On Decoding of the (89, 45, 17) Quadratic Residue Code

Lin Wang, *Senior Member, IEEE,* Yong Li, *Student Member, IEEE,* Trieu-Kien Truong, *Fellow, IEEE,*
and Tsung-Ching Lin, *Member, IEEE*

***Abstract*—In this paper, Three decoding methods of the (89, 45, 17) binary quadratic residue (QR) code to be presented are hard, soft and linear programming decoding algorithms. Firstly, a new hybrid algebraic decoding algorithm for the (89, 45, 17) QR code is proposed. It uses the Laplace formula to obtain the primary unknown syndromes, as done in Lin et al.'s algorithm when the number of errors v is less than or equal to 5, whereas Gaussian elimination is adopted to compute the unknown syndromes when $v \geq 6$. Secondly, an appropriate modification to the algorithm developed by Chase is also given in this paper. Therefore, combining the proposed algebraic decoding algorithm with the modified Chase-II algorithm, called a new soft-decision decoding algorithm, becomes a complete soft decoding of QR codes. Thirdly, in order to further improve the error-correcting performance of the code, linear programming (LP) is utilized to decode the (89, 45, 17) QR code. Simulation results show that the proposed algebraic decoding algorithm reduces the decoding time when compared with Lin et al.'s hard decoding algorithm, and thus significantly reduces the decoding complexity of soft decoding while maintaining the same bit error rate (BER) performance. Moreover, the LP-based decoding improves the error-rate performance almost without increasing the decoding complexity, when compared with the new soft-decision decoding algorithm. It provides a coding gain of 0.2 dB at BER = 2×10^{-6}.**

***Index Terms*—Berlekamp-Massey algorithm, Gaussian elimination, quadratic residue code, Chase algorithm, linear programming.**

I. Introduction

QR codes, first introduced by Prange [1] in 1958, are a nice family of cyclic codes which have code rates greater than or equal to 1/2 and generally have large minimum distances so that most of the known QR codes are the best-known codes. It is well known that there are 11 binary QR codes with code length less than 100; that is, 7, 17, 23, 31, 41, 47, 71, 73, 79, 89 and 97. A series of different algebraic decoding algorithms for these QR codes except for the (89, 45, 17) code have been proposed by the authors given in [2]-[11], but the algebraic decoding scheme for the (89, 45, 17) QR code is not available until 2008. Such a (89, 45, 17) QR code whose error-correcting capacity is up to 8 can be constructed in the finite field *GF*(2^{11}). Due to this small size of the finite field, the computational complexity can be dramatically reduced, and therefore, this code is considered to be one of the best in the family of the binary QR codes.

In the past decades, the most widely used methods for decoding binary QR codes are the Sylvester resultant [6], [8] or Gröbner basis methods [12]. These methods can be used to solve the Newton identities that are nonlinear and multivariate equations with high degree, so the calculations of identities require very high complexity when the weight of the occurred error becomes large. Moreover, different QR codes use different sets of conditions to calculate the error locations. As a sequence, a total enumeration of all conditions is impracticable for software implementation. Recently, using the inverse-free Berlekamp-Massey (BM) algorithm [13]-[15], an algebraic decoding algorithm for QR codes [9] has been applied to determine the error-locator polynomial. These facts also lead to designing the algebraic decoders for many other binary QR codes of lengths up to 113, see [11] and [16], except for the QR codes of lengths 31, 73 and 89.

Recently, Truong et al. [17] modified the previous algebraic decoding algorithms for decoding the (89, 45, 17) QR code, based on determining successfully the unknown syndromes and modifying the aforementioned inverse-free BM algorithm. In this new decoding algorithm, first, compute the unknown syndromes, and determine the error-locator polynomial by using the inverse-free Berlekamp-Massey (BM) algorithm. Then apply the Chien search [18] to find the roots of the error-locator polynomial. However, since a possibly distinct pair of primary unknown syndromes is not always unique, the inverse-free BM algorithm needs to be repeated recursively until the correct error-locator polynomial is obtained, thereby requiring very high computational complexity if the splitting field of $x^n - 1$ is very large. Towards this end, in 2010, Lin et al. proposed a fast and efficient algorithm [19] to determine the primary unknown syndromes, which significantly reduced the decoding complexity in terms of CPU time when compared with the previous algorithm [17]. In addition, this efficient algorithm together with the Chase-II algorithm [21] first successfully achieved the soft-decision decoding of the (89,

Paper approved by E. Ayanoglu, the Editor for Communication Theory and Coding Applications of the IEEE Communications Society. Manuscript received April 27, 2012; revised August 17, 2012 and September 26, 2012.

L. Wang is with the Department of Communication Engineering, Xiamen University, Xiamen, 361005 China (e-mail: wanglin@xmu.edu.cn).

Y. Li was with the Department of Communication Engineering, Xiamen University, Xiamen, 361005 China. He is now with the Key Lab of Mobile Communication in Chongqing, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, 400065 China (e-mail: liyongxmu@gmail.com).

T.-K. Truong and T.-C. Lin are with the Department of Information Engineering, I-Shou University, Kaohsiung Country 84001, Taiwan (e-mail: {truong, joe}@isu.edu.tw). T.-K. Truong is also with the Department of Computer Science and Engineering, National Sun Yat-sen University, Taiwan.

This work was supported in part by China NSF under Grant No. 60972053, and in part by Taiwan NSF under Grant No. 01010102.

Digital Object Identifier 10.1109/TCOMM.2012.09.120287

45, 17) QR code. However, since there are at most $2^{\lfloor d/2 \rfloor}$ error patterns considered in the process of decoding a codeword by the Chase algorithm, where the minimum Hamming distance d is equal to 17, it still requires high computational complexity needed in Lin et al.'s soft-decision decoder.

Maximum likelihood (ML) decoding of linear block codes, which is non-deterministic polynomial-time hard (NP-hard) [22], can be described as an integer programming (IP) problem. Linear programming (LP) decoding, as an approximation to ML decoding, was first introduced by Feldman et al. [23]. In the original formulation of LP decoding in [23], the number of constraints is exponential in the maximum check node degree. As a result, the computational complexity may be prohibitively high even for some small check degrees. To overcome this, recently, an adaptive linear programming (ALP) decoder was introduced [24], which reduces the number of constraints by adding only useful ones in an adaptive and selective way. Moreover, the performance of LP decoding can be improved by reducing the feasible solution space by adding more linear constraints generated by redundant parity checks (RPC) [24], [25], [26], [27]. Although most of references about LP decoding concentrate on LDPC codes, the advanced LP algorithms in [25], [27] also worked very well when used to decode BCH codes and Golay codes.

In this paper, a new hybrid scheme is proposed so as to calculate the primary unknown syndromes more rapidly than Lin et al.'s algorithm. A new algebraic hard-decision decoding algorithm that utilizes the hybrid scheme is approximately 13 times faster than Lin et al.'s decoder when a weight-8 error pattern occurs. In order to speed up the soft-decision decoding, a sufficient optimality condition is introduced to quickly terminate the Chase-II algorithm. The proposed hard decoding algorithm in conjunction with the modified Chase algorithm yields a new and fast soft-decision decoding algorithm of the (89, 45, 17) QR code. For fair comparison, the improved version of Chase-II algorithm based on the sufficient optimality condition instead of the conventional Chase-II algorithm combined with Lin et al.'s hard decoding algorithm is used to simulate the soft-decision decoding of the (89, 45, 17) QR code. Computer simulations show that the proposed new soft-decision decoder dramatically improves the decoding speed of approximate 7 times compared with the one whose kernel is Lin et al.'s algorithm. Moreover, the LP decoding performance of the (89, 45, 17) QR code is also investigated. Simulation results show that using the powerful cutting-plane technique, the LP decoding even provides more coding gain than the algebraic soft decoding algorithm with comparable decoding complexity.

