ISIT'2013 1569717313

Low-Complexity Encoding of Binary Quasi-Cyclic Codes Based on Galois Fourier Transform

Li Tang[1], Qin Huang[1], Zulin Wang[1] and Zixiang Xiong[2]
[1]School of Electronic and Information Engineering, Beihang University, Beijing, China, 100191
[2]Dept of ECE, Texas A&M University, College Station, TX, USA, 77843
(email:neathe@163.com; qhuang.smash@gmail.com; wzulin_201@163.com; zx@ece.tamu.edu)

***Abstract*—This paper presents a novel low-complexity encoding algorithm for binary quasi-cyclic (QC) codes based on matrix transformation. First, a message vector is encoded into a transformed codeword in the transform domain. Then, the transmitted codeword is obtained from the transformed codeword by the inverse Galois Fourier transform. Moreover, a simple and fast mapping is devised to post-process the transformed codeword such that the transmitted codeword is binary as well. The complexity of our proposed encoding algorithm is less than $ek(n-k)\log_2 e+ne(\log_2^2 e+\log_2 e)+\frac{n}{2}e\log_2^3 e$ bit operations for binary codes. This complexity is much lower than its traditional complexity $2e^2(n-k)k$. In the examples of encoding the binary $(4095, 2016)$ and $(15500, 10850)$ QC codes, the complexities are 12.09% and 9.49% of those of traditional encoding, respectively.**

I. Introduction

Quasi-cyclic (QC) codes [1] are an important class of linear error-correcting codes in both coding theory and their applications. These codes can asymptotically approach the Varshamov-Gilbert bound [2]. Moreover, their partial cyclic structure simplifies their encoding and decoding implementations by using simple shift registers and logic circuits [3]. In recent years, research on QC codes has focused on one of their subclasses, known as QC low-density parity-check (LDPC) codes [4]- [11], which have been shown to perform as well as other types of LDPC codes in most applications. QC-LDPC codes have advantages over other types of LDPC codes in hardware implementation of encoding [12] and decoding [5], [11]. Thus, most LDPC codes adopted as standard codes for various next-generation communication and storage systems are QC.

QC codes generally are encoded by multiplying a message vector $\mathbf{m}$ of length ek with an $ek \times en$ generator matrix $\mathbf{G}$ consisting of $e \times e$ circulants, where $e \in \mathbb{N}$. $\mathbf{G}$ usually is systematic [3] with $\mathbf{G} = [\mathbf{I} \vdots \mathbf{P}]$ and $\mathbf{I}$ being the identity matrix – in the rest of the paper, $\mathbf{G}$ is systematic if not stated otherwise. There are two issues with the implementation of QC code systems. First, the generator matrix $\mathbf{G}$ is usually not sparse, so it requires a large number, i.e., $e(n-k)k$, of memory units to store $\mathbf{P}$. Second, although encoding of QC codes can be partially parallelized so that the computation units are reduced by a factor of e, the total number of symbol operations is still $2e^2(n-k)k$, which is the same as that for general linear codes.

This work was supported by National Natural Science Foundation of China (61071070 and 61201156). The corresponding author is Qin Huang.

In this paper, we propose to encode binary QC codes in the Fourier domain rather than direct multiplications in the symbol domain. We are motivated by the fact that the $ek \times en$ transformed generator matrix is an $e \times e$ diagonal array of $k \times n$ matrices. Consequently, *encoding in the transform domain (ETD)* of the binary QC code is achieved by e times encoding of message vectors of length k with $k \times n$ generator matrices (rather than encoding of an ek message vector with an $ek \times en$ generator matrix). We call the resulting vector after ETD the *transformed codeword*. For any binary QC code, its transformed generator matrix satisfies the conjugacy constraint [13], [14], but the transmitted codeword from ETD usually is not binary. Thus, it costs several bits to transmit a code symbol, resulting in lower code rate. To make the transmitted codeword binary, we devise a simple and fast mapping to post-process the transformed codeword so that the new transformed codeword still satisfies the conjugacy constraint. The post-processing step consisting of a mapping with bases of subfields is the key of this algorithm. Then, the final transmitted QC codeword is obtained by permutations and n times inverse Galois transforms from the post-process transformed codewords. Furthermore, if we take advantages of the conjugacy constraint on the transformed generator matrix, the computational complexity of ETD can be reduced to less than $ek(n-k)\log_2 e + ne(\log_2^2 e + \log_2 e) + \frac{n}{2}e\log_2^3 e$. Its memory consumption is $e(n-k)k$, the same as the traditional one. We show that the computation complexities of ETD of the binary (4095, 2016) and (15500, 10850) QC codes are respectively 12.09% and 9.49% of those of traditional encoding.

The rest of this paper is organized as follows. Section II introduces binary QC codes and the matrix transformation. In Section III, we present ETD algorithm of binary QC codes and its simplification due to the conjugacy constraint. The key step involves construction of a post-processing mapping using subfield bases is derived. Section IV concludes the paper.

