AAECC 7, 299–308 (1996)

AAECC
Applicable Algebra in Engineering, Communication and Computing
© Springer-Verlag 1996

Fast Decoding of Non-Binary First Order Reed-Muller Codes

Alexey E. Ashikhmin[1], **Simon N. Litsyn**[2]

[1] Institute for Problems of Information Transmission, Moscow, Ermolovoy 19, GSP-4, 101447, Russia

[2] Department of Electrical Engineering – Systems, Tel-Aviv University, Ramat Aviv, 69978 Israel

Received April 23, 1993; revised version February 6, 1995

Abstract. A minimum distance decoding algorithm for non-binary first order Reed-Muller codes is described. Suggested decoding is based on a generalization of the fast Hadamard transform to the non-binary case. We also propose a fast decoding algorithm for non-binary first order Reed-Muller codes with complexity proportional to the length of the code. This algorithm provides decoding within the limits guaranteed by the minimum distance of the code.

Keywords: Fast decoding algorithms, Reed-Muller codes.

1 Introduction

First order Reed-Muller (RM) codes, gained a great amount of attention because of optimality of their parameters and existence of fast algorithms for their decoding. Decoding algorithms for binary first order Reed-Muller codes were considered in [4, 5, 6, 9, 11, 17, 18]. These algorithms provide minimum distance decoding and are based on ideas of the fast Hadamard (or Walsh-Hadamard, or Walsh) transform. Their complexity is proportional to $n \log n$, where n is the length of the code. Further generalizations and improvements of these algorithms for decoding **binary** codes were proposed in [1, 2, 3, 14, 15, 16]. In [16] an algorithm for decoding binary first order RM codes with complexity linear in n was described. This algorithm provides error correcting up to the limits guaranteed by the minimum distance of the code, and corrects some additional error patterns.

Non-binary generalizations of RM codes were also extensively studied [7, 13]. These codes contain optimal classes of first order RM codes and extended Hamming

Partly supported by the Guastallo Fellowship. This work was presented in part at the 9th International Symposium "Applied Algebra, Algebraic Algorithms and Error-Correcting Codes", New Orleans, USA, October 1991

codes, coincide with subcodes of extended BCH-codes. Decoders of these codes may be applied to decoding outer codes in cascade schemes, in communication systems with high level of noise, coset decoding of BCH-codes, etc. In [12] a majority-logic decoding algorithm for first-order RM codes over prime fields was presented. The complexity of the algorithm does not exceed $(p-1)n\log_2 n$ additions in $\mathbf{GF}(p)$ and $(p-1)n\log_2 n$ comparisons. It is applicable only to the codes over prime fields. For $p=3$ it gives the same complexity as the fast Hadamard transformation algorithm, but does not guarantee minimum distance decoding.

In the paper we consider a possibility of generalizing the well-known Hadamard transform to the non-binary case. This algorithm is widely regarded as being a real number algorithm relevant only for the case of binary codes. We demonstrate that a generalization to decoding non-binary codes with components from Galois fields is also possible. Further simplifications of the above algorithm are based on truncating intermediate results of the transform algorithm. We present a new simplified fast decoding algorithm which is relevant for first order RM codes over arbitrary fields having complexity linear in the length of the code.

The paper is organized as follows. In Section 2 we give a description of non-binary primitive first order RM codes and present a maximum likelihood (ML) algorithm for their decoding. This algorithm requires proportional to $n\log n$ number of operations. In Section 3 we describe the simplified decoding algorithm. We show that the suggested algorithm provides decoding within the limits guaranteed by the minimum distance.

We will use the following notation:

- $\bullet$ is multiplication in $\mathbf{GF}(q)$;

$\ominus, \oplus$ are subtraction and addition in $\mathbf{GF}(q)$;
$\lfloor x/n \rfloor$ is the integer part of x/n;
$\otimes$ is the Kronecker product of matrices;
δ_l^k is the Kronecker symbol, i.e. $\delta_l^k = 1$ if $l = k$ and $\delta_l^k = 0$ if $l \neq k$;
$\mathbf{E}_i = (i, i, \ldots, i)$ is the vector of suitable length consisting of identical entries equal to i.

To simplify notations, we suppose that $\mathbf{GF}(q) = \{0, 1, \ldots, q-1\}$, where 0 is the additive unit and 1 is the multiplicative unit of $\mathbf{GF}(q)$. It will be clear from the context whether i is regarded as a natural number or as an element of $\mathbf{GF}(q)$.

