Decoding of Projective Reed–Muller Codes by Dividing a Projective Space into Affine Spaces

Norihiro Nakashima,[*] Hajime Matsui[†]

Toyota Technological Institute, Nagoya 468-8511, Japan.

Abstract

A projective Reed–Muller (PRM) code, obtained by modifying a Reed–Muller code with respect to a projective space, is a doubly extended Reed–Solomon code when the dimension of the related projective space is equal to 1. The minimum distance and the dual code of a PRM code are known, and some decoding examples have been presented for low-dimensional projective spaces. In this study, we construct a decoding algorithm for all PRM codes by dividing a projective space into a union of affine spaces. In addition, we determine the computational complexity and the number of errors correctable of our algorithm. Finally, we compare the codeword error rate of our algorithm with that of the minimum distance decoding.

Key Words: error-correcting codes, affine variety codes, Gröbner basis, Berlekamp–Massey–Sakata algorithm, discrete Fourier transform.

1 Introduction

Projective Reed–Muller (PRM) codes have been investigated extensively since they were first introduced by Lachaud [1] in 1988. Sørensen [2] determined the minimum distances of PRM codes and proved that the dual code of a PRM code is also a PRM code or is spanned by a PRM code and a vector of ones. In addition, Berger and Maximy [3] presented conditions under which PRM codes are cyclic or quasi-cyclic. Recently, Ballet and Rolland [4] examined low-weight codewords of PRM codes and obtained an estimation of the second weight. The PRM codes of one-dimensional projective spaces are also considered to be doubly extended Reed–Solomon codes. Decoding examples for PRM codes related to low dimensional projective spaces are presented in [5], [6], [7].

To realize practical communication channels, many researchers constructed decoding procedures whose computational complexities are polynomial time. In addition, they investigated the numbers of errors correctable and the codeword error rates. Although the minimum distance decoding (MDD) [8], [9] achieves a good codeword error rate, the computational complexity of the MDD based on generating all codewords is known to be exponential. Pellikaan [10] developed a decoding algorithm for linear codes, which corrects t-errors if there exist t-error correcting pairs. The computational complexity of this algorithm is $O(n^3)$, where n is the code length. The Feng–Rao decoding algorithm

[11], [12] is also shown as a decoding method of $O(n^3)$ for linear codes. The number of errors correctable by the Feng–Rao algorithm is determined by Feng–Rao bounds [11], [12], [13], [14]. It is possible that these two algorithms can be applied to PRM codes. However, we cannot find any observations of t-error correcting pairs and Feng–Rao bounds for PRM codes, and it is difficult to determine the numbers of errors correctable.

The objective of the present study is to investigate a decoding algorithm for all PRM codes such that its computational complexity is less than $O(n^3)$ and the number of errors correctable is determined. We construct a new decoding algorithm by dividing a projective space into a union of affine spaces that a decoding algorithm proposed the second author [15] is applied for each affine component. In our algorithm, we adopt the Berlekamp–Massey–Sakata (BMS) algorithm [16], [17], [18], [19] to obtain a Gröbner basis whose zeros are the error positions, and we use the discrete Fourier transform (DFT) to determine the error values. After that, we prove that the computational complexity of our algorithm is strictly less than $O(n^3)$. In particular, the complexity of the error position determination is $O(zn^2)$ and that of the error value determination is $O(qn^2)$, where z is the maximum of the cardinalities of Gröbner bases obtained by BMS algorithm for all components and q is the finite field cardinality. We have $z < n/q$. Next, we determine the number of errors correctable by our algorithm component-wise. This implies the number of errors correctable at arbitrary positions. Finally, we compare the codeword error rate of our algorithm with that of the MDD and find them to be similar for some high-order PRM codes.

The remainder of this paper is organized as follows. In Section 2, we present some preliminary notation and recall the results of a previous study [15]. In Section 3, we present an example of PRM code that shows a difficulty to construct a decoding algorithm. In Section 4, we construct a decoding algorithm for all PRM codes. In Section 5, we determine the number of errors corrected by our algorithm. In Section 6, we present an example of a decoding procedure. In Section 7, we compute the computational complexity of our algorithm. In Section 8, we compare the codeword error rate of our algorithm with that of MDD. Finally, in Section 9, we summarize our findings and conclude the paper by briefly discussing the scope for future investigation.

2 Preliminaries

2.1 Reed–Muller codes

Throughout this paper, let q be a prime power and let $\mathbb{F}_q$ denote a finite field consisting of q elements. Let m be a positive integer. We define

$$\mathbb{A}_m(\mathbb{F}_q) = \left\{ (\omega_1, \ldots, \omega_m) \,\middle|\, \omega_1, \ldots, \omega_m \in \mathbb{F}_q \right\}, \tag{2.1}$$

where $\mathbb{A}_m(\mathbb{F}_q)$ is called an m-dimensional affine space over $\mathbb{F}_q$. We often omit a coefficient field $\mathbb{F}_q$ and write $\mathbb{A}_m(\mathbb{F}_q) = \mathbb{A}_m$ for short. Let $\mathbb{F}_q[X_1, \ldots, X_m]$ denote the polynomial ring over $\mathbb{F}_q$ in m variables. For a polynomial $f(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$, we often write $f(X_1, \ldots, X_m) = f$. Let $f(\omega_1, \ldots, \omega_m)$ denote the value obtained by substituting $(\omega_1, \ldots, \omega_m) \in \mathbb{A}_m$ for $f \in \mathbb{F}_q[X_1, \ldots, X_m]$.

Let $\mathbb{F}_q[X_1, \ldots, X_m]_{\leq \nu}$ denote the set of all polynomials in $\mathbb{F}_q[X_1, \ldots, X_m]$ of degree $\leq \nu$.

Definition 2.1 (Reed–Muller code, RM code) *A RM code over $\mathbb{F}_q$ of order ν and length q^m is defined by*

$$\mathrm{RM}_\nu(m, q) = \left\{ (f(P))_{P \in \mathbb{A}_m} \,\middle|\, f \in \mathbb{F}_q[X_1, \ldots, X_m]_{\leq \nu} \right\}. \quad \square \tag{2.2}$$

It has been shown (cf. [20]) that the dimension k and the minimum distance d of $\mathrm{RM}_\nu(m, q)$ are

$$k = \sum_{t=0}^{\nu} \sum_{j=0}^{m} (-1)^j \binom{m}{j} \binom{t - jq + m - 1}{t - jq}, \tag{2.3}$$

$$d = (q - s)q^{m-r-1}, \tag{2.4}$$

where r and s are respectively the quotient and remainder obtained when ν is divided by $q - 1$; that is, $0 \le r < m - 1$, $0 \le s < q - 1$, and $\nu = r(q - 1) + s$. For a finite set Ω, let $\mathbb{F}_q^\Omega = \{(c_P)_{P \in \Omega} \mid c_P \in \mathbb{F}_q\}$ denote the $\mathbb{F}_q$-linear space indexed by Ω. For a subset C of $\mathbb{F}_q^\Omega$, we denote the dual $C^\perp$ of C by

$$C^\perp = \left\{ (u_P)_{P \in \Omega} \in \mathbb{F}_q^\Omega \,\middle|\, \sum_{P \in \Omega} c_P u_P = 0 \text{ for all } (c_P)_{P \in \Omega} \in C \right\}. \tag{2.5}$$

The following is widely known (see, e.g., [2]).

Proposition 2.2 *Let $\mu = m(q - 1) - \nu$. The dual of $\mathrm{RM}_\nu(m, q)$ is obtained by*

$$\mathrm{RM}_\nu(m, q)^\perp = \mathrm{RM}_{\mu-1}(m, q). \qquad \square \tag{2.6}$$

2.2 Projective Reed–Muller codes

We define

$$\mathbb{P}_m(\mathbb{F}_q) = (\mathbb{A}_{m+1} \setminus \{0\}) / \sim \tag{2.7}$$

with the equivalence relation

$$P_1 \sim P_2 \quad \text{if} \quad P_1 = \lambda P_2 \text{ for some } \lambda \in \mathbb{F}_q \setminus \{0\}, \tag{2.8}$$

where $\mathbb{P}_m(\mathbb{F}_q)$ is called an m-dimensional projective space over $\mathbb{F}_q$. We often write $\mathbb{P}_m(\mathbb{F}_q) = \mathbb{P}_m$.

