A Novel Encoding/Decoding Scheme with Belief Propagation and LDPC Codes

Prashnatita Pal, Bikash Chandra Sahana and Jayanta Poray

September 28, 2021

A novel Encoding/Decoding scheme with Belief Propagation and LDPC Codes

Prashnatita Pal, [1]
Member IEEE & Corresponding Author
Electronics &Communication Engineering
National Institute Of Technology, Patna
Patna, India City
prashnatitap@gmail.com

Bikash Chandra Sahana [2]
Electronics &Communication Engineering
National Institute Of Technology, Patna
Patna, India
sahana@nitp.ac.in

Jayanta Poray [3]
Computer Science Engineering
Techno India University
Kolkata, India
jayanta.p@technoindiaeducation.com

Abstract— **Decoding plays an important role for the reliability and success of modern data communication because this communication is an inherent part of our daily lives nowadays. However, as with almost all types of communication, data communication is also affected by error. Error in communication mainly occurs across the communication channel and its effect is very severe at times. Depending on the type of error, the codeword transmitted across the channel can be partially or totally changed. So, the information transmitted can be lost permanently due to error. The concept of error correction is to introduce redundant data in the information during the transfer of data. The receiver can check the redundant data in the received message and can determine the consistency of the sent message. It can also check the redundant data to detect errors transmitted in the message and can correct the errors, if any. In this context, the principle of loopy Belief Propagation used by the LDPC codes comes with an excellent error correction performance. The goal of this thesis is to set up a framework to describe the LDPC code constraints, compute probabilities in trellises and finally generate transfer vectors for loopy Belief Propagation model.**

Keywords*— LDPC, Belief Propagation, Encoder/Decoder, Interleaver*

I. INTRODUCTION

Modern information and communication era demands efficient and reliable transmission schemes with a significant accuracy. For being efficient the process needs to be fast with putting less amount of effort and for being reliable it must be capable to produce the correct results consistently. These two expectations are always hard to achieve together at the same time. Before transmission, the message needs to be modelled into a suitable form. This modelling process is known as encoding and it results the encoded message called codeword. Normally this process is efficient and less complex. But there are concern about the related metrics derived and used in this phase. These must be good enough to recover the partly corrupted codeword very well. In transmission phase the sender wants to transmit the codeword (a sequence of binary digits) to a receiver via a channel. Unfortunately, the channels are not entirely reliable. Inside a channel different kinds of noises lead the possibility of disturbing the information. As a result the receiver does not receive the actual encoded message but a computed version. During last few decades a plainly of efforts were made to address the problem of recover the corrupted codeword, received via a communication channel. This recovering process is known as error correction or decoding. There are some good achievements until now. But still these are not enough. We can do even better. The overall picture of a transmission system is as Figure 1.1. Here the sender encodes the message from the message source by using an Encoder. This message is affected by noise inside the Channel. The receiver needs to correct the affected message. Therefore, it uses a Decoder to decode (recover) it and get the recovered message into the destination. The overall picture of a transmission system is as Figure 1.1. Here the sender encodes the message from the message source by using an Encoder. This message is affected by noise inside the Channel. The receiver needs to correct the affected message. Therefore, it uses a Decoder to decode (recover) it and get the recovered message into the destination.

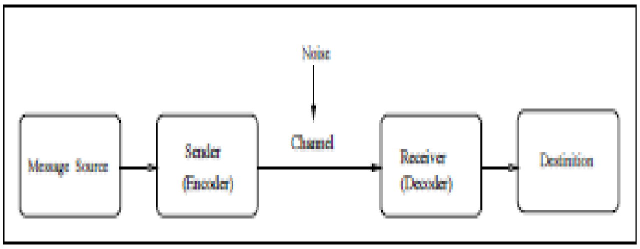

Figure1 Basic design of a digital transmission system

We have pointed that the reliable communication is a matter of concern. So in spite of several noises, keep and maintain the quality information is a big challenge to us. In order to deal with this challenge, the theory of error correcting codes needs to get our precise attention. Therefore we consider a class of error correcting codes which performs iterative decoding inside a graph. Using this graphical model we design an information exchange and update of takes place in an iterative way. Two component decoders get involved in this iteration process and leads to a good approximation. The overall objective is categorized into three phases : (1) Study the LDPC encoder/decoder framework and associated components. This includes the structure of LDPC codes and corresponding iterative decoder using Belief Propagation strategy. (2) Extend and refresh the model for practical application. Here we formally construct the LDPC code with the analysis of a graph, named as factor graph. Then use that code in our iterative decoder model. (3) Design the model with a simulator and test the result. In this context several code variants, algorithmic study according to the context of code construction and decoding are the prime achievements of our effort. The decoder works in an iterative way. This iterative process is motivated by loopy Belief Propagation technique.

