

Impact of the decoder connection schemes on iterative decoding of GPCB codes

Fouad Ayoub, Mohammed Lahmer, Mostafa Belkasmi, and El Houssine Bouyakhf

Abstract—In this paper we present a study of the impact of connection schemes on the performance of iterative decoding of Generalized Parallel Concatenated block (GPCB) constructed from one step majority logic decodable (OSMLD) codes and we propose a new connection scheme for decoding them. All iterative decoding connection schemes use a soft-input soft-output threshold decoding algorithm as a component decoder. Numerical result for GPCB codes transmitted over Additive White Gaussian Noise (AWGN) channel are provided. It will show that the proposed scheme is better than Hagenauer's scheme and Lucas's scheme [1] and slightly better than the Pyndiah's scheme.

Keywords—Generalized Parallel concatenated block codes, OSMLD codes, threshold decoding, iterative decoding scheme, and performance.

I. INTRODUCTION

THE decoding process of turbo codes is a suboptimal iterative processing in which each component decoder takes advantage of the extrinsic information produced by the other component decoder at the previous step. The way the extrinsic information is conveyed and how it is exploited by the component decoders to make their decision has not been closed yet. The original works in this context are due to Berrou [2] and Robertson [3] for convolutional codes, Pyndiah [4] and Lucas [5] for block codes. Hagenauer [6] gave an extrapolation of Robertson's scheme for block codes by using a trellis decoder. In this work the impact of the connection scheme on the performance of iterative decoding of Generalized Parallel Concatenated block codes (GPCB) [1] constructed from OSMLD codes is considered. These codes have proven to be a very good performance. On the other hand we will use the same component decoder for all schemes namely soft-in soft-out threshold algorithm [7] with a slight modification.

The organization of the paper is as follows. In Section II, we start with a description of the basic concept GPCB codes, and then we describe the soft-in soft-out decoding algorithm in section III. The connection schemes studied in this work are given in Section IV. Section V is dedicated to simulation results and analysis for different GPCB-OSMLD codes. Section VI concludes this paper.

F. Ayoub is with Faculty of Sciences, University Mohamed V Agdal, Rabat, Morocco. ayoubfouadn@gmail.com.

M. Lahmer is with High School of Technology, Moulay Ismail University, Meknes, Morocco. lahmer@est-umi.ac.ma.

M. Belkasmi is with National School of Computer Science and Systems Analysis, Rabat, Morocco. belkasmi@ensias.ma.

E. H. Bouyakhf is with Faculty of Sciences, University Mohamed V Agdal, Rabat, Morocco. bouyakhf@fsr.ac.ma.

II. GPCB CODES

A. One-step majority-logic decodable codes

One step majority logic decodable codes are based on orthogonal parity check sums for each bit. We relied on the work of [8] to develop algorithm for constructing various codes shown in table I. In this table we present some examples, using the abbreviation DSC for Difference Set Cyclic codes, EG for Euclidean Geometry codes and BCH for Bose Chaudhuri and Hocquenghem codes.

TABLE I
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

B. Structure of GPCB codes

The structure of generalized parallel concatenated block codes is shown in figure 1; it 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 II gives some examples of codes based on this construction where the two components codes are the same, and for different values for M.

TABLE II
SOME EXAMPLES OF GPCB CODES.

Component code	M	GPCB-OSMLD code	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

C. The interleaver

The function of the interleaver is to take each incoming block of bits and rearrange them in a pseudo-random fashion prior to encoding by second encoder. In our previous works [1] several interleaving techniques were invoked such as random, block, diagonal, helical, and cyclic interleaver, and simulation results have shown that the effect of the interleaver type is the same; that is why we used in this work the random interleaver type for all simulation results.

