A Design of Parity Check Matrix for Irregular LDPC Codes

Chutima Prasartkaew [*] and Somsak Choomchuay [†]

[*]College of Data Storage Technology and Applications, King Mongkut's Institute of Technology Ladkrabang, BKK, Thailand
Tel:/Fax: + 66-2-326-4731, E-mail: prasartkaew@yahoo.com

[†]Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, BKK 10520, Thailand
Tel: +66-2-326-4222 Ext.114, Fax: +66-2-739-2398, E-mail: kchsomsa@kmitl.ac.th

***Abstract*— This paper outlines a work on a design of parity check matrix for Irregular LDPC codes. The design is based on the adjustment of the modified array LDPC codes and interleave-modified array LDPC codes. The code rate of 0.930 is obtained. LDPC Codes based on this design suits the short and medium block length.**

I. INTRODUCTION

Error Correction Codes (ECC) is one of many tools made available for achieving consistent data transmission. Low-Density Parity-Check codes (LDPC), as ones of many kinds, are also linear block codes that have been studied vastly in this decade. The main advantages of the codes are that they provide the performance at that close to limited capacity for many different channels and linear time complex algorithms for decoding. They also suit well the parallel realization.

In this paper we propose a modified method to obtain parity check matrices. The obtained result is comparable to a published work [1] but with a higher code rate. The rest of this paper is organized as follows: General perspective of LDPC codes is given in section II where encoder and decoder are also included. A design of parity check matrices is outlined in section III and the corresponding performance tests are reported in section IV. The paper is concluded in section V.

II. LDPC CODES

Of its discovery in 1960 by Gallager [2], the LDPC code has been ignored for some ten years. This was because the code itself is quite complex. In the same time the more highly-structured code; Reed Solomon code was introduced [3]. The introduction of Turbo code by Berrou, Glavieux and Thitimajshima in 1993 [4] has drawn great attention to researchers since the code performance is close to Shannon limit. LDPC codes were recovered in 1998 by Richardson and Urbanke [5] and in 1999 by Mackey [6]. LDPC became more popular and widely developed for wider area of applications including communications and data storage.

There are two different ways to represent LDPC codes; matrix representation and graphical representation. In the matrix point of view, as it is named, LDPC codes hold small number of "1" in each row and column, i.e. $W_c << n$ and $W_r << m$ for a dimension $m \times n$ parity matrix. This can provide large minimum distance of the code. However such a circumstance results a large parity check matrix. In the graph point of view, Tanner graph [7] is an efficient graphical representation of LDPC codes. There are *m* check nodes (c-nodes; number of parity bits) and *n* variable nodes (v-nodes; number of bits in a codeword).

LDPC codes are said to be regular if W_c is constant for every column, and $W_r = W_c(n/m)$. If the parity matrix H is low density but the number of "1" in each row or column are not constant, the code is said to be an irregular one.

A. Encoder

Similar to all other linear block codes, we have the relation;

$$C_{(1\times n)} H^T_{(n\times m)} = 0 \tag{1}$$

where C is a codeword matrix, and H is a parity check matrix. In a systematic form, C can be written as:

$$C_{(1\times n)} = \left[m_{(1\times m)} \mid p_{(1\times n-m)} \right] \tag{2}$$

Where $p_{(1\times n-m)}$ denotes the parity portion, and $m_{(1\times m)}$ denotes the message portion respectivrly. With $H = \left[H_1 \; H_2 \right]$ or $H^T = \begin{bmatrix} H_1^T \\ H_2^T \end{bmatrix}$ we can have;

$$CH^T = \left[m \mid p \right] \begin{bmatrix} H_1^T \\ H_2^T \end{bmatrix} = mH_1^T + pH_2^T = 0 \tag{3}$$

Or

$$p = mH_1^T + \left(H_2^T \right)^{-1} \tag{4}$$

The task of the encoder is then to compute the parity matrix P that can be directly appended to the message to produce the codeword.

For the matrix H to be more manageable, the LU decomposition method can be preferably applied; i.e. $[H] = [L][U]$. Thus,

$$\begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \underbrace{\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}}_{Y} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} \tag{5}$$

Let $[Y]=[U][P]$, then we can use forward substitution to solve $[L][Y]=[M]$.

$$\begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} \tag{6}$$

Finally, the backward substitution is employed to solve for P of which $[U][P]=[Y]$. There, we can get $\{p_i\}$ as needed.

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{7}$$

B. Decoder

There are several methods used in decoding the LDPC codes. Each was derived individually. These are, for instance, Believe Propagation (BP), Sum-Product (SP), and Message Passing (MP).

