



Butterfly-Flow-Graph Based MAP Decoding Algorithm for Channel Quality Information in 3GPP-LTE

Qi Li^(✉), He Wang, Yunchuan Yang, Bin Yu, and Chengjun Sun

Samsung Research Institute China - Beijing (SRC-B),
Sun Palace Building, Chaoyang District, Beijing, China
{q1016.li, h0809.wang, yc0301.yang, bin82.yu, chengjun.sun}@samsung.com

Abstract. Channel quality information (CQI) is an essential element of uplink control signaling in Long Term Evolution (LTE) system. According to 3GPP standard, a linear block code based on Reed-Muller (RM) code has been employed for the CQI transmission for error control. In this paper, a low complexity *maximum a posteriori probability* (MAP) decoding algorithm for CQI decoding is described, which is performed by re-ordering the likelihood values of the received signal and all mapped codewords, and then calculating the probability of 0 and 1 of every transmitted information bit based on a butterfly-flow-graph (BFG). Compared to the standard MAP decoding algorithm, the proposed algorithm can reduce the addition calculation by 35.71% to 72.82% when the number of CQI bit is changing from 4 to 11, and the bit error rate (BER) performance is without degradation.

Keywords: 3GPP-LTE · CQI · MAP decoding

1 Introduction

In Long Term Evolution (LTE) system, channel quality information (CQI) is very crucial in evolved Node B (eNodeB) to decide the corresponding modulation and coding scheme [1]. According to the 3GPP standard [2], a $(32, k)$ linear block code based on Reed-Muller (RM) code is used for channel coding of CQI. Through fast and efficient decoding algorithm, receiver can retrieve the CQI report with low bit error rate (BER) and low latency [3].

In [4], a decoding algorithm with fast-Hadamard-transform (FHT) for CQI is proposed, which utilizes *maximum likelihood* (ML) algorithm based on the correlation between the Hadamard matrix and RM code. Rather than ML decoding algorithm, which can minimize the block error rate (BLER), the *maximum a posteriori probability* (MAP) decoding algorithm is one kind of decoding method which can not only minimize the BER, but also provide the corresponding soft information at the output of decoder [5–8]. However, due to the realization complexity, MAP decoding algorithm is not widely used for block codes [10, 11].

In this paper, we propose to use a butterfly-flow-graph (BFG) scheme for CQI decoding. The BFG scheme is derived from the fast-fourier-transform (FFT), which can greatly reduce the calculation complexity. Based on the BFG scheme, the proposed algorithm can reduce the addition calculation by 35.71% to 72.82% when the number of CQI bit is changing from 4 to 11, and the BER is without degradation compared with the standard MAP algorithm.

The rest of this paper is organized as follows. In Sect. 2, the introduction of linear block code and CQI encoder is described. In Sect. 3, the BFG based MAP decoding algorithm for CQI is proposed. Simulation result and complexity analysis are presented in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Linear Block Code and CQI Encoder

2.1 Linear Block Code

Linear block code is a subclass of block code. For an (n, k) binary linear block code, the generator matrix \mathbf{G} is a $k \times n$ matrix with the elements from $GF(2)$, and the rows of which are a basis of the code. Denote the information sequence as $\mathbf{u} = [u_0 u_1 \dots u_{k-1}]$, the corresponding code can be obtained by

$$\mathbf{c} = \mathbf{u} \cdot \mathbf{G}. \quad (1)$$

where $\mathbf{c} = [c_0 c_1 \dots c_{n-1}]$, and the calculation is performed based on modulo-2 operation [6].

2.2 The Encoder of CQI

As mentioned in Sect. 2.1, let's assume the bit number of CQI is k , and then the CQI is encoded using a $(32, k)$ linear block code, where $k \leq 11$. The codewords of the $(32, k)$ block code are a linear combination of 11 basis sequences as defined in Table 1. Based on the 11 basis sequences, we can construct the generator matrix by $\mathbf{G} = [\mathbf{G}_0^T, \mathbf{G}_1^T, \dots, \mathbf{G}_{10}^T]^T$, where T is the transpose operation.

It can be seen that \mathbf{G}_0 is an all-one sequence, and \mathbf{G}_1 to \mathbf{G}_5 are the rows of a matrix with all possible 5-tuples as columns, which are also the interleaved first order RM generating sequences. \mathbf{G}_6 to \mathbf{G}_{10} are five mask sequences. According to Eq. (1), the encoded CQI block is denoted by $\mathbf{c} = [c_0 c_1 \dots c_{n-1}]$, where $n = 32$ and

$$c_j = \sum_{i=0}^{k-1} (u_i \cdot G_{i,j}) \quad \text{mod} 2. \quad (2)$$

where $j = 0, 1, \dots, 31$ and $G_{i,j}$ is the j -th element of sequence \mathbf{G}_i .