III. THE SISO THRESHOLD DECODING

Threshold decoding is simply the logical extension of soft decisions of majority decoding described above [11]. Let us consider a transmission of block coded binary symbols $\{0, 1\}$ using a BPSK modulation over AWGN channel, the soft output for the j^{th} bit position of a given soft input $R(r_1, \dots, r_n)$ is defined as :

$$LLR_j = \ln \frac{P(c_j = 1/R)}{P(c_j = 0/R)} \quad j \in \{1, \dots, n\} \quad (1)$$

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

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

where $B_i, i \in \{0, \dots, J\}$ 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 $\{1, \dots, J\}$ the term B_i is computed by using the i^{th} orthogonal parity equation. By applying BAYES rule, (2) becomes

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

Since the parity check equations are orthogonal on the j^{th} symbol the individual probabilities are all independent and (3) can be rewritten as :

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

(4) can be written as:

$$LLR_j \simeq \underbrace{\left(\sum_{i=1}^J \ln \frac{P(\{B_i\}/c_j = 1)}{P(\{B_i\}/c_j = 0)} \right)}_{\text{extrinsic}} + \underbrace{\left(\ln \frac{P(\{B_0\}/c_j = 1)}{P(\{B_0\}/c_j = 0)} \right)}_{\text{channel}} + \underbrace{\left(\ln \frac{P(c_j = 1)}{P(c_j = 0)} \right)}_{\text{apriori}} \quad (5)$$

According to [12], (5) can be expressed as

$$LLR_j \simeq \underbrace{\left(\sum_{i=1}^J (1 - 2B_i) \cdot w_i \right)}_{\text{extrinsic}} + \underbrace{\left((1 - 2B_0) \cdot w_0 \right)}_{\text{channel}} + \underbrace{\left(\ln \frac{P(c_j = 1)}{P(c_j = 0)} \right)}_{\text{apriori}} \quad (6)$$

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:

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

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

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

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

$$L_{ik} = \frac{4E_s}{N_0} \cdot |r_{ik}| \quad (9)$$

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

$$LLR_j = \frac{4E_s}{N_0} \cdot r_j + W_j + \ln \frac{P(c_j = 1)}{P(c_j = 0)} \quad (10)$$

In each of connection scheme, the calcul LLR will involve each summand with different weighting.

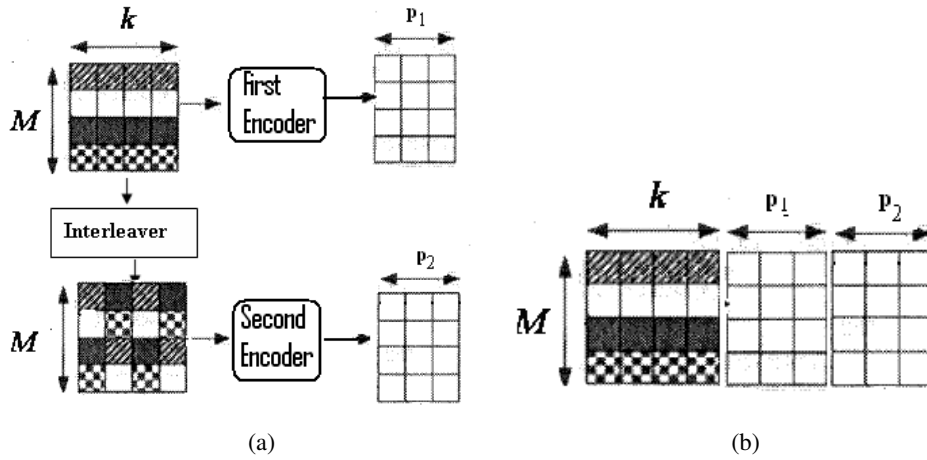


Fig. 1. Encoder structure of generalized parallel concatenated block codes:(a) Structure of encoder; (b) Transmitted codeword.

