High performance short-block binary regular LDPC codes

Latifa Mostari, Abdelmalik Taleb-Ahmed

Alexandria Engineering Journal (2018) 57, 2633–2639

HOSTED BY

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

ORIGINAL ARTICLE

High performance short-block binary regular LDPC codes

Latifa Mostari[a,*], Abdelmalik Taleb-Ahmed[b]

[a] *Department of Electronics, Faculty of Technology, Hassiba Benbouali University of Chlef, Algeria*
[b] *LAMIH UMR CNRS 8201, University of Valenciennes and Hainaut Combrésis, Valenciennes, France*

Received 31 December 2016; revised 21 September 2017; accepted 24 September 2017
Available online 15 November 2018

KEYWORDS

LDPC code;
Parallel concatenation;
Iterative decoding;
Sum-product algorithm;
Min-sum algorithm

Abstract LDPC code shows a good performance with long-block codes. However, certain channels are constrained to use short-block codes due to latency. Therefore, concatenated LDPC codes with iterative decoding is a good choice to get a good performance. Concatenated binary LDPC codes were introduced as a class of concatenated codes in which the LDPC codes are irregular codes having different parameters. Although irregular LDPC codes are more efficient than regular codes, irregular LDPC codes have an error floor and a higher encoding complexity than regular code. In this paper, in order to get a good performance/complexity trade-off with a short-block code, we investigate a parallel concatenation of two identical regular binary LDPC codes, using an interleaver. Simulation results show that the proposed code outperforms a single LDPC code. The proposed code needs less time decoding delay than a single LDPC code with the same iteration number. Also, the proposed code needs a less number of iterations to achieve the same performance of a single LDPC code that leads to less decoding complexity and decoding delay. A simplified algorithm, the min-sum algorithm, that is used to decode the component codes shows a small performance loss with respect to the sum-product algorithm.

1. Introduction

In 1948, Shannon [1] proves that there is a limit spectral efficiency that we cannot overcome if we want a transmission without errors. Shannon was able to give a limit, without giving the code that allows the correction of errors.

In 1993, a publication of Claude Berrou et al. presented an invention of a new error correcting code called turbo-code [2]. This code provides communications substantially without errors and obtains a better coding gain of all other existing codes. The turbo-code consists of two or several codes concatenated in serial, parallel or hybrid with short-block codes, separated by interleavers. Their decoding is done iteratively between the component codes which based on the Soft-Input Soft-Output decoding.

After the invention of the turbo-codes and grace to iterative technique used in their decoding. The binary LDPC codes, which were ignored during long year since their introduction by Gallager [3] in 1962, were rediscovered by Mackay [4] in 1995 to approach Shannon capacity.

[*] Corresponding author.
E-mail address: l.mostari@univ-chlef.dz (L. Mostari).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2017.09.016
1110-0168 © 2018 Faculty of Engineering, Alexandria University. Production and hosting by Elsevier B.V.

LDPC codes are block codes with parity-check matrices H that contain only a very small number of non-zero entries. This sparseness of H is essential for an iterative decoding complexity that increases only linearly with the code length. LDPC codes are decoded iteratively using a graphical representation of their parity-check matrix and so are much longer, less structured, and designed with the properties of H as a focus [5]. A significant contribution was introduced by Luby et al. [6], in 1997, on irregular LDPC codes, where the non-zero elements of the parity check matrix are distributed irregularly.

Although LDPC codes are good errors correcting codes, code concatenation of theses codes with iterative decoding is still attractive to construct powerful error correction codes with reasonable complexity. Concatenated binary LDPC codes were introduced as a class of concatenated codes in which the LDPC codes are irregular codes having different parameters interact in parallel [7,8] or in serial [9,10] with or without interleavers.

Although irregular LDPC codes are more efficient than regular codes, irregular LDPC codes have an error floor and a higher encoding complexity than regular code. In this work, we study a concatenation of two identical regular LDPC codes arranged in parallel, using an interleaver between the two LDPC codes that compose it. We restrict our description of proposed code to rate 1/3 codes constructed by combining two rate 1/2 binary LDPC codes. The two component codes are the same, decoded by Sum-Product Algorithm (SPA) and its simplified algorithm: Min-Sum Algorithm (MSA). An interleaver was used between the two codes to spread out any burst errors.

