Regular Article

A New High Performance Decoder for LDPC Codes

Nguyen Tung Hung[1], **Nguyen Van Duan**[2]

[1] Telecommunications University, Nha Trang-Khanh Hoa, Vietnam
[2] Le Quy Don Technical University, Ha Noi, Vietnam

Correspondence: Nguyen Tung Hung, tunghungus@yahoo.com

Manuscript communication: received 4 April 2013, accepted 10 January 2014

Abstract– **The article introduces a new decoder for LDPC codes based on the general check matrix and soft syndrome. Simulation result shows that the new decoder can improve the performance of LDPC codes. Compared with some other improvements, the new decoding algorithm is simpler, and it can detect errors and be applied to great length LDPC codes.**

Keywords– **General check matrix, soft syndrome, iterative decoding algorithm.**

1 Introduction

In BPA (Belief Propagation Algorithm) decoding of LDPC codes, dominant decoding failures are, in general, caused by trapping sets (TS) [1, 2]. The dominant TS are formed by a combination of short cycles present in the bipartite graph of the parity check matrix $\boldsymbol{H}$. At high E_b/N_0 values (E_b is the bit energy and N_0 is the noise spectral density), TS have been identified as one of the main factors causing error floors of BER (Bit Error Ratio) and FER (Frame Error Ratio).

To reduce the influence of TS, in [3, 4], the authors gave a solution which is to find the most dominant TS and propose the G-LDPC decoder (Generalized LDPC decoder), allowing improvement in decoding quality at the high E_b/N_0 values (the error floor region). However, defining TS is complex and difficult to implement with great length LDPC codes.

In [5, 6], the authors introduce BPA-OSD decoder (Ordered Statistic Decoding) which can improve performance at all E_b/N_0 values. However, the BPA-OSD decoder is very complex, difficult to apply to the medium and great length LDPC codes.

Based on the concept of the SS (Soft Syndrome) which is presented in [7], the article introduces a new G-LDPC decoder called G-LDPC-OSS (G-LDPC with the Order of Soft Syndrome decoder) having better performance and simpler than the above two decoders. The performance and complexity comparison between the G-LDPC-OSS with other decoders will be presented through the LDPC codes introduced in [8]. In addition, the article also presents simulation results to compare the performance of the G-LDPC-OSS decoder and the conventional BPA decoder in Wi-MAX system [9] and DVB -S2 (Digital Video Broadcasting S2 standard) [10].

2 BPA and G-LDPC decoders

Assume a binary (n, k) LDPC code with length n and dimension k, then the parity check matrix is $\boldsymbol{H}_{m\times n}$, where $m = n - k$ is the number of checksums. Information bits $\boldsymbol{u} = u_1, u_2, ..., u_k$ are encoded into a codeword $\boldsymbol{Y} = y_1, y_2, ..., y_n$, then modulated and transmitted through the channel. The input of the BPA decoder is the Log Likelihood Ratio (LLR):

$$L(\hat{y}_i) = log\frac{Pr(\hat{y}_i = 0 \mid \boldsymbol{r})}{Pr(\hat{y}_i = 1 \mid \boldsymbol{r})} \tag{1}$$

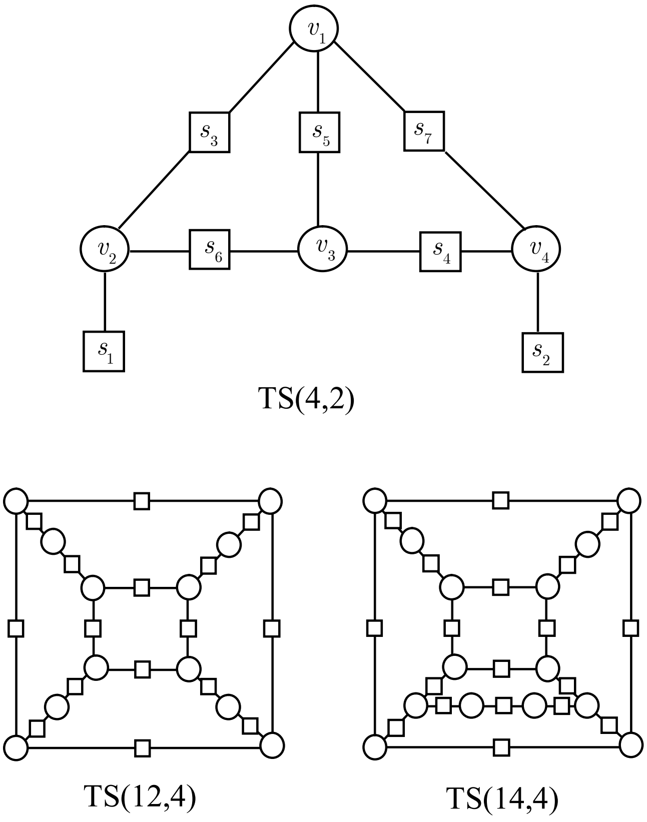
where $\boldsymbol{r}$ is the set of symbols acquired from channel and $Pr(\hat{y}_i = 0 \mid \boldsymbol{r})$ is the conditional probability. Normally, the check matrix $\boldsymbol{H}$ is a sparse matrix with a few number of "1" value in each row and each column. The matrix $\boldsymbol{H}$ of LDPC code set can be described by a bipartite Tanner graph [11] with the Bit Nodes (BN) $v_1, v_2, ..., v_n$ and Check nodes (CN) $s_1, s_2, ..., s_m$. Corresponding to the locations of $\boldsymbol{H}$ with the value "1", there is a connection between the BN and the CN. BPA algorithm [1, 12] is an iterative decoding algorithm with two main phases: 1) make calculation in rows to update information for all CN and send information from CN to BN, 2) make calculation in columns to update information for BN and send information from BN to CN. The output of the BPA decoder is the LLR of bits $L(\hat{y}_{i=1,2,...n})$, which is used for hard decision to become a codeword $\hat{\boldsymbol{Y}} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_n)$. If the syndrome is:

$$\hat{\boldsymbol{Y}}.\boldsymbol{H}^T = [0, 0, ..., 0] \tag{2}$$

stop the iteration. Otherwise, the process repeats until the number of iterations I reaches the maximum value I_{max} (assuming I_{max} in following studies).

In [2], a TS(z, w) is a set of z bit nodes, for which the subgraph of the z bit nodes and the check nodes that are directly connected to them contains exactly w odd-degree check nodes. Figure 1 shows an error TS(4,2) of an LDPC code with a degree of bit nodes and check nodes (3,6) in which the LLR value of BN is computed by the sum of intrinsic information and extrinsic information from three related CN.

Assume that an all-zero codeword ($\boldsymbol{Y} = 0$) is sent through an AWGN channel (logic 0 is encoded as +1,



TS(4,2)

TS(12,4)

TS(14,4)

Figure 1. Typical dominant trapping sets

while a logic 1 is encoded as -1), and all bits are received correctly except the 4 bits in the trapping set TS(4,2), we have $L(y_i) < 0$ with $1 \leq i \leq 4$, $L(y_i) > 0$ with $4 < i \leq n$. In this case, the LLR of the nodes v_1, v_3 will be errors because all the components have negative values and the LLR of the nodes v_2, v_4 have a large probability of errors. Then, the BPA decoder falls into the TS until the positive information from the nodes s_1, s_2 are large enough to change the sign of v_2, v_4.

To improve the performance of LDPC decoding at the error floor region when the decoder falls into the TS, it is required to conduct post-processing. In [13], the authors propose a post-processing technique to lower the error floor by using a look-up table of known trapping sets. After conventional BPA decoding, this table is used to process the residual error blocks much like a syndrome decoder. Regarding the great length LDPC codes, there will be a great number of TS, and the method of syndrome decoding and referring to the look-up table is very complicated. It was observed through simulations of many LDPC codes that, in the error floor region, a frame error event usually contains a single TS. The authors in [3, 4] provide the method of combination of CN into Super Check Nodes (SCN), so that information from the error-free TS can correct the error TS. For example, the Margulis code introduced in [8] with 1320 TS and 1320 TS (Figure 1) are the most dominant TS in the error floor region of the Margulis code. The parity check matrix generated by adding modulo 2 of the matrix rows $\boldsymbol{H}$ corresponding to the combination of CN is called general parity check matrix and the decoder is called G-LDPC (Generalized - LDPC decoder).

The disadvantage of the above methods is the it requires to predefine the most dominant TS. If the length of the LDPC code is large, the calculation to find TS will be very complicated. On the other hand, at the low E_b/N_0, a frame error event usually contains a combination of multiple TS and the TS with large z and w have not been considered yet. Therefore, the performance of the decoders cannot be improved on the low E_b/N_0.

3 Proposal of the New G-LDPC-OSS

The general parity check matrix $\boldsymbol{H}_G$ of the parity check matrix $\boldsymbol{H}$ of size $m \times n$ is a matrix of size of $(m+a) \times n$ under the form:

$$\boldsymbol{H}_G = \left[\begin{array}{c} \boldsymbol{H} \\ \boldsymbol{H}_a \end{array} \right] \tag{3}$$

where $\boldsymbol{H}_a$ is an additional matrix with a rows and n columns, each row is obtained from the sum of any two rows of the matrix $\boldsymbol{H}$.