II. Introduction to quasi-cyclic codes and its Fourier transform

In this section, we recall the characterization of QC codes and their Fourier transform which have presented in [13], [15].

Let $\mathbf{W} = [w_{ij}]$, $0 \leq i, j < e$, be an $e \times e$ circulant matrix over GF(q), i.e., every row is a *cyclic-shift* (one place to right) of the row above it, including end-around. Then, we write $\mathbf{W} = circ(\mathbf{w})$, where $\mathbf{w}$ is its top row, called the *generator* of $\mathbf{W}$ [16]. A binary (ne, ke) QC code $\mathcal{C}$ has generator matrix $\mathbf{G}$ which is given by the following $k \times n$ array of $e \times e$ submatrices over GF(2):

$$\mathbf{G} = \begin{bmatrix} \mathbf{W_{0,0}} & \mathbf{W_{0,1}} & \cdots & \mathbf{W_{0,n-1}} \\ \mathbf{W_{1,0}} & \mathbf{W_{1,1}} & \cdots & \mathbf{W_{1,n-1}} \\ \cdots & \cdots & \ddots & \cdots \\ \mathbf{W_{k-1,0}} & \mathbf{W_{k-1,1}} & \cdots & \mathbf{W_{k-1,n-1}} \end{bmatrix}. \quad (1)$$

Here we consider only the case that $e = 2^r - 1$. The other cases are similar.

Let α be an element over GF(2^r) with order e. Let $\mathbf{w} = (w_0, w_1, \cdots, w_{e-1})$ be a binary vector. The *Fourier transform* of $\mathbf{w}$, denoted by $\mathcal{F}[\mathbf{w}]$, is given by the $\mathbf{d} = (d_0, d_1, \cdots, d_{e-1})$ whose t-th component, d_t, for $0 \leq t < e$, is given by $d_t = \sum_{l=0}^{e-1} w_l \alpha^{-tl}$. It is clear that $\mathbf{d}$ is a e-tuple vector over GF(2^r). Similarly, $\mathbf{w}$ can be reconstructed by the *inverse Fourier transform* , denoted by $\mathcal{F}^{-1}[\mathbf{d}]$, which is given by following equation: $w_l = \sum_{t=0}^{e-1} d_t \alpha^{tl}$, for $0 \leq l < e$.

Define the following two $e \times e$ matrices over GF(2^r): $\mathbf{V} = [\alpha^{-ij}]$ and $\mathbf{V}^{-1} = [\alpha^{ij}]$, $0 \leq i, j < e$. Since both of $\mathbf{V}$ and $\mathbf{V}^{-1}$ are *Vandermonde* matrices, they are non-singular. Moreover, $\mathbf{V}^{-1}$ is the inverse of $\mathbf{V}$ and vice versa. Let $\mathbf{w} = circ(w_0, w_1, \cdots, w_{e-1})$ be an $e \times e$ circulant matrix. Then the Fourier transform of $\mathbf{W}$, denoted by $\mathbf{W}^{\mathcal{F}}$, which is a $e \times e$ diagonal matrix over GF(2^r), can be obtained as follows,

$$\mathbf{W}^{\mathcal{F}} = \mathbf{V}^{-1}\mathbf{W}\mathbf{V} = diag(d_0, d_1, ..., d_{e-1}), \quad (2)$$

where the diagonal vector $(d_0, d_1, ..., d_{e-1})$ is the Fourier transform of the generator $\mathbf{w}$ of $\mathbf{W}$. Recall that there is a necessary and sufficient condition to ensure $\mathbf{w}$ be a binary matrix that if and only if

$$d_{(2t)_e} = d_t^2, \quad (3)$$

for $0 \leq t < e$, where $(2t)_e$ denotes the nonnegative integer less than e and is congruent to $2t$ modulo e. The condition is called the *conjugacy constraint*.

Consider a generator matrix of (ne, ke) QC code as $\mathbf{G} = [\mathbf{W}_{i,j}]$, for $0 \leq i < k$, $0 \leq j < n$, which is an $k \times n$ array of $e \times e$ circulants $\mathbf{W}_{i,j}$ over GF(2^r), where $\mathbf{w}_{i,j}$ is the generator of the circulant $\mathbf{W}_{i,j}$. Define two matrices $\Omega(n)$ and $\Omega^{-1}(k)$, which are $n \times n$ diagonal array of $\mathbf{V}$ and $k \times k$ diagonal array of $\mathbf{V}^{-1}$, respectively, as follows,

$$\Omega(n) = diag(\underbrace{\mathbf{V}, ..., \mathbf{V}}_{n}), \quad (4)$$

$$\Omega^{-1}(k) = diag(\underbrace{\mathbf{V}^{-1}, ..., \mathbf{V}^{-1}}_{k}). \quad (5)$$

