Master Thesis
Electrical Engineering

Root LDPC Codes for Non Ergodic Transmission Channels

Tarique Inayat Bhutto
19830330-9432

School of Engineering
Blekinge Institute of Technology
371 79 Karlskrona
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

Supervisor and Examiner
Prof. Abbas Mohammed

External Supervisor

Asst prof. Iryna Andriyanova

School of Engineering
Blekinge Institute of Technology
371 79 Karlskrona
Sweden

ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to supervisor Prof. Abbas Mohammed and for his guidance and support throughout the whole thesis period. I also love to thank my friends whose moral support really worked as a catalyst during my thesis. I would like to convey my gratitude to my friends and families. Without their love and encouragement, it was really difficult for me to complete my thesis as well as degree in the specific time.

Je tiens à exprimer ma profonde et sincère reconnaissance à mon superviseur externe professeur adjoint Andriyanova Iryna et pour ses conseils et de soutien. Je suis également très reconnaissant à l'ENSEA institue de l'électronique et électrique France cergy génie qui m'a donné le stage et la forme de la plaque pour ce travail de thèse et mes remerciements au directeur de laboratoire ETIS et les chargés de recherche il m'a également motivé dans mon travail de thèse.

Last but not the least; I am very thankful and grateful to Blekinge Institute of Technology (BTH) for providing me the quality education which will definitely help me in upcoming days in my career.

This thesis work is dedicated to my loving parents.

ABSTRACT

Tremendous amount of research has been conducted in modern coding theory in the past few years and much of the work has been done in developing new coding techniques. Low density parity check (LDPC) codes are class of linear block error correcting codes which provide capacity performance on a large collection of data transmission and storage channels while Root LDPC codes in this thesis work are admitting implementable decoders with manageable complexity. Furthermore, work has been conducted to develop graphical methods to represent LDPC codes. This thesis implement one of the LDPC kind “Root LDPC code” using iterative method and calculate its threshold level for binary and non-binary Root LDPC code. This threshold value can serve as a starting point for further study on this topic. We use C++ as tool to simulate the code structure and parameters. The results show that non-binary Root LDPC code provides higher threshold value as compare to binary Root LDPC code.

Keywords: Non-Binary Root LDPC codes, LDPC codes, Binary-Root LDPC codes

Table of Contents

LIST OF FIGURES

1 INTRUDUCTION

1.1 Introduction

Low density parity check (LDPC) codes are class of linear block error correcting codes which provide capacity performance on a large collection of data transmission and storage channels while simultaneously admitting implementable decoders with manageable complexity [1]. They were invented by Gallager in his doctoral dissertation and were scarcely considered in the 35 years that followed.

One notable exception is Tanner, who wrote an important paper in 1981 [2] which generalized LDPC codes and introduced a graphical representation of these codes, now called Tanner graphs. Apparently independent of Gallager's work, LDPC codes were reinvented in mid-1990 by Mackay, Luby; others [3, 4, 5] also noticed the advantage of linear block codes which posses sparse (low density) parity check matrices.

Davey and Mackay in 1998 [6] investigated non-binary LDPC codes; these codes perform better than binary LDPC codes with higher order Galois fields. Recently Hu et al [7] constructed non-binary LDPC codes. The performance of these codes increases as the size of the Galois field increases.

In 2007, a family of LDPC codes has been developed that competes with multiplexed parallel turbo codes suitable for non-ergodic channels [8]. These codes are called Root LDPC codes (RLDPC)

The main suggestion of this thesis work is to combine the two latest LDPC coding techniques (i.e. non-binary LDPC codes over Galois field and the Root LDPC codes) in order to design a code for non-ergodic transmission channels. The constructed code will be called as non-binary Root LDPC code.

1.2 Study Area

The study area of this thesis work is modern coding theory, so the key principle of modern coding theory for capacity approaching codes (i.e. LDPC and Turbo codes,) is the performance that approaches Shannon limit. Shannon limit is actually channel capacity C. For any data rate $R < C$ there exist a code coding technique which allows the probability of error at the receiver to be made arbitrarily small and for $R > C$ probability of error at the receiver increases without bound as rate is increased. So no useful information can be transmitted beyond the channel capacity [9]. There are some elements of capacity approaching codes that must be taken into account.

- Defining the linear complexity graph of the code with pseudo-random interconnection.
- Using soft channel outputs (i.e., with reliability info), not hard, i.e., decode with Euclidean space not Hamming space.
- Decode with iterative, message-passing sum-product algorithm.

1.3 Motivation

Root LDPC codes achieve full diversity over non-ergodic transmission channels in the sense that their error probability declines at moderate or high signal to noise ratio. The LDPC code optimization for non-ergodic transmission channels do not follow the same criteria as those applied for standard ergodic erasure and Gaussian channels, the pervious known analysis is based on asymptotic bit threshold for information variables under iterative decoding. In this work we will investigate asymptotic block threshold for non binary Root LDPC codes for Rayleigh fading channel as a non-ergodic transmission channel. Non ergodic transmission takes place when a block (encoded) data is sent by sub blocks into several slow fading transmission channels. This model can describe the parallel (MIMO) system. [10]. A theoretical approach has been taken in [10] to obtain a code that can achieve full transmit diversity but has not been implemented; therefore we will try to implement the code in laboratory scenario. We hav**e** considered non-binary Root LDPC code.

To achieve full diversity here we mean to ensure that each information node receiving multiple messages undergoing distinct fading coefficients. This idea has been implemented in root LDPC codes [8] by means of root check nodes. Root check node guarantees a message that observes on the second channel state when variable node is observed on the first channel state. Root LDPC codes are full-diversity LDPC codes that can be devised for any diversity order but we will limit our study to rate ½ diversity 2 codes.

1.4 Study Type

The primary focus of this study is to generate non-binary root LDPC codes that can be implemented in the laboratory scenario. Different simulations will be conducted in order to obtain a threshold for non-ergodic transmission channels. Our aim is to compare these obtained results to highlight the good ones that can be suggested to field implementations. The study consists of literature review and exploratory study for collecting valuable information from research literature. We will conduct several simulations in order to collect the detailed data and plot it.

1.5 Report Outline

Chapter 2 provides an overview on the background of the thesis work. It deeply discusses the key challenges in the LDPC coding techniques and their origins. Further, the definitions will be presented for the key terms used in this report such as full diversity, non ergodicity in transmission channels. This chapter also summarizes the related work in the study area and identifies the improvements and opportunities in the current research.

In chapter 3 details on planning and conducting the research is presented. It motivates the selected design and research methodology and gives detail on the research design and conduction including data collection and analysis. It also reflects how the study was planned and conducted. Furthermore, the research and results are discussed in this chapter.

Chapter 4 concludes this thesis work by evaluating the research and its outcome and bringing a discussion in the study area of LDPC codes. This chapter will also include the set of recommendations proposed. It also draws the path for new research efforts.

2 RELATED STUDY

2.1 Background:

This section summarizes the background on the modern coding theory, LDPC codes, LDPC codes over $GF(q)$ (Galois finite field) and Root LDPC codes and their representation.

2.1.1 Over View of digital communication System and coding theory:

Any real world communication system subjected to noise which can cause errors in the transmission of information. Claude Shannon [11] laid the foundation for the methods of designing the communication systems such that the errors in transmission system that occur can be reduce to arbitrary small probability. These methods are collectively known as coding theory.

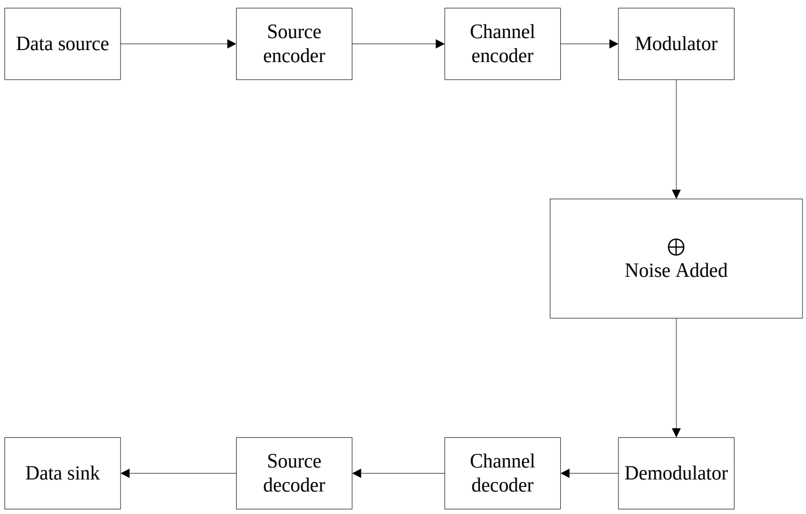

Fig 2.1 Digital Transmission System

Fig 2.1 is a digital transmission system in which channel encoder and channel decoder (fig 2.2) are the primary fields of this thesis work where we use coding techniques to encode and then decode our transmitted signal.

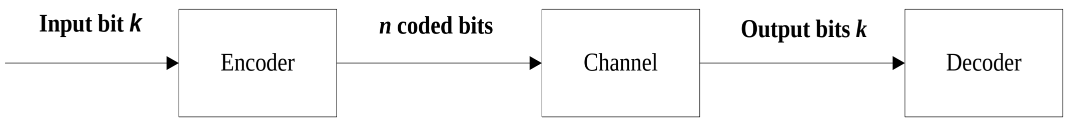

Fig 2.2 Channel Encoder and Decoder Diagram

Shannon's coding theorem became the base for the search of developing new data coding schemes therefore among all LDPC codes are the one.

Shannon's coding theory states that every (statically well defined) data communication channel has capacity C and for any data rate $R < C$ there exist a code $\mathcal{C}$ of $\mathcal{R}$ rate and decoding scheme such that the probability of decoding error $P_r\ (E)$ is arbitrarily small and for any data rate $R > C$ there exists no code $\mathcal{C}$ of rate $\mathcal{R}$ and decoding scheme such that $P_r\ (E)$ is small.

2.2 Introduction to Low Density Parity Check (LDPC) codes:

LDPC codes were invented by Robert Gallager in 1962 [1]. LDPC codes are class of linear block error correcting codes; these codes are well known for their capacity performance on the large collection of data transmission and storage channels with implementable decoders of less complexity. LDPC codes are obtained from sparse bipartite graph and their graphical representation can further be explained by matrix representation which will be next subtopics of LDPC codes.

2.2.1 Graphical representation of LDPCs:

The idea for the graphical representation of LDPC codes was given by Michael Tanner [2]. Tanner generalized LDPC codes and showed that how effectively LDPC codes were used by their Tanner (bipartite) graph.

2.2.2 Bipartite Graph:

Bipartite graph is the graph which has two types of nodes called "check node" and "variable node" where both are connected by undirected edges and the nodes of same type are not connected. For more understanding an example of LDPC Tanner graph is given in Fig 2.3.