This paper is organized as follows: The background of the binary QR codes is introduced in Section II. Section III proposes a new hard-decision decoding algorithm of the (89, 45, 17) QR code. In Section IV, a sufficient optimality condition described as a theorem needed in quickly terminating the Chase-II algorithm is given and proved. In Section V, LP decoding is briefly introduced. Simulation results are presented in Section VI for the hard-decision, soft-decision and LP decoding of the (89, 45, 17) QR code. Finally, this paper concludes with a brief summary in Section VII.

II. Terminology and Background of the QR Codes

Let n be a prime number of the form $n = 8l \pm 1$, where l is a positive integer. The set Q_n of quadratic residues modulo n is the set of nonzero squares modulo n. That is

$$Q_n = \{i | i \equiv j^2 \text{mod}\, n \qquad \text{for } 1 \leq j \leq n-1\}. \qquad (1)$$

Let m be the smallest positive integer such that n divides $2^m - 1$ and let α be a primitive element of the finite field $GF(2^m)$, such that each nonzero element of $GF(2^m)$ can be expressed as a power of α. Then the element $\beta = \alpha^u$, where $u = (2^m - 1)/n$, is a primitive nth root of unity in $GF(2^m)$. An (n, k, d) QR code which has the minimum distance d is a cyclic code with the generator polynomial $g(x)$ of the form $g(x) = \prod_{i \in Q_n} (x - \beta^i)$.

For an (n, k, d) QR code, an error pattern is said to be correctable if its weight is less than or equal to the error-correcting capacity $t = \lfloor (d-1)/2 \rfloor$, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. Now, let the codeword $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ be transmitted through a noisy channel. Also, let $e(x) = e_0 + e_1 x + \cdots + e_{n-1} x^{n-1}$ and $r(x) = r_0 + r_1 x + \cdots + r_{n-1} x^{n-1}$ be the error pattern occurred and the received vector, respectively. Then, the received word has the form $r(x) = c(x) + e(x)$.

The set of known syndromes computed by evaluating $r(x)$ at the roots of $g(x)$ is given by

$$S_i = r(\beta^i) = e(\beta^i), \qquad i \in Q_n \qquad (2)$$

Assume that there are v errors occurred in the received word $r(x)$. Then, the error pattern has v nonzero terms, namely, $e(x) = x^{l_1} + x^{l_2} + \cdots + x^{l_v}$, where $0 \leq l_1 < l_2 < \cdots < l_v \leq n-1$. The syndrome S_i can be written as $S_i = X_1^i + X_2^i + \cdots + X_v^i$, where $X_j = \beta^{l_j}$ for $1 \leq j \leq v$ are said to be the error locators. If i is not found in the set Q_n, the syndrome S_i is, what is called, the unknown syndrome.

For binary QR codes, there is an obvious relation between syndromes, namely, $S_{2i} = S_i^2$, with indices modulo n. It is well known [17] that the generator polynomial of the (89, 45, 17) QR code is reducible over *GF*(2), and all the known syndromes (resp., unknown syndromes) can be expressed as some powers of S_1, S_5, S_9, and S_{11} (resp., S_3, S_{13}, S_{19}, and S_{33}).

Suppose that v errors occur in the received word. The error-locator polynomial $\sigma(x)$ is defined to be a polynomial of degree v. One way to decode the QR code is to determine $\sigma(x)$ and the Chien search is then applied to find the roots of $\sigma(x)$. The inverse-free BM algorithm is known to be the most efficient method for determining the error-locator polynomial. The error locations are given by the inverse of the roots of $\sigma(x)$ provided they are no more than the error-correcting capacity t. In order to use the inverse-free BM algorithm to decode the QR code up to eight errors, i.e., $t = 8$, one needs to find, in sequence, the first $2t = 16$ consecutive syndromes $S_1, S_2, \ldots, S_{16}$. However, only the syndromes $S_1, S_2, S_4, S_5, S_8, S_9, S_{10}, S_{11}$ and S_{16} can be calculated directly from $r(x)$; the others, namely, $S_3, S_6, S_7, S_{12}, S_{13}, S_{14}$ and S_{15} not determined directly from

$r(x)$ are unknown syndromes. Obviously, these known syndromes (resp., unknown syndromes) can be expressed as some powers of S_1, S_5, S_9, S_{11} (resp., S_3 and S_{13}) that are the so-called primary known syndromes (resp., unknown syndromes) of the QR code.

Using a technique similar to that given in [11], a strategy in [17] was developed to obtain each of the needed primary unknown syndromes. It is based on solving the roots of the equation in a primary unknown syndrome with the coefficients that can be expressed in terms of certain primary known syndromes. Therefore, all of the unknown syndromes can be calculated once the values of the primary unknown syndromes are determined. The following is a brief review of the technique mentioned in [11] for the (89, 45, 17) QR code.

Assume that v errors occur in the received word. Let $\boldsymbol{I} = \{i_1, i_2, \ldots, i_{v+1}\}$ and $\boldsymbol{J} = \{j_1, j_2, \ldots, j_{v+1}\}$ denote two subsets of $\{0, 1, 2, \ldots, 88\}$, respectively. These index subsets can be found by an explicit use of the fast algorithm in [9]. Next, consider the matrix $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$ of size (v+1)× (v+1) given by

$$\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J}) = \begin{bmatrix} S_{i_1+j_1} & S_{i_1+j_2} & \cdots & S_{i_1+j_{v+1}} \\ S_{i_2+j_1} & S_{i_2+j_2} & \cdots & S_{i_2+j_{v+1}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i_{v+1}+j_1} & S_{i_{v+1}+j_2} & \cdots & S_{i_{v+1}+j_{v+1}} \end{bmatrix} \tag{3}$$

where the summation of the indices of S_i's is modulo n and the rank of $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$ is at most v, which, in turn, implies

$$\det \boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J}) = 0. \tag{4}$$

If all of the unknown syndromes among the entries of $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$ given in (3) can be expressed as some powers of one of the primary unknown syndromes, say S_r, and if det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) is a nonzero polynomial in S_r, then the actual value of S_r is one of the roots of (4). In other words, during the decoding process, one is able to calculate the value of S_r from the known syndromes. The determination of the primary unknown syndromes S_3 and S_{13} of the (89, 45, 17) QR code will be shown in Section III.

III. New Hard-Decision Decoding of the (89, 45, 17) QR Code

Based on the generator polynomial of the (89, 45, 17) QR code, there exists the mapping relationship between each error pattern with a weight less than or equal to eight and its primary known syndromes S_1, S_5, S_9 and S_{11}. Recently, according to the mapping relationship, Truong et al. developed the algebraic decoding technique [17], and Lin et al. went on to propose a fast algebraic hard-decision (HD) algorithm [19] which significantly reduced the decoding complexity in terms of CPU time when compared with the decoding algorithm in [17].

By simulating and analyzing the above decoding algorithm developed by Truong et al., one observes that calculating the unknown syndromes, which includes the calculation of the polynomial det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) and searching the roots of det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) = 0, requires high computational complexity, whereas both the inverse-free BM algorithm and Chien's search for finding the roots of the error-locator polynomial require very low computational complexity. Therein, the inverse-free BM algorithm and Chien's search require O($3t$) and O($v\rho$) operations for v-error case, respectively, where $v \leq t$ and $\rho = 89$ corresponding to the 89 components of a codeword. However, calculating the unknown syndromes requires O($q \cdot v \cdot (v+1)!$) operations, where $q = 2048$ corresponding to the 2048 elements in the field $GF(2^{11})$. In Lin et al.'s fast algebraic HD decoding algorithm, some excellent subsets $\boldsymbol{I}$ and $\boldsymbol{J}$ resulting in low degree polynomials are chosen in a manner such that the unknown syndrome exactly appears once in given $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$ for each $1 \leq v \leq 5$ case, whereas, when $v \geq 6$, more than two nonzero polynomials det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) are first constructed by different combinations of $\boldsymbol{I}$ and $\boldsymbol{J}$. Then the Euclid algorithm is used to find the greatest common divisor $F(S_r^{(v)})$ of those polynomials provided there exists a common divisor, where the notation "$S_r^{(v)}$" indicates that the formulae obtained are valid for v-error case only. If $F(S_r^{(v)})$ has degree one, then the unknown syndrome can be directly determined by unique root of $F(S_r^{(v)})$ = 0; otherwise, all the roots of $F(S_r^{(v)})$ = 0 are solved by Chien's search. Thus, the number of operations needed in Lin et al.'s algorithm is O($2v \cdot (v+1)! + \mu + \nu^2 + q\lambda$), where λ is the degree of $F(S_r^{(v)})$, μ and ν are the degrees of two polynomials det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$), respectively, and $\mu \geq \nu$. However, in some cases, only one nonzero polynomial is obtained by the subsets $\boldsymbol{I}$ and $\boldsymbol{J}$. As a result, Euclid's algorithm cannot be used and the Chien search is the only method to find the roots of det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) = 0. For these worst cases, the HD algorithm requires a complexity of O($q \cdot v \cdot (v+1)!$).