We express the equivalence class of a representative $(\omega_0, \omega_1, \ldots, \omega_m)$ as $(\omega_0 : \omega_1 : \cdots : \omega_m)$. For each $P = (\omega_0 : \omega_1 : \cdots : \omega_m) \in \mathbb{P}_m$, let i be the smallest index such that $\omega_i \ne 0$. Then, $(0, \ldots, 0, 1, \omega'_{i+1}, \ldots, \omega'_m)$ is a representative of P, where $\omega'_j = \omega_j / \omega_i$ for $j > i$. Let R denote the polynomial ring $\mathbb{F}_q[X_0, X_1, \ldots, X_m]$ over $\mathbb{F}_q$ in variables $X_0, X_1, \ldots, X_m$. The value $f(P)$ is defined by substituting the representative $(0, \ldots, 0, 1, \omega'_{i+1}, \ldots, \omega'_m)$ for $f = f(X_0, X_1, \ldots, X_m) \in R$; this is uniquely determined. A projective space is identified by a union of affine spaces, i.e.,

$$\mathbb{P}_m = \Psi_0 \cup \Psi_1 \cup \cdots \cup \Psi_m, \tag{2.9}$$

where $\Psi_i = \{(0 : \cdots : 0 : 1 : \omega_{i+1} : \cdots : \omega_m) \in \mathbb{P}_m \mid \omega_j \in \mathbb{F}_q, i + 1 \le j \le m\}$ is a subset of $\mathbb{P}_m$ for all $i \in \{0, 1, \ldots, m\}$ by which an $(m - i)$-dimensional affine space is identified.

Let n be the number of elements in $\mathbb{P}_m$. Then, $n = (q^{m+1} - 1)/(q - 1) = q^m + \cdots + q + 1$. Let R_ν denote the linear subspace of R consisting of homogeneous polynomials of degree ν.

Definition 2.3 (Projective Reed–Muller code, PRM code) *A PRM code over $\mathbb{F}_q$ of order ν and length n is defined by*

$$\mathrm{PRM}_\nu(m, q) = \{(f(P)_{P \in \mathbb{P}_m} \mid f \in R_\nu\}. \qquad \square \tag{2.10}$$

Table 1: Parameters of $\text{PRM}_\nu(2, 16)$

ν	5	8	11	14	17	20	23	26	29
k	21	45	78	120	168	207	237	258	270
d	192	144	96	48	15	12	9	6	3

A PRM code is trivial (i.e., $\dim \text{PRM}_\nu(m, q) = n$) if $\nu > m(q - 1)$ (see [2, Remark 3]). Therefore, in the rest of this paper, we assume that $0 < \nu \le m(q - 1)$. It is shown (cf. [2]) that $\text{PRM}_\nu(m, q)$ is an (n, k, d)-code with

$$k = \sum_{t=0}^{r} \left(\sum_{j=0}^{m+1} (-1)^j \binom{m + 1}{j} \binom{s + m - t + (t - j)q}{s + 1 - t + (t - j)q} \right), \tag{2.11}$$

$$d = (q - s)q^{m-r-1}, \tag{2.12}$$

where r and s are determined by $0 \le r < m$, $0 \le s < q - 1$, and $\nu - 1 = r(q - 1) + s$. Table 1 lists some dimensions and minimum distances of $\text{PRM}_\nu(2, 16)$. The following is used later in Lemma 4.1.

Theorem 2.4 ([2]) *Let $\mu = m(q - 1) - \nu$. The dual of $\text{PRM}_\nu(m, q)$ is obtained by the following:*

1. $\text{PRM}_\nu(m, q)^\perp = \text{PRM}_\mu(m, q)$ *if $\nu \not\equiv 0 \pmod{q - 1}$,*

2. $\text{PRM}_\nu(m, q)^\perp = \text{span}_{\mathbb{F}_q}\{\mathbf{1}, \text{PRM}_\mu(m, q)\}$ *if $\nu \equiv 0 \pmod{q - 1}$, where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{F}_q^n$.* $\square$

2.3 Affine variety codes

Let Ψ be a non-empty subset of $\mathbb{A}_m$, i.e., $\emptyset \ne \Psi \subseteq \mathbb{A}_m$. We define an ideal $Z(\Psi)$ of $\mathbb{F}_q[X_1, \ldots, X_m]$ as

$$Z(\Psi) = \{f \in \mathbb{F}_q[X_1, \ldots, X_m] \mid f(P) = 0 \text{ for all } P \in \Psi\}. \tag{2.13}$$

Definition 2.5 (Affine variety code) *For an $\mathbb{F}_q$-linear subspace L of a quotient ring $\mathbb{F}_q[X_1, \ldots, X_m]/Z(\Psi)$, we define an affine variety code as*

$$C(L, \Psi) = \{(f(P))_{P \in \Psi} \in \mathbb{F}_q^\Psi \mid f \in L\}. \quad \square \tag{2.14}$$

We previously proposed a decoding algorithm [15, Algorithm 2] for a class of affine variety codes using the BMS algorithm and DFT. The following definitions are required to explain this decoding algorithm. Let M be the set of all monomials whose exponent of each variable is less than q, i.e., $M = \{X_1^{a_1} \cdots X_m^{a_m} \mid (a_1, \ldots, a_m) \in \mathbb{N}_0^m, a_1, \ldots, a_m \le q - 1\}$, where $\mathbb{N}_0$ is the set of nonnegative integers.

Definition 2.6 (Discrete Fourier transform, DFT) *A linear map $\mathcal{F}$ is defined by*

$$\mathcal{F} : \mathbb{F}_q^{\mathbb{A}_m} \to \mathbb{F}_q^M, \quad (c_P)_{P \in \mathbb{A}_m} \mapsto \left(\sum_{P \in \mathbb{A}_m} c_P h(P) \right)_{h \in M}, \tag{2.15}$$

and $\mathcal{F}$ is called a DFT on $\mathbb{F}_q^{\mathbb{A}_m}$. $\square$

The following map is the inverse of $\mathcal{F}$, and is called an inverse discrete Fourier transform (IDFT) on $\mathbb{F}_q^{\mathbb{A}_m}$. For a finite set Ω, let $|\Omega|$ denote the number of elements in Ω.

Definition 2.7 *For each $P = (\omega_1, \ldots, \omega_m) \in \mathbb{A}_m$, we define a subset $\mathrm{supp}(P)$ of $\{1, \ldots, m\}$ by $\mathrm{supp}(P) = \{i \mid \omega_i \neq 0 \ (1 \leq i \leq m)\}$. Let $s = |\mathrm{supp}(P)|$. A linear map $\mathcal{F}^{-1}$ is defined by*

$$\mathcal{F}^{-1} : \mathbb{F}_q^M \to \mathbb{F}_q^{\mathbb{A}_m}, \quad (r_h)_{h \in M} \mapsto (c_P)_{P \in \mathbb{A}_m}, \tag{2.16}$$

where

$$c_P = (-1)^s \sum_{l_1=1}^{q-1} \cdots \sum_{l_s=1}^{q-1} \left\{ \sum_{J \subseteq \mathrm{supp}(P)^c} (-1)^{|J|} r_{h_{(P,l,J)}} \right\} \omega_1^{-l_1} \cdots \omega_s^{-l_s}, \tag{2.17}$$

J runs over all subsets of $\mathrm{supp}(P)^c = \{1, \ldots, m\} \setminus \mathrm{supp}(P)$, and $h_{(P,l,J)} = X_1^{b_1} \cdots X_m^{b_m}$ is a monomial such that

$$b_i = \begin{cases} l_i & \text{if } i \in \mathrm{supp}(P), \\ q - 1 & \text{if } i \in J, \\ 0 & \text{if } i \notin \mathrm{supp}(P) \cup J. \end{cases} \quad \square \tag{2.18}$$