Then we can obtain the Fourier transform of $\mathbf{G}$, $\mathbf{G}^{\mathcal{F}}$, by the following equation:

$$\mathbf{G}^{\mathcal{F}} = \Omega^{-1}(k)\mathbf{G}\Omega(n) = [\mathbf{W}_{i,j}^{\mathcal{F}}], \quad (6)$$

where $\mathbf{W}_{i,j}^{\mathcal{F}} = \mathbf{V}^{-1}\mathbf{W}_{i,j}\mathbf{V}$ is an $e \times e$ diagonal matrix with diagonal vector $(d_{i,j,0}, d_{i,j,1}, ..., d_{i,j,e-1})$, which is the Fourier transform of the generator $\mathbf{w}_{i,j}$ of $\mathbf{W}_{i,j}$.

To simplify of $\mathbf{G}^{\mathcal{F}}$, we can permute it to an $e \times e$ diagonal array of $k \times n$ matrices, which is denoted by $\mathbf{G}^{\mathcal{F},\pi}$. Define the following index sequences: for $0 \leq i, j < e$, $\pi_{row,i} = [i, e + i, ..., (k-1)e + i]$ and $\pi_{col,j} = [j, e + j, ..., (n-1)e + j]$. Let $\pi_{row} = [\pi_{row,0}, \pi_{row,1}, ..., \pi_{row,e-1}]$ and $\pi_{col} = [\pi_{col,0}, \pi_{col,1}, ..., \pi_{col,e-1}]$. Then π_{row} gives a permutation of the indices of the rows of $\mathbf{G}^{\mathcal{F}}$ while π_{col} represents a permutation of the columns of $\mathbf{G}^{\mathcal{F}}$. Their reverse permutations are denoted by π_{row}^{-1} and π_{col}^{-1}, respectively. We define the permutation π of a matrix that performs both row and column permutations. Its reverse permutation is denoted by π^{-1}. By the permutation π, $\mathbf{G}^{\mathcal{F}}$ results in $\mathbf{G}^{\mathcal{F},\pi}$ as follows,

$$\mathbf{G}^{\mathcal{F},\pi} = diag(\mathbf{D}_0, \mathbf{D}_1, ..., \mathbf{D}_{e-1}), \quad (7)$$

where $\mathbf{D}_i$, for $0 \leq i < e$, is a $k \times n$ matrix over GF(2^r). The transformation from $\mathbf{G}$ to $\mathbf{G}^{\mathcal{F},\pi}$ through $\mathbf{G}^{\mathcal{F}}$ is reversible. The reverse process is called the inverse matrix transformation, denoted by $\{\mathcal{F}^{-1}, \pi^{-1}\}$.

If the array $\mathbf{G}$ of circulants and zero matrices (ZM) is over GF(2) $\subseteq$ GF(2^r), the matrices on the main diagonal of the array $\mathbf{G}^{\mathcal{F},\pi}$ satisfy the conjugacy constraint,

$$\mathbf{D}_{(2t)_e} = \mathbf{D}_t^{\circ 2}, \quad (8)$$

where $\mathbf{D}_t^{\circ 2} = [d_{i,j,t}^{\circ 2}]$, for $0 \leq i < k$, $0 \leq j < n$, i.e., the entry at location (i, j) of $\mathbf{D}_{(2t)_e}$ is the square of the entry at location (i, j) of $\mathbf{D}_t$. We call the matrix $\mathbf{D}_{(2t)_e}$ a *conjugate matrix* of $\mathbf{D}_t$. Following the definition of conjugate matrix, we can group all the matrices on the main diagonal $\mathbf{D}_i$, $i = 0, 1, \ldots, e-1$, into conjugacy classes. Let λ be the number of distinct conjugacy classes and $\Psi_0, \Psi_1, \ldots, \Psi_{\lambda-1}$ represent these classes, with

$$\begin{aligned} \Psi_i &= \{\mathbf{D}_{t_i}, \mathbf{D}_{(2t_i)_e}, ..., \mathbf{D}_{(2^{\eta_i - 1}t_i)_e}\}, \\ &= \{\mathbf{D}_{t_i}, \mathbf{D}_{t_i}^{\circ 2}, ..., \mathbf{D}_{t_i}^{\circ 2^{\eta_i - 1}}\}, \end{aligned} \quad (9)$$

where η_i is the number of matrices in the conjugacy class Ψ_i, i.e., η_i is the smallest nonnegative integer such that $(2^{\eta_i} t_i)_e = t_i$, and t_i is the smallest number in the subscripts of the conjugate matrices in Ψ_i. The member matrix $\mathbf{D}_{t_i}$ is called the representative of the conjugacy class Ψ_i. In fact, $C_{t_i} = \{t_i, (2t_i)_e, ..., (2^{\eta_i - 1}t_i)_e\}$ is a cyclotomic coset modulo 2, t_i is its coset representative, and η_i divides r [14].

