# Raspberry Pi based Wireless Transmission of Text Data using Low Density Parity Check (LDPC)

Swaraj Patil[1], Dr. D.P Rathod[2]
[1]*Student M. Tech Electronics,* [2]*Professor*
*Department of Electrical Engineering, VJTI, Mumbai, Maharashtra, India*

***Abstract*— During transmission of data, errors are generally introduced which corrupts the actual information. Low Density Parity Check codes is an error correcting codes that are widely been used. In this paper, LDPC is used as error correcting codes with parity check matrix of size (8 x 16) having code rate 0.5.**

**Text data is been encoded with addition of noise and transmitted. The decoder used here is Sum Product Algorithm which decodes the message correctly by removing the error. The hardware used here is Raspberry Pi which has Wi-Fi module for transmission and reception. Encoder and Decoder GUI has been made to show transmission and reception of message.**

***Index Terms*—LDPC, Encoder, Decoder, Error, Raspberry Pi**

## I. INTRODUCTION

Data is one of the important integrity of communication in today's world. Large amount of data transfer takes place from one place to another. When transmission of data takes place between two systems, the data gets contaminated due to addition of noise. The noise can introduce error in the binary bits (data that is transmitted). That means a bit '0' may change to bit '1' or a bit '1' may change to bit '0'. This error can become serious threat to the accuracy of the system. Therefore, it is necessary to detect and correct such errors. Channel Coding (Forward Error Correction) is used extensively to eliminate data errors. Various error-correcting codes has been developed. In this paper, Low Density Parity Check Codes (LDPC) has been used as error correcting codes.

LDPC are class of linear block codes. It was introduced by Robert Gallager at MIT in 1960 in his PhD thesis. They are linear block codes formed using Generator matrix G in an encoder and parity check matrix H at the decoder [1]. The parity check matrix has M rows and N columns where M represents check nodes and N represents variable nodes. This parity check matrix can be represented in the form of Tanner Graph. It also helps to describe decoding algorithm. As the name LDPC ,it has less number of 1's and more number of 0's. Thus, reducing the computational complexity of the system. It is a iterative decoding algorithm. It was shown that LDPC allows data transmission rates very close to the theoretical maximum (Shannon Limit). Due to this, LDPC code is suitable for applications that require a largeBandwidth, high reliability, and channel noises too high such as (DVB-S2). It has been found that LDPC codes have lower complexity in the decoding process compared to the other FEC. With advances in technology and computing power, they have been adopted in many high speeds communication standards.

In this paper limitation like Code rate is ½ and the size of the parity check matrix 8 x 16. To encode LDPC, Lower Triangular algorithm is used and the decoder is sum-product algorithm.

## II. ENCODING

### A. *Parity Check Matrix*

An LDPC code parity-check matrix is called $(w_c,w_r)$*regular* if each code bit is contained in a fixed number $w_c$ of parity checks and each parity-checkequation contains a fixed number $w_r$ of code bits [12]. In this paper, (8 x 16) regular parity check matrix is used. The parity check matrix is shown below.

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

Fig. 1 Parity Check matrix with size (8 x 16)

### B. *Tanner Graph*

LDPC codes can be represented by a *bipartite* graph called *Tanner* Graph, theterm bipartite refers to a set of nodes partitioned into two subsets in such a way thatall edges have a vertex in the first subset and another one in the second, and no edgesconnect nodes within the same subset [12]. The two subsets are called *check nodes*representing rows, and *variable nodes* representing columns of the LDPC parity checkmatrix. The Tanner graph for the above parity check matrix is given below.

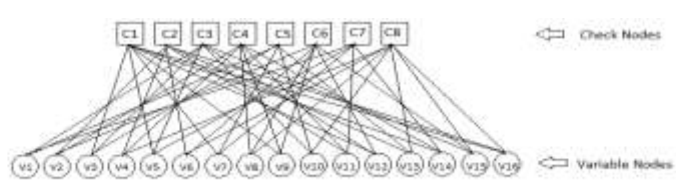



Fig. 2 Tanner Graph of Parity Check Matrix

### *C. Encode*

The LDPC based communication system block is shown below.