For an arbitrary vector $\mathbf{x} \in \mathbf{R}^n$ we denote by $\mathrm{Vmax}(\mathbf{x})$ the *value of its maximum component*, and $\mathrm{Nmax}(\mathbf{x})$ stands for the *index of its maximum component*. To compute the defined functions Vmax and Nmax it is necessary to fulfill $n-1$ comparisons.

2 Fast Maximum Likelihood Decoding of Non-Binary First Order Reed-Muller Codes

Roughly speaking, the main idea of the proposed algorithm is as follows. We associate with the code words of first order Reed-Muller code a matrix of powers of formal variable z, B_z, and show that the matrix B_z can be represented as product of sparse matrices A. Then we associate a vector $\mathbf{S}_w$ with the received vector $\mathbf{w}$. We define the product $B_z * \mathbf{S}_w$ and show that it gives the values of "correlation" $N(\mathbf{v}, \mathbf{w}) = |\{j : v_j = w_j\}|$ of the received vector $\mathbf{w}$ with all code words. The ML decoding algorithm consists of calculation of $B_z * \mathbf{S}_w$ and choice of the code word

$\mathbf{v}$ giving the maximum value $N(\mathbf{v}, \mathbf{w})$. A representation of the matrix B_z as a product of sparse matrices allows reducing complexity of the decoding algorithm.

Let us give a description of the primitive q-ary first order RM codes [7, 13]. Let $n = q^m$ (q is a prime power) stand for the length of the code, $k = m + 1$ be the number of its information symbols and $d = q^{m-1}(q-1)$ be its minimum distance. We denote this code by $R(q, m)$. The rows of its generator matrix

$$G = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_m \end{pmatrix}$$

have the following structure: $\mathbf{g}_m$ is the all-one vector of length n, the other rows $\mathbf{g}_0, \ldots, \mathbf{g}_{m-1}$ form a matrix with all possible q-ary m-tuples as the columns. We suppose that the columns are ordered lexicographically.

Example. Let $q = 4$, $n = q^2 = 16$. Then the generator matrix of $R(4, 2)$ is as follows:

$$G = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

□

Let $0 \leqq j \leqq q^m - 1$. We consider the elements of G as natural numbers. Then from the lexicographic order of columns it follows that elements $g_{0,j}, g_{1,j} \ldots, g_{m-1,j}$ form q-ary expansion of j, i.e.

$$j = g_{0,j} + g_{1,j} q + \cdots + g_{m-1,j} q^{m-1}. \tag{1}$$

Let

$$G' = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{m-1} \end{pmatrix}.$$

where $\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{m-1}$ are the corresponding rows from G. By $R'(q, m)$ we denote the code generated by G'. Denote the matrix consisting of all code words of $R'(q, m)$ by $B = [\mathbf{b}_i]$, $i = 0, \ldots, q^m - 1$. Write i in q-ary expansion $i = i_0 + i_1 q + \cdots + i_{m-1} q^{m-1}$. Then

$$\mathbf{b}_i = i_0 \bullet \mathbf{g}_0 \oplus i_1 \bullet \mathbf{g}_1 \oplus \cdots \oplus i_{m-1} \bullet \mathbf{g}_{m-1}, \tag{2}$$

where the values $i_0, i_1, \ldots, i_{m-1}$ are considered as elements of $\mathbf{GF}(q)$. Note that the code $R(q, m)$ is the union of q shifts of the code $R'(q, m)$, $R(q, m) = \{\mathbf{v} \oplus \mathbf{E}_i : \mathbf{v} \in R'(q, m), i = 0, \ldots, q-1\}$.

Let z be formal variable. Define $z^a z^b = z^{a \oplus b}$, $a, b \in \mathbf{GF(q)}$ and $0 \cdot z^x = 0$, $x \in \mathbf{GF(q)}$. Let us associate to B the matrix $B_z = [b^z_{i,j}]$, where $b^z_{i,j} = z^{b_{i,j}}$. Let $H = [h_{\alpha,\eta}]$, $h_{\alpha,\eta} = z^{\alpha \bullet \eta}$, $\alpha, \eta = 0, \ldots, q-1$.

Example. If $q = 3$

$$H = \begin{pmatrix} z^0 & z^0 & z^0 \\ z^0 & z^1 & z^2 \\ z^0 & z^2 & z^1 \end{pmatrix}.$$

□

Now we show that B_z can be factorized.

Lemma 1. *Let* $n = q^m$. *Then* $B_z = \underbrace{H \otimes H \otimes \cdots \otimes H}_{\text{m times}}$.

