IMPLEMENTATION OF DIFFERENT DECODING TECHNIQUES FOR QUASI CYCLIC LOW DENSITY PARITY CHECK CODES (QC-LDPC)

Sonia*, Geeta Arora

* M.Tech student, Deptt. of ECE Geeta Institute of Management and Technology, Kurukshetra University, Haryana, India
Assistant professor, Deptt. of ECE Geeta Institute of Management and Technology, Kurukshetra University, Haryana, India

ABSTRACT

This paper presents a different decoding techniques for Quasi-Cyclic (QC) Low Density Parity Check (LDPC) code. QC-LDPC code is proposed to reduce the complexity of the Low Density Parity Check code while obtaining the similar performance. The encoding and decoding processes of these codes are easy to simplify and implement. The algorithm used for encoding and decoding are based on matrices and implemented on MATLAB for simulation results. In this paper decoding technique QC-LDPC is introduced and compare the performance of LDPC and QC-LDPC. The decoding techniques is conducted by different algorithms such as Bit Flipping (BF), Belief Propagation (BP) Log, Belief Propagation (BP) Probabilistic (Prob), Belief Propagation (BP) Log Simple.

KEYWORDS: Low Density Parity Check Code(LDPC), Quasi-Cyclic Low Density Parity Check code (QC-LDPC), Bit Error Rate, Circulant Sub matrix, Decoding techniques.

INTRODUCTION

Low Density Parity Check code [LDPC]

Low density parity check codes are forward error-correction codes, first proposed in the 1962 PhD thesis of Gallager at MIT [1],[2]. Low density parity check codes are linear block codes using generator matrix G in an encoder and parity check matrix H in a decoder. The parity check matrix has M rows and N columns, where M represents the check nodes and N represents variable nodes [3]. Here the matrix is based on random construction techniques. Information bits depends on check nodes and code word bits are depends on variable nodes. Tanner Graph is the bipartite graph introduced to graphically represents these codes. They also helps to describe decoding algorithm There are various decoding algorithms, Bit flipping (BF) algorithm decoding algorithms for low density parity check (LDPC) and the message passing algorithms are the Belief propagation (BP), Sum product message passing algorithm (SPA) and so on [4]. These algorithms provide a wide range of tradeoffs among decoding speed and error rate performance. LDPC codes have been applied to application in communication and storage system.

Block diagram of LDPC [Figure 1.1]

Quasi-Cyclic Low Density Parity Check code[QC-LDPC]

Quasi-Cyclic Low Density Parity Check codes are codes in which rows or columns in a sub matrix have similar and cyclic connections. The construction of QCLDPC code is done by shifting identity sub-matrices. Numbers of cyclic shifts of columns in the identity sub-matrices are represented in a matrix which gives compact representation of H matrix [5], [6]. When designing a QC-LDPC code the parity check matrix H should have some properties that optimize the behavior of the belief propagation based decoder. The structure of QC-LDPC codes allows them to be decoded using shift registers and decoders architecture require simple address generation mechanisms, less memory and localized accesses [7].
QC-LDPC codes, based on circulant permutation matrices. If the weight w of circulant is equal to 1, then the circulant is called permutation matrix. The generator of the circulant is characterized by the first row on the first column of the same circulant.

QC-LDPC has been proposed to reduce the complexity of the LDPC while obtaining the similar performance [8]. In this paper, we have constructed a QC-LDPC code which is suitable for small and medium block length applications. We have tested this code using as Belief Propagation (BP) Log, Belief Propagation (BP) Probabilistic (Prob), Belief Propagation (BP) Log Simple, Bit Flipping (BF) decoding.

CIRCULANT PERMUTATION MATRICES
The main feature of QC-LDPC codes is that their parity check matrix consists of circulant sub matrices. It will be like an array of sparse circulates. A square matrix is a circulant where each row in the circulant matrix is rotated one element relative to the preceding row in the right direction, i.e. each row is the circular shift of the above row and the 1st row is the circular shift of the bottom row [9].

For such a cyclic matrix, every column is the downstairs circular shift of the left column and 1st column is circular shift of the last column. If a circular matrix the weight of row and column are the same, suppose ω. If ω=1, the circulant is a permutation matrix and it is called circulant permutation matrix. A circulant permutation matrix is known by its 1st row or 1st column, which is also called the generator of the matrix [10].

The regular LDPC codes have the same number of ones in every row and column. The irregular LDPC codes have a different number of ones in columns and rows.

ENCODING OF QC-LDPC
Each circulant sub matrix is a square matrix for which every row is the cyclic shift of the previous row, and the first row is obtained by the cyclic shift of the last row. In this way, every column of each circulant sub matrix is automatically the cyclic shift of the previous column, and the first column is obtained by the cyclic shift of the last column [8], [5]. The H matrix of dimension (m × Lm) for the QC-LDPC can be written as

\[ H = [H_1 \ H_2 \ H_3 \ \cdots \ H_L] \]

Where \( H_i \) is the i-th circulant sub matrix of dimension \( (m \times m) \), \( i = 1, \cdots, L \) for the circulant matrices, the row weight and column weight are the same and fixed.

