A Novel Coding Scheme for Encoding and Iterative Soft-Decision Decoding of Binary BCH Codes of Prime Lengths

Shu Lin and Khaled Abdel-Ghaffar  
University of California, Davis  
Davis, CA 95616, USA

Juane Li  
AppoTech USA Inc.  
Sunnyvale, CA 94085, USA

Keke Liu  
Broadcom Limited  
San Jose, CA 95131, USA

Abstract—A novel coding scheme is presented for encoding and iterative soft-decision decoding of binary BCH codes of prime lengths. The encoding of such a BCH code is performed on a collection of codewords which are mapped through Galois Fourier transform into a codeword of a nonbinary low-density parity-check (LDPC) code which has a binary parity-check matrix for transmission. Using this matrix, a binary iterative soft-decision decoding algorithm based on belief-propagation is applied to jointly decode a collection of codewords of the BCH code. The joint-decoding allows information sharing among the received vectors corresponding to the codewords in the collection during the iterative decoding process. For decoding a BCH code of prime length, the proposed decoding scheme not only requires low decoding complexity, but also yields superior performance. The proposed decoding scheme can achieve a joint-decoding gain over the maximum likelihood decoding of individual codewords.

I. Introduction

BCH (Bose-Chaudhuri-Hocquenghem) codes form a powerful class of cyclic error-correcting codes [1]–[5]. These codes have been widely used in both communication and data storage systems over the last 50 years and are still being used nowadays. They can be effectively decoded with the Berlekamp-Massey hard-decision decoding algorithm (BM-HDDA) [5], [6]. However, the BM-HDDA for BCH codes fails to exploit soft reliability information readily available at the output of the detector. It has long been known and shown that using soft reliability information of the received code symbols in decoding a BCH code can improve its performance over the BM-HDDA. The degree of this performance improvement very much depends on how and what amount of soft reliability information is being used. Improving the performance of a BCH code using soft reliability information for decoding is always achieved at the expense of the decoder complexity. Generally speaking, a larger amount of performance improvement requires a larger amount of decoding complexity.

In this paper, we present a novel coding scheme for encoding and iterative soft-decision decoding (ISDD) of BCH codes over GF(2) of prime lengths. The key to this coding scheme is to map a collection of codewords in a binary BCH code \( C_{\text{BCH}} \) of prime length into a codeword of a powerful nonbinary quasi-cyclic LDPC (QC-LDPC) code [7], [8] whose parity-check matrix is an array of binary circulant permutation matrices (CPMs). The Tanner graph of the LDPC code has girth at least 6. At the decoder, the received vector is first decomposed into a set of binary constituent received vectors. Then, each of these binary constituent received vectors is decoded iteratively based on the binary parity-check matrix of the LDPC code with a binary ISDD algorithm based on belief propagation (BP) [7]–[9]. The successfully decoded binary vectors are then combined and transformed into a collection of decoded codewords in \( C_{\text{BCH}} \) through inverse mapping and inverse permutations. Errors not corrected by the ISDD decoder can be decoded using a BM-HDDA decoder for the BCH code \( C_{\text{BCH}} \). The mapping involves symbol permutation within each codeword, combining codewords into a codeword in a composite BCH code over an extension field of the binary field GF(2), interleaving the permuted codewords in the composite BCH code, and Galois Fourier transform (GFT) [10]. The inverse mapping involves the inverse GFT, de-interleaving, de-combining codewords, and inverse symbol permutation of each decoded codeword.

The most important feature of the proposed ISDD scheme is that the decoding is performed on a collection of received BCH codewords jointly. During the decoding process, the reliability information of each decoded codeword is shared by others to enhance the overall reliability of all the decoded codewords. This joint-decoding and information sharing may result in an error performance per decoded codeword better than that of a received codeword decoded individually using the maximum-likelihood decoding (MLD).

The rest of this paper is organized as follows. Section II defines a class of binary BCH codes of prime lengths and their Hadamard-equivalents. Section III gives a construction of a composite BCH code by combining codewords in a binary BCH code of prime length. In Section IV, we apply the GFT to transform a collection of codewords in a composite BCH code into a codeword in an LDPC code over a finite field of characteristic 2 which has a binary parity-check matrix. This binary parity-check matrix has good structural properties which allow for excellent performance when used for ISDD. In Sections V and VI, we develop a coding scheme for encoding and decoding a collection of codewords in a BCH code of prime length in the GFT-domain. We show that the proposed collective encoding and decoding schemes require low implementation complexities. Section VII presents...
a method for constructing a family of rate-compatible GFT-BCH-LDPC codes with rates ranging from high to low based on a given BCH code of prime length. Codes in the same family can be encoded and decoded with the same encoder and the same decoder. In Section [VIII], experimental results are presented to demonstrate that the proposed coding scheme for BCH codes of prime lengths yields superior performance. Section [IX] concludes the paper with some remarks.

II. BINARY BCH CODES OF PRIME LENGTHS AND THEIR HADAMARD-EQUIVALENTS

Let $\alpha$ be a primitive element of $GF(2^t)$. Suppose $2^t - 1$ can be factorized as a product of two integers $c$ and $n$, where $n$ is a prime. Let $\beta = \alpha^c$. Then, $\beta$ is an element of $GF(2^t)$ of order $n$, i.e., $\beta^n = 1$. The set $S = \{\beta^0, \beta^1, \ldots, \beta^{n-1}\}$ forms a cyclic subgroup of $GF(2^t)$. Let $t$ and $m$ be two positive integers such that $1 \leq 2t \leq n < m$. Let $S = \{\beta^l, \beta^l, \ldots, \beta^{l+m-1}\}$ be a set of $m$ elements in the cyclic subgroup $S$, which consists of $2t$ consecutive powers of $\beta$, say $\beta^0, \beta^1, \ldots, \beta^{2t-1}$, and their conjugates, i.e., if $\eta$ is an element of $S$, then $\eta^t = \beta^t \eta$. Moreover, any of its generator polynomial 

BCH codes have the following structural property: any of its generator polynomial $B_{BCH}$ of degree $n$, i.e., $\pi_k(j) = (j-k)n$ for $1 \leq k < n$ and $0 \leq j < n$. Since $n$ is a prime, $\pi_k(j) \neq \pi_k(j')$ for $0 \leq j 
eq j' < n$. Hence, $\pi_k$ is a permutation on $\{0, 1, \ldots, n-1\}$ (the labels of the columns of $B_{BCH}$), which we call the $k$-th Hadamard-permutation. Therefore, for $1 \leq k < n$, $B_{BCH}^{\circ k}$ is simply a column-permutation of $B_{BCH}$. Clearly, $B_{BCH}^{\circ k}$, $1 \leq k < n$, has the same rank as $B_{BCH}$, which is $n$. Note that $B_{BCH}^{\circ k} = B_{BCH}$. For $k = 0$, $B_{BCH}^{\circ 0}$ is an all-one matrix of size $m \times n$. Since $B_{BCH}$ satisfies the $2 \times 2$ SNS-constraint, $B_{BCH}^{\circ k}$, $1 \leq k < n$, also satisfies the $2 \times 2$ SNS-constraint.