Let $<$ be a monomial order, and $\mathcal{G}_\Psi$ a Gröbner basis for the ideal $Z(\Psi)$ (see [21], [22], [23] or [24] for the theory of Gröbner bases). We write $X^a = X_1^{a_1} \cdots X_m^{a_m}$ for $a = (a_1, \ldots, a_m) \in \mathbb{N}_0^{m+1}$. Let $f \in \mathbb{F}_q[X_1, \ldots, X_m]$, where $f = \sum_{a \in \mathbb{N}_0^m} \lambda_a X^a$ for some coefficients $\lambda_a \in \mathbb{F}_q$. The leading monomial $\mathrm{LM}(f)$ of f is the maximum of the monomials arranged in $<$ that have nonzero coefficients in f, i.e., $\mathrm{LM}(f) = \max_< \{X^a \mid \lambda_a \neq 0\}$. For a subset Φ of Ψ, we define a set $D(\Phi)$ as

$$D(\Phi) = \{X^a \mid a \in \mathbb{N}_0^m\} \setminus \{\mathrm{LM}(f) \mid 0 \neq f \in Z(\Phi)\}. \tag{2.19}$$

Since $\{X_1^q - X_1, \ldots, X_m^q - X_m\} \subseteq Z(\Psi)$, we have $D(\Phi) \subseteq D(\Psi) \subseteq M$. We note that $D(\Psi)$ forms a basis for $\mathbb{F}_q[X_1, \ldots, X_m]/Z(\Psi)$ (see [23, Theorem 19]).

Let z be the number of elements in the Gröbner basis $\mathcal{G}_\Phi$, and $\{f^{(1)}, \ldots, f^{(z)}\}$ the set of elements in $\mathcal{G}_\Phi$.

Definition 2.8 *A linear map $\mathcal{E}_\Phi$ is defined by*

$$\mathcal{E}_\Phi : \mathbb{F}_q^{D(\Phi)} \to \mathbb{F}_q^M, \quad (r_h)_{h \in D(\Phi)} \mapsto (r_g)_{g \in M}, \tag{2.20}$$

where for $g \in M$,

$$r_g = \sum_{h \in D(\Phi)} v_h r_h, \tag{2.21}$$

v_h is obtained by the division algorithm by $\mathcal{G}_\Phi$:

$$g(X) = \sum_{0 \leq w < z} u^{(w)}(X) f^{(w)}(X) + v(X) \tag{2.22}$$

for some $u^{(w)}(X) \in \mathbb{F}_q[X_1, \ldots, X_m]$ and $v(X) = \sum_{h \in D(\Phi)} v_h h \in \mathbb{F}_q[X_1, \ldots, X_m]$. $\square$

Definition 2.9 *Let L be a subspace of* $\mathbb{F}_q[X_1, \ldots, X_m]/Z(\Psi)$ *over* $\mathbb{F}_q$. *We say that L has a monomial basis if*

$$L = \text{span}_{\mathbb{F}_q}(B) \text{ for some } B \subseteq D(\Psi). \quad \square \tag{2.23}$$

Example 2.10 *Let* $\Psi = \mathbb{A}_1(\mathbb{F}_4)$. *Then,* $Z(\Psi) = \langle X^4 + X \rangle$ *and* $D(\Psi) = \{1, X, X^2, X^3\}$. *The linear space* $L = \text{span}_{\mathbb{F}_4}\{1, X, X^2, X^3\}$ *has a monomial basis* $B = \{1, X, X^2, X^3\} \subseteq D(\Psi)$. *Next,* $L' = \text{span}_{\mathbb{F}_4}\{1 + X^2\}$ *does not have any monomial basis, since* $1 + X^2$ *is not in* $D(\Psi)$. $\quad \square$

Example 2.11 *Let* $\Psi = \mathbb{A}_2(\mathbb{F}_4)$. *Since* $Z(\Psi) = \langle X_1^4 + X_1, X_2^4 + X_2 \rangle$, *we have* $D(\Psi) = \{X_1^i X_2^j \mid 0 \leq i, j \leq 3\}$. *Then,* $L = \text{span}_{\mathbb{F}_4}\{1, X_1 + X_2, X_2\}$ *has a monomial basis* $B = \{1, X_1, X_2\}$, *since* X_1 *is a linear combination of* $X_1 + X_2$ *and* X_2. $\quad \square$

Example 2.12 *Let* $\Psi = \mathbb{A}_m$. *Then,* $Z(\Psi) = \langle X_1^q - X_1, \ldots X_m^q - X_m \rangle$. *We have that* $C(L, \Psi) = \text{RM}_\nu(m, q)$, *where* $B = \{\prod_{j=1}^m X_j^{a_j} \mid \sum_{j=1}^m a_j \leq \nu, \ 0 \leq a_1, \ldots, a_m \leq q - 1\}$ *and* $L = \text{span}_{\mathbb{F}_q}(B)$. *Thus, L has a monomial basis B.* $\quad \square$

Let $(r_P)_{P \in \Psi} = (c_P)_{P \in \Psi} + (e_P)_{P \in \Psi}$ be a received word, where $(c_P)_{P \in \Psi} \in C^\perp(L, \Psi)$ and $(e_P)_{P \in \Psi} \in \mathbb{F}_q^\Psi$. Let $\Phi = \{P \in \Psi \mid e_P \neq 0\}$ be the set of error positions of the received word $(r_P)_{P \in \Psi}$. We call $(\sum_{P \in \Psi} r_P h(P))_{h \in B}$ a syndrome of $(r_P)_{P \in \Psi}$ related to $C(L, \Psi)$. It follows from $(c_P)_{P \in \Psi} \in C^\perp(L, \Psi)$ that $(\sum_{P \in \Psi} r_P h(P))_{h \in B} = (\sum_{P \in \Psi} e_P h(P))_{h \in B}$. Thus, the syndrome is a B-component of $\mathcal{F}((e_P)_{P \in \mathbb{A}_m})$, where $e_P = 0$ if $P \in \mathbb{A}_m \setminus \Psi$. Let $\mathcal{R}_\Psi : \mathbb{F}_q^{\mathbb{A}_m} \to \mathbb{F}_q^\Psi$ be the restriction map. Algorithm 1 is a decoding algorithm for $C^\perp(L, \Psi)$. To apply Algorithm 1, it is sufficient that L has a monomial basis B. We note that a RM code is expressed as $C^\perp(L, \Psi)$ such that L has a monomial basis by Proposition 2.2 and Example 2.12.

Algorithm 1: Error correction for $C^\perp(L, \Psi)$ [15]

Input: $(r_P)_{P \in \Psi} \in \mathbb{F}_q^\Psi$, where $(r_P)_{P \in \Psi} = (c_P)_{P \in \Psi} + (e_P)_{P \in \Psi}$, $(c_P)_{P \in \Psi} \in C^\perp(L, \Psi)$ and $(e_P)_{P \in \Psi} \in \mathbb{F}_q^\Psi$
Output: $(\hat{c}_P)_{P \in \Psi}$
Step 1. $(S_h)_{h \in B} = (\sum_{P \in \Psi} r_P h(P))_{h \in B}$.
Step 2. Calculate $\mathcal{G}_\Phi$ from the syndrome $(S_h)_{h \in B}$ by
 the BMS algorithm (cf. [22], [25]).
Step 3. $(\hat{e}_P)_{P \in \Psi} = \mathcal{R}_\Psi \circ \mathcal{F}^{-1} \circ \mathcal{E}_\Phi((S_h)_{h \in B})$.

Step 4. $(\hat{c}_P)_{P \in \Psi} = (r_P)_{P \in \Psi} - (\hat{e}_P)_{P \in \Psi}$.