In additive white Gaussian noise (AWGN) channel, we can obtain

$$\begin{aligned}
 L_i &= \ln \frac{\sum_{\mathbf{c} \in \mathcal{C}\{u_i=0\}} \exp\{-\frac{1}{2\sigma^2} \|\mathbf{r} - \mathbf{s}(\mathbf{c})\|^2\}}{\sum_{\mathbf{c} \in \mathcal{C}\{u_i=1\}} \exp\{-\frac{1}{2\sigma^2} \|\mathbf{r} - \mathbf{s}(\mathbf{c})\|^2\}} \\
 &= \ln \frac{\sum_{\mathbf{c} \in \mathcal{C}\{u_i=0\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\}}{\sum_{\mathbf{c} \in \mathcal{C}\{u_i=1\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\}}
 \end{aligned} \tag{7}$$

where σ^2 is the noise variance and $\langle \cdot, \cdot \rangle$ denotes the inner product.

As we know, for the $(32, k)$ block code, the number of the codewords with the i -th information bit being 1 (or 0) is 2^{k-1} . The addition calculation in Eq. (7) will up to be $k(2^k - 2)$. Based on FFT, we can use a butterfly flow graph to reduce the calculation complexity [9]. The procedures are given in the following:

Step-1: For the $(32, k)$ block code, we denote all possible information sequence as $\mathbf{u}_0 = [00 \dots 0]$, $\mathbf{u}_1 = [00 \dots 1]$, \dots , $\mathbf{u}_{2^k-1} = [11 \dots 1]$, and the corresponding codewords are denoted as $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{2^k-1}$. After ± 1 mapping, the mapped symbol sets are expressed as $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{2^k-1}$.

Step-2: Define

$$\lambda_i = \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}_i \rangle\} \tag{8}$$

where $i = 0, 1, \dots, 2^k - 1$, and λ_i represents the likelihood value of the received sequence \mathbf{r} and the mapped i -th codeword \mathbf{s}_i .

Step-3: Re-order the generated λ_i : let $\mathbf{z}_i = [\lambda_i, \lambda_{2^k-1-i}]^T, i = 0, 1, \dots, 2^{k-1} - 1$, and introduce a $2^k \times 1$ sized vector $\mathbf{z} = [\mathbf{z}_0^T, \mathbf{z}_1^T, \dots, \mathbf{z}_{2^k-1}^T]^T$.

Step-4: Proceed on the BFG-based transform, we can obtain a $2k \times 1$ sized vector \mathbf{y} as

$$\mathbf{y} = [y_0, y_1, \dots, y_{2k-1}] = BFG(\mathbf{z}) \tag{9}$$

For simplicity, we take $k = 4$ for example, where $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \dots, \lambda_{15}]$, $\mathbf{z} = [\mathbf{z}_0^T, \mathbf{z}_1^T, \dots, \mathbf{z}_7^T]^T$, and \mathbf{y} is a 8×1 sized vector. The BFG-based transform can be shown in Fig. 1, from which it is obtained that:

$$\begin{aligned}
 y_0 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_0=0\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_1 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_0=1\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_2 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_3=0\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_3 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_3=1\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_4 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_2=0\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_5 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_2=1\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_6 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_1=0\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\} \\
 y_7 &= \sum_{\mathbf{c} \in \mathcal{C}\{u_1=1\}} \exp\{\frac{1}{\sigma^2} \langle \mathbf{r}, \mathbf{s}(\mathbf{c}) \rangle\}
 \end{aligned} \tag{10}$$

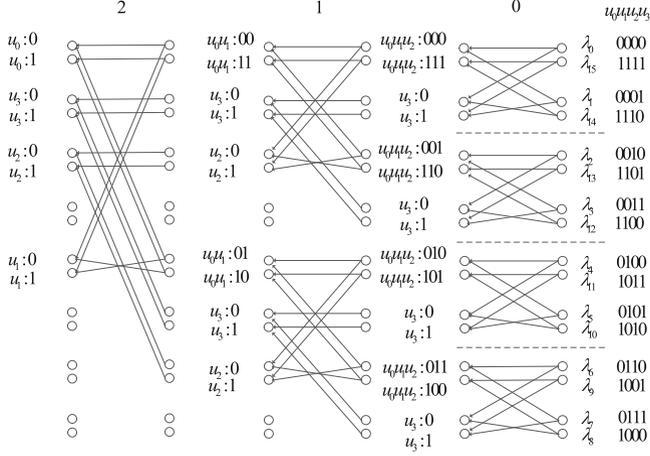


Fig. 1. The BFG-based transform of (32, 4) CQI decoder.