In the rest of this paper, we consider only the case $e = 2^r - 1$. The other cases that e is odd are similar. Moreover, if the circulant size e is even, it is required to decompose these circulants into smaller odd circulants of size e' [17], where $e' | e$ and e' is a factor of 2^r.

III. Encoding of binary QC codes in the transformed domain

In this section, we present the low-complexity ETD algorithm of binary QC codes and study its computational complexity and memory consumption.

Define the permutation π of a vector that performs only the column permutations. Its inverse permutation is denoted by π^{-1}. Consider an (en, ek) QC code $\mathcal{C}$ over GF(2) whose code rate is k/n defined by the $ek \times en$ generator matrix $\mathbf{G}$ in (1), which consists of circulants of size e. Suppose that α is a primitive element in GF(2^r), i.e., $e = q - 1 = 2^r - 1$. By the matrix transformation (7), the generator matrix results in the transformed generator matrix $\mathbf{G}^{\mathcal{F},\pi}$ over GF(2^r). Hence, it can be employed to encode a message vector $\mathbf{m}$ of ek bits over GF(2) into a transformed codeword $\mathbf{c}^{\mathcal{F},\pi}$ of en bits over GF(2^r) as following equation,

$$\mathbf{c}^{\mathcal{F}} = \{\mathbf{m} \cdot \mathbf{G}^{\mathcal{F},\pi}\}^{\pi^{-1}}. \tag{10}$$

Define the vector $\mathbf{c} \triangleq \{\mathbf{c}^{\mathcal{F},\pi}\}^{\pi^{-1}} \cdot \Omega^{-1}(n) = \mathbf{c}^{\mathcal{F}} \cdot \Omega^{-1}(n)$, where $\mathbf{c}^{\mathcal{F},\pi} = \mathbf{m} \cdot \mathbf{G}^{\mathcal{F},\pi}$, be named as *transmitted codeword*. Next we show that $\mathbf{c}$ is a codeword of the QC code $\mathcal{C}$. Since

$$\mathbf{G} \cdot \mathbf{H}^{\mathsf{T}} = \Omega(k) \cdot \Omega^{-1}(k) \cdot \mathbf{G} \cdot \Omega(n) \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} = 0,$$

where $\mathbf{H}$ is the parity-check matrix of $\mathcal{C}$, then

$$\Omega(k) \cdot \mathbf{G}^{\mathcal{F}} \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} = 0. \tag{11}$$

Because $\Omega(k)$ is a nonsingular matrix,

$$\mathbf{G}^{\mathcal{F}} \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} = 0. \tag{12}$$

Furthermore, permuting the rows of $\mathbf{G}^{\mathcal{F}}$ by π_{row} results in $\mathbf{G}^{\mathcal{F},\pi_{row}}$, then

$$\mathbf{G}^{\mathcal{F},\pi_{row}} \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} = 0. \tag{13}$$

The vector $\mathbf{c}$ is a codeword of the QC code $\mathcal{C}$ from (10) and (13), because

$$\begin{aligned} \mathbf{c} \cdot \mathbf{H}^{\mathsf{T}} &= \{\mathbf{c}^{\mathcal{F},\pi}\}^{\pi^{-1}} \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} \\ &= \{\mathbf{m} \cdot \mathbf{G}^{\mathcal{F},\pi}\}^{\pi^{-1}} \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} \\ &= \mathbf{m} \cdot \mathbf{G}^{\mathcal{F},\pi_{row}} \cdot \Omega^{-1}(n) \cdot \mathbf{H}^{\mathsf{T}} \\ &= \mathbf{0}. \end{aligned} \tag{14}$$

If the QC code is binary, the message vector $\mathbf{m}$ in (10) is binary, i.e., one bit per symbol. However, there is no guarantee that the transmitted codeword $\mathbf{c}$ from ETD is binary. The matrices $\mathbf{D}_t$'s on the main diagonal of $\mathbf{G}^{\mathcal{F},\pi}$ satisfy the conjugacy constraint (8), but the binary message vector $\mathbf{m}$ may not. Thus, the transformed codeword $\mathbf{c}^{\mathcal{F}}$, the product of $\mathbf{m}$ and $\mathbf{G}^{\mathcal{F},\pi}$(10), may not satisfy the conjugacy constraint. In other words, the transmitted codeword $\mathbf{c}$ may not be binary. In this case, many more bits are needed to transmit the codeword, resulting in code rate reduction. Here we propose a fast and simple mapping using subfield bases on the message vector $\mathbf{m}$ to make sure that the transmitted codeword $\mathbf{c}$ is also binary. In addition, to reduce the computational complexity of the first step of ETD, a post-processing step on the transformed codeword $\mathbf{c}^{\mathcal{F},\pi}$, which is equivalent to pre-processing of the message vector $\mathbf{m}$, will be presented.