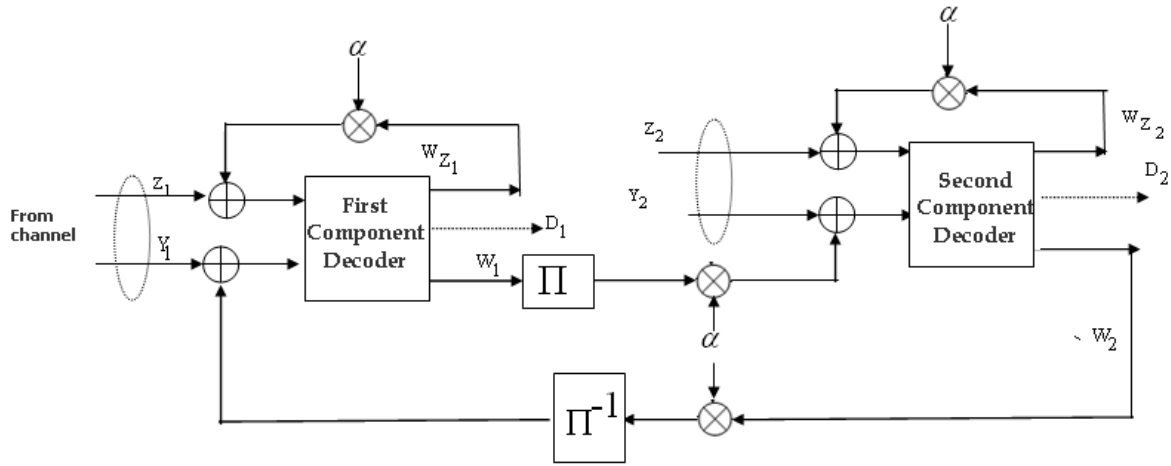


Fig. 2. Pyndiah's iterative decoding scheme.

IV. THE CONNECTION SCHEMES OF ITERATIVE DECODING OF GPCB CODES

Surprisingly the parallel concatenation of bloc codes can be decoded by offers a variety of iterative decoding schemes. In fact, the existence of an explicit collaboration between two component decoders, by exchange of extrinsic information, gives a great opportunity to improve decoding performance during iterations.

In this section, we present the different connection schemes for decoding GPCB codes,ie, Pyndiah's connection scheme, Hagenauer's scheme, and the proposed scheme.

We use the following notations for all connection schemes :

R : Received word, it consist of three parts $[Y, Z_1, Z_2]$:

Y : The systematic information.

Z_1 : The parity check information generated by first encoder.

Z_2 : The parity check information generated by second

encoder.

$Y_1 = Y$: 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 .

L_{Y_1} : The a priori information for Y_1 .

$L_c = \frac{4E_s}{N_0}$: The reliability value of the channel.

L_{Y_2} : The a priori information for Y_2 .

W_{Z_2} : The extrinsic information generated by the second decoder for Z_2 .

D : The hard decision(decoded word).

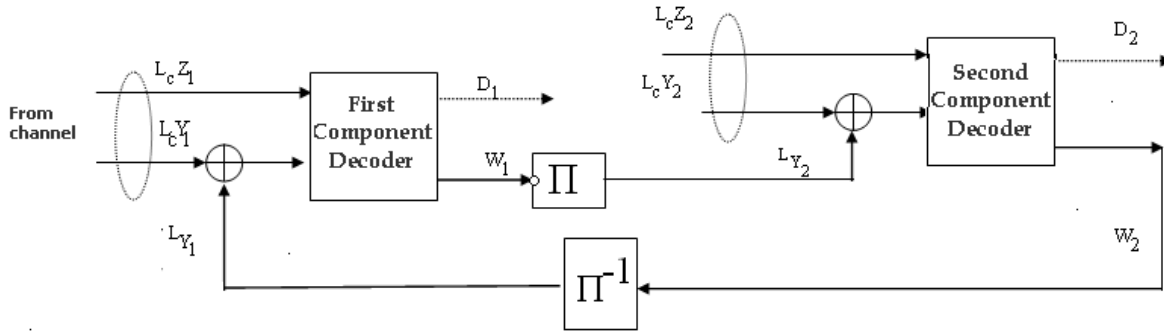


Fig. 3. Hagenauer's iterative decoding scheme.