For $1 \leq k < n$, let $C_{BCH}^{(k)}$ be the code generated by the null space over $GF(2)$ of $B_{BCH}^{\circ k}$. Then, $C_{BCH}^{(k)}$ is an $(n, n-m)$ code over $GF(2)$ which is equivalent to the $(n, n-m)$ BCH code $C_{BCH}$ generated by the null space over $GF(2)$ of $B_{BCH}$ defined by (1). The code $C_{BCH}^{(k)}$ can be obtained from $C_{BCH}$ by applying the permutation $\pi_k$ to its codewords (with the coordinates of each codeword labeled from $0$ to $n-1$). We call $C_{BCH}^{(k)}$ the $k$-th Hadamard-equivalent of $C_{BCH}$. Hereafter, we use the notation $\pi_k$ to denote both the $k$-th Hadamard-permutation of the columns of the matrix $B_{BCH}$, and the $k$-th Hadamard-permutation of the codewords of the code $C_{BCH}$. Applying the permutation $\pi_k$ to a codeword in $C_{BCH}$ results in a codeword in $C_{BCH}^{(k)}$. Note that a vector $v = (v_0, v_1, \ldots, v_{n-1})$ over $GF(2)$ is in the null space of over $GF(2)$ of $B_{BCH}$ if and only if the polynomial $v(X) = \sum_{i=0}^{n-1} v_i X^i$ associated with $v$ has $\beta^k v_0, \beta^k v_1, \ldots, \beta^k v_{n-1}$ as roots. Hence, for $1 \leq k < n$, $C_{BCH}^{(k)}$ is a cyclic code over $GF(2)$ whose generator polynomial is $g_{BCH}^{(k)}(X) = \prod_{i=0}^{k-1} (X - \beta^k \eta^i)$. Let $\omega = \beta^k$. Then, the generator polynomial $g_{BCH}^{(k)}(X)$ of $C_{BCH}^{(k)}$ has $\omega, \omega^2, \ldots, \omega^{2t}$ and their conjugates as roots. Hence, the Hadamard-equivalent $C_{BCH}^{(k)}$ of $C_{BCH}$ is also a BCH code with minimum distance $2t + 1$. For $k = 0$, the null space over $GF(2)$ of $B_{BCH}^{\circ 0}$ generates an $(n, n-1)$ code $C_{BCH}^{(0)}$ of length $n$ and dimension $n-1$. This code, $C_{BCH}^{(0)}$, which has a single parity-check (SPC) symbol, is called an SPC code [7]. It is also a cyclic code generated by $g_{BCH}^{(0)}(X) = (X-1)$. Note that for $k = 1$, $B_{BCH} = B_{BCH}^{\circ 1}$ and $C_{BCH} = C_{BCH}^{(1)}$.

III. CASCADING AND INTERLEAVING A COMPOSITE BCH CODE OF PRIME LENGTH AND ITS HADAMARD-EQUIVALENTS

For $0 \leq k < n$, let $c_{0,k}, c_{1,k}, \ldots, c_{t-1,k}$ be $t$ codewords in $C_{BCH}^{(k)}$. Combine these $t$ codewords of length $n$ into a composite vector $c_k$ of length $n$ over $GF(2^t)$ as follows:

$c_k = \sum_{i=0}^{t-1} c_{i,k} \alpha^i$, where $\alpha$ is a primitive element in $GF(2^t)$. Note that the $j$-th component of $c_k$, $0 \leq j < n$, is the weighted sum of the $j$-th components of $c_{0,k}, c_{1,k}, \ldots, c_{t-1,k}$, weighted by $\alpha^0, \alpha^1, \ldots, \alpha^{t-1}$, respectively. Let $C_{BCH,T}$ be the collection of all sequences $c_k$ corresponding to all vectors $(c_{0,k}, c_{1,k}, \ldots, c_{t-1,k})$ of $t$ codewords in $C_{BCH}^{(k)}$. It is an...
Let $C_{\text{BCH-casc}}$ be the code over $GF(2^q)$ of length $n^2$ obtained by cascading the composite BCH code $C_{\text{BCH}}$, over $C_{\text{BCH}}$, and its Hadamard-equivalents. A codeword $c_{\text{casc}}$ in $C_{\text{BCH-casc}}$ is in the form of $c_{\text{casc}} = (c_0, c_1, \ldots, c_{n-1})$ with $c_k$ in $C_{\text{BCH}}$ for $0 \leq k < n$. The code $C_{\text{BCH-casc}}$ has length $n^2$, dimension $(n-1)+(n-m)(n-1) = (n-m+1)(n-1)$, and is called a cascaded composite (CC) BCH code. A parity-check matrix of the CC-BCH code $C_{\text{BCH-casc}}$ is given by:

$$H_{\text{BCH-casc}} = \text{diag}(B_{0,0}^{\text{BCH}}, B_{1,0}^{\text{BCH}}, \ldots, B_{2(n-1)}^{\text{BCH}})$$

Let $C_{\text{BCH-casc}}$ be the collection of all vectors, $e_{\text{casc}}$, by interleaving all codewords, $c_{\text{casc}}$, in $C_{\text{BCH-casc}}$. Then, $C_{\text{BCH-casc}}$ is a linear code over $GF(2^q)$ of length $n^2$ and dimension $(n-m+1)(n-1)$ which are equal to those of $C_{\text{BCH-casc}}$. The code $C_{\text{BCH-casc}}$ is called the interleaved cascaded composite (ICC-BCH) code of the base BCH code $C_{\text{BCH}}$. Each codeword in the ICC-BCH code $C_{\text{BCH-casc}}$ contains $(n-1)$r codewords in $C_{\text{BCH}}$ and its Hadamard-equivalents, $\tau$ codewords from each, and $\tau$ codewords in $C_{\text{BCH}}$.

A parity-check matrix for $C_{\text{BCH-casc}}$ can be obtained by applying the permutations $\pi_{\text{col}}$ and $\pi_{\text{row}}$ to the columns and rows of the parity-check matrix $H_{\text{BCH-casc}}$ of $C_{\text{BCH-casc}}$, respectively. This gives

$$H_{\text{BCH-casc}}^{\tau} = [D_{e,f}]_{0\leq e < \tau m, 0\leq f < n},$$

where, for $0 \leq e < m$ and $0 \leq f < n$, $D_{e,f}$ is an $n \times n$ diagonal matrix over $GF(2^q)$ of the following form:

$$D_{e,f} = \text{diag}(1, \beta^e f_1, \beta^e f_2, \ldots, \beta^e (n-1) f_1).$$

Note that $D_{e,f}$ is formed by the root $\beta^e$ of the generator polynomial $g_{\text{BCH}}(X)$ of the base $(n,n-m)$ BCH code $C_{\text{BCH}}$.

IV. TRANSFORMING AN ICC-BCH CODE INTO A BCH-LDPC CODE IN THE GFT-DOMAIN

A. The GFT of an ICC-BCH Code of Prime Length

Let $V = [\beta^j_{i,j}]_{0 \leq i,j < n}$. Then, $V$ is an $n \times n$ Vandermonde matrix \[10, 12\] which is nonsingular with inverse $V^{-1} = [\beta^{-j}_{i,j}]_{0 \leq i,j < n}$. The Galois Fourier transform (GFT) of a vector $a = (a_0, a_1, \ldots, a_{n-1})$ over $GF(2^q)$ is the vector $F(a) = a\cdot V$. If $b = F(a)$, then $a = b\cdot V^{-1}$ is the inverse GFT of the vector $b$, denoted by $F^{-1}(b)$.