Recall the conjugacy constraint (3) for a binary vector. For the transformed codeword $\mathbf{c}^{\mathcal{F}}$ or $\mathbf{c}^{\mathcal{F},\pi}$, it means

$$c^{\mathcal{F}}_{ej+(2i)_e} = (c^{\mathcal{F}}_{ej+i})^2, \tag{15}$$

or

$$c^{\mathcal{F},\pi}_{n(2i)_e+j} = (c^{\mathcal{F},\pi}_{ni+j})^2, \tag{16}$$

for $0 \le i < e$ and $0 \le j < n$. For the sake of simplicity, we denote the transformed codeword and the codeword by n blocks of length e, $\mathbf{c}^{\mathcal{F}} = [\mathbf{c}^{\mathcal{F}}_j]$ and $\mathbf{c} = [\mathbf{c}_j]$, respectively, where $c^{\mathcal{F}}_{j,i} = c^{\mathcal{F}}_{ej+i}$ and $c_{j,i} = c_{ej+i}$. It is clear that $\mathbf{c}_j$ is the inverse Galois Fourier transform of $\mathbf{c}^{\mathcal{F}}_j$. Similarly, we denote $\mathbf{c}^{\mathcal{F},\pi}$ by e blocks of length n, $\mathbf{c}^{\mathcal{F},\pi} = [\mathbf{c}^{\mathcal{F},\pi}_i]$, where $c^{\mathcal{F},\pi}_i = c^{\mathcal{F},\pi}_{in+j}$. Then (15) and (16) can be rewritten as

$$c^{\mathcal{F}}_{j,(2i)_e} = (c^{\mathcal{F}}_{j,i})^2, \tag{17}$$

and

$$c^{\mathcal{F},\pi}_{(2i)_e,j} = (c^{\mathcal{F},\pi}_{i,j})^2, \tag{18}$$

respectively.

We now show that if the message vector $\mathbf{m}$ is pre-processed by bases of subfields, then (17) or (18) is satisfied and hence the codeword $\mathbf{c}$ is binary.

Again for the sake of simplicity, we denote the message vector $\mathbf{m}$ and the pre-processed message vector $\hat{\mathbf{m}}$ by e blocks of size k, $\mathbf{m} = [\mathbf{m}_i]$ and $\hat{\mathbf{m}} = [\hat{\mathbf{m}}_i]$, respectively, where $m_{i,j} = m_{ik+j}$ and $\hat{m}_{i,j} = \hat{m}_{ik+j}$, $i = 0, 1, \ldots, e-1$ and $j = 0, 1, \ldots, k-1$.

Recall the definition of the i-th conjugacy class $\Psi_i = \{\mathbf{D}_{t_i}, \mathbf{D}^{\circ 2}_{t_i}, ..., \mathbf{D}^{\circ 2^{\eta_i - 1}}_{t_i}\}$ in (9) with size η_i. Since C_{t_i} is cyclotomic coset, η_i divides r. Thus, $2^{\eta_i} - 1$ is a factor of $2^r - 1$. There exists a subfield of size 2^{η_i} whose element's 2^{η_i} power all equals to itself. Suppose that α is a primitive element, then a basis $\beta_{i,0} = \alpha^{t_i}, \beta_{i,1} = \alpha^{2t_i}, \ldots, \beta_{i,\eta_i-1} = \alpha^{2^{\eta_i-1}t_i}$ spans the subfield GF(2^{η_i}) of GF(2^r) whose element's 2^{η_i} power all equals to itself. Thus, if γ is an element of GF(2^{η_i}), then $\gamma^{2^{\eta_i}} = \gamma$ and $\gamma = \sum_{l=0}^{\eta_i - 1} u_l \beta_{i,l}$, where u_l is in the ground field GF(2). If $\eta_i = 1$, then the subfield only has two elements, 0 and 1. If $\eta_i = r$, then the subfield is the field GF(2^r).

Using the bases $\beta_{i,l}$'s, we map message $\mathbf{m}$ to its pre-processed version $\hat{\mathbf{m}}$ via

$$\hat{m}_{(2^\mu t_i)_e,j} = \left(\sum_{l=0}^{\eta_i - 1} \beta_{i,l} m_{(2^l t_i)_e,j} \right)^{2^\mu}. \tag{19}$$

First, the mapping from $\mathbf{m}$ to $\hat{\mathbf{m}}$ is one-to-one since $\beta_{i,l}$'s are linearly independent over GF(2). Second, $\hat{m}_{(2^\mu t_i)_e,j} = \hat{m}^{2^\mu}_{(t_i)_e,j}$. A detailed proof of this fact is given in the appendix of [18]. The following theorem shows that the transformed codeword

$$\hat{\mathbf{c}}^{\mathcal{F}} = \{\hat{\mathbf{m}} \cdot \mathbf{G}^{\mathcal{F},\pi}\}^{\pi^{-1}} \tag{20}$$

encoded from the pre-processed message $\hat{\mathbf{m}}$ satisfies the conjugacy constraint.