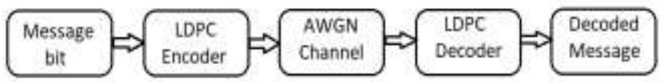
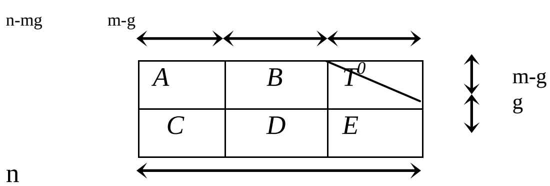



Fig. 3 LDPC based communication system block

LDPC encoder encodes the message bit (m-bits) and generates a code word of size (1 x n). Lower Triangular method is used here for encoding the data [4].The parity check matrix is of size (m x n) and the generated code word is of size (1 x n) where the multiplication of $H^T$* C= 0. Using only row and column permutation the parity check matrix is put into approximate lower triangular form.

$$H = \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}$$

Where the matrix T is a lower triangular matrix of size (m-g) x (m-g).A matrix (m-g) x (n-m), B (m-g) × g, T (m-g) x(m-g), C gx (n-m), D g x g and E g x (m-g). Each matrix issparse.




We know that in the code word C, u is message bit with the length (n-m), whereas p1 and p2 represents matrix that $p_1$= g and $p_2$ = (m-g).

$$C=[u \quad p_1 \quad p_2]$$

Once in approximate lower triangular form, Gauss–Jordan elimination is applied to clear *E* (make it zero). This is equivalent to multiplying *H* by

$$H_1=\begin{bmatrix} I_{m-g} & 0 \\ -ET^{-1} & I_g \end{bmatrix}\begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}$$

From $H_1$*$C^T$ =0 we get the following equations.

$$\boldsymbol{Au^T + Bp_1^T + Tp_2^T = 0}$$

$$\boldsymbol{(ET^{-1}A + C)u^T + (ET^{-1}B + D)p_1^T = 0}$$

Let ø **=(E $T^{-1}$ B+ D)** and the assumption ø is non-singular. After solving the above equation we get solution to give parity $\mathbf{p_1}$and $\mathbf{p_2}$as follows:

$$\boldsymbol{p_1^T = \emptyset^{-1}(ET^{-1}A + C)u^T}$$

$$\boldsymbol{p_2^T = T^{-1}(Au^T + Bp_1^T)}$$

## III. DECODING

The sum-product algorithm is a soft decision message-passing algorithm which is similar to the bit-flipping algorithm [12]. Bit-flipping decoding accepts an initial hard decision on the received bits as input whereas the sum-product algorithm is a soft decision algorithm which accepts the probability of each received bit as input. The flow chart for SPA is shown below.

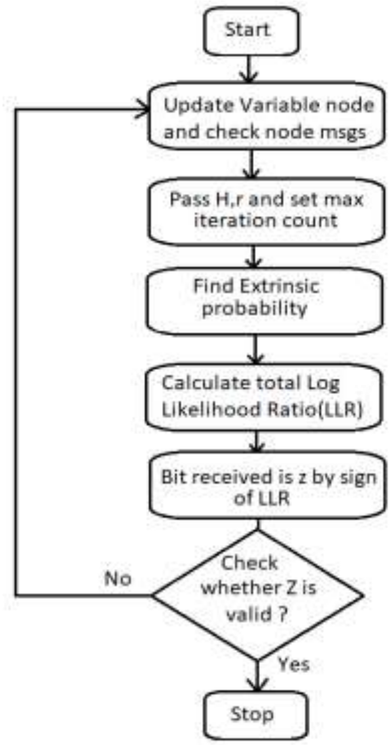



Fig. 4.Flow-Chart for Sum Product Algorithm (SPA)

## IV. METHODOLOGY

### *A. Proposed System*

The system designed here is capable of transmitting the text data. The text data is nothing but the stream of characters. Each character has its ASCII (American Standard Code for Information Interchange) value. This ASCII value is converted into 8-bit binary data.