Let $c_{\text{casc}} = (c_0^0, c_1^0, \ldots, c_{n-1}^0)$ be a codeword in the ICC-BCH code $C_{\text{BCH-casc}}$, where each of $c_0^0, c_1^0, \ldots, c_{n-1}^0$ is a vector over $GF(2^q)$ of length $n$. For $0 \leq i < n$, let $c_i^j = F^j(c_i^0) = c_i^0 e^j\cdot V$ and $c_{\text{casc}}^j = (c_0^j, c_1^j, \ldots, c_{n-1}^j)$. Then, $c_{\text{casc}}^j = c_{\text{casc}}^0\cdot \text{diag}(V, V, \ldots, V)$, where $\text{diag}(V, V, \ldots, V)$ is an $n \times n$ diagonal array with $n$ copies of $V$ lying on its main diagonal and zero matrices of size $n \times n$ elsewhere. The vector $c_{\text{casc}}^j$ is referred to as the GFT of $c_{\text{casc}}^0$. The inverse GFT of $c_{\text{casc}}^j$ gives back the ICC-BCH codeword $c_{\text{casc}}^0$, i.e., $c_{\text{casc}}^0 = c_{\text{casc}}^j\cdot \text{diag}(V^{-1}, V^{-1}, \ldots, V^{-1})$.

Let $C_{\text{BCH-casc}}$ be the collection of all vectors, $c_{\text{casc}}^j$, corresponding to all codewords, $c_{\text{casc}}^0$, in $C_{\text{BCH-casc}}$. Then, $C_{\text{BCH-casc}}$ is a linear code over $GF(2^q)$ of length $n^2$ and dimension $(n-m+1)(n-1)$, which are equal to those of $C_{\text{BCH-casc}}$. We call the code $C_{\text{BCH-casc}}$ the GFT of the ICC-BCH code $C_{\text{BCH-casc}}$, referred to as GFT-ICC-BCH code. The
code $C_{\text{BCH,casc}}^{\pi, F}$ is composed of $n-1$ copies of the base BCH code $C_{\text{BCH}}$ (in permuted and non-permuted forms) and one SPC code $C_{\text{BCH}}^{(0)}$ in GFT-domain. Based on the definition of $c_{\text{case}}^{\pi, F} = c_{\text{case}}^{\pi, F} \text{diag}(V, V, \ldots, V)$ and the fact that $H_{\text{BCH,casc}}^{\pi, F}$ is a parity-check matrix of $C_{\text{BCH,casc}}^{\pi, F}$, we readily see that the matrix defined by

$$H_{\text{BCH,casc}}^{\pi, F} = \text{diag}(V, V, \ldots, V) H_{\text{BCH}}^{\pi, F} \text{diag}(V^{-1}, V^{-1}, \ldots, V^{-1}).$$

(9)

is a parity-check matrix of the GFT-ICC-BCH code $C_{\text{BCH,casc}}^{\pi, F}$.

From (7) and (9), it follows that

$$H_{\text{BCH,casc}}^{\pi, F} = [VD_{c_{\text{case}}^{\pi, F}} V^{-1}]_{0 \leq e < m, 0 \leq f < n}.$$  

(10)

It is straightforward to check from (8) that for $0 \leq e < m$ and $0 \leq f < n$, $VD_{c_{\text{case}}^{\pi, F}} V^{-1}$ is a circulant over $GF(2)$ in which each row contains a single nonzero entry which equals 1. Such a circulant is called circulant permutation matrix (CPM). Let $D_{c_{\text{case}}^{\pi, F}}$ denote $VD_{c_{\text{case}}^{\pi, F}} V^{-1}$. Then, from (10),

$$H_{\text{BCH,casc}}^{\pi, F} = [D_{c_{\text{case}}^{\pi, F}} V^{-1}]_{0 \leq e < m, 0 \leq f < n},$$

(11)

which is an $m \times n$ array of CPMs over $GF(2)$ of size $n \times n$.

Each row-block of $H_{\text{BCH,casc}}^{\pi, F}$ consists of $n$ binary CPMs of size $n \times n$ and each column-block has $m$ binary CPMs of size $n \times n$. Hence, $H_{\text{BCH,casc}}^{\pi, F}$ has column and row weights $m$ and $n$, respectively.

It follows from the above developments that $H_{\text{BCH,casc}}^{\pi, F}$ given by (11) can be constructed directly from its $m \times n$ BCH matrix $B_{\text{BCH}} = [\beta^{j i}]_{0 \leq e < m, 0 \leq j < n}$ given by (1) with a replacement process. Recall that the entries of $B_{\text{BCH}}$ are elements from the cyclic subgroup $S = \{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$ of the extension field $GF(2)$ of order $n$. If $\beta$ is an element in $GF(2)$, then $n$ is a prime factor of $2^n - 1$.

If $c_{\text{case}}^{\pi, F}$ is carried out as follows. For $0 \leq i < m$ and $0 \leq j < n$, we replace the $(i,j)$-entry $\beta^{j i}$ of $B_{\text{BCH}}$ by an $n \times n$ binary CPM, with rows and columns labeled from 0 to $n-1$, whose generator (the top row) has its single nonzero component “1” at location $(j l_i)$, the least nonnegative integer congruent to $j l_i$ modulo $n$. We denote this CPM by $C_{\text{BCH,casc}}^{\pi, F}$ and call it the CPM-dispersion of $\beta^{j i}$. Replacing all the entries of $B_{\text{BCH}}$ by their corresponding CPM-dispersions, we obtain the matrix $H_{\text{BCH,casc}}^{\pi, F}$, which is called the CPM-dispersion of $B_{\text{BCH}}$.

B. The LDPC-Structure of a GFT-ICC-BCH Code

For large $n$, $H_{\text{BCH,casc}}^{\pi, F}$ is a low-density matrix, the null space over $GF(2^\tau)$ of which can be considered as an LDPC code over $GF(2^\tau)$. Since $C_{\text{BCH,casc}}^{\pi, F}$ is the null space over $GF(2^\tau)$ of $H_{\text{BCH,casc}}^{\pi, F}$, $C_{\text{BCH,casc}}^{\pi, F}$ is an LDPC code over $GF(2^\tau)$, called a BCH-LDPC code. Hence, we denote $H_{\text{BCH,casc}}^{\pi, F}$ simply by $H_{\text{BCH,LDPC}}$ and and its null space over $GF(2^\tau)$, $C_{\text{BCH,LDPC}}^{\pi, F}$, as $C_{\text{BCH,LDPC}}$. Since $H_{\text{BCH,LDPC}}$ is an array of binary CPMs, $C_{\text{BCH,LDPC}}$ is a QC-BCH-LDPC code.

The parity-check matrix $H_{\text{BCH,LDPC}}$ of the QC-BCH-LDPC code $C_{\text{BCH,LDPC}}$ associated with the base BCH code $C_{\text{BCH}}$ has several properties which are relevant to its use in iterative decoding based on an ISDD algorithm. First, since the parity-check matrix $B_{\text{BCH}}$ of $C_{\text{BCH}}$ satisfies the $2 \times 2$ SNS-constraint, it follows from [13] Corollary 1 and [14] Proposition 1 that $H_{\text{BCH,LDPC}}$ satisfies the RC-constraint [8], i.e., no two rows (or two columns) in $H_{\text{BCH,LDPC}}$ have more than one position where they both have 1-components. This RC-constraint ensures that the Tanner graph associated with $H_{\text{BCH,LDPC}}$ has girth at least 6, which is typically required for iterative decoding algorithms, such as the sum-product algorithm (SPA) and the min-sum algorithm (MSA), to achieve good performance. It follows from the RC-constraint structure of $H_{\text{BCH,LDPC}}$ and orthogonal principle of majority-logic decoding [7], [8] that the minimum distance of the QC-BCH-LDPC code $C_{\text{BCH,LDPC}}$ is at least $m + 1$. Note that the minimum distance of the base BCH code $C_{\text{BCH}}$ generated by the matrix $B_{\text{BCH}}$ given by (1) is at least $2 \tau + 1$. Since $2 \tau \leq m$, the minimum distance of the QC-BCH-LDPC code $C_{\text{BCH,LDPC}}$ may be much larger than that of the base BCH code $C_{\text{BCH}}$.