In the case when the dimension of $C^\perp(L, \Psi)$ is not 0, Algorithm 1 computes $(c_P)_{P \in \Psi}$ correctly, i.e., $(\hat{c}_P)_{P \in \Psi} = (c_P)_{P \in \Psi}$, if

$$2|\Phi| < d_{\text{FR}}(C^\perp(L, \Psi)), \tag{2.24}$$

where $d_{\text{FR}}(C^\perp(L, \Psi))$ is a Feng–Rao bound. In Step 1, we calculate a syndrome $(S_h)_{h \in B}$ of $(r_P)_{P \in \Psi}$. In Step 2, we calculate the Gröbner basis $\mathcal{G}_\Phi$ for $Z(\Phi)$ whose zeros are error positions. In Step 3, we extend the syndrome $(S_h)_{h \in B} = (\sum_{P \in \Psi} e_P h(P))_{h \in B}$ to $\mathcal{F}((e_P)_{P \in \mathbb{A}_m})$ by applying $\mathcal{E}_\Phi$. Then, by applying $\mathcal{R}_\Psi \circ \mathcal{F}^{-1}$, we obtain the error word $(e_P)_{P \in \Psi}$.

If the dimension of $C^\perp(L, \Psi)$ is 0, Algorithm 1 computes all error words correctly, i.e., $(\hat{c}_P)_{P \in \Psi} = (c_P)_{P \in \Psi}$ for all $(e_P)_{P \in \Psi} \in \mathbb{F}_q^\Psi$. Indeed, since L has a monomial basis $B = M$, we have $(S_h)_{h \in B} = (S_h)_{h \in M} = (\sum_{P \in \Psi} e_P h(P))_{h \in M} = \mathcal{F}((e_P)_{P \in \mathbb{A}_m})$. This means that the syndrome is the image of an error word by the DFT. Thus, by applying $\mathcal{R}_\Psi \circ \mathcal{F}^{-1}$ to the syndrome, we obtain the error word $(e_P)_{P \in \Psi}$. Hence, in this case, we do not calculate Step 2 and $\mathcal{E}_\Phi$ of Step 3.

3 Basis for PRM codes

In general, if L has a monomial basis and a Feng–Rao bound of $C^{\perp}(L, \Psi)$ is high, Algorithm 1 has a good codeword error rate. However, when $C^{\perp}(L, \Psi)$ is a PRM code, it is difficult to determine whether L has a monomial basis. In this section, we present an example of PRM code $C^{\perp}(L, \Psi)$ such that L does not have any monomial bases.

First, we prove that a PRM code is the dual of an affine variety code. A projective space $\mathbb{P}_m$ is identified by a set $\Psi = \bigcup_{i=0}^{m}\{(0, \ldots, 0, 1, \omega_{i+1}, \ldots, \omega_n) \mid \omega_{i+1}, \ldots, \omega_n \in \mathbb{F}_q\}$ of representatives in $\mathbb{A}_{m+1}$. Let ν be a positive integer and $\mu = m(q-1) - \nu$. Let $L = \mathrm{span}_{\mathbb{F}_q}\{X^{\boldsymbol{a}} \in R/Z(\Psi) \mid \boldsymbol{a} \in \mathbb{N}_0^{m+1}, |\boldsymbol{a}| = \mu\}$ if $\nu \not\equiv 0$ modulo $q-1$, and $L = \mathrm{span}_{\mathbb{F}_q}\{\mathbf{1}, X^{\boldsymbol{a}} \in R/Z(\Psi) \mid \boldsymbol{a} \in \mathbb{N}_0^{m+1}, |\boldsymbol{a}| = \mu\}$ if $\nu \equiv 0$ modulo $q-1$. Then, $C^{\perp}(L, \Psi) = \mathrm{PRM}_\nu(m, q)$ by Eq. (2.10), Eq. (2.14) and Theorem 2.4. To determine whether L has a monomial basis, we need to consider reductions in $R/Z(\Psi)$ and linear combinations of elements in L.

Next, we present an example of a PRM code such that L does not have any monomial bases. Let $|\boldsymbol{a}| = a_0 + a_1 + \cdots + a_m$ for $\boldsymbol{a} = (a_0, a_1, \ldots, a_m) \in \mathbb{N}_0^{m+1}$. In this section, we fix a monomial order $\prec$ in the following manner: $X^{\boldsymbol{a}} \prec X^{\boldsymbol{b}}$ if "$|\boldsymbol{a}| < |\boldsymbol{b}|$" or "$|\boldsymbol{a}| = |\boldsymbol{b}|$ and there exists an index ℓ such that $a_m = b_m, a_{m-1} = b_{m-1}, \ldots, a_{\ell+1} = b_{\ell+1}$ and $a_\ell < b_\ell$."

Definition 3.1 *A set of polynomials $\mathcal{G}$ is defined as follows:*

1. *When $m = 1$, we set $\mathcal{G} = \{X_1^q - X_1, (X_0 - 1)(X_1 - 1), X_0^2 - X_0\}$.*

2. *When $m = 2$, we set $\mathcal{G} = \{X_2^q - X_2, X_1^q - X_1, (X_0 - 1)(X_1 - 1)(X_2 - 1), (X_0 - 1)(X_1^2 - X_1), X_0^2 - X_0\}$.*
   $\square$

The inclusion $\mathcal{G} \subseteq Z(\Psi)$ immediately follows. Let $\langle \mathcal{G} \rangle$ denote the ideal of R generated by $\mathcal{G}$. By Buchberger's criterion (see [21, Theorem 2.6.6]), we can directly verify that $\mathcal{G}$ is a Gröbner basis for $\langle \mathcal{G} \rangle$. Thus, we can compute a basis for a quotient ring $R/\langle \mathcal{G} \rangle$, and we have $\dim_{\mathbb{F}_q}(R/\langle \mathcal{G} \rangle) = n$ by [21, Proposition 5.3.4]. At the same time, we have $\dim_{\mathbb{F}_q}(R/Z(\Psi)) = |\Psi| = n$ by [23, Theorem 19]. Therefore, $Z(\Psi)$ coincides with $\langle \mathcal{G} \rangle$. In particular, $\mathcal{G}$ is a Gröbner basis for $Z(\Psi)$. By (2.19), we have that

1. $D(\Psi) = \{X_1^{a_1} \mid 0 \le a_1 \le q - 1\} \cup \{X_0\}$ if $m = 1$,

2. $D(\Psi) = \{X_1^{a_1} X_2^{a_2} \mid 0 \le a_1, a_2 \le q - 1\} \cup \{X_0 X_2^{a_2} \mid 0 \le a_2 \le q - 1\} \cup \{X_0 X_1\}$ if $m = 2$.

We show monomial positions of $D(\Psi)$ in Fig. 1 and Fig. 2.

Example 3.2 *Let $q = 4, m = 2, \nu = 3$. By Theorem 2.4, we have $C^{\perp}(L, \Psi) = \mathrm{PRM}_3(2, 4)$, where $L = \mathrm{span}_{\mathbb{F}_4}\{1, X^{\boldsymbol{a}} \mid |\boldsymbol{a}| = 3\} \subseteq \mathbb{F}_4[X_0, X_1, X_2]/Z(\Psi)$. Monomials $X_0 X_1^2, X_0^2 X_1$ can be reduced in $\mathbb{F}_4[X_0, X_1, X_2]/Z(\Psi)$ as follows:*

$$X_0 X_1^2 = X_1^2 + X_0 X_1 - X_1, \quad X_0^2 X_1 = X_0 X_1. \tag{3.1}$$

Thus, $X_1^2 - X_1$ is obtained by a linear combination of elements in L. However, it follows from a direct calculation that any linear combination of elements in L containing $X_1^2 - X_1$ is not in $D(\Psi)$. This means that L does not have any monomial bases. $\square$
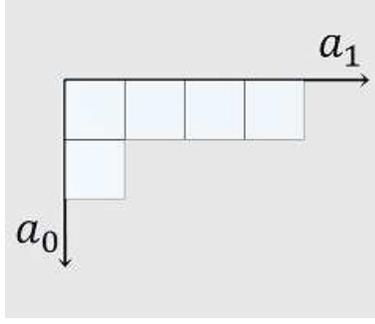
Figure 1: Monomial positions of $D(\Psi)$ if $m = 1, q = 4$
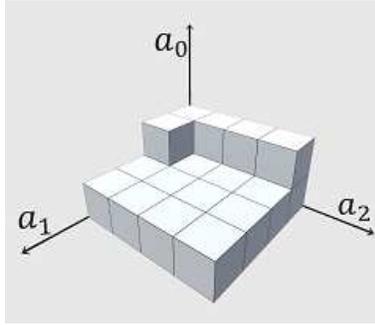
Figure 2: Monomial positions of $D(\Psi)$ if $m = 2, q = 4$