From the theory of linear codes, we have:

$$\boldsymbol{Y}.\boldsymbol{H}^T = [0, 0, ..., 0] \tag{4}$$

is a system of linear equations, so we also have:

$$\boldsymbol{Y}.\boldsymbol{H}_G{}^T = [0, 0, ..., 0] \tag{5}$$

This $\boldsymbol{H}_G$ is a parity check matrix of the LDPC code and $\boldsymbol{H}_G$ is also the specific case of G-LDPC matrix. If the number of extended rows is $a \ll n$, the matrix $\boldsymbol{H}_G$ which is a sparse matrix can be represented by a Tanner bipartite graph. The value of a is the number of SCN created in the Tanner graph of the matrix $\boldsymbol{H}_G$. An example of the construction of the matrix $\boldsymbol{H}_G$ of the matrix $\boldsymbol{H}$ is described in Figure 2. Node s_7, which is an SCN in Figure 2, is formed by adding modulo 2 between the 6th row and the first row (combination of node s_1 and node s_6). Similarly, s_8 is also an SCN generated by combining s_6 and s_4.

The soft syndrome defined in [7] is the LLR of the check nodes $s_{i=1,2,...m}$ with:

$$L(s_{i=1,2...m}) = \log \frac{\Pr\left(s_i = 1 \left| L\left(\hat{y}_{j=1,2...n}\right)\right.\right)}{\Pr\left(s_i = 0 \left| L\left(\hat{y}_{j=1,2...n}\right)\right.\right)} \tag{6}$$

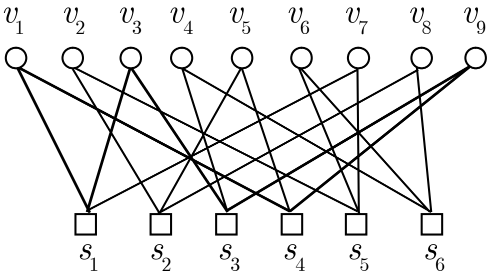
We have:

$$s_i = \sum_{\oplus} \left(\hat{y}_j \oplus \boldsymbol{H}\left(i,j\right)\right),\ j \in V_i,\ i = 1...m \tag{7}$$

where V_i is the set of branches connecting from bit node to check nodes on Tanner graphs (where the i^{th} row of the matrix $\boldsymbol{H}$ has a value of "1") and the calculation $\bigoplus$ is the modulo 2 operator. Based on the studies of LLR algebra in [14, 15], we have:

$$\begin{aligned} L(s_i) &= \log \frac{\prod\limits_{j\in V_i}\left(e^{L(\hat{y}_j)}+1\right)+\prod\limits_{j\in V_i}\left(e^{L(\hat{y}_j)}-1\right)}{\prod\limits_{j\in V_i}\left(e^{L(\hat{y}_j)}+1\right)-\prod\limits_{j\in V_i}\left(e^{L(\hat{y}_j)}-1\right)} \\ &\approx \prod_{j\in V_i} sign\left(L(\hat{y}_j)\right).\min_{j\in V_i}\left|L(\hat{y}_j)\right| \end{aligned} \tag{8}$$

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



$$\boldsymbol{H}_G = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

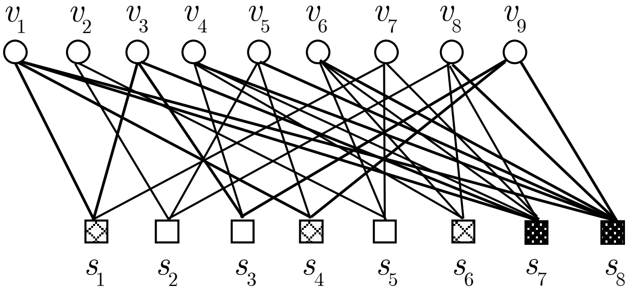


Figure 2. Parity check matrices $\boldsymbol{H}$, $\boldsymbol{H}_G$ and their Tanner bipartite graphs

Consider the case of the BPA decoder falling into the TS with the error bit node v_1 and signs of relevant nodes such as s_1, s_4 are not large enough to change the sign of v_1 as shown in Figure 2 and assuming the node s_6 is the node with the largest $|L(s_6)|$. According to (8), the bits corresponding to the bit nodes connected to the node s_6 will be the most reliability and less prone to error bits. If combine node s_6 with nodes s_1 and s_4, the most reliable information from highly reliable bit nodes will be passed through s_6 to v_1, allowing recovery from the trapping set error event.

The new decoding algorithm is based on the idea of combining the most reliable CN with the CN related to less reliable BN (the nodes with small $|L(\hat{y}_j)|$) to create the matrix $\boldsymbol{H}_G$ and then redecode. The decoding algorithm consists of two stages:

Stage 1: Decode LDPC with the input of (1) using the BPA algorithm with the conventional check matrix $\boldsymbol{H}$. Similar to the conventional BPA decoder, at each iteration, check the condition:

$$\hat{\boldsymbol{Y}}_1.\boldsymbol{H}^T = [0, 0, ..., 0] \tag{9}$$

if satisfied, stop the iterative decoding and give the codeword $\hat{\boldsymbol{Y}}_1$. If condition (9) is not satisfied, perform stage 2.

