

# Decoding Techniques of LDPC Codes

Devashree Marotkar<sup>1</sup>, Vivek Kapur<sup>2</sup>, Shital Bramhe<sup>3</sup>

Assistant Professor, Electronics & Telecommunication, Rajiv Gandhi college of Engg & Research, Nagpur, India<sup>1</sup>

Professor, Electronics & Telecommunication, Rajiv Gandhi college of Engg & Research, Nagpur, India<sup>2</sup>

Assistant Professor, Electronics & Telecommunication, Rajiv Gandhi college of Engg & Research, Nagpur, India<sup>3</sup>

**Abstract:** In this paper, we will Discuss Decoding technique of Low Density Parity check (LDPC) codes. Using Results we can evaluate which decoding procedure is better.

**Keywords:** Error Correcting Codes, Communication, Wireless Communication, LDPC.

## I. INTRODUCTION

Wireless Mobile Communication Now Demands Large Data Bandwidth To Accommodate Various Multimedia Services. Such A System Requires A Very High Speed Wireless Transmission Technique. A Wireless Channel Environment Is More Subject To Noise Than A Wired Channel Because The Signals Are Open To External Disturbances Such As Path Loss, Shadowing, And Fading Therefore, Channel Coding Is Inevitable For Wireless Communication. Channel Coding Has Been An Important Issue In Communication Systems. It Has The Ability To Detect And Correct Errors Caused By Noise In A Channel. The Bit-Error-Rate (BER) Can Be Reduced Without Increasing The Signal Power Since The Transmitted Data Carry Redundancies That Are Used To Detect And Correct Errors. This Coding Skill Is Useful In Transmission On Finite Power Channels Such As General Switched Telephone Networks [2]-[5].

LDPC codes are one of the hottest topics in coding theory today. Originally invented in the early 1960's, they have experienced an amazing ragged. comeback in the last few years. Unlike many other classes of codes LDPC codes are already equipped with very fast (probabilistic) encoding and decoding algorithms. These algorithms can recover the original codeword in the face of large amounts of noise. New analytic and combinatorial tools make it possible to solve the design problem. This makes LDPC codes not only attractive from a theoretical point of view, but also perfect for practical applications.

the data to be transmitted. The channel noise or interference might affect the transmitted data, changing some symbols. At the destination received data (source data plus noise) is demodulated and estimated using a pre-defined method defined by the decoder algorithm. Several error correction codes have been developed over time to encode and decode sent and received data respectively. The noise introduces errors in the transmitted data. Error detection and correction is achieved by adding redundant symbols to the original data. This realization has lead to the development of error correction codes (ECCs). Without ECCs data will need to be retransmitted if it could be detected that there is an error in the received data. Retransmission adds delay, cost and wastes system throughput.

## III. ERROR CORRECTING CODING

Error correcting coding (ECC) is a critical part of modern communications systems, where it is used to detect and correct errors introduced during a transmission over a channel [11], [12]. It relies on transmitting the data in an encoded form, such that the redundancy introduced by the coding allows a decoding device at the receiver to detect and correct errors. In this way, no request for retransmission is required, unlike systems which only detect errors (usually by means of a checksum transmitted with the data). The wide ranges of ECC applications [13] include space and satellite communications, data transmission, data storage and mobile communications.

## II. BASIC COMMUNICATION SYSTEM

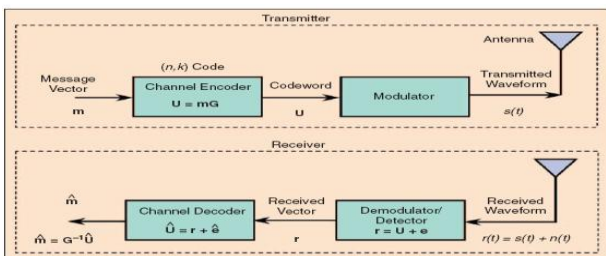


Figure 1. Block Diagram of Wireless communication System

Communication system diagram showing data movement from source to destination. Data from the source is encoded using an encoder based on error detection and correcting algorithm before it is modulated and sent through a channel. Encoding adds redundant symbols to

## IV. ERROR CORRECTING CODES

In telecommunication and information theory, forward error correction (FEC), also known as an error-correction code is a system of error control for data transmission, whereby the sender adds redundant data to its messages. Error correcting codes attach extra bits to the transmitted data. The extra bits are the redundancies which are then used to detect and correct errors on the received data. In a communication system that employs forward error-correction coding, a digital information source sends a data sequence to an encoder. The encoder inserts redundant (or parity) bits, thereby outputting a longer sequence of code bits, called a codeword. Such code words can then be transmitted to a receiver, which uses a suitable decoder to extract the original data sequence. Forward error-correction coding (also called channel

coding) improves data reliability by introducing a known structure into a data sequence prior to transmission or storage.

