Quantization methods of the reliability-based iterative decoding algorithm for LDPC codes

Haiqiang Chen1,2,3,a, Lingshan Luo1,b, Xiaoli Huang1,c, Daofeng Li1,d, Qi Liang1,e, Tuanfa Qin1,2,3,f

1School of Computer, Electronics and Information, Guangxi University, Nanning, China
2Guangxi Colleges and Universities Key Laboratory of Multimedia Communications and Information Processing (Guangxi University)
3Guangxi Key Laboratory of Multimedia Communications and Network Technology, Guangxi University (Cultivating Base)
4haiqiang@gxu.edu.cn, bshan092@mail.gxu.cn, cl183104748@qq.com, dldf_0123@163.com,
eqiangli@mail.gxu.cn, ftqin@gxu.edu.cn

Keywords: LDPC Code, Quantization, Reliability-based, Iterative, Decoding algorithm

Abstract. This paper investigates the quantization methods of the reliability-based iterative decoding algorithm for LDPC codes. For the uniform quantization, we present a new design criterion to determine the quantization parameters with the clipping interval/probability. Compared to the original parameters setting, the presented design criterion can achieve higher resolution with the allowable quantization distortions. The presented parameter design criterion can be extended to the non-uniform quantization with a slight modification. Simulation results show that, the decoding algorithm can achieve better BER performances and faster convergence rate with smaller quantization bit when compared to the original parameters setting.

Introduction

Recently, the reliability-based iterative decoding algorithms have attracted considerable attention since they can make efficient trade-offs between decoding performance and computational complexity. In 2009, Huang et al proposed a reliability-based iterative majority-logic decoding (RBI-MLGD) algorithm which performs well when compared to several existing reliability-based algorithms with very low decoding complexity [1]. In [2], we extended Huang’s work and presented a reliability-based min-sum decoding algorithm (RBI-MSD) by introducing the integer reliability measures and factor scaling techniques. Other relevant work can be seen in [3][4][5]. Reliability-based decoding algorithms for non-binary LDPC codes have also been investigated, see [6][7] and references therein.

For the reliability-based algorithms mentioned above, real reliability messages are required to be quantized into integer values, usually using a uniform quantization function. It is shown that the parameters for the uniform quantizer (such as the quantization level, the quantization threshold) may have strong influence on the decoding performance and the convergence speed. However, no detailed discussions are given to determine these parameters. Motivated by this fact, this paper focuses on the quantization method of the reliability-based decoding algorithm for LDPC codes. For the uniform quantization method, we design a clipping criterion to determine the quantization parameters using a specified clipping interval and clipping probability. Then we extend such design criterion to the non-uniform quantization case. Simulation results show that, the decoding algorithm can achieve better BER performances and faster convergence rate with smaller quantization bit when compared to the original parameters setting.
Preliminary

**Basic concepts and notation.** Let $C$ denote an $n$-bit LDPC code, which is defined by a sparse parity-check matrix $H = (h_{i,j})_{n \times m}$. An LDPC code can also be interpreted by a Tanner graph, which is a bipartite graph containing two types of node, the variable node $V_{j,i}$ ($0 \leq j < n$) and the check node $C_i$ ($0 \leq i < m$). A variable node corresponds to a column of $H$ and a check node corresponds to a row of $H$. For any parity-check matrix, we define the following two sets

$$N_j = \{ j: 0 \leq j < n-1, h_{j,n-1} = 1 \}$$ \hspace{1cm} (1)

for each row $i$ of $H$ and

$$M_i = \{ i: 0 \leq i < m-1, h_{n-1,i} = 1 \}$$ \hspace{1cm} (2)

for each column $j$ of $H$.

**System model.** The input binary information is denoted by $u = (u_1, u_2, \ldots, u_{n-1}) \in \mathbb{F}_2^{n-1}$. This input vector is encoded by a binary LDPC encoder, resulting in an $n$-bit codeword vector $c = (c_1, c_2, \ldots, c_m) \in \mathbb{F}_2^m$. The codeword is then modulated into a real vector $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ with $x_j = 2c_j - 1$ (Assume that the simplest BPSK modulation is employed). The modulated signal $\mathbf{x}$ is transmitted over an additive white Gaussian noise (AWGN) channel. Then the receiver takes as the input the vector $y = (y_0, y_1, \ldots, y_{n-1})$ with $y_j = x_j + n_j$, where $n_j \sim \mathcal{N}(0, \sigma^2)$ is a white Gaussian noise sample with the standard deviation $\sigma$. The received signal is required to be quantized into integer value before passing to the decoder. The corresponding quantized signal is denoted by $q = (q_0, q_1, \ldots, q_{n-1})$. The binary LDPC decoder then can be performed with the input $q$ and obtains the output $\hat{u}$, an estimation of the information vector $u$. The system model described above can be seen in Fig. 1.