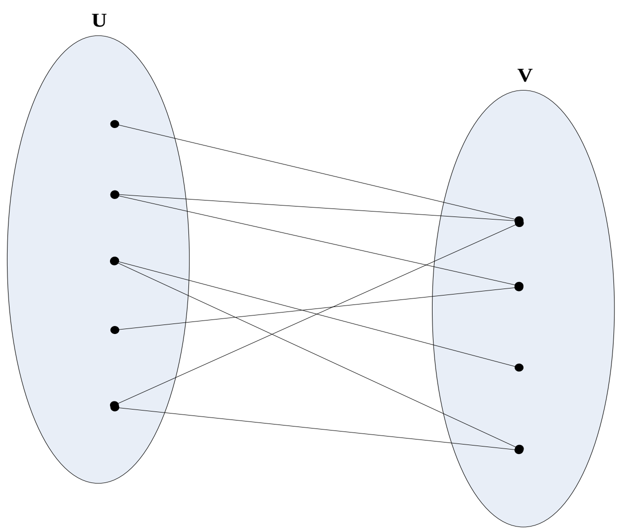

Fig 2.3 Example of Bipartite Graph

One can define bipartite graph as a graph whose vertices are divided into two disjoint sets U and Vsuch that each edge has two vertexes in U and in V each, U and V are independent sets. Furthermore it will explained by a figure 2.3 above.

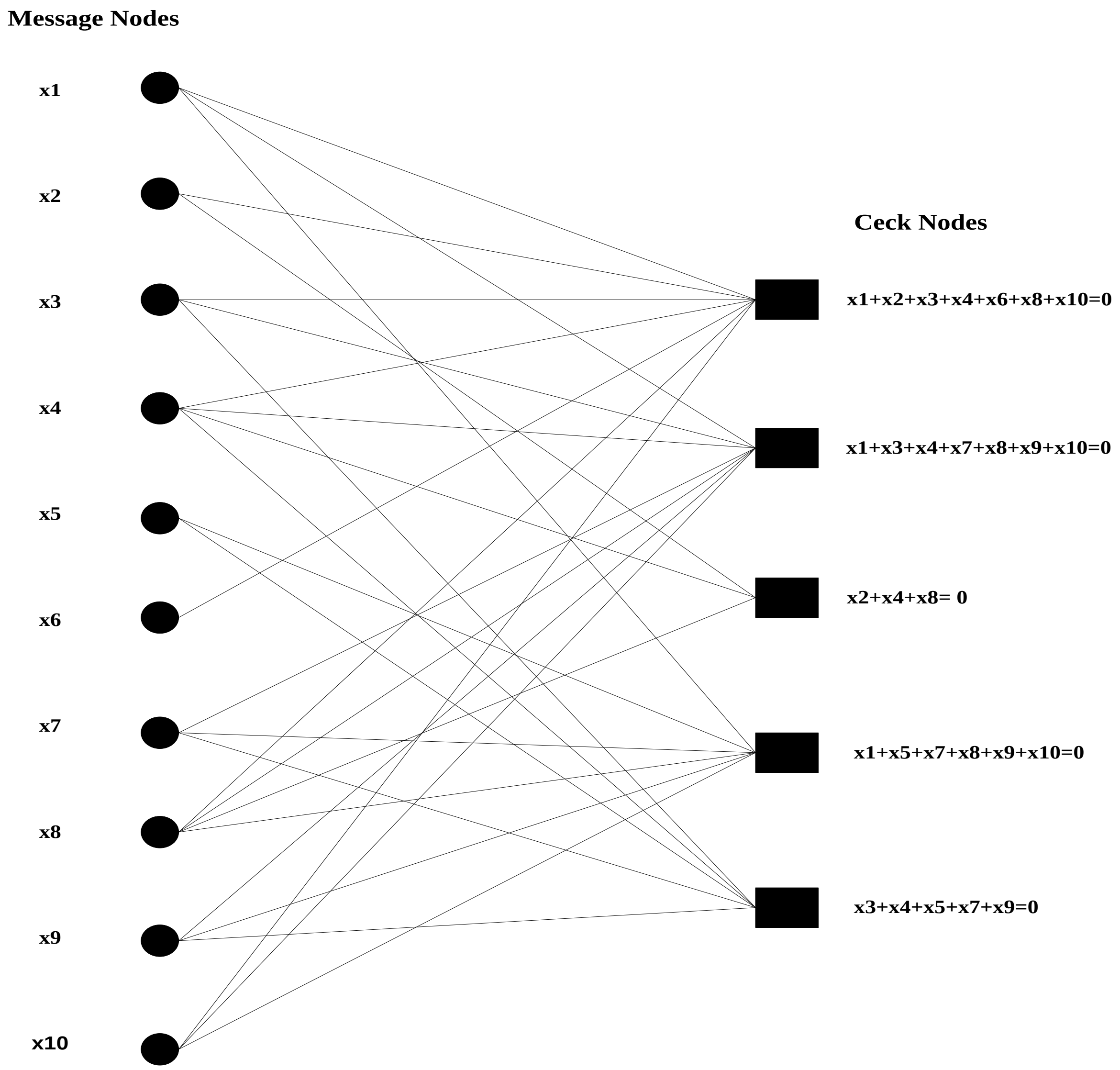

Fig 2.4 Tanner Graph Model

2.2.3 Tanner graph Representation of LDPC codes:

Low Density Parity Check (LDPC) codes are block linear type codes which are obtained from bipartite graphs. Let's say that G is the graph which has n number of nodes on the left side (called message or variable nodes or information nodes) and r number of right nodes (called check nodes). A linear code of block length n can be deduced from the graph which will have dimension at least $n - r$ in the following way: The n coordinates of the codewords are generated from n message nodes. The codewords are those vectors $(c_1, \ldots., c_n)$ such that for all check nodes the sum of neighboring positions among the message nodes is zero. Figure 2.4 gives an example.

The matrix and graph representation are analogous to each other while looking at the adjacency matrix of the graph. Let H be $n \times r$ binary matrix which has entry (i, j) is 1 only in case if the i^{th} check node is connected to j^{th} message node in the graph. Then the LDPC code defined by the graph will be set of vectors $c = (c_1, \ldots., c_n)$ such that $H.C^T = 0$ where H is the parity check matrix for the code. Conversely any binary $n \times r$ matrix gives rise to bipartite graph between n messages and r check

nodes where the code is defined as the null space of H is precisely the code associated to this graph. Therefore any linear code has representation as a code associated to bipartite graph (note that this graph is not uniquely defined by the code). However, not every binary linear code has representation by a sparse bipartite graph. If it does then the code is called a Low-Density Parity-Check (LDPC) code.
Variable nodes can be specified as v-nodes and check nodes are specified as c-nodes and their respective equation are mentioned below.

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1} \tag{2.1}$$

λ_d denotes the fraction all edges which are connected to degree-d variable nodes (v-nodes) and d_v indicates the maximum number of variable nodes similarly in the polynomial

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1} \tag{2.2}$$

ρ_d indicates the fraction of all edges which are connected to degree-d check nodes (c-nodes) and d_c denotes the maximum check node degree.

The messages are sent to and from nodes in order to perform required operation at variable node and for those decoding algorithms are used to decode the messages. There are different decoding algorithms but we will discuss here only Message Passing Algorithm (MAP) as it is used in our thesis work.

2.3 Iterative Decoding Algorithms:

Gallager [1] in addition to introducing LDPC codes also provided a decoding algorithm which is typically near optimal, but during that time many other researchers have also independently discovered near optimal algorithm and other related algorithms, albeit sometimes for different application [4][12]. The computation process of the algorithm is iterative and it computes the distributions of variable in the graph based model as show in figure 2.4. The algorithm comes into different names depending on the context. The message passing algorithm (MPA), the belief propagation algorithm (PBA), and the sum product algorithm (SPA). The term "message passing" refers to all the iterative algorithms mentioned above.

Much as optimal (maximum a posteriori, MAP) decoding of trellis codes symbol by symbol we here are keen to compute the a posteriori probability (APP) such that in a given bit the codeword $\boldsymbol{c} = [c_0 c_1 \ldots . c_{n-1}]$ which is transmitted is equals to 1, given that the received word $\boldsymbol{y} = [y_0\, y_1 \ldots . y_{n-1}]$. Therefore, now we will focus on the decoding of bit c_i with loss of generality, so that we are interested in computing the APP:

$$\mathrm{P}_r(c_i = 1 |\, \boldsymbol{y}\,) \tag{2.3}$$

or the $\widehat{App}$ ratio (called as likelihood ratio, LR)

$$l(c_i) \triangleq \frac{\mathrm{P}_r(c_i=0|\boldsymbol{y})}{\mathrm{P}_r(c_i=1|\boldsymbol{y})} \tag{2.4}$$

For further more stable numerical computation we will take the log-APP ratio which is also called log-likelihood ratio (LLR)

$$l(c_i) \triangleq \log \frac{P_r(c_i=0|\boldsymbol{y})}{P_r(c_i=1|\boldsymbol{y})} \tag{2.5}$$

Here in the subsequence the natural logarithm is assumed i.e. LLR.

2.4 How Message Passing Algorithm Works:

For the computation of $P_r(c_i = 1\,|\boldsymbol{y})$,$l(c_i)$ or $L(c_i)$ the MPA is an iterative algorithm which is based on the Tanner graph of the code in which v-node represent processors of one type and c-node represent processors of another type and the edges represent the message paths. The word iteration is defined as one half iteration, where each v-node processes its input messages and passes the resulting output messages to neighboring check nodes (c-nodes), where as neighboring nodes are those which are connected with same edge.

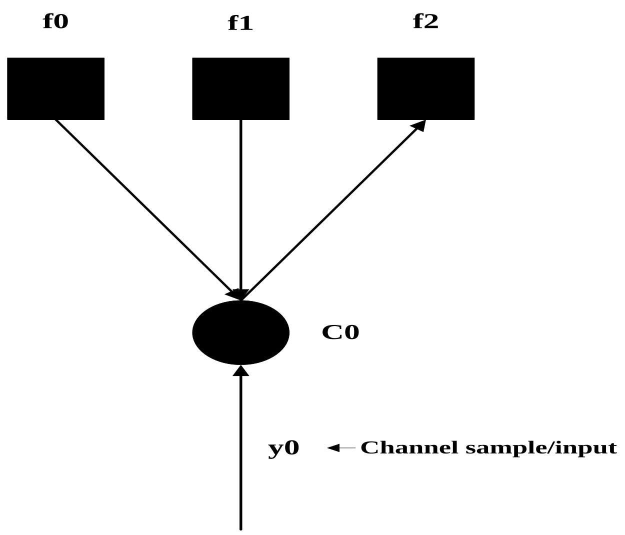

Fig 2.5 Check node Messages

As depicted in figure 2.5 the message $m_{\uparrow}$ from variable node c_0 to f_2 check node (the arrow in the subscription indicates the direction of message as in case of above figure all check nodes are lying above and message or variable node lay below. The information that passed is concerns the probability $\mathrm{P_r}(c_0 = b \mid \text{input messages})$,$b \in \{0,1\}$, the ratio of such probabilities or the logarithm of the ratio of such probabilities. From figure 2.5 we can see that the information send to check node f_2 from the variable node c_0 is the total information gathered at variable node i.e. the information from the channel and from its neighbors excluding the check node f_2, Therefore, only extrinsic information is passed.