Proof. Let $\underbrace{H \otimes H \otimes \cdots \otimes H}_{m \text{ times}} = [h'_{i,j}]$, $i, j = 0, \ldots, q^m - 1$. We show that the expressions for $h'_{i,j}$ and $b^z_{i,j}$ are the same.

Write the values i, j in q-ary expansion

$$i = i_0 + i_1 q + \cdots + i_{m-1} q^{m-1}; \; j = j_0 + j_1 q + \cdots + j_{m-1} q^{m-1}.$$

Consider $b^z_{i,j} = z^{i_0 \bullet g_{0,j} \oplus i_1 \bullet g_{1,j} \oplus \cdots \oplus i_{m-1} \bullet g_{m-1,j}}$. From (1) and (2) we get

$$b^z_{i,j} = z^{i_0 \bullet j_0 \oplus i_1 \bullet j_1 \oplus \cdots \oplus i_{m-1} \bullet j_{m-1}}.$$

It is well known [10, 19] that $h'_{i,j} = \prod_{l=0}^{m-1} j_{i_l, j_l}$. Hence

$$h'_{i,j} = \prod_{l=0}^{m-1} h_{i_l, j_l} = \prod_{l=0}^{m-1} z^{i_l \bullet j_l} = z^{i_0 \bullet j_0 \oplus i_1 \bullet j_1 \oplus \cdots \oplus i_{m-1} \bullet j_{m-1}} = b^z_{i,j}. \quad \square$$

Now we can use the standard technique [8, 19] to factorize B_z.

Lemma 2. $B_z = A^m$, *where* $A = [a_{i,j}]$, $i, j = 0, \ldots, q^m - 1$, *and*

$$a_{i,j} = h_{i_{m-1}, j_0} \delta^{\lfloor j/q \rfloor}_{i \bmod q} = z^{i_{m-1} \bullet j_0} \delta^{\lfloor j/q \rfloor}_{i \bmod q};$$

$$i = i_0 + i_1 q + \cdots + i_{m-1} q^{m-1}; \; j = j_0 + j_1 q + \cdots + j_{m-1} q^{m-1}.$$

Example. For $q = 3$, $m = 2$ we have (only nonzero elements are shown)

$$A = \begin{pmatrix}
z^0 & z^0 & z^0 & & & & & & \\
 & & & z^0 & z^0 & z^0 & & & \\
 & & & & & & z^0 & z^0 & z^0 \\
z^0 & z^1 & z^2 & & & & & & \\
 & & & z^0 & z^1 & z^2 & & & \\
 & & & & & & z^0 & z^1 & z^2 \\
z^0 & z^2 & z^1 & & & & & & \\
 & & & z^0 & z^2 & z^1 & & & \\
 & & & & & & z^0 & z^2 & z^1
\end{pmatrix}. \quad \square$$

Given $c \in \mathbf{GF}(q)$ and $\mathbf{s} = (s_0, s_1, \ldots, s_{q-1})$, $s_i \in \mathbf{R}$, we define the $*$-product of z^c and vector $\mathbf{s}$ as follows

$$z^c * \mathbf{s} = (s_{0 \oplus c}, s_{1 \oplus c}, \ldots, s_{(q-1) \oplus c})$$

Example. The $*$-products $z^c * \mathbf{s}$ for $c \in \mathbf{GF}(4)$, $c = 0, 1, 2, 3$, equal

$$(s_0, s_1, s_2, s_3), (s_1, s_0, s_3, s_2), (s_2, s_3, s_0, s_1), (s_3, s_2, s_1, s_0),$$

respectively. $\square$

Analogously, we define the vector $*$-multiplication. Let $\mathbf{c}_z = (z^{c_0}, \ldots, z^{c_{n-1}})$, $c_j \in \mathbf{GF}(q)$, and let $\mathbf{S} = (\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n-1)})$, $\mathbf{s}^{(i)} = (s^{(i)}_0, s^{(i)}_1, \ldots, s^{(i)}_{q-1})$, $s^{(i)}_j \in \mathbf{R}$. Then the $*$-product of $\mathbf{c}_z$ and $\mathbf{S}$ is determined as follows

$$\mathbf{c}_z * \mathbf{S} = z^{c_0} * \mathbf{s}^{(0)} + z^{c_1} * \mathbf{s}^{(1)} + \cdots + z^{c_{n-1}} * \mathbf{s}^{(n-1)}.$$

Note that sums of the vectors $z^{c_i} * \mathbf{s}^{(i)}$ are calculated in $\mathbf{R}^q$.