![Fig. 1 System model](image)

**The uniform quantization method**

**Brief review.** Let $\Delta > 0$ denote the quantization interval and $b > 1$ denote the quantization level. The RBI-MLGD algorithm and the RBI-MSD algorithm employ the uniform quantization function as follows.

$$q_j = \begin{cases} 
(2^b - 1), & y_j / \Delta \leq (2^b - 1) \\
\lfloor y_j / \Delta \rfloor, & (2^b - 1) < y_j / \Delta < 2^b - 1 \\
2^b - 1, & y_j / \Delta \geq (2^b - 1) 
\end{cases}$$ \hspace{1cm} (3)

for $0 \leq j \leq n-1$, where $\lfloor x \rfloor$ denotes the nearest integer to $x$. With the quantization function defined above, the real reliability message $y = (y_0, y_1, \ldots, y_{n-1})$ is quantized into integer $q = (q_0, q_1, \ldots, q_{n-1})$, which can be viewed as the input of the decoder. It is shown from the quantizer above that the received signal is clipped into the interval $[-y_{n}, y_{n}]$, where $y_{n} = (2^b - 1) \Delta$ denotes the maximum value of the clipped signal.
Discussion and design criterion. The quantization method above is easy to be implemented and is commonly used. The quantization parameters $\Delta, b$ and $y_{m}$ are of importance in practice, which may have a strong influence on the decoding performance. To ensure more original received symbols being included in the quantization processing, we need to increase the magnitude $y_{m}$. However, for a fixed quantization bit $b$, a large value of $y_{m}$ means a large quantization interval $\Delta$. This may result in very low resolution and leads to performance degradation. Therefore, there exists an optimal selection among the quantization interval $\Delta$, the quantization bit $b$ and consequently the maximum clipping value $y_{m}$. In this paper, we suggest to determine the quantization parameters by defining the “reliable” region similar to the confidence interval. Intuitively, to ensure the decoding performance, the received signal should be clipped carefully to avoid clipping distortion. For the AWGN channel, the received variable $y_j$ has the normal distribution as follows

$$y_j = X_i(c_j, \sigma^2),$$

where $\phi(c)$ denotes the constellation point of the code bit $c$. Given a SNR value, we then can easily compute the probability that the signal $y_j$ falls into the interval $[-y_m, y_m]$. For convenience, we call such probability the “clipping” probability $p^*$ as the interval associated with such probability the “clipping” region for a certain reliability-based decoding algorithm. For a required quantization bit $b$, we then can describe the design criterion for the quantization parameters as follows. Firstly, set the clipping probability $p^*$ under a certain SNR value; Secondly, compute the maximum clipping parameter $y_m$ to determine the clipping interval $[-y_m, y_m]$. Finally, with the parameter $b$ and $y_m$, compute the quantization interval $\Delta$ accordingly.

For convenience, we can first determine the clipping interval then examine the clipping probability in practice. Table 1 gives an example of the design criterion for a fixed SNR value under 4-bit quantization. It can be seen that, the clipping probability is large enough (say, larger than 0.8) when $y_m$ is larger than 1.5. Besides, we observe a very large clipping probability (very close to 1) for the RBL-MLGD/RBL-MSD algorithms with their own quantization parameter setting. However, such large clipping probability/interval may not necessary in practice. Actually, as is shown by simulations in the following section that, a clipping probability around $p^* = 0.8$ is adequate for performance/complexity tradeoff. The resulting clipping interval $[-y_m, y_m]$ is about $[-1.5, 1.5]$. For comparisons, we also give the design criterion for quantization parameters with such clipping interval (4-bit quantization) in table 2. It is shown that, for 4-bit quantization with $y_m = 1.5$, the resulting clipping probability is around 0.8 and the quantization interval is 0.1, which is quite different form the parameters given in [1][2] for the RBL-MLGD/RBL-MSD algorithm. One of the difference is that the quantization interval has become smaller (from 0.25 to 0.1), which indicates that the quantization resolution becomes higher, especially for the small received signal region. This improvement is meaningful for the iterative decoding algorithm and may achieve performance gain.