The computation is done for such extrinsic information $m_{\uparrow ij}$ for every variable node (c_i) node/ check node (f_i) pair at each half iteration.

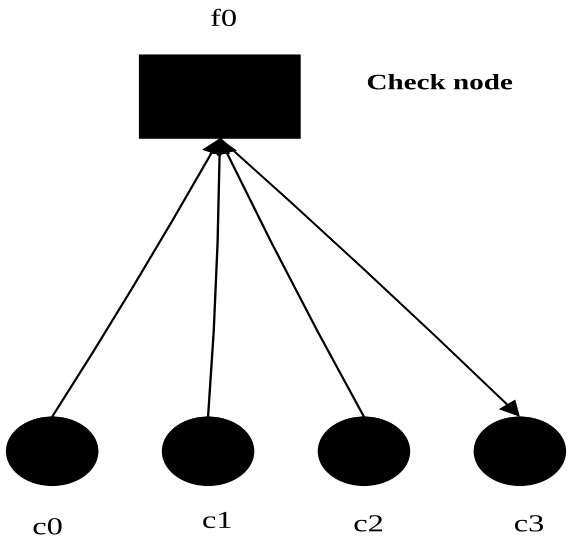

Fig 2.6 Variable Node Messages

Before it was half iteration depicted in figure 2.5 now in figure 2.6 is the other half iteration where the check node f_0 computes all input messages and pass the resulting output messages to its neighboring variable nodes, i.e. messages $m_{\downarrow}$ from check f_0 to variable nodes c_4. The information which is passed to neighbors concerns the Probability $\mathrm{P_r}(\text{check equation } f_0 \text{ is satisfied } |\text{input messages }) \; b \in \{0,1\}$ the ratio of such probabilities, or the logarithm of the ratio of such probabilities. Note, as in the previous case only extrinsic information is passed to variable node c_4. Therefore such extrinsic information $m_{\downarrow ji}$ is computed for each check node (f_j)/variable node (c_i) pair at each other half iteration.

Thus iteration process continues till some defined maximum number of iteration in the programming reaches, let us say thousand iterations are defined or some stopping criteria has been met, decoder will then computes APP, the LR or the LLR from which the decisions on the bits c_i are made. One stopping criteria is to stop iterating when $\hat{c}H^T = 0$ where ĉ is a tentatively decoded codeword.

It is assumed in the message passing decoding algorithm that the messages which are passed in the iterating process are statically independent throughout the decoding process. When y_i are independent, this independency assumption will hold true if there are no cycles in the Tanner graph.

2.5 Cycles in the Tanner graph:

To understand the cycles in the tanner graph we will take an example of Tanner graph in figure 2.7 which will show us some repetitions in the graph (blue edges indicated in figure 2.7).

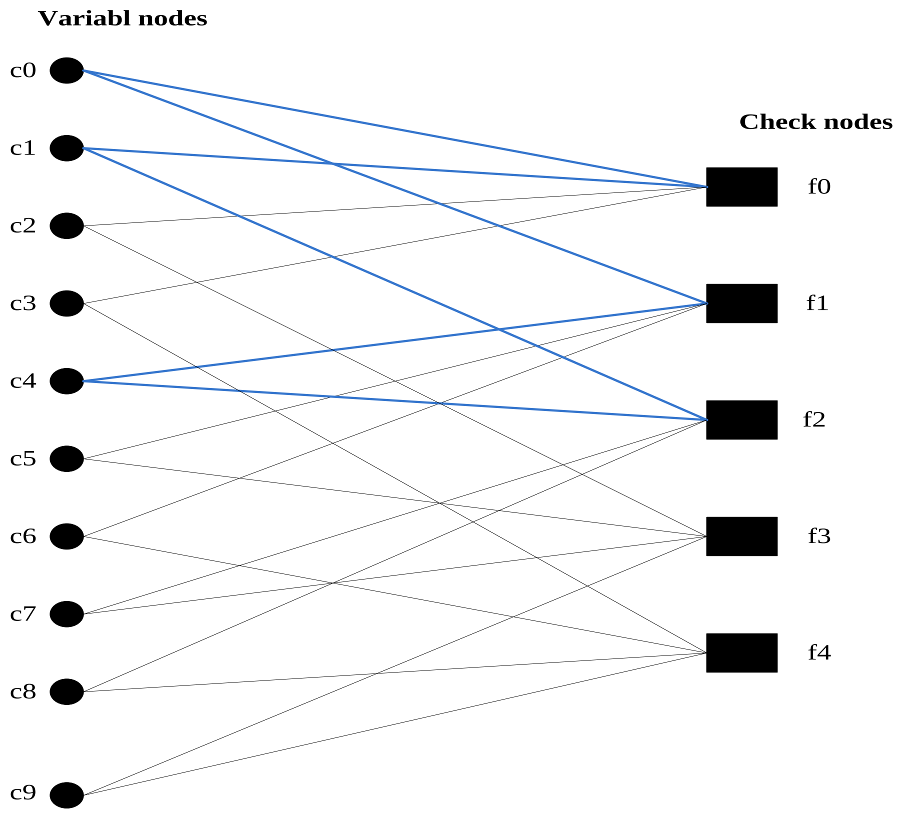

Fig 2.7 Tanner Graph with Cycles

A cycle (or loop) in a Tanner graph is defined as a path which contains some number edges v which are closed back on itself.

The Tanner graph in figure 2.7 have six cycles as exemplified by six blue bold edges. The girth γ of a Tanner graph is the minimum cycle length of the graph. Short cycles are usually preferred because they degrade the performance of iterative decoding algorithm used for LDPC codes. Lin et al. [13] showed that some configuration of length-four cycles is not harmful which means code will converge to some finite value during iteration process.

2.6 LDPC Codes Over Galois Field($GF(q)$)/ Non-Binary LDPC codes:

LDPC codes over Galois field are also known as non-binary LDPC codes. As for the sake of understanding the difference between non-binary and binary LDPC codes we will review the binary LDPC code along with non-binary section.

As we know that Gallager [1] was the father of LDPC codes (binary LDPC) he invented them in 1963, where as the non-binary LDPC codes were discovered by Davey [6]. The main difference between a binary LDPC code and non-binary LDPC code is that binary LDPC code is defined over Galois field of order 2 $GF(2)$ while the non-binary LDPC codes is defined over the Galois field of order q, $GF(q)$. Let's consider $q = 2^p$. The sparse parity check matrix **H** of size $(N - K) \times N$, where the codeword length is denoted here by N and the number information symbols by K. The number of redundancy symbol here is $M = N - K$, and the code rate is given by $R = K/N \geq 1 - M/N$, with equality if **H** is full rank (i.e., its row rank is equal to M). Connection of nodes in Tanner graph in LDPC codes defined by parameters dv and dc, where dv indicates the number of degrees of node connected to variable or message node and dc shows the number of degrees of nodes which connected with check node. Therefore this (dv, dc) made connection in Tanner graph easier for all parity check-matrices. In case of non-binary LDPC codes the non-zero value of the parity check matrices are chosen uniformly at random in $GF(q) \setminus \{0\}$ (finite Galois field).

Figures 2.4 and 2.7 are two examples of binary LDPC bipartite graph which also called factor graph [14] and Tanner graph [2]. The non-zero values of the non-binary LDPC codes of the parity check matrix **H** belongs to $GF(q) \setminus 0$. Elements of **H** on column j and row i is denoted h_{ij}. Two nodes are connected i.e. i^{th} check node and j^{th} variable node if $h_{ij} \neq 0$. Let x_j represents the variable node where j is the value of symbol, therefore the ith parity check equation is formed if

$$\sum_{j=0}^{N-1} h_{ij} x_j = 0 \qquad (2.6)$$

As per above equation two mathematical operations, addition and multiplications are performed over $GF(q)$. Same as in LDPC codes dv here is the degree of connection of variable node (same for check node) is the number of edges linked to this node where a node is said i connected or of degree i if it connected to i edges.

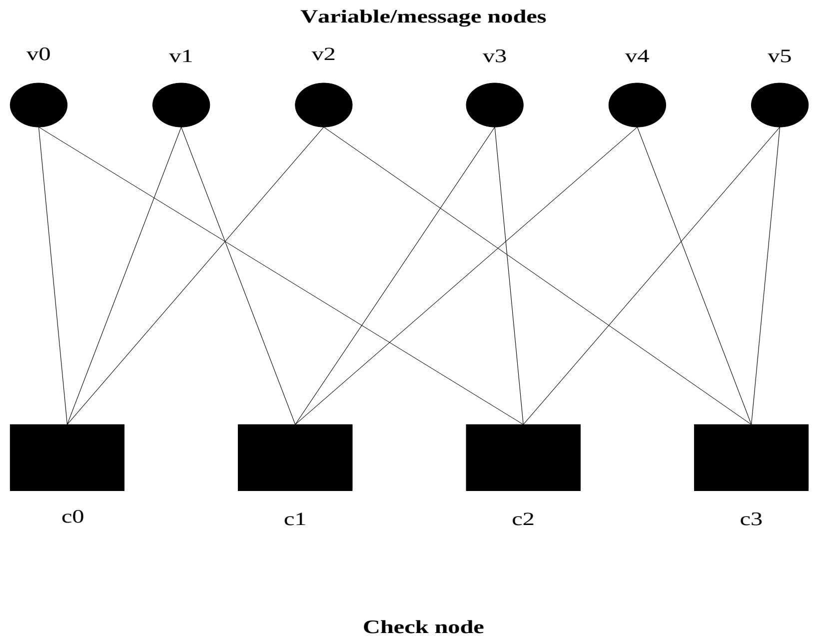

Fig 2.8 Tanner Graph for Non-binary LDPC

Figure 2.8 is bipartite graph of non-binary LDPC code and it can further explained by following polynomials for variable nodes and for check nodes.
Polynomial for variable nodes is:

$$\lambda(x) = \sum_{i=2}^{d_{v_{max}}} \lambda_i \, x^{i-1} \qquad (2.7)$$

λ_i here defined as the proportion of edges of the graph which are connected to degree i variable nodes, and $d_{v_{max}}$ is the maximum degree of variable node.