Stage 2: Calculate $L(s_i)$ based on $L(\hat{y}_{i=1,2...n})$ and matrix $\boldsymbol{H}$ in (8). Arrange $|L(s_i)|$ by descending value and we have the permutation π. Starting from the node $s_{\pi(1)}$ with the largest $\left|L(s_{\pi(1)})\right|$, combine node $s_{\pi(1)}$ with a SCN nodes connected to e consecutive bit nodes with $|L(\hat{y}_i)|$ arranged in ascending order to get matrix $\boldsymbol{H}_G$. Decode BPA with the input of (1) and the general parity check matrix $\boldsymbol{H}_G$. At each re-decoding γ, the check condition is:

$$\hat{\boldsymbol{Y}}_2.\boldsymbol{H}_G{}^T = [0, 0, ..., 0] \tag{10}$$

where $\hat{\boldsymbol{Y}}_2$ is the codeword of decision of hard decoding in accordance with $\boldsymbol{H}_G$. If condition (10) is satisfied, replace $\hat{\boldsymbol{Y}}_1$ with $\hat{\boldsymbol{Y}}_2$ and exit the loop. If condition (10) is not satisfied, keep combining $s_{\pi(1)}$ with the next a SCN nodes. If using node $s_{\pi(1)}$ to decode but still not successful, continue to use nodes $s_{\pi(2)}$, $s_{\pi(2)}$, ... until $\gamma = \gamma_{max}$ (γ_{max} is the maximum number of times of redecoding).

The above decoder is based on the general parity check matrix $\boldsymbol{H}_G$ and on the order of the soft syndrome, so it is called G-LDPC-OSS (G-LDPC with the Order of Soft decoder Syndrome). The value of e needs to be obtained through simulation and depends on the LDPC code. Figure 3 depicts simulation results of the G-LDPC-OSS decoder for the 96.33.964 and 252.252.3.252 codes [8] corresponding to $E_b/N_0 = 3.5$ dB and

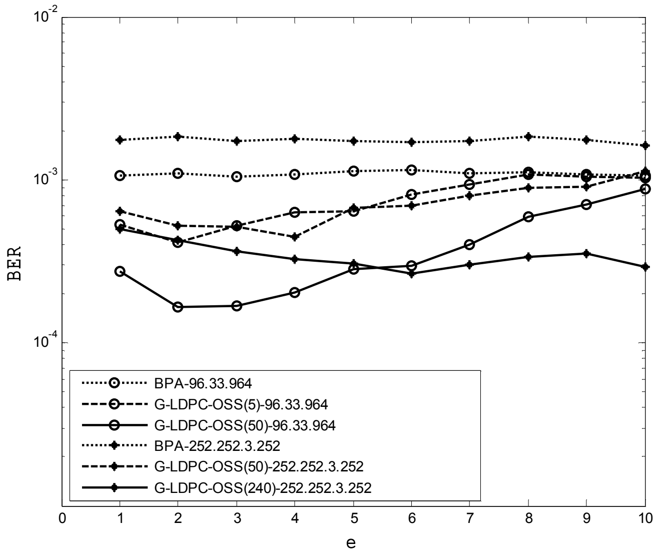


Figure 3. BER with different values of e

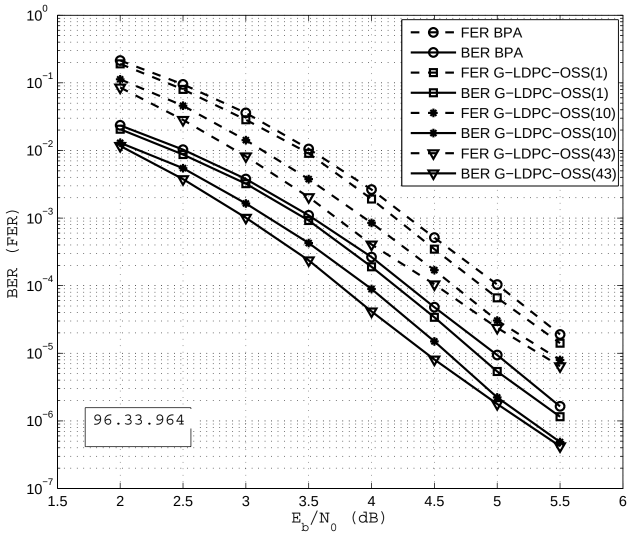


Figure 4. BER and FER comparisons between BPA and G-LDPC-OSS decoders for 96.33.964 code