Analogously, we define the matrix $*$-multiplication. The product matrix $F = B*D$, where $b_{i,t}$ is a power of z, and $\mathbf{d}_{t,j} \in \mathbf{R}^q$, is determined by

$$\mathbf{f}_{i,j} = \sum_t b_{i,t} * \mathbf{d}_{t,j}.$$

It is easy to verify that the $*$-multiplication has the following properties

$$(z^a z^b) * \mathbf{s} = z^a * (z^b * \mathbf{s}),\ a, b \in \mathbf{GF}(q),\ \mathbf{s} \in \mathbf{R}^q; \tag{3}$$

$$z^a * (\mathbf{s}_1 + \mathbf{s}_2) = z^a * \mathbf{s}_1 + z^a * \mathbf{s}_2,\ a \in \mathbf{GF}(q),\ \mathbf{s}_1, \mathbf{s}_2 \in \mathbf{R}^q; \tag{4}$$

$$z^a * (\mu \mathbf{s}) = \mu (z^a * \mathbf{s}),\ a \in \mathbf{GF}(q),\ \mu \in \mathbf{R},\ s \in \mathbf{R}^q. \tag{5}$$

Let $\mathbf{v} \in R(q, m)$ be transmitted, and denote the output of a discrete memoryless symmetric channel as $\mathbf{w} = (w_0, \ldots, w_{n-1})$, $w_i \in \mathbf{GF}(q)$. Write $w_i = v_i \oplus e_i$, $e_i \in \mathbf{GF}(q)$.

Let us associate to $\mathbf{w}$ a vector $\mathbf{S}_\mathbf{w} = (\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n-1)})$, where $\mathbf{s}^{(i)} = (s_0^{(i)}, s_1^{(i)}, \ldots, s_{q-1}^{(i)})$, and

$$s_t^{(i)} = \begin{cases} 1, & \text{if } t = w_i, \\ 0, & \text{otherwise.} \end{cases}$$

Let us also associate to $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ a vector of powers of z, $\mathbf{v}_z = (z^{v_0}, z^{v_1}, \ldots, z^{v_{n-1}})$.

By $N(\mathbf{v}, \mathbf{w}) = |\{j: v_j = w_j\}|$, we denote the "correlation" of the vectors $\mathbf{v}$ and $\mathbf{w}$. In other words, $N(\mathbf{v}, \mathbf{w})$ is the number of positions where elements of $\mathbf{v}$ and $\mathbf{w}$ coincide.

Lemma 3. *Let* $\mathbf{v}_z * \mathbf{S}_\mathbf{w} = (k_0, k_1, \ldots, k_{q-1})$. *Then* $k_i = N(\mathbf{v} \oplus \mathbf{E}_i, \mathbf{w})$.

Proof. Let $I = \{l: v_l \oplus i = w_l\}$. Write k_i as follows

$$k_i = s_{i \oplus v_0}^{(0)} + s_{i \oplus v_1}^{(1)} + \cdots + s_{i \oplus v_{n-1}}^{(n-1)} = \sum_{j \in I} s_{i \oplus v_j}^{(j)} + \sum_{j \notin I} s_{i \oplus v_j}^{(j)}.$$

Suppose $j \in I$. Evidently, $s_{i \oplus v_j}^{(j)} = s_{w_j}^{(j)} = 1$. If $j \notin I$ then $i \oplus v_j \neq w_j$ and therefore $s_{i \oplus v_j}^{(j)} = 0$. Hence $k_i = \sum_{j \in I} s_{i \oplus v_j}^{(j)} = |I| = N(\mathbf{v} \oplus \mathbf{E}_i, \mathbf{w})$. □

From Lemma 3 it follows that $B_z * \mathbf{S}_\mathbf{w}$ gives $N(\mathbf{v}, \mathbf{w})$ for all code words of $R(q, m)$. Thus, we can calculate this product and then choose as the result of decoding the code word $\mathbf{v}_\beta + \mathbf{E}_\gamma$ with the maximum value of $N(\mathbf{v}_\beta + \mathbf{E}_\gamma, \mathbf{w})$. The value of γ and q-ary expansion of β give the resulting information symbols.

Let us define

$$0 * (s_0, s_1, \ldots, s_{q-1}) = (0, 0, \ldots, 0).$$

Taking into account Lemma 2, the shape of A and (3), (4), one can see that

$$B_z * \mathbf{S}_\mathbf{w} = (\underbrace{A * (A * \cdots * (A}_{m \text{ times}} * \mathbf{S}_\mathbf{w}) \cdots))$$

Note, that computing $A * \mathbf{S}_\mathbf{w}$ requires $q(q-1)n$ additions.

Now we are in the position to describe the ML decoding algorithm for $R(q, m)$.

Algorithm A. *ML decoding for* $R(q, m)$.