Similarly polynomial which associated with check nodes is:

$$\rho(x) = \sum_{j=2}^{d_{c_{max}}} \rho_i \, x^{j-1} \tag{2.8}$$

where ρ_i is the proportion of the edges of the graph which are connected to degree j check nodes, and $d_{c_{max}}$ is the maximum degree of check nodes. The code rate of non-binary LDPC code is defined as below:

$$R = 1 - \frac{\sum_{j=2}^{d_{c_{max}}} p_j/j}{\sum_{i=2}^{d_{v_{max}}} \lambda_i/i} \tag{2.9}$$

2.7 Decoding Algorithm for non-binary LDPC codes:

Belief Propagation Algorithm (BPA) is used as a decoding algorithm for non-binary LDPC codes. Belief propagation algorithm is also known as Sum-Product [14]. Message forwarding probabilities are spread along the edges in this algorithm where to edge two types of messages are associated, each message for one direction. Bays rule is the basic principle of BP algorithm which is applied locally and iteratively to estimate a posteriori probabilities (APP) of every codeword symbol. In this algorithm, the messages which are going into a node are independent from each other and thus the exact computation of APP has been made possible by local factorization of Bays rule in a cycle free graph.

In the decoding principle of $GF(q)$ codes where $q > 2$, messages on the edges of the graph are q sized vectors and non-binary symbols are considered as random variables in $GF(q)$ therefore BPA to computes the APP for each codeword symbol. For example, the algorithm handles the probability vector for the symbol that corresponds to variable node v_i and that probability vector is $\mathbf{p}_i = (P(v_i = \alpha_0|y_i, S_i), \dots, P(v_i = \alpha_{q-1}|y_i, S_i))$, where $P(v_i = \alpha_0|y_i, S_i)$ is the probability that shows the sent codeword symbol i is equal to $\alpha_0$following conditions are fulfilled that the output for the channel y_i is the i^{th} symbol where S_i is the event that that shows parity check equations are connected to v_i the variable node. The computation of $\mathbf{p}_i$ mainly depends on the structure of the Tanner graph through event S_i for all i.The probabilities on the graph are computed exactly up to $\frac{g}{4}$ iteration, where considering g is the shortest cycle in the Tanner graph given that input messages y_i on edges are independent ; this is also called girth of the graph.

Before describing that how BP algorithm works, some message types must be defined. To an each edge, there are two types of messages;

The messages getting in a variable/message node v of degree d_v at the t^{th} iteration
$\left\{\mathbf{l}_{p_i v}^{(t)}\right\} i \in \{ 1, \dots, d_v\}$

The messages which are going out/away from this variable/message node are
$\left\{\mathbf{r}_{v p_i}^{(t)}\right\} i \in \{ 1, \dots, d_v\}$.

Here pv show the message direction from permutation node to variable node and vp denotes the message direction from variable node to permutation node.

Similarly there are two types of messages for check node

The messages which getting in to a parity check node c are denoted as $\left\{\mathbf{r}_{p_i c}^{(t)}\right\} i \in \{1, \ldots, d_c\}$.

The messages which are going out from a check node c are denoted as $\left\{\mathbf{l}_{cp_i}^{(t)}\right\} i \in \{1, \ldots, d_c\}$.

The algorithm is composed of six different stages which is a complete description that how this algorithm works in orders to decode non-binary LDPC codes.

Initialization:
The messages which are going out from a variable node to a check node are $\mathbf{r}^{(0)}$ which are initialized with some priori information is computed at the channel output $\{p_{ch_i}[a]\} i = \{0, \ldots, N-1\}$ with probability

$$p_{ch_i}[a] = P(y_i | c_i = \alpha_a), \quad \alpha_a \in GF(q) \tag{2.10}$$

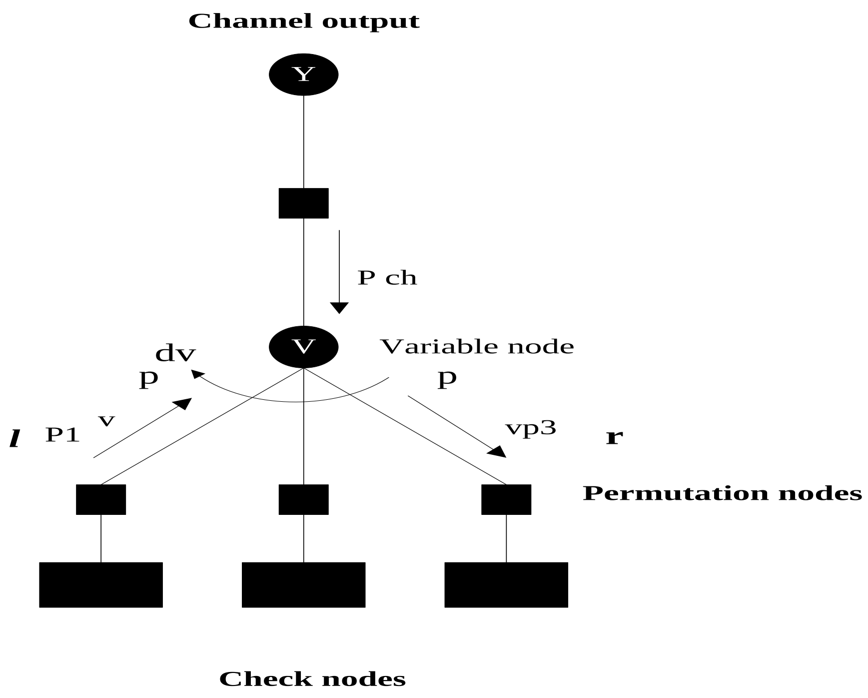

Fig 2.9 Variable Node Update

Variable node update:
The probability is sent to check node c from a variable node v for the symbol which corresponds to v to be equals to $\alpha_a \in GF(q)$ is also shown in figure 2.9.
The messages which are going out from a variable node are updated with following equation.

$$r_{vp_c}^{(t+1)}[a] = \mu_{vc} p_{ch_v}[a] \quad \prod_{j=1, j\neq c}^{d_v} l_{p_j v}^{(t)}[a] \,, \tag{2.11}$$

is $c \in \{1, \ldots, d_v\}$ and μ_{vc} is the normalization vector such that $\sum_{a=0}^{q-1} r_{vp_c}^{P}[a] = 1$.

Permutation nodes update:
This stage is the result of parity check equation

$$\sum_{j=0}^{N-1} h_{ij} x_j = 0 \tag{2.12}$$

Permutation nodes do the multiplication of non-zero value with the symbol value on each edge as depicted in figure 2.9. This multiplication corresponds to cyclic permutation of the vector messages as the non-zero value and symbol value belong to $GF(q)$.

$$r_{pc}\left[h_{ij} \otimes \alpha_k\right] = r_{vp}[\alpha_k] \quad k = \{0, \ldots, q-1\}. \tag{2.13}$$

Inverse transform is achieved with the help of inverse symbol h_{ij}^{-1} permutation when messages are going from check node to variable node ($l_{cp} \rightarrow l_{pv}$).

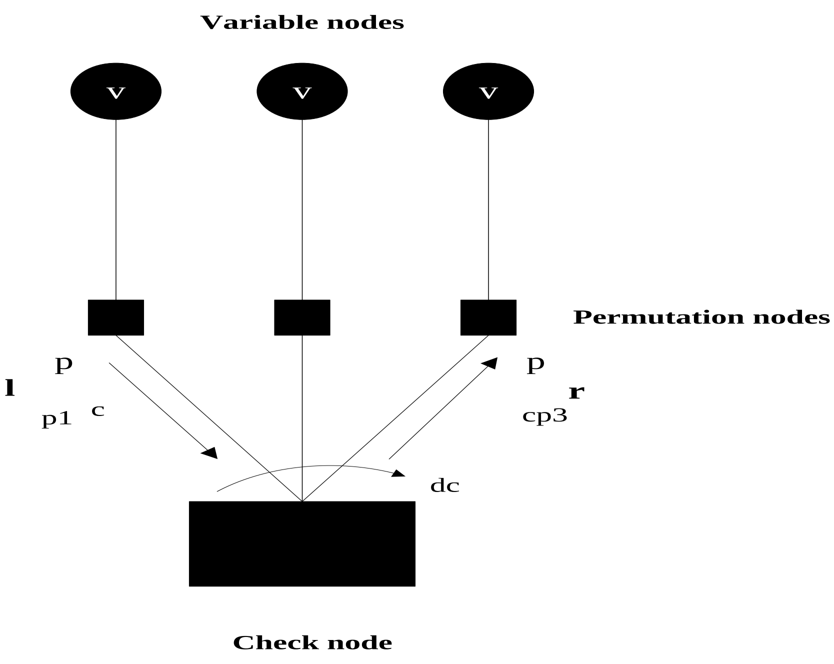

Fig 2.10 Check Node Update

Check node update:
At check node update stage $\mathbf{l}^{(t)}$ are updated messages when $\mathbf{r}^{(t)}$ are incoming messages to check node, processed and sent out to all neighbors of check node as shown in figure 2.10. Check node actually sends the probability to its neighboring variables that the parity check equation is fulfilled given its incoming messages.

$$l_{cp_v}^{P}[a] = \sum_{\substack{\alpha_{1},\ldots,\alpha_{v-1},\alpha_{v+1},\ldots,\alpha_{d_c}: \\ \oplus_{i=1,i\neq\alpha_i=\alpha_a}^{d_c}}} \prod_{j=1,j\neq v}^{d_c} r_{p_jc}^{(t)}\left[\alpha_j\right] \tag{2.14}$$

where $\oplus$ operator shows that the addition is performed over $GF(q)$. This operator is also known as addition. Therefore, addition is only performed over $GF(q)$ if the elements are summed up, where $\mathbf{l}_{cp_v}^{(t)}$ can also be expressed directly in terms of $\mathbf{r}_{vp_c}^{(t+1)}$

$$l_{cp_v}^{P}[a] = \sum_{\substack{\alpha_{1},\ldots,\alpha_{v-1},\alpha_{v+1},\ldots,\alpha_{d_c},g_1,\ldots,g_{v-1},g_{v+1},\ldots,g_{d_c}: \\ \oplus_{i=1,i\neq v}^{d_c}(g_i\otimes\alpha_i)=\alpha_a}} \prod_{j=1,j\neq v}^{d_c} r_{v_jp_j}^{(t)}\left[\alpha_j\right] \tag{2.15}$$

This equation is the update of the component a of the output vector $\mathbf{l}_{cp_v}^{(t)}$.
Figure 2.10 depicts the updated equation 2.15 where the element $l_{cp_3}^{P}[a]$ update consisting of the sum of all products $r_{p_2c}^{(t)}[a_1] \cdot r_{p_2c}^{(t)}[a_2]$ by satisfying the condition $a_1 \otimes a_2 \otimes a = 0$ with $a_1, a_2, a \in \{0, \ldots\ldots, q-1\}$.

Stopping criteria:
Following equation corresponds to the decision rule on symbol value:

$$l_n = \max_{\alpha_a} P_{ch}[a] \prod_{j=1}^{d_v} V_{p_jv}^{P}[a]. \tag{2.16}$$

The update of $\mathbf{r}^{(t)}$ and $\mathbf{l}^{(t)}$ messages is done iteratively until $\mathbf{H}\hat{\mathrm{C}} = \mathbf{0}$ condition is reached and this is also called that the decoder has converged to a codeword or the maximum number of iteration is reached and which means that the decoder did not succeed in converging to a codeword.