## V. VARIOUS TYPES OF ERROR CORRECTING CODES (ECC)

- Viterbi codes
- Convolution codes
- Bose-Chaudhary-Hocquenghen(BCH)
- Reed Solomon codes
- Turbo codes
- LDPC codes

All the above error correcting codes have a different correcting performance, computation technique and implementation complexity.

## VI. LDPC CODES

In the past few years, LDPC codes have received much attention because of their excellent performance, and have been widely considered as the most promising candidate ECC scheme for many applications in telecommunications and storage devices, [8]. LDPC codes have been the subject of intense research lately because of their capacity-achieving performance and linear decoding complexity by using an iterative decoding algorithm, the so-called belief propagation or sum-product algorithm [3]. They were originally proposed in 1962 by Robert Gallager [6]. In the late 90's LDPC codes were rediscovered by Mackay and Neal [1] and also by Wiberg [5]. Current hardware speeds make them a very attractive option for wired and wireless systems.

Low-density parity-check code (LDPC code) is an error correcting code, a method of transmitting a noise free message over a noisy transmission channel. LDPC was the first code to allow data transmission rates close to the theoretical maximum, the Shannon Limit. These codes are the class of linear block codes. These codes can perform within 0.0045 dB of Shannon limit. Low density in the LDPC code refers to the less number of 1's. Low-density parity-check codes are a class of linear block code defined by a sparse  $M \times N$  parity-check matrix,  $H$  [5], where  $N > M$  and  $M = N - K$ . Although LDPC codes can be generalized to non-binary symbols, we consider only binary codes. The parity-check matrix has a small number of '1' entries compared to '0' entries, making it sparse. The number of '1's in a parity-check matrix row is called the row-weight,  $k$ , and the number of '1's in a column is the column-weight,  $j$ . A regular LDPC code is one in which both row and column weights are constant, otherwise, the parity check matrix is irregular. Row and column weights are much smaller than the matrix dimensions, with row weights greater than column weights. The rate of the parity check or code matrix is the fraction of information bits in the codeword. It is given by

$$KN = N - M$$

$$N = 1 - M. \text{ The number of '1'}$$

$$NM$$

$$Nj = k.$$

## A. Classes of LDPC Codes

The LDPC codes are classified into two classes according to the number of ones in every row and column.

- 1) Regular LDPC codes
- 2) Irregular LDPC codes

## B. LDPC Encoding

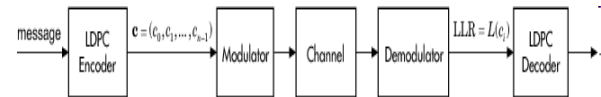


Figure 2. Communication system with LDPC Encoder & Decoder

There are mainly two methods of LDPC Encoding:

- i) General encoding method
- ii) Efficient encoding method

## VII. LDPC DECODING

The algorithm used to decode LDPC codes was discovered independently several times and as a matter of fact comes under different names. The most common ones are the belief propagation algorithm, the message passing algorithm and the sum-product algorithm. In order to explain this algorithm, a very simple variant which works with hard decision, will be introduced first. Later on the algorithm will be extended to work with soft decision which generally leads to better decoding results. Only binary symmetric channels will be considered. LDPC Decoding is of basically two types:

1. Hard Decision Decoding
2. Soft Decision Decoding

### Hard Decision Decoding

The algorithm will be explained on the basis of the example code already introduced in equation and figure. An error free received codeword would be e.g.  $c = [1 \ 0 \ 0 \ 1 \ 0 \ 0]$ . Let's suppose that we have a BHC channel and the received codeword with two error bit  $c_1$  &  $c_5$  flipped to 1.

1. In the first step, all  $v$ -nodes  $c_i$  send a "message" to their  $c$ -nodes  $f_j$  containing the bit they believe to be the correct one for them. At this stage the only information a  $v$ -node  $c_i$  has, is the corresponding received  $i$ -th bit of  $c$ ,  $y_i$ . That means for example, that  $c_0$  sends a message containing 1 to  $f_1$  and  $f_2$ , node  $c_1$  sends messages containing  $y_1$  (1) to  $f_0$  and  $f_1$ , and so on.