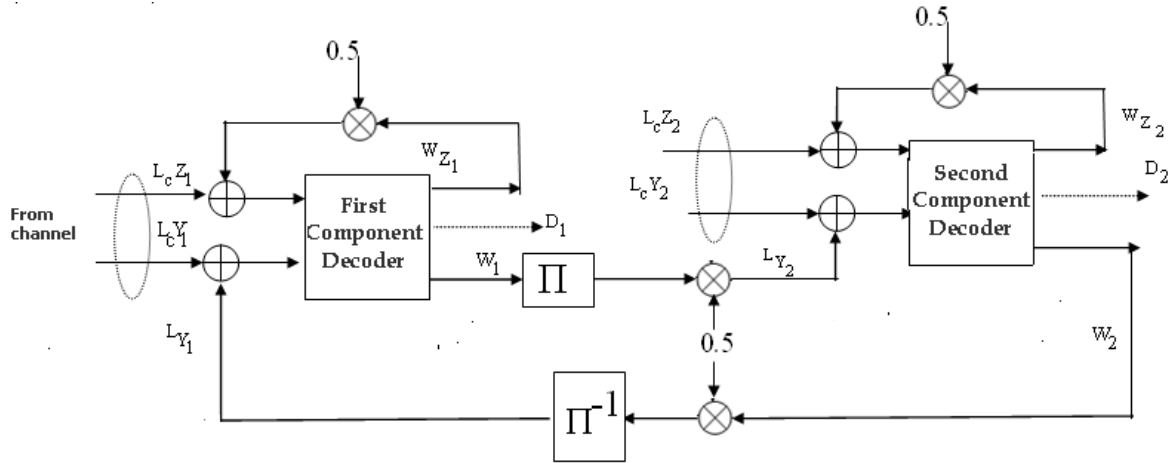


Fig. 4. The proposed iterative decoding scheme.

TABLE III
THE VALUES OF α FOR SOME GPCB-OSMLD CODES.

GPCB-OSMLD code	value of α
(101,45)	0.08
(1010,450)	0.07
(10100,4500)	0.08
(30300,13500)	0.08
(3550,1910)	0.07

A. Pyndiah's connection scheme

The iterative decoding process presented here is introduced in [4] to decode product codes. In this work, we use the same principle of Pyndiah's connection scheme to decode GPCB-OSMLD codes using soft-input soft-output threshold algorithm (see figure 2). Recall that the LLR is computed by Pyndiah et al. [7] as follow :

$$LLR = \begin{cases} L_c Y_2 + \alpha W_2 & \text{For systematic bits} \\ L_c Z_2 + \alpha W_{Z_2} & \text{For parity bits.} \end{cases}$$

Where α is a scaling factor, it is constant for all iterations, depending to GPCB-OSMLD code, whereas Pyndiah uses a non constant value. The table III resumes the values of α for each code. These values were obtained by an optimization procedure by using simulations.

B. Hagenauer's connection scheme

In the Hagenauer's connection scheme [6], the iterative decoding process can be described as follows : in the first iteration, each decoder computes the extrinsic information using the observation of the associated systematic and parity

symbols coming from the transmission channel and the a priori information L_{Y_1} and L_{Y_2} , since no a priori are available from the decoding process at the beginning of the iterations. For the subsequent iterations, the extrinsic information coming from the other decoder are used as a priori information for the current decoder (see figure 3). The decision can be computed by the second decoder as follows :

$$LLR = \begin{cases} L_c Y_2 + W_2 + L_{y_2} & \text{For systematic bits} \\ L_c Z_2 & \text{For parity bits.} \end{cases}$$

C. Proposed connection scheme

In the proposed scheme, we have introduced two important modifications in the Hagenauer's scheme as shown in figure 4 :

1) In contrary to Hagenauer's scheme which calculates and updates extrinsic information only for systematic bits, the proposed scheme calculates and updates extrinsic information for systematic and redundancy bits.

2) In each entry of elementary decoder, the extrinsic information is multiplied by a constant value equal to 0.5 in iterations based on these two modifications, the decision can be computed by the second decoder as follows :

$$LLR = \begin{cases} L_c Y_2 + W_2 + 0.5 L_{y_2} & \text{For systematic bits} \\ L_c Z_2 + 0.5 W_{Z_2} & \text{For parity bits.} \end{cases}$$

V. THE SIMULATION RESULTS

In this section, we present simulation results and analysis for some GPCB-OSMLD codes. Transmission over the additive white Gaussian noise (AWGN) channel and BPSK modulation are used. We would like to notify that for all simulations we have used a minimal residual error number of 200 and the maximum number of iterations was set at 20 (where there is no significantly more gain for more than 20 iteration). We present a comparison between the three decoding schemes and results of [1] in terms of BER.

