helical and random interleaver. The figure 8 shows the results. We observe that the block and helical are little good than random, diagonal and cyclic ones.

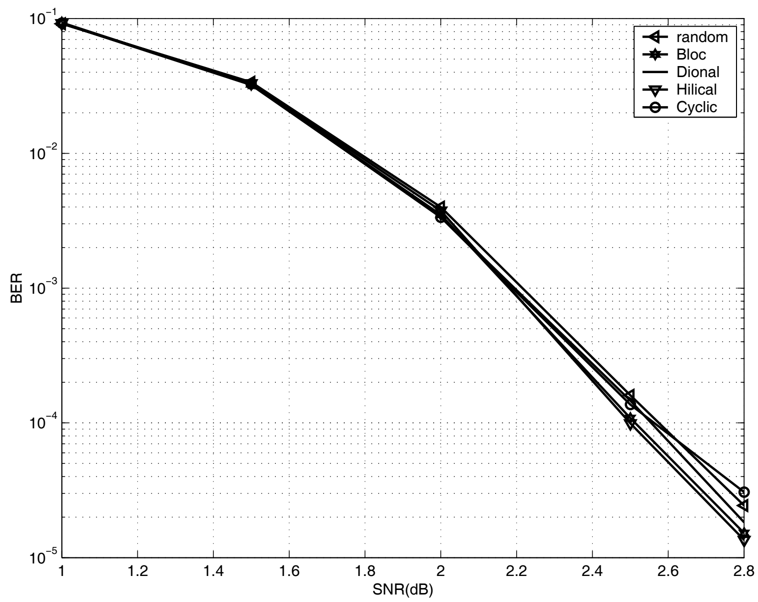


Figure 8: Interleaver structure effect on the performance of a GPCB-OSMLD codes

5.4 Comparisons with other works

In this subsection we present on one hand, the comparison between GPCB-OSMLD decoded by our algorithm and GPCB-BCH decoded by the algorithm proposed by Belkasmi et al [3].
For the rate 0.45 we compare the GPCB-OSMLD (101,45) and GPCB BCH(169,85) codes obtained from Farchan [13], and for the rate 0.53 we compare the GPCB-OSMLD(355,191) and GPCB-BCH(127,78) codes obtained from Farchan [13].
Figure 9 shows that our proposed algorithm is too better in term of BER than that of Belkasmi et al [3] for both rates mentioned above.
In the other hand the comparison between our decoder and the algorithm of Lucas [11], for the compenent code OSMLD (73,45) and M=1 the figure 10 shows that our decoder outperforms the Lucas algorithm for this code.

5.5 Shannon limits

In this subsection we evaluate the performance at 20^{th} iteration of the GPCB-OSMLD (30300, 13500) and GPCB-OSMLD (106500, 57300) codes. Their code rates are, respectively 0.45 and 0.53. Our decoder performances and Shannon limits are shown in figure 11. From this figure, we observe that these codes are respectively 2.4dB and 2dB away from their Shannon limits.

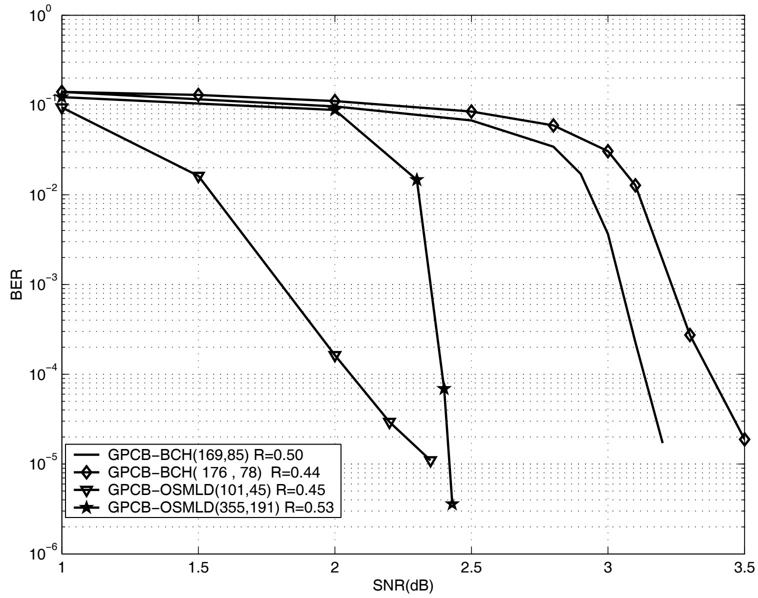


Figure 9: Comparison of two decoders for GPCB-OSMLD and GPCB-BCH

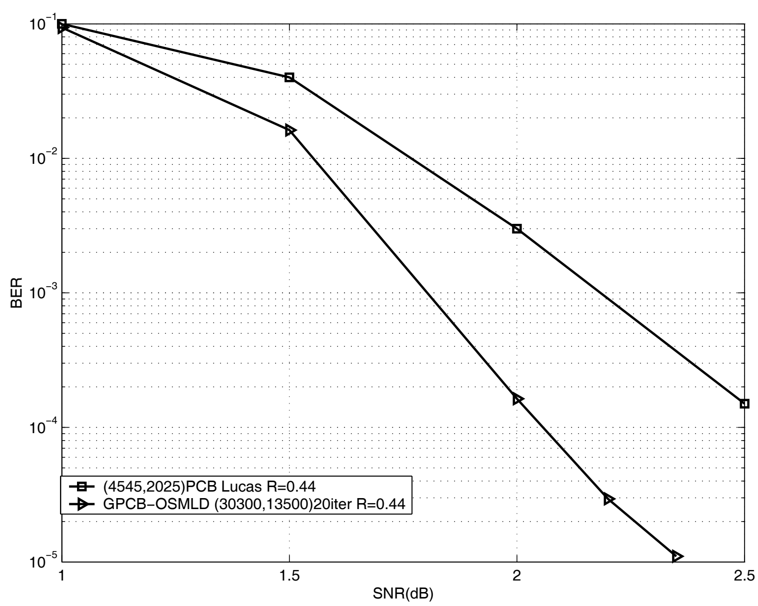


Figure 10: Comparison of our algorithm and Lucas for PCB(4545,2025) code and GPCB-OSMLD(30300,13500) code

