Advanced Science Letters
Vol. 23, 3754–3757, 2017

LDPC Code for OFDM Transmission using Bit Flipping and Sum Product Algorithm

Robby Sahputra, Dwi Astharini, Faisal
Department of Electrical Engineering, Universitas Al Azhar Indonesia, Jakarta

LDPC code is a powerful FEC coding scheme with good error performance under very low BER. For implementation in OFDM scheme, particular decoding algorithm is needed. The project is to implement FEC for OFDM and analyze the system performance with LDPC codes along with two decoding algorithm on OFDM over a AWGN channel. Bit flipping (BF) and sum product algorithm (SPA) are compared for their decoding performance. The parameter used for observation are BER over EbN0. The results obtained shows that the LDPC codes on OFDM can reduce error 1.5 % in average compared with without LDPC. Also that SPA decoding can perform better than the BF.

Keywords: LDPC, OFDM, Bit Flipping, Sum product algorithm

1. INTRODUCTION

Forward error correction (FEC) is a process adding extra symbols to the message to be transmitted and use symbols in the decoding process to correct the errors introduced by the channel. FEC is implemented in form of turbo codes, conventional code, or LDPC.

Low Density Parity Check (LDPC) code is a powerful FEC coding scheme that can achieve good error performance or very low BER. Specifically, data is transmitted by a sequence of pulses, and the system must ensure that these pulses are received with a acceptable rate of error. LDPC was first proposed by Gallager in 1962[1], but its potential remained undiscovered for many years, due to the computational demands for simulation. The field of forward error correction was by then dominated by highly structured algebraic block and convolutional codes.

Basically there are two possibilities to represent LDPC codes. As a linear block codes it can be described via matrix, and with graphical representation[2]. The characteristics of today's signal processing makes it likely to implement LDPC the matrix representation, while using the graph as an illustrious means.

For the decoding, various applicable algorithms were discovered independently several times and comes under different names. This paper worked with two of those algorithms. The bit flipping (BF) is based on a hard decision between 0 and 1 for every bit received, while sum product algorithm (SPA) employs iterative method of decoding[3].

This work has implemented LDPC and test its performnace on OFDM system, by using two variations of decoding algorithms.

LDPC for OFDM has commended considerable attention from many related researchers, particularly in conjunction with optical communication. The urge for next generation optical network led to the development of optical OFDM. And the different nature of signals and disturbance for this needs gave rise to importance of FEC i.e. LDPC[4,5]. This work is part of continuing project on software defined communication which previously weighted more on OFDM [6,7], and is currently directed toward OOFDM and SDOT.

This paper is organized as follows. Section 2 present the LDPC methods, along with the decoding algorithms. In section 3, implementation and obsevation method is given. The simulation results are presented in section 4. Finally the work is summarized in the last section.

[*] Email Address: astharini @uai.ac.id

2. LDPC CODE AND DECODER

We begin by defining LDPC and its generation [8,9,10]. One can define LDPC to matrix with dimension m x n. The rows and column *m,n,* are associated with the check nodes and bit nodes[11]. For the example matrix (3,6), we can look the figure 1.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

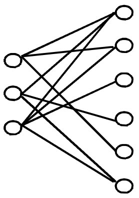

Fig.1. LDPC Representation with matrix (left) and Tanner graph (right)

Initially parity check matrix is generated, and used to create generator matrix by Gaussian elimination method. Two types of parity matrices in LDPC coding are Regular and irregular matrices. Regular matrix is one with fixed number of column weight *Wc* and row weight which is given by *Wr* = *Wc*(*n*/*m*). Irregular matrix has columns and rows with varying weights, meaning the bit nodes and check nodes are not fixed. In this project, we used irregular parity check matrix, having been reported to have better performance than the regular ones [12].

In implementation, the matrix dimension is based on the system, i.e. how the matrix is used on further process. Let us consider a 3x6 parity matrix as shown below.

