International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

LDPC Codes Based On Fullerene Graphs

Manisha Kumari
M.Tech Scholar, Marudhar Engineering College, Bikaner, India
manishapoonia16@gmail.com

Dr. Javed Khan Bhutto
Professor, Marudhar Engineering College, Bikaner, India
bhutto_javed786@yahoo.com

***Abstract* — In this paper, LDPC Codes based on fullerene graphs have been generated. Here we used extension process of fullerene graph to design the parity check matrix for the Low Density parity check Codes. We design the code by using this matrix and evaluated performance for AWGN, Rayleigh and Rician Channels. And it is found that the codes generated are fast in encoding and better in terms of error performance on AWGN Channel.**

***Keywords* — LDPC Codes, Fullerene Graphs, Parity Check Matrix, Incidence Matrix.**

I. Introduction

Modern digital communication systems often require error-free transmission. In 1948, the fundamental concepts and mathematical theory of information transmission were laid by C. E. Shannon [1]. Shannon perceived that it is possible to transmit digital information over a noisy channel with an arbitrarily small error probability by proper channel encoding and decoding. Low-Density parity-check (LDPC) codes were discovered by Gallager in early 1960s [2]. After being overlooked for almost 35 years, this class of codes were recently rediscovered by Mackay and Neal and Wiberg [4] and shown to form a class of Shannon limit approaching codes [5]. This class of codes decoded with iterative decoding, such as the sum-product algorithm (SPA) [2, 8], performs amazingly well for a lot of different channels. Since their rediscovery, LDPC codes have become a focal point of research for a variety of applications such as distributed source coding [6] and Forward error correction (FEC) [7].

It was proved in [9] that LDPC codes with column weight (number of 1's in code matrix column) $j \geq 3$ have a minimum distance that grows linearly with the block length (number of columns) n for given j and row-weight (number of 1's in matrix row) k and that the minimum distance for codes with $j = 2$ grows logarithmically with n. However, compared with $j \geq 3$ codes, codes with $j = 2$ are easier to implement and require less storage making them best for low complexity applications [9]. Column weight 2 codes can be generated from regular graphs called cages. LDPC codes based on cages were discussed in [10], and it was shown that the codes are suitable for magnetic storage devices [11].

One of the limitations of the codes generated by cages has limited code size which depends on the vertices and edges of these cages. Large size cages are very complex to design. In this paper we have used family of fullerene graphs to generate LDPC codes. Fullerene graphs have a property that it can be extended up to large number of vertices [12]. We have analyzed the time taken in encoding of LDPC Codes and its Bit Error Rate (BER) performance over AWGN, Rayleigh and Rician channels. A typical configuration of the code generated from fullerene graph has given better BER performance and also took less time for encoding. We have compared the results for all three channels. The rest of the paper is organized as follows, in section II a short introduction to LDPC Codes is giving; in section III parity check matrix from defected fullerene graph is generated. Performance evaluation is giving in section IV followed by conclusion in section V.

II. LDPC Codes

An LDPC code is a linear block code specified in terms of a sparse $m \times n$ parity check matrix H whose elements are 0 and 1. A (n, j, k) LDPC code represents a code of length n where H has j number of 1's in each column and k number of 1's in each row. For the case where all bit nodes have the same degree and all the check nodes have same degree, then it is known as a regular LDPC code [13]. For such a case the code rate can be given by $1 - \frac{j}{k}$. These degrees are different for the case of irregular LDPC codes where the irregularity is typically specified using two polynomials called bit node and check node degree profiles or degree distributions [14][15][16]. The name LDPC comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. Such a structure guarantees both: a lower decoding complexity and good distance properties [1].

For encoding, we have to generate the generator matrix G. To generate the G matrix, H matrix should be converted into systematic form $[P_{m\times k}\ I_{n-k}]$. This can be done by first converting H matrix into reduced row echelon form (rref) and then with some column permutations it is converted into systematic form. Generator matrix then can be easily generated as $[I_k\ P^T]$.

