Iterative Decoding Algorithms for a Class of Non-Binary Two-Step Majority-Logic Decodable Cyclic Codes

Hsiu-Chi Chang and Hsie-Chia Chang

***Abstract*—This paper presents two iterative decoding algorithms for a class of non-binary two-step majority-logic (NB-TS-MLG) decodable cyclic codes. A partial parallel decoding scheme is also introduced to provide a balanced trade-off between decoding speed and storage requirements. Unlike non-binary one-step MLG decodable cyclic codes, the Tanner graphs of which are 4-cycle-free, NB-TS-MLG decodable cyclic codes contain a large number of short cycles of length 4, which tend to degrade decoding performance. The proposed algorithms utilize the orthogonal structure of the parity-check matrices of the codes to resolve the degrading effects of the short cycles of length 4. Simulation results demonstrate that the NB-TS-MLG decodable cyclic codes decoded with the proposed algorithms offer coding gains as much as 2.5 dB over Reed-Solomon codes of the same lengths and rates decoded with either hard-decision or algebraic soft decision decoding.**

***Index Terms*—Extended min-sum algorithm, majority-logic decoding, non-binary LDPC codes, cyclic codes.**

I. Introduction

FINITE geometry codes received considerable attention in the late 1960s and 1970s [1]–[3]. These codes form an important class of cyclic codes, which can be systematically encoded with linear shift registers and decoded with majority-logic decoding (MLGD) [4]. Based on finite geometries, there are two types of cyclic codes: one-step and multi-step MLG decodable. One-step MLG decodable cyclic codes were rediscovered in 2001 [5] as finite geometry low-density parity-check (FG-LDPC) codes with 4-cycle-free Tanner graphs [6]. Long FG-LDPC codes provide error correction performance approaching to Shannon's theoretical limit [7] when decoded using belief propagation algorithms, such as the *sum-product algorithm* [8] and the *min-sum algorithm* [9]. In contrast, numerous short cycles of length 4 involved in multi-step MLG decodable cyclic codes limit the effectiveness of the standard belief propagation algorithm [10]. Consequently, only a small amount of coding gain is achieved at a considerable increment in decoding complexity. Efforts to overcome this key disadvantage have led to the development of efficient iterative decoding algorithms, which utilize the orthogonal structure of the parity-check matrices of the two-step MLG (TS-MLG) decodable cyclic codes [10], [11].

Binary LDPC codes typically demonstrate weakness in error performance for short and moderate code lengths [12]. In these cases, non-binary LDPC (NB-LDPC) codes in higher order Galois fields provide excellent alternatives. NB-LDPC codes constructed based on finite geometries have been discussed in [13], [14]. These codes are non-binary one-step MLG decodable. The associated Tanner graphs of the parity-check matrices of the codes are 4-cycle free, which enables NB-LDPC codes perform very well over the *additive white Gaussian noise* (AWGN) channel using standard belief propagation algorithms such as FFT-QSPA [12] or EMS [15] algorithm. However, the development of an efficient belief propagation algorithm for decoding non-binary multi-step MLG decodable cyclic codes has yet to be achieved. In this paper, a subclass of NB-TS-MLG decodable cyclic codes is presented. From our simulation studies, standard belief propagation algorithm for decoding NB-TS-MLG decodable cyclic codes is not effective due to the large number of short cycles of length 4. These short cycles produce decoding correlations after a few decoding iterations, thereby preventing convergence to *maximum-likelihood* decoding. As a result, coding gains are marginal and the speed of convergence is slow. To overcome this major drawback, we modify standard belief propagation by introducing the geometric structure of the parity-check matrices of the codes [4]. Two efficient decoding algorithms based on the orthogonal structure of the parity-check matrices of the codes are proposed to reduce or eliminate the degrading effects of short cycles of length 4. Furthermore, the orthogonal structure of NB-TS-MLG decodable cyclic codes allows a decomposition on the parity-check matrices, resulting in a partial parallel decoding scheme.

FFT-QSPA presents the best performance among the belief propagation algorithms developed for decoding NB-LDPC; however, complex operations, such as multiplication and division tend to increase decoding complexity. The EMS algorithm overcomes this issue by utilizing the log-domain operations that turn multiplications into log-domain additions and avoid divisions. In this paper, we propose an algorithm called *iterative two-step EMS* (ITS-EMS) by modifying the standard EMS algorithm. The NB-TS-MLG decodable cyclic codes decoded with the proposed ITS-EMS achieve as much as 2.5 dB coding gain over Reed-Solomon (RS) codes of the same lengths and rates decoded using either the hard-decision Berlekamp-Massey (HD-BM) algorithm [4] or the algebraic soft-decision

Manuscript received August 16, 2013; revised December 17, 2013, February 11, 2014, March 16, 2014, and April 4, 2014; accepted April 16, 2014. Date of publication April 25, 2014; date of current version June 18, 2014. This work was supported by the NSC, Taiwan, under Contract NSC 101-2628-E-009-013-MY3. The associate editor coordinating the review of this paper and approving it for publication was L. Dolecek.

The authors are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: jasper.ee94g@nctu.edu.tw; hcchang@mail.nctu.edu.tw).

Digital Object Identifier 10.1109/TCOMM.2014.2320508

Koetter-Vardy (ASD-KV) algorithm [16]. Unfortunately, ITS-EMS suffers from high computational complexity because many of its computations involve real numbers. A low complexity iterative message passing decoding algorithm was developed previously to decode non-binary one-step MLG decodable cyclic codes [17], called *iterative soft reliability-based MLGD* (ISRB-MLGD) algorithm. We further generalize the ISRB-MLGD algorithm as *iterative reliability two-step MLGD* (IRTS-MLGD) algorithm to decode the NB-TS-MLG decodable cyclic codes. The IRTS-MLGD requires far lower computational complexity by employing only finite field and integer operations, compared to the ITS-EMS using computations in real numbers. Moreover, the decoding process is different between the ISRB-MLGD and the IRTS-MLGD. The ISRB-MLGD uses a fully parallel decoding scheme; instead, the IRTS-MLGD employs a partial parallel decoding scheme. The partial parallel decoding scheme can be generalized for decoding the binary TS-MLG decodable cyclic codes presented in [10], resulting in a more balanced trade-off between decoding speed and memory usage. In addition, we compare the error performances of ITS-EMS decoding with the NB-TS-MLG decodable cyclic codes and standard EMS decoding with the one-step MLG decodable NB-LDPC codes constructed based on Euclidean geometries via matrix dispersion [14], [18]. Simulation results show that in a small number of decoding iterations, the NB-TS-MLG decodable cyclic codes outperform one-step MLG decodable NB-LDPC codes.

The remainder of this paper is organized as follows. Section II briefly introduces a subclass of NB-TS-MLG decodable cyclic codes and the hard-decision non-binary two-step MLGD (NB-TS-MLGD) algorithm. The proposed ITS-EMS is introduced in Section III, together with a parity-check matrix decomposition for partial parallel decoding. We also discuss the computational complexity of ITS-EMS and investigate its memory requirements. Section IV gives the low complexity IRTS-MLGD algorithm and evaluates its computational complexity. Section V concludes the paper.

II. Class of NB-TS-MLG Decodable Cyclic Codes

In this section, we consider a special class of TS-MLG decodable cyclic code, referred to as two-fold Euclidean geometry (EG) codes. This subclass of binary MLG decodable cyclic codes was constructed based on Euclidean geometries by Lin [19] in 1973, called *multifold Euclidean geometry* codes. We generalize the binary two-fold EG codes to the non-binary cases known as NB-two-fold EG (NB-TF-EG) codes, then investigate this special case of NB-TS-MLG decodable cyclic codes.

A. Code Construction

Consider a d-dimensional Euclidean geometry EG(d,q) over the field GF(q), where q is a power of prime. The field GF(q^d) as an extension filed of the field GF(q) is a realization of EG(d,q). Let α be a primitive element of GF(q^d). Then the powers of α, $\alpha^{-\infty}=0$, $\alpha^0=1$, $\alpha, \ldots \alpha^{q^d-2}$, represent the q^d points of EG(d,q) and $\alpha^{-\infty}=0$ represents the origin of EG(d,q). Let EG$^*(d,q)$ be the subgeometry by removing the origin and all the lines passing through the origin in EG(d,q). Let $n=q^d-1$. There are n non-origin points and $J_0=n(q^{d-1}-1)/(q-1)$ lines not passing through origin in EG$^*(d,q)$ [4]. Let $L=\{\alpha^{j_1},\alpha^{j_2},\ldots,\alpha^{j_q}\}$ be a line in EG$^*(d,q)$ comprising points $\alpha^{j_1},\alpha^{j_2},\ldots,\alpha^{j_q}$, where $0\le j_1,j_2,\ldots,j_q<q^d-1$. Let $\mathbf{v}_L$ be the (q^d-1)-tuple over GF(q^d) as $\mathbf{v}_L=(v_0,v_1,\ldots,v_{q^d-2})$. The components in $\mathbf{v}_L$ correspond to the n non-origin points of EG$^*(d,q)$, where the j_1th, j_2th$,\ldots,j_q$th components are $v_{j_1}=\alpha^{j_1}$, $v_{j_2}=\alpha^{j_2}$, $v_{j_3}=\alpha^{j_3},\ldots,v_{j_q}=\alpha^{j_q}$ and other components are zero element in GF(q^d). This (q^d-1)-tuple $\mathbf{v}_L$ is called a q^d-ary incidence vector of line L. This vector $\mathbf{v}_L$ has q points, each point represents its location and value by the element of GF(q^d). Let $\mathbf{L}$ be a $J_0\times n$ matrix which is formed by the J_0 lines in EG$^*(d,q)$. Let $\mathbf{v}_{L_0},\mathbf{v}_{L_1},\ldots,\mathbf{v}_{L_{J_0-1}}$ be the rows of $\mathbf{L}$. Let $\mathbf{v}_{L_i}$ be the q^d-ary incidence vector of line L_i denoted as $\mathbf{v}_{L_i}=(v_{i,0},v_{i,1},\ldots,v_{i,n-1})$, where $0\le i<J_0$. For $0\le j<n$, we define $N_i=\{j:0\le j<n, v_{i,j}\ne 0\}$ and $M_j=\{i:0\le i<J_0, v_{i,j}\ne 0\}$. The indices in N_i denote the location of nonzero components in the ith row of $\mathbf{L}$. The indices in M_j denote the location of nonzero components in the jth column of $\mathbf{L}$. The Tanner graph [6] of matrix $\mathbf{L}$ has two disjoint classes of nodes: variable nodes (VN) and check nodes (CN). The jth VN corresponds to the jth q^d-ary received symbol in $\mathbf{L}$, while the ith CN corresponds to the ith row of $\mathbf{L}$. If $v_{i,j}\ne 0$, the jth VN is connected to the ith CN by an edge.