Theorem 1. *The transformed codeword* $\hat{\mathbf{c}}^{\mathcal{F}} = \{\hat{\mathbf{m}} \cdot \mathbf{G}^{\mathcal{F},\pi}\}^{\pi^{-1}}$ *encoded from the pre-processed message* $\hat{\mathbf{m}}$ *satisfies the conjugacy constraint* $\hat{c}^{\mathcal{F}}_{j,(2i)_e} = (\hat{c}^{\mathcal{F}}_{j,i})^2$ *in (17) or* $\hat{c}^{\mathcal{F},\pi}_{(2i)_e,j} = (\hat{c}^{\mathcal{F},\pi}_{i,j})^2$ *in (18).*

Proof: From the definition of $\hat{\mathbf{c}}^{\mathcal{F},\pi}$, its $(ni+j)$-th symbol for $i = 0, 1, \ldots, e-1$, $0 \le s < k$ and $0 \le j < n$ is

$$\hat{c}^{\mathcal{F},\pi}_{i,j} = \sum_{s=0}^{k-1} \hat{m}_{i,s} D_{i,s,j}, \tag{21}$$

where $\mathbf{D}_i = [D_{i,s,j}]$. From the definition of $\hat{\mathbf{m}}$ (20) and the conjugacy constraint on $\mathbf{D}_i$ in (8), its $(n(2i)_e + j)$-th symbol is

$$\begin{aligned} \hat{c}^{\mathcal{F},\pi}_{t'} = \hat{c}^{\mathcal{F},\pi}_{(2i)_e,j} &= \sum_{s=0}^{k-1} \hat{m}_{(2i)_e,s} D_{(2i)_e,s,j}, \\ &= \sum_{s=0}^{k-1} \hat{m}^2_{i,s} D^2_{i,s,j}, \\ &= \left(\sum_{s=0}^{k-1} \hat{m}_{i,s} D_{i,s,j}\right)^2 \\ &= (\hat{c}^{\mathcal{F},\pi}_{i,j})^2 = (\hat{c}^{\mathcal{F},\pi}_{t})^2. \end{aligned} \tag{22}$$

■

Direct computation of (19) involves k Galois multiplications and $k-1$ Galois additions over GF(2^r), since both $\hat{\mathbf{m}}$ and $\mathbf{D}_i$ are non-binary symbols over GF(2^r). Thus, we rewrite (21) as

$$\begin{aligned} \hat{c}^{\mathcal{F},\pi}_{(2^\mu t_i)_e,j} &= \sum_{s=0}^{k-1} \hat{m}_{(2^\mu t_i)_e,s} D_{(2^\mu t_i)_e,s,j}, \\ &= \sum_{s=0}^{k-1} \left(\sum_{l=0}^{\eta_i-1} \beta_{i,l} m_{(2^l t_i)_e,s}\right)^{2^\mu} D^{2^\mu}_{t_i,s,j}, \\ &= \sum_{l=0}^{\eta_i-1} \beta^{2^\mu}_{i,l} \left(\sum_{s=0}^{k-1} m^{2^\mu}_{(2^l t_i)_e,s} D^{2^\mu}_{t_i,s,j}\right), \\ &= \left(\sum_{l=0}^{\eta_i-1} \beta_{i,l} c^{\mathcal{F},\pi}_{(2^l t_i)_e,j}\right)^{2^\mu}. \end{aligned} \tag{23}$$

Equation (23) shows that the mapping on message $\mathbf{m}$ (19) and the mapping on $\mathbf{c}^{\mathcal{F},\pi}$ (23) result in the same $\hat{\mathbf{c}}^{\mathcal{F},\pi}$, which satisfies the conjugacy constraint.

However, the mapping on $\mathbf{c}^{\mathcal{F},\pi}$ (23) involves much less Galois field multiplications than the one on message $\mathbf{m}$ (19). According to (23), computation of $\hat{c}^{\mathcal{F},\pi}_{t_i,j}$ is carried out in two steps. In the first step, the summation of $c^{\mathcal{F},\pi}_{(2^l t_i)_e,j} = \sum_{s=0}^{k-1} m_{(2^l t_i)_e,s} D_{(2^\mu t_i)_e,s,j}$ is calculated, which only involves $k-1$ additions, since $\mathbf{m}$ is a binary vector; in the second step, $\hat{c}^{\mathcal{F},\pi}_{t_i,j}$ is calculated, which involves η_i multiplications and $\eta_i - 1$ additions. The other codeword symbol $\hat{c}^{\mathcal{F},\pi}_{(2^\mu t_i)_e,j}$ in the conjugacy class of j can be simply calculated from $(\hat{c}^{\mathcal{F},\pi}_{t_i,j})^{2^\mu}$.