Our work contains two The basic design of our model iterative decoder illustrated in Figure 1.2. This decoder is motivated by two types of knowledges (1) the information coming from the channel, termed as channel input and (2) the code knowledge i.e., the knowledge about the LDPC codes which were constructed in some point of time and used for encode that information before send via channel. Two component decoders shown in this diagram are the central part of our attention. They exchange and update their corresponding beliefs i.e., $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$ in an iterative fashion. Note that in information and communication science, the meaning of the word of ”belief” often synonymous with the words knowledge or information. All these words have some meaningful scene according to the context of their applications. Here the belief is the marginal values from component decoders and these marginal values gives a good approximation. Finally it generates the decoded (error corrected) output.

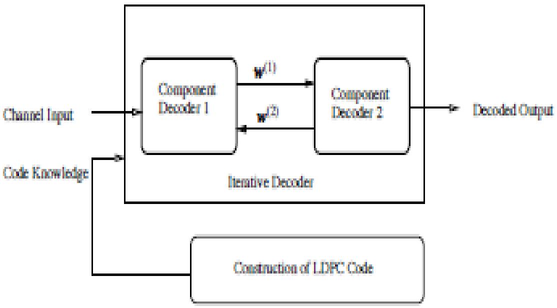

Figure 2 Basic design of the iterative decoder

The organization of this paper is as follows:

After an introduction in the first section, then basic background related to this work has been described. The details of constructing Low Density Parity Check (LDPC) codes, encoding infrastructure and transmission via a Binary Symmetric Channel has been discussed at next. The Decoding details of LDPC code has been presented in after that. There we use the loopy Belief Propagation algorithm, contains message passing and knowledge update mechanisms. Finally, we discuss the scope of future work along with the conclusion.

II. LITERATURE REVIEW

Our work is motivated with the study of Error correcting code. The related background knowledge about coding theory, we needed for this paper are considered from [1] and [2]. Throughout this work we discuss about Low Density Parity Check (LDPC) codes and its several variants. These are described well in the doctoral work by Ingmar L and Amin S and the report by [3] [4] respectively. LDPC codes works 2 well inside a graph and use loopy Belief Propagation to pass the message in an iterative way. We have studied Belief Propagation in a message passing framework using graph from [5], [6] and [7]. To design the iterative decoder, we use an extended version of loopy Belief Propagation algorithm from [8].

