Optimization of Low Density Parity Check Based on Logdomain

Farah L. Joey[1], Dr. Jinan F. Mahdi[2], Dr. Bassem G. Al-saadi[1], Dr.Mahmood F. Mosleh[2]
[1]Al-Esraa University College/ faraaaah86@yahoo.com , dr.basimalsaady@gmail.com
[2]Middle technical University jinanf2008@yahoo.com

Abstract. The Low-Density Parity-Check Codes (LDPC) become the most important channel codes used to address the huge Bit Error Rate (BER) stim from highly new mobility of the next generation. An iterative decoder of LDPC which employed many algorithms that are different according to their performance and complexity. Logdomain decoder is one the most technique involve for LDPC decoding according to its better performance and acceptable complexity. In this research, a system including Logdomain decoder is simulated and evaluated in terms of BER against SNR. The results show that such decoder with optimum parameters is better than hard decision decoder which needs only half the amount of SNR that consumed by hard decision.

Keywords: LDPC, Logdomain, hard decision decoder.

1 Introduction

Recently, one of the most important linear block codes is the Low-Density Parity Check (LDPC), which proposed by the thesis of Gallager in 1962 [1]. LDPC got more attention by many researchers because of its capability to correct great errors perfectly and leads to more reliability. It outperforms other iterative decoders like turbo codes for modern communications due to its success in responding to the requirements of next generations like Fifth Generation (5G). By using LDPC, it can be simply obtained any code rate and block length by specifying the form of the matrix of parity check. In addition, the guaranty of the codeword can be validated easily by its own parity checks matrix [2], [3]. On the other hand, there are many algorithms for decoding such code, it is mainly divided into two categories soft and hard decision, one of the familiar decoders is the log domain [4].

LDPC codes are widely interested by many researchers because of their flexibility like [5] who has proposed an improved algorithm of the weighted method depending on the bit-flipping. This proposal aims to address the drawback of conventional algorithms by decreasing the number of iteration. In other words, the number of iterations will be terminated according to good results without needed to approach the maximum of iteration which leads to more delay without improving its behaviour. In [6], a practical scheme of quasi-cyclic for LDPC in order to reduce the complexity of message passing decoder. Their proposal got acceptable Bit Error Rate (BER) with suitable data rate and length for the next generation. Also, the authors of [7] are proposed a modified LDPC decoder for useful optical telecommunications. They deployed a simple scheme of binary phase-shift keying modulation. The results of the simulation show that the Bit Error Rate (BER) approach zero at 0.5 Signal to Noise Ratio (SNR). On the other hand, [8] proposed a modification log –domain decoder by simplifying the equation of parity

IMDC-SDSP 2020, June 28-30, Cyberspace

DOI 10.4108/eai.28-6-2020.2298222

check using permutation which is unlike the current log domain. Mathematically it is an equivalent LDPC decoder based on log domain but it has advantages in terms of complexity.

In this paper, a communication system included LDPC is simulated by Matlab package to evaluate its performance with the Additive White Gaussian Noise (AWGN) channel. The decoder used in this paper is a log domain for its low complexity of the decoding circuit. The parameters will be selected to achieve prefer performance which depends on the amount of BER against Signal to Noise Ratio (SNR).

2 Low-Density Parity-Check Code

LDPC is represented by a special matrix (H) called parity check having (N) rows and (M) columns with a code rate of $R \geq \frac{N-M}{N}$ and $N > M$. Such a matrix has a low number of ones in each row (Wr) and (Wc) ones in each column. The codeword C must be satisfied with the condition [8]:

$$CH^T = 0 \,. \tag{1}$$

Other representations of LDPC can be represented by the tanner graph illustrated in Fig. 1 which consists of M bit nodes and N check nodes with a 4 and 2 ones in each row and column respectively.