The BFG-based MAP decoding algorithm can be summarized as follows.

- (a) Define $\mathbf{Q}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{Q}'_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, we can obtain \mathbf{Q}_l by $\mathbf{Q}_l = \begin{bmatrix} \mathbf{Q}_{l-1} & \mathbf{Q}_{l-1} \\ \mathbf{Q}'_{l-1} & \mathbf{Q}'_{l-1} \end{bmatrix}$ and $\mathbf{Q}'_l = \begin{bmatrix} \mathbf{Q}'_{l-1} & \mathbf{Q}'_{l-1} \\ \mathbf{Q}_{l-1} & \mathbf{Q}_{l-1} \end{bmatrix}$ recursively, $l = 1, 2, \dots, k-1$.
- (b) In order to obtain the $2k \times 1$ sized vector \mathbf{y} , denote $\mathbf{x}_i = [y_{2i}, y_{2i+1}]^T$ firstly, $i = 0, 1, \dots, k-1$. We can get

$$\mathbf{x}_i = \begin{cases} \mathbf{Q}_{k-1,0} \cdot \mathbf{z}, & i = 0 \\ \mathbf{Q}_{k-1,2^i} \cdot \mathbf{z}, & i = 1, 2, \dots, k-1 \end{cases} \quad (11)$$

where $\mathbf{Q}_{k-1,i}$ denotes the i -th and $i+1$ -th row of \mathbf{Q}_{k-1} .

Based on $\mathbf{x}_i = [y_{2i}, y_{2i+1}]^T$, we can construct the $2k \times 1$ sized vector \mathbf{y} correspondingly, where $\mathbf{y} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_{k-1}^T]^T$. Besides, with \mathbf{Q}_0 and \mathbf{Q}'_0 , the BFG-based transform can be represented as shown in Fig. 2.

Step-5: From Eq. (7) and the definition of vector \mathbf{y} , we can calculate the decoded LLR by

$$L_i = \begin{cases} \ln \frac{y_0}{y_1}, & i = 0 \\ \ln \frac{y_{2^{(k-i)}}}{y_{2^{(k-i)}+1}}, & i = 1, 2, \dots, k-1 \end{cases} \quad (12)$$

Step-6: Based on the decoded LLR, we can obtain the corresponding hard decision by

$$\hat{u}_i = \begin{cases} 0, & L_i > 0 \\ 1, & L_i < 0 \end{cases} \quad (13)$$

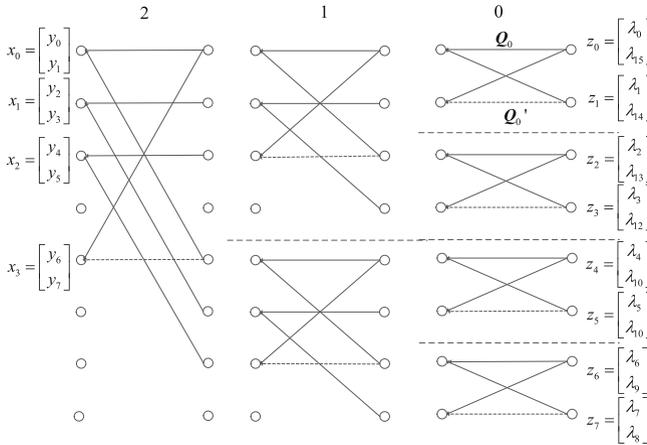


Fig. 2. Another representation of the BFG-based transform

4 Numerical Results

4.1 Performance Simulation

The simulation results are given in this section, where the simulation is carried out on AWGN channel with different signal-to-noise ratio (SNR), and the information bit number $k = 11$. The BER performance of standard MAP decoding algorithm and BFG-based MAP decoding algorithm are provided in Fig. 3. We can observe that the proposed BFG-based MAP decoding algorithm can obtain exactly the same BER performance as the standard MAP decoding algorithm.

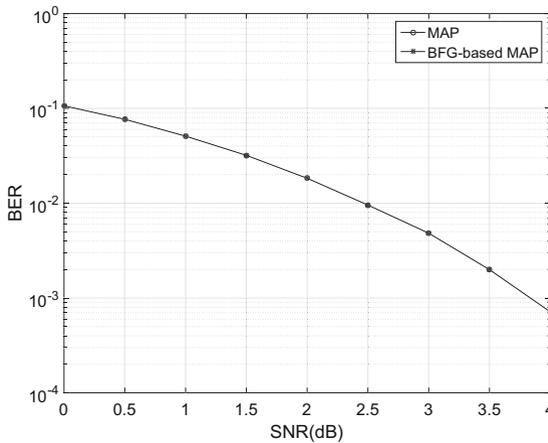


Fig. 3. The BER performance of the standard MAP decoding and BFG-based MAP decoding

4.2 Complexity Analysis

The addition calculation complexity of the standard MAP decoding algorithm and BFG-based MAP decoding algorithm with different information bit number k are shown in Fig. 4, and the relative complexity reduction is shown in Fig. 5. From the figure, it is known that, with the proposed BFG-based MAP decoding algorithm, the addition calculation complexity can be reduced by 35.71% when $k = 4$ and 72.82% when $k = 11$ accordingly. As a result, with the proposed BFG-based MAP decoding algorithm, the addition calculation complexity can be efficiently reduced.