2.1 In this section we review various research papers related to work done in error correction using LDPC code.

Morii and other explain their paper [9], irregular LDPC code is decoded using Belief Propagation and Gallagher A algorithm. The error correction capacity of the irregular LDPC codes is also checked. However, the drawback of this paper is that it is difficult to process and display the results of the error correction capability using Belief Propagation theoretically. Next [10] paper mainly deals with LDPC code for holographic memory purposes. Here a decoding framework is used for decoding the LDPC code. The channel over which the code is transmitted contains non-uniform error. The drawback of the work in this paper is that this process can only work for channel with non-uniform error. After that deals with the basic steps involved in using LDPC code. It includes generation of the generator matrix, the LDPC parity-check matrix, LDPC decoder and also some of the concepts involved in generating LDPC code such as Sum-Product algorithm, Factor graph, etc. [11]. The drawback of the work in this paper is that it considers only burst error, but across a channel there are various types of errors, not just burst error. Next paper [12] deals with a decoding method called Multiple Logic Decoder (MLD) is used. MLD is useful for applications where a large number of errors need to be corrected. MLD is used for memory storage. A different type of LDPC code is used called Euclidean Geometry Low Density Parity Check (EG-LDPC) Code. EG-LDPC code has a separate detector to be used at receiver in order to detect the errors. However, the drawback of using MLD is that it normally works on EG-LDPC, which is more complex than normal LDPC code. High performance short-block binary regular LDPC codes is explain in next paper [13]. Here deals with concatenated LDPC codes to be used in channel of short block code capacity. Some channels have a reduced capacity to support short codewords to decrease latency. So, in this paper, two regular LDPC codes are concatenated parallelly by the use of Inter-leaver. This process reduces codeword length but increases security. However, the drawback is that, concatenating LDPC codes is difficult to perform and operate on. Concatenated LDPC codes are also difficult to decode. Good error correcting codes based on very sparse matrices. [14]. It contains a type of LDPC code called Mac-Kay codes. In this case, an (m x n) matrix is created with random weight per column and random weight per row. A parity check matrix can be created by permutation of the (m x n) matrix. The drawback of Mac-Kay codes is that it lacks sufficient structure to make it less complex for encoding and decoding. Design of Low-Density Parity Codes for Deep-Space Applications [15] Here a technique called Progressive Edge Growth (PEG) is discussed. PEG overcomes the drawbacks of Gallagher Code by a different process of designing the parity check matrix instead of random selection. It uses a computer search algorithm. Design of efficiently- encodable moderate -length high rate irregular LDPC codes.” [16] It deals with three codes- RA, IRA, eIRA. Rapid Accumulator (RA) code has characteristics of both serial turbo code and LDPC code. They repeat the user codes, permutes them and sends them through the accumulator. They have performance near capacity limits but have low codeword rates. An advancement over RA is IRA (Irregular Repeat Accumulator) code. They repeat some bits more than others. They higher code rates than RA but are non-systematic codes. To overcome drawbacks of RA and IRA, eIRA (extended IRA) is introduced. They allow both high and low rates and are systematic in nature. Design of LDPC Codes: A Survey and New Results. [17] This paper provides a fundamental design of LDPC codes, where EXIT chart technique is used to determine the optimal degree distributions. This technique is further extended to the special case of photograph based LDPC codes. The algebraic LDPC code design leads to cyclic, quasi-cyclic, and structured codes. Mapping Interleaving Laws to Parallel Turbo and LDPC Decoder Architectures [18] For high data rate applications, implementations of turbo-like codes (i.e., Turbo Codes and LPDC Codes) use parallel architecture for collision free constraints. In this paper the researchers proved wrong about this literary myth, that ad hoc design is only used for parallelism requirement. They stated and proved that it gives a desired collision-free mapping algorithm by giving examples of this algorithm on both Turbo and LDPC codes for better illustration. Low Complexity LDPC Code Decoders for Next Generation Standards [19] This paper represents design of LDPC codes for next generation Wi-Fi, WiMAX, and DVB-S2 standards. Here for feasible high throughput decoder, a low complexity decoder is designed, and also node processing approximations to the top-level decoder architecture is discussed here. LDPC Decoder's Error Performance over AWGN Channel using Min-Sum Algorithm [20] LDPC code meets the Shannon limit at a very close proximity than any other codes, having complexity and latency during execution by using large code lengths and a greater number of iterations during decoding. In this paper they have used 16 Quadrature Amplitude Modulation Technique in LDPC decoder, which provides better error performance over AWGN channel with Min-Sum Algorithm, including effects on Bit Error Rate (BER) over Signal to Noise Ratio (SNR). Block-LDPC: A Practical LDPC Coding System Design Approach [21]. This paper worked with Block LDPC codes, which means that there is a joint LDPC code where code-encoder-decoder is present. The main objective was to construct an LDPC code which would be suitable for some hardware-based constraints and would ensure effective hardware implementations for both encoder and decoder. Here we can also find some computer simulations demonstrating good error correcting performances of Block LDPC codes with effective encoder and decoder design. An Approach Based on Edge Coloring of Tripartite Graph for Designing Parallel LDPC Inter-leaver Architecture [22] This paper focuses on design of partially parallel hardware architecture, which proves efficient in terms of area, cost, flexibility and performances. The design of this architecture is complex due to memory mapping, but it can be solved using an algorithm of tripartite graph modelling and modified edge coloring algorithm for designing parallel LDPC inter-leaver architecture. A Family of Irregular LDPC Codes with Low Encoding Complexity [23] Here irregular quasi-cyclic LDPC are derived from different families, where the resulted codes are encoded with low complexity and can perform well in the Sum-Product Algorithm. Estimation of Bit and Frame Error Rates of Finite Length Low Density Parity Check Codes on Binary Symmetric Channels [24] Here estimation of performance of LDPC codes using hard decision iterative decoding algorithm on Binary Symmetric Channels is stated. The small weight error patterns which a decoder cannot be easily corrected, but by using this method it can be done. It estimates the frame error rate (FER) and bit error rate (BER) of an LDPC code for both regular and irregular, with a good precision rate than the previous one for all crossover possibilities of practical interest. After comparing to Monte Carlo Simulation, it is found that this method has small computational complexity particularly for lower error rates.