III. Design of Parity Check Matrix from Fullerene Graphs

The role of Parity Check Matrix is very important in the performance analysis of LDPC Codes. After the design of parity check matrix its respective generator matrix can be constructed. In this section the parity check matrix of the LDPC codes is generated with the help of fullerene graphs. A Fullerene graph is a cubic planar graph with all faces 5- cycles or 6-cycles. If the number of 5-cycles (pentagons) in a given Fullerene is p and number of 6-cycles (hexagons) is h. And if v be the number of vertices, and e be the number of edges then for Fullerene graph p = 12, $v = 2h + 20$ and $e = 3h+30$. Hence the size of the fullerene graph depends on the number of hexagons [5]. We use the extension process to design our parity check matrix.

Extending process: Let F be a fullerene with a hexagon face neighbored by pentagons. We may add a vertex to each edge of the hexagon to make all 6 neighboring pentagons, hexagon, and add an edge to each new vertex and finally join the ends of new edges to make a new hexagon. With this process, we get a new fullerene with 6 more hexagons. The new fullerene has the same property and we may do the process again to get new fullerenes. So, we can construct fullerenes with h = i + 6k hexagons, for i = 2, 3, 4, 5, 7 and k = 1, 2, 3,... [5].

Once fullerene graph is constructed its incidence matrix can be use as parity check matrix. After this corresponding generator matrix has to be generated. Here an approach is taken to generate code with the help of Extension of fullerene graph. First we take the fullerene graph with h=2 then we apply extension process on it as described above. The process is shown in Figure 1 (the extending process on h = 2 to get h = 8).

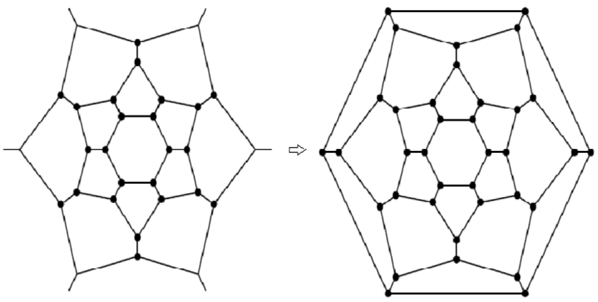

Fig.1. Extension of Fullerene Graph from h = 2 to h = 8.

Now we first design the incidence matrix. For designing the incidence matrix we first give numbering to each edge and each vertex. Now, we take nodes or we can say edges as the rows and vertexes as the columns. We put 1 at corresponding columns where a vertex connecting two edges. By this way we construct an incidence matrix.

For easy understanding here we first take fullerene graph with parameter i = 2 and k = 10 such that number of hexagons h = i + 6k = 64 and the code rate is 1/3. The matrix then use as parity check matrix for LDPC Codes.

The parameters of sparsity pattern are shown in table 1 where h=2.

Table 1: Parameters for sparsity pattern of Fig.2

Parameters	Values
i	2
k	10
$h = i + 6k$	62
$v = 2h + 20$	144
e = $3v/2 = 3h + 30$	216

Here Figure 2 shows the sparsity pattern of the Fullerene Graph generated from MATLAB with the above parameters.

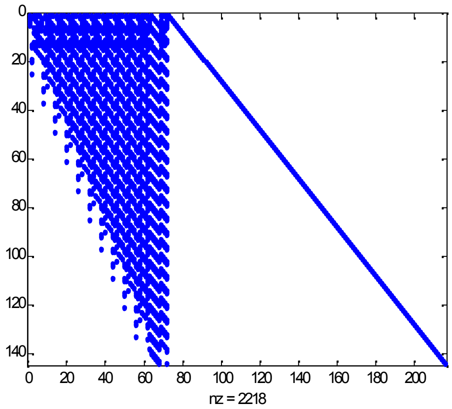

Fig.2. The sparsity pattern of the fullerene graph for h=2

Similar extension process is applied for h=4 and we get the sparsity pattern.

The parameters of sparsity pattern are shown in table 2 for h=4.

Table 2: Parameters for sparsity pattern of Fig.3

Parameters	Values
i	4
k	5
$h = i + 6k$	34
$v = 2h + 20$	88
e = $3v/2 = 3h + 30$	132

Fig.3 shows the sparsity pattern of the Fullerene Graph generated from MATLAB with above parameters where we take i=4 and k= 5.