The Tanner graph (Fig.1) can be drawn directly from the H matrix (given in (8)) as shown be below:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{8}$$

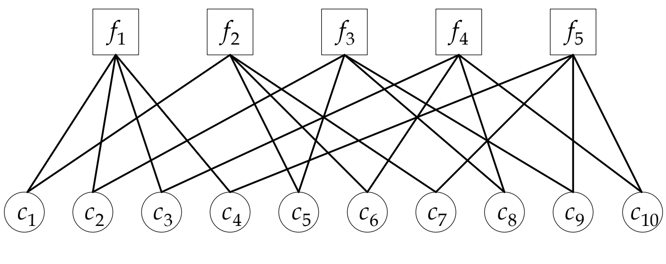

Fig. 1. Tanner graph of the *H* matrix given in (8)

The graph contains *m* check nodes (number of parity bits) and *n* variable notes (number of bits in a codeword). Check node f_i is connected to a variable node c_i if the element h_{ij} of H is a “1”.

In the Log-domain Sum-Product algorithm, the message passes between check nodes and variable nodes. In each pass the log likelihood ratio (*LLR*) is recorded for is probability of its likely symbol. In summary, the decoder goes through 5 steps as follows:

Step 1:

Compute the initial value of $L(q_{ij})$ transmitted from the variable node *i* to check node *j*; for all $i; 1 \le i \le n$.

$$L(q_{ij}) = L(c_i) = \frac{2y_i}{\sigma^2} = LLR_i = \log\left(\frac{p_{(c_i=0)|y_i}}{p_{(c_i=1)|y_i}}\right) \tag{9}$$

Where $L(c_i)$ denotes log likelihood ratio

σ^2 denotes derivation of white noise

$p_{(c_i=x)} \mid y_i$ denotes probability for given input y_i

Step 2:

Compute $L(r_{ji})$ transmitted from the check node *j* to variable node *i*; for all $i; 1 \le i \le n$. Let $\phi(x) = \log\left(\frac{e^x+1}{e^x-1}\right)$.

$$L(r_{ji}) = \prod_{i' \in V_{j \setminus i}} \alpha_{i'j}\, \phi\left(\sum_{i' \in V_{j \setminus i}} \phi\left(\beta_{i'j}\right)\right) \tag{10}$$

Where $\alpha_{ij} = \operatorname{sgn}\left\{L\left(q_{ij}\right)\right\}$, and $\beta_{ij} = \left|L\left(q_{ij}\right)\right|$.

Step 3:

Modify $L(q_{ij})$ and used it as the data transmitted from the variable node *i* to check node *j*; for all $i; 1 \le i \le n$.

$$L(q_{ij}) = L(c_i) + \sum_{j' \in C_{i \setminus j}} L\left(r_{j'i}\right) \tag{11}$$

Step 4:

Compute the soft output.

$$L(Q_i) = L(c_i) + \sum_{j \in C_i} L\left(r_{ji}\right) \tag{12}$$

Step 5:

The soft output obtained in step 4 is then used in the hard decision as,

$\hat{c}_i = 1$ if $L(Q_i) < 0$, otherwise $\hat{c}_i = 0$.

III. DESIGNS OF PARITY CHECK MATRICES

Resulted design of a H matrix is a crucial step in obtaining the code performance. A design of the parity check matrix outlined in this paper is based on the adjustment of the modified array LDPC and interleave-modified array LDPC.

A. Some Previous Works

Fan [8] has introduced the array structure parity matrix that can offer comparable performance when compared to a random generated parity matrix. Some other features are: low noise floor and no existence of cycle of 4. The said matrix is shown below.

$$H(p,j,k) \triangleq \begin{bmatrix} I & I & I & \cdots & I \\ I & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\ I & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & \alpha^{j-1} & \alpha^{2(j-1)} & \cdots & \alpha^{(j-1)(k-1)} \end{bmatrix}_{(jp\times kp)} \quad (13)$$

Where I is an identity matrix $(p\times p)$.

α is a position permutation matrix ($p\times p$).

This yields the code rate of $R = 1-(\frac{pj-j+1}{p^2})$.