1. For $i = 0$ to $m-1$ do steps 2–4.

2. From $\mathbf{S}_w$ construct the matrix

$$S = \begin{pmatrix} \mathbf{s}^{(0)} & \mathbf{s}^{(q)} & \dots & \mathbf{s}^{(n-q)} \\ \mathbf{s}^{(1)} & \mathbf{s}^{(1+q)} & \dots & \mathbf{s}^{(n-q+1)} \\ \dots & \dots & \dots & \dots \\ \mathbf{s}^{(q-1)} & \mathbf{s}^{(2q-1)} & \dots & \mathbf{s}^{(n-1)} \end{pmatrix}.$$

3. Compute

$$T = H * S = [\mathbf{t}_i] = [\mathbf{t}_{i,j}] \; i = 0, \dots, q-1, j = 0, \dots, n/q-1.$$

(Note that this operation is equivalent to the operation $A * \mathbf{S}_w$)

4. Organize

$$\mathbf{S}_w = (\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n-1)}) = (\mathbf{t}_0, \mathbf{t}_1, \dots, \mathbf{t}_{q-1}).$$

5. Compute $\mathbf{y} = (y_0, \dots, y_{n-1})$, where $y_k = \mathrm{Vmax}(\mathbf{s}^{(k)})$.
6. Determine $\beta = \mathrm{Nmax}(\mathbf{y})$.
7. Determine $\gamma = \mathrm{Nmax}(\mathbf{s}^{(\beta)})$.
8. $v = (\beta_0, \dots, \beta_{m-1}, \gamma)$, where $\beta = \sum_{i=0}^{m-1} \beta_i q^i$, is the resulting information vector.
9. End.

Example. *Let* $q = 4$, $n = 16$, and $\mathbf{u} = (u_0, u_1, u_2) = (1, 2, 0)$ be the information vector. Then we get the codeword

$$\mathbf{v} = u_0 \bullet \mathbf{g}_0 \oplus u_1 \bullet \mathbf{g}_1 \oplus u_2 \bullet \mathbf{g}_2 = (0, 1, 2, 3, 2, 3, 0, 1, 3, 2, 1, 0, 1, 0, 3, 2).$$

Let us consider the case of the absence of errors. In this case the output word $\mathbf{w}$ coincides with $\mathbf{v}$. We present the decoding procedure for this word.

a) $i = 0$. We compute

$$S = \begin{pmatrix} (1000) & (0010) & (0001) & (0100) \\ (0100) & (0001) & (0010) & (1000) \\ (0010) & (1000) & (0100) & (0001) \\ (0001) & (0100) & (1000) & (0010) \end{pmatrix},$$

$$T = \begin{pmatrix} (1111) & (1111) & (1111) & (1111) \\ (4000) & (0040) & (0004) & (0400) \\ (1111) & (1111) & (1111) & (1111) \\ (1111) & (1111) & (1111) & (1111) \end{pmatrix}.$$

b) $i = 1$. We compute

$$S = \begin{pmatrix} (1111) & (4000) & (1111) & (1111) \\ (1111) & (0040) & (1111) & (1111) \\ (1111) & (0004) & (1111) & (1111) \\ (1111) & (0400) & (1111) & (1111) \end{pmatrix},$$

$$T = \begin{pmatrix} (4444) & (4\,444) & (4444) & (4444) \\ (4444) & (4\,444) & (4444) & (4444) \\ (4444) & (16\,000) & (4444) & (4444) \\ (4444) & (4\,444) & (4444) & (4444) \end{pmatrix},$$

$$\mathbf{y} = (4, 4, 4, 4, 4, 4, 4, 4, 4, 16, 4, 4, 4, 4, 4, 4),$$

$$\beta = 9, \beta_0 = 1, \beta_1 = 2, \gamma = 0.$$

Thus $v = (\beta_0, \beta_1, \gamma) = (1, 2, 0) = \mathbf{u}$. □

Algorithm A allows stating the following theorem.

Theorem 1. *Maximum likelihood decoding of* $R(q,m)$ *requires at most* $q(q-1)n\log_q n$ additions and $n(q-1)+n-1$ *comparisons.* □

For the binary case the algorithm is twice as complicated as the fast Hadamard transformation. It is a consequence of use of vectors $\mathbf{s}^{(i)}$ of dimension 2 instead of 1. It is possible to decrease complexity of the algorithm by approximately a factor of $q/(q-1)$ if one uses another shape of the initial vectors $\mathbf{s}^{(i)}$. Organize $\mathbf{s}^{(i)}=(s_0^{(i)}, s_1^{(i)}, \ldots, s_{q-1}^{(i)})$,

$$s_t^{(i)} = \begin{cases} q-1, & \text{if } t=w_i, \\ -1, & \text{otherwise,} \end{cases}$$

and take into account that the sum of components in every $\mathbf{s}_i$ equals zero. Therefore, computing $s_{q-1}^{(i)}$ is redundant. We do not give here the complete description of the algorithm because of its awkwardness.