A. Performance of the proposed scheme

1) *The turbo effect:* Figures 5 and 6 show that the performances increase with number of iterations. According to these figures we can see that the improvement is great after the first iterations. So, we note that the turbo effect of the proposed scheme is established for this family of codes.

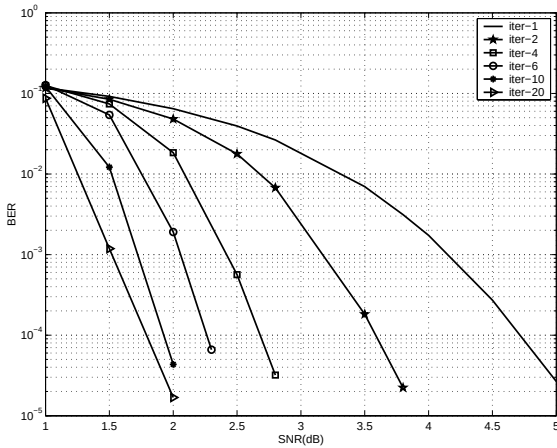


Fig. 5. Turbo effect of the proposed scheme for GPCB-OSMLD(10100,4500) code.

2) *The effect of M:* In figure 7, we present the simulation results for the GPCB OSMLD (101,45) code for different values of M (1, 10 and 100). By increasing M we obtain about a gain of 3.0dB at 10^{-5} , the amelioration becomes negligible when M is greater than 100.

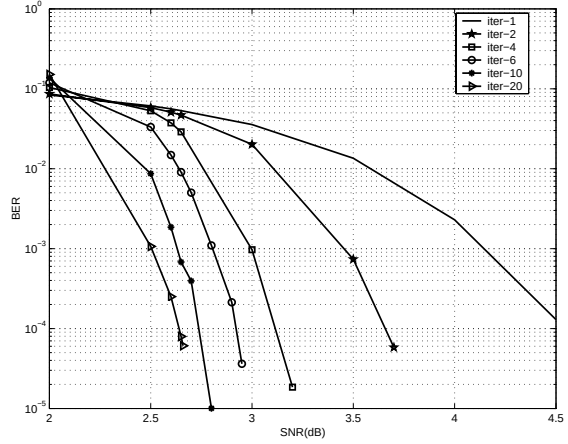


Fig. 6. Turbo effect of the proposed scheme for GPCB-OSMLD(3550,1910) code.

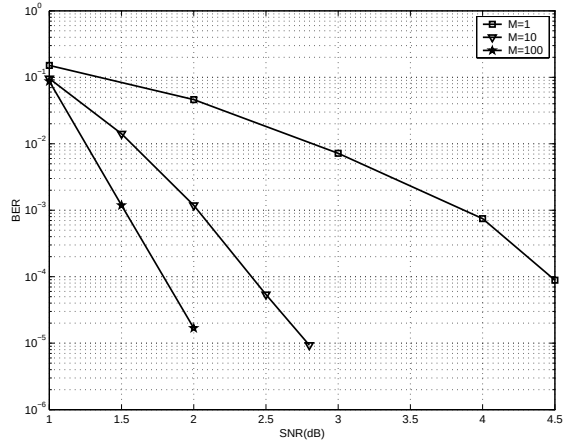


Fig. 7. Effect of M in proposed for GPCB-OSMLD(101,45) code.

B. Performance of the Pyndiah's scheme

In this subsection we present the simulation results of the decoding iterative of the GPCB-OSMLD codes using the Pyndiah's scheme that presented in section IV-A. In figure 8 we show the improvement of performances during iterations, so the turbo effect is confirmed by Pyndiah's scheme. Figure 9 shows that the effect of M provides about 3.0dB gain passing from M=1 to M=100.