4 Decoding algorithm

In this section, we construct a decoding algorithm for all PRM codes following the decomposition $\mathbb{P}_m = \bigcup_{i=0}^{m} \Psi_i$. As described in Section 3, there exists a PRM code that does not have any monomial bases. On the other hand, for each Ψ_i-component, we can find a suitable monomial basis B_i such that $C^\perp(\text{span}_{\mathbb{F}_q}(B_i), \Psi_i)$ is a RM code. Then, we obtain a Ψ_i-component of an error ward $(e_P)_{P \in \mathbb{P}_m}$ by applying Algorithm 1 from a syndrome related to the RM code. By repeating this for all $i \in \{0, 1, \ldots, m\}$, we obtain the error word $(e_P)_{P \in \mathbb{P}_m}$. We describe a non-trivial procedure to calculate the syndrome in Lemma 4.1.

Let v be an integer where $0 < v \leq m(q-1)$, and let $\mu = m(q-1) - v$. Let $(c_P)_{P \in \mathbb{P}_m}$ be a codeword in $\text{PRM}_v(m, q)$. After an error word $(e_P)_{P \in \mathbb{P}_m}$ occurs, we assume that we receive the word $(r_P)_{P \in \mathbb{P}_m} = (c_P)_{P \in \mathbb{P}_m} + (e_P)_{P \in \mathbb{P}_m}$. Using the following settings, we can construct a decoding algorithm by which the error word $(e_P)_{P \in \mathbb{P}_m}$ may be corrected.

Let $i \in \{0, 1, \ldots, m\}$. We define a subset B_i of R_μ by

$$B_i = \left\{ \prod_{j=i}^{m} X_j^{a_j} \;\middle|\; \begin{array}{l} \sum_{j=i}^{m} a_j = \mu, \; 0 < a_i, \\ 0 \leq a_{i+1}, \ldots, a_m \leq q - 1 \end{array} \right\}. \tag{4.1}$$

We recall that Ψ_i is identified by $\{(0, \ldots, 0, 1, \omega_{i+1}, \ldots, \omega_m) \mid \omega_{i+1}, \ldots, \omega_m \in \mathbb{F}_q\} \subseteq \mathbb{A}_{m+1}$. Since $Z(\Psi_i) \subseteq R$ is generated by $\{X_0, \ldots, X_{i-1}, X_i - 1, X_{i+1}^q - X_{i+1}, \ldots, X_m^q - X_m\}$, we have $R/Z(\Psi_i) = \mathbb{F}_q[X_{i+1}, \ldots, X_m]/\langle X_{i+1}^q - X_{i+1}, \ldots, X_m^q - X_m \rangle$. Then, $B_i = \{\prod_{j=i+1}^{m} X_j^{a_j} \mid \sum_{j=i+1}^{m} a_j \leq \mu - 1, \; 0 \leq$

$a_{i+1}, \ldots, a_m \leq q - 1\}$ in $R/Z(\Psi_i)$, which is the set of monomials in $D(\Psi_i)$ of degree $\leq \mu - 1$. By Eq. (2.2) and Eq. (2.14),

$$C(\text{span}_{\mathbb{F}_q}(B_i), \Psi_i) = \text{RM}_{\mu-1}(m - i, q). \tag{4.2}$$

Therefore, $\left(\sum_{P \in \Psi_i} e_P h(P)\right)_{h \in B_i}$ is a syndrome of $(e_P)_{P \in \Psi_i}$ related to $\text{RM}_{\mu-1}(m - i, q)$. If we calculate the syndrome $\left(\sum_{P \in \Psi_i} e_P h(P)\right)_{h \in B_i}$, we can apply Step 2 and Step 3 of Algorithm 1 as $\Psi = \Psi_i$, $B = B_i$, $(S_h)_{h \in B} = \left(\sum_{P \in \Psi_i} e_P h(P)\right)_{h \in B_i}$ and $C^\perp(\text{span}_{\mathbb{F}_q}(B), \Psi) = \text{RM}_{\mu-1}(m - i, q)^\perp$. A procedure to obtain the syndrome is described later in Lemma 4.1.

Algorithm 2: Decoding algorithm for $\text{PRM}_\nu(m, q)$

Input: $(r_P)_{P \in \mathbb{P}_m} \in \mathbb{F}_q^{\mathbb{P}_m}$, where $(r_P)_{P \in \mathbb{P}_m} = (c_P)_{P \in \mathbb{P}_m} + (e_P)_{P \in \mathbb{P}_m}$, $(c_P)_{P \in \mathbb{P}_m} \in \text{PRM}_\nu(m, q)$ and $(e_P)_{P \in \mathbb{P}_m} \in \mathbb{F}_q^{\mathbb{P}_m}$

Output: $(\hat{e}_P)_{P \in \mathbb{P}_m}$

for $i \in \{0, 1, \ldots, m\}$ **do**

    (Step 1)

    if $i = 0$ **then**

        $r_P^{(0)} = r_P$ for $P \in \mathbb{P}_m$.

    else

        $r_P^{(i)} = r_P - \hat{e}_P$ for $P \in \bigcup_{j=0}^{i-1} \Psi_j$.

        $r_P^{(i)} = r_P$ for $P \in \bigcup_{j=i}^{m} \Psi_j$.

    end

    (Step 2)

    Calculate $S_h^{(i)} = \sum_{P \in \mathbb{P}_m} r_P^{(i)} h(P)$ for $h \in B_i$.

    (Step 3)

    Calculate $(\hat{e}_P)_{P \in \Psi_i}$ by Algorithm 1 as $\Psi = \Psi_i$, $B = B_i$ and $(S_h)_{h \in B} = \left(S_h^{(i)}\right)_{h \in B_i}$.

end

In Algorithm 2, $(\hat{e}_P)_{P \in \mathbb{P}_m} = (e_P)_{P \in \mathbb{P}_m}$ if $(\hat{e}_P)_{P \in \Psi_i} = (e_P)_{P \in \Psi_i}$ for all $i \in \{0, 1, \ldots, m\}$. Let i_0 be the smallest integer satisfying $\mu \geq (m - i_0)(q - 1) + 1$, i.e.,

$$i_0 = m - \left\lfloor \frac{\mu - 1}{q - 1} \right\rfloor. \tag{4.3}$$

If $i_0 \leq i \leq m$, then $(\hat{e}_P)_{P \in \Psi_i} = (e_P)_{P \in \Psi_i}$ for all $(e_P)_{P \in \Psi_i} \in \mathbb{F}_q^{\Psi_i}$. Indeed, since $\text{RM}_{\mu-1}(m - i, q) = \mathbb{F}_q^{\Psi_i}$, the dimension of $\text{RM}_{\mu-1}(m - i, q)^\perp$ is 0 (see the last paragraph of Section 2.3).