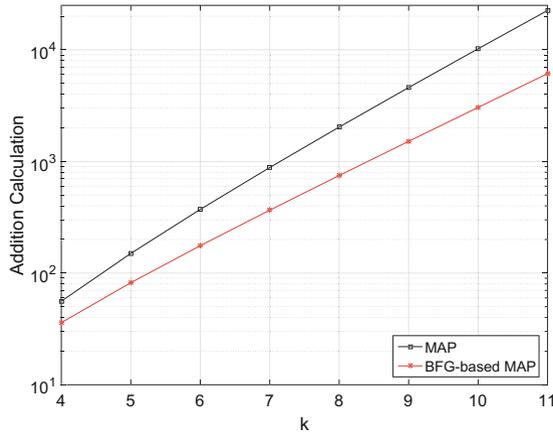


Fig. 4. The addition calculation complexity of the standard MAP decoding and BFG-based MAP decoding

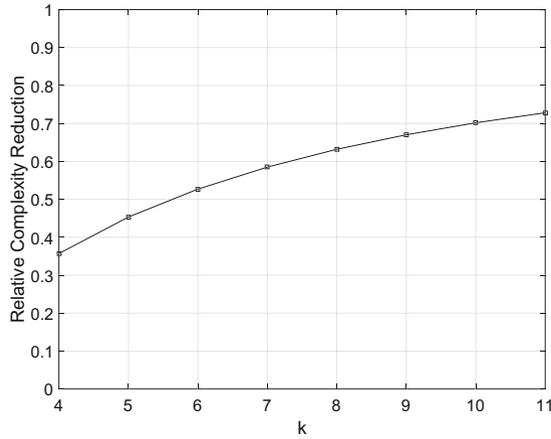


Fig. 5. The relative complexity reduction of BFG-based MAP decoding compared to the standard MAP decoding

5 Conclusion

In this paper, a BFG-based MAP decoding algorithm for CQI in 3GPP-LTE is described. From the simulation result, it is seen that compared to the standard MAP decoding algorithm, the propose algorithm can greatly reduce the complexity while the BER performance is without degradation. As a result, with the proposed algorithm, the receiver can retrieve the CQI report fast and efficiently.

References

1. Ramli, H.A.M., Sukor, M.A.: Performance analysis on automated and average channel quality information (CQI) reporting algorithm in LTE-A. In: 2016 International Conference on Computer and Communication Engineering (2016)
2. 3GPP: TS36.212, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 12) (2015)
3. Moreira, J.C., Farrell, P.G.: Essentials of Error-control Coding. Wiley, New York (2006)
4. Li, M., Han, D., Ma, S.: Decoding algorithm with fast Hadamard transform for channel quality indication (CQI) in 3GPP-LTE, In: 2010 International Conference on Computer Application and System Modeling (2010)
5. Wolf, J.K.: Efficient maximum likelihood decoding of linear block codes using a trellis. *IEEE Trans. Inf. Theory* **24**, 76–80 (1978)
6. Lin, S., Costello, D.J.: Error Control Coding. Pearson Education Inc., London (2004)
7. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)
8. Ashikhmin, A., Litsyn, S.: Simple MAP decoding of first-order Reed-Muller and Hamming codes. *IEEE Trans. Inf. Theory* **50**(8), 1812–1818 (2004)
9. Li, Q., Yin, L.G., Lu, J.H.: Low rate coding and decoding scheme based on LDPC-BCH graph. *J. Tsinghua Univ. (Sci. Technol.)* **53**(11), 1515–1520 (2013)
10. Feng, W., Wang, Y., Ge, N., Lu, J., Zhang, J.: Virtual MIMO in multi-cell distributed antenna systems: coordinated transmissions with large-scale CSIT. *IEEE J. Select. Areas Commun.* **31**(10), 2067–2081 (2013)
11. Feng, W., Wang, Y., Lin, D., Ge, N., Lu, J., Li, S.: When mmWave communications meet network densification: a scalable interference coordination perspective. *IEEE J. Select. Areas Commun.* **35**(7), 1459–1471 (2017)