Note in passing that we can form the vectors $\mathbf{S}_w=(\mathbf{s}^{(0)}, \ldots, \mathbf{s}^{(n-1)})$, $\mathbf{s}^{(i)}=(s_0^{(i)}, \ldots, s_{q-1}^{(i)})$, as follows

$$s_t^{(i)} = \log \Pr(v_i = t \mid w_i),$$

where $\Pr(v_i=t\mid w_i)$ is the conditional probability of the event $v_i=t$. In this case, the described algorithm will find the code word $\mathbf{v}=(v_0, v_1, \ldots, v_{n-1})$ with the maximum value of $\sum_{i=0}^{n-1} \log \Pr(v_i=t\mid w_i)$, and therefore it is a ML soft decision algorithm.

3 Bounded Distance Decoding

Further simplifications of the decoding algorithm are possible if we do not demand maximum likelihood decoding, but only correcting error patterns of Hamming weight less than $d/2=q^{m-1}(q-1)/2$. In this case we can truncate intermediate results of the ML algorithm, saving rows $\mathbf{t}_i$ with the maximum sum $\sum_l \mathrm{Vmax}(\mathbf{t}_{i,l})$ and deleting the other rows.

Using the preceding notations we may present the following decoding algorithm.

Algorithm B. *Decoding of* $R(q,m)$ *up to half minimum distance.*

1. Set $j=n$.
2. For $i=0$ to $m-1$ do steps 3–8.
3. Organize from $\mathbf{S}_w$ the matrix

$$S = \begin{pmatrix} \mathbf{s}^{(0)} & \mathbf{s}^{(q)} & \ldots & \mathbf{s}^{(j-q)} \\ \mathbf{s}^{(1)} & \mathbf{s}^{(1+q)} & \ldots & \mathbf{s}^{(j-q+1)} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbf{s}^{(q-1)} & \mathbf{s}^{(2q-1)} & \ldots & \mathbf{s}^{(j-1)} \end{pmatrix}.$$

4. Compute

$$T = H * S = [\mathbf{t}_{i,l}],\ i=0,\ldots,q-1,\ l=0,\ldots,j/q-1.$$

5. Compute $\mathbf{y}=(y_0,\ldots,y_{q-1})$, where $y_k=\sum_{l=0}^{j/q-1} \mathrm{Vmax}(\mathbf{t}_{k,l})$.
6. Determine $\beta_i=\mathrm{Nmax}(\mathbf{y})$.
7. Organize the new vector $\mathbf{S}_w=(\mathbf{s}^{(0)},\ldots,\mathbf{s}^{(j/q-1)})=(\mathbf{t}_{\beta_i,0},\ldots,\mathbf{t}_{\beta_i,j/q-1})$.

8. $j \Leftarrow j/q$.
9. Compute $\gamma = \mathrm{Nmax}(\mathbf{t}_{\beta_{m-1},0})$.
10. $v = (\beta_0, \ldots, \beta_{m-1}, \gamma)$ is the resulting information vector.
11. End.

Example. Let q, n and $\mathbf{w}$ be as in the previous example.

a) $j = 16$, $i = 0$. On this step the matrices S and T are the same as in Algorithm A and

$$\mathbf{y} = (4, 16, 4, 4), \beta_0 = 1;$$

b) $j = 4$, $i = 1$,

$$S = \begin{pmatrix} (4000) \\ (0040) \\ (0004) \\ (0400) \end{pmatrix},$$

$$T = \begin{pmatrix} (\ 4 & 4 & 4 & 4\ ) \\ (\ 4 & 4 & 4 & 4\ ) \\ (\ 16 & 0 & 0 & 0\ ) \\ (\ 4 & 4 & 4 & 4\ ) \end{pmatrix},$$

$$\mathbf{y} = (4, 4, 16, 4), \beta_1 = 2;$$

c) $\gamma = 0$;
d) $v = (\beta_0, \beta_1, \gamma) = (1, 2, 0)$ and the result coincides with one in the previous example. □

Algorithm B allows to state the following theorem.