Since the LDPC matrix $H_{\text{BCH,LDPC}}$ is RC-constrained with constant column weight $m$, it has no absorbing set of size less than $m/2 + 1$ and no small trapping set of size less than $m - 3$ [15]. This implies that the $2^\tau$-ary QC-BCH-LDPC code $C_{\text{BCH,LDPC}}$ given by the null space of $H_{\text{BCH,LDPC}}$ decoded with either the SPA or the MSA does not suffer from high error-floors if $m$ is reasonably large. Furthermore, note that $H_{\text{BCH,LDPC}}$ is an array of CPMs which simplifies wire routing in the decoder and allows for using the reduced-complexity decoding schemes proposed in [16] to reduce the hardware implementation complexity of an iterative decoder.

Finally, the most striking structure is that $H_{\text{BCH,LDPC}}$ is a binary matrix, even though its null space $C_{\text{BCH,LDPC}}$ is a nonbinary code over $GF(2^\tau)$. This binary property considerably simplifies the decoding of the QC-BCH-LDPC code $C_{\text{BCH,LDPC}}$. A vector $r = (r_0, 1, \ldots, r_{n-1})$ over $GF(2)$ is in the null space of the binary matrix $H_{\text{BCH,LDPC}}$ if and only if each of the $\tau$ binary constituent vectors $r_{\tau i} = (r_{0,i}, r_{1,i}, \ldots, r_{n^2-1,i})$ for $0 \leq i < \tau$, is in the null space over $GF(2)$ of $H_{\text{BCH,LDPC}}$, where $(r_{j,0}, r_{j,1}, \ldots, r_{j,\tau-1})$ is the binary vector representation of the received symbol $r_j$ which is an element in $GF(2^\tau)$ for $0 \leq j < n^2$. The subscript “$b$” in $r_{\tau i}$ stands for “binary”. Thus, the decoding of a received vector $r$ over $GF(2^\tau)$ can be implemented by performing $\tau$ decodings of its $\tau$ binary constituent vectors using a binary ISDD algorithm. This reduces the decoding complexity from a function of $(2^\tau)^2$ for direct implementation of the $2^\tau$-ary SPA [17], or $2^\tau$ for fast Fourier transform implementation of the $2^\tau$-ary SPA [18], to a function of $\tau$, i.e., $\tau$ times the complexity of a binary ISDD algorithm.

The mapping of a collection of codewords in a binary BCH code of prime length into a codeword in a BCH-LDPC code...
code given above is similar to the mapping of a collection of codewords in a Reed-Solomon (RS) code of prime length into an RS-LDPC code presented in our recent paper [19], except that the mapping carried out above is through an ICC-BCH code.

V. ENCODING OF AN ICC-BCH CODE IN THE GFT-DOMAIN

The encoding of a collection of codewords in a binary BCH code of prime length and mapping the collection to a codeword in a BCH-LDPC code through GFT can be carried out with the following five steps:

- **Enc-1.** Encode $\tau$ messages, $M_{i,0}$, each composed of $n - 1$ binary symbols, into $\tau$ codewords, $v_{i,0}$, $0 \leq i < \tau$, in the SPC code $C_{BCH}$ and $(n - 1)\tau$ messages, $M_{i,k}$, each composed of $n - m$ binary symbols, into $(n - 1)\tau$ codewords, $v_{i,k}$, $0 \leq i < \tau$, $1 \leq k < n$, in the base BCH code $C_{BCH}$.

- **Enc-2.** Apply the Hadamard-permutation $\pi_k$ on each $v_{i,k}$, for $0 \leq i < \tau$, $1 \leq k < n$, to obtain a codeword $c_{i,k}$ of length $n$ over GF(2) in the code $C_{BCH}$. No permutation is performed on the code $c_{BCH}$ and we set $c_{i,0} = v_{i,0}$ for $0 \leq i < \tau$.

- **Enc-3.** Combine the $\tau$ codewords in $C_{BCH}$ to form a composite codeword $c_k$ of length $n$ over GF(2) in the composite code $C_{BCH,casc}$ for each $k, 0 \leq k < n$. Cascade the $n$ composite codewords $c_0, c_1, \ldots, c_{n-1}$ to form a cascaded codeword $c_{casc} = (c_{0}, c_{1}, \ldots, c_{n-1})$ over GF(2) in the cascaded composite code $C_{BCH,casc}$.

- **Enc-4.** Apply the permutation $\pi_{col}$, defined by (9), to $c_{casc}$, i.e., interleave $c_{casc}$ to obtain the interleaved cascaded codeword $c_{casc} = (c_{0}, c_{2}, \ldots, c_{n-1})$ in the ICC-BCH code $C_{BCH,casc}$.

- **Enc-5.** Take the GFT of each component word $c^e_j$, $0 \leq j < n$, in $c_{casc}$ to obtain $c^{\tau \times \tau}_{casc}$ over GF(2) and form the vector $c^{\tau \times \tau}_{casc} = (c^e_{0}, c^e_{2}, \ldots, c^e_{n-1})$ of length $n^2$ over GF(2) for transmission. The vector $c^{\tau \times \tau}_{casc}$ is a codeword in the QC-BCH-LDPC code $C_{BCH,LDPC}$ which has $H_{BCH,LDPC}$ given by (10) as a parity-check matrix.

Note that with the above GFT encoding, encoding operations, messages to codewords, are only performed at the first encoding step, and the operations performed in the rest four steps are simply Hadamard-permutations, compositions, cascading, interleaving, and GFTs. The encoding operations can be carried out with two simple feedback shift-registers, one for $C_{BCH}^{(0)}$ and one for $C_{BCH}^{(k)}$. Combining the $\tau$ codewords in $C_{BCH}^{(k)}$ to form a codeword $c_k$ in $C_{BCH,casc}$ is based on representing a sequence of $\tau$ symbols in GF(2) as a symbol in GF(2$^\tau$). This can be easily implemented by a look-up-table. The cascading and interleaving can be performed together using a memory unit to store the code symbols of $c_0, c_1, \ldots, c_{n-1}$ in an $n \times n$ array $A$ with $c_0$ as the top row and $c_{n-1}$ as the bottom row. For $0 \leq k < n$, the $k$-th column of $A$ gives the word $c_k^e$. The GFT is performed on each column of the array $A$ as it is read out from the memory for transmission. Computing the GFTs of the columns of the array $A$ can be implemented efficiently with a fast algorithm [20].

The codeword $c^{\tau \times \tau}_{casc}$ at the output of the transmitter comprises $\tau$ codewords in $C_{BCH}^{(0)}$ and $(n - 1)\tau$ codewords in the base BCH code $C_{BCH}$, $\tau$ un-permuted and $(n - 2)\tau$ permuted. We can perform the encoding of $C_{BCH}^{(0)}$ in a way such that all the $n\tau$ constituent codewords in $c^{\tau \times \tau}_{casc}$ are codewords in $C_{BCH}$, permuted or un-permuted. This is done as follows. A message $M_{i,0}, 0 \leq i < \tau$, of $n - m$ information symbols with the first symbol set to zero, is first encoded into a codeword $v_{i,0}$ in $C_{BCH}$. In systematic form, the first code symbol in $v_{i,0}$ is zero. Removing this zero symbol, we obtain $n - 1$ code symbols. Then, we encode these $n - 1$ code symbols into a codeword $c_{i,0}$ in $C_{BCH}^{(0)}$, i.e., adding an overall parity-check symbol. The $\tau$ codewords $c_{i,0}, c_{i,1}, \ldots, c_{i,1-1}$ in $C_{BCH}^{(0)}$, we obtain a composite codeword $c_{i} \in C_{BCH,casc}$ and use it in the rest of encoding steps. This results in an output codeword $c^{\tau \times \tau}_{casc}$ in the QC-BCH-LDPC code $C_{BCH,LDPC}$ which comprises $n\tau$ codewords in the base BCH code $C_{BCH}$.