This paper is organized as follows: in Section 2, we define the LDPC codes. Sections 3 and 4 present respectively LDPC encoding and LDPC decoding algorithms such as: SPA and MSA. In Section 5, the parallel concatenated LDPC encoding and decoding are investigated. Section 6 presents simulation results of a rate 1/3 parallel concatenated binary LDPC code compared with a single LDPC code of the same rate and block-length. Finally, in Section 7 we conclude our findings.

2. LDPC codes

LDPC codes are linear block codes, based on low-density parity check matrices H, i.e the number of non-zero elements in the matrix is much less than the number of zeros. The code rate R of an LDPC code is given by:

$$R = \frac{N - M}{N} \tag{1}$$

where N and M are respectively the number of columns and rows of H. The number of columns N defines the code length. The number of rows M determines the number of parity check equations.

According to the regular or irregular distribution of the non-zero elements in the matrix, the LDPC codes can be regular [11,12] or irregular. The parity check matrix of a regular LDPC code has constant column and row weights. Irregularity of an irregular LDPC code results in the non-uniform distribution of non-zero elements on columns and/or rows.

An example of a parity check matrix H of size 4×6 of a regular LDPC code is given by:

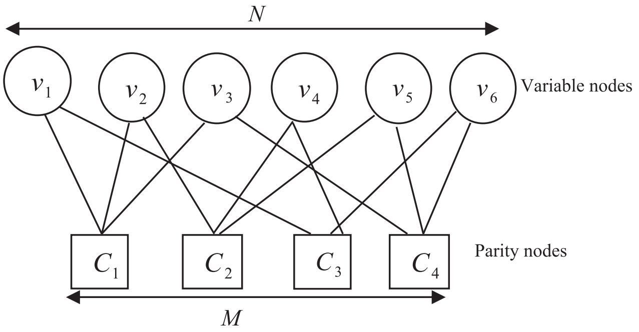

Fig. 1 Tanner Graph corresponds to the matrix H of the Eq. (2).

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2}$$

An example of a parity check matrix H of size 4×8 of an irregular LDPC code is given by:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{3}$$

The LDPC codes can be described by a graphical representation, called Tanner graph [13], corresponds to the matrix H.

Tanner graph is a bipartite graph composed of two types of nodes: the variable nodes representing the symbols of the codeword and the parity nodes representing the parity control equations. These two types of nodes are connected by branches according to the non-zero elements in the parity check matrix.

The number of the variable nodes $N_m(m \in \{1, \cdots, N\})$ and the parity nodes $M_n(n \in \{1, \cdots, M\})$ correspond respectively to the number of columns N and the number of rows M of the matrix H.

Fig. 1 shows an example of a Tanner graph corresponding to the matrix H of the Eq. (2).

The parity check matrix H allowed us to determine the Tanner graph which is used as a support for the decoder. Also, this matrix is used for the LDPC encoder.

3. LDPC encoding

The encoding is doing by several methods. In this work, we use the encoding methods by decomposition LU. This coding type is systematic. This means that the codeword is in the form of a concatenation of an information word C_I, of size N-M symbols, and of a redundancy block C_R:

$$C = [C_R \quad C_I] \tag{4}$$

Codewords should verify the following equation:

$$C.H' = 0 \tag{5}$$

The encoding operation uses this equation. One decompose H into two sub-matrices H_1 and H_2 such that H is the concatenation of H_1 and H_2.

$$H = [H_1 \quad H_2] \tag{6}$$

where H_1 is a square matrix of size $M \times M$ and occupies the first M columns of H. H_2 is a matrix of size $M \times (N - M)$ and occupies the last N-M columns.