If a point is on a line in EG$^*(d,q)$, we say that the line passes through the point (or is orthogonal on the point). Every point in EG$^*(d,q)$ is intersected by $J_1=n/(q-1)-1$ lines. For the ith line L_i in EG$^*(d,q)$, where $0\le i<J_0$, it has $J_2=q^{d-1}-2$ parallel lines denoted as $L_{t,i}$, where $0\le t<J_2$. $\{L_i,L_{t,i}\}$ forms a (1,2)-frame which consists of $2q$ points in EG$^*(d,q)$. The corresponding q^d-ary incidence vector of $\{L_i,L_{t,i}\}$ is denoted as $\mathbf{v}_{L_i}+\mathbf{v}_{L_{t,i}}$, where $\mathbf{v}_{L_i}$ and $\mathbf{v}_{L_{t,i}}$ are two (q^d-1)-tuple over GF(q^d) without any points in common. Let $\{L_i,L_{0,i}\},\{L_i,L_{1,i}\},\ldots,\{L_i,L_{J_2-1,i}\}$ be the J_2 (1,2)-frames that intersect on line L_i. We say that these J_2 (1,2)-frames are orthogonal on L_i. There are a total of $m_r=n(q^{d-1}-1)(q^{d-1}-2)/2(q-1)$ (1,2)-frames F in EG$^*(d,q)$. These (1,2)-frames form a $m_r\times n$ matrix $\mathbf{H}$ over GF(q^d) with each row as a q^d-ary incidence vector of the (1,2)-frames in EG$^*(d,q)$. Then the null space of $\mathbf{H}$ gives a cyclic code of length n, referred to as NB-TF-EG code. The *generator polynomial* of a NB-TF-EG code can be derived as the following steps [13]. Each row of $\mathbf{H}$ is represented by a polynomial of degree q^d-2 or less over GF(q^d). Let $\mathbf{h}(X)$ be the *greatest common divisor* of the row polynomials of $\mathbf{H}$. Let $\mathbf{h}^*(X)$ be the *reciprocal polynomial* of $\mathbf{h}(X)$. The *generator polynomial* of a NB-TF-EG code is derived by $\mathbf{g}(X)=(X^n-1)/\mathbf{h}^*(X)$.

B. NB-TS-MLGD Algorithm

Consider GF(q^d) as the field on which to construct the NB Euclidean geometry. For simplicity of illustration, we consider $q^d=2^r$. Although this paper considers only the case for 2-powers, the codes and the decoding algorithms can be generalized to any prime-powers. Assume that transmission uses *binary phase-shift keying* (BPSK) or m-QAM over the AWGN channel with two-sided power spectral density $N_0/2$. We use $\mathbb{R}^r$ for BPSK and $\mathbb{R}^2$ for m-QAM. Let $\mathbf{u} = (u_0, u_1, u_2, \ldots, u_{n-1})$ be a transmitted n-tuple codeword of a NB-TF-EG code over GF(q^d). For $0 \le j < n$, the jth symbol u_j of $\mathbf{u}$ can be converted into a sequence of $\mathrm{r} = \log_2(q^d)$ bits and denoted as $u_j = (u_{j,0}, u_{j,1}, \ldots, u_{j,r-1})$ over GF(2). Let $\mathbf{z} = (z_0, z_1, z_2, \ldots, z_{n-1}) \in Z^n$ be the hard-decision received sequence, where Z is the received alphabet for a single NB-TF-EG symbol. For $0 \le j < n$, each component z_j of $\mathbf{z}$ is an element in GF(q^d). The hard-decision received sequence is a NB-TF-EG codeword if and only if $\mathbf{H}\mathbf{z}^T = 0$ (or the polynomial representation $\mathbf{z}(X)$ of $\mathbf{z}$ is divisible by the generator polynomial $\mathbf{g}(X)$).

The NB-TS-MLGD algorithm is generalized from the non-binary one-step MLGD algorithm [17]. Assume that α^j in EG$^*(d, q)$ is updated. The corresponding received symbol for α^j is z_j. Let z_j be the jth received symbol in $\mathbf{z}$ participating in L_i. Let $S(L_i)$ be the line-sum (or check-sum), which can be derived by the inner product of the non-zero element $v_{i,j}$ in $\mathbf{v}_{L_i}$ and the received symbol z_j in $\mathbf{z}$ as

$$S(L_i) = \sum_{j \in N_i} v_{i,j} z_j. \tag{1}$$

Consider a (1,2)-frame $F = \{L_i, L_{t,i}\}$ in EG$^*(d, q)$. The frame-sum of F denoted by $S(F) = S(L_i) + S(L_{t,i})$ is the inner product of $\mathbf{z}$ and the q^d-ary incidence vector of (1,2)-frame F comprising two lines L_i and $L_{t,i}$ in EG$^*(d, q)$. We omit the subscript t of $L_{t,i}$ for calculating $S(L_{t,i})$ by (1) since $L_{t,i}$ is also a line in EG$^*(d, q)$. Let L_u^j denote the J_1 lines passing through z_j, where $0 \le u < J_1$. The line-sum of L_u^j is denoted as $S(L_u^j)$. For $0 \le t < J_2$, J_2 lines are denoted as $L_{t,u}^j$ parallel to L_u^j. The line-sum of $L_{t,u}^j$ is denoted as $S(L_{t,u}^j)$. The first step of decoding is to decode $S(L_u^j)$ with the J_2 (1,2)-frames in EG$^*(d, q)$ orthogonal on L_u^j. Let $F^{j,u,t} = \{L_u^j, L_{t,u}^j\}$ be a (1,2)-frame in EG$^*(d, q)$ orthogonal on L_u^j. The frame-sum of $F^{j,u,t}$ is denoted as $S(F^{j,u,t}) = S(L_u^j) + S(L_{t,u}^j)$. Note that $S(L_u^j)$, $S(L_{t,u}^j)$ and $S(F^{j,u,t})$ are the elements in GF(q^d). The line-sum $S(L_{t,u}^j)$ of $S(F^{j,u,t})$ is the *extrinsic information* for decoding $S(L_u^j)$. A received symbol in $\mathbf{z}$ not contained in L_u^j can appear in at most one $L_{t,u}^j$. Thus, we can correctly decode the value of $S(L_u^j)$ from the J_2 $S(F^{j,u,t})$ orthogonal on $S(L_u^j)$ provided that no more than $\lfloor J_2/2 \rfloor$ symbol errors in $\mathbf{z}$. The second step is to decode z_j with J_1 $S(L_u^j)$ orthogonal on z_j. Any received symbols of $\mathbf{z}$ other than z_j can appear in at most one of these J_1 lines. The symbols orthogonal on z_j are the *extrinsic information* for z_j. Since $J_1 > J_2$, the value of z_j can be correctly determined if there are no more than $\lfloor J_2/2 \rfloor$ symbol errors in $\mathbf{z}$. This completes the decoding process of the NB-TS-MLGD for the NB-TS-MLG decodable cyclic codes.

III. Iterative Two-Step Extended Min-Sum Algorithm With Partial Parallel Decoding

Serial and parallel decoding algorithms have been developed for binary TS-MLG decodable cyclic codes [10]. If we consider hardware implementation, serial decoding algorithm has the advantage of requiring a simple decoding circuit at the cost of large number of decoding cycles. In contrast, parallel decoding has the advantage of fast decoding but requires hardware of greater complexity. A partial parallel decoding scheme can be a good trade-off between serial and parallel decoding with regard to decoding speed and hardware complexity.

A. Parity-Check Matrix Decomposition and Partial Parallel Decoding Scheme

In this subsection, we present a partial parallel decoding scheme via a decomposition on parity-check matrix. Unlike traditional method used to represent the parity-check matrix for a NB-TF-EG code with points on the column side and frames on the row side, we decompose the parity-check matrix into two parts. One contains q^d-ary incidence vectors representing the relationship between points and lines, and the other comprises binary incidence vectors describing the relationship between lines and frames. In the following, we illustrate the construction of these two matrices. Consider the d-dimensional Euclidean geometry EG$^*(d, q)$ over the GF(q^d). For $d = 2$, let $\beta = \alpha^{J_1+1}$. Then, $\{0, 1, \beta, \beta^2, \beta^3, \ldots, \beta^{J_2}\}$ form a subfield GF(q^{d-1}) of the field GF(q^d). Consider a *parallel bundle* P [4] in EG$^*(d, q)$ comprising lines $\{L, \beta^1 L, \ldots, \beta^{J_2} L\}$. The corresponding q^d-ary incidence vector is $\mathbf{v}_{P_L} = \{\mathbf{v}_L, \mathbf{v}_{\beta^1 L}, \ldots, \mathbf{v}_{\beta^{J_2} L}\}$. By multiplying P by α, we obtain $\alpha P = \{\alpha L, \alpha\beta^1 L, \ldots, \alpha\beta^{J_2} L\}$, where $\mathbf{v}_{\alpha P_L} = \{\mathbf{v}_{\alpha L}, \mathbf{v}_{\alpha\beta^1 L}, \ldots, \mathbf{v}_{\alpha\beta^{J_2} L}\}$ is its q^d-ary incidence vector. Each line in αP is the right cyclic shift of the line in P. The J_0 lines in EG$^*(d, q)$ can be divided into $J_3 = J_1 + 1$ groups of parallel bundles [4] and denoted as $\{P, \alpha P, \ldots, \alpha^{J_3 - 1} P\}$. Each group of parallel bundles comprises $J_4 = J_2 + 1$ lines. A $J_0 \times n$ matrix $\mathbf{L}_P$ can be formed by the q^d-ary incidence vectors of the parallel bundle of lines via $\{\mathbf{v}_{P_L}, \mathbf{v}_{\alpha P_L}, \ldots, \mathbf{v}_{\alpha^{J_3-1} P_L}\}$. Matrix $\mathbf{L}_P$ represents the relationship between points and lines with q^d-ary incidence vectors of lines as rows. This completes the first part of the decomposition of the parity-check matrix $\mathbf{H}$. The parallel bundle P has cyclic property; therefore, we only need to store the q^d-ary incidence vectors in $\mathbf{v}_{P_L}$ as the indices for iterative decoding. The q^d-ary incidence vectors of the other parallel bundles of lines can simply be derived by cyclically shifting the elements in $\mathbf{v}_{P_L}$ when the corresponding block is decoded. Next, we construct the matrix with binary incidence vectors. In each parallel bundle, $J_5 = (q^{d-1} - 1)(q^{d-1} - 2)/2$ different frames are formed by J_4 lines. Let $\mathbf{F} = \{F_0, F_1, \ldots, F_{J_3 - 1}\}$ be the frames constructed by $\mathbf{P} = \{P_0, P_1, \ldots, P_{J_3 - 1}\}$, where $P_0 = P$, $P_1 = \alpha P_0$ and so on. Consider the ath parallel bundle P_a and its corresponding frames F_a, where $0 \le a < J_3$. We express the relationship between frames and lines by defining a $J_5 \times J_4$ matrix, referred to as a *double identity matrix* (DIM). This matrix can be decomposed vertically into J_5/J_4 blocks. For $1 \le k \le J_5/J_4$, the kth block is equal to $I_0 + I_k$, where I_0 is a $J_4 \times J_4$ identity matrix, and I_k is a k-times right cyclically-shifted matrix of I_0. The rows of a DIM represent (1,2)-frames in F_a, while the columns represent the q^d-ary incidence vectors of lines in P_a. Each row includes two values of 1, representing two parallel lines in P_a that participate the corresponding frame in F_a. The column and row weights of DIM are J_2 and 2, respectively. Different frames in F_a and the corresponding parallel bundles P_a share the same DIM, such that the partial parallel decoding scheme can be operated using cyclically-shifted q^d-ary incidence vectors of lines in $\mathbf{L}_P$ as inputs on the column side to form the corresponding (1,2)-frames on the row side.