2.2 Comparative study of various decoders used to decode LDPC code

Abhishek Bairwa and adale Self-Learning Tool for LDPC Codes[25] explain a software tool for self-learning different aspects of LDPC codes is developed, useful for beginners in channel coding. A prototype of LDPC game was designed as a self-learning tool based on fun loving activities, using step by step procedure to get an idea of LDPC codes for beginners. Next paper An Approach Based on Edge Colouring of Tripartite Graph for Designing Parallel LDPC Inter-leaver Architecture by Awais SANI, and other [22]. A practical and feasible solution for LDPC decoder is to design partially-parallel hardware architecture. But there are some memory mapping problem, that is solved in this paper, an approach based on a tripartite graph modelling and a modified edge collaring algorithm to design parallel LDPC inter-leaver architecture. After that Block-LDPC: A Practical LDPC Coding System Design Approach" - Hao Zhong and Tong Zhang.[21]. This paper introduces to a set of hardware-oriented constraints, subject to which a semi-random approach is used to construct Block-LDPC codes with good error-correcting performance. Correspondingly, an efficient encoding strategy and a pipelined partially parallel Block-LDPC encoder architecture, and a partially parallel Block-LDPC decoder architecture is designed. This paper further deals with the estimation of Block-LDPC coding system implementation key metrics including the throughput and hardware complexity for both encoder and decoder. The good error-correcting performance of Block-LDPC codes has been demonstrated through computer simulations. With the effective encoder/decoder design and good error-correcting performance, Block-LDPC provides a promising vehicle for real-life LDPC coding system implementations.

III. LDPC SYSTEM TO DETECT AND CORRECT SINGLE BIT ERROR

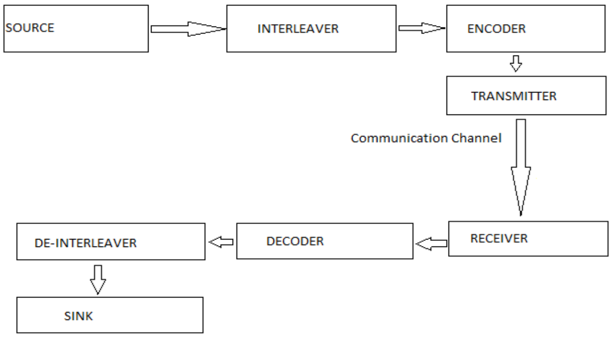

Figure 3: Block diagram of overall procedure

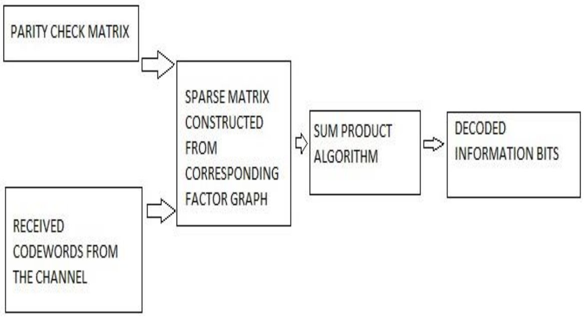
PARITY CHECK MATRIX
RECEIVED CODEWORDS FROM THE CHANNEL
SPARSE MATRIX CONSTRUCTED FROM CORRESPONDING FACTOR GRAPH
SUM PRODUCT ALGORITHM
DECODED INFORMATION BITS

Figure 4: Block Diagram of decoder to be constructed

3.1 Mathematical Analysis

The decoding of LDPC code is done using factor graphs. Factor graph is a part of bipartite graph.

Bipartite graph is a graph which fulfils two main conditions. They are as follows:

i)Every vertex must be grouped into two groups which are called parts. Hence the name bipartite meaning two parts.

ii)Each edge must connect two vertices from each part. No edge can connect any two vertices which belong to the same part. There are two main types of bipartite graph- Tanner

Graph and Factor Graph. Factor graph is mainly used in creation of the decoding matrix in LDPC and Turbo codes. Factor graphs represent a multivariable polynomial function as factors of its subsets. The diagram of factor graph is shown figure 5.