Applying Eqs. (4) and (5) in relation (6), one obtain:

$$C_R.H_1' + C_I.H_2' = 0 \tag{7}$$

C_I is already known, it is the information block. Remain to calculate C_R by the following operation:

$$C_R' = inv(H_1).C_I.H_2' \tag{8}$$

With $inv(H_1')$ is the inverse of H_1'. To avoid the computational complexity of $inv(H_1')$, we write H_1 in the form:

$$H_1 = L.U \tag{9}$$

With L and U are respectively lower triangular matrix and upper triangular matrix. This is the LU decomposition. Therefore, Eq. (9) becomes:

$$\begin{aligned} C_R.L'.U' + C_I.H_2' = 0 &\Rightarrow C_R.L'.U' = C_I.H_2' \\ &\Rightarrow L'\underbrace{(C_R.U')}_{Y} = \underbrace{C_I.H_2'}_{Z} \\ &\Rightarrow L'.Y = Z \\ &\Rightarrow Y = Z/L' \end{aligned} \tag{10}$$

Therefore, the redundancy block is:

$$C_R = Y/U' \tag{11}$$

Example 1. The LDPC encoder encodes the information block of length N-M = 5, $C_I = [1 \ 0 \ 1 \ 0 \ 1]$, as follows:

Let us take the following parity check matrix:

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{12}$$

The decomposition of H gives us:

$$H_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}, H_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 0 \end{pmatrix} \tag{13}$$

The factorization LU of H_1 gives us:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}, U = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

Using the Eqs. (10) and (11), the redundancy block C_R is calculated as follows:

$$\begin{aligned} Z = C_I.H_2' = [0 \ 0 \ 1 \ 1 \ 1], &\quad \mathrm{Y = Z/L'} \\ = [0 \ 0 \ 1 \ 1 \ 1], &\quad C_R = Y/U' = [0 \ 0 \ 0 \ 1 \ 0] \end{aligned} \tag{15}$$

Therefore, the codeword is:

$$C = [C_R \ \ C_I] = [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1] \tag{16}$$

4. LDPC decoding

LDPC decoders uses the Tanner graph as a support and their operations can be explained by the passage of messages along the branches of a Tanner graph. The decoding of the LDPC code is an iterative decoding. LDPC decoders performs as follows:

a. Initialization of the variable nodes by the input messages of the decoder.
b. At each iteration the following steps are repeated:
   1. Each parity node receives messages arriving from the variable nodes connected to it by branches and then calculates and sends the resultant message that is related to all messages arriving from the variable nodes except the message where the output message will be sent;
   2. Each variable node receives messages arriving from the parity nodes connected to it by the branches and then calculates and sends the resultant message that is linked to all messages arriving from the parity nodes except the message where the output message will be sent;
   3. Then, the a posteriori information associated with each variable node is calculated before the decision is taken.

Finally, after a certain number of iterations or in the case where the syndrome is zero (*i.e. estimated codeword.*$H' = 0$), the algorithm stops.

The first iterative decoding algorithm of the LDPC codes is the Sum-Product Algorithm (SPA), also known as the belief propagation algorithm, is an optimal iterative decoding algorithm, but with high computational complexity. Several algorithms have been proposed to reduce the complexity of the SPA. In the following, the SPA and its simplified algorithm, Min-Sum algorithm (MSA), are described.

4.1. Sum-product algorithm

The SPA performs the following operations [14]:

■ Initialization of variable nodes

$$\gamma_n(x) = log\frac{Pr(v_n = 1|c_n')}{Pr(v_n = 0|c_n')}, \quad m\epsilon\{1, \cdots, M\}, \ n\epsilon\{1, \cdots, N\} \tag{17}$$

$$\mu_{mn}(x) = \gamma_n(x) \tag{18}$$

– Iteration
■ Calculation of parity nodes

$$\beta_{mn} = 2 \times \tanh^{-1}\left(\prod_{n' \in N_m/n} \tanh(\mu_{mn'}/2)\right) \tag{19}$$

- Calculation of variable nodes

$$\mu_{mn} = \gamma_n + \sum_{m' \in M_n/m} \beta_{m'n} \tag{20}$$

- A posteriori information

$$\tilde{\gamma} = \gamma + \sum_{m \in M_n} \beta_{mn} \tag{21}$$

- Decision

$$\hat{c}_n = \arg\max_x \left(\tilde{\gamma}_n\right) \tag{22}$$

Finally, the algorithm stops if the maximum number of iterations is reached or if the syndrome is zero.