It is well-known that Gaussian elimination is a very efficient method to compute the determinant of a $N \times N$ matrix and its asymptotic complexity is equal to O(N^3). Thus, assume that v errors occur, the roots of det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) = 0 can be obtained by substituting each element in the field $GF(2^{11})$ into the matrix $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$ and checking whether its determinant is zero or not, which requires O($q(v+1)^3$) operations. As a result, this Gaussian elimination method is more efficient than Lin et al.'s method when v is large (v = 6, 7, 8). Moreover, in this method, the nonzero polynomials det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) don't need to be stored. As for the case $v \leq 5$, solving det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) = 0 directly is available due to the fact that the nonzero polynomials, derived from the given subsets $\boldsymbol{I}$ and $\boldsymbol{J}$ in [19], only include two monomials because the unknown syndrome exactly appears once in given $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$. It only requires O($(v+1)!$) operations. Hence, an algorithm, called the hybrid unknown syndrome calculation (HUSC) algorithm proposed in this paper can be utilized to compute the primary unknown syndromes with less computational complexity.

In order to obtain the polynomial det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) with low degrees and guarantee that the primary unknown syndromes can be determined for any error pattern of weight less than or equal to eight, the subsets $\boldsymbol{I}$ and $\boldsymbol{J}$ provided in [19] are applied to construct the matrix $\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$. Suppose that v errors occur, there are two sub-cases corresponding to S_3^v and S_{13}^v, respectively.

Now, the HUSC algorithm is depicted in detail as follows:

1) Initially, γ is the total number of distinct pairs $(\boldsymbol{I}, \boldsymbol{J})$ in

each sub-case and $i = 1$.

2) If $i \leq \gamma$, obtain the matrix $\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$ from Case v; otherwise, stop.
3) If $v \leq 5$, compute det($\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$) by the Laplace formula; otherwise, go to step 5.
4) If det($\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$) is a nonzero polynomial, find all the roots by solving the equation det($\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$) = 0 directly and stop; otherwise, let $I = I + 1$ and return back to step 2.
5) Substitute each element a in the finite field $GF(2^{11})$ into the matrix $\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$, then check whether det($\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$) = 0 or not by Gaussian elimination. If all the elements satisfy the equation det($\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$) = 0, which means that det($\boldsymbol{S}(\boldsymbol{I}_v^i, \boldsymbol{J}_v^i)$) is a zero polynomial, then let $i = i+1$ and go back to step 2. Otherwise, stop.

Finally, one can obtain all possible pairs $(S_3^{(v)}, S_{13}^{(v)})$ according to distinct combinations of the possible $S_3^{(v)}$'s and $S_{13}^{(v)}$'s determined by the above-mentioned algorithm. It should be noted that for any subcase, once some roots of det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) = 0 are found, the HUSC algorithm will be terminated no matter whether all the subsets $\boldsymbol{I}$, $\boldsymbol{J}$ have been used or not. However, in the Lin et al.'s algorithm, the remainder of the subsets $\boldsymbol{I}$ and $\boldsymbol{J}$ still need to be considered unless a degree-1 polynomial of S_3 (or S_{13}) is obtained or all the sets $\boldsymbol{I}$ and $\boldsymbol{J}$ are exhausted, even if some roots of det($\boldsymbol{S}(\boldsymbol{I}, \boldsymbol{J})$) = 0 are solved. As a result, this new method determines the unknown syndromes S_3 and S_{13} more efficiently, thereby reducing the total decoding time. It is verified by computer simulations in Section VI.

The new hard-decision decoding algorithm, called Algorithm 1 and listed in Appendix A, has the almost same procedure as the ones in [17], [19]. The only difference depends on how to determine the unknown syndromes.

IV. Soft-decision Decoding Based on the Chase Algorithm

It is well-known that Chase has proposed three algorithms, namely, Chase-I, Chase-II, and Chase-III algorithm, see [21]. Among them, the most widely used one is the Chase-II algorithm due to the trade-off between performance and complexity, and there are at most $2^{\lfloor d/2 \rfloor}$ error patterns considered by this decoding algorithm.

Chase-II decoding is very time-consuming for the (89, 45, 17) QR code because the hard-decision decoder may run repeatedly 256 times for decoding a received sequence. Towards this end, Taipale et al. developed a modification of the generalized-minimum-distance decoding algorithm [28], which can be used in Chase-II decoding as a stopping condition in order to reduce the average decoding complexity. Based on two decoded codewords, a sufficient optimality condition without derivation (see, for example, Lin et al.'s classic book [29]) can be used to terminate the Chase decoding process more rapidly. In this paper, this sufficient optimality condition is redescribed as a theorem and its corresponding proof is derived mathematically.

Suppose a QR codeword is transmitted through an AWGN channel. Let C be a set of codewords and $\boldsymbol{v} = (v_0, v_1, \ldots, v_{n-1})$ be a codeword in the set C. Also, let $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$, where $c_i = 2v_i - 1$, $0 \leq i < n$, and $\boldsymbol{r} = (r_0, r_1, \ldots, r_{n-1})$ be the corresponding BPSK signal sequence and the soft received sequence, respectively. Next, the hard decoding vector is denoted as $\boldsymbol{z} = (z_0, z_1, \ldots, z_{n-1})$ which satisfies the following formula:

$$z_i = \begin{cases} 0 & \text{for } r_i < 0, \\ 1 & \text{for } r_i \geq 0. \end{cases} \tag{5}$$

Two index sets are defined as

$$D_0(\boldsymbol{v}) \triangleq \{i : v_i = z_i \quad \text{with } 0 \leq i < n\} \tag{6}$$

$$\begin{aligned} D_1(\boldsymbol{v}) &\triangleq \{i : v_i \neq z_i \quad \text{with } 0 \leq i < n\} \\ &= \{0, 1, \ldots, n-1\} \setminus D_0(\boldsymbol{v}) \end{aligned} \tag{7}$$

Define the correlation discrepancy between $\boldsymbol{r}$ and $\boldsymbol{v}$ as

$$\lambda(\boldsymbol{r}, \boldsymbol{v}) = \lambda(\boldsymbol{r}, \boldsymbol{c}) \triangleq \sum_{i: r_i \cdot c_i < 0} |r_i|, \tag{8}$$

and define

$$n(\boldsymbol{v}) = |D_1(\boldsymbol{v})|\,, \tag{9}$$

where $|\cdot|$ denotes the cardinality of a set.

With the aid of the definition of $\boldsymbol{c}$ and Formula (5), the expression $r_i \cdot c_i < 0$, where $0 \leq i < n$, is valid if and only if $z_i \neq v_i$. Thus, (8) can be written as

$$\lambda(\boldsymbol{r}, \boldsymbol{v}) = \sum_{i \in D_1(\boldsymbol{v})} |r_i|. \tag{10}$$

The index set $D_0(\boldsymbol{v})$ has $n - n(\boldsymbol{v})$ elements, which can be arranged in the order of the reliability measurements of received symbols as follows:

$$D_0(\boldsymbol{v}) \triangleq \{l_1, l_2, \ldots, l_{n-n(\boldsymbol{v})}\}, \tag{11}$$

where $|r_{l_i}| < |r_{l_j}|$ for $1 \leq i < j \leq n - n(\boldsymbol{v})$.