| Table 1 Design criterion for quantization parameters with fixed SNR (4-bit quantization) |
|-----------------------------------|---|---|---|---|
| Design criterion     | SNR | $\sigma$ | $p^*$ | $y_m$ | $\Delta$ |
| Presented            | 4.0 | 0.15     | 0.50000 | 1.0   | 0.0667    |
| Presented            | 4.0 | 0.15     | 0.60000 | 1.288 | 0.0753    |
| Presented            | 4.0 | 0.15     | 0.70000 | 1.2678| 0.0845    |
| Presented            | 4.0 | 0.15     | 0.80000 | 1.4327| 0.0955    |
| Presented            | 4.0 | 0.15     | 0.90000 | 1.6593| 0.1106    |
| Presented            | 4.0 | 0.15     | 0.95000 | 1.8447| 0.1230    |
| Presented            | 4.0 | 0.15     | 0.99000 | 2.2001| 0.1467    |
Table 2 Design criterion for quantization parameters with fixed clipping interval (4-bit quantization)

<table>
<thead>
<tr>
<th>SNR</th>
<th>$\sigma$</th>
<th>$\rho$</th>
<th>$\gamma_b$</th>
<th>$\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>0.552</td>
<td>0.819</td>
<td>1.5</td>
<td>0.1000</td>
</tr>
<tr>
<td>3.6</td>
<td>0.539</td>
<td>0.824</td>
<td>1.5</td>
<td>0.1000</td>
</tr>
<tr>
<td>3.8</td>
<td>0.527</td>
<td>0.829</td>
<td>1.5</td>
<td>0.1000</td>
</tr>
<tr>
<td>4.0</td>
<td>0.515</td>
<td>0.834</td>
<td>1.5</td>
<td>0.1000</td>
</tr>
<tr>
<td>4.2</td>
<td>0.503</td>
<td>0.839</td>
<td>1.5</td>
<td>0.1000</td>
</tr>
<tr>
<td>4.4</td>
<td>0.492</td>
<td>0.846</td>
<td>1.5</td>
<td>0.1000</td>
</tr>
</tbody>
</table>

The non-uniform quantization method The quantization methods in RBI-MLGD/RBI-MSD algorithms are both the uniform quantization. For the uniform quantization, the quantization interval always takes the same value, which is not efficient for some applications. For example, for the equal prior probability source, the decision threshold is exactly zero, indicating that errors may occur at this point with very high probability. Therefore, the received signals around zero (the small signals) may require higher quantization resolution. Motivated by this fact, we have presented a non-uniform quantization method in [8]. The design criterion described above can also be extended to the non-uniform case. The only modification is to enlarge the clipping interval from $[-1, +1]$ to $[-\gamma_b, \gamma_b]$. Given a required quantization bit $b$, the non-uniform quantization method can be described as follows.

$$
q_j = \begin{cases} 
\text{sgn}(y_j)(2^b - 1), & |y_j| \geq \gamma_b \\
\text{sgn}(y_j)(2^b - 1 - p), & \rho \gamma_b \leq |y_j| < r \gamma_b \\
\text{sgn}(y_j), & 0 \leq |y_j| < r \gamma_b - 2 
\end{cases},
$$

for $0 \leq j \leq n-1$. $p$ is an integer with $0 \leq p \leq 2^b - 3$, the parameter $r (0 < r < 1)$ is relevant to the quantization resolution and can be determined by simulations.

Algorithm description We take the RBI-MLGD algorithm and the RBI-MSD algorithm as examples to briefly describe the main decoding procedures. At the $k$-th iteration, the decoder mainly performs the following three steps. For the detailed notation, see [1][2].

 Syndrome computation: Let $Z^{(k)} = (z_{1}^{(k)}, z_{2}^{(k)}, ..., z_{n-1}^{(k)})$ be the hard-decision vector. The syndrome vector $S^{(k)} = (s_{1}^{(k)}, s_{2}^{(k)}, ..., s_{n-1}^{(k)})$ can be computed by

$$
s_j^{(k)} = \sum_{i \in \{1, 2, ..., n-1\}} \oplus z_i^{(k)} h_j = \sum_{i \in \{1, 2, ..., n-1\}} \oplus z_j^{(k)},
$$

where $\oplus$ denotes addition modulo 2.