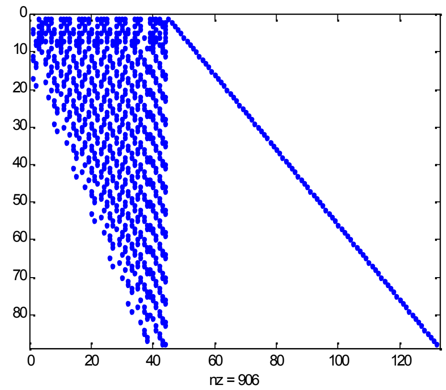

Fig.3. The sparsity pattern of the fullerene graph for h=4.

IV. Performance Evaluation

The LDPC Codes generated from fullerene graphs and defected fullerene graphs has been simulated in MATLAB for BPSK modulated signals passed through AWGN channel for different SNR. The LDPC encoder and decoder objects, modulator and demodulator objects, and the channel have been generated by inbuilt standard MATLAB commands. The variance (σ^2) of additive noise can be calculated from the expression [17] of SNR given by:

$$SNR(dB) = 10log_{10}\frac{a^2}{2R\sigma^2}$$

In this section, LDPC Codes based on Fullerene Graphs are discussed. The simulation results for the Fullerene graph (designed in previous section) are shown here. The number of iterations performed on decoder side was 20.

A. Results for h=2

The parameters for the fullerene graph are shown in table 3.

Table 3: Parameter for a Typical Fullerene Graph. (Results Fig. 4)

Parameters	Values
i	4
k	5
	34
H matrix	144×216
Message vector size	144
Code vector size	216

The description of the parameters are given in previous section

Figure 4 show the performance of LDPC Codes Fullerene graphs with parameters given in Table 3.

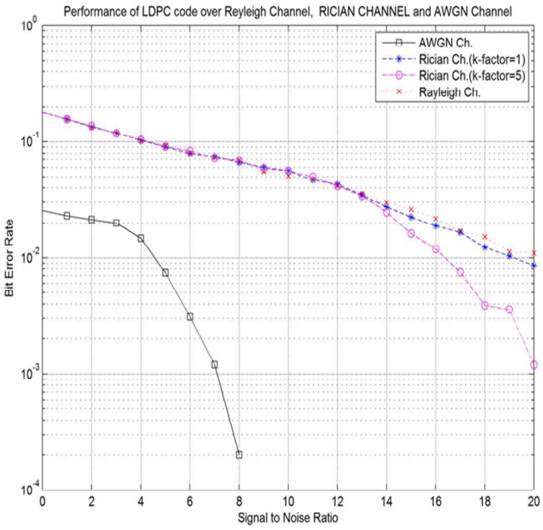

Fig.4. The simulation results on AWGN, Rayleigh and Rician Channel (k =1 and k = 5) for performance of LDPC codes over BPSK for h = 2.

From figure 4 on AWGN channel, BER of 10-2 reaches at approx. 4.2 dB SNR while it reaches at 20 dB on Rayleigh channel and for Rician channel, BER of 10-2 at approx 19 dB SNR with k-factor of 1 and with k = 5 BER of 10-2 achieve at approx 16 dB SNR. The performance on the Flat Rayleigh channel is quite poor then Rician with K = 5 and reaches 10-2 BER at SNR of 16 dB.

B. Results for h=4

The parameters for the fullerene graph are shown in table 4.

Table 4: Parameter for a Typical Fullerene Graph. (Results Shown in Fig. 5)

Parameters	Values
i	4
k	5
	34
H matrix	144×216
Message vector size	144
Code vector size	216

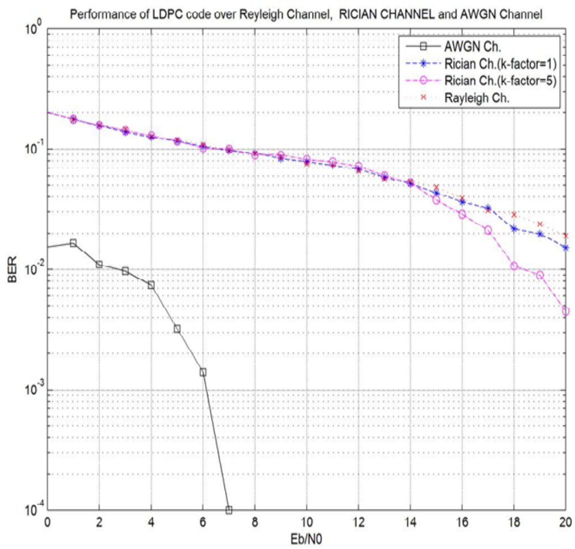

Fig.5: The simulation results on AWGN, Rayleigh and Rician Channel (k = 1 and k = 5) for performance of LDPC codes over BPSK for h = 4.

