LETTER

Fast Decoding of the p-Ary First-Order Reed-Muller Codes Based on Jacket Transform

Moon Ho LEE\textsuperscript{a)}, Member and Yuri L. BORISSOV\textsuperscript{b)}, Nonmember

SUMMARY We propose a fast decoding algorithm for the p-ary first-order Reed-Muller code guaranteeing correction of up to \(n/4\sin\left(\frac{p-1}{2}\sqrt{p}\right)\) errors and having complexity proportional to \(n\log n\), where \(n = p^m\) is the code length and \(p\) is an odd prime. This algorithm is an extension in the complex domain of the fast Hadamard transform decoding algorithm applicable to the binary case.

\textbf{key words:} p-ary first-order Reed-Muller codes, decoding algorithms, Jacket matrix

1. Introduction

The binary first-order Reed-Muller (shortly RM) codes attracted a great deal of attention in the past because of the optimality of their parameters and the existence of fast decoding algorithms. These algorithms are based on ideas of the fast Hadamard transform and their complexity is proportional to \(n\log n\), where \(n\) is the code length (see [1] or [2], for more details).

Generalizations of RM codes over other finite fields were also extensively studied (see [4], [5]). Decoding algorithms for the first-order RM codes over such fields were presented in [8] and [10]. The proposed algorithms (some of them based on a generalization of the Hadamard transform) are highly efficient and provide maximum likelihood decoding as well as decoding within the limits guaranteed by the minimum distance. Another contribution to this subject is [9].

In this paper, we consider yet another possibility of generalizing the Hadamard transform approach for decoding of the first-order RM codes. Jacket matrices and corresponding Jacket transforms are introduced and investigated in [11] and [12]. Here, we present a new bounded-distance decoding algorithm for the first-order RM codes over the prime field \(F_p = GF(p)\) which is based on fast Jacket transform.

The paper is organized as follows. In the next section we recall some basic definitions and facts. In third section we show explicitly how to associate Jacket matrices with considered codes. In fourth section we present our main results. Finally, we end the paper with some conclusions.

2. Definitions and Preliminaries

For our purposes it is convenient to use the following definition of the (primitive) \(q\)-ary first-order RM codes. Let \(F_q = \{\alpha_1 = 0, \alpha_2, \ldots, \alpha_q\}\) be the finite field of \(q\) elements, where \(q = p^n\), \(p\) prime. Arrange the elements of \(F_q^n\):

\[
\begin{align*}
v_1 &= (0, 0, 0, \ldots, 0) \\
v_2 &= (0, 0, 0, \ldots, \alpha_2) \\
\vdots \\
v_q^n &= (\alpha_q, \alpha_q, \ldots, \alpha_q)
\end{align*}
\]

Let \(A(X) = \sum_{k=0}^{m} a_k X^k + d\), where \(X = (X_1, X_2, \ldots, X_m)\) and \(a_i, d \in F_q\), be an arbitrary affine function of \(m\) variables over \(F_q\). The vector \(i = (a_1, a_2, \ldots, a_m, d)\) is called information vector corresponding to \(A(X)\). Consider the vector \(c = (A(v_1), A(v_2), \ldots, A(v_q^n))\) of values of the function \(A\).

\textbf{Definition 2.1:} The linear subspace of \(F_q^n\) formed by the set of vectors \(c\), when \(A\) runs through all affine functions over \(F_q\) is called the first-order \(q\)-ary RM code and denoted by \(RM_q(1, m)\).

\(RM_q(1, m)\) has length \(q^m\), dimension \(m + 1\), and minimum distance \(d_{\text{min}} = q^m - q^{m-1}\).

\textbf{Definition 2.2:} ([11]) Let \(M = (m_{kl})\) be an \(n \times n\) matrix whose entries are nonzero elements of the field \(F\). Denote by \(M^t\) the transpose matrix of the matrix of inverses of \(M\) i.e. \(M^t = (m_{kl}^{-1})\). The matrix \(M\) is called Jacket matrix if \(MM^t = nI_n\), where \(I_n\) is the identity matrix over \(F\).

Note that \(n\) is assumed not to be a multiple of \(\text{char}(F)\), the characteristic of the field \(F\), when \(\text{char}(F) > 0\).

The above definition can be rewritten as follows. \(M\) is a Jacket matrix if and only if:

\[
\sum_{i=1}^{n} m_{ki} m_{li}^{-1} = 0
\]

for all pairs \((k, l)\), \(k \neq l\).