Check node processing: Let $R_j^{(k)}$ be the reliability message with respect to $V_j$ in the $k$-th iteration. Initially, $R_j^{(0)}$ is set to be $q_j$. The extrinsic messages from the check node $C_j$ to the variable node $V_j$ can be calculated as

$$
\sigma_j^{(k)} = \sum_{i \in \{1, 2, ..., n-1\}} \oplus z_i^{(k)} \text{ (for RBI-MLGD)},
$$

$$
U_j^{(k)} = (2\sigma_j^{(k)} - 1) \min \{| R_j^{(k)} |} \text{ (for RBI-MSD)}.
$$

Variable node processing: The total extrinsic messages collected by the variable node $V_j$ from all adjacent check nodes are calculated as
\[ \xi_j^{(k)} = \sum_{i \in M_j} (2\sigma_{ij}^{(k)} - 1) \text{ (for RBI-MLGD)}, \]  
\[ \xi_j^{(k)} = \alpha \sum_{i \in M_j} U_{ij}^{(k)} \text{ (for RBI-MSD)}, \]  
where the scaling factor \( \alpha \) can be optimized by discretized density evolution.

**Updating:** The updating rule at the variable node \( V_j \) can be computed as follows
\[ R_j^{(k+1)} = R_j^{(k)} + \xi_j^{(k)} \text{ (for RBI-MLGD)}, \]  
\[ R_j^{(k+1)} = q_j + \xi_j^{(k)} \text{ (for RBI-MSD)}. \]

With the notation given above, the RBI-MLGD and the RBI-MSD algorithms can be easily described. For details, see [1][2].

**Simulation results**

**Example 1:** Consider the uniform quantization method with a fixed clipping interval \([-1.5, +1.5]\) (the associated clipping probability \( p^2 \) is around 0.8). To show the effect of the parameters design criterion, we take the RBI-MSD decoding algorithm as an example to decode the \((961, 721)\) regular QC-LDPC code. For 4-bit quantization, the resulting quantization interval \( \Delta = 0.1 \); for 6-bit quantization, the resulting quantization interval \( \Delta = 0.0238 \). The maximum iteration number is set to be 30. The BER performance is shown in Fig. 2. It can be seen that, with 4-bit parameters setting, the RBI-MSD algorithm achieves a performance gain about 0.2dB over the original parameters setting around \( BER = 10^{-4} \); with 6-bit parameter setting, the algorithm performs as well as the original parameters setting. Fig. 3 shows the convergence rate of the algorithm. It can be seen that, with 4-bit parameters setting, the RBI-MSD algorithm converges almost as fast as the original parameters setting, especially for high SNR region.

![Fig. 2 The BER performance for the uniform method](image1)

![Fig. 3 The convergence rate for the uniform method](image2)

**Example 2:** Consider the non-uniform quantization method. Similarly, the clipping interval is set to be \([-1.5, +1.5]\). The RBI-MLGD decoding algorithm is employed to decode the \((961, 721)\) regular QC-LDPC code. For 4-bit quantization, \( r \) is set to be 0.82; for 6-bit quantization, \( r \) is set to be 0.97. The maximum iteration number is set to be 30. The BER performance is shown in Fig. 4. It can be seen that, with 4-bit parameters setting, the RBI-MLGD suffers little performance degradation (within 0.1dB) compared to the original 8-bit parameter setting; while the 6-bit parameters setting can outperform the original 8-bit parameter setting. Fig. 5 shows the convergence rate of the algorithm. It can be seen that, both the 4-bit and the 6-bit parameters setting converge faster than the original 8-bit parameters setting.
Conclusions

In this paper, we have investigated of the quantization methods of the reliability-based iterative decoding algorithms for LDPC codes. For the uniform quantization, we have presented a design criterion for the quantization parameters setting, using the clipping interval/probability. Then we extend such design criterion to the non-uniform case. Simulation results show that, the decoding algorithm can achieve better BER performances and faster convergence rate (at different levels) with small quantization bit when compared to the original parameters setting.

Acknowledgements

This work was supported by NSF of China (No. 61102090, 61261023, and 61362010) and NSF of Guangxi (2012GXNSFAA053217, 2011 GXNSFD018024) and Foundation of Guangxi Educational Committee (YB2014008).

References