Eleftheriou and Olcer [9] have proposed the modified array structure (MAC) by applying cyclic shift to Fan's array. This structure yields the code rate of $R = 1-(j/k)$. MAC offers superior performance to Fan's array as it can reduce number of "1" in the lower triangle. The effort also led to a simpler encoder.

$$H = \begin{bmatrix} I & I & \cdots & I & I & \cdots & \cdots & I \\ 0 & I & \alpha & \cdots & \alpha^{(j-2)} & \alpha^{(j-1)} & \cdots & \alpha^{(k-2)} \\ 0 & 0 & I & \cdots & \alpha^{2(j-3)} & \alpha^{2(j-2)} & \cdots & \alpha^{2(k-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & I & \alpha^{(j-1)} & \cdots & \alpha^{(j-1)(k-j)} \end{bmatrix}_{(jp\times kp)} \quad (14)$$

Singhaudom *et. al.* [10] has proposed the interleaved modified array LDPC or IMAC by introducing the quasi-cyclic matrix into the cyclic shift of (13). This interleaved LDPC of which the parity matrix given below is superior to Fan's array LDPC when the block length is particularly long.

$$H = \begin{bmatrix} I & I & I\omega & I\omega^2 & I\omega^3 & \cdots & I\omega^j \\ 0 & I & \alpha\omega & \alpha^2\omega^2 & \alpha^3\omega^3 & \cdots & \alpha^{(k-2)}\omega^j \\ 0 & 0 & I & \alpha^2\omega^2 & \alpha^4\omega^3 & \cdots & \alpha^{2(k-3)}\omega^j \\ \vdots & \vdots & \vdots & I & \alpha^3\omega^3 & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & I & \cdots & \alpha^{(j-1)}\omega^{(k-j)} \end{bmatrix}_{(jp\times kp)} \quad (15)$$

B. Our Construction

The implementation complexity can be greatly reduced significantly if a suitable structured parity check matrix [11] is assigned. In our construction, the matrix is based on Eleftheriou's and Singhaudom's works. We reduced "1" in the upper triangle by applying matrix transposition, and row and column interchanging to avoid cycle of 4. As a result we can have 4 forms of the new matrix. Unfortunately, the first two have less potential. The third offers the feature of generality and simplified encoding as well as the ability of error correction. The fourth has failed its decoding capability. We, hence, consider the third achieve in this paper. The procedure of obtaining the parity matrix is summarized below.

$$\begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \Rightarrow \begin{bmatrix} I & 0 \\ I & I \end{bmatrix} \Rightarrow \begin{bmatrix} I & I & I & 0 \\ 0 & I & I & I \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} I & I & I & I \\ 0 & I & I & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} I & \alpha & \alpha^2 & \alpha^3 \\ 0 & I & I & 0 \end{bmatrix}$$

Therefore,

$$H = \begin{bmatrix} I & \alpha & \alpha^2 & \alpha^3 \\ 0 & I & I & 0 \end{bmatrix} \quad (16)$$

1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

IV. Performance Evaluation

Upon preliminary investigation of long block length (such as 4096 bits) decoding, we found that our matrix cannot compete the IMAC one. But for the short and medium block length, the proposed matrix offers considerably good result. The test parameters are tabulated in table 1, 2 and 3 below.

TABLE 1: PARAMETERS FOR SHORT BLOCK TESTING

j	3	3	4	4
k	15	18	15	18
p	17	29	17	29
$R = 1-(j/k)$	0.80	0.833	0.733	0.778
$m = p(k-j)$	204	435	187	406
Parity = jp	51	87	68	116
$c = kp$	255	522	255	522
Iterations	5, 10, 20			

TABLE 2: PARAMETERS FOR MEDIUM BLOCK TESTING

j	3	3	4	4
k	31	43	31	43
p	37	47	37	47
$R = 1-(j/k)$	0.903	0.93	0.871	0.907
$m = p(k-j)$	1,036	1,880	999	1,833
Parity = jp	111	141	148	188
$c = kp$	1,147	2,021	1,147	2,021
Iterations	5, 10, 20			

TABLE 3: PARAMETERS FOR LONG BLOCK TESTING

j	4	4	5	5
k	61	47	61	47
p	67	89	67	89
R = 1-(j/k)	0.934	0.915	0.918	0.894
m = p(k-j)	3,819	3,827	3,752	3,738
Parity = jp	268	356	335	445
c = kp	4,087	4,183	4,087	4,183
Iterations	10, 20, 30			

The designed LDPC code offers the code rate of 0.833, 0.930 and 0.934 when applied to short block (j=3, c=522), medium block (j=3, c=2021) and long block (j=4, c=4087) respectively. With these parameters, BER are investigated and the obtained results are shown in Fig.2 and Fig.3 below.

