Implementation of Low Density Parity Check (LDPC) codes

Theresh Babu Benguluri, Sandeep Kumar Khyalia and G V V Sharma*

Contents

***Abstract*—A brief description of the design and implementataion of LDPC codes using (7,4) Hamming parity check matrix is provided. Calculation of LLR's for higher order mapping schemes are decribed by taking examples of QPSK and 8-SPK.**

1. Introduction

Let the Channel model be,

$$Y_k = X_k + V_k, \quad k = 0, \ldots, 6 \tag{1.1}$$

where X_k is the transmitted symbol in the kth time slot using the BPSK modulation and $V_k \sim \mathcal{N}\left(0, \sigma^2\right)$.

2. Encoding

LDPC codes are popular linear block codes with closest Shannon limit channel capacity [?]. As an example, lets take (7,4) Hamming parity check matrix.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

The above H matrix has $k = 4$ i.e $\mathbf{m} = \begin{pmatrix} m_0 & \ldots & m_3 \end{pmatrix}$ information bits and $m = n-k = 3$ i.e $\mathbf{p} = \begin{pmatrix} p_0 & p_1 & p_2 \end{pmatrix}$ parity bits. The code word length $n = 7$ i.e $\mathbf{c} = \begin{pmatrix} \mathbf{m} & \mathbf{p} \end{pmatrix}$. Encoding can be carried out

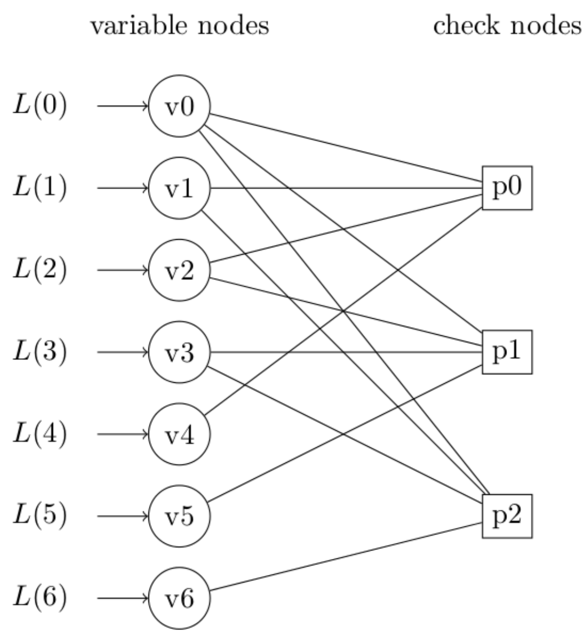

Fig. 1: Tanner Graph Representation for (7,4) Hamming parity check matrix

by using

$$H \times c^T = 0 \tag{2.2}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ p_0 \\ p_1 \\ p_2 \end{bmatrix} = 0 \tag{2.3}$$

solving we get

$$p_0 = m_0 \oplus m_1 \oplus m_2 \tag{2.4}$$

$$p_1 = m_0 \oplus m_2 \oplus m_3 \tag{2.5}$$

$$p_2 = m_0 \oplus m_1 \oplus m_3 \tag{2.6}$$

This is called Systematic Encoding.i.e Encoder will ensures information bits followed by parity bits.

3. LDPC Encoding- DVB S2

A. Parity Check Matrix Generation

To determine $n_{ldpc} - k_{ldpc}$ parity bits $\left(p_0 \quad p_1 \quad \dots \quad p_{n_{ldpc}-k_{ldpc}-1}\right)$