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \qquad (1)$$

The parity matrix is later to be used to check the codeword (information signals). Therefore it is accommodating to be presented in the standard form as in formula 2 [2], and Gauss Jordan elimination method suit the puspose well.

$$H = [A \mid I_{n\text{-}k}] \qquad (2)$$

The resulting parity check matrix in its standard form is as shown below:

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \qquad (3)$$

Obtained parity matrix *H* is then translated to standard form generator matrix as in formula 4.

$$G = [I_k | A^T] \qquad (4)$$

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \qquad (5)$$

The information bits *U* are then encoded by multiplying it with the generator matrix above, to obtain the codeword *C* as given in formula 6.

$$C = U\,G \qquad (6)$$

A valid codeword can be verified using formula 7. If the equation results in nonzero, it is identified that there is error bit within the message.

$$H\,C^T = 0 \qquad (7)$$

For LDPC decoding, as said this works used two common schemes, i.e. the bit flipping and sum product algorithm.

Bit flip decoding is based on a hard decision (‘0’ or ‘1’) for every bit received. Part of significant of decoding bit flip is message passing between node on Tanner graph. Bit flipping or hard decision decoding is described as follows[3]:

1. Initialization: every variable node is a receive channel and sending message to check node with connection,in which case a value is assigned in the tanner graphic.
2. Parity update: with the use of variable node, every node checked the equation needed to be completed by parity check. When all equation is complete, algorithm will stop. Every check node also sends message to variable node through parity check concerning the status, whether it is complete or not.
3. Variable update; if the majority of data received are not completed, variable node change is activated, i.e. flipped. And then back to the second step, when total iteration maximum much and codeword not yet valid, then algorithm stop and the message is fail to converge report.

The sum product algorithm is an iterative process, whereby each iteration is divided into two half iteration processes. The first half includes processing the received message (v nodes) and sending the output to the connected nodes (c nodes). The second half includes the c node which process the received message and sending the output back to the v nodes connected [13].

Using the log version of the sum product algorithm, the probabilities calculations are performed as follows [3].

Count probability value priorilog - likehood ratios from bit received.

$$L(q_{ij}) = L(v_i) = \frac{2y_i}{\sigma^2} = LLR_i \qquad (8)$$

Process from check node go in the variable node, count probability value extrinsic from a check node to-j to variable node to-i. Probability value extrinsic depend on probability variable node to-i’ each to connect to check node to-j except variable node to-i (*Lrji*).

$$L_{rji} = \log\left(\frac{1+\prod i\in B_{j},i\text{-}'\neq i\,\tanh\left(M_{j,i'}/2\right)}{1-\prod i\in B_{j},i\text{-}'\neq i\,\tanh\left(M_{j,i'}/2\right)}\right) \qquad (9)$$

Combination probability value intrinsic and extrinsic from every variable node, and than count value total LLR from variable node to-i. If total LLR from negative channel, so value bit ‘1’, and otherwise when the total LLR from positive channel, so value bit ‘0’.

$$L_{qij} = r_i + \sum j \in A\, L_{rji} \qquad (10)$$

With use value from LLR total from every variable node, every check node checked what parity check equation completed ($s=cH^T$). If amount parity check equation is complete, algorithm will be stop. If not, process will be continue for the next iteration.

For sent back value L_{ji} to every information check, message from variable node to-i to check node to-j is sum of Lq_{ij} without component $Lr_{j'i}$ for received from check node to-j.

$$LQ_i = r_i + \sum j \in A\, L_{rj'i} \qquad (11)$$

3. CODE IMPLEMENTATION

LDPC encoder is implemented on according to the flow in Figure 2, with consideration of [14]. The steps starts with initiolization which basically where the parameters are declared or determined, among others are the matrix dimensions. The of the process steps are as explained in part 2.

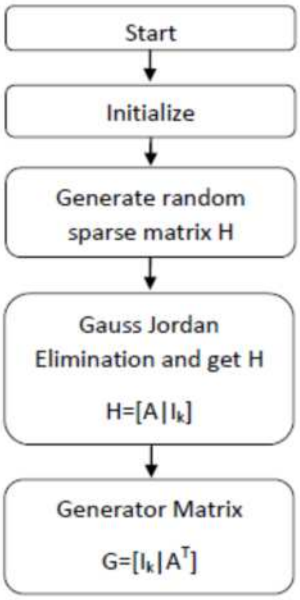
Start
Initialize
Generate random sparse matrix H
Gauss Jordan Elimination and get H
H=[A|I_k]
Generator Matrix
G=[I_k|A^T]

Figure 2. LDPC implementation

The LDPC process is then tested as part of transmission system, i.e. OFDM. First it was tested as a pre-process in PSK modulation scheme, as a step toward OFDM implementation scheme.