Next, we demonstrate the partial parallel decoding scheme. Recall that there are J_4 lines in a parallel bundle in $\mathrm{EG}^*(d, q)$, The bth line participating in the ath parallel bundle P_a is denoted as $\mathcal{L}_{a,b}$, where $0 \le b < J_4$. Let $\mathcal{L}_{a,b'}$ be another line in P_a, where $0 \le b' < J_4$ and $b' \neq b$. For each $\mathcal{L}_{a,b}$, there are J_2 parallel lines $\mathcal{L}_{a,b'}$. We can form $J_4 \times J_2$ pairs of (1,2)-frames in F_a denoted by $\mathcal{F}_{a,b} = \{\mathcal{L}_{a,b}, \mathcal{L}_{a,b'}\}$. For each P_a, we need to calculate $J_4 \times J_2$ frames. The decoding process in P_a is continued until all of the $J_4 \times q$ symbols participating in these J_4 lines in P_a have been updated. We redefine the index set N_i and M_j in II-A so as to represent the partial parallel decoding scheme. For $0 \le a < J_3$, $0 \le b < J_4$, the bth line in the ath group is identical to the the ith line in $\mathrm{EG}^*(d, q)$, where $i = a \times J_4 + b$ and $0 \le i < J_0$. Therefore, we use the notation $(a, b) \equiv i$ to represent the one and only one corresponding index for the ith line in $\mathrm{EG}^*(d, q)$. For $0 \le i < J_0$, and $0 \le j < n$, we define the index sets $N_{(a,b)}$ and M'_j by replacing i with (a, b) as $N_{(a,b)} = \{j : 0 \le j < n, v_{(a,b),j} \neq 0\}$ and $M'_j = \{(a, b) \equiv i : 0 \le (a, b) < J_0, v_{(a,b),j} \neq 0\}$, respectively. In the following, we present an example to illustrate the decomposition of the parity-check matrix of the NB-TF-EG code for partial parallel decoding.

Example 1: Let $d = 2$ and $q = 8 = 2^3$, and consider the two-dimensional Euclidean geometry $\mathrm{EG}(2, 2^3)$ over the field $\mathrm{GF}(2^3)$. The subgeometry $\mathrm{EG}^*(2, 2^3)$ comprises 63 non-origin points and 63 lines not passing through the origin of $\mathrm{EG}(2, 2^3)$. These 63 lines in $\mathrm{EG}^*(2, 2^3)$ form 189 (1,2)-frames. The parity-check matrix $\mathbf{H}$ of this code is a 189×63 matrix with 189 (1,2)-frames on the row side and 63 64-ary symbols on the column side. The null space over $\mathrm{GF}(2^6)$ of this parity-check matrix gives a 64-ary (63,45) NB-TF-EG code over $\mathrm{GF}(2^6)$. We decompose $\mathbf{H}$ as follows. A 63×63 matrix $\mathbf{L}_P$ with 64-ary incidence vectors of lines is formed to represent the relationship between points and lines in $\mathrm{EG}^*(2, 2^3)$. We divide 63 lines into 9 groups of parallel bundles, each group contains 7 lines. A 21×7 DIM is formed to represent the relationship between lines and (1,2)-frames in $\mathrm{EG}^*(2, 2^3)$. The decoding process is accomplished by decoding a 21×7 DIM 9 times using the corresponding 64-ary incidence vectors of 7 lines in $\mathbf{L}_P$ on the column side.

B. Proposed Iterative Two-Step Extended Min-Sum Algorithm

NB-TF-EG codes contain large numbers of short cycles of length 4. There are a total of $\binom{J_2}{2} \times \binom{q}{2}$ short cycles of length 4 in the Tanner graphs of these codes [10]. Using a standard belief propagation algorithm, such a large number of short cycles of length 4 would degrade decoding performance. The proposed ITS-EMS employs the orthogonal structure of NB-TF-EG codes to overcome the performance degradation resulting from short cycles.

Before outlining the decoding algorithm, we define some notation for later use. Upper script w represents the iteration index, and $w_{\max}$ is the maximum number of iterations to be performed. Suppose V_j is the jth received symbol, where $0 \le j < n$. A soft message of the jth code symbol at the wth decoding iteration is a vector comprising q^d sub-messages $\lambda_j^{(w)} = [\lambda_j^{(w)}(0), \lambda_j^{(w)}(1), \ldots, \lambda_j^{(w)}(q^d - 1)]$. The initial value $\lambda_j^{(0)} = [\lambda_j^{(0)}(0), \lambda_j^{(0)}(1), \ldots, \lambda_j^{(0)}(q^d - 1)]$ is the *a priori* information of the jth code symbol from the channel. The log-likelihood reliability (LLR) of the xth sub-message of $\lambda_j^{(w)}(x)$ is defined as

$$\lambda_j^{(w)}(x) = \ln \frac{Pb(V_j = z_j)}{Pb(V_j = x)} \tag{2}$$

with $Pb(V_j = x)$ as the probability of V_j equal to $x \in \mathrm{GF}(q^d)$. We define $z_j = \arg\max_{x \in \mathrm{GF}(q^d)} \lambda^{(w)}(x)$ as the *most likely* symbol for V_j, which also represents the hard-decision of the jth received symbol. Let $\oplus$ be the elementary CN operation (ECN) [15] with two-input messages and one output message. Notation $\sum^{\oplus}$ implies that the equation sums up the input messages using the operation of ECN and stores the smallest soft value. Let $\otimes$ be the multiplication in $\mathrm{GF}(q^d)$. Let $\delta_{i,j}^{(w)}$ and $\eta_{i,j}^{(w)}$ represent the VN-to-CN (V2C) and CN-to-VN (C2V) soft messages between the ith CN and the jth VN, respectively. For $x \in \mathrm{GF}(q^d)$, the xth LLR of $\delta_{i,j}^{(w)}$ and $\eta_{i,j}^{(w)}$ are denoted as $\delta_{i,j}^{(w)}(x)$ and $\eta_{i,j}^{(w)}(x)$, respectively. Let $\gamma_{i,j}$ be the symbol with the lowest reliability. With $\mathrm{v}_{i,j} = v_{i,j} \otimes V_j$, we let $\delta_{i,j}^{(w)}(x) = \ln(Pb(\mathrm{v}_{i,j} = \gamma_{i,j})/Pb(\mathrm{v}_{i,j} = x))$ and $\eta_{i,j}^{(w)}(x) = \ln(Pb(\mathrm{v}_{i,j} = \gamma_{i,j})/Pb(\mathrm{v}_{i,j} = x))$, where $\delta_{i,j}^{(w)}(\gamma_{i,j}) = 0$ and $\eta_{i,j}^{(w)}(\gamma_{i,j}) = 0$, respectively. To initialize the decoding process, we set $z_j = \arg\min_{x \in GF(q^d)} \lambda_j^{(0)}(x)$ and $\delta_{i,j}^{(0)}(v_{i,j} \otimes x) = \lambda_j^{(0)}(x)$.