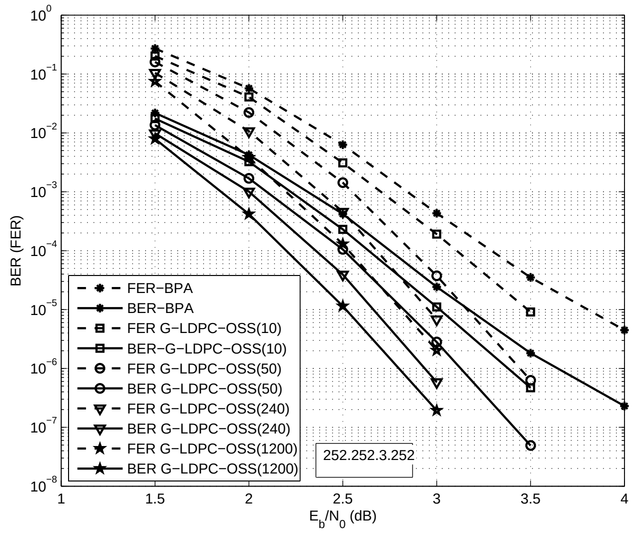


Figure 5. BER and FER comparisons between G-LDPC-OSS and BPA decoders for 252.252.3.252 code

$E_b/N_0 = 2.2$ dB with the error number being 150. Based on Figure 3, we can see that for the 96.33.964 code, we should choose $e = 2$. For the 252.252.3.252 code, when $\gamma_{max} = 50$, choose $e = 4$ and when $\gamma_{max} = 240$, choose $e = 6$. These codes are regular LDPC codes and there are 3 values of "1" in a column, so we have $a = 3 \times e$.

Figure 4 and Figure 5 depicts simulation results for the 96.33.964 and the 252.252.3.252 codes with different maximum re-decoding time γ_{max}. Based on Figure 4, we note that the G-LDPC-OSS(1) (with the number of re-decoding times $\gamma_{max} = 1$) achieves 0.1 dB coding gain over the BPA decoder. If we increase the number of re-decoding γ_{max}, BER and FER can be improved to 0.6 dB for small E_b/N_0. For the 252.252.3.252 code depicted in Figure 5, G-LDPC-OSS with different values of γ_{max} achieved about 0.1 dB to 0.5 dB coding gain over BPA at small E_b/N_0. At the high E_b/N_0, the performance can be improved more than 0.8 dB at a BER of 10^{-7} and higher than 1.0 dB in the error floor region.

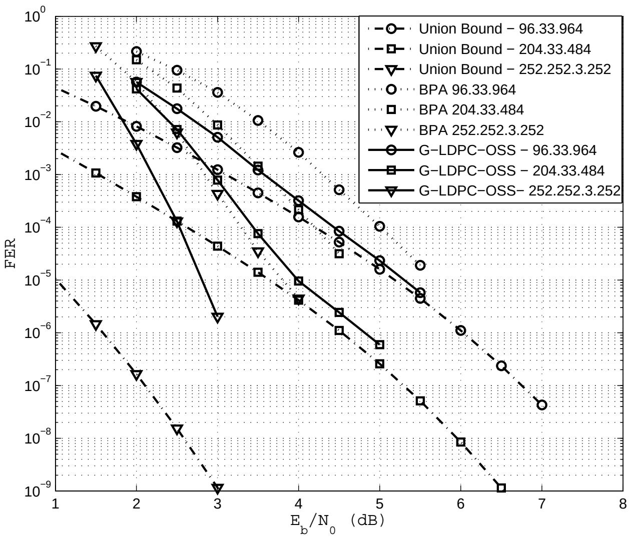


Figure 6. FER comparisons between G-LDPC-OSS, BPA, MLD decoders

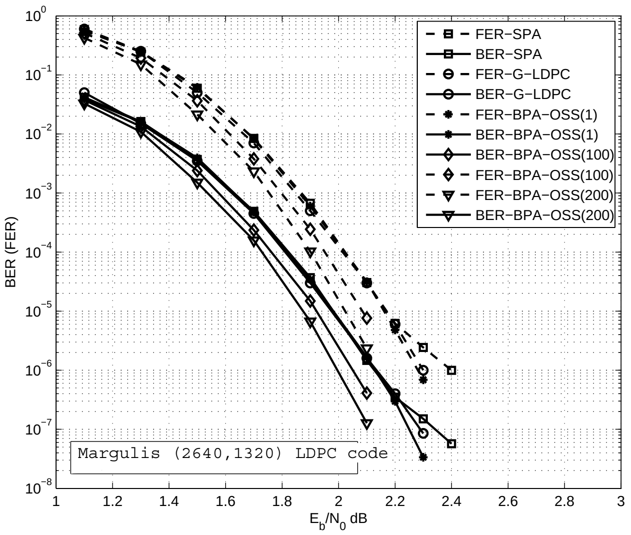


Figure 7. Performance of the G-LDPC-OSS, G-LDPC and BPA decoders

4 Compareisons of the Performance of the G-LPDC-OSS with Other Decoders

First, we will compare the performance of the G-LDPC-OSS with the Union Bound (UB) used to evaluate the Maximum Likelihood decoder (MLD), which is the ideal decoder that is too complex to implement in reality. The UB is calculated using the formula:

$$\mathrm{P_e} < \sum_{\mathrm{d=d_{free}}} aQ\left(\sqrt{2dRE_b/N_0}\right) \tag{11}$$

where a is the codeword number with distance d, d_{free} is the minimum distance. The pair $[a, d]$ is called the distance spectrum of the code. The distance spectrum of LDPC codes calculated as in [16] with 96.33.964 code is $a = [2\ 0\ 9\ 0\ 420\ 470]$, $d_{free} = 6$. This means that this code has two codewords with minimum distance $d_{free} = 6$, 9 codewords with the distance 8, and 420 codewords with distance 10, etc.

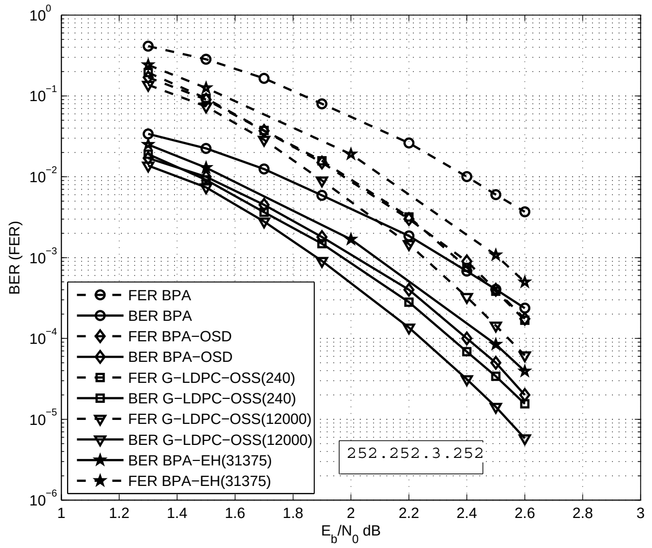


Figure 8. Performance of the G-LDPC-OSS, BPA-OSD, BPA-EH and BPA decoders

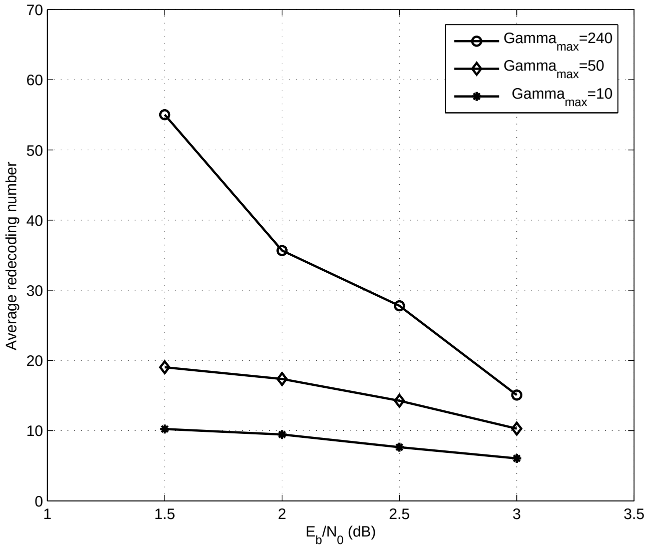


Figure 9. γ_a vs E_b/N_o for different values of γ_{max} for the 252.252.3.252 code

The distance spectrum of the 204.33.484 code is $a = [1\ 0\ 1\ 0\ 5\ 0\ 43\ 0\ 314]$, $d_{free} = 8$, of the 252.252.3.252 code is $a = [2\ 0\ 22\ 0\ 117\ 0\ 481]$ $d_{free} = 20$. The simulation results in Figure 6 show that in the case of short codewords such as 96.33.964 and 204.33.484, the performance of G-LDPC-OSS decoding with the re-decoding time $\gamma_{max} = 215$ is quite close to the MLD performance. For the medium code length such as 252.252.3.252, the performance of the G-LDPC-OSS with $\gamma_{max} = 1200$, can approach the performance of MLD for very low FER.

Figure 7 compares the performance of G-LDPC-OSS with the G-LDPC decoder in [3, 4], used for the Margulis code. We can see that the performance of G-LDPC only improves the error floor. Meanwhile, the performance of G-LDPC-OSS is improved at low E_b/N_0 and is better than 0.3 dB at 10^{-6}.