VI. DECODING OF AN ICC-BCH CODE IN THE GFT-DOMAIN

A. A Binary ISDD Scheme

Let $r = (r_0, r_1, \ldots, r_{n^2-1})$ be a received vector (the output of the detector) corresponding to the transmitted LDPC code $c^{\tau \times \tau}_{casc} = (c_{0}, c_{2}, \ldots, c_{n-1})$. Decoding of the vector $r$ of $n^2$ received symbols over GF(2$^\tau$) is based on the BCH-LDPC matrix $H_{BCH,LDPC}$ given by (10) using an ISDD algorithm to obtain an estimate $c^{\tau \times \tau}_{casc} = (c_{0}, c_{2}, \ldots, c_{n-1})$ of the transmitted codeword $c^{\tau \times \tau}_{casc}$ formed by the encoder in encoding step Enc-5. Recall that $H_{BCH,LDPC}$ is a binary matrix. Based on the properties developed in Section IV, the received vector $r$ can be decoded in binary using the binary matrix $H_{BCH,LDPC}$ as follows. For $0 \leq j < n^2$, let $(r_{j,0}, r_{j,1}, \ldots, r_{j,\tau-1})$, a $\tau$-tuple over GF(2), be the binary vector representation of the received symbol $r_j$. Decompose the received vector $r$ into $\tau$ vectors of length $n^2$ over GF(2), $r_{b_1} = (r_{0,b_1}, r_{1,b_1}, \ldots, r_{n^2-1,b_1})$, $0 \leq i < \tau$. These $\tau$ binary vectors are called the constituent received vectors of $r$. Suppose the received vector $r$ is free of errors. Then, $r = c^{\tau \times \tau}_{casc}$ and $H_{BCH,LDPC}r^T = 0$, and since $H_{BCH,LDPC}$ is binary, $H_{BCH,LDPC}r^T = 0$ for $0 \leq i < \tau$. Based on this fact, we can decode $r$ by decoding its $\tau$ binary constituent vectors. The decoding of each binary constituent received vector based on the binary BCH-LDPC matrix $H_{BCH,LDPC}$ is carried out with a binary ISDD algorithm, such as the SPA or the MSA. Thus, decoding a received vector over GF(2$^\tau$) can be implemented by decoding $\tau$ binary vectors. This significantly reduces the decoding complexity.

The decoding process at the receiver side consists of the following six steps:

- **Dec-1.** Decompose the received vector $r$ into $\tau$ binary constituent vectors. Decode each binary constituent vector of $r$ by using a binary ISDD algorithm based on
the binary matrix $H_{BCH,LDPC}$. Combine the $\tau$ decoded binary constituent vectors to form a decoded vector $\tilde{c}_{\text{casc}}^{n,\tau} = (\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{n-1})$ over GF$(2^\tau)$ of length $n^\tau$ which is the estimate of the transmitted codeword $c_{\text{casc}}^{n,\tau}$ formed by the encoder in Step Enc-5.

- **Dec-2.** Take the inverse GFT of the decoded vector $\tilde{c}_{\text{casc}}^{n,\tau}$ to obtain an estimate $\tilde{c}_{\text{casc}}^{n,\tau}$ of the interleaved cascaded codeword $c_{\text{casc}}^{n,\tau}$ formed by the encoder at Step Enc-4.

- **Dec-3.** Apply the inverse permutation $\pi_{\text{col}}^{-1}$ to $\tilde{c}_{\text{casc}}^{n,\tau}$, i.e., de-interleave $\tilde{c}_{\text{casc}}^{n,\tau}$, to obtain a vector $\tilde{c}_{\text{casc}}^{n,\tau}$ of $n$ sequences, each of length $n$, which is an estimate of $c_{\text{casc}}^{n,\tau}$ formed by the encoder at Step Enc-3.

- **Dec-4.** Decompose $\tilde{c}_k$, $0 \leq k < n$, into $\tau$ binary vectors $\tilde{c}_{0,k}, \tilde{c}_{1,k}, \ldots, \tilde{c}_{\tau-1,k}$ which are the estimates of the $\tau$ codewords $c_{0,k}, c_{1,k}, \ldots, c_{\tau-1,k}$ in $C_{BCH}$ formed at Step Enc-2.

- **Dec-5.** Permute $\tilde{c}_{i,k}$, for $0 \leq i < \tau$ and $1 \leq k < n$, using the inverse Hadamard-permutation $\pi_{\text{col}}^{-1}$ to obtain an estimate $\tilde{v}_{i,k}$ for the codeword $\tilde{v}_{i,k}$ in $C_{BCH}$ formed by the encoder at Step Enc-1. If $\tilde{v}_{i,k}$ is a codeword in $C_{BCH}$ for $0 \leq i < \tau$ and $1 \leq k < n$, then a message $M_{i,k}$ can be deduced as an estimate of the message $c_{i,k}$. If $\tilde{v}_{i,k}$ is a codeword in $C_{BCH}$, then a message $M_{i,0}$ can be deduced as an estimate of the message $c_{i,0}$.

- **Dec-6.** (Optional step): Apply the BM-HDDA to decode $\tilde{v}_{i,k}$, for $0 \leq i < \tau$, $1 \leq k < n$, based on the base BCH code $C_{BCH}$ to correct errors if detected.

At Step Dec-1, the $\tau$ binary constituent vectors of a received vector $r$ can be decoded in three modes: (1) decode them in parallel by using $\tau$ identical binary ISDD-decoders; (2) decode them in serial, one at a time, using a single ISDD-decoder; and (3) decode them in parallel-parallel using $s < \tau$ ISDD-decoders. Grouping operation is performed when and only when all the binary constituent received vectors are successfully decoded. Before all the binary constituent received vectors are successfully decoded, those that have been successfully decoded are saved in the memory, and the rest are continuously processed until all of them are successfully decoded and stored in the memory as a $\tau \times n^\tau$ array. At this point, grouping the $\tau$ decoded binary vectors is completed. If not all of the $\tau$ binary received constituent vectors are successfully decoded by the time when a preset maximum number of ISDD-iterations for all the ISDD-decoders is reached, a decoding failure is declared.

The BM-HDDA option at Step Dec-6 is not available for $k = 0$ since the code $C_{BCH}^{(0)}$ has minimum distance 2. However, if we use the modified encoding for $C_{BCH}^{(0)}$ as described at the end of the last section, then the BM-HDDA can be applied to decode the decoded words, $\tilde{v}_{i,0}, \tilde{v}_{i,1}, \ldots, \tilde{v}_{i,\tau-1}$. For each word $\tilde{v}_{i,0}$, $0 \leq i < \tau$, we remove the last symbol which corresponds to the overall parity-check symbol of a codeword in $C_{BCH}^{(0)}$ and add a zero information symbol at the first position. This modified word gives a codeword in $C_{BCH}$ if it is error free; otherwise, we can decode it based on $C_{BCH}$ using the BM-HDDA.