2.8 Root LDPC Codes:

Root LDPC codes are the codes which are design to achieve full diversity and (i.e. minimum number of cycles in the tanner graph), If there are minimum numbers of cycles in the Tanner graph then the code will be stable. LDPC codes were designed [15] for slow varying fading channel. These codes are designed for the block fading (BF) channel, which was first introduced in [16]. The block fading channel model is convenient model which is affected by slow varying fading. Example of slow varying fading are on wireless communications involving time frequency hopping or multicarrier modulation using orthogonal frequency multiplexing (OFDMA). Therefore it is a challenging task to design a code for block fading (BF) channel as compare to additive white Gaussian noise (AWGN) or independent fading channels [17].The reason is that in BF channels the random channel gain is constant during the block of symbols and it takes independent values from block to block. The word error probability of independent channel depends on hamming distances between code words while in BF channel it depends on block wise hamming distance. Therefore it is not necessary that if the code that exhibits large minimum hamming distance may not have large blockwise hamming distance; in other words we can say that if codes which are good for independent fading channels may not be good for BF channel. Another property that permutations for the symbols in independent channel have an effect on the code performance but in the case of BF channel it causes variations in the code patterns. If the code which is designed for independent fading channel can be used for transmission over a BF fading channel required to use the best permutation of its symbols. Therefore one must consider BF fading channel as non ergodic channel. Thus the word error probability of any coding scheme cannot achieve theoretical rate limits (i.e. one cannot use channel capacity but rather outage probability) [18]. The classical random-like codes are designed to achieve ergodic capacity but cannot generally approach the ideal performance limits of BF channel. So efforts are made to in [15] to design codes which are suited to the non ergodic nature of the channel.

There are two main parameters which determine error rate of coded BF channels for high signal to noise ratio (SNR) ratio: diversity order and coding gain. As the slope of the error rate curve is a function of the SNR on a log–log scale. The error probability of any coding scheme is lower-bounded by the outage probability, and the diversity order is upper-bounded by the intrinsic diversity of the channel, and it reflects the slope of the outage limit. The coding gain yields a measure of SNR proximity to the outage limit when a diversity order is achieved by a code. In [17] the maximum achievable diversity order with discrete input constellation by singleton bound. The codes which achieve singleton bound are termed as blockwise maximum distance separable (MDS). The blockwise MSD codes are outage achieving for noisy BF erasure channel [19], but that may not be outage probability limit achievable on noisy BF channels. As per facts MDS codes are important and necessary but not sufficient to achieve outage probability of the channel [17].

For BF channels, codes which include the near outage schemes based on suitable permutation are parallel turbo codes [20]. Multiplexers for convolution, turbo and repeat-accumulate code [17] which appeared one decade after the analysis of random and periodic interleaving of convolution codes on the block-erasure channel [21]. Some random ensembles of LDPC designed for ergodic AWGN channel [22], LDPC codes of irregular structures have an excellent decoding threshold but that does not have full diversity, and therefore exhibit a poor performance over BF channel. In [10] the decoding threshold for LDPC codes over BF channel has been studied. These codes unfortunately, are not designed for blockwise MDS, and thus fail to achieve the outage limit in the non ergodic setup.

Thus new families of blockwise MDS LDPC codes perposed [15] are called root LDPC codes which are based on special type of check nodes termed as rootchecks. Root LDPC codes achieve outage probability limit on block erasure channel [15], and they also perform close to that limit on Rayleigh BF channels under iterative message passing decoding.

2.9 Root LDPC Proposed Model:

In [15] a model for root LDPC code has been proposed which can attain the full diversity; and this model will be explained below. There are four types variable nodes v and and two check node types as mentioned in figure 2.11.

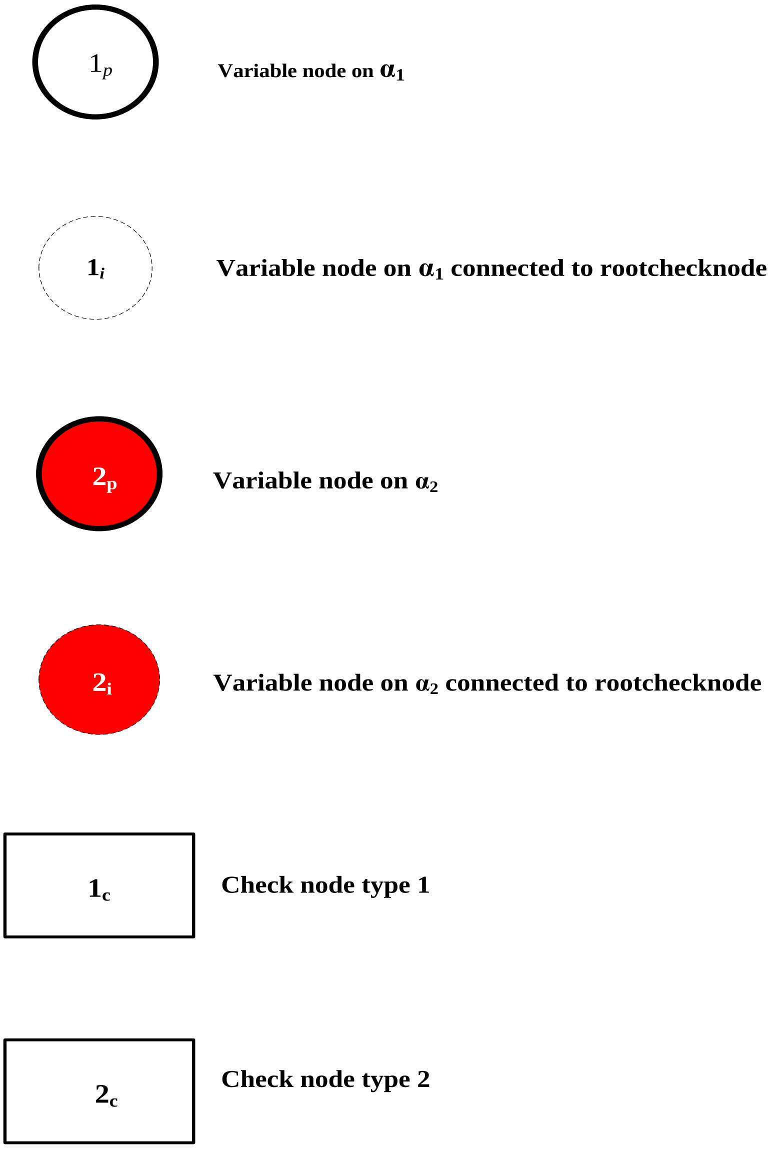

Fig 2.11 Notations for Graph Representation

In figure 2.11 p represents the parity bit or parity variable node and *i* represents the information bit or information variable node. The particular connection of the variable nodes v with the rootchecknodes is shown in figure 2.12.

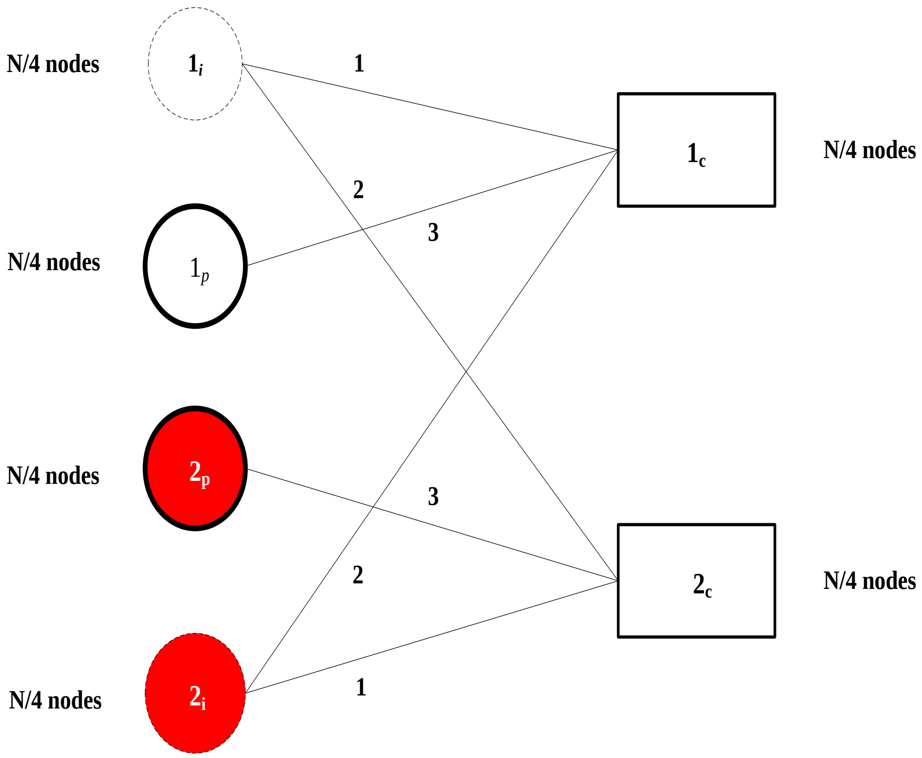

Fig 2.12 Tanner Graph of Root LDPC

As In [15] a proposition has been made for binary root LDPC codes by considering a rate 1/2, $(\lambda(x), (\rho(x))$ root LDPC code transmitted on a Rayleigh block-fading channel with $n_c = 2$ then, root LDPC code has full diversity under belief propagation decoding. This proposition is restricted to a regular (3,6) LDPC and this is elaborated below.