The main part of LDPC is the decoding procedure which divided into two main categories; hard and soft decision. The bit flipping is wider used as hard decision while the Sum

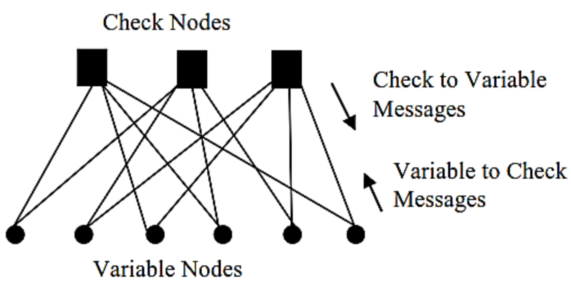

Fig.1. Tanner graph LDPC representation.

Product Algorithm (SPA) is the preferred one for soft decisions. In this paper, a log domain which is one of SPA types will be concerned to show its potential for error-correcting.

3 Log Domain Algorithm

This algorithm uses the Log-Likelihood Ratio (LLR) instead of real probability which applied in Prob domain algorithm. Log domain algorithm can be summarized as follow [10]:

i- Step 1: information from i^{th} variable nodes to j^{th} check nodes are initialized as LLR $L(p_i)$ denoted as $L(q_{i,j})$, can be calculated as:

$$L(p_i) = L\left(q_{i,j}\right) = log\,\frac{p_i^0}{p_i^1} = \frac{2}{\sigma^2}y_i \,. \tag{2}$$

Where p_i^0 and p_i^1 are the probabilities of zero and one respectively.

ii- Step 2: Back information from j^{th} check nodes to i^{th} variable nodes are calculated as LLR (variable node process):

$$L(r_{j,i}) = 2.tan^{-1}\left(\prod_{i' \in \frac{row[j]}{\{i\}}} \tanh\left(\frac{L(q_{i',j})}{2}\right)\right). \quad (3)$$

Where $i' \in \frac{row[j]}{\{i\}}$ which mean that the indices i' $(1 \leq i' \leq n)$ for all received bits in j $(1 \leq j \leq m)$ which have value one.

iii- Step 3: The messages transferred from bit nodes to the check nodes are to be calculated as LLR (check node process).

$$L(q_{i,j}) = L(p_i) + \sum_{j' \in col[i]/\{j\}} L(r_{j,i}) . \quad (4)$$

iv- Step 4: Calculate the extrinsic LLR values of decoder output bits and the hard diction is made according to Q_i value.

$$L(Q_I) = L(p_i) + \sum_{j' \in col[i]/\{j\}} L(r_{j,i}) . \quad (5)$$

$$\hat{c_i} = \begin{cases} 1 & if\ Q_i^1 > 0.5 \\ 0 & Otherwice \end{cases}$$

v- Step 5: Syndrome check.

$$\hat{c_i} \times H^T = \hat{S} . \quad (6)$$

If $\hat{S}$ is not equal to zero vector this means that the received codeword must be repeated the algorithm beginning with step 2 until it arrived at the maximum number of iterations. Otherwise ($\hat{S} = 0$), the received codeword is decoded correctly and the iteration is terminated.

4 Results

The first step of the simulation is to obtain the preferred number of iteration for our simulation. The number of frames in this step is (20), while the code rate is 1\2 (2000 rows, 4000 columns). Figure (2) illustrates the effect of the number of iteration for each round.