The obtained 8-bit binary data is encoded to generate a code word of 16-bit by encoder. Encoder adds parity bits to the message bit to generate a desired code word.

A ⇒ 010000010100 ⇒ 00110110

(Character) (Binary value) (Encoded data)

Before the transmission of the data, random noise is added to encoded data (Code word). After the addition of error theerror code word is then transmitted to the recipient via transmitter.

At the receiver, the transmitted code word is received. The decoder checks whether the received code word has any error or not. If there are any error in the received code word, the decoder corrects those errors.

After obtaining the correct code word, message bits are extracted from it. These binary message bits are converted back to get their desired value (ASCII value).

010000010111 ⇨ 00100000100110110 ⇨ 01000001

(Error code word) (Decode code word) (Message)

A

⇨ (Character)

### *B. Hardware*

The hardware used in this system is the Raspberry Pi 3 model B. Two Raspberry Pi (R-pi) are used, one acts as a transmitter and the other as receiver. A Graphical User Interface (GUI) is developed at both the ends of communication. The GUI has been programmed in python 3. Also, the encoder and decoder are programmed in python 3.

The transmission takes place using Wi-Fi that is embedded in both R-Pi. Both the R-Pi are connected to the same network of Wi-Fi. Once they are connected to the network, then transmission and reception takes place using client server model.

## V. RESULTS

The image shown below is the Encoder GUI. The data that is to be sent is entered in “Input text”. The button “Convert” converts this text to binary data which is displayed in the text box “Binary Message”. The button “Encode” encodes this text by adding parity bits and the code word is displayed in the “Encoded Message” text box. The “Error” button introduces error in the code word. On pressing the button “Send” the text present in “Error Message” is transmitted to the receiver.

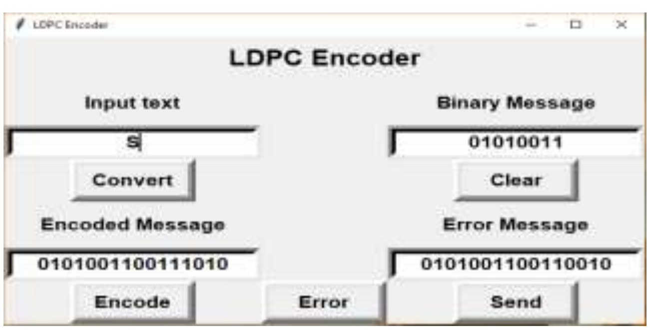



Fig. 5. Encoder GUI with character ‘S’

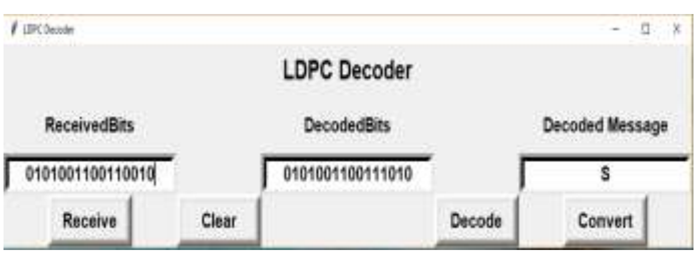



Fig. 6. Decoder GUI with character ‘S’

The image above shows the Decoder GUI. On pressing the button “Receive”, the transmitted text is received and displayed in the “Received Bits” text box. The button “Decode” checks whether received bits are in error or not. If error is present, then it corrects the code word. The corrected code word is shown in “Decoded Bits” text box. Now the button “Convert” will extract the message bits from the corrected code word and convert it back to ASCII. The converted message is shown in the “Decoded Message” text box. It is been clearly observed that the character ‘S’ has been correctly decoded at the receiving end.

Now, the input text to be transmitted would be a stream of characters. User can enter any input text. e.g. “hello world”.