Let $\wedge_i^a, i = 1 \ldots . d_c - 1$ denote the input probabilistic messages to checknode c in log-ratio of degree d_c and the output message $\wedge^e$ for belief propagation is

$$\wedge^e = 2th^{-1}\left(\prod_{i=1}^{d_c-1} th\left(\frac{\wedge_i^a}{2}\right)\right) \tag{2.17}$$

where $th(x)$ denotes the hyperbolic-tangent function. The superscripts a and e stands for a priori and extrinsic, respectively. Considering the min-sum decoder, the checknode c produce the output message defined as

$$\wedge^e = \min(|\wedge_i^a|) \prod_{i=1}^{d_c-1} \text{sign}\,(\wedge_i^a). \tag{2.18}$$

2.9.1 First decoding iteration:

It has been assumed that all-zero codeword has been transmitted firstly, then after studying the first iteration, the channel crossover probability that is associated with fading $\alpha_j, j = 1,2,$ is

$$\epsilon_j = Q\left(\sqrt{2}\gamma\alpha_j^2\right). \tag{2.19}$$

The channel message for a bit variable/message node v transmitted over fading coefficient α is

$$\wedge_0 = \left(\frac{p(y|v=0,\alpha)}{p(y|v=1,\alpha)}\right) = \frac{2\alpha y}{\sigma^2} = \frac{2}{\sigma^2}(\alpha^2 + \alpha z) \quad (2.20)$$

where $y = \alpha + z$ and $z \sim \aleph\,(0, \sigma^2)$ (assuming E_s). Where E_s is the average energy per symbol

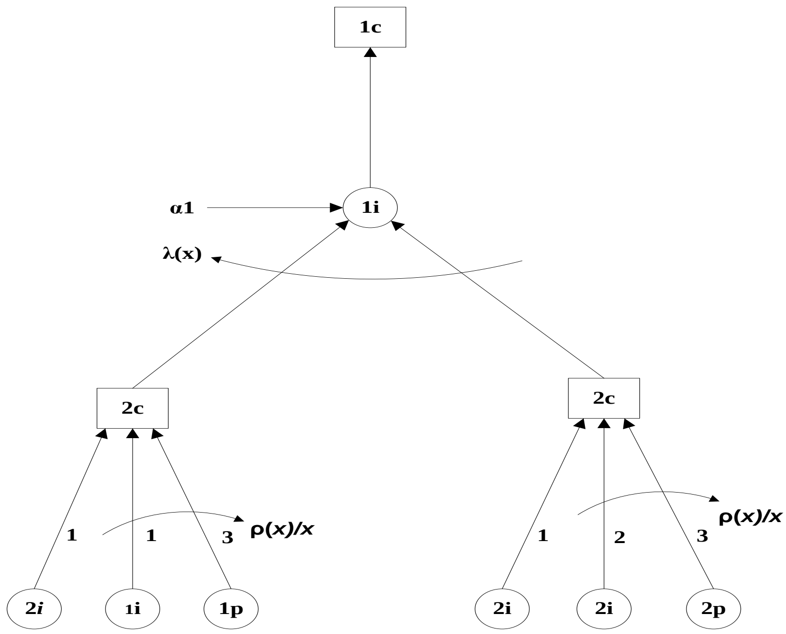

Fig 2.13 Local Neighborhood of Variable nodes 1i, this figure shows the Evolution of Messages from 1i to 1c

In figure: 2.13 the variable node v of class 1i has $\wedge_0 = \frac{2}{\sigma^2}\,(\alpha_1^2 + \alpha_1 z_0)$. Whereas variable node v also receives 3 messages $\wedge_i^e, i = 1 \ldots 3$ from its 3 neighboring check nodes. Thus the total posteriori message corresponding to variable v is $\wedge = \wedge_0 + \wedge_1^e + \wedge_2^e + \wedge_3^e$. consider $\wedge_1^e$ an extrinsic message which is generated by the rootcheck of class $1c$ connected to v. The error rate at $P_e(1i)$ is given by the negative tail of the density of $\wedge$ messages. The addition of $\wedge_2^e + \wedge_3^e$ to $\wedge_0^e + \wedge_1^e$ will not degrade the $P_e(1i)$ because the convolution with the density messages from non-rootchecks can only physically upgrade resulting density. Therefore it is sufficient to prove that message $\wedge_0^e + \wedge_1^e$ brings full diversity. The expression $\wedge_1^e$ is found by applying equation 2.17. The input messages of rootcheck are negative with probability ϵ_2

$$\wedge_1^e = S_1 \frac{2}{\alpha^2}\,(\alpha_2^2 + \alpha_2 z_1) \quad (2.21)$$

where

$$S_1 = \sum_{i\,even}^{4} \epsilon_2^i\,(1 - \epsilon_2)^{4-i} - \sum_{i\,odd}^{4} \epsilon_2^i\,(1 - \epsilon_2)^{4-i} \quad (2.22)$$

obtained

$$\wedge_1^e = (1-2\epsilon_2)^4 \; \frac{2}{\alpha^2} \; (\alpha_2^2 + \alpha_2 z_1) \; . \tag{2.23}$$

Hence the partial a posteriori log-ratio message becomes

$$\wedge_0 + \wedge_1^e = \frac{2}{\alpha^2} (\alpha_1^2 + (1-2\epsilon_2)^4 \alpha_2^2) + \alpha_1 z_0 + \; (1-2\epsilon_2)^4 \alpha_2 z_1) \tag{2.24}$$

The embedded metric $Y = \alpha_1^2 + (\alpha_1^2 + (1-2\epsilon_2)^4 \; \alpha_2^2$ guarantees full diversity. At high SNR (i.e. when $E_b/N_0 \rightarrow \infty$) Y behaves exactly as $\alpha_1^2 + \alpha_2^2$.

2.9.2 Further decoding iterations:

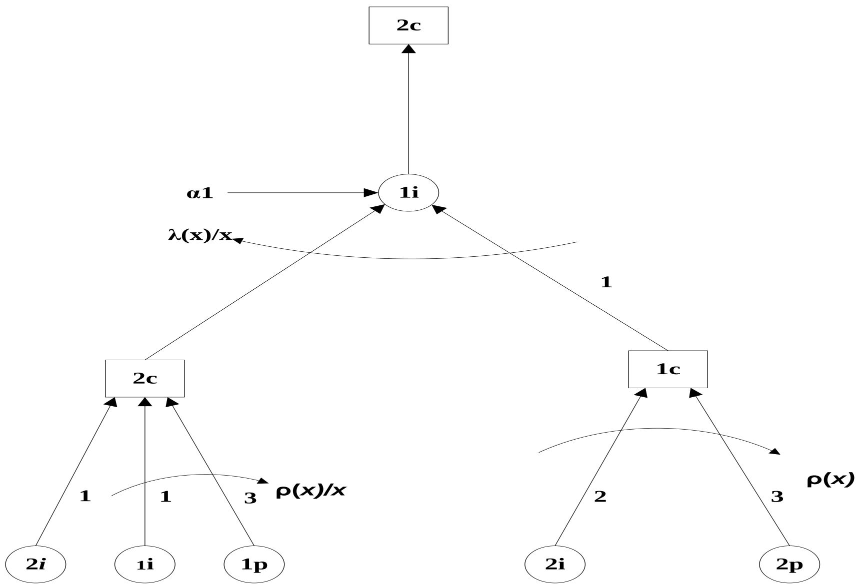

Fig 2.14 Local neighborhood of Variable node 1i, this figure shows the evolution of messages from 1i to 2c

Diversity order 2 is maintained after the first iteration in the decoding tree of bit node 1i shown in figure 12.The input to the rootcheck the information bits $2i$ have full diversity and parity bits 2_p bring always a term which is proportional to α_2^2. Density message $\wedge_0 + \wedge_1^e$ can only be improved with respect to its first iteration due the some particular structure of the root LDPC codes, therefore the full diversity is preserved.

Let us now examine the diversity for the parity bits (parity variable node). A parity bit v of class 1_p has message $\wedge_0 = \frac{2}{\alpha^2} (\alpha_1^2 + \alpha_1 z_0)$ and there are three other messages for this parity bit from its neighboring check nodes which are all of class 2_c and they are $\wedge_i^e, i = 1 \ldots 3$. Therefore, the total a posteriori message of v is $\wedge = \wedge_0 + \wedge_1^e \wedge_2^e \wedge_3^e$. Now we will determine the nature of $\wedge_i^e$ on the basis of the input messages to the check node of class 2_c as shown in figure 2.14 (root LDPC code).

After the decoding iteration it has been shown that extrinsic message produced by checked node of class 2_c satisfy the following expression:

In the case of $\alpha_2 \leq \alpha_1$

$$\wedge_i^e = \begin{cases} s\ \frac{2}{\sigma^2}(\alpha_2^2 + \alpha_2 z) \text{ with probability } \mathrm{G}^4 \geq \frac{1}{16} \\ s\ \frac{2}{\sigma^2}(\alpha_1^2 + \alpha_1 z) \text{ with probability } 1 - \mathrm{G}^4 \leq \frac{15}{16} \end{cases} \qquad (2.25)$$

where G is defined as

$$G(\alpha, \alpha, \sigma^2) = \frac{1}{2} \quad \sigma^2 > 0.$$

G defined as non-decreasing function of α_1 and decreasing function of α_2. Thus, $G \leq \frac{1}{2}$ if $\alpha_1 \leq \alpha_2$ and $G \geq \frac{1}{2}$ if $\alpha_2 \leq \alpha_1$.
For fixed σ^2 and $\alpha_2, G \rightarrow 1$ as $\alpha_1 \rightarrow +\infty$.
For fixed σ^2 and $\alpha_1, G \rightarrow 0$ as $\alpha_2 \rightarrow +\infty$.

Now determine the messages produced by checked node of class 2_c.
In case of $\alpha_2 \geq \alpha_1$

$$\wedge_i^e = \begin{cases} s\ \frac{2}{\sigma^2}(\alpha_2^2 + \alpha_2 z) \text{ with probability } \mathrm{G}^4 \leq \frac{1}{16} \\ s\ \frac{2}{\sigma^2}(\alpha_1^2 + \alpha_1 z) \text{ with probability } 1 - \mathrm{G}^4 \geq \frac{15}{16} \end{cases} \qquad (2.26)$$

Thus it is concluded for parity bits, that the output message has the first diversity order with a probability $> 1/2 \times 15/16$. The error probability of parity bits will have diversity order 1 instead of diversity order 2. Hence the above description will be valid for further decoding iterations.

2.10 Block Fading Channel Model:

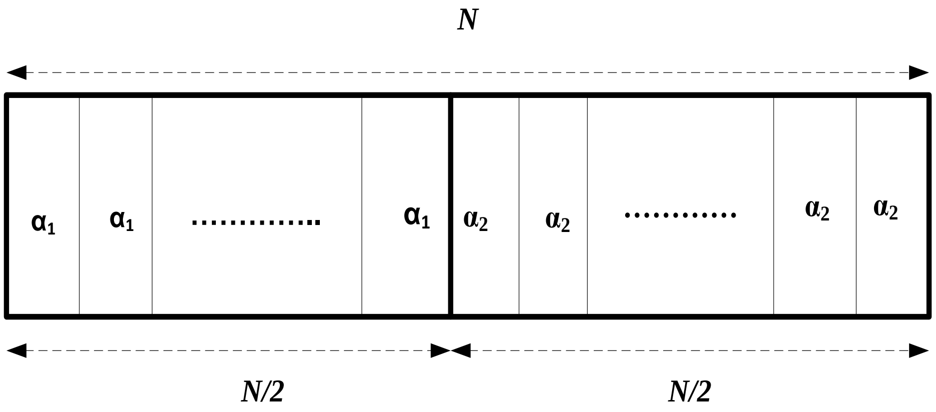

Fig 2.15 Block fading channel model

Figure 2.15 shows the channel model for a BF fading channel. Consider that N binary digits are being transmitted on a BF channel, with n_c independent fading gains (whose values are from the channel state) affect each codeword. The length N is a multiple of fading gains (n_c) with $l \triangleq N/n_c$ that denotes the number of bits per fading block. The received signal when symbol x_i is transmitted.

$$y_i = \alpha_j x_i + z_i \ . \tag{2.27}$$

where $y_i \in \mathbb{R},\ i = 1 \dots . N$ and $j = 1 + \left[\frac{(i-1)}{l}\right]$, with $[r]$ that denotes the integer part of the real number r. Where α_j is the nonnegative real number which is the fading gain at block $j = 1 \dots n_c$. The symbol x_i which are chosen from the BPSK alphabet, where $x_i = \mp\sqrt{E_s}$ and $z_i \sim \aleph\,(0, \sigma^2)$, $\sigma^2 = N_0/2$ are noise samples. We assume perfect channel state information (CSI) at the receiver side and the channel gains are Rayleigh distributed from codeword to codeword and from block to block. The average SNR per symbol is $\gamma = \frac{E_s}{N_0}$, when the information rate is R bits per channel use. And the average SNR per bit is $\frac{E_b}{N_0} = \frac{\gamma}{R}$. Figure: 2.15 illustrate the channel model for $n_c = 2$ and $l = N/2$.