1) every block of k_{ldpc} information bits $\left(i_0 \quad i_1 \quad \dots \quad i_{k_{ldpc}-1}\right)$
2) Initialize $p_0 = p_1 = p_2 = ... = p_{n_{ldpc}-k_{ldpc}-1} = 0$
3) Accumulate the first information bit, i_0 at parity bit address specified in the row of table of Addresses of parity bit for $n_{ldpc} = 64800$ and $n_{ldpc} = 16200$
4) Example- First row for Rate $\frac{1}{2}$ for $n_{ldpc} = 64800$ is $[54, 9318, 14392, 27561, 26909, 10219, 2534, 8597]$
5) update parity bit:-
$p_{54} = p_{54} \oplus i_0$
$p_{9318} = p_{9318} \oplus i_0$
$p_{14392} = p_{14392} \oplus i_0$
$p_{27561} = p_{27561} \oplus i_0$
$p_{26909} = p_{26909} \oplus i_0$
$p_{10219} = p_{10219} \oplus i_0$
$p_{2534} = p_{2534} \oplus i_0$
$p_{8597} = p_{8597} \oplus i_0$
6) For next 359 information bits, i_m, $m = 1, 2, ...359$ accumulate i_m at parity bit address $(x + ((m)mod360) * q)mod(n_{ldpc} - k_{ldpc})$
where x denotes the address of the parity bit accumulatro corresponding to the first bit i_0 and q is a code rate dependent constant given in standard DVB-S2.
7) For example q=90 for rate $\frac{1}{2}$, so for information bit i_1-
$p_{144} = p_{144} \oplus i_1$
$p_{9408} = p_{9408} \oplus i_1$
$p_{14482} = p_{14482} \oplus i_1$
$p_{27651} = p_{27561} \oplus i_1$
$p_{26999} = p_{26999} \oplus i_1$
$p_{10309} = p_{10309} \oplus i_1$
$p_{2624} = p_{2624} \oplus i_1$
$p_{8687} = p_{8687} \oplus i_1$
8) For the 361^{st} information bit i_{360}, the address of the parity bit accumulators are given in the second row of the DVB-S2 Addresses of parity bit table.
9) In similiar manner the address of the parity bit accumulator for the following 359 information bits, i_m, m=361,362,....,719 are obtained using the formula-
$(x + ((m)mod360) * q)mod(n_{ldpc} - k_{ldpc})$
10) In similar manner for every group of 360 new information bits and a new row from DVB-S2 Addresses of parity bit table are used to find the addresses of parity bit accumulators.
11) After all of the information bits are exhausted, the final parity bits are obtained as follows-
$p_i = p_i \oplus p_{i-1},\ i = 1, 2, ...n_{ldpc} - k_{ldpc} - 1$
12) Final content of p_i $i = 1, 2, ...n_{ldpc} - k_{ldpc} - 1$ is equal to the parity bit p_i.

4. Decoding

A. Useful Calculations for proceeding LDPC Decoding

1) Calculation of Input Channel Log Likelihood Ratio LLR

$$L(x_j) = \log\left(\frac{Pr(x_j = 1|y)}{Pr(x_j = -1|y)}\right) \quad X = 1 - 2c \tag{4.1}$$

$$= \log\left(\frac{f(y|x_j = 1)Pr(x_j = 1)}{f(y|x_j = -1)Pr(x_j = -1)}\right) \tag{4.2}$$

$$= \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_j-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_j+1)^2}{2\sigma^2}}}\right) \tag{4.3}$$

$$= \log\left(e^{\frac{2y_j}{\sigma^2}}\right) \tag{4.4}$$

$$L(x_j) = \frac{2y_j}{\sigma^2} \tag{4.5}$$

2) Check Node Operation :
Lets assume that we have initilized all LLR values to variable nodes and we sent to check nodes. V_j represents all the variable nodes which are connected to j^{th} check node. Using the min-sum approximation [?], the message from j^{th} check node to i^{th} variable node given by, since parity node equation for the first check node is $p_0 = m_0 + m_1 + m_2 + m_4$. we need to calculate

$$L_{ext0,0} = \log\left(\frac{Pr(x_0 = 0|y_1, y_2, y_4)}{Pr(x_0 = 1|y_1, y_2, y_4)}\right) \tag{4.6}$$

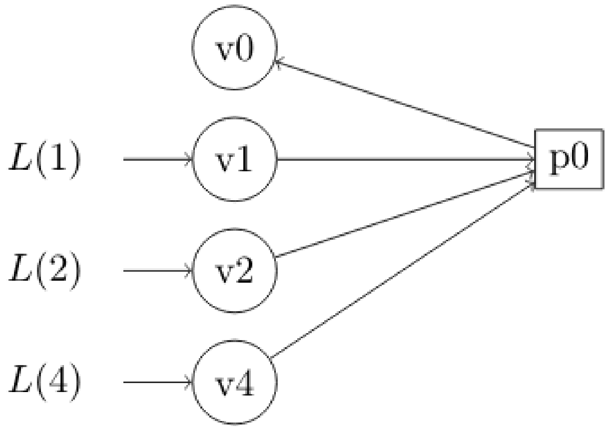

Fig. 2: Check node operation

Defining,

$$L_1 = \log\left(\frac{Pr(x_1 = 0|y_1)}{Pr(x_1 = 1|y_1)}\right) = \log\left(\frac{p_1}{1 - p_1}\right) \quad (4.7)$$

$$L_2 = \log\left(\frac{Pr(x_2 = 0|y_2)}{Pr(x_2 = 1|y_2)}\right) = \log\left(\frac{p_2}{1 - p_2}\right) \quad (4.8)$$

$$L_4 = \log\left(\frac{Pr(x_4 = 0|y_4)}{Pr(x_4 = 1|y_4)}\right) = \log\left(\frac{p_4}{1 - p_4}\right) \quad (4.9)$$