2. In the second step every check nodes  $f_j$  calculate a response to every connected variable node. The response message contains the bit that  $f_j$  believes to be the correct one for this  $v$ -node  $c_i$  assuming that the other  $v$ -nodes connected to  $f_j$  are correct. In other words: If you look at the example, every  $c$ -node  $f_j$  is connected to 4  $v$ -nodes. So a  $c$ -node  $f_j$  looks at the message received from two  $v$ -nodes and calculates the bit that the third  $v$ -node should have in order to fulfill the parity check equation. Important is, that this might also be the point at which the decoding algorithm terminates. This will be the case if all check equations are fulfilled. We will later see that the whole

algorithm contains a loop, so another possibility to stop would be a threshold for the amount of loops.

3. Next phase: the v-nodes receive the messages from the check nodes and use this additional information to decide if their originally received bit is OK. A simple way to do this is a majority vote. When coming back to our example that means, that each v-node has two sources of information concerning its bit. The original bit received and two suggestions from the check nodes. Table 2 illustrates this step. Now the v-nodes can send another message with their (hard) decision for the correct value to the check nodes.

4. Go to step 2.

Initial: In hard, variable nodes receive hard 0's and 1's. In soft, they receive  $\Pr(v=0)$  and  $\Pr(v=1)$ .

**Step 1:** Check node finds the predicted correct value for v based on values of other variable nodes connected to it and sends it back. In case of soft, it sends probability.

**Step 2:** Variable node gets suggestions from all check nodes and accepts majority decision. Go to step 1. In case of soft algorithm, it considers probability. Repeat till all parity equations are satisfied.

### VIII. SOFT DECISION DECODING

The above description of hard-decision decoding was mainly for educational purpose to get an overview about the idea. Soft-decision decoding of LDPC codes, which is based on the concept of belief propagation, yields in a better decoding performance and is therefore the preferred method. The underlying idea is exactly the same as in hard decision decoding. Before presenting the algorithm lets introduce some notations:

- $P_i = \Pr(c_i = 1 | y_i)$
- $q_{ij}$  is a message sent by the variable node  $c_i$  to the check node  $f_j$ . Every message contains always the pair  $q_{ij}(0)$  and  $q_{ij}(1)$  which stands for the amount of belief that  $y_i$  is a "0" or a "1".
- $r_{ji}$  is a message sent by the check node  $f_j$  to the variable node  $c_i$ . Again there is a  $r_{ji}(0)$  and  $r_{ji}(1)$  that indicates the (current) amount of believe in that  $y_i$  is a "0" or a "1". The step numbers in the following description correspond to the hard decision case.

1. All variable nodes send their  $q_{ij}$  messages. Since no other information is available at this step,  $q_{ij}(1) = P_i$  and  $q_{ij}(0) = 1 - P_i$ .

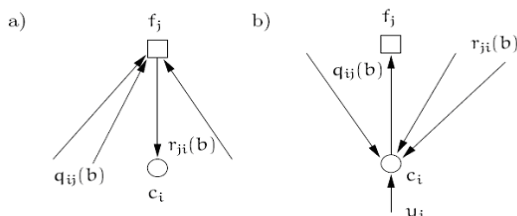


Figure 3 Illustrate the calculation of  $r_{ji}(b)$  and  $q_{ij}(b)$ . The check nodes calculate their response messages  $r_{ji}$ :

$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in C_i \setminus i} (1 - 2q_{i'j}(1))$$

$$r_{ji}(1) = 1 - r_{ji}(0)$$

2. So they calculate the probability that there is an even number of 1's among the variable nodes except  $c_i$  (this is exactly what  $V_{ji}$  means). This probability is equal to the probability  $r_{ji}(0)$  that  $c_i$  is a 0. This step and the information used to calculate the response is illustrated in figure.

3. The variable nodes update their response messages to the check nodes. This is done according to the following equations,

$$q_{ij}(0) = K_{ij} (1 - P_i) \prod_{j' \in C_i \setminus j} r_{j'i}(0)$$

$$q_{ij}(1) = K_{ij} P_i \prod_{j' \in C_i \setminus j} r_{j'i}(1)$$

The probability that the whole sequence contains an even number of 1's is

$$\frac{1}{2} + \frac{1}{2} \prod_{i=1}^M (1 - 2p_i)$$

Where by the constants  $K_{ij}$  are chosen in a way to ensure that  $q_{ij}(0) + q_{ij}(1) = 1$ .  $C_i \setminus j$  now means all check nodes except  $f_j$ . Again figure illustrates the calculation in this step. At this point the v-nodes also update their current estimation  $\hat{c}_i$  of their variable  $c_i$ . This is done by calculating the probabilities for 0 and 1 and voting for the bigger one. The used equations

$$Q_i(0) = K_i (1 - P_i) \prod_{j \in C_i} r_{ji}(0)$$

and

$$Q_i(1) = K_i P_i \prod_{j \in C_i} r_{ji}(1)$$

are quite similar to the ones to compute  $q_{ij}(b)$  but now the information from every c-node is used.

$$\hat{c}_i = \begin{cases} 1 & \text{if } Q_i(1) > Q_i(0), \\ 0 & \text{else} \end{cases}$$

If the current estimated codeword fulfill the parity check equations the algorithm terminates. Otherwise termination is ensured through a maximum number of iterations.