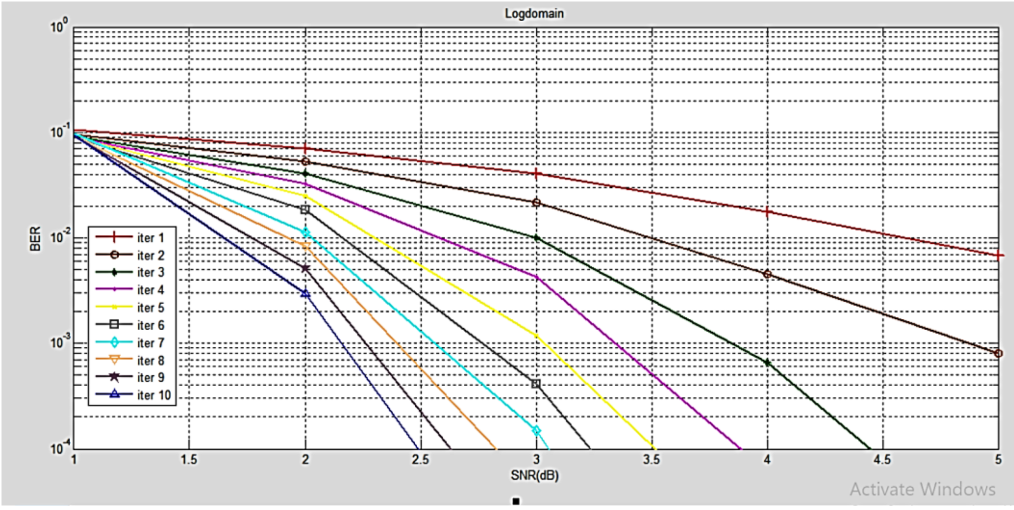

Fig. 2. The performance of Logdomain decoder with variable iteration.

It is clear that the performance of log domain decoder is improved by increasing the number of iteration. But as it is known, increasing the number of iterations more and more leads to more delay time which is not suitable for real-time communications. So we suggest that the number of iteration must not exceed 5 iterations, which is deployed in all the rest simulations. The second step of the simulation is to illustrate the effect of the number of variations of 1s per column. In this simulation number of 1s has been changed from (1-5) ones for each column, the results are shown in figure (3)

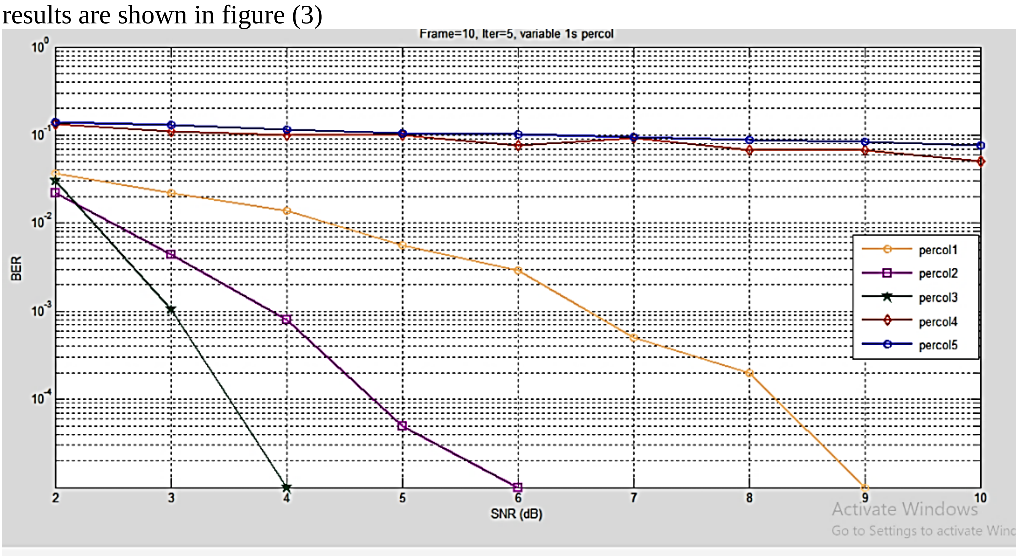

Fig.3. The effect of the number of 1s per column variation

The results of step 2 show that the performance is improved significantly by increasing the number of 1s per column from (1-3) while huge degradations are occurred when the number of

1s are exceeded 3. So that we suggest that better performance can be achieved with 3 ones per columns

Finally, a comparison between Logdomain, Probdomain and Bitflipping are applied to highlight the robustness of logdomain decoder compared with other decoders. Figure(4) shows the results of Brobdomain, Logdomain and Bitflipping.