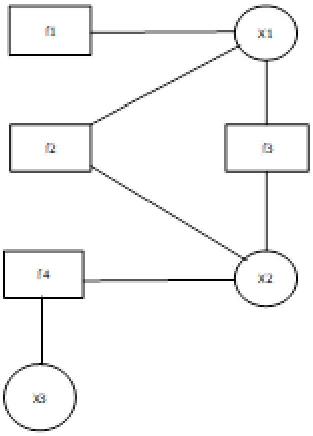

Figure 5: Example of Factor graph

This is represented by:

g (x1, x2, x3) = f1(x1) f2(x1, x2) f3(x1, x2) f4(x2, x3)

3.1.1 Encoder Structure

The generator matrix G, is defined as:

$$c = G^T m$$

where,

$c = [c_1, c_2, \ldots\ldots\ldots, c_N]^T$ – Codeword

$m = [m_1, m_2, \ldots\ldots\ldots, m_k]^T$ – Message Word

G = k by n Generator Matrix

In order to determine the relationship of the parity bits to the H matrix, we will use the following definition of syndrome. We have defined a complete set of successful parity–checks as:

$H_C = 0$

$H_{N\text{-}k \text{ X } N} = N$ – k by N Parity Check Matrix

Codeword formed is such that:

$c = [p: m]^T$

where,

$p = [p_1, p_2, \ldots\ldots, p_{N\text{-}k}]^T$ – Parity Bits

Therefore:

$H\,[p: m]^T = 0$

H can be partitioned as:

H = [X : Y]

where:

X = N – k by N – k Sub-matrix

Y = N – k by k Sub-matrix

Therefore

$X_p + Y_m = 0$

Using modulo – 2 arithmetic we can solve for as:

$p = X^{-1} Ym$

then we solve for c as :

$c = [(X^{-1}Y)^T : I\,]^T m$

where I is the k-by-k identity matrix

and hence we defined G as:

$G = [(X^{-1}Y)^T\!: I]$

3.1.2 Decoder Structure

Let a binary codeword x_i $(x_1, x_2, \ldots, x_n)$ be transmitted over an Additive White Gaussian Noise (AWGN) channel and the received codeword be y_i $(y_1, y_2, \ldots, y_n)$. We assume that after modulation $x_i = 0$ is transmitted as -1 and $x_i = 1$ is transmitted as +1. The n code bits must satisfy all parity checks and we will this fact to compute the posterior probability.

$$p\big(x_i = b|s_{i|y}\big), b\ \in \{0,1\}$$

where, $s_{i|y}$ is the event that all parity checks associated with x_i are satisfied.

v_j = set of variable nodes connected to c_j

$v_{j|i}$ = set of variable nodes connected to c_j except the variable node x_i

c_i = set of check nodes connected to variable node x_i

$c_{i|j}$ = set of check nodes not connected to variable node x_i excluding v_j

$M_v(i)$ = message formed from all variable nodes except x_i

$M_c(j)$ = message formed from all check nodes except c_j

In the first half of the iteration, each variable node processes its input message received (y_i) from the channel and passes it to the neighboring connected check node. At the first pass, there is no message incoming back from the check node. Now, having the input bit, the check equation is evaluated whether it is satisfied or not. Then, the resulting outputs are passed to the corresponding neighboring variable nodes which are connected to the check node c_j excluding the information from $x_{i.}$ P_r(check equation is c_i is satisfied | input message)

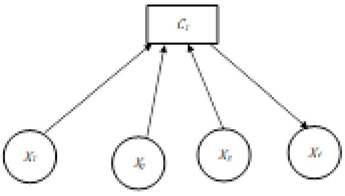

Figure 6: Process of decoding

In the other half of the iteration, each variable node processes its input message and passes it to its neighboring check node using sample channel and incoming messages from all other check nodes connected to the variable node x_i , excluding check node c_j.

The information passes concerns,

$$p_r(x_1 = b \mid input\ messages), b \in \{0,1\}$$

The information passed to check node c_3 is all the information that is available to variable node X_1 from the sample channel y_1 through its neighboring check nodes excluding c_3.

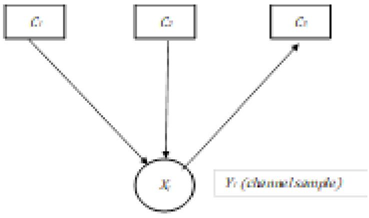

Figure 7: Process of finding end value of check node

Now, the check node has got new messages from the variable node and it has to check whether the check criteria is satisfied and passes all the information the variable node. Then, at every bit in the variable node, x_i is updated by using the channel information and messages from the neighboring check nodes, without excluding any check node c_j information. After all the iterations, it will check whether the valid codeword is found or not. If codeword is found then, $H \otimes \hat{x}$ = 0. Thus, the iteration terminates, else it continues.