The SPA achieves the best performance for LDPC codes. But the operations of $tanh^{-1}$ and *tanh* are very complex to realize. Otherwise, the MSA makes approximations to simplify parity node calculations.

4.2. Min-Sum algorithm

The MSA performs the following operations [15]:

- Initialization of variable nodes

$$\gamma_n(x) = log\frac{Pr\left(v_n = 1|c_n'\right)}{Pr\left(v_n = 0|c_n'\right)}, \quad m\epsilon\{1,\cdots,M\}, \quad n\epsilon\{1,\cdots,N\} \tag{23}$$

$$\mu_{mn}(x) = \gamma_n \tag{24}$$

$$\mu_{mn} = \mu^*_{mn} \times \mu^{**}_{mn}, \quad \text{where} \quad \mu^*_{mn} = sign(\gamma_n), \mu^{**}_{mn} = |\gamma_n| \tag{25}$$

– Itération

- Calculation of parity nodes

$$\beta_{mn} = \prod_{n' \in N_m/n} \mu^*_{mn'} \times \min_{n' \in N_m/n} \left(\mu^{**}_{mn'}\right) \tag{26}$$

- Calculation of varialble nodes

$$\mu_{mn} = \gamma_n + \sum_{m' \in M_n/m} \beta_{m'n} \tag{27}$$

- A posteriori information

$$\tilde{\gamma} = \gamma + \sum_{m \in M_n} \beta_{mn} \tag{28}$$

- Decision

$$\hat{c}_n = \arg\max_x \left(\tilde{\gamma}_n\right) \tag{29}$$

Finally, the algorithm stops if the maximum number of iterations is reached or if the syndrome is zero.

5. Parallel concatenation of LDPC codes

5.1. Encoding

Fig. 2 represents the block diagram of a rate 1/3 parallel concatenation encoder. It is built using a parallel concatenation of two systematic encoders separated by an interleaver noted π. The two systematic encoders are often referred to as component encoders. The global encoder rate is 1/3; that is, for every input block, the encoder produces three code blocks. One is the interleaved information block, and the other two are the parity blocks generated by the two systematic binary LDPC encoders.

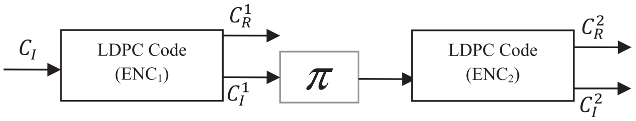

Fig. 2 Structure of a rate 1/3 parallel concatenation LDPC encoder.

The first component encoder ENC_1 encodes the information block

C_I of size *N-M*, $C_I = [C_1 C_2 \cdots C_{N-M}]$, using a parity check matrix H of size $M \times N$, and generates the coded information block of size N:

$$\left[C_R^1 C_I^1\right] = \left[C_{R1}^1 C_{R2}^1 \cdots C_{RM}^1 C_1^1 C_2^1 \cdots C_{N-M}^1\right] \tag{30}$$

where C_I^1 is a systematic block $C_I^1 = C_I$, and C_R^1 is a parity block.

The second encoder ENC_2 uses only the interleaved systematic block $C^1_{I.interleaved}$, and generates the coded block of size N:

$$\left[C_R^2 C_I^2\right] = \left[C_{R1}^2 C_{R2}^2 \cdots C_{RM}^2 C_1^2 C_2^2 \cdots C_{N-M}^2\right] \tag{31}$$

where C_I^2 is an interleaved systematic block $C_I^2 = C^1_{I.interleaved} = C_{I.interleaved}$, and C_R^2 is a parity block.

Thus, the global encoder encodes the information block C_I of size *N-M*, $C_I = [C_1 C_2 \cdots C_{N-M}]$, and generates the coded information block of size N:

$$\left[C_R^1 C_R^2 C_I^2\right] = \left[C_{R1}^1 C_{R2}^1 \cdots C_{RM}^1 C_{R1}^2 C_{R2}^2 \cdots C_{RM}^2 C_1^2 C_2^2 \cdots C_{N-M}^2\right] \tag{32}$$

The interleaver in the global encoder plays an important role. It permutes the data block sent to the second component encoder so as to generate a second parity that is independent from the parity generated by the first component encoder. It is important for the interleaver in turbo coding to have a random property and make the interleaver block as independent as possible from the original block [16].