Here, we explain how we obtain the syndrome $\left(\sum_{P \in \Psi_i} e_P h(P)\right)_{h \in B_i}$ and how we apply Algorithm 1 in Algorithm 2. We fix an integer i where $0 \leq i \leq m$. In Step 1, if $i = 0$, then we set $r_P^{(0)} = r_P$ for $P \in \mathbb{P}_m$. If $0 < i \leq m$, we assume that we already know the $\Psi_0, \Psi_1, \ldots, \Psi_{i-1}$ components of the error word, i.e., $\hat{e}_P = e_P$ for all $P \in \bigcup_{j=0}^{i-1} \Psi_j$. We set a modified received word $(r_P^{(i)})_{P \in \mathbb{P}_m}$ by removing the $\Psi_0, \Psi_1, \ldots, \Psi_{i-1}$ components of the error word, i.e.,

$$r_P^{(i)} = \begin{cases} r_P - e_P & \text{if } P \in \bigcup_{j=0}^{i-1} \Psi_j, \\ r_P & \text{if } P \in \bigcup_{j=i}^{m} \Psi_j. \end{cases} \tag{4.4}$$

Then, $r_P^{(i)} = c_P$ if $P \in \bigcup_{j=0}^{i-1} \Psi_j$, and $r_P^{(i)} = c_P + e_P$ if $P \in \bigcup_{j=i}^{m} \Psi_j$.

In Step 2, we calculate $S_h^{(i)} = \sum_{P \in \mathbb{P}_m} r_P^{(i)} h(P)$ for $h \in B_i$. Since $(h(P))_{P \in \mathbb{P}_m} \in \mathrm{PRM}_\nu(m, q)^\perp$ for $h \in B_i$ by Theorem 2.4, we have that

$$\sum_{P \in \mathbb{P}_m} c_P h(P) = 0 \quad \text{for } h \in B_i. \tag{4.5}$$

Lemma 4.1 *We have that* $\left(S_h^{(i)}\right)_{h \in B_i}$ *is the syndrome of* $(e_P)_{P \in \Psi_i}$ *related to* $\mathrm{RM}_{\mu-1}(m-i, q)$*, i.e.,*

$$\left(S_h^{(i)}\right)_{h \in B_i} = \left(\sum_{P \in \Psi_i} e_P h(P)\right)_{h \in B_i}. \tag{4.6}$$

Proof: Let $h \in B_i$. It follows from Eq. (4.4) and Eq. (4.5) that

$$S_h^{(i)} = \sum_{P \in \mathbb{P}_m} r_P^{(i)} h(P) \tag{4.7}$$

$$= \sum_{P \in \mathbb{P}_m} c_P h(P) + \sum_{P \in \bigcup_{j=i}^m \Psi_j} e_P h(P) \tag{4.8}$$

$$= \sum_{P \in \bigcup_{j=i}^m \Psi_j} e_P h(P) = \sum_{P \in \Psi_i} e_P h(P), \tag{4.9}$$

where $h(P) = 0$ for $P \in \bigcup_{j=i+1}^m \Psi_j$, since the i-th exponent of h is positive and the i-th entry of P is 0.
$\square$

In Step 3, if $0 \le i < i_0$, then we apply Algorithm 1 from Eq. (4.6) as $\Psi = \Psi_i$, $B = B_i$. Thus, we obtain the Ψ_i-component of the error word. If $i_0 \le i \le m$, we obtain the Ψ_i-component of the error word by applying the IDFT to Eq. (4.6). By repeating Steps 1, 2 and 3 for $i \in \{0, 1, \ldots, m\}$, we complete the decoding procedure. We remark that corresponding codes to which we apply Algorithm 1 are listed in the middle column of Table 2.

5 Number of errors correctable

Let $0 < \nu \le m(q-1)$ and $\mu = m(q-1) - \nu$. Let Ψ be $\mathbb{P}_m$ (or resp. Ψ_i). The number of errors correctable for $\mathrm{PRM}_\nu(m, q)$ (or resp. $\mathrm{RM}_{\mu-1}(m-i, q)^\perp$) is defined by

$$\max\left\{|\Phi| \;\middle|\; \Phi \subseteq \Psi, \quad \begin{array}{c} (\hat{e}_P)_{P \in \Psi} = (e_P)_{P \in \Psi} \\ \text{for } (e_P)_{P \in \Psi} \in \mathbb{F}_q^\Psi \text{ with} \\ \Phi = \{P \in \Psi \mid e_P \neq 0\} \end{array} \right\}, \tag{5.1}$$

where $(\hat{e}_P)_{P \in \Psi}$ is the output of $(e_P)_{P \in \Psi}$ by applying Algorithm 2 to $\mathrm{PRM}_\nu(m, q)$ (or resp. Algorithm 1 to $\mathrm{RM}_{\mu-1}(m-i, q)^\perp$). We note that the output of $(e_P)_{P \in \Psi}$ coincides with that of $(c_P)_{P \in \Psi} + (e_P)_{P \in \Psi}$ for all codewords $(c_P)_{P \in \Psi}$, since the syndrome does not depend on codewords.

In this section, we determine the number of errors correctable for $\mathrm{PRM}_\nu(m, q)$. We recall that Algorithm 2 computes an error word correctly if Algorithm 1 computes the Ψ_i-component of the error word correctly for all $i \in \{0, 1, \ldots, m\}$. We set

$$t_0 = \left\lfloor \frac{(q-s)q^{m-r-1} - 1}{2} \right\rfloor, \tag{5.2}$$

where $\nu = r(q-1) + s$, $0 \le s < q-1$, $0 \le r \le m-1$. The numbers of errors correctable for $\mathrm{RM}_{\mu-1}(m-i, q)^\perp$ are determined in Proposition 5.1.

Proposition 5.1 *Let i_0 be the integer defined in* (4.3).

1. *If $0 \leq i < i_0$, then the number of errors correctable for $\mathrm{RM}_{\mu-1}(m-i,q)^{\perp}$ is t_0.*

2. *If $i_0 \leq i \leq m$, then the number of errors correctable for $\mathrm{RM}_{\mu-1}(m-i,q)^{\perp}$ is q^{m-i}.*

Proof: Assertion 2 has already been proved. Here, we prove Assertion 1. Let $0 \leq i < i_0$. By (2.6), we have

$$\mathrm{RM}_{\mu-1}(m-i,q)^{\perp} = \mathrm{RM}_{(m-i)(q-1)-(\mu-1)-1}(m-i,q) \tag{5.3}$$
$$= \mathrm{RM}_{\nu-i(q-1)}(m-i,q). \tag{5.4}$$

In addition, by [26, Proposition 4.16], there exists an ordered basis for $\mathrm{RM}_{\nu-i(q-1)}(m-i,q)$ such that $d_{\mathrm{FR}}(\mathrm{RM}_{\nu-i(q-1)}(m-i,q)) = d_{\min}(\mathrm{RM}_{\nu-i(q-1)}(m-i,q))$. Thus, by (2.24), the number of errors correctable is

$$\left\lfloor \frac{d_{\mathrm{FR}}(\mathrm{RM}_{\nu-i(q-1)}(m-i,q)) - 1}{2} \right\rfloor \tag{5.5}$$

$$= \left\lfloor \frac{d_{\min}(\mathrm{RM}_{\nu-i(q-1)}(m-i,q)) - 1}{2} \right\rfloor \tag{5.6}$$

$$= \left\lfloor \frac{(q-s)q^{(m-i)-(r-i)-1} - 1}{2} \right\rfloor \quad \text{(by Eq. (2.4))} \tag{5.7}$$

$$= \left\lfloor \frac{(q-s)q^{m-r-1} - 1}{2} \right\rfloor = t_0. \quad \square \tag{5.8}$$

The result of Proposition 5.1 is listed in the rightmost column of Table 2.

Corollary 5.2 *Let t be the number of errors correctable for $\mathrm{PRM}_{\nu}(m,q)$. Then, we have $t = t_0$.* $\quad\square$

Proof: By Theorem 5.1, we have $t \geq t_0$. If $\{P \in \mathbb{P}_m \mid e_P \neq 0\} \subseteq \Psi_1$ and $|\{P \in \mathbb{P}_m \mid e_P \neq 0\}| = t_0 + 1$, it does not always hold that $(\hat{e}_P)_{P \in \mathbb{P}_m} = (e_P)_{P \in \mathbb{P}_m}$. Hence, we have $t \leq t_0$ $\quad\square$

Thus, the number of errors correctable for $\mathrm{PRM}_{\nu}(m,q)$ is the same as that for $\mathrm{RM}_{\mu-1}(m,q)^{\perp}$. In special error cases, Algorithm 2 can correct more errors than t_0 which is described in Section 8.