We illustrate the ITS-EMS using partial parallel decoding for the case of V_j participating in the (1,2)-frames $\mathcal{F}_{a,b}$ formed by P_a, where $0 \le a < J_3$, and $0 \le b < J_4$. In the following, we use the relation $(a, b) \equiv i$ to rewrite the notation $\delta_{i,j}^{(w)}(x)$, $\eta_{i,j}^{(w)}(x)$, $\gamma_{i,j}$, and $\mathrm{v}_{i,j} = v_{i,j} \otimes V_j$ as $\delta_{(a,b),j}^{(w)}(x)$, $\eta_{(a,b),j}^{(w)}(x)$, $\gamma_{(a,b),j}$, and $\mathrm{v}_{(a,b),j} = v_{(a,b),j} \otimes V_j$. The soft messages of lines $\mathcal{L}_{a,b}$ in P_a are calculated first with scaling factor c by

$$LLR_{(a,b)}^{(w)}(x) = c \times \sum\nolimits_{j \in N_{(a,b)}}^{\oplus} \delta_{(a,b),j}^{(w)}\left(\mathrm{v}_{(a,b),j}\right), \tag{3}$$

where $0 \le c < 1$. The *extrinsic information* of $\mathcal{L}_{a,b}$ contributed by other lines $\mathcal{L}_{a,b'}$ participating in P_a with scaling factor κ is given by

$$E_{(a,b)}^{(w)}(x) = \kappa \times \sum_{\mathcal{L}_{a,b'} \in P_a, b \neq b'} LLR_{(a,b')}^{(w)}(x), \tag{4}$$

where $0 \le \kappa < 1$. The *extrinsic information* of V_j contributed by other received symbols participating in $\mathcal{L}_{a,b}$ except V_j is obtained by

$$E_{(a,b),j}^{(w)}(x) = \sum\nolimits_{j' \in N_{(a,b)} \backslash j}^{\oplus} \delta_{(a,b),j'}^{(w)}\left(\mathrm{v}_{(a,b),j'}\right) \tag{5}$$

where $\mathrm{v}_{(a,b),j'} = v_{(a,b),j'} \otimes V_j$. Let $\eta_{(a,b),j}^{(w)}(x)$ be a tentative C2V message for V_j in P_a. This can be derived by using the

ECN step with $E^{(w)}_{(a,b),j}(x)$ and $E^{(w)}_{(a,b)}$, which is formulated as

$$\eta^{(w)}_{(a,b),j}(x) = E^{(w)}_{(a,b),j}(x) \oplus E^{(w)}_{(a,b)}(x). \tag{6}$$

After finishing the partial decoding process from (3) to (6) for all symbols participating in J_3 parallel bundles, the post-processing for V_j is executed as

$$\lambda_j^{(w+1)}(x) = \lambda_j^{(w)}(x) + \sum_{(a,b)\in M'_j} \eta^{(w)}_{(a,b),j}\left(v_{(a,b),j\otimes x}\right), \tag{7}$$

where $0 \le j < n$. By letting $w \leftarrow w+1$, we obtain

$$z_j^{(w)} = \arg \min_{x\in \mathrm{GF}(q^d)} \lambda_j^{(w+1)}(x). \tag{8}$$

A new received vector $\mathbf{z}^{(w)}$ is formed from (8) for syndrome calculation. For $0 \le j < n$, we execute the VN processing to derive the new V2C messages $\delta^{(w+1)}_{(a,b),j}(x)$ for the next iteration. First, we compute the primitive V2C messages by

$$\hat{\delta}^{(w+1)}_{(a,b),j}\left(v_{(a,b),j}\otimes x\right) = \lambda_j^{(w+1)}(x) - \eta^{(w)}_{(a,b),j}\left(v_{(a,b),j}\otimes x\right). \tag{9}$$

Thereafter, the $(w+1)$-th V2C messages are derived by normalizing primitive V2C messages with respect to the *most likely* symbol $\gamma_{(a,b),j}$ as

$$\delta^{(w+1)}_{(a,b),j}(x) = \hat{\delta}^{(w+1)}_{(a,b),j}(x) - \hat{\delta}^{(w+1)}_{(a,b),j}\left(\gamma_{(a,b),j}\right), \tag{10}$$

where

$$\gamma_{(a,b),j} = \arg \min_{x\in \mathrm{GF}(q^d)} \hat{\delta}^{(w+1)}_{(a,b),j}(x). \tag{11}$$

Based on the above updating process and notation, the proposed ITS-EMS is formulated in Algorithm 1.

Algorithm 1 ITS-EMS

```
1) Initialization: For $0 \le i < J_0$ and $0 \le j < n$, set $z_j = \arg\min_{x\in GF(q^d)} \lambda_j^{(0)}(x)$, $\delta_{i,j}^{(0)}(v_{i,j} \otimes x) = \lambda_j^{(0)}(x)$ with $v_{i,j} \neq 0$, $w = 0$, and the maximum number of iterations to $w_{\max}$.
2) Let $\mathbf{S}^{(w)}(X)$ be the syndrome derived by dividing the received polynomial $\mathbf{z}^{(w)}(X)$ by the generator polynomial $\mathbf{g}(X)$ of the codes. If $\mathbf{S}^{(w)}(X) = 0$, then stop the decoding process and output $\mathbf{z}^{(w)}$ as the decoded codeword.
3) If $w = w_{\max}$, then stop the decoding process. If $\mathbf{S}^{(w)}(X) \neq 0$, declare a decoding failure.
4) CN processing:
     For $0 \le a < J_3$, $0 \le b, b' < J_4$, and $i = a \times J_4 + b$,
   a) Compute soft messages for lines in $P_a$ by (3).
   b) Calculate (4) and (5).
   c) Update tentative C2V messages by (6).
5) Post processing:
     For $0 \le j < n$, execute the post processing for $V_j$ by (7). Let $w \leftarrow w+1$, and form a new received vector $\mathbf{z}^{(w)}$ by (8).
6) VN processing:
   a) Compute the V2C messages by (9).
   b) Normalize the V2C messages by (10) and (11).
7) Go to Step 2.
```

TABLE I
MEMORY REQUIREMENTS FOR FULLY AND PARTIAL PARALLEL DECODING

Type(Bits)	Input	C2V	V2C	LS	EL	ES
Fully parallel	nU	nJ_1U	nqU	J_0U	J_0U	J_0qU
Partial parallel	nU	nJ_1U	nqU	J_4U	J_4U	J_4qU

LS: The line-sum defined in (3).
EL: The *extrinsic information* contributed by other lines defined in (4).
ES: The *extrinsic information* contributed by other symbols defined in (5).

Next, we demonstrate the complexity analysis of the ITS-EMS with q as a power of 2. To ensure the best performance for the code, we take q^d elements of field GF(q^d) as the input for each symbol. We also have $d = 2$ for the NB-TF-EG codes constructed using the two-dimensional Euclidean geometry. At the Step 2, a $(n-k)$-stage syndrome calculation necessarily employs at most $(n-k)$ finite field additions and $(n-k)$ finite field multiplications, where k is the number of information symbols. We use the bubble check [20] to calculate the ECN. Each stage in the ECN requires $2 \times q^2$ additions and q^3 comparisons. At the Step 4, $q-1$ ECN steps are required for (3). $2J_4$ multiplications are required for the scaling factors c and κ in (3) and (4), and J_4J_2 additions are required for (4). Moreover, to update each symbol in a line in (5) and (6), we need $2q-4$ and q ECN operations, respectively. Therefore, it takes $J_4(4q-5)$ ECN operations to calculate all the line-sum of the lines and update each symbol in each line in P_a. Since there are J_3 blocks, a total of $J_3J_4(4q-5)$ ECN operations, $2J_3J_4$ multiplications, and $J_3J_4J_2$ additions are needed to perform one iteration. At the step 5, J_1n additions are needed for (7), and nq^2 comparisons are needed for (8). At the Step 6, nq^3 additions and comparisons are requried for (9), and nq^3 additions are required for (10). With some translations, we summarize the computational complexity with code length n and q. To carry out one iteration of the ITS-EMS algorithm, $(10n^2 + 12n)(q-1)$ real-number additions, $9(n^2+n)(q-1) - nq$ real-number comparisons, and $2n$ real-number multiplications are required. Both the addition and the comparison operations are on the order of $O(n^2q)$, and the multiplication operations are on the order of $O(n)$.

Table I presents the memory requirements for fully and partial parallel decoding. Each value in the table has N bits of finite precision represented by $U = q^d(N + \log_2 q^d)$. It turns out that partial parallel decoding saves on storage for line-sums, *extrinsic information* contributed by other lines, and *extrinsic information* contributed by other symbols at a factor of J_0/J_4. Thus, partial parallel decoding provides an alternative for fully parallel decoding if memory is limited.

In the following, two examples are presented to demonstrate the frame error rate (FER) performances of the NB-TF-EG codes decoded using the proposed ITS-EMS and various decoding algorithms for short to moderate code lengths. Note that the decoding complexity of the NB-LDPC codes is in proportion to the field size of the finite field [12], [15]. For constructing NB-TF-EG codes with longer block length, the construction needs to be modified as in [18] to decrease the field size of the codes and thus reduce decoding complexity. We also include the error performances of the RS codes with same lengths and rates decoded using the HD-BM and the ASD-KV algorithm.

TABLE II

NUMBER OF COMPUTATIONS REQUIRED FOR ITERATIVE DECODING OF NB-TF-EG CODES AND ASD-KV DECODING OF RS CODES

Codes	Decoding algorithm	(N, K)	
		$(255, 191)$	$(63, 45)$
RS	ASD-KV[1], $\mu = 9.99$	4.26×10^8	2.6×10^7
	ASD-KV[1], $\mu = 4.99$	1.6×10^7	10^6
NB-TF-EG	ITS-EMS[2], 5	2.83×10^8	1.45×10^6
	ITS-EMS[2], 3	1.7×10^8	8.73×10^5
	IRTS-MLGD[2], 10	1.11×10^7	3.67×10^5
	IRTS-MLGD[2], 5	5.55×10^6	1.83×10^5

1) Only the number of computations for the interpolation step of the ASD-KV algorithm is considered in this table.

2) The IRTS-MLGD employs integer operations, while the ITS-EMS employs operations in real numbers.

The computational complexity of the ASD-KV algorithm is on the order of $(\lfloor\lambda\rfloor^4 N^2)$ (the interpolation step), where N is the length of the code and λ is the parameter of multiplicity assignment in the interpolation steps. We use $\lambda = \infty$, $\lambda = 9.99$, and $\lambda = 4.99$ for comparison [21]. Scaling factors c and κ for decoding the NB-TF-EG codes in BPSK with the ITS-EMS are determined by the points with the lowest signal to noise ratio via extensive simulation, as illustrated in Fig. 5. We use the same scaling factors for the higher order modulations. We also examine the performance of one-step MLG decodable NB-LDPC codes with similar lengths and rates constructed based on Euclidean geometries via matrix dispersion [14] and [18]. Furthermore, Table II illustrates the number of computations required for the proposed two iterative decoding algorithms decoding the NB-TF-EG codes and the ASD-KV algorithm decoding RS codes. The numbers for the corresponding NB-TF-EG codes are derived by summing up all of the operations of the ITS-EMS algorithm. In addition, the major computational complexity to carry out the ASD-KV algorihtm comes from the interpolation step [21]; therefore, we only consider this type of calculation for comparison.