Theorem 2. *Correcting up to* $(q^m - q^{m-1})/2$ *errors in* $R(q, m)$ *requires at most* $nq(q^2 - q + 1)/(q - 1) - q\log_q n - q(q^2 - q + 1)/(q - 1)$ *additions and* $nq + (q - 1)\log_q n - 1$ *comparisons.*

Proof. At the i-th step (i.e. $j = n/q^i$) the algorithm requires $q(q-1)j$ additions to compute T, $(q-1)j$ comparisons and $(j-q)$ additions to determine $\mathbf{y}$, $(q-1)$ comparisons to compute β_i, and, finally, $(q-1)$ comparisons to determine γ. This totals to $nq(q^2 - q + 1)/(q-1) - q\log_q n - q(q^2 - q + 1)/(q-1)$ additions and $nq + (q-1)\log_q n - 1$ comparisons. Hence, for $q = \mathrm{const}$ the proposed algorithm has a *linear complexity* in n.

We will prove that the algorithm corrects errors at least within the limits guaranteed by its minimum distance. Let the number ε of errors be less than $(q^m - q^{m-1})/2 = d/2$. W.l.o.g. assume that the transmitted vector is all-zero. Consider first the case of absence of errors ($\varepsilon = 0$). Then at the i-th step

$$y_0 = \sum_{k=0}^{q^{m-i-1}-1} \mathrm{Vmax}\{\mathbf{t}_{0,k}\} = \sum_{k=0}^{q^{m-i-1}-1} \mathrm{Vmax}\{\mathbf{s}^{(kq+0)} + \mathbf{s}^{(kq+1)} + \cdots + \mathbf{s}^{(kq+(q-1))}\}, \tag{6}$$

and for $r \neq 0$

$$y_r = \sum_{k=0}^{q^{m-i-1}-1} \mathrm{Vmax}\{\mathbf{t}_{r,k}\} =$$

$$= \sum_{k=0}^{q^{m-i-1}-1} \mathrm{Vmax}\{h_{r,0} * \mathbf{s}^{(kq+0)} + h_{r,1} * \mathbf{s}^{(kq+1)} + \cdots + h_{r,q-1} * \mathbf{s}^{(kq+(q-1))}\}. \tag{7}$$

Let $i=0$. In the absence of errors, the vectors $\mathbf{s}^{(l)}$, $l=0,\ldots,q^m-1$, have the shape $(1,0,\ldots,0,0)$, hence by (6) $\mathbf{t}_{0,k}=(q,0,\ldots,0)$, $k=0,\ldots,q^{m-1}-1$, and, therefore, $y_0=q^m$. From the definition of $*$-multiplication we see that $*$-products of values z^a, z^b, $a\neq b$. a, $b\in\mathbf{GF}(q)$ with a vector containing one non-zero entry give two different vectors again with one nonzero entry having the same value. The entries $h_{r,0}, h_{r,1},\ldots,h_{r,q-1}$ of the r-th row of H are powers of z, where the powers run through different elements of $\mathbf{GF}(q)$, hence by (7) $\mathbf{t}_{r,k}=(1,1,\ldots,1)$, $k=0,\ldots,q^{m-1}-1$, and $y_r=q^{m-1}$. To the following step $(i=1)$ we pass $\mathbf{S}_w=(\mathbf{s}^{(0)},\ldots,\mathbf{s}^{(n/q-1)})$, $\mathbf{s}^{(k)}=\mathbf{t}_{0,k}=(q,0,\ldots,0)$. Taking into account (5), we conclude that the same arguments are valid for the first step, and so on. By induction at the i-th step we get $\mathbf{t}_{0,k}=(q^{i+1},0,\ldots,0)$, and, therefore, $y_0=q^m$, and $\mathbf{t}_{r,k}=(q^i,q^i,\ldots,q^i)$, $k=0,\ldots,q^{m-i-1}-1$, and $y_r=q^{m-1}$. Furthermore, note that at the i-th step,

$$\mathbf{t}_{0,k}=\mathbf{s}^{(kq^{i+1}+0)}+\mathbf{s}^{(kq^{i+1}+1)}+\cdots+\mathbf{s}^{(kq^{i+1}+(q^{i+1}-1))}, \tag{8}$$

where in the right-hand side the vectors $\mathbf{s}$ of the 0-th step are summed. Analogously, for $r\neq 0$ we have

$$\begin{aligned}\mathbf{t}_{r,k}=&h_{r,0}*(\mathbf{s}^{(kq^{i+1}+0)}+\cdots+\mathbf{s}^{(kq^{i+1}+(q^i-1))})\\&+h_{r,1}*(\mathbf{s}^{(kq^{i+1}+q^i)}+\cdots+\mathbf{s}^{(kq^{i+1}+2q^i-1)})\\&+h_{r,q-1}*(\mathbf{s}^{(kq^{i+1}+(q-1)q^i)}+\cdots+\mathbf{s}^{(kq^{i+1}+(q^{i+1}-1))}).\end{aligned} \tag{9}$$