3.2 Algorithm for single bit error detection and correction

The algorithm is as follows:

3.2.1: Design of the Inter-leaver

Prerequisites: The length of the Interleaver needs to be known. Let length be n= M.d_c = N.$d_{v.}$ An irregular array Π_p of length n.
M= row dimension of the Interleaver.
d_c= number of check nodes.
d_v= number of variable nodes.
Steps:

1. Π_p= $Z^n{}_2$ = 1, 2, 3, 4, …, n
2. While the Interleaver Π_p is not random and unacceptable, go to step 3.
3. Shuffle the positions of the array Π_p randomly.
4. End while loop
5. Return the randomly arranged array. This array is the finished Interleaver.

3.2.2: Check if the Inter-leaver is acceptable or not

Prerequisites: The row dimension (M) of the interleaver array is needed along with the number of check nodes (d_c) and variable nodes (d_v).

Steps:

1. for i starting from 1 to M, go to step 2.
2. for j starting from 1 to d_c and k starting from 1 to d_c, go to step 3.
3. Check if j ≠ k and if $\Pi_p[i*d_c + j]/d_v = \Pi_p[i*d_c + k]/d_v$, is true then go to step 4, else go to step 5.
4. Return interleaver not accepted.
5. Return interleaver accepted.
6. End of for loop.
7. End of for loop.

3.2.3: Generation of the Parity Check Matrix H using the Inter-leaver

Prerequisites: The Interleaver $\Pi_p[M \text{ x } d_c]$ and the LDPC code parameters (M, d_c, d_v).
Steps:

1. for i starting from 1 to M, go to step 2.
2. Take a temporary local storage L[d_c].
3. for j starting from 1 to d_c, go to step 4.
4. Calculate $L[j] = \Pi_p[(i-1) * M + j]/d_v$
5. End of for loop.
6. Arrange L[] in increasing order of the probabilities of the elements.
7. Take counter variable k = 1
8. for j starting from 1 to N, go to step 9.
9. Check if L[k=k+1] = j, is true then go to step 10, else go to step 11.
10. H[i][j] ← 1
11. H[i][j] ← 0
12. End of for loop
13. End of for loop
14. Return Parity Check Matrix H.

3.2.4: Use of Binary Gaussian Elimination

Prerequisite: The Parity Check Matrix is needed.
Steps:

1. for each row i starting from 1 to M, go to step 2.
2. Let, s = i.
3. While array a[i][s] = 0, go to step 4.
4. Do s=s+1
5. Exchange a_i column with a_s column
6. for each column j starting from 1 to N, go to step 7
7. check if j ≠ i and if a[i][j] = 1, is true then go to step 8
8. for k starting from i to M, go to step 9
9. do a[j][k] = a[j][k] + a[i][k]
10. end of for loop
11. end of for loop
12. end of while loop
13. return H matrix in systematic form (H = [I|P])

3.2.5: Simulation of Binary Symmetric Channel

Prerequisites: Input codeword c is needed along with error probability of the channel p_c.
Steps:

1. for I starting from 1 to codeword length (|c|), go to step 2.
2. Generate a random number Rand(i) such that 0<Rand(i)<1.
3. Check if Rand(i) ≤ p_c, is true then go to step 4, else go to step 5.
4. $r_i = (-1)c_i$
5. $r_i = c_i$
6. end of for loop
7. return Received codeword after transmission $r_{i.}$

3.2.6: Usage of Belief Propagation Technique for iterative decoding of the LDPC codeword

Prerequisites: Initial dynamic knowledges of the Decoder = $w^{(2)} = 0$ and initial static knowledge V is required.

Steps:

1. for i starting from 1 to Number of loops n, go to step 2.
2. Calculate new belief for Decoder 1 = The Dynamic Knowledge $w^{(1)}$
3. Calculate new belief for Decoder 2= The Dynamic Knowledge $w^{(2)}$
4. Compute the symbol-by-symbol sign $c^{\wedge}_j = \text{sign}(V_j + w_i^{(1)} + w_i^{(2)})$ for all j
5. End of for loop
6. Return the optimal decisive sign vector $c^{\wedge}$

3.2.7: Computation of new belief for Decoder l

Prerequisites: Static knowledge $V^{(l)}$ is required along with extrinsic dynamic knowledge $w^{(h)}$ from the conjugate decoder h.