The parity check matrix H is defined, the generator matrix is obtained. The matrices are created such that they should satisfy the constraint \( GH^T = 0 \). All the bits to be encoded are run through the generator matrix, and therefore, all valid code words obey the property \( CH^T = 0 \) where C is the codeword. The QC generator matrix of rate \( R = (L - 1)/L \) has the following structure.
As one of the requirements is $GH^T = 0$, we can write
\[
G^T H = \begin{bmatrix}
P_2^T & I_m & 0 & 0 & \cdots & 0 \\
P_3^T & 0 & I_m & 0 & \cdots & 0 \\
P_4^T & 0 & 0 & I_m & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\
P_L^T & 0 & 0 & 0 & I_m \\
\end{bmatrix} \begin{bmatrix}
H_1^T \\
H_2^T \\
H_3^T \\
\vdots \\
H_L^T \\
\end{bmatrix} = 0
\]

From the above relation, we can get $P_i = H_i^{-1}H_i$, where $i = 1 \cdots L$. The inverse of a circulant matrices is also a circulant matrix [11].

Therefore, the QC-LDPC of different rates $(L - 1)/L$ can be produced from the above defined generator matrix $G$. By using this construction, the QC notes of generator matrix is preserved. Since the generator matrix is QC, the first row of each circulant sub matrix is stored, and successive rows can be generated by a shift register generator. This greatly simplifies the encoder design.

MODULATION/CHANNEL
The modulation scheme used is Binary Phase Shift Keying (BPSK). BPSK modulator maps the input binary signals, to an analog signal for transmission. Additive White Gaussian Noise (AWGN) channel is used as a transmission medium from transmitter to the receiver.

DECODING OF QC-LDPC
The decoding techniques used in this paper are Sum Product Algorithm (SPA) and Bit Flipping (BF) algorithm. Sum product algorithm also called belief propagation. Bit flipping algorithm is also called iterative decoding.

HARD DECISION DECODING
In this decoding scheme the check nodes finds the bit in error by checking the even/odd parity. The messages from message nodes are transmitted to check nodes, check node checks the parity of the data stream received from message nodes connected to it. If number of 1’s received at check nodes satisfies the required parity,[12],[13] then it sends the
same data back to message node, else it adjust each bit in the received data stream to satisfy the required parity and then transmits the new message back to message nodes.

The bit-flipping decoder can be immediately terminated whenever a valid codeword has been found by checking if all of the parity-check equations are satisfied.

**Bit Flipping Algorithm (BF):**

In the Bit flipping algorithm the MSG passed along the Tanner graph edges and a bit node sends a MSG declaring if it is a one or a zero, and then each check node sends a MSG to each connected bit node finally declaring that what value the bit is based on the information available to the check node.

The bit flipping algorithm is an example of hard decision message passing algorithm for QC-LDPC codes. The bit flipping decoder can be immediately terminated whenever a valid codeword has been found by checking if all of the parity check equations are satisfied [14]. This is true of all MSG passing decoding of QC-LDPC codes and has two important benefits; firstly additional iterations are avoided once a solution has been found, and secondly a failure to converge to a codeword is always detected.

**SOFT DECISION DECODING**

A second type is, called “soft decision decoding” uses a unique measure of reliability for each symbol. Each symbol is given a unique value between 0 and 1 or any range of values. The soft decision decoding algorithms, capable of correcting more errors and therefore being more robust and of higher performance, tend to be very complicated. Hence, practical implementations of soft-decision decoding is slower in comparison with other decoding algorithms and if sufficient hardware is used to increase speed, costly and often impractical.

Soft-decision decoding gives enhanced performance in decoding procedure of QC-LDPC codes which is based on the idea of belief propagate. In soft scheme, the messages are the conditional probability that in the given received vector received bit is 1 or a 0. The sum-product algorithm is a soft decision message passing algorithm. Posterior probabilities for the received bits is the input probabilities and they were known in advance before running the QC-LDPC decoder. The bit probabilities returned by the decoder are called as posterior probabilities.

**Sum Product Message Passing Algorithm (SPA):**

The sum-product algorithm is similar to the bit-flipping algorithm, but the major difference is that the messages representing each decision with probabilities in SPA whereas bit-flipping decoding on the received bits given as input, accepts an initial hard decision and the sum-product algorithm is a soft decision algorithm which accepts the probability of each received bit as input.

For a binary variable X it is easy to find \( P(x = 1) = 1 - P(x = 0) \) and so we only need to store one probability value for \( x \).[15] Log likelihood ratios are used to represent the metrics for a binary variable by a single value.

\[
L(x) = \log \left( \frac{P(X=0)}{P(X=1)} \right)
\]

The belief propagation decoding can be conducted either in the probabilistic[16] or logarithmic domain [1],[2]. The advantages of using logarithmic probabilities in that a product of several messages will be converted to a sum. This will decrease the complexity of the decoding process since a sum is more convenient to implement in hardware. The two decoding algorithms have almost equal bit error rate performances.