If $\varepsilon>0$, then from (8) and (9) it follows that $\mathrm{Vmax}\{\mathbf{t}_{0,k}\}$ and $\mathrm{Vmax}\{\mathbf{t}_{r,k}\}$ may change their value by maximum ε, since the non-distorted shape of $\mathbf{s}$ is $(1,0,\ldots,0)$ and each error changes a component of $\mathbf{s}$ maximum by 1 (it simply permutes the coordinates of vector $\mathbf{s}$). Therefore, the minimum value of y_0 is $q^m-\varepsilon$, and the maximum value of y_r is $q^{m-1}+\varepsilon$. From the constraints imposed on ε, namely, $\varepsilon<d/2=(q^m-q^{m-1})/2$, we conclude that the algorithm provides correct decoding. □

References

1. Ashikhmin, A., Litsyn, S.: Analysis of quasi-optimal decoding algorithms of biorthogonal codes. Radioelectronica, **31**(11), pp. 30–34 (1988) (in Russian), English translation: Radio Electronics Commun Syst, **31**(11), 26–30 (1988)
2. Ashikhmin, A., Litsyn, S.: List algorithm for search of the maximal element of Walsh transform. Radioelectronica, **32**(11), 15–22 (1990) (in Russian), English translation: Radio Electronics Commun Syst, **32**(11), 37–41 (1990)
3. Ashikhmin, A., Litsyn, S.: Fast decoding of first order Reed-Muller and related codes, submitted
4. Be'ery, Y., Snyders, J.: Optimal soft decision block decoders based on fast Hadamard transform, IEEE. Trans. Inf. Theory, IT-**32**, 355–364 (1986)
5. Berlekamp, E. R.: The technology of error-correcting codes, Proc. IEEE **68**, 564–593 (1980)
6. Conway, J. H., Sloane, N. J. A.: Sphere Packings, Lattices and Groups, Berlin, Heidelberg, New York: Springer 1988
7. Delsarte, P., Goethals, J.-M., MacWilliams, F.J.: On generalized Reed-Muller codes and their relatives, Info. Control, **16**, 403–442 (1974)
8. Glassman, J. A.: A generalization of the fast Fourier transform, IEEE Trans., v. **C-19**(2), (1970)
9. Golomb, S. W. (ed.), Digital communications with space applications. Prentice-Hall, Englewood Cliffs, NJ 1964
10. Good, I. J.: The Intercation algorithm and practical Fourier analysis, J. R. Stat. Soc. (London) **B-20**, 361–372 (1958)

11. Green, R. R.: A serial orthogonal decoder, JPL Space Programs Summary. **37-39**-IV, 247–253 (1966)
12. Grushko, I.: Majority logic decoding of generalized Reed-Muller codes, Problemy peredachi informatsii **26**(3), 12–21 (1990) (in Russian)
13. Kasami, T., Lin, S., Peterson, W.: New generalizations of the Reed-Muller codes, Part 1: Primitive codes, IEEE Trans. Inf. Theory **IT-14**, 189–199 (1968)
14. Litsyn, S.: Fast algorithms for decoding orthogonal and related codes. Lecture Notes in Computer Science, vol. 539, Mattson H. F., Mora T., Rao T. R. N. (eds.) Applied Algebra, Algebraic Algorithms and Error Correcting Codes, pp. 39–47 (1991)
15. Litsyn, S., Mikhailovskaya, G., Neimirovsky, E., Shekhovtsov, O.: Fast decoding of first order Reed-Muller codes in the Gaussian channel, Problems Control Information Theory **14**(3), 189–201 (1985)
16. Litsyn, S., Shekhovtsov, O.: Fast decoding algorithm for first order Reed-Muller codes. Problemy Peredachi Informatsii, **19**(2), 3–7 (1983)
17. MacWilliams, F. J., Sloane, N. J. A.: The theory of error-correcting codes, Amsterdam, The Netherlands: North-Holland 1977
18. Manley, H. J., Mattson, H. F., Schatz, J. R.: Some applications of Good's theorem, IEEE Trans. Inform. Theory IT-**26**, 475–476 (1980)
19. Trakhtman, A., Trakhtman, V.: Basics of a theory of digital signals on finite intervals. Moscow, Sovetskoe Radio, 1975 (In Russian)