3 DESIGN ANALYSIS AND SIMULATION

3.1 Design Analysis:

The design analysis is the implementation of the following polynomials in this thesis work. The equations being used in the simulations.

$$\lambda_{ME} = \frac{1}{2}\sum_i \left(\frac{\lambda_i}{i}\mu_1 x_1^i + \frac{(i-1)\lambda_i}{i}\mu_1 x_2^i + \lambda_i \mu_1 x_3^i + \lambda_1 \mu_2 x_4^i + \frac{(i-1)\lambda_i}{i}\mu_2 x_5^i + \frac{\lambda_i}{i}\mu_2 x_6^i\right). \quad (3.1)$$

$$\rho_{ME} = \frac{1}{2}\sum_j \binom{i}{j} f_e^j x_4^j g_e^{i-j} x_5^{i-j} + x_6 \sum_k \binom{i}{k} f_e^k x_3^k g_e^{i-k} x_2^{i-k}. \quad (3.2)$$

Equation (3.1) represents the variable/message nodes v on the left side of the bipartite graph mentioned. Figure 2.12 while the equation (3.2) represents the rootchecks (check node c). The variable μ_1 and μ_2 are related to two fading channels and $x_1, x_2, \ldots. x_6$ are variable which represent the six different types of edges used in the bipartite graph in the figure 2.7. We have two types variable node which are information bitnodes and parity bitnodes, so remember that edge x_3 and x_4 are the parity edges, and all other remaining edges are the information edges. f_e represents the probability that an edge that connects parity node to check node. Similarly g_e represents the probability that an edge which connects to an information node to check node. Clearly $f_e + g_e = 1$.

The following equations are for the six different edge types in the Tanner graph.

$$\bar{d}_v = \sum_i \lambda_i / i \qquad \bar{d}_c = \sum_i \rho_i / i \quad (3.3)$$

$$\dot{\lambda}(x) = \bar{d}_v \sum_i \lambda_i / i\, x^{i-1} \qquad \dot{\rho}(x) = \bar{d}_c \sum_i \rho_i / i\, x^{i-1} \quad (3.4)$$

$$\tilde{\lambda}(x) = \frac{\bar{d}_v}{\bar{d}_v - 1}\sum_i \frac{\lambda_i (i-1)}{i} x^{i-2} \qquad \tilde{\rho}(x) = \frac{\bar{d}_c}{\bar{d}_c - 1}\sum_i \frac{\rho_i (i-1)}{i} x^{i-2} \quad (3.5)$$

3.2 Tanner Graph Model of Diversity 2:

Tanner graph as shown in figure 3.1 has promised diversity of order 2. Figure 2.12 show that there are four types of variable node. $(1_i, 1_p, 2_i, 2_p)$ and two checks node $(1_c, 2_c)$ and six different type of edge. The variable node $1_i, 1_p$ shows the information and parity nodes which are in a codeword and sent through the fading channel1. Similarly variable node $2_i, 2_p$ are the second information and parity bits respectively which sent through fading channel 2.

Thus 1_i and 2_i are information nodes, and $1_p, 2_p$ are parity nodes, consideration has been made that two information nodes are connected to same rootcheck (check node) while other all other edges are connected to another rootcheck (check node) and each parity variable node connected to different root check (check node). Thus edge connections are made accordingly $1_i \rightarrow 1_c, 1_i \rightarrow 2_c, 1_p \rightarrow 1_c, 2_p \rightarrow$

$2_c, 2_i \rightarrow 1_c, 2_i \rightarrow 2_c$. The following structure guarantees code rate of $\frac{1}{2}$ and transmit diversity of 2, which is the maximum diversity that can be obtained for two transmitting channels. For non-binary root LDPC codes symbols are assigned to variable node from $GF(q)$. Where q^{2m} and $m = 0,1,2,3 \ldots$

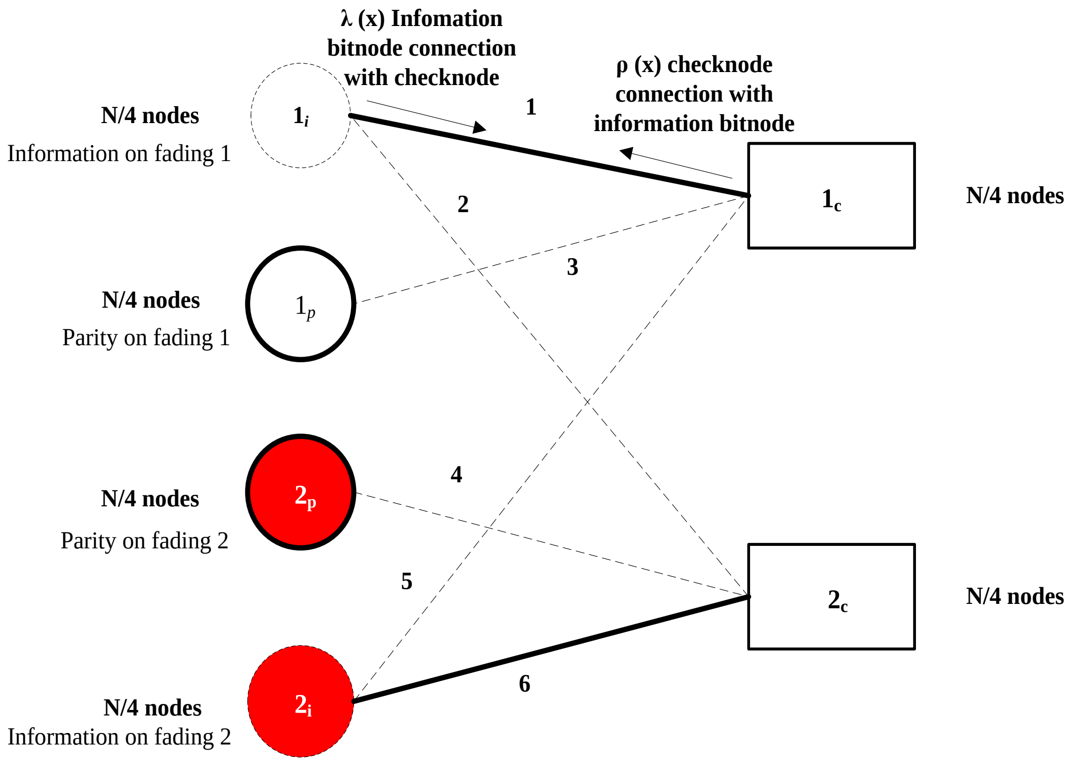

Fig 3.1 Structure of a (λ,ρ) Root LDPC Ensemble of Diversity 2

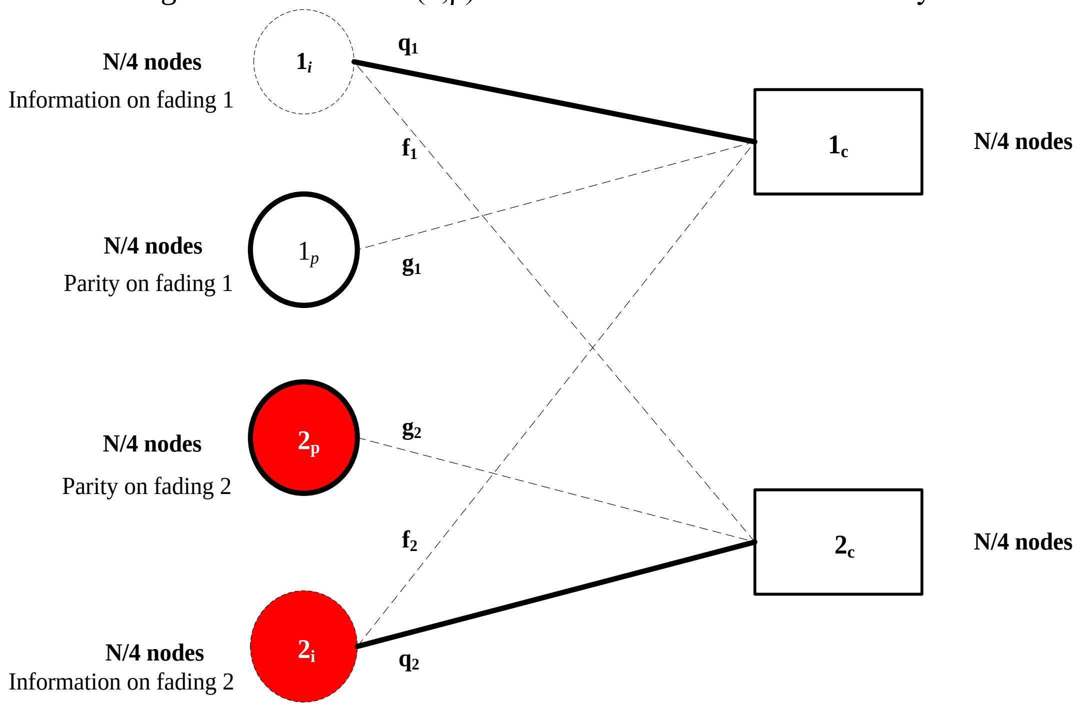

Fig 3.2 Probabilistic edge connection structure of a Root LDPC (λ,ρ) ensemble of diversity 2

Therefore for the non-binary root LDPC symbols are being divided into two equal sub-words in a codeword and those two subwords are transmitted on two independent BF channels of coefficients

$\alpha_1$and α_2. Parity and information symbols are differentiated inside each subword. It should be taken into account that full diversity property holds only for information symbols and not the parity symbols. Two subwords are erased by the fading channels (i.e. $\alpha_1 = \alpha_2 = 0$). Due to channel independency it happens with the probability p^2 where p is considered as probability to get $\alpha = 0$ for BF channel. Thus the power of p is 2, so we have obtained the diversity 2 which is the maximum possible value obtain.