C. Performance Comparisons

In this subsection we present the comparison between GPCB-OSMLD decoded by the proposed scheme and the tree connection schemes: Hagenauer's scheme, Lucas's scheme[1] and Pyndiah's scheme. The figure 10 shows that the proposed scheme outperforms the Hagenauer's scheme for the code GPCB-OSMLD(10100,4500) at iteration 20 by about 1.5dB at 10^{-4} . In Figure 11, we present a comparison between the four schemes for GPCB-OSMLD (3550, 1910) code at 20 iterations. Figure 12 confirms that the proposed scheme provides performances better than performances of Lucas's

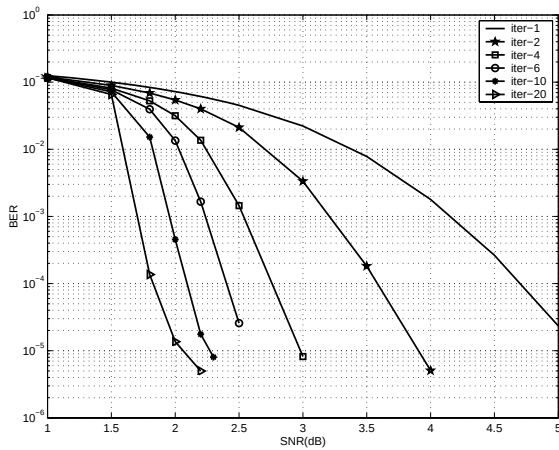


Fig. 8. Turbo effect of the Pyndiah's scheme for GPCB(10100,4500) code.

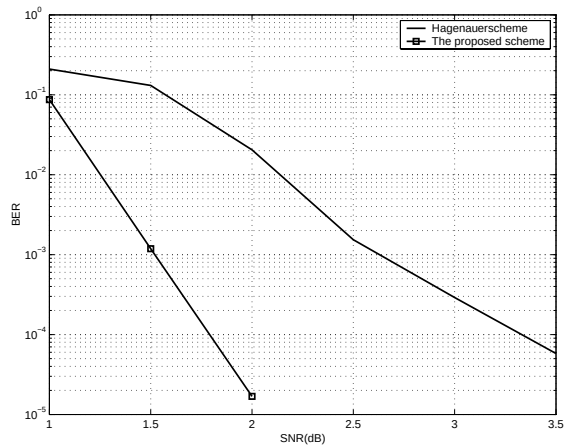


Fig. 10. Comparison of the proposed scheme and Hagenauer's scheme for GPCB(10100,4500) code.

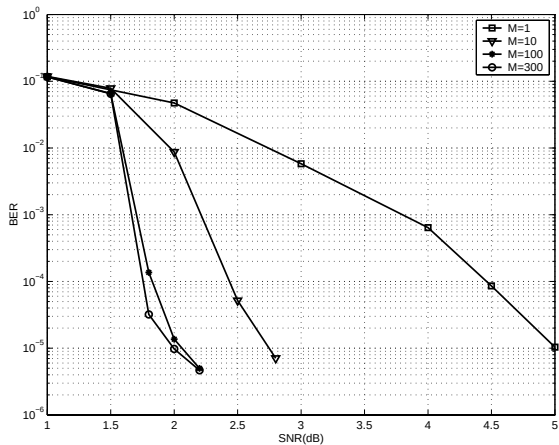


Fig. 9. Effect of M in the Pyndiah's scheme for GPCB-OSMLD(101,45) code.