The BPA-OSD decoder presented by the authors of [5, 6] includes the two above stages. In stage 2, in each iteration, the BPA-OSD algorithm arranges $|L_1(y_{i=1,2\ldots n})|$ by ascending value to find permutation π_1. Rearrange the $\boldsymbol{H}$ matrix following π_1 into $\boldsymbol{H}(\pi_1)$, then estimate the Gauss-Jordan $\boldsymbol{H}(\pi_1)$ from left to right to find the systematic matrix $\boldsymbol{G}(\pi_2)$. Next, re-encode all the codewords with the total number of error bits from 1 to p based on systematic form of $\boldsymbol{G}$, and compute the decoding metric associated with each constructed codeword. Select the most likely codeword among the $\sum_{\ell=0}^{p} \binom{k}{\ell}$ constructed candidate codewords. The performance of BPA-OSD depends on the value of p and if p is large, it may approach to the performance of the MLD.

Figure 8 depicts the simulation results of the order-p BPA-OSD decoder ($p = 2$) and G-LDPC-OSS for the 252.252.3.252 code. We observe that the FER of G-LDPC-OSS with $\gamma_{max} = 240$ approximates the performance of BPA-OSD and the BER is improved by 0.05 dB. When the number of redecoding is large, $\gamma_{max} = 12000$, the performance of G-LDPC-OSS is about 0.15 dB better than BPA-OSD. The performance of the BPA-EH decoder (BPA based on Equivalent $\boldsymbol{H}$) with equivalent matrix [7] is also shown in Figure 8. Compared to the quality of the BPA-EH with a number of re-decoding $\gamma_{max} = 31375$, we can see that the performance of G-LDPC-OSS is better and has requires less re-decoding times.

5 The Complexity of the G-LDPC-OSS Decoder and Application

Compared with the G-LDPC algorithm, the G-LDPC-OSS algorithm does not have to predefine a dominant TS. The performance of the G-LDPC-OSS algorithm with only one time of redecoding can also improve the performance of the Margulis code in the error floor region (Figure 7). For great length LDPC codes, it will be difficult to find the TS, so the G-LDPC-OSS algorithm has the generality for all LDPC codes and better performance.

The BPA-OSD decoder has a large complexity. According to [6], the number of binary calculations in estimating Gaussian-Jordan is an exponential function $O(n^3)$. The number of binary operations in re-encoding is the sum of the exponential functions $O(n^{(l+1)})$, $0 \leq n \leq p$. Thus, if value of n is large, the complexity increases by an exponential function of n, it is very difficult to implement in reality. Because matrix $\boldsymbol{H}_G$ received from the matrix $\boldsymbol{H}$ is also a sparse matrix, the complexity of the G-LDPC-OSS algorithm is similar to the BPA algorithm which means an increase by the function $\gamma_a O(n)$ where γ_a is the average number of times of redecoding). Figure 9 compares γ_a with different values of γ_{max} for the 252.252.3.252 code. The result shows that when E_b/N_o is high, γ_a is reduced, so the complexity of the decoder decreases. To compare the complexity of the G-LDPC decoder and BPA-OSD-OSS decoder, we compare the processing time in stage

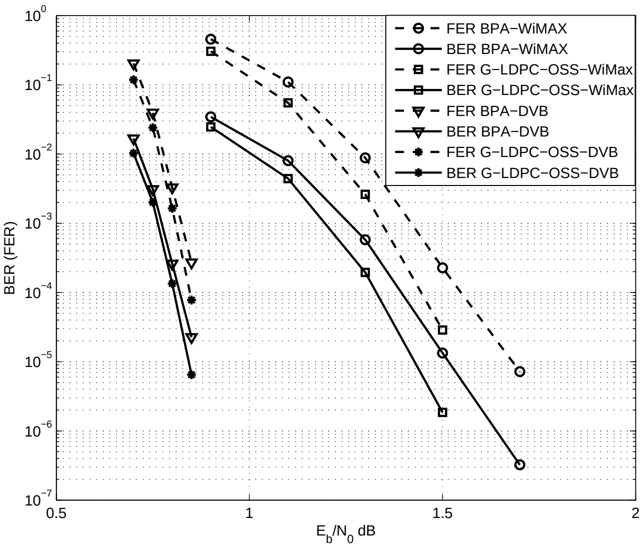


Figure 10. Performance of the G-LDPC-OSS and BPA decoders in Wi-MAX and DVB-S2 systems

2 of the 252.252.3.252 code on the same event of errors and the same computer. The result shows that, in each iteration, the total time to process Gauss-Jordan and re-encode is 3.63 seconds, so processing 50 iterations will take 181.5 seconds. The BPA-OSD algorithm must always fulfill Gauss-Jordan estimation and re-encoding, so the complexity of this algorithm is very large and is difficult to apply to great length LDPC codes. For the G-LDPC-OSS decoder, it only takes 0.25 seconds for 50 times of re-decoding. Figure 9 compares γ_a with different values γ_{max} for 252.252.3.252 code. We observe that at $E_b/N_o = 3$ dB, due to $\gamma_a \approx 0.2\gamma_{max}$, the average period for re-decoding is just 0.05 seconds (much smaller than 181.5 seconds). Besides, the BPA-OSD algorithm has no characteristics of error detection because the output is always a valid codeword, while G-LDPC-OSS algorithm can still detect errors and facilitate the application of Automatic Repeat Request systems (ARQ).