3.3 Implementation:

3.3.1 Iterative Threshold γ^* versus Fading:

Two fading coefficients of two channels are denoted by $\alpha_{1,}\ \alpha_2$, where $\gamma = {}^{1}/_{\sigma^2}$ is SNR and $\gamma^*(\alpha_{1,}\alpha_2)$ is the function for iterative decoding threshold the root ensemble. There fore

$$\gamma^*(\alpha_{1,}\alpha_2) = \frac{\gamma^*\left({}^{\alpha_{1,}}/_{\alpha_2},1\right)}{\alpha_2^2} = \gamma^*(\alpha_{1,}\alpha_2) = \frac{\gamma^*\left(1,{}^{\alpha_{2,}}/_{\alpha_1}\right)}{\alpha_1^2} \qquad (3.6)$$

The pdf (probability density function) of the LLR channel estimate, relating to the threshold is,

$$p \wedge_0^* (\alpha_{1,}\alpha_2) = \frac{1}{2}\aleph\,(2\alpha_1^2\gamma^*, 42\alpha_1^2\gamma^*) + \frac{1}{2}\aleph\,(2\alpha_2^2\gamma^*, 42\alpha_2^2\gamma^*) = \frac{1}{2\alpha_1^2}\aleph\,(2\gamma^*, 4\gamma^*) + \frac{1}{2\alpha_1^2}\aleph\,\left(2\frac{\alpha_2^2}{\alpha_1^2}\gamma^*, 4\frac{\alpha_2^2}{\alpha_1^2}\gamma^*\right) = \frac{p\wedge_0^*\left(1,\frac{\alpha_2}{\alpha_1}\right)}{\alpha_1^2} \qquad (3.7)$$

Similarly $p \wedge_0^* (\alpha_{1,}\alpha_2) = \frac{p\wedge_0^*\left(1,\frac{\alpha_1}{\alpha_2}\right)}{\alpha_2^2}$

Equation (3.7) states that for a given root ensemble it is sufficient to compute the function $\gamma^*(1,\alpha)(\alpha \epsilon R^+)$ in order to define iterative threshold for any couple of fading coefficients (α_1, α_2).Thus from Eq. (3.7) it is observed, that $\gamma^*(1,\alpha)$ for the case of 2 fading channel is of the following form

$$\gamma^*(1,\alpha) = \frac{b(\alpha)}{\alpha^2} + \frac{d(\alpha)}{\alpha} + b(\alpha), \qquad (3.8)$$

where

$$b(\alpha) = K_b e^{-C_b\alpha} \qquad (3.9)$$

$$d(\alpha) = K_d(1 - e^{-C_d\alpha}) \qquad (3.10)$$

Note that in case of Root LDPC codes, $b(\alpha) \approx 0$, thus Eq. (3.9) will be $\gamma^*(1,\alpha) = \frac{d(\alpha)}{\alpha}$. By approximating 2 parameters (numerically for the moment) for a Root LDPC ensemble to estimate $\gamma^*(1,\alpha)$. In figure (3.3) we have presented the estimation done for (2,4) code.

3.3.2 Simulations Results/Graphs:

In non-binary Root-LDPC code, each variable node is assigned a symbol instead of bit (in case of binary Root LDPC code) from $GF(q)$ field, while check node performs operation over $GF(q)$. Each edge and the linear transform of the form $\alpha x, \quad \alpha \in GF^*(q)$ is being assigned.

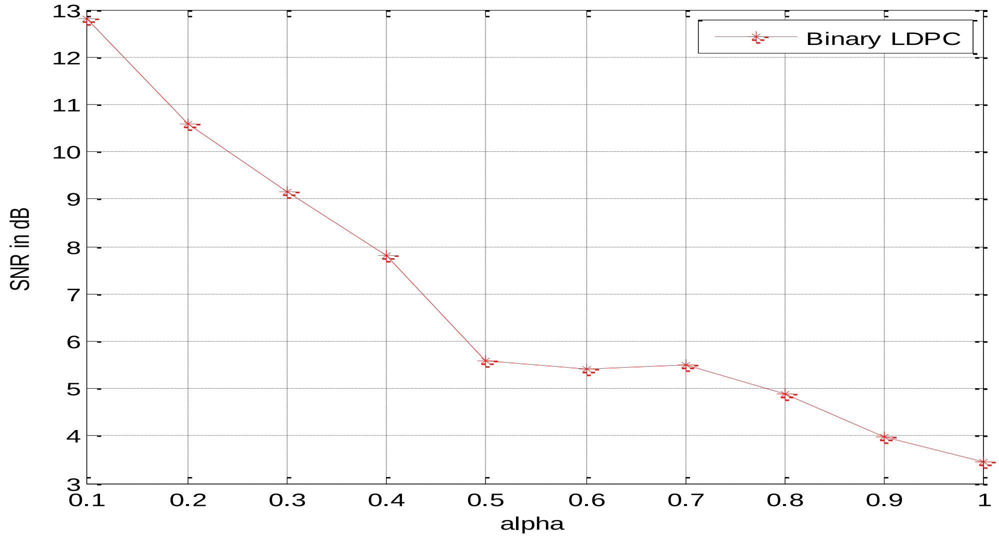

Fig 3.3 Estimated threshold level for (2,4) binary root LDPC code

In figure 3.3 the estimated threshold level for (2,4) binary Root LDPC starts from 3.4dB and it goes to 12.80dB at different values of alpha.

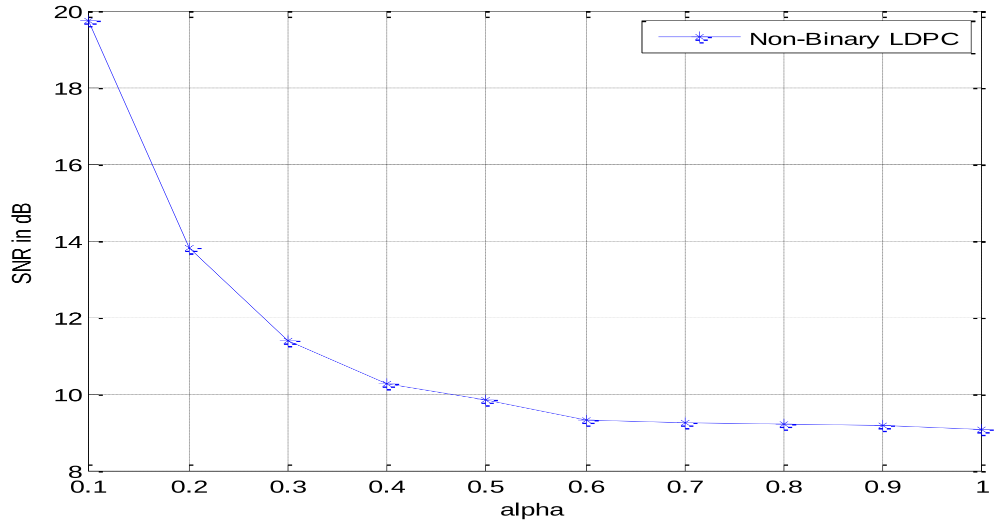

Fig 3.4 Estimated threshold level for (2,4) Non-Binary Root LDPC code

In figure 3.4 the estimated threshold level for (2,4) non-binary Root LDPC starts from 9.1 dB and it goes to 19.74 dB at different values of alpha. So if we compare the threshold level of the two different codes (binary and non-binary Root LDPC codes,) it is clear from the graph that in case of non-binary Root LDPC code the SNR value is higher than the in case binary Root LDPC code at different values of alpha. Alpha is the fading coefficient of the block fading channel.

CONCLUSION:

In this thesis/research work we studied the full diversity LDPC (root LDPC) codes for non-ergodic transmission channels. The study was mainly divided in two sections one was the theoretical part which included study of different research papers and second part was the implementation of the idea. The threshold level has been computed for binary and non-binary root LDPC using C++. Iterative decoding method has been used for detecting the received code word. Therefore on basis of Simulation results it can be concluded that non-Binary Root LDPC code provides higher threshold value as compare to binary Root LDPC code.

REFERNCES

1- R. Gallager, “Low-density parity check codes,” *IRE Trans. Information theory,* 21-28, Jan. 1962.

2- R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Information Theory*, 533-547, Sept. 1981.

3- D. Mackay and R. Neal, “Good codes based on very sparse matrices,” in Cryptography and coding, 5th IMA conf., C. Boyd, Ed., Lecture notes in computer science, 100-111, Berlin Germany, 1995.

4- D. Mackay “Good codes based on very sparse matrices,” IEEE *trans Information Theory*, March. 1999.

5- Alon and M.luby, “A linear time erasure-resilient code with near optimum recovery,” *IEEE Trans Inf. Theory,*1732-1736, Nov. 1996.

6- M. C. Davey and D. Mackay, “Low density parity check codes over GF (q),” *IEEE comm. Letters,*165-167, June. 1998.

7- X. Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge-growth tanner graphs,” *IEEE Trans. Inform. Theory* 386–398, Jan. 2005.

8- L. Barnault and D. Declercq, “Fast Decoding Algorithm for LDPC over GF (2q),” The Proc. 2003 Inform*. Theory Workshop,* 70–73, 2003.

9- G. David Forney, Jr., “Introduction to modern coding theory,” *IEEE on Trans Information theory* 2002.

10- X. Jin, A. W. Eckford, and T. E. Fuja, “Analysis of LDPC decoding for correlated and uncorrelated block fading channels,” 2004 *IEEE Int. Symp. Inf. Theory*, Chicago, IL, USA, 2004.

11- C. E. Shannon, “Communication in the presence of noise,” *Proceedings of the Institute of Radio Engineers,* 10–21, 1949.

12- J.Pearl, “Probabilistic Reasoning in intelligent systems,” San Mateo, CA: Morgan Kaufmann, 1988.

13- H. Tang, J. Xu, S. Lin, and K. Abdel-Ghaffar, “Codes on finite geometries,” *IEEE Trans. Info. Theory,* 2002.

14- F. Kschischang , B. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm,"*IEEE Transactions on Information Theory,* 498-519, Feb. 2001.

15- Joseph J. Boutros, Albert Guill´en i F`abregas, Ezio Biglieri, and Gilles Z´emor "Low-Density Parity-Check Codes for Nonergodic Block-Fading Channels," *IEEE Transactions on Information Theory,* 5 Oct. 2007.

16- L. H. Ozarow, S. Shamai (Shitz), and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Trans. Vehicular Tech.*, 359–378, May 1994.

17- A. Guill´en i F`abregas and G. Caire, "Coded modulation in the block-fading channel: Coding theorems and code construction," *IEEE Trans. Inf. Theory,* 91–114, Jan. 2006.

18- E. Biglieri," Coding for wireless channels", New York: Springer, 2005.

19- A. Guill´en i F`abregas, "Coding in the block-erasure channel," *IEEE Trans. Inf. Theory,* 5116–5121, Nov. 2006.

20- J. J. Boutros, G. M. Kraidy, and N. Gresset,"Near outage limit space-time coding for MIMO channels," *Inaugural ITA Workshop, UCSD,* San Diego, California, Feb. 2006.

21- A. Lapidoth, "The performance of convolutional codes on the block erasure channel using various finite interleaving techniques," *IEEE Trans. Inf. Theory,* 1459–1473, Sept. 1994.

22- J. Hou, P. H. Siegel, and L. B. Milstein, "Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels," *IEEE J. Selec. Areas Commun* , May 2001.