Example 2: Let $d = 2$ and $q = 8 = 2^3$. Consider the two-dimensional Euclidean geometry $\mathrm{EG}(2, 2^3)$ over the field $\mathrm{GF}(2^3)$. From *Example 1*, we know that the null space of the parity-check matrix of this code is the 64-ary (63,45) NB-TF-EG code with $J_1 = 8$, $J_2 = 6$. By using NB-TS-MLGD, 3 symbol errors can be corrected. The Tanner graph of this code has 79380 cycles of length 4. From Fig. 5, we set $c = 0.2$ and $\kappa = 0.21$. Fig. 1 shows the FER performances of the 64-ary (63,45) NB-TF-EG code over the AWGN channel with BPSK transmission decoded using the proposed ITS-EMS with 3 and 5 iterations, standard EMS with 50 iterations, and NB-TS-MLGD. We also include the error performances of the (63,45) RS code over $\mathrm{GF}(2^6)$ decoded using the HD-BM and the ASD-KV algorithms. In addition, the FER performance of the standard EMS algorithm in decoding one-step MLG decodable NB-LDPC code with same rate and length is also included. This code is a 64-ary (63,45) NB-LDPC code with two different column weights 2 and 3, and row weight 8. At the FER of 10^{-6}, the NB-TF-EG code decoded using the proposed ITS-EMS with 5 iterations achieves a coding gain of 2.2 dB over the RS code decoded using the HD-BM algorithm, as well as a coding gain of 1 dB, 1.3 dB and 1.6 dB over the RS code decoded using the ASD-KV algorithm with $\lambda = \infty$, $\lambda = 9.99$, and $\lambda = 4.99$, respectively. Due to the degrading effect of short cycles of length 4, the NB-TF-EG code decoded using the standard EMS algorithm with 50 iterations gains only 0.5 dB over the RS code decoded using the HD-BM algorithm, and degrades by 1.6 dB, compared to the proposed ITS-EMS with 5 iterations. Moreover, the NB-TF-EG code decoded with 5 iterations of the ITS-EMS outperforms the NB-TS-MLGD by 4.3 dB. At the FER of 10^{-4}, we find that the low column weights of the one-step MLG decodable 64-ary (63,45) NB-LDPC code decoded with 10 iterations of standard EMS result in an error floor phenomenon. The 64-ary (63,45) NB-TF-EG code decoded with 5 iterations of the ITS-EMS achieves a coding gain of 1 dB over the 64-ary (63,45) NB-LDPC code decoded with 10 iterations of standard EMS.

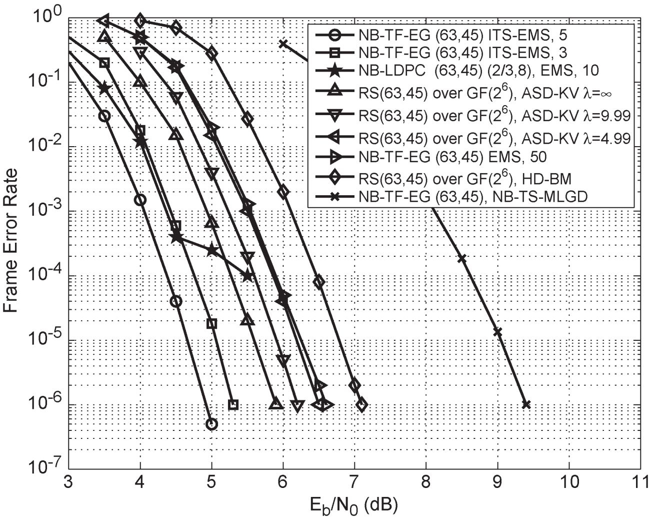

Fig. 1. Frame error rates of various decoding algorithms forthe 64-ary (63,45) NB-TF-EG code, the 64-ary (63,45) NB-LDPC code, and the (63,45) RS code over $\mathrm{GF}(2^6)$ decoded with the HD-BM and the ASD-KV algorithms using BPSK over the AWGN channel.

Fig. 2 shows the FER versus E_b/N_0 performance of the 64-ary (63,45) NB-TF-EG code and the (63,45) RS code over the AWGN channel using 64-QAM. At the FER of 10^{-5}, the NB-TF-EG code decoded with 5 iterations of the ITS-EMS achieves a coding gain of 2.5 dB over the RS code decoded using the HD-BM, as well as a coding gain of 1.2 dB, 1.6 dB, and 1.9 dB over the RS code decoded using the ASD-KV algorithm with $\lambda = \infty$, $\lambda = 9.99$, and $\lambda = 4.99$, respectively. In addition, the ITS-EMS with 5 iterations outperforms the standard EMS with 50 iterations by 2 dB for decoding the NB-TF-EG code.

In Table II, the number of computations for decoding the 64-ary (63,45) NB-TF-EG code with 5 iterations of the ITS-EMS is on the order of 1.45×10^6. On the other hand, the number of computations for the (63,45) RS code decoded using the ASD-KV algorithm in the interpolation step with $\lambda = 9.99$ is on the order of 2.6×10^7. From Fig. 1 and Table II, the 64-ary (63,45) NB-TF-EG code decoded with 5 iterations of the ITS-EMS achieves a 1.3 dB coding gain over the (63,45) RS code decoded using the ASD-KV algorithm with $\lambda = 9.99$, representing an order of magnitude reduction in the number of computations.

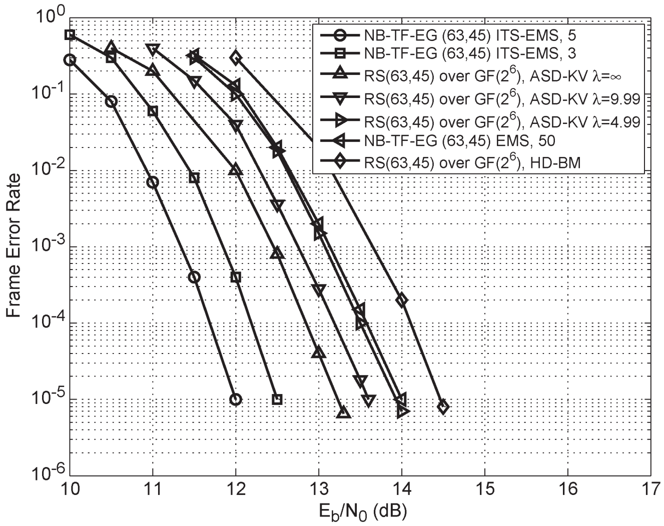

Fig. 2. Frame error rates of various decoding algorithms forthe 64-ary (63,45) NB-TF-EG code and the (63,45) RS code over GF(2^6) decoded with the HD-BM and the ASD-KV algorithms using 64-QAM over the AWGN channel.

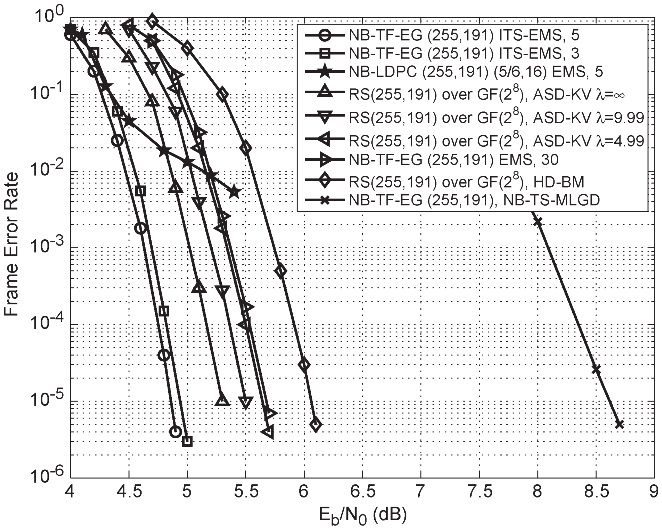

Fig. 3. Frame error rates of various decoding algorithms for the 256-ary (255,191) NB-TF-EG code, the 256-ary (255,193) NB-LDPC code, and the (255,191) RS code over GF(2^8) decoded with the HD-BM and the ASD-KV algorithms using BPSK over the AWGN channel.

Example 3: Let $d = 2$ and $q = 16 = 2^4$. Consider the two-dimensional Euclidean geometry EG$(2, 2^4)$ over the field GF(2^4). The subgeometry EG$^*(2, 2^4)$ consists of 255 non-origin points and 255 lines not passing through the origin of EG$(2, 2^4)$, which form 1785 (1,2)-frames. The parity-check matrix $\mathbf{H}$ of this code is a 1785 × 255 matrix with 1785 (1,2)-frames on the row side and 255 256-ary symbols on the column side. The null space over GF(2^8) of this parity-check matrix gives a 256-ary (255,191) NB-TF-EG code over GF(2^8).

The decomposition of $\mathbf{H}$ is as follows. With the 255 lines and the 255 points in EG$^*(2, 2^4)$, a 255 × 255 matrix $\mathbf{L}_P$ with 256-ary incidence vectors of lines is formed. The 255 lines in EG$^*(2, 2^4)$ can be divided into 17 groups of parallel bundles, with each of them consisting of 15 lines and forming 105 (1,2)-frames. A 105 × 15 DIM with binary incidence vectors of (1,2)-frames is formed as a unit for partial parallel decoding. The decoding process is accomplished by decoding a 105 × 15 DIM 17 times with the corresponding 256-ary incidence vectors of 15 lines on the column side. The values of J_1 and J_2 for the 256-ary (255,191) NB-TF-EG code code are 16 and 14, respectively. This code can correct up to 7 symbol errors with NB-TS-MLGD. The Tanner graph of this code contains 19,492,200 short cycles of length 4. From Fig. 5, we set $c = 0.2$ and $\kappa = 0.05$, respectively. Fig. 3 shows the FER performances of the 256-ary (255,191) NB-TF-EG code over the AWGN channel with BPSK signaling decoded using the proposed ITS-EMS with 3 and 5 iterations, standard EMS with 30 iterations, and NB-TS-MLGD. We also include the error performances of the (255,191) RS code over GF(2^8) decoded using the HD-BM algorithm and ASD-KV algorithms using $\lambda = \infty$, $\lambda = 9.99$, and $\lambda = 4.99$, respectively. In addition, the FER performance of the standard EMS algorithm in decoding one-step MLG decodable NB-LDPC code with same rate and length is also included. The code is a 256-ary (255,193) NB-LDPC code with two different column weights 5 and 6, and row weight 16. At the FER of 10^{-5}, we see that the NB-TF-EG code decoded using 5 iterations of the ITS-EMS achieves a coding gain of 1.3 dB over the RS code decoded using the HD-BM algorithm, and a coding gain of 0.4 dB, 0.6 dB and 0.75 dB over the RS code decoded using the ASD-KV algorithm with $\lambda = \infty$, $\lambda = 9.99$, and $\lambda = 4.99$, respectively. Note that the performance gap between 3 and 5 iterations of the ITS-EMS is less than 0.1 dB. Moreover, the 256-ary (255,191) NB-TF-EG code decoded with 5 iterations of the ITS-EMS outperforms the NB-TS-MLGD by 3.7 dB. We notice that the NB-TF-EG code decoded with the standard EMS algorithm performs poorly due to the degrading effect of short cycles of length 4. The ITS-EMS with 5 iterations achieves a 0.9 dB coding gain over the standard EMS with 30 iterations. At the FER of 10^{-2}, note that the low column weights of the one-step MLG decodable 256-ary (255,193) NB-LDPC code decoded with 5 iterations of the standard EMS result in an error floor phenomenon. In Fig. 4, we demonstrate the FER versus E_s/N_0 performance of the 256-ary (255,191) NB-TF-EG code and the (255,191) RS code over the AWGN channel using 256-QAM. At FER $= 10^{-5}$, the NB-TF-EG code decoded using 5 iterations of the ITS-EMS achieves a coding gain of 1.5 dB over the RS code decoded using the HD-BM, as well as a coding gain of 0.5 dB, 0.7 dB and 0.8 dB over the RS code decoded using the ASD-KV algorithm with $\lambda = \infty$, $\lambda = 9.99$ and $\lambda = 4.99$, respectively. Also, the 256-ary (255,191) NB-TF-EG code decoded using the ITS-EMS with 5 iterations outperforms the standard EMS with 30 iterations by 0.8 dB.