\textbf{Definition 2.3:} Let \(G\) be a finite abelian group and \(T\) be the multiplicative group of complex numbers with magnitude equal to 1. A mapping \(\chi : G \rightarrow T\) is called a character of \(G\) if:

\[
\chi(g_1 * g_2) = \chi(g_1)\chi(g_2)
\]

for any \(g_1, g_2 \in G\), where \(*\) denotes the operation in \(G\).
Proposition 2.4: If $\chi$ is a nontrivial character (i.e. \( \neq 1 \)) of a finite abelian group $G$ then
\[
\sum_{g \in G} \chi(g) = 0
\]

In this paper, we make use of the character $\chi$ of the additive group of $\mathbb{F}_p$, $p$ prime, given by: $\chi(a) = e^{2\pi ia/p}$, where $a \in \mathbb{F}_p \equiv \mathbb{Z}_p$. The character $\chi$ is called sometimes canonical additive character of $\mathbb{F}_p$.

3. Jacket Matrices Associated with the $p$-Ary First-Order Reed-Muller Codes

Further on we deal only with codes $RM_p(1,m)$, where $p$ is a prime and $m$ is a positive integer.

Let us consider the following $p^m \times p^m$ array $\Lambda_m$ whose entry $\Lambda_m(r,s)$ is equal to $\langle v_r, v_s \rangle$, where $\langle, \rangle$ denotes the inner product of two vectors from $\mathbb{F}_p^m$ and $1 \leq r, s \leq p^m$. Note that $r$th row (column) of $\Lambda_m$ is a codeword of $RM_p(1,m)$ determined by the linear function $L_r(X) = \sum_{k=1}^{m} v_{r,k}X_k$, where $v_r = (v_{r,1}, v_{r,2}, \ldots, v_{r,m})$. So, the rows/columns of $\Lambda_m$ form a subcode of $RM_p(1,m)$ denoted by $LRM_p(1,m)$. Applying the canonical additive character $\chi$ of $\mathbb{F}_p$ to every entry of $\Lambda_m$ we obtain a $p^m \times p^m$ matrix $J_m$, whose entries are complex numbers (in fact $p$th roots of unity). Clearly, $J_m$ is a symmetric matrix.

The following theorem holds.

Theorem 3.1: The matrix $J_m$ defined above is a Jacket matrix.

Proof: Let us take two arbitrary rows of the matrix $J_m$, say the $a$th row $ja$ and the $b$th row $jb$, where $a \neq b$. They correspond to two distinct linear functions $L_a = \sum_{k=1}^{m} v_{a,k}X_k$ and $L_b = \sum_{k=1}^{m} v_{b,k}X_k$, respectively. We will prove that $S_{a,b} \overset{\text{def}}{=} \sum_{k=1}^{m} \gamma_{a,b,k}$ equals to 0. Indeed, since $\chi$ is an additive character we have:
\[
S_{a,b} = \sum_{k=1}^{m} \frac{\chi(L_a(v_k))}{\chi(L_b(v_k))} = \sum_{k=1}^{m} \frac{\chi(L_a(v_k) - L_b(v_k))}{\chi(L_b(v_k))} = \sum_{k=1}^{m} \chi(L_c(v_k))
\]
where $L_c = L_a - L_b = \sum_{k=1}^{m} (v_{a,k} - v_{b,k})X_k$ for some $c$: $1 < c \leq p^m$ is the difference between $L_a$ and $L_b$ and therefore $L_c \neq 0$. Since each non-zero linear function admits any value of $\mathbb{F}_p$ exactly $p^{m-1}$ times, we have: $S_{a,b} = p^{m-1} \sum_{v \in \mathbb{F}_p} \chi(v)$. Hence, by Proposition 2.4 it follows that $S_{a,b} = 0$. Finally, by (1) we conclude that $J_m$ is a Jacket matrix.