Before we study the computational complexity of our proposed ETD algorithm, we summarize its encoding steps. For the sake of implementational simplicity, we assume that the permutation π^{-1} operation is included in the initialization stage.

We present the computational complexity of the proposed ETD algorithm step by step. Since $\mathbf{m}$ is binary, the first step of encoding involves only $ek(n-k)$ binary field multiplication and $e(k-1)(n-k)$ Galois field additions. In the second step, each $\hat{c}^{\mathcal{F}}_{j,t_i}$ needs η_i Galois field multiplications and $\eta_i - 1$ Galois field additions; the computational complexity of this step is thus about ne Galois field multiplications and ne Galois field additions. Since the GFFT over GF(2^r) is about $\frac{1}{2} e \log_2 e$ Galois field operations, so the complexity of the third step, which requires n times GFFT, is less than $\frac{n}{2} e \log_2 e$.

In terms of the memory consumption, it is clear that most memory is spent on storing the transform matrix $\mathbf{G}^{\mathcal{F},\pi}$. Since it is a block diagonal matrix, we only need to store e matrices, each of which requires to store $(k(n-k))$ Galois symbols if it is systematic. Moreover, considering the conjugacy constraint (8), we only need to store λ representative matrices. Each symbol in the i-th representative matrix cost η_i bits to store. Thus, the overall memory consumption of ETD is $\sum_{i=0}^{\lambda} \eta_i k(n-k) = ek(n-k)$, which is the same as that of traditional encoding.

Considering that each Galois addition costs r bit operations, each Galois multiplication costs r^2 bit operations, and each Galois symbol costs r bits memory, we provide the specific comparison of complexity between ETD of binary QC codes and the traditional encoding algorithm by a table in [18]. Since

Algorithm 1 The Encoding of Binary QC Codes in the Transform Domain

Input:

The message $\mathbf{m}$ of ek bits;

The $ek \times en$ transformed generator matrix $\mathbf{G}^{\mathcal{F},\pi}$;

Output:

The binary transmitted codeword $\mathbf{c}$ of en bits;

Steps:

1) The message $\mathbf{m}$ is encoded into the transformed codeword $\mathbf{c}^{\mathcal{F}}$ by the transformed generator matrix $\mathbf{G}^{\mathcal{F},\pi}$

$$\mathbf{c}^{\mathcal{F}} = \{\mathbf{m} \cdot \mathbf{G}^{\mathcal{F},\pi}\}^{\pi^{-1}}$$

2) The transformed codeword $\mathbf{c}^{\mathcal{F}}$ is mapped into the conjugacy constraint satisfied codeword $\hat{\mathbf{c}}^{\mathcal{F}}$,

$$\hat{c}^{\mathcal{F}}_{j,(t_i 2^\mu)_e} = \left(\sum_{l=0}^{\eta_i-1} \beta_{i,l} c^{\mathcal{F}}_{j,(t_i 2^l)_e}\right)^{2^\mu}.$$

3) The binary transmitted codeword $\mathbf{c}$ is obtained by the inverse Galois Fourier transform from $\hat{\mathbf{c}}^{\mathcal{F}}$'s,

$$c_{j,i} = \sum_{l=0}^{e-1} \hat{c}^{\mathcal{F}}_{j,l} \alpha^{il}.$$

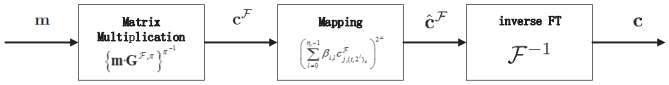

Fig. 1. The block diagram of the encoding of binary QC codes in the transformed domain.

$r \approx \log_2 e$, the overall computation complexity of ETD is less than $ek(n-k)\log_2 e + ne(\log_2^2 e + \log_2 e) + \frac{n}{2}e\log_2^3 e$. Thus the complexity of the transformed encoding is about $\mathcal{R} = \frac{2(n-k)ke^2}{ek(n-k)\log_2 e + ne(\log_2^2 e + \log_2 e) + \frac{n}{2}e\log_2^3 e}$ times lower than the complexity of the traditional encoding. Moreover, if n and k are much greater than e, Step 1) costs most computation and dominates the computational complexity, and thus the complexity of the transformed encoding can be simplified as $\mathcal{R} \approx \frac{2e}{\log_2 e}$.

Furthermore, as stated in Section II, if the circulant size e is even and there exists an odd number, e', where $e'|e$, we can decompose these circulants into smaller circulants of size e'. Then $\mathbf{G}$, the generator matrix of the (en, ek) QC-code, becomes a $\frac{ke}{e'} \times \frac{ne}{e'}$ matrix of $e' \times e'$ circulant matrices. If $e' = 2^r - 1$, the computational complexity is less than $\frac{e^2 k(n-k)\log_2 e'}{e'} + ne(\log_2^2 e' + \log_2 e') + \frac{n}{2}e\log_2^3 e'$.