Define the set of first j elements in $D_0(\boldsymbol{v})$ arranged as

$$D_0^{(j)}(\boldsymbol{v}) = \{l_1, l_2, \ldots, l_j\}, \tag{12}$$

where $D_0^{(j)}(\boldsymbol{v}) \triangleq \varnothing$ for $j \leq 0$ and $D_0^{(j)}(\boldsymbol{v}) \triangleq D_0(\boldsymbol{v})$ for $j \geq n - n(\boldsymbol{v})$.

The results without proof given in [29] is summarized as a theorem as follows:

Theorem 1: Let $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ be two codewords in C, and let $\boldsymbol{v}$ be the one with smaller correlation discrepancy between them. Define $\delta_1 \triangleq d - n(\boldsymbol{v}_1)$, $\delta_2 \triangleq d - n(\boldsymbol{v}_2)$, $D_{00} \triangleq D_0(\boldsymbol{v}_1) \cap D_0(\boldsymbol{v}_2)$, $D_{01} \triangleq D_0(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_2)$, $D_{10} \triangleq D_1(\boldsymbol{v}_1) \cap D_0(\boldsymbol{v}_2)$, and $D_{11} \triangleq D_1(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_2)$. In general, assume $\delta_1 \geq \delta_2$. We define $I(\boldsymbol{v}_1, \boldsymbol{v}_2) \triangleq (D_{00} \cup D_{01}^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)})^{(\delta_1)}$, where $X^{(q)}$ represents the first q indexes of the index set X arranged, and $G(\boldsymbol{v}_1, d; \boldsymbol{v}_2, d) \triangleq \sum_{i \in I(\boldsymbol{v}_1, \boldsymbol{v}_2)} |r_i|$. If

$$\lambda(\boldsymbol{r}, \boldsymbol{v}) \leq G(\boldsymbol{v}_1, d; \boldsymbol{v}_2, d), \tag{13}$$

then $\boldsymbol{v}$ is the maximum likelihood codeword of $\boldsymbol{r}$.

Proof: For detailed proof, see Appendix B. ■

The flowchart of the new soft-decision decoding algorithm of the (89, 45, 17) QR code adapting Algorithm 1 to the Chase-II algorithm with the sufficient optimality condition (13), called Algorithm 2, is depicted in Fig. 1.

The flowchart of the new soft-decision decoding algorithm of the (89, 45, 17) QR code adapting Algorithm 1 to the Chase-II algorithm with the sufficient optimality condition (13), called Algorithm 2, is depicted in Fig. 1.

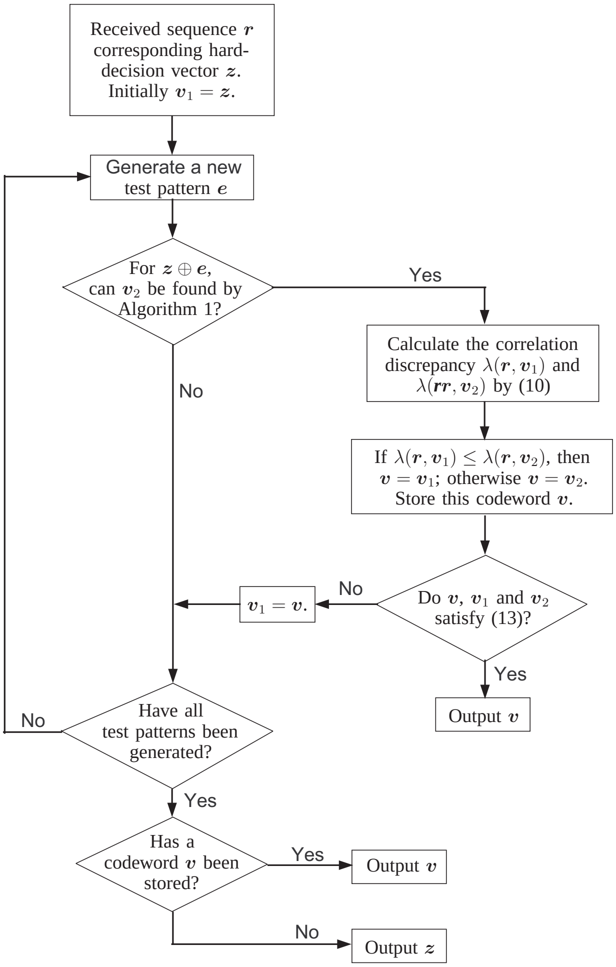
Received sequence $\boldsymbol{r}$ corresponding hard-decision vector $\boldsymbol{z}$. Initially $\boldsymbol{v}_1 = \boldsymbol{z}$.
Generate a new test pattern $\boldsymbol{e}$
For $\boldsymbol{z} \oplus \boldsymbol{e}$, can $\boldsymbol{v}_2$ be found by Algorithm 1?
Yes
No
Calculate the correlation discrepancy $\lambda(\boldsymbol{r}, \boldsymbol{v}_1)$ and $\lambda(\boldsymbol{rr}, \boldsymbol{v}_2)$ by (10)
If $\lambda(\boldsymbol{r}, \boldsymbol{v}_1) \leq \lambda(\boldsymbol{r}, \boldsymbol{v}_2)$, then $\boldsymbol{v} = \boldsymbol{v}_1$; otherwise $\boldsymbol{v} = \boldsymbol{v}_2$. Store this codeword $\boldsymbol{v}$.
Do $\boldsymbol{v}$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ satisfy (13)?
No
$\boldsymbol{v}_1 = \boldsymbol{v}$.
Yes
Output $\boldsymbol{v}$
No
Have all test patterns been generated?
Yes
Has a codeword $\boldsymbol{v}$ been stored?
Yes
Output $\boldsymbol{v}$
No
Output $\boldsymbol{z}$

Fig. 1. Flowchart of Algorithm 2.

V. Linear Programming Decoding

The ML decoding for any binary linear code can be written as an optimization problem. That is,

$$\min \boldsymbol{\gamma}^T \boldsymbol{u} \qquad \text{s.t. } \boldsymbol{u} \in \text{conv}(C) \tag{14}$$

Here, $\boldsymbol{\gamma}$ is the cost vector obtained by the log-likelihood ratios $\gamma_i = \log(P(y_i|u_i = 0)/P(y_i|u_i = 1))$ for a given channel output y_i, and conv(C) is the codeword polytope.

As an approximation to ML decoding, Feldman et al. [23] relaxed the codeword polytope onto the so-called fundamental polytope so as to convert (14) into a linear programming problem. This polytope includes both integral and nonintegral vertices, with the former corresponding exactly to the codewords of C. Thus, the LP relaxation yields the ML certificate property; that is, if the LP decoder outputs an integral solution, it is guaranteed to be an ML codeword.

In Feldman's original linear programming problem, the total number of constraints, and hence the computational complexity of building and solving the LP problem is exponential in terms of the maximum check degree $d_i^{\max}$. This results in that the explicit description of the fundamental polytope via parity inequalities is inapplicable for high-density codes. To overcome this, Feldman et al. proposed an equivalent formulation which requires O(N^3) constraints [23]. More recently, an alternative polytope, that has size linear in the code length and the maximum check node degree, was proposed [30], [31].

To reduce the computational complexity, Taghavi and Siegel [24] proposed an adaptive approach, called adaptive linear programming (ALP) decoding, which can be applicable to high density codes instead of the direct implementation of the original LP decoding algorithm. It should be pointed out that the ALP decoder doesn't yield an improvement from the view of frame error rate. However, it has a very positive effect on the decoding time because of converging with fewer constraints than the original LP decoder.

Recently, Tanatmis et al. developed a separation algorithm (SA) to improve the error-correcting performance of LP decoding [25], which is abbreviated as SALP in the after-mentioned contents. Zhang and Siegel [26] proposed a novel decoder combining the new adaptive cut-generating (ACG) algorithm with the ALP algorithm, called ACG-ALP decoder, and its two variations called ACG-MALP-B, ACG-MALP-C decoders, respectively. Here we omit the details of the SALP decoder, the ACG-ALP decoder and its variations due to limited space.