6 Numerical example

In this section, we present a numerical example of a decoding procedure related to a three-dimensional projective space. To the best of our knowledge, this is the first example for three-dimensions in the literature. We consider the case when $m = 3, q = 4, \nu = 5$. The code length and dimension of $\mathrm{PRM}_5(3,4)$ are $n = 85$ and $k = 50$, respectively. By Theorem 2.4, we have $\mathrm{PRM}_5(3,4)^{\perp} = \mathrm{PRM}_4(3,4)$. Let α be a generator of a cyclic group $\mathbb{F}_4^{\times}$ satisfying $\alpha^2 + \alpha + 1 = 0$, and β denotes α^2. Then, $\mathbb{F}_q = \{0, 1, \alpha, \beta\}$.

Fig. 3 presents a numerical example for applying Algorithm 2 to $\mathrm{PRM}_5(3,4)$. At Information polynomial of Fig. 3, we show the coefficients of $f \in R_5$. The (i,j)th entry of the 4×4 matrix named $a_3 = l$ of B_0 is the coefficient of $X_0^{5-i-j-l}X_1^i X_2^j X_3^l$. Similarly, we show coefficients of B_1, B_2 and B_3 by matricies. For example, the coefficient of $X_0^3 X_1^2$ is α, that of $X_1^4 X_2$ is β. At Codeword, we show the values c_P indexed by $P \in \mathbb{P}_3$. For example, $c_{(1:0:1:\beta)} = \alpha$, $c_{(0:0:1:\alpha)} = \beta$.

Table 2: (Left) Components of $\mathbb{P}_m$, (Middle) corresponding codes to which we apply Algorithm 1 and (Right) component-wise numbers of errors correctable

Components	Corresponding codes	Numbers of errors correctable
Ψ_0	$\mathrm{RM}_{\mu-1}(m,q)^\perp$	t_0
Ψ_1	$\mathrm{RM}_{\mu-1}(m-1,q)^\perp$	t_0
Ψ_2	$\mathrm{RM}_{\mu-1}(m-2,q)^\perp$	t_0
$\vdots$	$\vdots$	$\vdots$
Ψ_{i_0-1}	$\mathrm{RM}_{\mu-1}(m-i_0+1,q)^\perp$	t_0
Ψ_{i_0}	$\left(\mathbb{F}_q^{\Psi_{m-i_0}}\right)^\perp$	$q^{m-i_0} = |\Psi_{m-i_0}|$
$\vdots$	$\vdots$	$\vdots$
Ψ_{m-1}	$\left(\mathbb{F}_q^{\Psi_1}\right)^\perp$	$q^1 = |\Psi_1|$
Ψ_m	$\left(\mathbb{F}_q^{\Psi_0}\right)^\perp$	$1 = |\Psi_0|$

We have $i_0 = 2$ and $t_0 = 3$. In the Ψ_i-component for $i \in \{0,1\}$, we use the monomial order $\prec$ defined in Section 3, and correct three errors. For example, if $i = 0$, monomials arranged as follows: $1 \prec X_1 \prec X_2 \prec X_3 \prec X_1^2 \prec X_2 X_1 \prec X_2^2 \prec X_3 X_1 \prec \cdots$. Moreover, we obtain and use Gröbner bases $\mathcal{G}^{(0)} = \{g_1^{(0)} = X_2^2 + \alpha X_2 + \beta X_1, g_2^{(0)} = X_2 X_1 + X_2 + \alpha X_1 + \alpha, g_3^{(0)} = X_1^2 + X_1, g_4^{(0)} = X_3 + \alpha X_2 + 1\}$ in the Ψ_0-component, and $\mathcal{G}^{(1)} = \{g_1^{(1)} = X_3^2 + \beta X_2 + \beta, g_2^{(1)} = X_3 X_2 + X_3 + \alpha, g_3^{(1)} = X_2^2 + \beta X_2 + 1\}$ in the Ψ_1-component.

We correct all error words in the Ψ_i-component if $i \in \{2,3\}$. The number of errors correctable are four and one in the Ψ_2- and the Ψ_3-component, respectively.

7 Computational complexity

In this section, we calculate computational complexities of Algorithm 2 based on the total number of finite-field operations. For each Ψ_i-component of Algorithm 2, the error positions are determined in Step 2 of Algorithm 1 and the error values e_P for all $P \in \Psi_i$ are determined in Step 3 of Algorithm 1. To observe a precise complexity, we separate the decoding procedure into the error position determination and the error value determination.

Definition 7.1 *Let $f(q)$ and $g(q)$ be two functions defined on a subset of real numbers. We write $f(q) = O(g(q))$ if and only if there exist constants q_0 and C such that $|f(q)| \leq C|g(q)|$ for all $q > q_0$.* $\square$

Let $N_i = q^{m-i}$ be the cardinality of Ψ_i, and z_i the cardinality of the Gröbner basis obtained by the BMS algorithm for the Ψ_i-component for $i \in \{0, 1, \ldots, m\}$.

Theorem 7.2 *Let $n = (q^{m+1} - 1)/(q - 1) = q^m + \cdots + q + 1$ the length of $\mathrm{PRM}_\nu(m,q)$.*

1. *The computational complexity of the error position determination of Algorithm 2 is $O(zn^2)$, where $z = \max\{z_0, z_1, \ldots, z_m\} \leq N_0/q = q^{m-1} < n/q$.*

2. *The computational complexity of the error value determination of Algorithm 2 is $O(qn^2)$.*

3. *The total complexity of Algorithm 2 is $O(wn^2)$, where $w = \max\{q, z\} \leq q^{m-1} < n/q$.*

Figure 3: Decoding example for $PRM_4(3, 4)$

Proof: For the Ψ_i-component, the computational complexities of the error position determination and the error value determination are $O(z_i N_i^2) = O(z_i q^{2m-2i})$ [22], [25] and $O(q N_i^2) = O(q^{2m-2i+1})$ [15], respectively. According to [22], [25], we have $z_i \leq N_i/q = q^{m-i-1} < N_i$ for all $i \in \{0, 1, \ldots, m\}$. Hence, the computational complexity of the error position determination in Algorithm 2 is $O(\sum_{i=0}^{m} z_i q^{2m-2i})$, and that of the error value determination is $O(\sum_{i=0}^{m} q^{2m-2i+1})$.

Since the proofs of assertions 1 and 2 are similar and assertion 3 follows from 1 and 2, we verify only assertion 1. For all $q > 1$, we have $q^2/2 < q^2 - 1$. Thus,

$$z_0 q^{2m} + z_1 q^{2m-2} + z_2 q^{2m-4} + \cdots + z_m \tag{7.1}$$

$$\leq z(q^{2m} + q^{2(m-1)} + q^{2(m-2)} + \cdots + 1^2) \tag{7.2}$$

$$= z \frac{q^{2m+2} - 1}{q^2 - 1} < z \frac{2q^{2m+2}}{q^2} = 2z q^{2m}. \tag{7.3}$$

This means $\sum_{i=0}^{m} z_i q^{2m-2i} = O(z q^{2m})$. It is clear that $z q^{2m} < z n^2$ for all $q > 1$, and then $z q^{2m} = O(z n^2)$. $\square$

We note that Theorem 2 does not depend on ν, because ν only affects $|B_i|$ which can be replaced by an upper bound $|\Psi_i| = q^{m-i}$ during the complexity analysis.

From the proof of Theorem 7.2, the computational complexities are $O(yq^{2m})$ and $yq^{2m} = O(yn^2)$, where $y = z$, $y = q$ or $y = w$. We also have $yn^2 = O(yq^{2m})$. Indeed, since $(q-1)^2 - (q^2/2) = (1/2)(q^2 - 4q + 2) = (1/2)(q-2)^2 - 1 > 0$ for all $q > 3$, we have

$$yn^2 = y\left(\frac{q^{m+1} - 1}{q - 1}\right)^2 < y\frac{q^{2m+2}}{(q-1)^2} < 2yq^{2m}. \tag{7.4}$$

In this sense, Theorem 7.2 is an optimal evaluation for the computational complexity of Algorithm 2.