(4.10)

Using Table. I we can find the

c_0	c_1	c_2	c_4
0	0	0	0
1	0	0	1
1	0	1	0
0	0	1	1
1	1	0	0
0	1	0	1
0	1	1	0
1	1	1	1

TABLE I: Probability of a varibale node from other check nodes

$$p_0 = Pr(c_0 = 0|c_1, c_2, c_4) \quad (4.11)$$

$$\begin{aligned} p_0 &= p_1p_2p_4 + p_1(1 - p_2)(1 - p_4) \\ &\quad + (1 - p_1)p_2(1 - p_4) + (1 - p_1)(1 - p_2)p_4 \\ 1 - p_0 &= p_1p_2(1 - p_4) + p_1(1 - p_2)p_4 \\ &\quad + (1 - p_1)p_2p_4 + (1 - p_1)(1 - p_2)(1 - p_4) \end{aligned}$$

by rearranging above equations,

$$p_0-(1-p_0) = p_1-(1-p_1)+p_2-(1-p_2)+p_4-(1-p_4) \quad (4.12)$$

Where p_i is the probability. getting message from check to variable node by taking all variable node informations.

$$L_{ext0,0} = \left(\prod_{k\in V_j\backslash i} \alpha_{k,0}\right) f\left(\sum_{k\in V_j\backslash i} f\left(\beta_{k,0}\right)\right) \quad (4.13)$$

Where,

$$\alpha_{k,j} = sign(L_{k,j}) \quad (4.14)$$

$$\beta_{k,j} = \left|L_{k,j}\right| \quad (4.15)$$

$$f(x) = -\log\left(\tanh\frac{x}{2}\right) \quad (4.16)$$

Using the Fig 3 and using its 45° symmetricity,

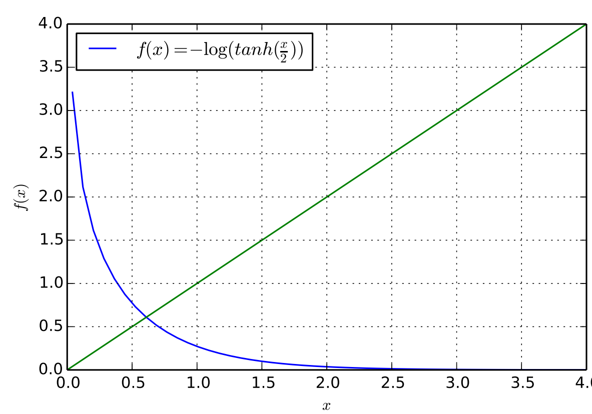

Fig. 3: Plot of function $f(x)$

we can approximate the above equation as, given by minimum sum approximation [?]

$$f\left(\sum_{k\in V_j\backslash i} f\left(\beta_{k,0}\right)\right) \approx f\left(f\left(\min_{k\in V_j\backslash i}\left(\beta_{k,0}\right)\right)\right) \quad (4.17)$$

$$= \min_{k\in V_j\backslash i}\left(\beta_{k,0}\right) \quad (4.18)$$

Combining (4.18) in (4.13),

$$L(r_{j=0,i=0}) = \left(\prod_{k\in V_j\backslash i} \alpha_{k,0}\right)\left(\min_{k\in V_j\backslash i}\left(\beta_{k,0}\right)\right) \quad (4.19)$$

3) Variable Node Operation :
Let C_i denotes all the check nodes connected to i^{th} variable node. The message from i^{th} variable

node to j^{th} check node given by,

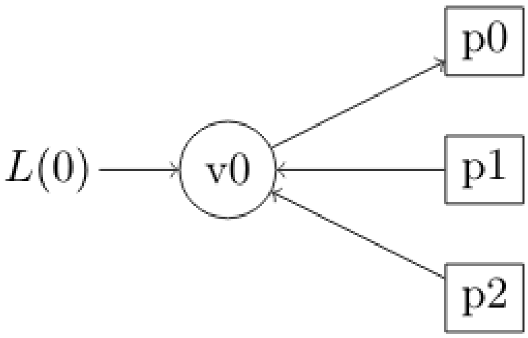

Fig. 4: Variable node operation

$$L(q_{i=0,j=0}) = \log\left(\frac{Pr(x_j = 1|y_0, y_1, y_2)}{Pr(x_j = -1|y_0, y_1, y_2)}\right) \quad X = 1 - \tag{4.20}$$