As shown in Table II, the number of computations for decoding the 256-ary (255,191) NB-TF-EG code with 3 iterations of the ITS-EMS is on the order of 1.7×10^8. In contrast, the number of computations for decoding the (255,191) RS code with the ASD-KV algorithm in the interpolation step with $\lambda = 9.99$ is on the order of 4.26×10^8. From Fig. 3 and Table II, the 256-ary (255,191) NB-TF-EG code decoded with 3 iterations of the ITS-EMS outperforms the (255,191) RS code decoded with ASD-KV $\lambda = 9.99$ by 0.4 dB, providing 60% reduction in computational complexity.

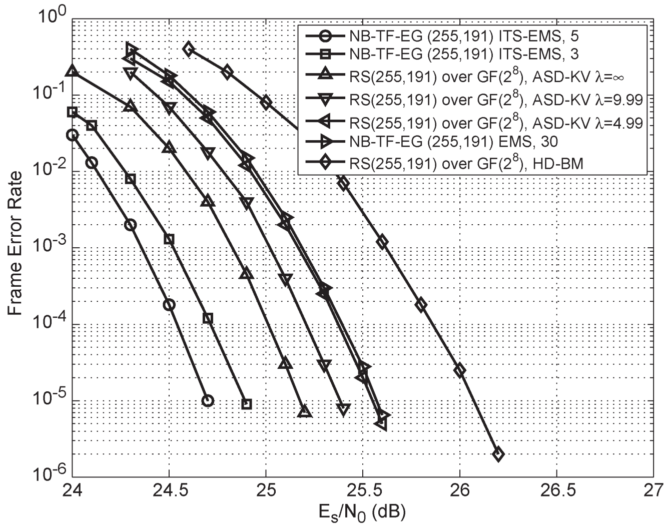

Fig. 4. Frame error rates of various decoding algorithms for the 256-ary (255,191) NB-TF-EG code and the (255,191) RS code over GF(2^8) decoded with the HD-BM and the ASD-KV algorithms using 256-QAM over the AWGN channel.

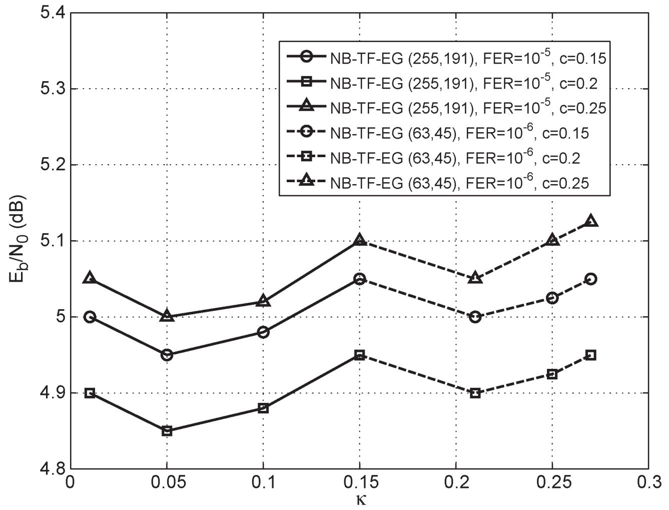

Fig. 5. Scaling factors c and κ for the 64-ary NB-TF-EG (63,45) and the 256-ary NB-TF-EG (255,191) in BPSK decoding with 5 iterations of ITS-EMS with target FER of 10^{-5} and 10^{-6}, respectively.

IV. Iterative Reliability Two-Step MLGD Algorithm

The computational complexity of the ITS-EMS algorithm is high because a large number of operations are performed using real numbers. In this section, we present a simplified decoding algorithm, called IRTS-MLGD algorithm. The IRTS-MLGD only utilizes finite field and integer operations, which greatly reduce computational complexity, compared to the ITS-EMS using operations in real numbers. In addition, compared to the *one-pass* NB-TS-MLGD employs only hard-decision values from the received symbols, the IRTS-MLGD utilizes the soft information of the received symbols in conjunction with an iterative decoding process. As a result, a considerable coding gain can be achieved.

For practical applications, we devise the algorithm over GF(2^r). Consider NB-TF-EG code C over GF(2^r) of length n. Let $\mathbf{y}_s$ be the soft received sequence at the received sampler represented by $\mathbf{y}_s = (y_0, y_1, \ldots, y_{n-1})$, where "s" stands for soft information. For $0 \le j < n$, each element of $\mathbf{y}_s$ in GF(2^r) is represented as an r-tuple $y_j = (y_{j,0}, y_{j,1}, \ldots, y_{j,r-1})$ over GF(2). The hard-decision received sequence $\mathbf{z} = (z_0, z_1, \ldots, z_{n-1})$ over GF(2^r) is determined by $\mathbf{y}_s$, where z_j is an estimate of the jth transmitted symbol, for $0 \le j < n$. Let $\rho_{j,k}$ be the quantized value of sample $y_{j,k}$, where $0 \le j < n$ and $0 \le k < r$. The quantized value is an integer representation of the $2^p - 1$ quantized intervals symmetric to the origin. Each interval has a length $\triangle$ and each sample is represented by p bits. Therefore, $\rho_{j,k}$ is in the range of $[-(2^{(p-1)} - 1), +2^{(p-1)} - 1]$. For $0 \le j < n$, the jth group $(\rho_{j,0}, \rho_{j,1}, \ldots, \rho_{j,r-1})$ is decoded into element a in GF(2^r) $= \{a_0, a_1, \ldots, a_{2^r-1}\}$. For $0 \le l < 2^r$, the binary representation of the lth element $a_l \in$ GF(2^r) is denoted by an r-tuple $a_l = (a_{l,0}, a_{l,1}, \ldots, a_{l,r-1})$ over GF(2). For each element $a_l \in$ GF(2^r), we calculate the reliability measure of a_l as

$$\phi_{j,l} = \sum_{k=0}^{r-1} (1 - 2a_{l,k})\rho_{j,k} \tag{12}$$

which is in the range of $[-r(2^{(p-1)-1}), +r(2^{(p-1)-1})]$. Let a be the element in GF(2^r) with the highest reliability, and a is selected as z_j. For $0 \le j < n$, let $\boldsymbol{\phi_j} = (\phi_{j,0}, \phi_{j,1}, \ldots, \phi_{j,2^r-1})$ which is called the *decision vector* of the jth received symbol z_j. For $0 \le i < J_0$, the reliability measure of the jth received symbol is given by

$$\varphi_{i,j} = \min_{j' \in N_i \backslash j} \max_{l} \phi_{j',l} \tag{13}$$

which can be regarded as a reliability measure of the *extrinsic information* contributed to z_j by other received symbols in $S(L_i)$. Consider the jth received symbol z_j participating in $F^{j,u,t}$ which consists of two lines L_u^j and $L_{t,u}^j$, where $0 \le u < J_1$ and $0 \le t < J_2$. Frame-sum $S(F^{j,u,t})$ is actually a check-sum in $\mathbf{H}$. There are $J_1 J_2$ check-sums that contain z_j. Assume that $S(F^{j,u,t})$ participates in the ith line of EG$^*(d, q)$, where $0 \le i < J_0$, $0 \le j < n$, $0 \le u < J_1$, and $0 \le t < J_2$. $S(F^{j,u,t})$ can be normalized for decoding the jth received symbol as

$$\begin{aligned} S'(F^{j,u,t}) &= v_{i,j}^{-1} S(F^{j,u,t}) \\ &= v_{i,j}^{-1} S\left(L_u^j\right) + v_{i,j}^{-1} S\left(L_{t,u}^j\right) \\ &= z_j + v_{i,j}^{-1} \sum_{l \in N_i \backslash j} v_{i,l} z_l + v_{i,j}^{-1} S\left(L_{t,u}^j\right). \end{aligned} \tag{14}$$

Next, we consider the partial parallel decoding scheme mentioned in III-A for the proposed IRTS-MLGD algorithm. The bth line in which the jth received symbol participates in the a-group is denoted as $\mathcal{L}_{a,b}$. The *extrinsic information* of the jth received symbol comprises two parts. The first part is the *extrinsic symbol information*, which comes from the frame-sum $S(F^{j,u,t})$ without the jth symbol. The other part is the magnitude of the reliability measure, which comes from the reliability measure of the parallel lines of $\mathcal{L}_{a,b}$ and the reliability measure of the symbols participating in the same line as the jth symbol. Recall that $w_{\max}$ is the maximum number of iterations to be performed. At the wth iteration, the jth received symbol is denoted as $z_j^{(w)}$. The *extrinsic information* of $\mathcal{L}_{a,b}$ can be derived by the line-sum and the reliability measure of $\mathcal{L}_{a,b}$ as

$$S^{(w)}(\mathcal{L}_{a,b}) = \sum_{j \in N_{(a,b)}} v_{(a,b),j} z_j^{(w)}, \qquad (15)$$

$$\Gamma_{\mathcal{L}_{a,b}} = \min_{j \in N_{(a,b)}} \max_{l} \phi_{j,l}. \qquad (16)$$