VI. Simulation Results

Simulations were conducted using the C programming language on a 2.93 GHz Intel Core i3 Processor to determine the decoding complexity of Algorithm 1. For comparison purpose, the simulation for Lin et al.'s algorithm was also conducted. The fast decoding algorithm [19] decoded an eight-error pattern perfectly with an average speed of 1.83s per codeword, whereas the proposed Algorithm 1 decoded an eight-error pattern only in 0.141s. Such an improvement is very important in low SNR region, where many errors with weight larger than or equal to 7 occur. The detailed comparison results for the v-error case, where $1 \leq v \leq 8$, are shown in Table I. Simulation results of the bit error rate (BER) via a bit-energy-to-noise-spectral-density ratio (E_b/N_0) which takes seven values from 0 dB to 6 dB for Lin et al.'s algorithm and Algorithm 1 are illustrated in Fig. 2. Upon the inspection of this figure, one observes that the BER curves are the same for both algorithms. Obtaining the BER curves for Algorithm 1 and Lin et al.'s algorithm require 118s and 667s, respectively. In other words, Algorithm 1 is about 5.7 times faster than Lin et al.'s algorithm, which is consistent with the complexity analysis in Section III.

For further comparison, Algorithm 2 and the soft-decision decoding algorithm combining Lin et al.'s hard decoding algorithm with the modified Chase-II algorithm based on the aforementioned sufficient optimality condition, called Algorithm 3, are also used to simulate the soft-decision decoding of the (89, 45, 17) QR code. Its corresponding computational time for each E_b/N_0 is listed in Table II. It can be seen that the proposed new soft decoder dramatically accelerates approximately 7 times when compared with the one using Algorithm 3. With SNR increasing, the former decoder converges more quickly than the latter one, which coincides with the fact that maximum likelihood codewords can be found more easily and rapidly in high SNR region because the number of errors is reduced when the value of SNR is increased.

TABLE I
COMPARISON OF COMPUTATIONAL TIME (SECONDS) BETWEEN ALGORITHM 1 AND LIN ET AL.'S DECODING ALGORITHM

v-error occur	The proposed decoding algorithm	Lin et al.'s decoding algorithm
1	0.000195	0.000200
2	0.000359	0.000358
3	0.000542	0.000539
4	0.000748	0.000784
5	0.00152	0.00173
6	0.00850	0.0160
7	0.0641	0.212
8	0.141	1.830

TABLE II
COMPARISON OF COMPUTATIONAL TIME (SECONDS) BETWEEN ALGORITHM 2 AND ALGORITHM 3

E_b/N_0(dB)	Algorithm 2	Algorithm 3
0	3400	30198
1	4461	31193
2	8547	57599
3	15132	110631
4	35149	244021
4.5	54890	379022
Total time	121579	852664

As illustrated in Fig. 2, using the new soft-decision algorithm proposed in this paper, namely Algorithm 2, we obtain the BER vs. E_b/N_0 curve as the same as that of Lin et al.'s algorithm for the soft-decision decoder for the (89, 45, 17) QR code in AWGN with BPSK modulation. Thus, the validity of the new algorithm is verified by computer simulations. It is obvious that at BER of 10^{-5}, the curve of hard-decision decoding of the (89, 45, 17) QR code is approximately 1.5 dB away from that of the soft-decision case.

In this paper, we also investigate the error-rate performance of the (89, 45, 17) QR code when the ALP decoder, the SALP decoder, the ACG-ALP decoder and its variation the ACG-MALP-B decoder are used, respectively. As shown in Fig. 3, the ACG-ALP decoder yields the best performance among all the LP-based decoders. At BER = 2×10^{-5} and BER = 2×10^{-6}, it provides 0.7 dB and 0.2 dB coding gain, respectively, when compared with the SALP decoder and the algebraic soft-decision decoder. Furthermore, the ACG-MALP-B decoder performs almost as well as the ACG-ALP decoder, but the ALP decoder is inferior to all other three decoders.

Fig. 4 compares the frame error rate (FER) performance of aforementioned decoders. At FER = 2×10^{-4}, the ACG-ALP decoder outperforms the SALP decoder about 0.7 dB, and has 0.25 dB coding gain compared with the ACG-MALP-B decoder at FER of 3×10^{-5}. In the whole simulated SNR range, the ACG-ALP decoder is slightly superior to the decoder based on Algorithm 2.

Fig. 5 compares the average decoding time of different LP decoding algorithms. We implement these algorithms by C++ code, and use the Simplex method from the open-source GNU Linear Programming Kit (GLPK) as our LP solver [32]. It seems to be reasonable because all the LP based decoders use the same LP solver. The simulation time is averaged over the total number of transmitted codewords required for each decoder to collect 100 erroneous codewords. In the high SNR region, compared with the SALP decoder, the ACG-ALP decoder and the ACG-MALP-B decoder not only correct more errors, but also decode the codewords more rapidly.

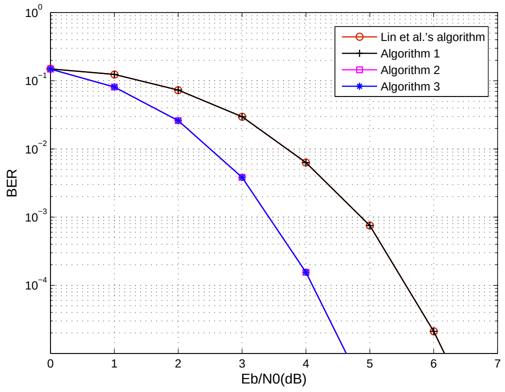

Fig. 2. BER performance of the (89, 45, 17) QR code in AWGN channel.

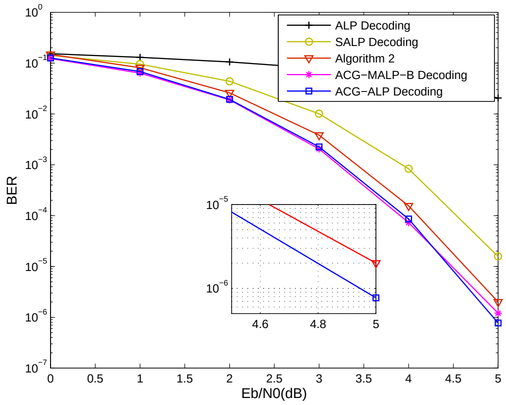

Fig. 3. BER performance comparison between Algorithm 2 and LP-based algorithms for the (89, 45, 17) QR code.

In the following, we just compare the algorithm complexity between the ACG-ALP decoder and the algebraic soft-decision decoder based on Algorithm 2, because only the ACG-ALP decoder among all the LP-based decoders outperforms the algebraic soft-decision decoder both in terms of BER and FER.

It is difficult to compare the detailed arithmetic operations between the ACG-ALP decoder and the decoder based on Algorithm 2. The algorithm complexity of the ACG-ALP decoder is mainly decided by two factors: the number of iterations (i.e., the number of LP problems) of decoding a codeword, and the complexity of the LP problem in each iteration. In our simulations, the simplex algorithm is used as the LP solver. It needs $2n \sim 3n$ pivot steps typically [33] and requires O(n^3) operations in each pivot step, where n is the number of primal variables, i.e., the code length. Thus the ACG-ALP decoder requires O($c \cdot n^4$) operations to decode a codeword, where c is the average number of iterations listed

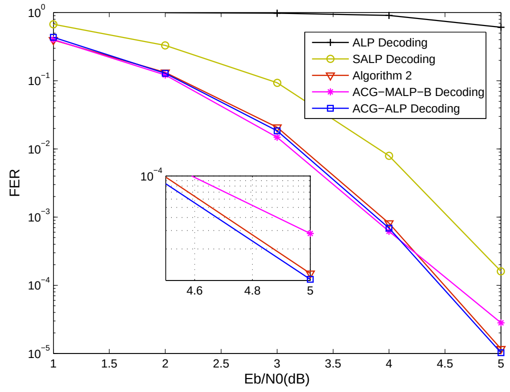

Fig. 4. FER performance comparison between Algorithm 2 and LP-based algorithms for the (89, 45, 17) QR code.