Example 3.2: Let $p = 3$ and $m = 2$. $F_3 = \{0, 1, 2\} \equiv \mathbb{Z}_3$. Let $\omega$ be a primary third root of unity. The Jacket matrix $J_2$ looks as:
\[
J_2 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & 1 & \omega & \omega^2 & 1 & \omega \\
1 & \omega^2 & 1 & \omega & \omega & 1 & \omega^2 & \omega \\
1 & 1 & 1 & \omega & \omega & \omega & \omega & \omega \\
1 & \omega & \omega^2 & 1 & \omega & \omega^2 & 1 & \omega \\
1 & \omega^2 & 1 & \omega & \omega & 1 & \omega^2 & \omega \\
1 & 1 & 1 & \omega & \omega & \omega & \omega & \omega \\
1 & \omega & \omega^2 & 1 & \omega & \omega^2 & 1 & \omega \\
\end{pmatrix}
\]

Remark 3.3: The Jacket matrices associated above with the first-order RM codes fall into the class of generalized Butson-type Hadamard matrices [3]. However, we prefer to state the results of this article in terms of Jacket matrices and stress that their property expressed by the equation (1) is essential for the proof of Theorem 4.2 from the next section. Note also, that in the considered case, the inverses of the entries of the matrix $J_m$ are precisely their complex conjugates.

4. Decoding Algorithm Based on Jacket Transform

We assume that the noisy channel changes any symbol $x \in \mathbb{F}_p$ with small probability $\epsilon$ that does not depend on $x$.

Based on the construction from previous section, we design a fast decoding algorithm for $RM_p(1,m)$. For this aim let us define a Jacket transformation of a vector $r = (r_1, r_2, \ldots, r_{p^m})$ with components from $\mathbb{F}_p$ to be the vector $J(r) = (\chi(r_1), \chi(r_2), \ldots, \chi(r_{p^m}))$ with components from $\mathbb{T}$.

Herein, we summarize the Jacket transform decoding algorithm:

Step 1. Given a received vector $r$, compute its Jacket transformation: $\tilde{j} = J(r)$.

Step 2. Compute the reverse Jacket transform $\tilde{j} = J_j^\top$, where $J_j^\top$ is the matrix whose entries are complex conjugates of the entries of $J_m$ (see Remark 3.3).

Step 3. Find the coordinate $a$ where $\tilde{j}$ has the greatest magnitude.

Step 4. Compute the codeword $c'$ determined by the linear function $L_a = \sum_{k=1}^{m} v_{a,k}X_k$ corresponding to the vector $v_a$.

Step 5. Compute the vector $d' = r - c'$. Find the element of $\mathbb{F}_p$ with the highest frequency of occurrence among the components of $d'$, say $d$.

Step 6. Output the codeword $c = c' + d$, where $d$ is all $d$'s vector.

When $p = 2$ this algorithm coincides with the well-known Hadamard transform decoding algorithm [1],[2]. Further on we consider only the case $p > 2$.

We shall make use of the following lemma:

Lemma 4.1: Let $\omega_k$ and $\omega_l$ be two arbitrary $p$th roots of unity, where $p$ is an odd prime. Then $|\omega_k - \omega_l| \leq 2 \sin\left(\frac{\pi}{2p}\right)$.

Proof: Without loss of generality we may assume $\omega_l = 1$. Recall that the $p$th roots of unity are equally spaced around the unit circle, as a regular $p$-gon, when $p > 2$. Thus it can be easily seen that the maximum magnitude of the difference $\omega_k - 1$ is reached when $\omega_k = \omega = e^{2\pi ik/p} = e^{\pi i k/p}$. Using
the law of cosines then we get: \[ |w-1|=2 \sin\left(\frac{p-1}{2p} \pi\right). \]

Before considering the next theorem, we note, recalling the construction of \( \mathbf{J}_n \), that the columns (rows) of \( \mathbf{J}_n \) are vectors with coordinates complex conjugates (i.e., again \( p \)th roots of unity) of the coordinates of the Jacket transformations of codewords in \( \mathcal{LRM}_p(1, m) \).

Now we are in position to prove the main theorem:

**Theorem 4.2:** The Jacket transform decoding algorithm is capable of correcting up to \( \lceil p^n/4 \sin(\frac{p-1}{2p} \pi) \rceil \) errors.

**Proof:** We may assume that \( d=0 \) in information vector. Otherwise all intermediate results must be multiplied by \( \chi(d) \) and it does not confuse the considerations below. So, the codeword \( \mathbf{c} \) to be transmitted, is assumed to belong to \( \mathcal{LRM}_p(1, m) \).

Let \( \mathbf{r} \) be the received vector which differs from \( \mathbf{c} \) in \( t \) coordinates, say \( i_1, i_2, \ldots, i_t \). There are two essentially different cases to be considered when \( \mathcal{J}(\mathbf{r}) \) is multiplied with columns of the \( \mathbf{J}_m \) in Step 2.