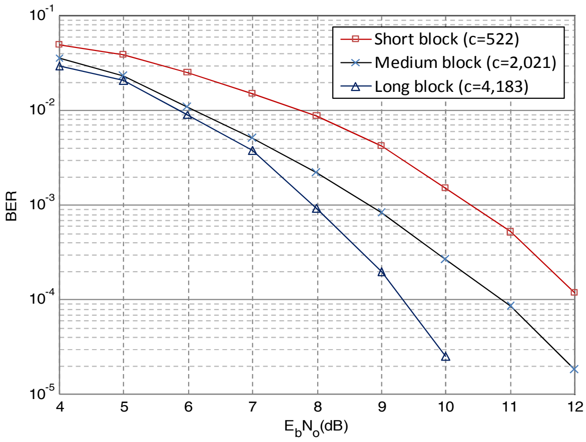

Fig. 2. Short block length and medium block length's performance

We also has compared our result to an existing published work proposed by Rakibul *et al* [1]. With similar block length, Rakibul's LDPC gives slightly better performance compared to ours. However, their code rate is only 0.5 whilst ours is 0.930. BER versus E_bN_0 plot is shown in Fig. 3 below.

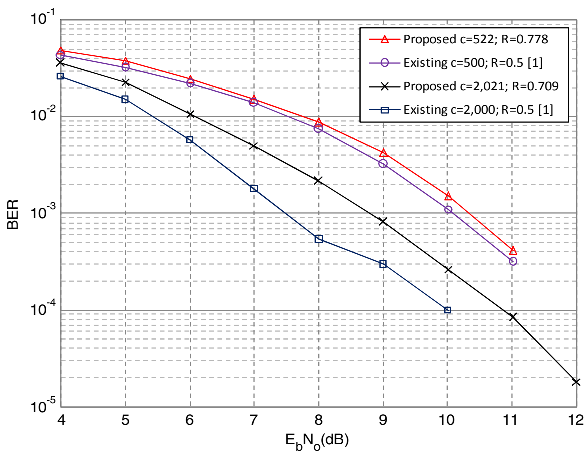

Fig. 3. The proposed LDPC and the existing LDPC's performance

In additions, we investigated number of iterations versus the performance obtained. Our LDPC codes were tested at 5, 10, and 20 iterations. At 5 iterations, BER of both short and medium block length are similar. BER can be improved when we increased the number of iterations to 10. When the number of iteration is of 20, only the BER of a medium block length can be slightly improved. However, the obtained BER is still not so low. This is quite common that one cannot get so low BER when the block length is not large.

V. CONCLUSIONS

Combining modified array with interleaved modified array by avoiding the cycle of 4, we can get a different parity matrix that leads to a moderate performance irregular LDPC code. This code has comparable performance compared with the regular one; in particular when the block lengths are of short and medium. For the long block length, the obtained LDPC cannot compete the existing the interleaved modified array. Based on our idea, there could be opportunity to develop a parity check matrix to obtain the performance of IMAC but with less complexity.

REFERENCES

[1] Mohammad Rakibul, Jinsang Kim, **"**On the use of QC-LDPC code for data transfer using short and medium block length," *Conference ICACT 2009* Feb. 15-18, 2009.

[2] R. Gallager, "Low-density Parity-check Code," *IRE Trans. Information Theory*, Jan. 1962, pp.21-28.

[3] Reed, I. S. and Solomon, G., "Polynomial Codes Over Certain Finite Fields," *SIAM Journal of Applied Math.*, vol. 8, 1960, pp.300-304**.**

[4] C Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting Coding and Decoding," *Proc. IEEE Int. Conf.Comm.*, pp.1064-1070, May 1993.

[5] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, "Design of Capacity-approaching Low-density Parity-check Codes," *IEEE Trans. on Info. Theory*, Vol. 47, pp.619-637, Feb. 2001.

[6] D.J.C. Mackey and R. Neal, "Near Shannon Limit Performance of Low Density Parity Check Code," *Electronics Letter*, Vol.33, Mar 1997, pp.457-458.

[7] R.M. Tanner, "A Recursive Approach to Low Complexity Code," *IEEE Trans. Information Theory*, Sept. 1981, pp.533-547.

[8] J.L. Fan, "Array Codes as low-density parity-check codes," *Proc. 2nd Int. Symp. Turbo Code*, Beit, France, pp. 543-546, Sep. 2000.

[9] E. Eleftheriou and S. Olcer, "Low-density parity-Check Codes for Digital Subscriber Lines," *Proc. 2002 Int. Conf. on Comm.*, pp.1752-1757, April-May 2002.

[10] W. Singhaudom, S. Noppankeepong, P. Suphithi**, "**Design of Hign-Rate Modified Array Codes for Magnetic Recording System," *ECTI International Conference***,** May 2007.

[11] O. Othman Khalifa, S. Khan, M. Zaid, and M. Nawawi, **"**Performance Evaluation of Low Density Parity Check Codes," *International Journal of Computer Science and Engineering*, Vol.2, No.2, 2008**.**