Example 2. Let us take the same parity check matrix H and the information block $C_I = [1\ \ 0\ \ 1\ \ 0\ \ 1]$ given in the example 1. The global encoder, illustrated in Fig. 2, encodes C_I as follows:

The encoder ENC_1 encodes the information block $C_I = [1\ \ 0\ \ 1\ \ 0\ \ 1]$ and generates the following coded information block:

$$C_1 = \left[C_R^1\ \ C_I^1\right] = [0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1] \tag{33}$$

The second encoder ENC_2 uses the interleaved block, and generates the coded block. If the interleaver permutes the information block to $[0\ \ 0\ \ 1\ \ 1\ \ 1]$, the encoder ENC_2 generates the following coded block

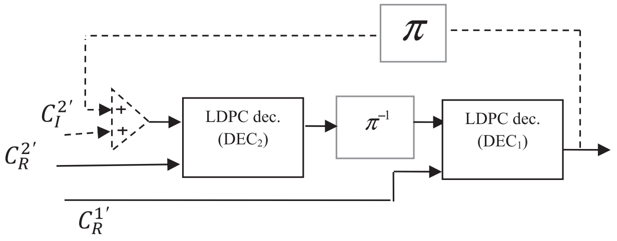

Fig. 3 Concatenated decoders.

$$C_2 = \left[C_R^2\ C_I^2\right] = [1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 1] \tag{34}$$

Thus, the global encoder encodes the information block $C_I = [1\ 0\ 1\ 0\ 1]$, and generates the followning coded information block:

$$C = \left[C_R^1\ C_R^2\ C_I^2\right] = [0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 1] \tag{35}$$

5.2. Decoding

Concatenated decoders [17] presented at Fig. 3, consists of two decoders DEC$_1$ and DEC$_2$ associated respectively to the encoders ENC$_1$ and ENC$_2$, of two interleavers and a deinterleaver noted π^{-1}.

Each component code is decoded by using soft-input soft-output decoding algorithm. The component LDPC decoders estimate the message, as described in Section 4.

The global decoder receives the soft observations $C_R^{2'}$, $C_R^{1'}$ and $C_I^{2'}$ and estimates the message transmitted. Where $C_I^{2'}$ denotes the received block corresponding to the interleaved systematic information block, while $C_R^{1'}$ and $C_R^{2'}$ denotes the received blocks corresponding to the parity block of the first and second decoders respectively.

In the first iteration, the first decoder DEC$_2$ generates the soft information block $I^2 = \left[I_1^2 I_2^2 \cdots I_{N-M}^2\right]$, using the following received block:

$$\left[C_R^{2'} C_I^{2'}\right] = \left[C_{R1}^{2'} C_{R2}^{2'} \cdots C_{RM}^{2'} C_1^{2'} C_2^{2'} \cdots C_{N-M}^{2'}\right] \tag{36}$$

The second decoder DEC$_1$ generates the soft information block I^1, using the received block $C_R^{1'}$ and the soft information I^2 generated by the first decoder DEC$_2$:

$$\left[C_R^{1'} I^d\right] = \left[C_{R1}^{1'} C_{R2}^{1'} \cdots C_{RM}^{1} I_1^d I_2^d \cdots I_{N-M}^d\right] \tag{37}$$

where $I^d = I^2_{deinterleaved}$

For next iterations, the first decoder DEC$_2$ uses the soft information I^1 generated by the second decoder DEC$_1$ to do the decoding, the decoder input becomes:

$$\left[y^{2'} I\right] = \left[y_1^{2'} y_2^{2'} \cdots y_M^{2'} I_1 I_2 \cdots I_{N-M}\right] \tag{38}$$

where $I = x^{2'} + I^1_{interleaved}$

The presence of the interleaver π and deinterleaver π^{-1} respectively at the output of decoder DEC$_1$ and DEC$_2$ are used to decorrelate the soft decisions at the output of each decoder.