The reliability measure of the jth received symbol participating in $\mathcal{L}_{a,b}$ is calculated as

$$\varphi_{(a,b),j} = \min_{j' \in N_{(a,b)} \setminus j} \max_{l} \phi_{j',l}. \qquad (17)$$

The *extrinsic symbol information* of the bth line is contributed by other lines in the a-th group as

$$\xi_{(a,b),j}^{(w)} = \sum_{\mathcal{L}_{a,b'} \in P_a, b \neq b'} v_{(a,b),j}^{-1} S^{(w)}(\mathcal{L}_{a,b'}). \qquad (18)$$

The frame-sum (14) can be rewritten as

$$S'^{(w)}(F^{j,u,t}) = z_j^{(w)} + v_{(a,b),j}^{-1} \left(\sum_{l \in N_{(a,b)} \setminus j} v_{(a,b),l} z_l^{(w)} + \xi_{(a,b),j}^{(w)} \right). \qquad (19)$$

Let

$$\sigma_{(a,b),j}^{(w)} = v_{(a,b),j}^{-1} \left(\sum_{l \in N_{(a,b)} \setminus j} v_{(a,b),l} z_l^{(w)} + \xi_{(a,b),j}^{(w)} \right). \qquad (20)$$

The normalized check-sum $S'^{(w)}(F^{j,u,t})$ can be rewritten as

$$S'^{(w)}(F^{j,u,t}) = z_j^{(w)} + \sigma_{(a,b),j}^{(w)}. \qquad (21)$$

The *extrinsic symbol information* of the jth received symbol can be derived by

$$\sigma_{(a,b),j}^{(w)} = S'^{(w)}(F^{j,u,t}) - z_j^{(w)}. \qquad (22)$$

From (22), we can see that: 1) if $S'^{(w)}(F^{j,u,t}) = 0$ and $\sigma_{(a,b),j}^{(w)}$ is error free, then $z_j^{(w)}$ must be error free; 2) if $S'^{(w)}(F^{j,u,t}) \neq 0$ and $\sigma_{(a,b),j}^{(w)}$ is error free, then $z_j^{(w)}$ contains an error e_j. The value of $z_j^{(w)}$ must be changed to $z_j^{(w)} - e_j = -\sigma_{(a,b),j}^{(w)}$ to make the normalized check-sum $S'^{(w)}(F^{j,u,t})$ equal to zero when $e_j = S'^{(w)}(F^{j,u,t})$. Next, we consider updating the magnitude of the reliability measure of the jth received symbol which participates in $\mathcal{L}_{a,b}$. The first step is to calculate the reliability measure contributed by J_2 $\mathcal{L}_{a,b'}$ parallel to $\mathcal{L}_{a,b}$ and denoted as

$$\beta_{\mathcal{L}_{a,b}} = \min_{\mathcal{L}_{a,b'} \in P_a} \Gamma_{\mathcal{L}_{a,b'}}, \qquad (23)$$

where $\Gamma_{\mathcal{L}_{a,b'}}$ is derived as (16) with $\mathcal{L}_{a,b}$ replaced with $\mathcal{L}_{a,b'}$. In the second step, we update the reliability measure of the received symbol $\mathbf{z}$ participating in each $\mathcal{L}_{a,b}$. Let $\boldsymbol{\psi}_j^{(w)}$ be the *decision vector* of the magnitude of the reliability measure for the jth received symbol contributed by other symbols participating in the line $\mathcal{L}_{a,b}$ except $z_j^{(w)}$ and other lines parallel to $\mathcal{L}_{a,b}$ at the wth iteration. For $(a,b) \in M'{}_j$ and $0 \leq j < n$, the *decision vector* is denoted as $\boldsymbol{\psi}_j^{(w)} = (\psi_{j,0}^{(w)}, \psi_{j,1}^{(w)}, \ldots, \psi_{j,2^r-1}^{(w)})$, and derived by summing up the the minimum value between $\beta_{\mathcal{L}_{a,b}}$ and $\varphi_{(a,b),j}$ when the *extrinsic symbol information* $-\sigma_{(a,b),j}^{(w)}$ equals $a_l \in \mathrm{GF}(2^r)$. The lth element in $\boldsymbol{\psi}_j^{(w)}$ is calculated by

$$\psi_{j,l}^{(w)} = \sum_{\substack{(a,b) \in M'_j \\ -\sigma_{(a,b),j}^{(w)} = a_l}} \min\left(\beta_{\mathcal{L}_{a,b}}, \varphi_{(a,b),j}\right). \qquad (24)$$

Let $\mathbf{R}_j^{(w)} = \{R_{j,0}^{(w)}, R_{j,1}^{(w)}, \ldots, R_{j,2^r-1}^{(w)}\}$ be the reliability measure vector of $z_j^{(w)}$ at the wth iteration, where $R_{j,l}^{(w)}$ is the reliability measure, such that a_l is taken to be $z_j^{(w)}$. In the $(w+1)$-th iteration, the reliability measure of $z_j^{(w+1)}$ is updated by

$$\mathbf{R}_j^{(w+1)} = \mathbf{R}_j^{(w)} + \boldsymbol{\psi}_j^{(w)}. \qquad (25)$$

For $w = 0$ and $0 \leq j < n$, we set $R_{j,l}^{(0)} = \epsilon \phi_{j,l}$, where the parameter ϵ is called a scaling factor which is selected to optimize the performance of a given code.

Algorithm 2 IRTS-MLGD

```
1) Initialization: For $0 \leq j < n$, set $R_{j,l}^{(0)} = \epsilon \phi_{j,l}$, $w = 0$,
   and the maximum number of iterations to $w_{\max}$.
2) Let $\mathbf{S}^{(w)}(X)$ be the syndrome derived by dividing the received
   polynomial $\mathbf{z}^{(w)}(X)$ by the generator polynomial$\mathbf{g}(X)$ of the codes.
   If $\mathbf{S}^{(w)}(X) = 0$, then stop the decoding process and output
   $\mathbf{z}^{(w)}$ as the decoded codeword.
3) If $w = w_{\max}$, then stop the decoding process. If $\mathbf{S}^{(w)}(X) \neq 0$,
   then declare a decoding failure.
4) For $0 \leq a < J_3$, $0 \leq b, b' < J_4$ and $i = a \times J_4 + b$:
       Update the elements in $\boldsymbol{\psi}_j^{(w)}$ using (24) by selecting the minimum
       value between (17) and (23) when symbol extrinsic information
       $-\sigma_{(a,b),j}^{(w)}$ equals $a_l \in \mathrm{GF}(2^r)$.
5) For $0 \leq j < n$, update the reliability measure vector $\mathbf{R}_j^{(w+1)}$
   using (25). Make the hard-decision $z_j^{(w)} = \arg\max_{a_l} R_{j,l}^{(w)}$.
   Let $w \leftarrow w + 1$, and form a new received vector $\mathbf{z}^{(w)}$.
6) Proceed to Step 2.
```

Due to the limit of quantization bit widths, we need to bound the range of the reliability for each symbol. Let $\Delta = (2^{p-1} - 1)$ be the range of quantization. If $R_{j,\max}^{(w+1)} \triangleq \max_l(R_{j,l}^{(w)} + \psi_{j,l}^{(w)})$ is greater than Δ, then $R_{j,\max}^{(w+1)}$ is truncated at Δ. By defining $\pi = R_{j,\max}^{(w+1)} - \Delta$, we obtain

$$R_{j,l}^{(w+1)} = \begin{cases} -\Delta, & \text{if } R_{j,l}^{(w+1)} - \pi < -\Delta; \\ R_{j,l}^{(w+1)} - \pi, & \text{otherwise} \end{cases}$$

Using the above updating process, the proposed IRTS-MLGD algorithm is formulated in Algorithm 2.

The computational complexity of the IRTS-MLGD is analyzed as follows. The initialization of the decoding algorithm needs $n2^r \log 2^r$ integer additions for (12) and $n2^r$ integer multiplications for ε to compute the reliability measure of all $\mathbf{R}_j^{(0)}$. In addition, $J_3J_4(3q-5)(2^r-1)$ integer comparisons are required to calculate (16) and (17). At the Step 2, an $(n-k)$-stage syndrome calculation must be performed using no more than $(n-k)$ finite field additions and $(n-k)$ finite field multiplications. At the Step 4, $J_4(q-1)$ finite field additions and J_4q finite field multiplications are required for the calculation of J_4 line-sums using (15) in each P_a. For two-step decoding, J_2J_4 finite field additions and J_2J_4 finite field multiplications are required for (18). J_4q finite field additions and J_4q finite field multiplications are required to calculate (20), and q finite field additions are required for (22) to update the symbols in the corresponding line-sum. Finally, $J_4(q-2)(2^r-1)$ integer comparisons are required to calculate (24). $J_3[J_4(J_2-1+2q)+q]$ finite field additions, $J_3J_4(J_2+2q)$ finite field multiplications, and $J_3J_4(q-2)(2^r-1)$ integer comparisons are required to complete J_3 blocks for partial parallel decoding. At the Step 5, nJ_1 integer additions are required to update (25). Moreover, a maximum of $n2^r$ integer additions and $n(2^r-1)$ integer comparisons are required for normalization, and $n(2^r-1)$ integer comparisons are required to make hard decisions. The computational complexity is summarized with some translations with code length n and q. A total of $n(4q-2)$ finite field additions, $n(3q-2)$ finite field multiplications, $nq(q+1)$ integer additions, and $nq(n+1)$ integer comparisons are required to carry out one iteration of the IRTS-MLGD. In the following examples, the bit widths are respectively 10-bits and 12-bits and the interval length of both codes is $\triangle=0.3125$. For convenience, the computational complexity of the ITS-EMS and the IRTS-MLGD is evaluated according to the number of operations. The IRTS-MLGD is shown to reduce computational complexity to a degree exceeding that of real numbered ITS-EMS with 32-bit floating point format in IEEE Standard 754 [22]. Moreover, the number of computations for the interpolation of ASD-KV algorithm exceeds that of the two proposed iterative decoding algorithms as λ increases to $\lambda=9.99$.