From figure 5 on AWGN channel, BER of 10-2 reaches at approx. 2 dB SNR while it reaches at 20 dB on Rayleigh channel and for Rician channel, BER of 10-2 at approx 19 dB SNR with k-factor of 1 and with k = 5 BER of 10-2 achieve at approx 18 dB SNR. The performance on the Flat Rayleigh channel is quite poor then Rician with K = 5 and reaches 10-2 BER at SNR of more than 18 dB

V. Conclusion

In findings it has been concluded that the methods shows best performance for AWGN channel and gives same performance for Rayleigh and Rician with k = 1 channel and quite better performance for Rician with factor k = 5.

References

[1] C.E. Shannon, “A mathematical theory of communication”, Bell Syst. Tech.J., pp. 372–423, 1948.
[2] R.G. Gallager,“ Low-Density Parity Check Codes”. Cambridge, MA: MIT Press, 1963.
[3] G. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error correcting coding: Turbo codes”, in Proc. 1993 Int. Conf. Comm., Geneva, Switzerland, May 1993, pp. 1064–1070.
[4] D.J.C. Mackay and R.M. Neal,“ Good codes based on very sparse matrices”, in Cryptography and Coding, 5th IMA Conference (Lecture Notes in Computer Science), C. Boyd, Ed. 1995, vol. 1025, pp. 110–111.
[5] Shu Lin and Costello, “Error Control Coding”, Pearson-Prentice Hall, 2004.
[6] Angelos D. Liveris, Zixiang Xiong and Costas N. Georghiades, “Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes”, IEEE Communications Letters, Vol. 6, NO.10, October 2002.

[7] E.A. Lee and D.G. Messerschmitt,“ Digital Communication”. Boston/Dordrecht/London: Kluwer, 1994.

[8] Amendment: Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-T, IEEE *Draft* P802.3an/D2.1.

[9] Hongwei Song, Jingfeng Liu, B.V.K.V. Kumar, *Low Complexity LDPC Codes for Partial Response Channels*, IEEE Globecom 2002, vol. 2, pp. 1294-1299, November 2002.

[10] Gabofetswe Malema and Michael Liebelt, *High Girth Column-Weight- Two LDPC Codes Based on Distance Graphs*, EURASIP Journal on Wireless Communications and Networking Volume 2007, Article ID 48158, 5 pages.

[11] G. Malema, and M. Liebelt, *Low Complexity Regular LDPC codes for Magnetic Storage Devices*, Proc. of World Academy of Science, Engineering and Technology, Issue 7, 2005.

[12] Rashid Zaare-Nahandi, *Some Infinite Classes of Fullerene Graphs*, International Mathematical Forum, 1, 2006, no. 40, 1997-2002.

[13] S. –Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Commun. Lett*., vol. 5, pp. 58-60, Feb. 2001.

[14] D. J. C. MacKay, “Good error-correction codes based on very sparse matrices,” IEEE Trans. Inform. Theory, vol. 45, pp. 399-431, Mar. 1999.

[15] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Improved low-density parity check codes using irregular graph,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 585-598, Feb. 2001.

[16] Y. Kou, S. Lin, and M. Fossorier, “Construction of low-density parity-check codes: A geometric approach,” in Proc. 2nd Int. Symp. *Turbo Codes*, Brest, France, Sept. 4-7, 2000.

[17] D. J. C. MacKay and R. M. Neal, *Near Shannon Limit Perfoemance of Low Density Parity Check Codes*, IEEE Electronics Letters Vol. 32 No. 18, pp. 1645-1646, 29th August 1996.