Decoding stops at the end of a fixed number of iterations, and the final decision come from DEC$_1$. One iteration corresponds to one pass from DEC$_2$ to DEC$_1$.

6. Simulation results

In this section, by means of computer simulations, using Matlab programming language, we evaluate the Bit Error Rate (BER) performance of LDPC codes and concatenated codes, as a function of signal to noise ratio (E_b/N_0) in decibel (dB), where E_b is the received energy per bit of transmitted information and N_0 is the power spectral density of the noise.

First, we evaluate in Figs. 4 and 5, for a BPSK modulation and under a gaussian channel, the performance of a binary LDPC code with code rate 1/2 as a function of the iteration number and its block length N ($N = 256$ in Fig. 4 and $N = 2000$ in Fig. 5).

Figs. 4 and 5 show that the performance of LDPC code can achieved a good performance. We can see that the performance of LDPC codes with $N = 2000$ (Fig. 5) is better than that obtained with $N = 256$ (Fig. 4). Therefore, the performance of LDPC code increases with block length, but the

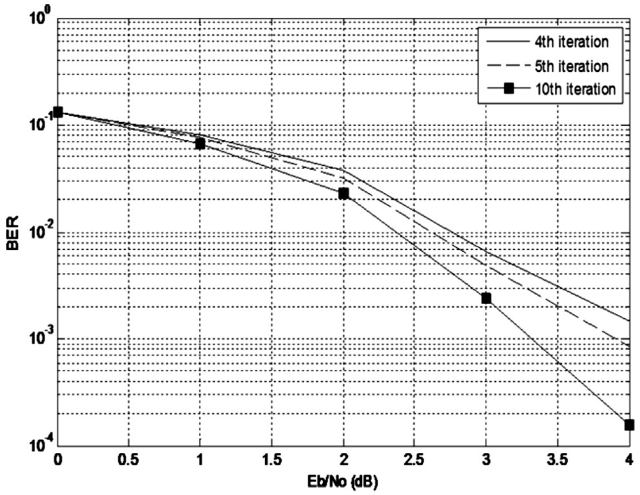

Fig. 4 Performance of LDPC codes with block length $N = 256$, for a BPSK modulation and under a gaussian channel.

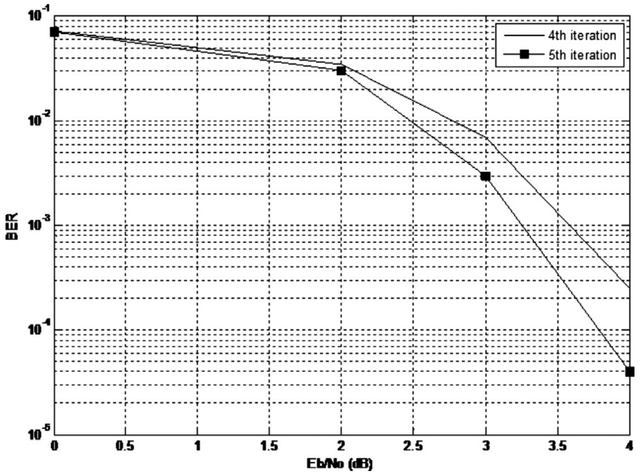

Fig. 5 Performance of LDPC codes with block length $N = 2000$, for a BPSK modulation and under a gaussian channel.

longue block length caused to coding and decoding complexities.

Also, we can see that the performance can be increased more and more by increasing the number of iteration as shown in Figs. 4 and 5, but the large increase here leads to excessive time delay and a computational complexity which make such system unsuitable for real time applications.

The results presented in the literature [7,8,18–20] show that the concatenation of irregular binary LDPC codes is more efficient than a single binary LDPC code. Note that the main characteristic of irregular LDPC codes is that they perform better than regular codes [21]. However, irregular LDPC codes have a higher error floor and coding complexity than regular codes. Therefore, we propose the concatenation of regular codes with an interleaver between them in order to introduce diversity.