First, let us consider the scalar product: \( \alpha = \langle \mathcal{J}(\mathbf{r}), \mathbf{J}'(\mathbf{c}) \rangle \), where \( \mathbf{J}'(\mathbf{c}) \) is the column corresponding to \( \mathbf{c} \). Thus all intermediate products in \( \alpha \) are equal to 1 with an exception of those corresponding to positions of errors, which are \( \frac{t}{p} \)th roots of unity \( \omega_{i_k} \neq 1, k=1, \ldots, t \). Hence, we have:

\[
\alpha = p^n - t + \sum_{k=1}^{t} \omega_{i_k} = p^n - t + \sum_{k=1}^{t} (1 - \omega_{i_k}).
\]

Using the triangle inequality and Lemma 4.1 we get:

\[
|\alpha| \geq p^n - \sum_{k=1}^{t} (1 - \omega_{i_k}) \geq p^n - 2 \sin\left(\frac{p-1}{2p} \pi\right) t
\]  

(2)

It remains to consider the case of scalar product \( \beta = \langle \mathcal{J}(\mathbf{c}), \mathbf{J}'(\mathbf{c}') \rangle \) for an arbitrary codeword \( \mathbf{c}' \) belonging to \( \mathcal{LRM}_p(1, m) \) and \( \mathbf{c}' \neq \mathbf{c} \). It is easy to see that:

\[
\beta \leq \sum_{k=1}^{t} (\xi_k - \eta_k), \quad \text{where} \quad \xi_k \neq \eta_k, k=1, \ldots, t.
\]

Using again the triangle inequality and Lemma 4.1 we get:

\[
|\beta| \leq \sum_{k=1}^{t} |\xi_k - \eta_k| \leq 2 \sin\left(\frac{p-1}{2p} \pi\right) t
\]  

(3)

If the number of errors \( t \) satisfies the inequality: \( t \leq \lceil p^n/4 \sin(\frac{p-1}{2p} \pi) \rceil \) then one easily gets that \( 2 \sin(\frac{p-1}{2p} \pi) t < p^n - 2 \sin(\frac{p-1}{2p} \pi) t \). Hence, from (3) and (2) we obtain: \( |\beta| < |\alpha| \), for all \( \beta \)'s. Therefore in this case Jacket transform decoder takes the correct decision, which completes the proof.

For instance this algorithm can correct up to 2 errors when applied to the \( \mathcal{RM}_4(1, 2) \) (it attains the designed error-correction capability of that code). We show the working of the algorithm with the following example.

**Example 4.3:** We make use of the ordinary lexicographic arrangement of the elements of \( \mathbb{F}_3^{2} \), namely:

\[
(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2).
\]

Then the information vector \( \mathbf{c} = (1, 0, 1, 0, 0, 0, 0, 0, 0, 1) \) corresponding to affine function \( X_1 + 2X_2 + 1 \) is encoded into the codeword \( \mathbf{e} = (1, 0, 2, 1, 0, 0, 2, 1, 0) \). Suppose, there are two errors in the received vector \( \mathbf{r} \). So, let \( \mathbf{r} = (1, 0, 1, 0, 1, 0, 0, 2, 1) \). We do the following computations:

**Step 1.** \( \mathcal{J}(\mathbf{r}) = (w, 1, w, 1, w, 1, 1, w^2, w) \).

**Step 2.** \( \mathbf{j} = \mathcal{J}(\mathbf{r}) \mathbf{J}_2^\dagger = (-3w^2, 0, 3, 0, 3w^2, 6w, 3w, -3, 0) \), where \( \mathbf{J}_2^\dagger \) is the matrix whose entries are complex conjugate of entries of \( \mathbf{J}_2 \) (see Example 3.2).

**Step 3.** The sixth coordinate of \( \mathbf{j} \) is that with the greatest magnitude.

**Step 4.** The linear function corresponding to this coordinate is \( L(X_1, X_2) = X_1 + 2X_2 \) and hereby \( \mathbf{c}' = (0, 2, 1, 1, 0, 2, 2, 1, 0) \).

**Step 5.** \( \mathbf{d}' = \mathbf{r} - \mathbf{c}' = (1, 1, 0, 2, 1, 1, 1, 1, 1) \) and the element of \( \mathbb{F}_3 \) with the highest frequency of occurrence among the components of \( \mathbf{d}' \) is \( d = 1 \).