Consider the two LDPC codes used in Wi-MAX system with the size of (4512, 2256) and the LDPC code used for the Digital Video Broadcasting standard S2 (DVB-S2) with size of (64800, 32400). These are great length LDPC codes, so it is difficult to apply the G-LDPC or BPA-OSD algorithms. Simulation result of evaluation of BER and FER of the two above codes with G-LDPC-OSS at $\gamma_{max} = 500$ is described in Figure 10. The simulation results show that for the Wi-MAX code, the performance can be improved in all areas of E_b/N_o and the range 0.1 dB at FER = 10^{-5}. For DVB-S2 code, at 0.85 dB, when the BPA algorithm has 7 frame errors, the G-LDPC-OSS algorithm only has 2 errors, so the number of errors can be reduced by more than a half. Therefore, the performance of G-LDPC-OSS can improve 0.05 dB even if the code is very large in size with the performance approaching the Shannon limit.

6 Conclusion

The article introduces a new decoder allowing improvement of the LPDC decoding performance. The new decoder is based on a generalized parity check matrix and order of soft syndrome and is denoted as G-LDPC-OSS. Simulation results show that, for LDPC codes with short length, the performance of the G-LDPC-OSS decoder is better than BPA and can approach the MLD performance.

The complexity of the G-LDPC-OSS decoder is a first order linear function with the codeword length, so it can be applied to decode the LDPC codes with great length. Compared to other LDPC decoder improvements, such as the decoders of G-LDPC and BPA-OSD, the G-LDPC-OSS decoder is simpler and has better performance. If this new decoder is applied to the systems using great length LDPC codes such as Wi-MAX, DVB-S2, the performance can be significantly improved with acceptable complexity.

References

[1] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[2] T. Richardson, "Error floors of ldpc codes," in *Proceedings of the annual Allerton conference on communication control and computing*, vol. 41, no. 3. The University; 1998, 2003, pp. 1426–1435.

[3] Y. Han and W. Ryan, "Low-floor decoders for ldpc codes," *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1663–1673, 2009.

[4] Y. Han and W. E. Ryan, "Ldpc decoder strategies for achieving low error floors," in *Information Theory and Applications Workshop, 2008*. IEEE, 2008, pp. 277–286.

[5] M. Fossorier, "Iterative reliability based decoding of ldpc codes," in *IEEE International Symposium on Information Theory, 2001. Proceedings.* IEEE, 2001, p. 233.

[6] G. Li and G. Feng, "Generalised reliability-based syndrome decoding of ldpc codes," *European Transactions on Telecommunications*, vol. 19, no. 8, pp. 873–877, 2008.

[7] N. T. Hung, "A new decoding algorithm based on equivalent parity check matrix for ldpc codes," *REV Journal on Electronics and Communications - The Radio Electronics Association of Vietnam*, vol. 3, no. 12, 2013.

[8] D. J. MacKay, "Encyclopedia of sparse graph codes," 2005.

[9] Y. Xiao, "Turbo and ldpc codecs and their applications," *Press of Posts and Telecommunications, Beijing*, 2010.

[10] (2005) Dvb-s.2 standard specification.

[11] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[12] M. P. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 673–680, 1999.

[13] E. Cavus and B. Daneshrad, "A performance improvement and error floor avoidance technique for belief propagation decoding of ldpc codes," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005.*, vol. 4. IEEE, 2005, pp. 2386–2390.

[14] G. Battail, "Building long codes by combination of simple ones, thanks to weighted-output decoding," *Proc, URSIISSSE*, pp. 634–637, 1989.

[15] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[16] D. Declercq and M. Fossorier, "Improved impulse method to evaluate the low weight profile of sparse binary linear codes," in *IEEE International Symposium on Information Theory, 2008. ISIT 2008.* IEEE, 2008, pp. 1963–1967.

Nguyen Tung Hung was born in Hung Yen, Vietnam, in 1969. He received the B. Eng, the M. Eng. and Dr. Eng from Le Qui Don Technical University, Hanoi, Vietnam respectively, in 1992, 1999 and 2006. Since 2006, he has been working at Telecommunications University, Nha Trang, Vietnam. He is currently the dean of the Faculty of Telecommunications Technology. His main interests are channel coding theory and its applications.

Nguyen Van Duan was born in 1974. He received the Master degree from Le Qui Don Technical University, Hanoi, Vietnam, in 2009. From 2010 to 2011, he was lecturer at Telecommunications University, Nha Trang, Vietnam. He is currently a PhD student in Le Qui Don Technical University, Hanoi, Vietnam. His main interests are channel coding theory and its applications.