In Figs. 6 and 7, for a BPSK modulation, respectively on a Gaussian channel and on a Rician channel, performances of LDPC codes and concatenated codes are compared with the same 256 input bits (N-M = 256) and code rate of 1/3. LDPC codes are made by 512 × 768 parity check matrix. Concatenated codes are composed of two identical rate 1/2 LDPC code with 256 × 512 parity check matrix, decoded by Sum-Product Algorithm (SPA) and its simplified algorithm Min-Sum Algorithm (MSA) [22]. The number of iterations in concatenated codes is set to 6 using one iteration in each component code, and the maximum number of a single LDPC code's iterations is set to 6.

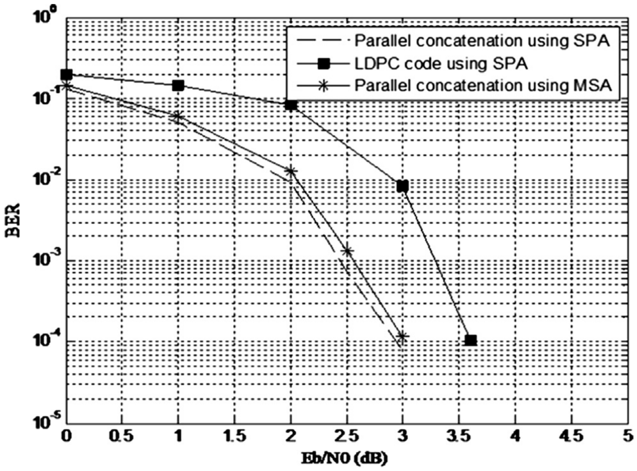

Fig. 6 Performance comparison of a rate 1/3 parallel concatenation LDPC codes employing MSA and SPA with a rate 1/3 LDPC code using SPA, under a Gaussian channel.

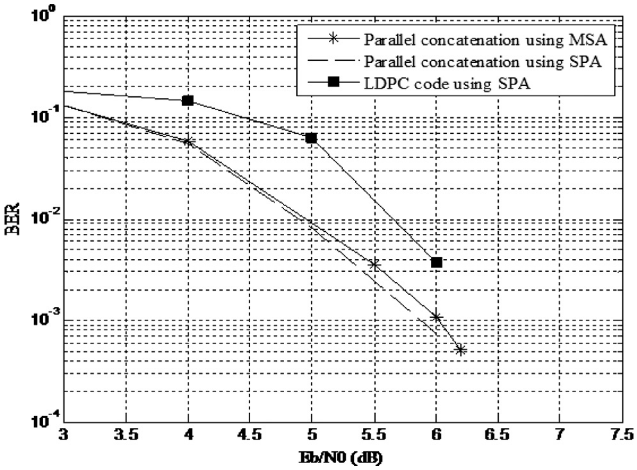

Fig. 7 Performance comparison of a rate 1/3 parallel concatenation LDPC codes employing MSA and SPA with a rate 1/3 LDPC code using SPA, under a Rician channel.

Table 1 Time delay comparison between a single ldpc code and parallel concatenated ldpc codes decoded by spa and msa.

	Codes with 6 iterations		
	LDPC code using SPA	Parallel concatenated LDPC Codes using SPA	Parallel concatenated LDPC Codes using MSA
Time delay	20 min: 25 s	6 min: 8 s	5 min: 45 s

The simulation result presented in Figs. 6 and 7 shows that the concatenated codes outperform LDPC codes. As seen in Figs. 6 and 7, there is a few performance degradation between concatenation coded using SPA and that using MSA. As results, since the performance increases with concatenated codes, it can be use the MSA to more reduce the decoding complexity.

Also, we can see that, with only 6 iterations, there is an important gain between concatenated codes and a single code. The concatenated codes need a less number of iterations to achieve the same performance of a single LDPC code that leads to low decoding complexity and less time delay.

Also, the concatenated codes need less time delay than the single LDPC code, with the same iterations number, as shown in Table 1. The simulation time presented in Table 1, is started from the encoder input and finished at the decoder output.

In the table, one can see that the parallel concatenation LDPC codes needs less time delay than single LDPC code with same iteration numbers. It means that the proposed code not only has higher decoding performance, but also lower time delay than single LDPC code.