4. Go to step 2.

The explained soft decision decoding algorithm is a very simple variant, suited for BSC channels and could be modified for performance improvements. Beside performance issues there are numerical stability problems due to the many multiplications of probabilities.

The results will come very close to zero for large block lengths. To prevent this, it is possible to change into the log-domain and doing additions instead of multiplications.

The result is a more stable algorithm that even has performance advantages since additions are less cost.

### RESULTS

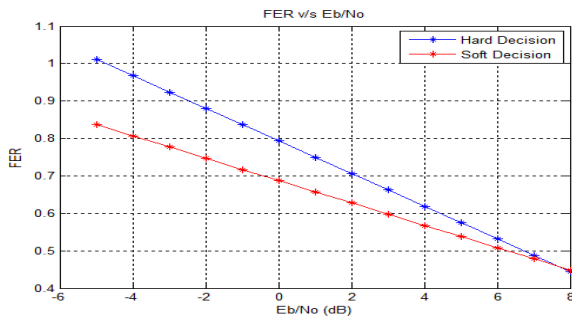


Figure 4 Comparison of hard & soft decision decoding in terms of FER v/s Eb/No.

### CONCLUSION

Here we can observe From the Graph the FER for Hard Decision Decoding is greater as compared to soft Decision decoding technique.

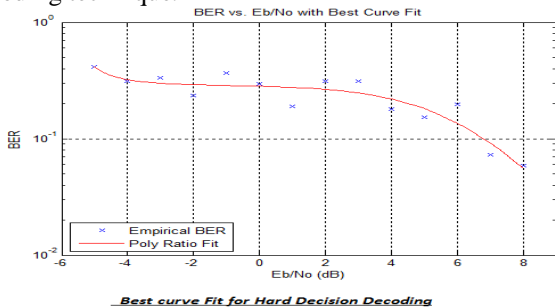


Figure 5 Matlab Result for Hard Decision Decoding

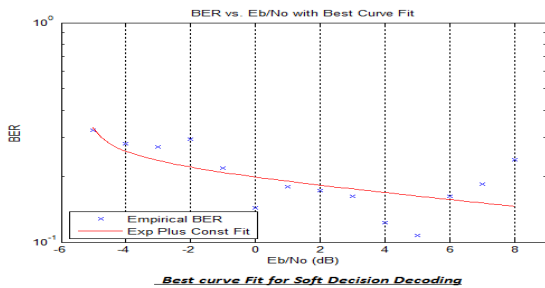


Figure 6 Matlab Result for Soft Decision Decoding

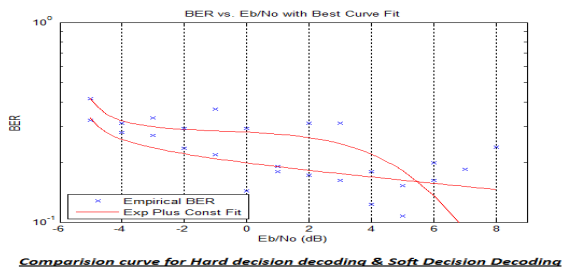


Figure 7 Comparison of Hard Decision Decoding & soft Decision Decoding

### CONCLUSION

Here we can observe From the Graph the BER for Hard Decision Decoding is greater as compared to soft Decision decoding Technique.

### ACKNOWLEDGMENTS

Our thanks to Department of ETC of our College who helped us to perform the practical's on our topic & also to the family members who supported a lot to achieve this.

### REFERENCES

- [1] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," IEE Electron Lett., vol 32, no. 18, pp. 1645–1646, Aug. 1996.
- [2] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. Inform. Theory, vol. IT-45, no. 2, pp. 399–431, March 1999.
- [3] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and sum-product algorithm," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 498-519, Feb. 2001.
- [4] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in Proc. 30th Annu. ACM Symp. Theory of computing, 1998, pp. 249-258.
- [5] N. Wiberg, "codes and decoding on general graphs," Linköping studies in science and technology, No. 440, 1996
- [6] R. G. Gallager, Low-Density Parity-Check Codes, Cambridge, MA:MIT Press, 1963.
- [7] R.H. Morelos-Zaragoza. The Art of Error Correcting Coding. John Wiley & Sons, 2002.
- [8] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching low-density parity-check codes," IEEE Trans. Inform. Theory, vol. 47, pp. 619-637, Feb.2001.