Example 4: Consider the 64-ary (63,45) NB-TF-EG code in Example 1 with $\epsilon=8$. Fig. 6 shows the FER performances of the NB-TF-EG code over the AWGN channel with BPSK transmission using the ITS-EMS with 5 iterations, the IRTS-MLGD with 5 and 10 iterations, standard EMS with 50 iterations, and NB-TS-MLGD. The FER performances of the (63,45) RS code over GF(2^6) decoded using the HD-BM and the ASD-KV algorithm are also included. As shown in Table II, the number of integer operations for the IRTS-MLGD with 10 iterations is on the order of 3.67×10^5. In contrast, the number of computations in real numbers using the ITS-EMS with 5 iterations is on the order of 1.45×10^6. At the FER of 10^{-6}, the IRTS-MLGD with 10 iterations reduces the number of computations by 75% with a 1.1 dB in performance loss, compared to the ITS-EMS with 5 iterations. Besides, the IRTS-MLGD with 10 iterations outperforms the NB-TS-MLGD by 3 dB and achieves 1 dB coding gain over the RS code decoded using the HD-BM algorithm. Furthermore, the NB-TF-EG code decoded using the IRTS-MLGD with 10 iterations nearly exceeds the RS code decoded using the ASD-KV algorithm with $\lambda=\infty$, and achieves a 0.5 dB coding gain over the RS code decoded using the ASD-KV algorithm with $\lambda=4.99$.

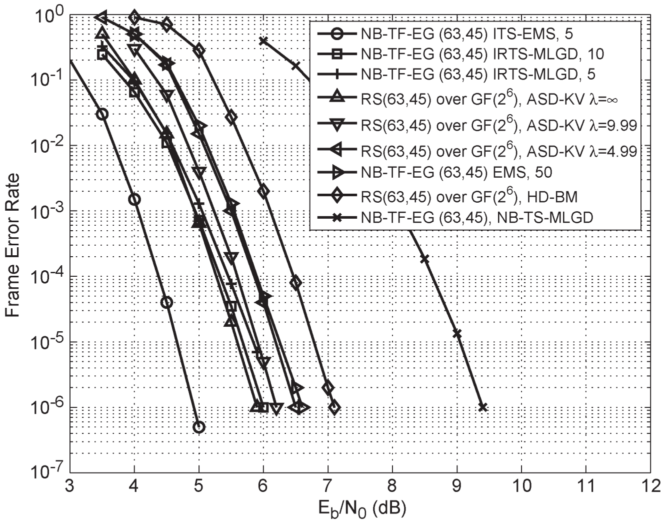

Fig. 6. Frame error rates of the IRTS-MLGD algorithm, and other decoding algorithms for the 64-ary (63,45) NB-TF-EG code, and the (63,45) RS code over GF(2^6) decoded with the HD-BM and the ASD-KV algorithms using BPSK over the AWGN channel.

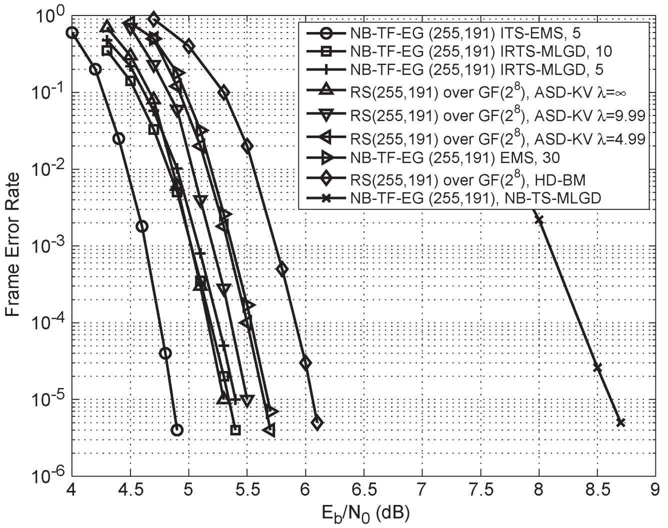

Fig. 7. Frame error rates of the IRTS-MLGD algorithm, and other decoding algorithms for the 256-ary (255,191) NB-TF-EG code, and the (255,191) RS code over GF(2^8) decoded with the HD-BM and the ASD-KV algorithms using BPSK over the AWGN channel.

Example 5: Consider the 256-ary (255,191) NB-TF-EG code given in Example 2 with $\epsilon=16$. Fig. 7 presents the FER performances of the NB-TF-EG code over the AWGN channel with BPSK signaling decoded using the ITS-EMS with 5 iterations, the IRTS-MLGD with 5 and 10 iterations, standard EMS with 30 iterations, and NB-TS-MLGD. The FER performances of the (255,191) RS code over GF(2^8) decoded using the HD-BM and the ASD-KV algorithm are also included.

From Table II, decoding with 5 and 10 iterations of the IRTS-MLGD require the integer operations on the order of 5.55×10^6 and 1.11×10^7, respectively. In contrast, the number of computations required for real numbers using the ITS-EMS with 5 iterations is on the order of 2.83×10^8. At the FER of 10^{-5}, the IRTS-MLGD with 5 iterations reduces the number of computations by 99% with a 0.5 dB in performance loss, compared to the ITS-EMS with 5 iterations. In addition, the

NB-TF-EG code decoded using the IRTS-MLGD with 10 iterations achieves a 3.2 dB coding gain over the NB-TS-MLGD, and outperforms the RS code by 0.7 dB when decoded using the HD-BM algorithm. Moreover, decoding the NB-TF-EG code with 10 iterations of IRTS-MLGD nearly exceeds the RS code decoded using the ASD-KV algorithm with $\lambda = \infty$, and outperforms the RS code decoded using the ASD-KV algorithm by 0.3 dB with $\lambda = 4.99$.

V. Conclusion

This paper presents a subclass of the TS-MLG decodable cyclic codes based on Euclidean geometries to non-binary cases, termed as NB-TF-EG codes. We also present two corresponding algorithms for decoding NB-TS-MLG decodable cyclic code. Our results demonstrate that the proposed iterative decoding algorithms are capable of efficient decoding of NB-TS-MLG decodable cyclic codes with Tanner graphs including a large number of short cycles of length 4. This is achieved by utilizing the orthogonal structure of the parity-check matrices of the codes to avoid performance degradation resulting from numerous short cycles of length 4. In addition, the proposed partial parallel decoding scheme strikes a reasonable balance between decoding speed and memory usage by incorporating a decomposition of the parity-check matrices of the codes. Simulation results demonstrate that the NB-TF-EG codes decoded using the proposed ITS-EMS algorithm in a small number of decoding iterations outperform the RS codes with similar lengths and rates decoded using either hard-decision or algebraic soft-decision decoding algorithms. Moreover, the IRTS-MLGD provides an alternative for ITS-EMS in decoding NB-TF-EG codes with far lower computational complexity.

Acknowledgment

The authors would like to acknowledge the contributions of Prof. Shu Lin to the development of this paper. The authors also thank the anonymous reviewers and the Editor for their helpful suggestions and careful reading which have improved the quality and presentation of the paper.

References

[1] L. D. Rudolph, "Geometric configuration and majority logic decodable codes," M.S. thesis, Univ. Oklahoma, Norman, OK, USA, 1964.

[2] T. Kasami, S. Lin, and W. W. Peterson, "Polynomial codes," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 6, pp. 807–814, Nov. 1968.

[3] T. Kasami and S. Lin, "On majority-logic decoding for duals of primitive polynomial codes," *IEEE Trans. Inf. Theory*, vol. IT-17, no. 3, pp. 322–331, May 1971.

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

[5] Y. Kou, S. Lin, and M. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[6] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[7] H. Tang, J. Xu, S. Lin, and K. Abdel-Ghaffar, "Codes on finite geometries," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 572–596, Feb. 2005.

[8] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algortihm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[9] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check code," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.

[10] L. Zhang, Q. Huang, and S. Lin, "Iterative algorithms for decoding a class of two-step majority logic decodable cyclic codes," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 416–427, Feb. 2011.

[11] H.-C. Chang, C.-L. Chen, and H.-C. Chang, "An iterative weighted reliability decoding algorithm for two-step majority-logic decodable cyclic codes," *IEEE Commun. Lett.*, vol. 17, no. 10, pp. 1980–1983, Oct. 2013.

[12] D. J. MacKay and M. C. Davey, "Evaluation of gallager codes for short block length and high rate applications," in *Proc. IMA Workshop Codes, Syst. Graph. Models*, 1999, pp. 113–130.

[13] L. Zeng *et al.*, "Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: A finite geometry approach," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 378–387, Mar. 2008.

[14] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[15] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[16] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[17] C.-Y. Chen, Q. Huang, C.-C. Chao, and S. Lin, "Two low-complexity reliability-based message-passing algorithms for decoding non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 58, no. 11, pp. 3140–3147, Nov. 2010.

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

[19] S. Lin, "Multifold Euclidean geometry codes," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 537–548, Jul. 1973.

[20] E. Boutillon and L. Conde-Canecia, "Bubble check: A simplified algorithm for elementary check node processing in extended min-sum nonbinary LDPC decoders," *Electron. Lett.*, vol. 46, no. 9, pp. 633–634, Apr. 2010.

[21] W. Gross, F. Kschischang, R. Koetter, and P. Gulak, "Applications of algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1224–1234, Jul. 2006.

[22] *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Std. 754, 1985.

Hsiu-Chi Chang received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2005 and 2007, respectively, where he is currently working toward the Ph.D. degree in electronics engineering. From January 2011 to July 2011, he was a visiting scholar in electrical engineering at the University of California, Davis, CA, USA.

His research interests include iterative algorithms, error control codes, and machine learning.

Hsie-Chia Chang received the B.S., M.S., and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1995, 1997, and 2002, respectively.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Corporation, working in the area of decoding architectures for Combo single chip. In February 2003, he joined the Faculty of Electronics Engineering Department, National Chiao Tung University, where he has been a Professor since August 2010. His research interests include algorithms and VLSI architectures in signal processing, especially for error control codes and cryptosystems. Recently, he has also committed himself for designing high code-rate ECC schemes for flash memory and multi-Gb/s chip implementations for wireless communications.

He served as the Associate Editor of IEEE Transactions on Circuits and System I: Regular papers since 2012. He also served as Technique Program Committee (TPC) member of IEEE A-SSCC 2011 and 2012. Dr. Chang was the recipient of the Outstanding Youth Electrical Engineer Award from Chinese Institute of Electrical Engineering in 2010, and the Outstanding Youth Researcher Award from Taiwan IC Design Society in 2011.