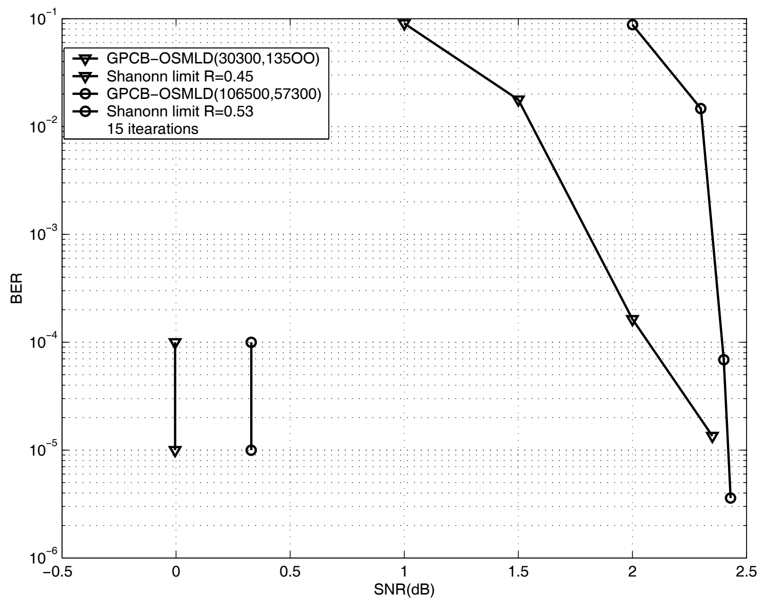


Figure 11: Performance of two GPCB-OSMLD codes and their positions from Shannon limits

6 Conclusion

In this work, we have applied the threshold algorithm as a Soft In Soft Out(SISO) compenent decoder and a Lucas connexion scheme to decode the GPCB-OSMLD codes. For this decoder we have studied the effect of various parameters like component codes, the number of iterations, interleaver size (parameter M) and pattern using simulations. The results show that by increasing the number of iterations and/or the block size (parameter M) we ameliorate the performance of the decoder. We have also demonstrated through simulation that there is a relationship between parameter M and the number of iterations. Finaly by comparing the coding of the codes GPCB-OSMLD (30300, 13500) and GPCB-OSMLD (106500, 57300), we find that these codes are respectively 2.4dB and 2dB away from their Shannon limits.
The obtained results by applying the above construction and decoding on OSMLD family codes look very promising and open new perspectives. The extension of this study is to apply other connection sachems.

References

[1] C. Berrou, A. Glavieux and P. Thitimajshima,Near Shannon limit error-correcting coding and decoding : Turbo-codes,*IEEE Int. Conf. on Comm. ICC'93,* 1993,pp. 1064-1071.

[2] M. Belkasmi, M. Lahmer and M. Benchrifa ,Iterative Threshold Decoding of Parallel Concatenated Block Codes,*Turbo Coding Conf,*Munich,2006.

[3] M.Belkasmi and A.Farchane,Iterative decoding of parallel concatenated block codes,*Proceedings of ICCCE'08,Kuala Lumpur, Malaysia,* May, 2008

[4] J.L Massey,Threshold Decoding,*Cambridge Ma, M.I.T. Press,* 1963

[5] R. Pyndiah,Near-Optimum Decoding of Product Codes: Block Turbo Codes,*IEEE Trans. Commun.*Aug. 1998,pp. 1003-1010

[6] J. Hagenauer, E. Offer, and L. Papke,iterative decoding of binary block and convolutional codes,*IEEE Trans.Inform. Theory.* Mar. 1996, Vol. 42, pp. 429-446.

[7] C. Clark and B. Cain,Error-Correction Coding for digital communications,*Plenum Press,* 1981

[8] C. R. P. Hartman and L. D. Rudolph,An optimum symbol by symbol decoding rule for linear codes,*IEEE Trans. Inform. Theory.*Sept. 1973, Vol. IT-22, pp. 514-517.

[9] J. Nilsson, R Kotter,Iterative decoding of product code constructions,*Proceedings of ISIT-94, Sydney, Australia,* November 1994, pp.1059-1064

[10] S. Benedetto and G. Montorsi,Average Performance of Parallel Concatenated Block Codes,*Electronics Letters, vol. 31, no. 3, pp. 156-158,* February 1995.

[11] R. Lucas, M. Bossert and M. Breitbach,On Iterative Soft-Decision Decoding of Linear Binary Block Codes and Product Codes,*IEEE Journal on selected areas in communications.* February 1998, Vol. 16, N 2, pp. 276-296.

[12] S. Lin and D. J. Costello, Error Control Coding, Fundamentals and Applications, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[13] Previte, A.Farchane, Iterative decoding of parallel concatenated block codes, May, 2008

Received: November, 2009