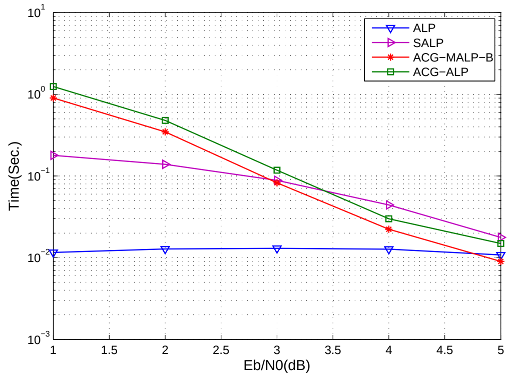

Fig. 5. Average simulation time for decoding one codeword of the (89, 45, 17) QR code.

in Table III. As seen in Section III, when v errors occur and Gaussian elimination is used, the complexity of Algorithm 1 is O($q \cdot (v+1)^3 + 3t + v\rho$), where $q = 2048$, $6 \leq v \leq t$, $t = 8$, $\rho = 89$. In low SNR region, the weights of most of error patterns are larger than t, which violates the condition (13). As a result, Algorithm 2 cannot be terminated until it exhausts all the 2^t possible error patterns, namely, hard-decision decoding is executed 2^t times. Hence, the complexity of the decoder using Algorithm 2 is O($(q \cdot (t+1)^3 + 3t + t\rho) \cdot 2^t$). However, in the high SNR region, fewer errors occur and Algorithm 2 can often be terminated quickly by (13) because of finding the ML codeword. Thus, its complexity drops down to O($((v+1)! + 3t + v\rho) \cdot k$), where $v \leq 5$ and $k \ll 2^t$. It should be noted that the ACG-ALP decoder operates on the real field, and the algebraic soft-decision decoder works on the finite field $GF(2^{11})$. In the finite field, the arithmetic operations refer to look-up table and module operations additionally. In our simulations, the ACG-ALP decoder is faster than the algebraic soft-decision decoder in the low SNR region. However, in the high SNR region, the soft-decision decoder is slightly faster. Thus, the total decoding complexity of them are comparable. In addition, the ACG-ALP decoder can be accelerated by other powerful LP solver, say, interior-point solver.

TABLE III
AVERAGE NUMBER OF ITERATIONS FOR DECODING A CODEWORD BY USING THE ACG-ALP DECODING ALGORITHM

E_b/N_0(dB)	Average number of iterations
1.0	50
2.0	26
3.0	13
4.0	8
5.0	6

VII. Conclusion

In this paper, a fast hybrid algebraic decoding algorithm of the (89, 45, 17) QR code is proposed. When v = 6, 7 and 8, it results in 46.9%, 69.8% and 92.3%, respectively, reduction of decoding complexity in terms of CPU time while maintaining the same BER. This improvement is very valuable in the low SNR region in which many error patterns whose weights are large than or equal to 7 occur. Therefore, such a decoding algorithm is more suitable for the soft-decision decoding of the (89, 45, 17) QR code than Lin et al.'s algorithm. Using this new algebraic decoding algorithm together with the Chase-II algorithm modified by the sufficient optimality condition based on two candidate codewords, the average decoding speed is improved approximately 7 times when compared with the soft-decision decoding algorithm consisting of Lin et al.'s algorithm and the improved version of the Chase algorithm.

The error-rate performance of LP decoding of the (89, 45, 17) QR code is also investigated. Simulation results show that, using powerful cutting-plane techniques, the performance of LP-based decoding can be somewhat superior to the algebraic soft-decision decoding in terms of error-correcting performance while maintaining the decoding complexity. At BER = 2×10^{-6}, the ACG-ALP decoder provides 0.2 dB coding gain than the algebraic soft-decision decoder. Moreover, LP-based decoding can be further improved by more powerful cuts. However, it is very difficult to improve the error-rate performance of the algebraic soft-decision decoding without increasing the complexity significantly. It should be pointed out that the algebraic soft-decision decoding algorithm can be further accelerated by using a new hard decoding algorithm in which there is eight different conditions corresponding to v-error cases, where $1 \leq v \leq 8$, if such an algorithm indeed exists. According to these conditions, we know how many errors occur, and then decode into the corrected codeword directly instead of the current methods that always begin to guess from the 1-error case.

Appendix A
Procedure of Algorithm 1

1) Obtain the known syndromes S_1, S_2, S_4, S_5, S_8, S_9, S_{10}, S_{11}, and S_{16}.
2) If $S_1 = S_5 = S_9 = S_{11} = 0$, no error occurs and stop; otherwise, initially set v = 1.
3) Compute all possible pairs $(S_3^{(v)}, S_{13}^{(v)})$ by using the HUSC algorithm.
4) If there exists pairs $(S_3^{(v)}, S_{13}^{(v)})$ from step 3 not yet chosen previously, choose one of the possible pairs

$(S_3^{(v)}, S_{13}^{(v)})$; otherwise, set $v = v + 1$ and go to step 9.

5) Obtain the unknown syndromes $S_3^{(v)}$, $S_6^{(v)}$, $S_7^{(v)}$, $S_{12}^{(v)}$, $S_{13}^{(v)}$, $S_{14}^{(v)}$, and $S_{15}^{(v)}$ from $(S_3^{(v)}, S_{13}^{(v)})$.
6) Apply the inverse-free BM algorithm to determine $\sigma(x)$.
7) If deg($\sigma(x)$) = v, find $u \leq v$ roots of $\sigma(x)$, which are clearly primitive 89th roots of unity, by a use of Chien's search. Then one error pattern is thus obtained and go to step 10.
8) If another possible pair $(S_3^{(v)}, S_{13}^{(v)})$ obtained from step 3 can be chosen, return to step 5; otherwise, set $v = v+1$.
9) If $v > 8$, stop; otherwise, go back to step 3.
10) If u equals v, recalculate the ordered 4-tuple (S'_1, S'_5, S'_9, S'_{11}) of the error pattern obtained; otherwise, return to step 8.
11) If (S'_1, S'_5, S'_9, S'_{11}) = (S_1, S_5, S_9, S_{11}), according to *Corollary* 2 in [17], the corrected codeword is obtained; otherwise, return to step 8.

Appendix B
Proof of Theorem 1

Proof: It follows from [29] that the codeword $\boldsymbol{v}$ with the smallest $\lambda(\boldsymbol{r}, \boldsymbol{v})$ is the maximum likelihood codeword. Thus, we only need to prove that $\boldsymbol{v}$ satisfying (13) has the smallest $\lambda(\boldsymbol{r}, \boldsymbol{v})$.

Without loss of generality, assume $\lambda(\boldsymbol{r}, \boldsymbol{v}_1) \leq \lambda(\boldsymbol{r}, \boldsymbol{v}_2)$, then $\boldsymbol{v} = \boldsymbol{v}_1$. According to (6), (7), one yields

$$\begin{aligned}
& d_H(\boldsymbol{v}_1, \boldsymbol{v}_3) + d_H(\boldsymbol{v}_2, \boldsymbol{v}_3) \\
&\quad = |D_0(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_3)| + |D_1(\boldsymbol{v}_1) \cap D_0(\boldsymbol{v}_3)| + \\
&\quad\quad |D_0(\boldsymbol{v}_2) \cap D_1(\boldsymbol{v}_3)| + |D_1(\boldsymbol{v}_2) \cap D_0(\boldsymbol{v}_3)| \\
&\quad = |D_0(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_3)| + |D_1(\boldsymbol{v}_1)| - |D_1(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_3)| + \\
&\quad\quad |D_0(\boldsymbol{v}_2) \cap D_1(\boldsymbol{v}_3)| + |D_1(\boldsymbol{v}_2)| - |D_1(\boldsymbol{v}_2) \cap D_1(\boldsymbol{v}_3)| \\
&\quad = 2\,|D_0(\boldsymbol{v}_1) \cap D_0(\boldsymbol{v}_2) \cap D_1(\boldsymbol{v}\boldsymbol{v}_3)| + n(\boldsymbol{v}_1) + n(\boldsymbol{v}_2) - \\
&\quad\quad 2\,|D_1(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_2) \cap D_1(\boldsymbol{v}_3)| \\
&\quad = 2\,|D_{00} \cap D_1(\boldsymbol{v}_3)| + n(\boldsymbol{v}_1) + n(\boldsymbol{v}_2) - 2\,|D_{11} \cap D_1(\boldsymbol{v}_3)|
\end{aligned} \tag{15}$$

Combining $d_H(\boldsymbol{v}_1, \boldsymbol{v}_3) \geq d$ with $d_H(\boldsymbol{v}_2, \boldsymbol{v}_3) \geq d$ yields

$$|D_{00} \cap D_1(\boldsymbol{v}_3)| \geq \frac{\delta_1 + \delta_2}{2} + |D_{11} \cap D_1(\boldsymbol{v}_3)| \geq \lceil (\delta_1 + \delta_2)/2 \rceil \tag{16}$$

In what follows, the proof of this theorem will be completed on two cases.