$$= \log\left(\frac{f(y_0, y1, y_2|x_j = 1)Pr(x_j = 1)}{f(y_0, y_1, y_2|x_j = -1)Pr(x_j = -1)}\right) \tag{4.21}$$

$$= \log\left(\frac{\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^3 e^{\frac{-(y_0-1)^2}{2\sigma^2}} e^{\frac{-(y_1-1)^2}{2\sigma^2}} e^{\frac{-(y_2-1)^2}{2\sigma^2}}}{\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^3 e^{\frac{-(y_0+1)^2}{2\sigma^2}} e^{\frac{-(y_1+1)^2}{2\sigma^2}} e^{\frac{-(y_2+1)^2}{2\sigma^2}}}\right) \tag{4.22}$$

$$= \log\left(e^{\frac{2(y_0+y_1+y_2)}{\sigma^2}}\right) \tag{4.23}$$

$$L(q_{i=0,j=0}) = \frac{2(y_0 + y_1 + y_2)}{\sigma^2} = L(x_i) + \sum_{k \in C_i \setminus j} L(r_{ki}) \tag{4.24}$$

B. Message Passing Algorithm using min-sum Approximation

Transmitted frames = N, Total number of bits = N × 7 and Total number of information bits = N × 4. For Each Frame,

1) Initialize $L(q_{ij})$ using (4.5) for all i, j for which $h_{ij} = 1$ with channel LLR's.
2) Update $\left\{L(r_{ji})\right\}$ using (4.19)
3) Update $\left\{L(q_{ji})\right\}$ using (4.24).
4) Update $\{L(V_i)\}$ using,

$$L(V_i) = L(x_i) + \sum_{k \in C_i} L(r_{ki}) \quad i = 0, \ldots, 6. \tag{4.25}$$

5) Proceed to step 2.

After maximum specified iterations, Decoding can be done using,

$$\hat{c}_i = \begin{cases} 1 & L(V_i) < 0 \\ 0 & else \end{cases} \tag{4.26}$$

C. Simulation Results

For frames N=10000. Fig 5 Shows the Comparison of Probability error with channel coding and without channel coding. Since the parity check matrix taken was not much sparse, we are not getting near shannon limit performance. (Good sparse matrix i.e number of entries in H << $m \times n$)

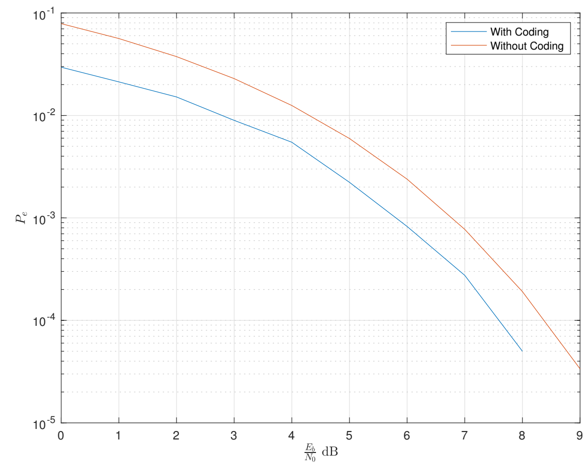

Fig. 5: SNR vs BER curves using LDPC channel coding and no channel coding

5. LDPC Decoding using Min-Sum

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{5.1}$$

received bits is-

$$= \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 \end{bmatrix} \tag{5.2}$$

Row operation-

1) Step-1

$$H = \begin{bmatrix} r_1 & r_2 & r_3 & 0 & r_5 & 0 & 0 \\ 0 & r_2 & r_3 & r_4 & 0 & r_6 & 0 \\ r_1 & r_2 & 0 & r_4 & 0 & 0 & r_7 \\ r_1 & 0 & r_3 & 0 & r_5 & r_6 & r_7 \end{bmatrix} \tag{5.3}$$

2) For each row- Magnitude
   Min1= minimum absoluate value of all nonzero entries in row
   Min2= next higher absolute value
   Set magnitude of all values (except minimum)= Min1
   Set magnitude of minimum value =Min2
3) For each row- Sign
   Parity= product of signs of entries in row New sign of an entry= (old sign)* (Parity)

Column operation-

1) For each Column-
   $sum_j = r_j$ + sum of all entries in column j
   new entry = Sum - (old entry)