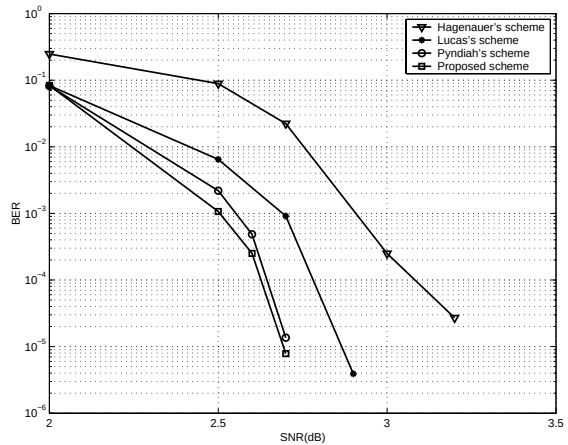


Fig. 11. Comparison of the four schemes for GPCB (3550, 1910) code at 20 iterations.

scheme published in [6] and slightly better than the Pyndiah's scheme for GPCB-OSMLD codes, from this figure, we also observe that the GPCB-OSMLD(10100,4500) code is 2dB away from its Shannon limit.

VI. CONCLUSION

In this paper, we have presented the impact of the connection schemes on iterative decoding of generalized parallel concatenated codes base on one-step majority logic decodable codes. For all connection schemes we have studied the effect of various parameters like component code, the number of iterations, interleaver size(parameter M) and interleaver pattern using simulations. The simulation result shows that the proposed scheme is better than Hagenauer's scheme and Lucas's scheme [1] and slightly better than the Pyndiah's scheme. We conclude that the impact of connection scheme iterative decoding of GPCB-OMLD codes is an important feature. These results open new perspectives namely, the application of such connection schemes on the product code and serial concatenated block codes.

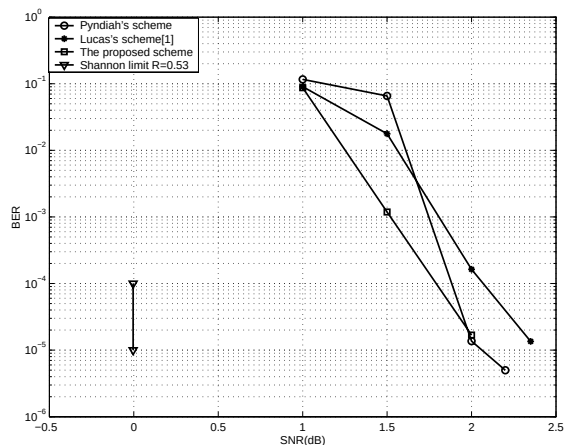


Fig. 12. Performance of connection schemes for GPCB-OSMLD(10100, 4500) code and their positions from Shannon limit.

REFERENCES

- [1] F. Ayoub, M. Belkasm and I. Channa, Iterative Decoding of Generalized

- Parallel Concatenated OSMLD Codes, *Applied Mathematical Sciences journal*, Vol.4, no. 41, pp. 2021-2038, 2010
- [2] 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.
 - [3] P. Robertson, Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes, in Proc. IEEE Global Commun. Conf. (GLOBECOM94), San Francisco, CA, Dec. 1994, pp. 1298-1303.
 - [4] R. Pyndiah, Near-Optimum Decoding of Product Codes: Block Turbo Codes, *IEEE Trans. Commun.*, Aug. 1998, Vol. 46, N 8, pp. 1003-1010.
 - [5] R. Lucas, M. Bossert and M. Breitbart, 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.
 - [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] M. Belkasm, M. Lahmer and M. Benchrifa, Iterative Threshold Decoding of Parallel Concatenated Block Codes. Turbo Coding Conf, Munich, 2006.
 - [8] Martin Tomlinson, Cen Jung Tjhai, Marcel A. Ambroze and Mohammed Z. Ahmed, Binary Cyclic Difference Set Codes Derived from Idempotents Based on Cyclotomic Cosets, *submitted to IEEE Transactions on information theory*, May 2004
 - [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] J.L. Massey, Threshold Decoding, *Cambridge Ma, M.I.T. Press*, 1963
 - [12] C. Clark and B. Cain, Error-Correction Coding for digital communications, *Plenum Press*, 1981
 - [13] Yuri V. Svirid and Sven Riedel, Threshold Decoding of Turbo-Codes, *IEEE Int. Symposium on Information Theory*, 1995, pp. 39.
 - [14] S. Lin and D. J. Costello, Error Control Coding, Fundamentals and Applications, Englewood Cliffs, NJ: Prentice-Hall, 1983.