1) If $|D_{01} \cap D_1(\boldsymbol{v}_3)| \geq \lfloor (\delta_1 - \delta_2)/2 \rfloor$, then

$$\begin{aligned}
& |D_{00} \cap D_1(\boldsymbol{v}_3)| + |D_{01} \cap D_1(\boldsymbol{v}_3)| \\
&\quad \geq \lceil (\delta_1 + \delta_2)/2 \rceil + \lfloor (\delta_1 - \delta_2)/2 \rfloor \\
&\quad = \delta_1,
\end{aligned} \tag{17}$$

and

$$\begin{aligned}
\lambda(\boldsymbol{r}, \boldsymbol{v}_3) &= \sum_{i \in D_1(\boldsymbol{v}_3)} |r_i| \geq \sum_{i \in D_1(\boldsymbol{v}_3) \cap (D_{00} \cup D_{01})} |r_i| \\
&\geq \sum_{i \in (D_1(\boldsymbol{v}_3) \cap D_{00}) \cup (D_1(\boldsymbol{v}_3) \cap D_{01})^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)}} |r_i| \\
&\geq \sum_{i \in ((D_1(\boldsymbol{v}_3) \cap D_{00}) \cup (D_1(\boldsymbol{v}_3) \cap D_{01})^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)})^{(\delta_1)}} |r_i| \\
&\geq \sum_{i \in ((D_1(\boldsymbol{v}_3) \cap D_{00}) \cup D_{01}^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)})^{(\delta_1)}} |r_i| \\
&\geq \sum_{i \in (D_{00} \cup D_{01}^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)})^{(\delta_1)}} |r_i| \\
&= G(\boldsymbol{v}_1, d; \boldsymbol{v}_2, d) \geq \lambda(\boldsymbol{r}, \boldsymbol{v}),
\end{aligned} \tag{18}$$

where the first three '$\geq$' are valid because the left side of each inequality contains all the terms of the right side, and the fourth and fifth '$\geq$' are also valid because the right side of each inequality selects δ_1 elements with smallest decoding measurements from a bigger set containing the one on the left side.

2) If $|D_{01} \cap D_1(\boldsymbol{v}_3)| < \lfloor (\delta_1 - \delta_2)/2 \rfloor$, we have

$$D_1(\boldsymbol{v}_3) \cap (D_{00} \cup B_{01}^{\lfloor (\delta_1 - \delta_2)/2 \rfloor}) = D_1(\boldsymbol{v}_3), \tag{19}$$

where $B_{01}^{\lfloor (\delta_1 - \delta_2)/2 \rfloor}$ denotes a subet of D_{01}. It contains the intersection $D_{01} \cap D_1(\boldsymbol{v}_3)$ and its cardinality is $\lfloor (\delta_1 - \delta_2)/2 \rfloor$. Additionally,

$$\begin{aligned}
|D_1(\boldsymbol{v}_3)| &\geq |D_0(\boldsymbol{v}_1) \cap D_1(\boldsymbol{v}_3)| \\
&= d_H(\boldsymbol{v}_1, \boldsymbol{v}_3) - |D_1(\boldsymbol{v}_1) \cap D_0(\boldsymbol{v}_3)| \\
&\geq d - |D_1(\boldsymbol{v}_1)| = \delta_1.
\end{aligned} \tag{20}$$

So

$$\begin{aligned}
\lambda(\boldsymbol{r}, \boldsymbol{v}_3) &= \sum_{i \in D_1(\boldsymbol{v}_3)} |r_i| = \sum_{i \in D_1(\boldsymbol{v}_3) \cap (D_{00} \cup B_{01}^{\lfloor (\delta_1 - \delta_2)/2 \rfloor})} |r_i| \\
&\geq \sum_{i \in (D_1(\boldsymbol{v}_3) \cap (D_{00} \cup B_{01}^{\lfloor (\delta_1 - \delta_2)/2 \rfloor}))^{(\delta_1)}} |r_i| \\
&\geq \sum_{i \in (D_{00} \cup B_{01}^{\lfloor (\delta_1 - \delta_2)/2 \rfloor})^{(\delta_1)}} |r_i| \\
&\geq \sum_{i \in (D_{00} \cup D_{01}^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)})^{(\delta_1)}} |r_i| \\
&= G(\boldsymbol{v}_1, d; \boldsymbol{v}_2, d) \geq \lambda(\boldsymbol{r}, \boldsymbol{v}).
\end{aligned} \tag{21}$$

Thus, for any decoded candidate codeword $\boldsymbol{v}_3$, $\lambda(\boldsymbol{r}, \boldsymbol{v}) \leq \lambda(\boldsymbol{r}, \boldsymbol{v}_3)$ is always tenable. Together with the hypothesis $\lambda(\boldsymbol{r}, \boldsymbol{v}) \leq \lambda(\boldsymbol{r}, \boldsymbol{v}_2)$, one yields the codeword $\boldsymbol{v}$ with the smallest correlation discrepancy, which is the maximum likelihood codeword. If $\delta_1 < \delta_2$, the proof of this case is similar to that used in the condition $\delta_1 \geq \delta_2$. ■

Acknowledgment

The authors would like to thank Prof. Ding, Dr. W. K. Su, Dr. P. Y. Shih, and Mr. Y. S. Duan for their constructive comments and suggestions on improving the quality as well as the presentation of this paper, and especially thank Helming and Xiaojie Zhang for their help in the simulations of LP decoding.

References

[1] E. Prange, “Some cyclic error-correcting codes with simple decoding algorithms,” Air Force Cambridge Research Center-TN-156, 1958.

[2] R. He, I. S. Reed, T. K. Truong, and X. Chen, “Decoding of the (47, 24, 11) quadratic residue code,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 1181-1186, Mar. 2001.

[3] R. W. Hamming, “Error detecting and error correcting codes,” *Bell Syst. Tech. J.*, vol. 29, pp. 147-160, 1950.

[4] I. S. Reed, X. Yin, T. K. Truong, and J. K. Holmes, “Decoding the (24, 12, 8) Golay code,” *Proc. Inst. Electr. Eng.*, vol. 137, pp. 202-206, May 1990.

[5] M. Elia, “Algebraic decoding of the (23, 12, 7) Golay code,” *IEEE Trans. Inf. Theory*, vol. IT-33, no. 1, pp. 150-151, Jan. 1987.

[6] I. S. Reed, X. Yin, and T. K. Truong, “Algebraic decoding of the (32, 16, 8) quadratic residue code,” *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 876-880, July 1990.

[7] I. S. Reed and X. Chen, *Error-Control Coding for Data Networks*. Boston, MA: Kluwer, 1999.

[8] I. S. Reed, T. K. Truong, X. Chen, and X. Yin, “The algebraic decoding of the (41, 21, 9) quadratic residue code,” *IEEE Trans. Inf. Theory*, vol. 38, no. 3, pp. 974-985, May 1992.

[9] Y. H. Chen, T. K. Truong, Y. Chang, C. D. Lee, and S. H. Chen, “Algebraic decoding of quadratic residue codes using Berlekamp-Massey algorithm,” *J. Inf. Sci. Eng.*, vol. 23, pp. 127-145, Jan. 2007.