For the decoding part, the SPA and bit flip algorithm were implemented as shown in Table 1 and 2 [3,8].

Table 1. Bit Flipping implementation

```
Received vector
Set maximum iteration & hard decision
Check node
    $C H^T = 0$
Calculate BER
```

Table 2. SPA implementation

```
Initialize
    Variable update, Lrji
        Check update, Lqij
    Code Estimation
    Syndrome Check $C H^T = 0$
Calculate BER
```

The implementation on OFDM system was as shown in Figure 3, continuoing the previous work[6], aligned with other references[15,16]. The data was fed to the system as series of 256 bit packet, thus the bit error rate was observed each of those packet, and averaged for one transmission.

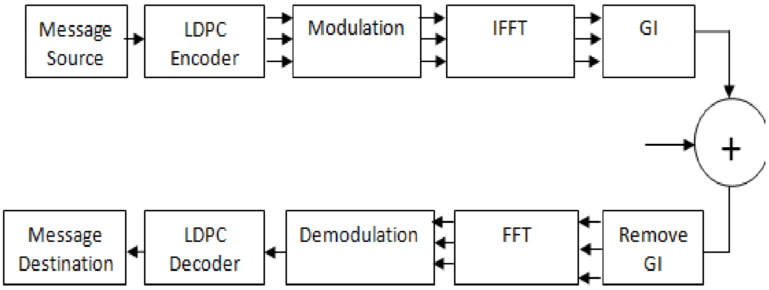

Figure 3. OFDM system scheme testing

The parameter for the data collecting are listed in table 3. Some were set i.t the modulation, channel and the LDPC matrix dimension. The noise level was varied for both methods of decoding.

Table 3. Testing parameters

Modulation	BPSK
Channel	AWGN
LDPC Matrix	Rows =190 and Cols=576
Decoding	1. Bit Flipping Algorithm 2. Sum Product Algorithm
Maximum number of iterations	100
Eb/No	2-5

4. IMPLEMENTATION RESULT

Figure 4 shows the error performances for LDPC implementation with both the bit flipping and SPA decoder over AWGN channel. The y - axis represents the bit error rate, which is the ratio of the number of bits incorrectly received to the total number of bits sent within a specified time interval. For a given communication system, the bit error ratio will affected by the data transmission rate. The results comparison analysis was performed.

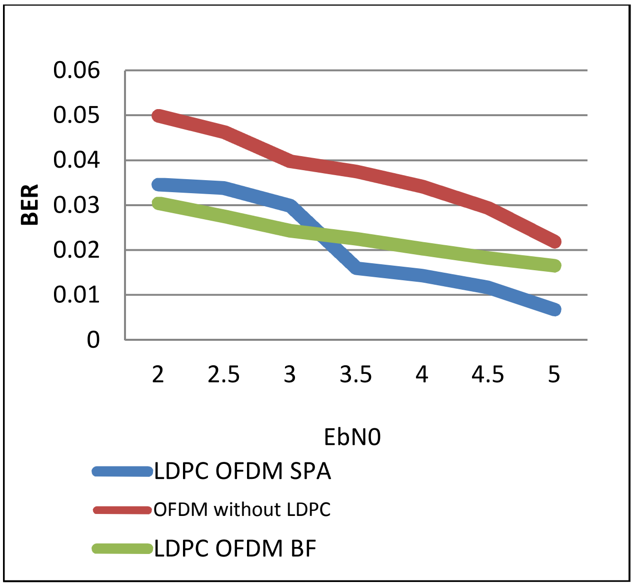

Figure 4. BER result of OFDM with various LDPC

Comparison also made, between OFDM without LDPC and with LDPC on OFDM using BFA and SPA decoding. The figure 2 shows that Bit Error Rate (BER) in variation EbN0. The BER of the LDPC on OFDM modulation is lower than the BER for OFDM without LDPC. This shows that using LDPC code is a more efficient code. The implementation of LDPC in simulation results that SPA Decoding is better than BFA Decoding because SPA has a lower BER than BFA Decoding.