**Probabilistic BP decoding algorithm:** A posteriori probability \( f_j^0 \) and \( f_j^1 \) for each bit \( c_j \) for an AWGN channel.

\[
f_j^i = \rho(c_j = 1 | r_j) = \frac{1}{1 + \exp(-2r_j/\sigma^2)}
\]

Logarithmic BP decoding algorithm is an enhanced version of the probabilistic BP algorithm, introducing logarithmic likelihood ratios (LLR) which reduce most multiplications to additions.
SIMULATION PERFORMANCE

We simulate the code error-correcting performance with the assumption that each code is modulated by BPSK and transmitted over AWGN channel.

The simulation parameters used here are shown in table: (Table 6.1)

<table>
<thead>
<tr>
<th>Simulation Tool used</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>10</td>
</tr>
<tr>
<td>Modulation</td>
<td>BPSK</td>
</tr>
<tr>
<td>Channel Model</td>
<td>AWGN</td>
</tr>
<tr>
<td>Number of rows</td>
<td>30</td>
</tr>
<tr>
<td>Number of columns</td>
<td>60</td>
</tr>
<tr>
<td>Decoding Algorithms</td>
<td>BF, BP LOG, BP PROB, BP LOG SIMPLE</td>
</tr>
</tbody>
</table>

COMPARE THE PERFORMANCE OF LDPC & QC-LDPC

Comparison between LDPC and QC-LDPC using Belief Propagation (BP) Log, Belief Propagation (BP), Probabilistic (Prob), Belief Propagation (BP) Log Simple, Bit Flipping (BF) decoding.

(LDPC & QC-LDPC Codes can be decoded by using different decoding algorithms)

QC-LDPC and LDPC code is tested on Bit flipping (BF), Belief propagation (BP) Log, BP Prob, BP log simple decoding algorithms. When we compare the LDPC & QC-LDPC codes using these four decoding techniques than BP LOG, BP PROB or BP LOG SIMPLE algorithms shows reduce BER of QC-LDPC.

Below fig. shows the simulated Bit Error Rate (BER) verses Signal to Noise ratio (SNR). (Table 6.2)

<table>
<thead>
<tr>
<th>Eb/No</th>
<th>BF</th>
<th>BP_LOG_SIMPLE</th>
<th>BP_LOG</th>
<th>BP_PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4667</td>
<td>0.2000</td>
<td>0.6000</td>
<td>0.6000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3000</td>
<td>0.2333</td>
<td>0.2000</td>
<td>0.2000</td>
</tr>
<tr>
<td>1</td>
<td>0.3333</td>
<td>0.1333</td>
<td>0.1333</td>
<td>0.1333</td>
</tr>
<tr>
<td>1.5</td>
<td>0.2000</td>
<td>0.1000</td>
<td>0.0333</td>
<td>0.0333</td>
</tr>
<tr>
<td>2</td>
<td>0.3333</td>
<td>0.0667</td>
<td>0.0667</td>
<td>0.0667</td>
</tr>
</tbody>
</table>

Elapsed time is 0.476958 seconds.
(Figure 6.2)

(Table 6.3)

<table>
<thead>
<tr>
<th>Eb/No</th>
<th>BF</th>
<th>BP_LOG_SIMPLE</th>
<th>BP_LOG</th>
<th>BP_PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4333</td>
<td>0.1667</td>
<td>0.5333</td>
<td>0.5333</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4000</td>
<td>0.1333</td>
<td>0.2333</td>
<td>0.2333</td>
</tr>
<tr>
<td>1</td>
<td>0.2667</td>
<td>0.1667</td>
<td>0.0667</td>
<td>0.0667</td>
</tr>
<tr>
<td>1.5</td>
<td>0.4333</td>
<td>0.0667</td>
<td>0.0333</td>
<td>0.0333</td>
</tr>
<tr>
<td>2</td>
<td>0.3000</td>
<td>0.0667</td>
<td>0.0333</td>
<td>0.0333</td>
</tr>
</tbody>
</table>

Elapsed time is 0.399352 seconds.
In Fig. 6.2 and Fig. 6.3 we have blue line which is showing the Bit flipping, green line is for Belief propagation (BP) Log simple, red line is for BP Log, BP Prob is shown through sky blue line. In these fig. BP Log and BP Prob is shows overlapping. QC-LDPC code is reduce the complexity of the LDPC while obtaining the similar performance.

CONCLUSION
The QC-LDPC decoding may be best understood by the sum product algorithm and bit flipping algorithm showing the QC-LDPC code can be implemented easily. Here QC-LDPC is tested on four decoding algorithms. In this paper QC-LDPC decoding techniques explained in detail, when we compare the LDPC & QC-LDPC codes using four decoding techniques than BP LOG or BP PROB algorithms shows reduce BER of QC-LDPC and overlap with each other. Above Table 6.3 shows elapsed time complexity (sec.) is reduce of the QC-LDPC as compare to LDPC.

REFERENCES