[10] X. Chen, I. S. Reed, and T. K. Truong, “Decoding the (73, 37, 13) quadratic residue code,” *Proc. Inst. Electr. Eng.*, vol. 141, pp. 253-258, Sept. 1994.

[11] Y. Chang, T. K. Truong, I. S. Reed, H. Y. Cheng, and C. D. Lee, “Algebraic decoding of (71, 36, 11), (79, 36, 11), and (97, 49, 15) quadratic residue codes,” *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1463-1473, Sept. 2003.

[12] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, “Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance,” *IEEE Trans. Commun.*, vol. 40, no. 9, pp. 1654-1661, Sept. 1994.

[13] I. S. Reed, M. T. Shih, and T. K. Truong, “VLSI design of inverse-free Berlekamp-Massey algorithm,” *Proc. Inst. Electr. Eng.*, vol. 138, pp. 295-298, Sept. 1991.

[14] X. Youzhi, “Implementation of Berlekamp-Massey algorithm without inversion,” *Proc. Inst. Electr. Eng.*, vol. 138, pp. 138-140, 1991.

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

[16] T. K. Truong, Y. Chang, Y.-H. Chen, and C. D. Lee, “Algebraic decoding of (103, 52, 19) and (113, 57, 15) quadratic residue codes,” *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 749-754, May 2005.

[17] T. K. Truong, P. Y. Shih, W. K. Su, C. D. Lee, and Yaotsu Chang, “Algebraic decoding of the (89, 45, 17) quadratic residue code”, *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5005-5011, Nov. 2008.

[18] R. T. Chien, “Cyclic decoding procedure for the Bose-Chaudhuri-Hocquenghem codes,” *IEEE Trans. Inf. Theory*, vol. 10, no. 4, pp. 357-363, Oct. 1964.

[19] T. C. Lin, W. K. Su, P. Y. Shih, and T. K. Truong, “Fast algebraic decoding of the (89, 45, 17) quadratic residue code,” *IEEE Commun. Lett.*, vol. 15, no. 2, pp. 226-228, Feb. 2011.

[20] T. C. Lin, W. K. Su, P. Y. Shih, and T. K. Truong, “Modified algebraic decoding of the (89, 45, 17) binary quadratic residue code,” in *Proc. PIMRC 2009 IEEE*, pp. 1824-1828, Tokyo, Japan, Sept. 2009.

[21] D. Chase, “A class of algorithms for decoding block codes with channel measurement information,” *IEEE Trans. Inf. Theory*, vol. IT-18, pp. 170-182, Jan. 1972.

[22] E. Berlekamp, R. J. McEliece, and H. van Tilborg, “On the inherent intractability of certain coding problems,” *IEEE Trans. Inf. Theory*, vol. IT-24, no. 3, pp. 384-386, May 1978.

[23] J. Feldman, M. Wainwright, and D. Karger, “Using linear programming to decode binary linear codes,” *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954-972, Mar. 2005.

[24] M. H. Taghavi and P. H. Siegel, “Adaptive methods for linear programming decoding,” *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5396-5410, Dec. 2008.

[25] A. Tanatmis, S. Ruzika, H. W. Hamacher, M. Punekar, F. Kienle, and N. Wehn, “A separation algorithm for improved LP-decoding of linear block codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3277-3289, July 2010.

[26] Xiaojie Zhang and Paul H. Siegel, “Adaptive cut generation algorithm for improved linear programming decoding of binary linear codes,” submitted to *IEEE Trans. Inf. Theory*.

[27] M. Miwa, T.Wadayama, and I. Takumi, “A cutting-plane method based on redundant rows for improving fractional distance,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 1005-1012, Aug. 2009.

[28] D. J. Taipale and M. B. Pursley, “An improvement to generalized-minimum-distance decoding,” *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 167-172, Jan. 1991.

[29] Shu Lin, and Daniel J. Costello, *Error Control Coding*, Prentice Hall, June, 2004.

[30] M. Chertkov and M. Stepanov, “Pseudo-codeword landscape,” in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, June 2007, pp. 1546-1550.

[31] K. Yang, X. Wang, and J. Feldman, “A new linear programming approach to decoding linear block codes,” *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1061-1072, Mar. 2008.

[32] GNU Linear Programming Kit, [Online]. Available: http://www.gnu.org/software/glpk

[33] Zsuzsanna Szabo, and Marta Kovacs, “On interior-point methods and simplex method in linear programming,” *An. St. Univ. Ovidius Constanta*, vol. 11, no. 2, 2003, pp. 155-162.

Lin Wang (S’99–M’03–SM’09) received the B.Sc. degree in Mathematics (with first class honors) from the Chongqing Normal University, Chongqing, China, in 1984, the M.Sc. degree in Applied Mathematics from the Kunming University of Technology, Kunming, China, in 1988, and the Ph.D. degree in Electronics Engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2001.

From 1984 to 1986, he was Teaching Assistant in Mathematics Department of Chongqing Normal University. From 1989 to 2002, he was Teaching Assistant, Lecturer, and then Associate Professor in Applied Mathematics and Communication Engineering in the Chongqing University of Post and Tele-communication, Chongqing, China. From 1995 to 1996, he spent one year with Mathematics Department of the University of New England, Australia. In 2003, he spent three months as visiting researcher in the Center for Chaos and Complexity Networks at the City University of Hong Kong. Since 2003, he has been full-time Professor and Associate Dean in the School of Information Science and Engineering of Xiamen University, Xiamen, China. Recently he has become the editor of *ACTA Electronica Sinica* and Guest Associate Editor of *International Journal of Bifurcation and Chaos*. He holds 8 patents in the field of physical layer in digital communications and has published over 60 international journal and conference papers. His current research interests are in the area of channel coding, chaos modulation, and their applications for wireless communication and storage systems.

Yong Li (S’11) received the B.Sc. degree in Electronic and Information Engineering from Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2003, and the M.S. degree in Communication Engineering from Xiamen University, Fujian, China, in 2006. Now he is currently pursuing the Ph.D. degree in the Department of Communication Engineering, Xiamen University, Fujian, China. From Sep. 2006 to Jan. 2007, he was a research assistant with the department of Electronic Engineering, City University of Hong Kong. From Feb. 2007 to Aug. 2009, he was with Gallop Inc., Chongqing, China. From Sep. 2011 to Aug. 2012, he visited University of California, Davis, USA, as a visiting scholar. His primary research interests include channel coding, MIMO-OFDM, joint blind equalization and decoding.

Trieu-Kien Truong (M'82–SM'83–F'99) was born in Vietnam on December 4, 1944. He received the B.S. degree from National Cheng Kung University, Tainan, Taiwan, in 1967, the M.S. degree from Washington University, St. Louis, MO, in 1971, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1976, all in electrical engineering. From 1975 to 1992, he was a Senior Member of Technical Staff (E6) with the Communication System Research Section, Jet Propulsion Laboratory, Pasadena, CA. Currently, he is a Chair Professor of Collage of Electrical and Information Engineering, I-Shou University, Kaohsiung, Taiwan. His research interests include error-correcting codes, VLSI architecture design, communication systems, signal processing, and image compression. Dr. Troung served as an Editor in the Asia area for the *Journal of Visual Communication and Image Representation* and as an Editor in the area of Coding Theory and Techniques for the IEEE TRANSACTIONS ON COMMUNICATIONS.

Tsung-Ching Lin (M'08) was born in Taiwan on April 6, 1970. He received the B.S. degree in mechanical engineering from Chung Yuan Christian Univ., Chung Li, Taiwan, in 1993, the M.S. degree in mechanical engineering from Nation Chiao Tung Univ., Hsinchu, Taiwan, in 1995, and the Ph.D. degree in electrical engineering from National Taiwan Univ. Taipei, Taiwan, in 2005. Currently, he is an Associate Professor of Information Engineering Department, I-Shou University, Kaohsiung, Taiwan. His research interests include image compression, digital information, data mining, control system, and analysis of high-tech industries.