5. CONCLUSIONS

The simulation with higher value of Eb/No produces the better result. The results obtained shows that the LDPC codes on OFDM can reduce error 1.5 % in average compared with without LDPC. The lowest BER average value of LDPC on OFDM over AWGN channel is 0.008. The highest BER average value over AWGN channel is 0.0434.

SPA Decoding is better than BFA Decoding because SPA has a 1.5 % lower BER than BFA Decoding.

Further investigations on higher Eb/No is to be done to confirm the pattern seen between the two decoder. Also investigation for transmission over multifarious channel such as Rayleigh or Rician.

ACKNOWLEDGMENTS

This work was supported by Indonesian Ministery of Research and Higher Education under Hibah Bersaing Grant 2016.

REFERENCES

[1] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," in *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966-2984, Dec. 2004

[2] Manjunatha P.N, T.S. Bharath Kumar and M.Z.Kurian, “Architecture for Low Density Parity Check Encoder”, International Journal of Application or Innovation in Engineering & Management, Volume 3, Issue 3, pp.414-417. March 2014

[3] S. J. Johnson, “Introducing Low-Density Parity-Check Codes”. [Online]. http://sigpromu.org/sarah/SJohnsonLDPCintro.pdf

[4] Kachris, C.; Tzimpragos, G.; Soudris, D.; Tomkos, I., "Reconfigurable FEC codes for software-defined optical transceivers," Optical Communications and Networks (ICOCN), 13th International Conference on , vol., no., pp.1,4, 9-10, 2014

[5] Wei, J.L.; Geng, L.; Cunningham, D.G.; Penty, R.V.; White, I.H., "Comparisons between gigabit NRZ, CAP and optical OFDM systems over FEC enhanced POF links using LEDs," Transparent Optical Networks (ICTON), 14th International Conference on, pp.1,4, 2-5, 2012

[6] Dwi Astharini, T. Bastian, R. Mustika, O.N. Samijayani, R. Safitri, "Zero Padding and Cyclic Prefix for OFDM on Multipath Rayleigh Fading Channel," Journal of Mobile Multimedia JMM, Vol.11 No.3&4, pp.330-338, 2015

[7] W. Pradani, F. Mubarik, Dwi Astharini; “SDR UAI, Virtual Machine for a Software Defined Radio Toolpack,” Journal of Mobile Multimedia, Vol.11 No.3&4, pp. 339-345, 2015

[8] Bernhad M. J Leiner, “LDPC Codes a brief tutorial“, 2005. [Online]. http://www.bernh.net/media/download/papers/ldpc.pdf

[9] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," in *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619-637, Feb 2001

[10] R.G. Gallager, “Low Density Parity- Check Codes”, 1963. [Online]. Available: http://www.rle.mit.edu/rgallager/documents/ldpc.pdf

[11] Nguyen Thi Dieu Linh, Gang Wang, A. Barkana and G. Rugumira, "Performance and processing time enhancement of LDPC decoder using stopping node modified Sum Product algorithm," Communication Technology (ICCT), 2011 IEEE 13th International Conference on, Jinan, 2011, pp. 118-121

[12] V. Kulkarni and K. J. Sankar, "Design of structured irregular LDPC codes from structured regular LDPC codes," Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on, Hooghly, 2015, pp. 1-4

[13] William Shieh, Ivan Djordjevic, “OFDM for Optical Communications”, Academic Press, 2009

[14] Matthew Pregara and Zachary Saigh, “Low Density Parity Check Code Implementation”, 2013. [Online]. Available: http://ee.bradley.edu/projects/proj2013/ldpcci/deliverables/Final Paper_rev14.pdf

[15] HF Juwono, D Gunawan, “Prinsip-prinsip OFDM”, Penerbit Andi, Yogyakarta 2011

[16] Shieh, William, "OFDM for Flexible High-Speed Optical Networks," Lightwave Technology, Journal of , vol.29, no.10, pp.1560,1577, 2011