Example 1. *Considering a (4095, 2016) QC-LDPC with circulant size 63, i.e.,* $n = 65$, $k = 32$, $n - k = 33$, $e = 63$ *and* $r = 6$, *we have,* $\mathcal{R} = 8.27$. *In other words, the computational complexity of ETD is only* 12.09% *of that of traditional encoding.*

Example 2. *Considering a (15500, 10850) QC-LDPC with circulant size 31, i.e.,* $n = 500$, $k = 350$, $n - k = 150$, $e = 31$ *and* $r = 5$, *we have,* $\mathcal{R} = 10.54$. *In other words, the computational complexity of ETD is only* 9.49% *of that of traditional encoding.*

IV. Conclusion

In this paper, we have proposed an efficient encoding algorithm for binary QC codes in the transform domain. Its complexity is much lower than traditional encoding for binary. For example, the computation complexities of ETD of the binary $(4095, 2016)$ and $(15500, 10850)$ QC codes are respectively 12.09% and 9.49% of those of traditional encoding.

References

[1] R. L. Townsend and E. J. Weldon, Jr., "Self-orthogonal quasi-cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-13, no. 2, pp. 183-195, Apr. 1967.

[2] T. Kasami, "A Gilbert-Varshamov bound for quasi-cycle codes of rate 1/2," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 5, p. 679, Sep. 1974.

[3] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd edition. Upper Saddle River, NJ: Prentice Hall, 2004.

[4] N. Bonello, C. Sheng Chen, and L. Hanzo, "Construction of regular quasi-cyclic protograph LDPC codes based on Vandermonde matrices," *IEEE Trans. Vehicular Technology*, vol. 57, no. 4, pp. 2583–2588, Jul. 2008.

[5] Y. Y. Tai, L. Lan, L. Zheng, S. Lin and K. Abdel-Ghaffar, " Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels," *IEEE Trans. Commun.*, vol 54, no. 7, pp. 1765–1774, Oct. 2006.

[6] J. L. Fan, "Array codes as low-density parity-check codes," in *proc. Int. Symp. Turbo Codes,* Brest, France, Sep. 2-7, 2000, pp.545-546.

[7] S. Myung and K. Yang, " A combining method of quasi-cyclic LDPC codes by the Chinese remainder theorem," *IEEE Commun. Lett.*, vol. 9, no. 9, pp. 823–825, Sep. 2005.

[8] K. Lally and P. Fitzpatrick, "Algebraic structure of quasicyclic codes," *Disc. Appl. Math.*, vol. 111, pp. 157–175, 2001.

[9] S. J. Johnson and S. R. Weller, "A family of irregular LDPC codes with low encoding complexity," *IEEE Commun. Lett.*, vol. 7, no. 2, pp. 79–81, Feb. 2003.

[10] M. Yang and W. E. Ryan, "Performance of efficiently encodable low-density parity-check codes in noise bursts on the EPR4 channel," *IEEE Trans. Magn.*, vol. 40, no. 2, part 1, pp. 507–512. Mar. 2004.

[11] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: A finite field approach," *IEEE Trans. Inform. Theory*, vol. 53, no. 7, pp. 2429–2458, Jul. 2007.

[12] Z. Li, L. Chen, L. Zeng, S. Lin and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, 2006.

[13] Q. Diao, Q. Huang, S. Lin, and K. Abdel-Ghaffar, "A matrix theoretic approach for analyzing quasi-cyclic low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 58, no. 6, pp. 4030–4048, June. 2012.

[14] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.

[15] Q. Diao, Q. Huang, S. Lin, and K. Abdel-Ghaffar, "A transform approach for computing the ranks of parity-check matrices of quasi-cyclic LDPC codes," *Proc. 2011 IEEE Int. Symp. Inform. Theory*, SaintPetersburg, Russia, pp. 366-379, July 31-Aug. 5, 2011.

[16] R. M. Tanner, "A transform theory for a class of group-invariant codes," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, pp. 752–775, Jul. 1988.

[17] Q. Huang, Q. Diao, S. Lin, and K. A. Ghaffar, "Cyclic and quasi-cyclic LDPC codes on constrained parity-check matrices and their trapping sets," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2648-2671, May 2012.

[18] Q. Huang, L. Tang, Z. Wang, Z. Xiong and S. He, "A Low Complexity Encoding of Quasi-Cyclic Codes Based on Galois Fourier Transform," *arXiv:1301.3220v1*.

[19] F. P. Preparata and D. V. Sarwate, "Computational complexity of Fourier transforms over finite field," *Mathematics of Computation*, vol. 31, no. 139, pp. 740,751, Jul. 1977.