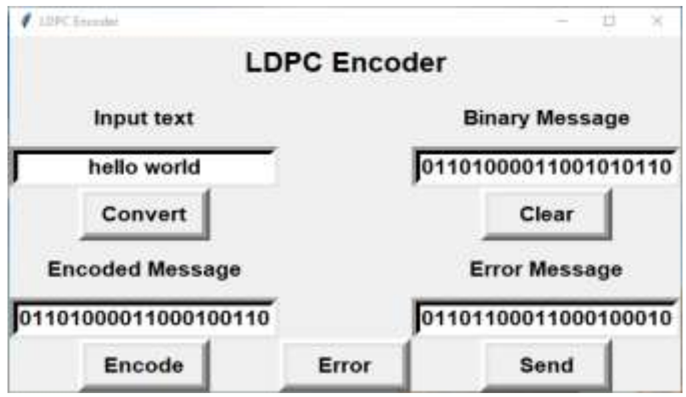



Fig. 7. Encoder GUI with stream of characters


Fig. 8. Decoder GUI with stream of characters

As it can be seen from the above Encoder and Decoder GUI, the transmitted text with error has been correctly been decoded by the decoder. The hardware of the system is shown below.

Fig. 9. Hardware of the system

## VI. CONCLUSIONS

The LDPC based encoder and decoder was successfully implemented on Raspberry Pi. Single bit character was transmitted with addition of error in to it. This was correctly decoded by the decoder. Also a stream of character was also transmitted with error. The LDPC decoder was able to successfully recover the correct data.

## REFERENCES

[1] Gallager, R.G, “Low Density Parity Check codes”, *IRE Transactions on Information Theory Engineering,* Vol. IT-8, January 1962.

[2] Heng Li, YanChunZhong,” A Novel Method for H.264 Video Transmission Using LDPC Codes over High BER Wireless Network”, 2nd International Conference on Power Electronics and Intelligent Transportation System,2009.

[3] Rui GUO, Lixin WANG, Xiaoxia JIANG,” Stereo Video Transmission Using LDPC Code”, I. J. Communications, Network and System Sciences, 3, 207-283, 2008.

[4] Rita Purnamasari, HeroeWijanto, IswahyudiHidayat,” Design and Implementation of LDPC(Low Density Parity Check) Coding Technique on FPGA(Field Programmable Gate Array) for DVB-S2 (Digital VideoBroadcasting-Satellite)”, IEEE International Conference on Aeroelectronics and Remote Sensing Technology ,2014.

[5] M.UshaswiniChowdary et al,“ZigBee Based Wireless Data Transmission with LDPC codes using FPGA”, International Journal of Engineering and Technology (IJET), Vol 8 No 2 Apr-May 2016.

[6] KavitaSogale,Dr.D.J.Pete,” Wireless Transmission of Karhunen-Loeve Transformed Images over A WGN”, IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS)2015.

[7] PradeepShah,PrakashVyavare and Anjana Jain,”Modern error correcting codes for 4G and beyond: Turbo codes and LDPC codes ” ,Radio and Antenna Days of the Indian Ocean (RADIO), 2015.

[8] KritikaPuri, Madhwendra N. Tripathi,” Designing and the Performance Estimation of Wireless Communication Systems using LDPC codes using Sum Product algorithm and Bit-flipping algorithm”, *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 4, Issue 8, August 2015.*

[9] WojciechSulek,”Seed Expansion for construction of structured LDPC codes”, International Symposium on Wireless Communication Systems (ISWCS) 2009.

[10] Timothy WeylinGerken,“Implementation of LDPC codes using the IEEE 802.16e standard”, College of Engineering and Mineral Resources at West Virginia University,2009.

[11] Rinu Jose and Ameenudeen P.E,”Channel Coding using Low Density Parity Check Codes in AWGN”, *International Conference on Emerging Trends in Technology and Applied Sciences(ICETTAS) 2015*

[12] Sarah Johnson,Chapter“Iterative error correction”, 2, ISBN-13 978-0-511-69129-4,E-book.

[13] “Low-density parity-check code”, *https://en.wikipedia.org/wiki/Low-density-parity-check*