**Step 6.** \( \mathbf{c} = (1, 0, 2, 2, 1, 0, 0, 2, 1) \). Note that in Step 5 we found that errors are in positions 3 and 4, as actually they are.

**Remark 4.4:** Note that the correcting bound \( \lceil p^n/4 \sin\left(\frac{p-1}{2p} \pi\right) \rceil \) is asymptotically equal to \( d_{\text{min}}/4 \) for large \( p \). Hence, a decoding up to approximately the half of the designed error-correction capability of the \( \mathcal{RM}_p(1, m) \) is realized by the proposed algorithm.

By analogy with the binary case, where the Sylvester construction for Hadamard matrices can be expressed in terms of a series of Kronecker products, this can be done for Jacket matrices associated with the \( p - \)arty first-order RM codes. The seed in this case is the Jacket matrix \( \mathbf{J}_1 \) obtained from \( \mathcal{RM}_p(1, 1) \). Recalling the construction of the array \( \mathbf{A}_m \) it can be easily seen that for any \( m \geq 1 \) holds:

\[
\mathbf{J}_{m+1} = \mathbf{J}_1 \otimes \mathbf{J}_m
\]  

(4)

The last equation helps for the proof of the following theorem analogous to the fast Hadamard transform theorem [1], p.422.

**Theorem 4.5:** For an arbitrary prime number \( p \) and positive integer \( m \), the following holds:

\[
\mathbf{J}_m = \mathbf{M}_m^{(1)} \mathbf{M}_m^{(2)} \ldots \mathbf{M}_m^{(m)}
\]

where

\[
\mathbf{M}_m^{(i)} = \mathbf{I}_{p^{m-i}} \otimes \mathbf{J}_1 \otimes \mathbf{I}_{p^{i-1}}, 1 \leq i \leq m,
\]

and \( \mathbf{I}_n \) denotes an \( n \times n \) identity matrix.

We omit the proof, which is similar to that given in [1].

The same representation holds for the Jacket matrix of complex conjugates \( \mathbf{J}_m^* \), where \( \mathbf{J}_1 \) must be set to \( \mathbf{J}_1^\dagger \).

In the case of Example 3.2 we have: \( \mathbf{J}_2^\dagger = \mathbf{M}_2^{(1)} \mathbf{M}_2^{(2)} \), where two matrices \( \mathbf{M}_2^{(1)} \) and \( \mathbf{M}_2^{(2)} \) are sparse matrices equal...
to $I_3 \otimes J^*_1$ and $J^*_1 \otimes I_3$, respectively, and

$$J^*_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega^2 & \omega \\ 1 & \omega & \omega^2 \end{pmatrix}. $$

Based on Theorem 4.5 we propose a fast version of the Jacket transform decoding algorithm. This modification is distinguished from the basic version in the most computational costly Step 2 taking into account that the matrix $J^m_1$ can be expressed as a product of matrices with special sparse structure. In Table 1 we give a comparison between the numbers of necessary operations (additions and multiplications of complex numbers) of the two versions of the algorithm.

Comparing the fast decoding algorithm for $R\!M_p(1,m)$ proposed in this paper with that in [8], it should be pointed out, that both algorithms have roughly the same complexity since they rely on matrix factorization into sparse factors. However, the former algorithm is closer to original fast Hadamard decoding algorithm and its description is much simpler. But of course, it is more inferior in terms of decoding capability, since the latter provides a maximum likelihood decoding.

5. Conclusion

In this paper, we propose a decoding algorithm for the first-order RM code $R\!M_p(1,m)$, based on Jacket transform. The algorithm works in the complex domain and provides \( \lfloor p^m/4 \sin \left( \frac{p-1}{2p} \pi \right) \rfloor \)-error correcting bounded-distance decoding of this code. The fast variant of the algorithm can be regarded as an extension of the so-called Green machine (see [1], [2], [6] or [7], for more details).

Although, in general, the proposed decoding algorithm cannot attain the designed error-correcting capability of the code it may be preferable for channels with moderate error-rate because of its low computational complexity and simplicity of implementation.

Acknowledgments

The authors would like to thank the anonymous referee for his valuable comments and suggestions that improved substantially the Letter. This work was supported in part by the KRF 2007-521-D00330, Korea. This research was supported by the MIC (Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Assessment).

References