7. Conclusion

In this paper, we propose a code scheme that concatenates two regular binary LDPC codes in parallel concatenation separated by an interleaver, using the turbo principle. We concluded that several benefits could be obtained from considering this approach, such as, less decoding delay, less complexity and a good BER performance when compared to a single LDPC code with the same block code.

References

[1] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948), pp. 379-423 and 623-656.

[2] C. Berrou, A. Glavieux, P. Thitimajshima, Near Shannon limit error correcting coding and decoding: turbo-codes, IEEE ICC'93, vol. 2/3, Geneva, May 1993, pp. 1064–1070.

[3] R. Gallager, Low-density parity-check codes, IEEE Trans. Inform. Theory 8 (1) (1962) 21–28.

[4] D. MacKay, Good error-correcting codes based on very sparse matrices, IEEE Trans. Inf. Theory 45 (2) (1999) 399–431.

[5] S. Johnson, Iterative Error Correction Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes, Cambridge University Press, 2010.

[6] M.G. Luby, M. Mitzenmachery, M.A. Shokrollahiz, D.A. Spielman, V. Stemann, Practical loss-resilient codes. STOC'97 Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, 1979, pp. 150–159.

[7] H. Behairy, S.C. Chang, Parallel concatenated Gallager codes, Electron. Lett. 36 (2000) 2025–2026.

[8] B. Belgheit, A. Boukelif, A. Moulay Lakhder, M. Kamline, Parallel concatenated Gallager codes matrix and the effect of interleaver, Int. J. Electron. 99 (9) (2012) 1281–1289.

[9] Zhe Wang, Meng Zhang, A serial concatenated scheme for LDPC code to achieve better error correction performance, 2nd International Conference on Consumer Electronics, Communications and Networks, 21–23 April 2012, pp. 1587–1589.

[10] Hatim M. Behairy, Mohammed Benaissa, Multiple parallel concatenated gallager codes: code design and decoding techniques, IETE J. Res. 59 (2014).

[11] H.A.N. Jian-Bing, H.E. Chen, H.E. He Yun, Research on regular LDPC codes with better performance than turbo codes, International Conference on Information Engineering ICIE '09, WASE, 2009.

[12] Su-Chang Chae, Yun-Ok Park, Low complexity encoding of improved regular LDPC codes, Vehicular Technology Conference IEEE, 2004.

[13] R.M.A. Tanner, Recursive approach to low complexity codes, IEEE Trans. Inform. Theory IT-27 (5) (1981) 533–547.

[14] S.J. Johnson, Low-density parity-check codes: design and decoding, Chapter in Wiley Encyclopedia of Telecommunications (2002).

[15] J. Chen, A. Dholakia, E. Eleftheriou, Marc P.C. Fossorier, Hu Xiao-Yu, Reduced-complexity decoding of LDPC codes, IEEE Trans. Commun. 53 (8) (2005) 1288–1299.

[16] Y. Jiang, A practical guide to error-control coding using MATLAB, Artech House (2010).

[17] S. Benedetto, G. Montorsi, D. Divsalar, F. Pollara, Soft-output decoding algorithms in iterative decoding of turbo-codes. JPL TDA Progress Report 42-124, February 15, 1996.

[18] H.M. Behairy, M. Benaissa, Multiple parallel concatenated gallager codes: code design and decoding techniques, IETE J. Res. 59 (2014).

[19] Z. Wang M. Zhang, A serial concatenated scheme for LDPC code to achieve better error correction performance, 2nd International Conference on Consumer Electronics, Communications and Networks, 21–23 April 2012, pp. 1587–1589.

[20] H. Behairy, S.C. Chang, Parallel concatenated gallager codes for CDMA applications, Global Telecommunications Conference GLOBECOM'01, IEEE, 2001.

[21] L. Mostari, R. Meliani, A. Bounoua, Iterative effect on LDPC code performance, Mediterr. Telecommun. J. 01 (02) (2011) 63–67.

[22] L. Mostari, A. Bounoua, A. Taleb-Ahmed, R. Meliani, Decoding for binary LDPC Codes under Gaussian channel, The 4th Workshop on Codes, Cryptography and Communication System, Meknes, Morocco, 07–08 November, 2013.