Steps:

1. Compute Overall Knowledge $m^{(l)} = V^{(l)} + w^{(h)}$ for all i
2. Calculate the logarithmic probability ratio
   $L_i^{(l)}(m^{(l)}) = ½ \log_2 [P^{(l)}(+1|m^{(l)}) / P^{(l)}(-1|m^{(l)})]$ for all i
3. Calculate the Extrinsic Logarithmic Probability Ratio
   $\overline{L}_i^{(l)} = L_i^{(l)} – V^{(l)} – w^{(h)}$ for all i
4. Calculate the symbol-by-symbol new belief $w_i^{(l)} = \overline{L}_i^{(l)}$ for all i
5. Return the new belief vector $w^{(l)}$

1. RESULT ANALYSIS

In this section we have included the input and subsequent results obtained by us.

INPUT VALUES:
The Input message array x = {1, 1, 0}
The channel error probability $p_c = 0.2$

CODE CONSTRUCTION:

1. The Inter-leaver π = {0, 3, 6, 1, 4, 8, 2, 5, 10, 7, 9, 11}
2. The De-Inter-leaver π^{-1} = {0, 3, 6, 1, 4, 7, 2, 9, 5, 10, 8, 11}
3. The Parity Check Matrix

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

ENCODING:

1. The Generator Matrix

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

2. The codeword with BPSK map

c = {0, 1, 1, 1, 1, 0} → {+1, -1, -1, -1, -1, +1}

CHANNEL:

The transmitted word t = {+1, -1, -1, -1, -1, +1}

The received word r = {+1, +1, -1, -1, -1, +1}

INTERMEDIATE RESULTS OF THE ITERATIVE DECODER:
The received codeword r = 1 1 -1 -1 -1 1

2. CONCLUSION

The purpose of the present work is to study the scope of Low-Density Parity Check (LDPC) codes for design an encoder/decoder framework using loopy Belief Propagation method. The LDPC codes have represented by the cut set of two constituent codes, namely repetition code and single parity check code. The relation between these two constituent codes have represented by an undirected graph, called factor graph. In the first part of our work, we have designed the factor graph via inter-leaver/de-inter leaver pairs. Our encoding and decoding processes have governed by these. In this sequel, we have observed that the parity check matrix has constructed from the inter-leaver in order to produce the codeword (encoded message) and in the other hand the iterative decoding process also operated by the inter-leaver/de-inter leaver pairs. During that experiment we have noticed that, the LDPC code have defined by all these three variants i.e., factor graph, inter-leaver and parity check matrix. In the second part of our work, we have designed a model iterative decoder. The goal of our model decoder is to take the optimal decision from the available information. In that context there were three available information to our model decoder (1) the received word (2) channel error probability (3) the knowledge about the LDPC code. The decoding process by using all these information is complicated. Therefore, in our work we have considered two component decoders. The component decoders were work inside a loopy Belief Propagation framework. There the component decoders have initiated by the information received from the channel, namely static knowledge and have computed their subsequent dynamic knowledge in an iterative way. Finally, the decision was taken when the knowledge was not change further. This were produced the symbol-by-symbol decoded transfer vector. Our claim was that the decision was close to the optimal. There is no available theoretical proof for that. The claim we have made were supported by the result, obtained from our simulator. LDPC code is an extremely efficient error correction code which has a low error floor. It is one of the codes to come very close to Shannon's Theoretical Channel Limit. LDPC code is used in 5G communication and in the DVB-S2 system for data transfer and video streaming. However, LDPC code is still growing in use and it will continue to grow and become one of the most important and one of the most commonly used codes in the future.

REFERENCES

[1] Richard E. Blahut. Algebric Codes for Data Transmission. Cambridge University Press, Cambridge, 2003.

[2] Ulrich Sorger. Lecture Notes on Error Correcting Codes. University of Luxembourg.

[3] Ingmar Land. Reliability Information in Channel Decoding, 2005. Christian Albrechts University of Kiel.

[4] Amin Shokrollahi. LDPC Codes : An Introduction, 2003. Digital Fountain Inc.

[5] Daniel P. Huttenlocher Pedro F. Felzenszwalb. Efficient Belief Propagation for Early Vision.International Journal of Computer Vision, 70(1):41 – 54, October 2006.