### B. An ISDD Algorithm

Let $I_{\text{max}}$ denote the preset maximum number of ISDD-iterations to be performed for decoding each of the $\tau$ binary received constituent vectors $r_{b,i}$, $0 \leq i < \tau$. For $0 \leq l \leq I_{\text{max}}$ and $0 \leq i < \tau$, let $r_{b,i}^{(l)}$ denote the updated hard-decision of $r_{b,i}$ at the end of the $l$-th iteration. For $l = 0$, $r_{b,i}^{(0)} = r_{b,i}$, $0 \leq i < \tau$. The ISDD-decoding of a received vector $r$ is carried out in the following four steps:

1. **ISDD-1.** Initialization: Set $l = 0$ and log-likelihood ratios (LLRs) for the received code bits of all the $\tau$ constituent received vectors based on the channel output detector.

2. **ISDD-2.** Perform the $l$-th ISDD-iteration to update the LLRs of the bits in $r_{b,i}$, $0 \leq i < \tau$ for which $H_{BCH,LDPC}(r_{b,i}^{(l)})^T \neq 0$. Compute $r_{b,i}^{(l)}$.

3. **ISDD-3.** Calculate $s_{b,i} = H_{BCH,LDPC}(r_{b,i}^{(l)})^T$, $0 \leq i < \tau$. If $s_{b,i} = 0$ for all $0 \leq i < \tau$, go to Step ISDD-4. If not, save $r_{b,i}^{(l)}$ for which $s_{b,i} = 0$. If $l = I_{\text{max}}$, declare a decoding failure; otherwise, set $l \leftarrow l + 1$ and go to Step ISDD-2.

4. **ISDD-4.** Stop ISDD-decoding and group the $\tau$ decoded binary vectors into an estimate of the transmitted codeword $c_{\text{casc}}^{n,\tau}$.

Since the ISDD of the above decoding scheme is performed in the GFT-domain, we call it the *GFT-ISDD scheme*. The most important feature of the proposed coding scheme is that the decoding is performed on a *collection* of received codewords *jointly*. During the decoding process, the reliability information of each decoded codeword is shared by the others to enhance the *overall reliability* of all the decoded codewords. This *joint-decoding and information sharing* may result in an error performance per decoded codeword better than the error performance of a received codeword decoded *individually* using the MLD. This will be demonstrated in several examples given in the later sections. This gain over the MLD is referred to as a *joint-decoding gain*. For a long code, the joint-decoding gain can be large. However, for a long code, computing its MLD performance is practically impossible. In this case, we will use the *union bound* (UB) on its MLD performance [4], [12], denoted by UB-MLD, for comparison. For large SNR, the UB is very tight.

### C. Measures of the Error Performance of the GFT-ISDD

In measuring the error performance of the cascaded code $C_{BCH,\text{casc}}$ of the base BCH code $C_{BCH}$ at the output of the BCH-LDPC decoder, we use frame error probability (i.e., the probability that a frame of $n \tau$ codewords is decoded incorrectly) in terms of frame error rate (FER), denoted by $P_{\text{FER}}$. After de-interleaving of the decoded frame and performing the inverse of the Hadamard-permutation on the $n$ symbols of each decoded codeword, the block error probability (i.e., the probability that a codeword in a frame is decoded
incorrectly) in terms of block error rate (BLER), denoted by $P_{BLER}$, is computed. The block error probability $P_{BLER}$ of a codeword in a decoded frame can be computed from the frame error probability $P_{FER}$. Let $\lambda$ be the average number of codewords in a decoded frame which are incorrectly decoded. Then, the error probability $P_{BLER}$ of a decoded codeword is $P_{BLER} = \lambda/(n\tau)P_{FER}$. It is clear that the frame error probability $P_{FER}$ is an upper bound on the block error probability $P_{BLER}$ of a decoded codeword.

In simulations, we found that the gap between these two error probabilities is very small corresponding to a difference of one-tenth of a dB in SNR. This reflects the fact that if a frame is not decoded correctly, then most of the $n\tau$ codewords are not decoded correctly. Suppose we transmit $n\tau$ codewords in the base code $C_{BCH}$ independently and decode them individually using a certain decoding algorithm. Let $P_{BLER}^*$ be the block error probability of a single decoded codeword. Then, the total error probability, denoted by $P_{total}^*$, of $n\tau$ individually decoded codewords (i.e., the probability that at least one of the $n\tau$ decoded codewords is incorrect) is at least $P_{BLER}^*$. When we compare the error performance of the cascaded BCH code constructed from the base BCH code $C_{BCH}$ using the above proposed decoding scheme with the error performance of the base BCH code $C_{BCH}$ with other decoding algorithms or schemes in which each received vector is decoded individually, we compare $P_{BLER}$ with $P_{BLER}^*$.

D. Decoding Complexity

Any binary iterative soft-decision BP-decoding algorithm can be used in conjunction with the GFT-ISDD scheme to decode the $2^\tau$-ary BCH-LDPC code $C_{BCH-LDPC}$ in binary based on its binary $mn \times n^\tau$ parity-check matrix $H_{BCH-LDPC}$. The two commonly used iterative BP-algorithms for decoding LDPC codes are the SPA [9] and the MSA [21]. Both algorithms require real number computations including multiplications, additions, and comparisons. The MSA is a simplified (or approximated) version of the SPA, which requires mainly real number additions and comparisons, but much less multiplications which are required when scaling $[22]$ is performed. Since the MSA is an approximated version of the SPA, decoding an LDPC code with the MSA results in some performance loss. However, if optimal scaling is used in updating LLRs of the received symbols, the MSA may perform just as well as (or very close to) the SPA. For this reason, the MSA or its variations are most commonly used in practice.

Suppose we combine the MSA with the GFT-ISDD scheme in decoding the $2^\tau$-ary BCH-LDPC code $C_{BCH-LDPC}$ based on the binary parity-check matrix $H_{BCH-LDPC}$. We denote such a combination as the GFT-ISDD/MSA. In computing the complexity of the GFT-ISDD/MSA, we only count the number of real number comparisons and additions. For a scaled MSA, each check node (CN) requires only 2 real number multiplications, which are a very small part of the total computations. Thus, we ignore the real number multiplications when estimating the computational complexity of the GFT-ISDD/MSA. When we say a real number computation, it means either a real number addition or comparison.

Suppose we use $\tau$ MSA decoders to decode the $\tau$ binary constituent received vectors in parallel. Decoding a BCH-LDPC code $C_{BCH-LDPC}$ with the GFT-ISDD/MSA, the total number of real number computations required to update a $2^\tau$-ary received vector (or very close to) the SP A. For this reason, the MSA or its loss. However, if optimal scaling is used in updating LLRs of calculations which are required when scaling [22] is performed. Since approximated) version of the SP A, which requires mainly real number computations including multiplications and additions, we may use the GFT-ISDD/MSA converges very fast. Setting $I_{max}$ to 5 gives an error performance only 0.2 dB away from the error performance obtained by setting $I_{max}$ to 50.