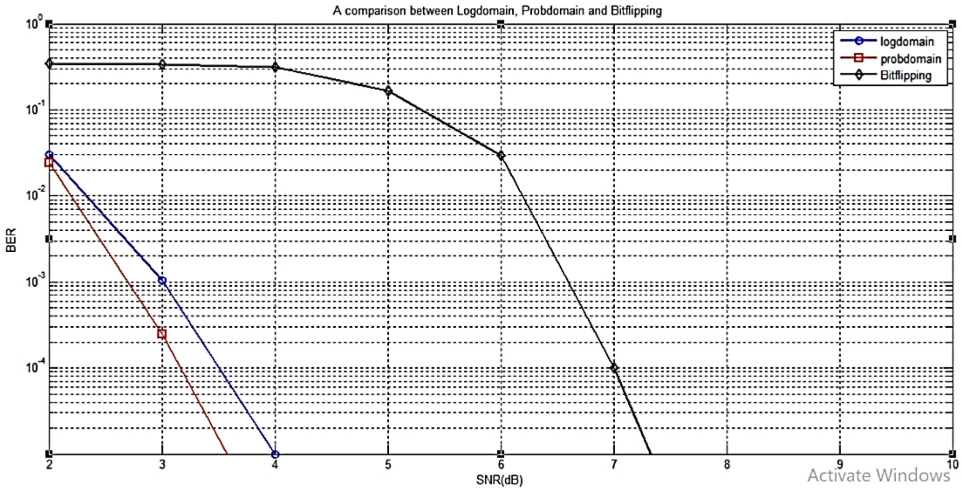

Fig.4. A comparison between Logdomain, Probdomain and Bitflipping

From the last results, it is clear that the Probdomain is outperformed the Bitflipping by 3.5 dB of SNR at 10-4 of BER, while it slightly differs from Logdomain about 0.3 dB of SNR.
In contrast, in terms of complexity, the Bitflipping which is a hard decision decoder is more simple than soft-decision (Probdomain and Logdomain), but it does not match the amount of the gain between them. On the other hand, the Log domain is more stable from the other types of decoders.

5 Conclusions

In this paper, the performance evaluation of Logdomain decoder for LDPC Codes is applied. A system included LDPC Codes with a decoder of Logdomain is simulated using MATLAB package. For simplicity, AWGN channel is employed to estimate Logdomain decoder as compare with other decoders. The results show that the Logdomain is better than Bitflipping decoder which is needed only half SNR that used for Bitflipping, while it is closed to Bropdomain decoder. In terms of complexity, the Bitflipping which is a hard decision decoder have a simplicity feature but it is not matched to the code gain of Logdomain. We expect that Logdomain will be used strongly of the next generation of communication because of its excellent performance to mitigate the higher BER.

References

[1] Gallager, R. G.: Low Density Parity Check Codes. pp. 1-6. MI Press, Cambridge, Mass, (1963)
[2] Peng, X., Goto, S.: Performance Analysis of Sphere Detection Combined with the LDPC Decoding. 5th International Colloquium on Signal Processing & Its Applications (CSPA). pp. 228-232 (2009)

[3] Mahmood F. M.: Evaluation of Low Density Parity Check Codes over Various Channel Types. Engineering and Technology Journal. Vol. 29. pp 1-11 (2011)
[4] Lanxun W., Weizhen L.: Quasi-orthogonal Space Time Block Codes (STBC) with full transmit rate Concatenated LDPC codes. The Eighth International Conference on Electronic Measurement and Instruments. pp. 3207-3210 (2007)
[5] Chang T. C. and Yu. T. Su.: Dynamic weighted Bit-Flipping Decoding Algorithm for LDPC Codes. IEEE Transections on Communications, [sc. IT]. pp. 3950 - 3963 (2015)
[6] Kiran K. G., et. al.: Decoding of Quasi-cyclic LDPC Codes Using an On-the-Fly Computation. IEEE Fortieth Asilomar Conference on Signals. Systems and Computers. USA. pp. 1192-1199. (2006)
[7] Liu Y., Olmos P. M. and Mitchell D. G. M.: On Generalized LDPC Codes for 5G Ultra Reliable Communication. 2018 IEEE Information Theory Workshop.15 Oct. pp. 1-25 (2019)
[8] Hossain Md., et. al.: Modified Log Domain Decoding Algorithm for LDPC Codes over GF (q). Journal of Selected Areas in Telecommunications, June Edition, (2011)
[9] Wymeersch H., Steendam H., Moeneclaey M.: Log-Domain Decoding of LDPC Codes over GF (q). IEEE International Conference on Communications. France. pp. 772-776 (2004)
[10] Hagenauer J., Ofer E. and Papke L.: Iterative decoding of binary block and convolutional codes. IEEE Trans. Inform. Theory. vol. 42. pp. 429–445, Mar. (1996)