[6] Ulrich Sorger. Discriminated Belief P ropagation (Technical Report TR-CSE-07-01). University of Luxembourg.

[7] Ulrich Sorger. Lecture Notes on Error Correcting Codes. University of Luxembourg.

[8] Richardson Thomas J and Rdiger L. Urbanke. The Capacity of Low-Density Parity-Check

[9] Morii, Tamiya, Hirotomo "Error correction capability of irregular LDPC codes under the Gallagher A algorithm", IEEE, February, 2017.

[10] H Pishro Nik, N Rahnavard, F Fekri "Nonuniform error correction using low-density parity-check codes", IEEE Volume: 51, Issue: 7, July 2005.

[11] Mrugesh R. Patel, Prof. Neeta Chapatwala, "Low Density Parity Check Code for Burst Error Correction", ResearchGate, January, 2013.

[12] Anvesh Thatikonda, C Chitra "Efficient Design to Error Detection in EG-LDPC Codes", IJRTE, Vol: 8, Issue: 5, January, 2020.

[13] Latifa Mostari, Abdelmalik Taleb-Ahmed "High performance short-block binary regular LDPC codes", Alexandria Engineering Journal, November, 2018.

[14] William E. Ryan "An Introduction to LDPC Codes", Department of Electrical and Computer Engineering, University of Arizona,Box 210204,Tucsan,AZ 85721,August 19,2003.

[15] D. MacKay "Good error correcting codes based on very sparse matrices", IEEE Trans Information Theory, pp. 399-431, March 1999.

[16] M.Yang,W.E. Ryan and Y.Li "Design of Low-Density Parity Codes for Deep-Space Applications", IEEE Trans. Comm. ,2003.

[17] Liva, Gianluigi & Song, Shumei & Lan, Lan & Zhang, Yifei & Lin, Shu & Ryan, William. (2006). Design of LDPC codes: A survey and new results. Journal of Communication Software and Systems. 2. 191-211. 10.24138/jcomss.v2i3.283.

[18] A. Tarable, S. Benedetto and G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoder architectures," in IEEE Transactions on Information Theory, vol. 50, no. 9, pp. 2002-2009, Sept. 2004, doi: 10.1109/TIT.2004.833353.

[19] T. Brack et al., "Low Complexity LDPC Code Decoders for Next Generation Standards," 2007 Design, Automation & Test in Europe Conference & Exhibition, 2007, pp. 1-6, doi: 10.1109/DATE.2007.364613.

[20] Yadav, A., Kakde, S., Khobragade, A., & Bhoyar, D. (2018). LDPC Decoder's Error Performance over AWGN Channel using Min-Sum Algorithm.

[21] Hao Zhong and Tong Zhang, "Block-LDPC: a practical LDPC coding system design approach," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, no. 4, pp. 766-775, April 2005, doi: 10.1109/TCSI.2005.844113.

[22] A. Sani, P. Coussy, C. Chavet and E. Martin, "An approach based on edge coloring of tripartite graph for designing parallel LDPC interleaver architecture," 2011 IEEE International Symposium of Circuits and Systems (ISCAS), 2011, pp. 1720-1723, doi: 10.1109/ISCAS.2011.5937914.

[23] S. J. Johnson and S. R. Weller, "A family of irregular LDPC codes with low encoding complexity," in IEEE Communications Letters, vol. 7, no. 2, pp. 79-81, Feb. 2003, doi: 10.1109/LCOMM.2002.808375.

[24] H. Xiao and A. H. Banihashemi, "Estimation of Bit and Frame Error Rates of Finite-Length Low-Density Parity-Check Codes on Binary Symmetric Channels," in IEEE Transactions on Communications, vol. 55, no. 12, pp. 2234-2239, Dec. 2007, doi: 10.1109/TCOMM.2007.910589.

[25] Abhishek Bairwa, Quynh Le, Apirujee Rungruang, Pruk Sasithong, Lunchakorn Wuttisitttikulkij, Suvit Nakpeerayuth, Chairat Phongphanphanee, Pisit Vanichchanunt, Piya Kovintavewat, Ambar Bajpai, "Self Learning Tool for LDPC Codes".

[26] Pal P, Chandra Sahana B, Poray J. RSA encrypted FSK RF transmission powered by an innovative microwave technique for invulnerable security. The Journal of Defense Modeling and Simulation. August 2021. doi:10.1177/15485129211031670