VII. RATE-COMPATIBLE GFT-BCH-LDPC CODES

In some applications, we may want to construct a family of codes of the same nature (or the same structure) with different rates which can be encoded and decoded with the same encoder and the same decoder. Codes in such a family are said to be rate-compatible. In this section, we present a method to construct a family of rate-compatible GFT-BCH-LDPC codes based on a given BCH code of prime length. We start with an $(n, n-m)$ BCH code $C_{BCH}$ over $GF(2)$ of prime length $n$ as the base code for the construction of a GFT-ICC-BCH code $C_{π,λ}^{ICC}$ (or $C_{π,λ}^{ICC-casc}$) of length $n^\tau$. Using the encoding scheme presented in Section [22] for $0 \leq k < n$, let $\lambda_k$ be a nonnegative integer such that $0 \leq \lambda_k < \tau$. In the first step of encoding, $\text{Enc-1}$, we set $\tau - \lambda_k$ of the $\tau$ messages into zero messages. Encode the $\lambda_k$ nonzero messages into $\lambda_k$ nonzero codewords in $C_{BCH}$. Then, we follow the rest of the encoding steps. At the end of Step $\text{Enc-5}$, we obtain a codeword $c_{π,λ}^{ICC,\tau}$ over $GF(2^\tau)$ of length $n^2$ in the BCH-domain, which contains $(n-1)\lambda_0 + (n-m)(\lambda_1 + \lambda_2 + \ldots + \lambda_{n-1})$ information symbols. Therefore, the rate of the resultant GFT-BCH-LDPC code $C_{π,λ}^{ICC}$ of $(n-1)\lambda_0 + (n-m)(\lambda_1 + \lambda_2 + \ldots + \lambda_{n-1})/n^\tau$ which is less than or equal to the rate of $C_{BCH}$ (equality holds if and only if $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = \tau$). As an example, if we set $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 1$, the code $C_{π,λ}^{ICC,\tau}$ is a GFT-BCH-LDPC code over $GF(2^\tau)$ with rate $(n-1)(n - m + 1)/n^\tau$. The code $C_{π,λ}^{ICC,\tau}$ is a subcode of $C_{π,λ}^{ICC}$. If $C_{π,λ}^{ICC}$ is a proper subcode of $C_{π,λ}^{ICC,\tau}$, we call it a descendant code of $C_{π,λ}^{ICC}$. Differ-
ent choices of the set \( \{ \lambda_0, \lambda_1, \ldots, \lambda_{n-1} \} \) of parameters result in different descendant codes of \( C_{\text{BCH, cas}} \) with the same length \( n^2 \) but different rates. The code \( C_{\text{BCH, cas}} \) is referred to as the mother code. The subscript “des” in \( C_{\pi, \Sigma, \text{cas}, \text{des}} \) and \( C_{\text{case, des}} \) stands for “descendant”. The mother code \( C_{\text{BCH, cas}} \) and its descendants have the same structure. Two codewords from two different descendant codes of \( C_{\text{BCH, cas}} \) consist of two different collections of codewords from the base BCH code \( C_{\text{BCH}} \).

From the above construction of descendant codes of \( C_{\text{BCH, cas}} \), it is clear that all the descendant codes can be encoded and decoded with the same encoder and the same decoder as the mother code \( C_{\text{BCH, cas}} \). The decoding of a descendant code \( C_{\pi, \Sigma, \text{cas}, \text{des}} \) of \( C_{\text{BCH, cas}} \) takes exactly the same steps as those for decoding the mother code \( C_{\pi, \Sigma, \text{BCH, cas}} \) using the parity-check array \( H_{\text{BCH, LDPC}} \).

What we have shown above is that based on a given BCH code \( C_{\text{BCH}} \) of prime length \( n \), we can construct a family of rate-compatible GFT-BCH-LDPC codes of the same length \( n^2 \) but different rates. In some communication applications, rate-compatible codes of different lengths may be required. In this case, we can construct rate-compatible GFT-BCH-LDPC codes of different lengths using the above construction method in conjunction with shortening the base BCH code and its Hadamard-equivalents to different degrees (using the cyclic code shortening method [7]).

### VIII. EXAMPLES

In this section, we give three examples to demonstrate the error performances of three BCH codes of prime lengths decoded in the GFT-domain through their associated BCH-LDPC codes using the GFT-ISDD/MSA. We assume transmission over the binary-input AWGN channel using BPSK signaling. In simulations, we found that whether or not the BM-HDDA is performed on the base BCH code \( C_{\text{BCH}} \) at the final stage of decoding has very little effect on performance, which does not exceed one-tenth of a dB in SNR. This can be attributed to the fact that the decoding of the BCH-LDPC code \( C_{\text{BCH, LDPC}} \) using the GFT-ISDD/MSA corrects most of the errors, and in case it fails then the residual errors exceed the error-correcting capability of the base BCH code \( C_{\text{BCH}} \). For this reason, we do not perform the BM-HDDA on \( C_{\text{BCH}} \).

**Example 1.** Let \( \tau = 5 \) and \( \alpha \) be a primitive element of the field \( \text{GF}(2^5) \). Note that \( 2^5 - 1 = 31 \) is a prime. Set \( n = 31, t = 3, \) and \( \beta = \alpha \). Consider the triple-error-correcting (31, 16) BCH code \( C_{\text{BCH}} \) whose generator polynomial has \( \beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6 \) and their conjugates as roots, a total of 15 roots. This code has minimum distance 7. Using the 15 roots of the generator polynomial of this BCH code, we form a \( 15 \times 31 \) parity-check matrix \( H_{\text{BCH, LDPC}} \) over \( \text{GF}(2^5) \). The null space over \( \text{GF}(2^5) \) of the BCH-LDPC matrix \( H_{\text{BCH, LDPC}} \) gives a 32-ary BCH-LDPC code \( C_{\text{BCH, LDPC}} \) over GF(2^5) associated with the base BCH code \( C_{\text{BCH}} \). Since \( H_{\text{BCH, LDPC}} \) satisfies the RC-constraint and has column weight 15, the BCH-LDPC code \( C_{\text{BCH, LDPC}} \) associated with the base BCH code \( C_{\text{BCH}} \) has minimum distance at least 16 which is more than twice of the minimum distance of the base code \( C_{\text{BCH}} \). Decode the cascaded BCH code \( C_{\text{BCH, cas}} \) using the GFT-ISDD/MSA based on the parity-check array \( H_{\text{BCH, LDPC}} \). At the LDPC decoding stage, the received vector over GF(2^5) is decomposed into 5 binary constituent received vectors and each of these 5 binary vectors is decoded with 50 iterations of the MSA. The block error performance of the cascaded code \( C_{\text{BCH, cas}} \) of the (31, 16) BCH code \( C_{\text{BCH}} \) is shown in Fig. 2 labeled by BCH-FT-50, together with the block error performances of the base BCH code \( C_{\text{BCH}} \) decoded using the BM-HDDA and the MLD.

At a BLER of \( 10^{-5} \), the GFT-ISDD/MSA achieves about 3.8 dB coding gain over the BM-HDDA and 1.4 dB joint-decoding gain over the MLD of the code which decodes each received codeword individually. From Fig. 2 we see that the MLD and the UB-MLD curves basically overlap with each other. Hence, the UB-MLD is a very tight bound on the MLD. For the BLER lower than \( 10^{-7} \), the GFT-ISDD/MSA even starts to improve upon the sphere packing bound (SPB) which decodes each received codeword individually.

The average number of real number computations required to update a single codeword in \( C_{\text{BCH}} \) with the GFT-ISDD/MSA per iteration is 1440. With 50 iterations, the average number of real number computations required to decode a single codeword in \( C_{\text{BCH}} \) is at most 72000.

![Fig. 1. Block error performances of the (31, 16) BCH code in Example 1 decoded by the GFT-ISDD/MSA, the BM-HDDA, and the MLD.](image)

**Example 2.** Let \( \tau = 7 \) and \( \alpha \) be a primitive element of the field \( \text{GF}(2^7) \). Note that \( 2^7 - 1 = 127 \) is a prime. Set \( n =
127, $t = 2$, and $\beta = \alpha$. The double-error-correcting BCH code $C_{\text{BCH}}$ constructed from the field $\text{GF}(2^7)$ is a $(127, 113)$ cyclic code of prime length 127 with rate 0.89. The generator polynomial of this code has $\beta, \beta^2, \beta^3, \beta^4$ and their conjugates as roots, a total of 14 roots. Based on the 14 roots of its generator polynomial, we construct a $14 \times 127$ BCH-LDPC array $H_{\text{BCH},\text{LDPC}}$ of binary CPMs of size $127 \times 127$. It is an RC-constrained binary matrix with column weight 14 and row weight 127. Even though the $(127, 113)$ base BCH code has minimum distance 5, its associated 2-power BCH-LDPC code $C_{\text{BCH,LDPC}}$ given by the null space over $\text{GF}(2^7)$ of the BCH-LDPC matrix $H_{\text{BCH,LDPC}}$ has minimum distance at least 15.

The block error performances of the $(127, 113)$ BCH code decoded in the GFT-domain with the GFT-ISDD/MSA using 5, 10, and 50 iterations are shown in Fig. 2 labeled by BCH-FT-5, BCH-FT-10, and BCH-FT-50, respectively. We see that the GFT-ISDD/MSA decoding of this code converges very fast and with 10 iterations of the MSA, at a BLER of $10^{-7}$, it achieves 3.7 dB coding gain over the BM-HDDA and 1.8 dB joint-decoding gain over the UB-MLD of the code. From Fig. 2 we see that at BLER of $10^{-7}$, the GFT-ISDD/MSA decoding of the cascaded $(127, 113)$ BCH code with 50 iterations achieves more than 1 dB coding gain over the ML of the $(127, 113)$ BCH code with each codeword decoded individually. Furthermore, we see that for SNR greater than 5 dB, the MLD performance and the UB on the ML performance of the BCH code overlap with each other, i.e., the UB-MLD is tight.

If we set the maximum number $I_{\text{max}}$ of decoding iterations to be performed with the GFT-ISDD/MSA to 10. Then, the average number of real number computations required to decode a single codeword in $C_{\text{BCH}}$ is at most $52640$.

Example 3. In this example, we show that even a cyclic Hamming code (a single-error-correcting BCH code) 7 of prime length decoded with the GFT-ISDD/MSA can achieve a very impressive error performance. Again, let $\text{GF}(2^7)$ be the field for code construction. Set $n = 127$ and $t = 1$. Consider the $(127, 120)$ cyclic Hamming code $C_{\text{Ham}}$ of rate 0.945 and minimum distance 3 generated by a primitive polynomial of degree 7 over $\text{GF}(2)$, $q_{\text{Ham}}(X) = X^7 + X^3 + 1$. The subscript “Ham” in $C_{\text{Ham}}$ and $q_{\text{Ham}}(X)$ stands for “Hamming”. This polynomial has a primitive root $\beta$ of $\text{GF}(2^7)$ and its 6 conjugates as roots which are $\beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32}$, and $\beta^{64}$. Using these 7 roots, we form a $7 \times 127$ Hamming-LDPC array $H_{\text{Ham,LDPC}}$ of $127 \times 127$ binary CPMs, which is an $889 \times 16129$ binary matrix with column and row weights 7 and 127, respectively. Decode the $(127, 120)$ Hamming code $C_{\text{Ham}}$ in the GFT-domain using the GFT-ISDD/MSA based on the Hamming-LDPC matrix $H_{\text{Ham,LDPC}}$.

The block error performances of the code decoded with 5, 10, and 50 iterations of the MSA are shown in Fig. 3 labeled by Hamming-FT-5, Hamming-FT-10, and Hamming-FT-50, respectively. We see that the decoding converges fast. The performance gap between 10 and 50 iterations is about 0.1 dB. At a BLER of $10^{-8}$, the GFT-ISDD/MSA with 50 iterations achieves 5.2 dB coding gain over the BM-HDDA and 3.7 dB joint-decoding gain over the MLD, a very large joint-decoding gain over the MLD. Even with 5 iterations of the MSA, the GFT-ISDD/MSA outperforms the BM-HDDA and the MLD by 4.4 dB and 2.9 dB gains at a BLER of $10^{-7}$, respectively. It even improves upon the SPB applied to each codeword decoded individually by 0.7 dB.

If we set the maximum number $I_{\text{max}}$ of decoding iterations to be performed with the GFT-ISDD/MSA to 10. Then, the average number of real number computations required to decode a single codeword in $C_{\text{Ham}}$ (or $C_{\text{BCH}}$) is at most 25320.

Using the $(127, 120)$ Hamming code $C_{\text{Ham}}$, we can construct a family of rate-compatible GFT-Hamming-LDPC codes of length 16129 with a wide range of rates from 0.945 down. Suppose we choose $\lambda_0 = 0$ and $\lambda_1 = \lambda_2 = \ldots = \lambda_7 = 7$, i.e., we set all the 7 messages $M_{i,0}, 0 \leq i < 7$, to zero messages in the encoding step Enc-1. With those choices of parameters, we can construct a GFT-Hamming-LDPC code $C_{\text{Ham,casc,des}}$ of length 16129 with rate 0.937. If we choose $\lambda_0 = \lambda_1 = \ldots = \lambda_6 = 3$, then the resultant GFT-Hamming-LDPC code $C_{\text{Ham,casc,des}}$ has length 16129 and rate 0.405. Setting $\lambda_0 = \lambda_1 = \ldots = \lambda_6 = 1$, we obtain a GFT-Hamming-LDPC code of length 16129 with rate 0.135. So the rates of the rate-compatible GFT-Hamming-LDPC codes constructed based on the $(127, 120)$ Hamming code range from very low to very high. All these codes can be decoded with the same decoder based on the 889 $\times$ 16129 parity-check matrix $H_{\text{Ham,LDPC}}$ given above.

IX. CONCLUSION AND REMARKS

In this paper, we presented a novel and effective coding scheme for encoding and iterative soft-decision decoding of
binary BCH codes of prime lengths in the GFT-domain. The keys to this coding scheme are joint-encoding and joint-decoding of a collection of codewords from a BCH code by applying: (1) Hadamard-permutations (or their inverses) to symbols of the encoded or decoded codewords in the collection; (2) combining (or de-combining) encoded or decoded codewords into a codeword over a larger field; (3) interleaving (or de-interleaving) encoded or decoded codewords; and (4) taking GFTs (or their inverses) on the collection of interleaved encoded or decoded codewords.

The joint-decoding and information sharing may result in an error performance per decoded codeword better than the error performance of a received codeword decoded individually using the MLD as demonstrated in examples. Another important feature of the proposed GFT-ISDD scheme is that the decoding of a nonbinary received vector is carried out in binary based on a binary LDPC matrix. The binary iterative decoding can be performed efficiently and reduces the decoding complexity significantly. The proposed coding scheme not only requires low decoding complexity, but also yields superior performance as demonstrated by examples. Since the parity-check matrix $H_{BCH,LDPC}$ of the BCH-LDPC code $C_{BCH,LDPC}$ is an array of binary CPMs, the hardware complexity of the GFT-ISDD-decoder of a BCH-LDPC code can be significantly reduced by using the revolving iterative decoding scheme proposed in [15]. Another feature of the proposed coding scheme is that before applying the ISDD to the binary constituent received vectors of a received vector $r$, a simple decoding scheme, such as hit-flipping or one-step majority-logic decoding, can be applied to decode the binary constituent received vectors of $r$. If $r$ contains only a small number of errors, the adopted simple decoding scheme can correct the errors to avoid using the more complicated ISDD. This reduces both the decoding latency and the decoding complexity. Such a feature is desired for flash memory and optical communication systems.

In this paper, we also presented a method for constructing a family of rate-compatible GFT-BCH-LDPC codes of the same length with different rates based on a given BCH code of prime length. All the codes in the same family can be encoded and decoded with the same encoder and the same decoder.

The proposed coding scheme can be applied to any cyclic code, not necessary a BCH code, of prime length.

REFERENCES