8 Codeword error rate comparison with MDD

In this section, we investigate the codeword error rate of Algorithm 2 and compare it with that of the MDD which achieves the best rate of the three previous methods described in Introduction. We consider two types of errors correctable. In the first type, the number of errors correctable is t_0, and such errors are always correctable (see Corollary 5.2). The second type is a specialized case, for which the number of errors correctable has been listed component-wise in Table 2. These two types have different codeword error rates. We refer to the decoding method for the first and second cases as Proposed Method 1 (PM1) and Proposed Method 2 (PM2), respectively. Let p be a symbol error rate. The codeword error rate of PM1 is then $1 - P$, where $P = \sum_{j=0}^{t_0} \binom{n}{j} p^j (1-p)^{n-j}$. The codeword error rate of PM2 is $1 - \prod_{i=0}^{i_0 - 1} P_i$, where $P_i = \sum_{j=0}^{t_0} \binom{q^{m-i}}{j} p^j (1-p)^j$ for $i \in \{0, 1, \ldots, i_0 - 1\}$.

Tables 3 and 4 list numerical examples of the number of errors correctable by PM1 and the MDD. In these tables, the double lines indicate the turning positions of the quotient obtained when ν is divided by $q - 1$. The difference between the number of errors correctable decreases when the above-mentioned quotient increases. Let t_{MD} be the number of errors correctable by the MDD. The codeword error rate of the MDD is $1 - \sum_{j=0}^{t_{MD}} \binom{n}{j} p^j (1-p)^{n-j} = 1 - P - \sum_{j=t_0+1}^{t_{MD}} \binom{n}{j} p^j (1-p)^{n-j}$. Recall that $1 - P$ is the codeword error rate of PM1. Therefore, the lower the difference $t_{MD} - t_0$ between the number of errors correctable by PM1 and the MDD, the lower the difference between their codeword error rates. In the right hand side of Table 3, i.e., where the quotient obtained by dividing ν by $q - 1$ is $m - 1$, the difference is one or less. Further, in some cases, the codeword error rate of PM1 coincides with that of the MDD.

Figs. 4 and 5 show the codeword error rates for $\mathrm{PRM}_{17}(2, 16)$ and $\mathrm{PRM}_9(3, 8)$. When ν is sufficiently large, the performance curves of PM1 and PM2 are close to that of the MDD, as shown in Fig. 4. In Fig. 5, the performance curve of PM2 is distinct from that of PM1 because the cardinality and number of errors correctable are not negligible.

9 Conclusion

In this paper, we have constructed a decoding algorithm for all PRM codes by dividing a projective space into a union of affine spaces. We have determined the number of errors correctable for $\mathrm{PRM}_\nu(m, q)$. Although it is the same as the number of errors correctable for $\mathrm{RM}_\nu(m, q)$, advantages of Algorithm 2 are that the codeword is longer and the code parameters are more flexible. We have also proved that the computational complexities of Algorithm 2 is $O(wn^2)$, where $w = \max\{q, z_0, z_1, \ldots, z_m\}$ is less than n/q. Finally, we compared the codeword error rate of three types of decoding procedures. When the order of a PRM code is sufficiently high, the codeword error rate of Algorithm 2 is close to that of the MDD. Further improvement of our algorithm is required to decrease the difference between

Table 3: Number of errors correctable by Algorithm 2 and the MDD for $\mathrm{PRM}_\nu(2, 16)$

ν	5	8	11	14	17	20	23	26	29
Algorithm 2	87	63	39	15	6	5	3	2	0
MDD	95	71	47	23	7	5	4	2	1
Difference	8	8	8	8	1	0	1	0	1

Table 4: Number of errors correctable by Algorithm 2 and the MDD for $\mathrm{PRM}_\nu(3, 8)$

ν	2	4	6	9	12	14	16	18
Algorithm 2	191	127	63	23	11	7	3	2
MDD	223	159	95	27	15	7	3	2
Difference	32	32	32	4	4	0	0	0

its codeword error rate and that of the MDD. This could be a topic for future studies regarding the decoding theory of PRM codes.

Acknowledgments

References

[1] G. Lachaud, "Projective Reed–Muller codes," *Coding Theory and Applications,* pp.125–129, vol.311, Berlin: Springer 1988.

[2] A. B. Sørensen, "Projective Reed–Muller codes," *IEEE Trans. Inf. Theory,* vol.37, no. 6, pp.1567–1576, Nov. 1991.

[3] T. P. Berger, L. de Maximy, "Cyclic projective Reed–Muller codes," *In Proc. of Applied Algebra, Algebraic Algorithms and Error Correcting Codes,* vol. 2227, pp.77–81, Oct. 2001.

[4] S. Ballet, R. Rolland, "On low weight codewords of generalized affine and projective Reed–Muller codes," *Designs Codes Cryptogr.,* vol.73, no.2, pp.271–297, Nov. 2014

[5] A. Dür, "The decoding of extended Reed–Solomon codes," *Discrete Math.,* vol.90, no.1, pp.21–40, June 1991.

[6] I. Duursma, "Decoding codes from curves and cyclic codes," Thesis, Technische Universiteit Eindhoven. Sep. 1993.

[7] N. Nakashima, H. Matsui, "A decoding algorithm for projective Reed–Muller codes of 2-dimensional projective space with DFT," *2014 International Symposium on Information Theory and its Applications (ISITA2014),* Melbourne, Australia, pp.371–375, Oct. 2014.
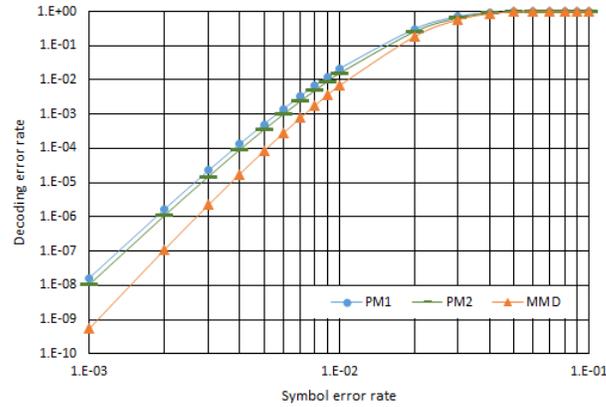
Figure 4: Comparison of codeword error rates for $\mathrm{PRM}_{17}(2, 16)$
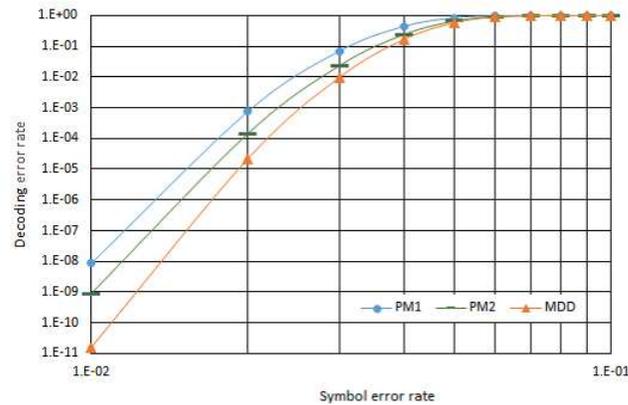
Figure 5: Comparison of codeword error rates for $\mathrm{PRM}_9(3, 8)$

[8] J. Justesen, T. Høholdt, *A Course in Error-Correcting Codes,* European Mathematical Society Publishing House, 2004.

[9] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, 1977.

[10] R. Pellikaan, "On decoding by error location and dependent sets of error positions," *Discrete Math.,* vol.106, pp.369–381, Sep. 1992.

[11] G. L. Feng, T. R. N. Rao, "Decoding algebraic geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory,* vol.39, pp.37–45, Jan. 1993.

[12] S. Miura, "Linear codes on affine algebraic varieties," *IEICE Trans. Fundam. Electron. Commun. Comput.,* vol.J81-A, no.10, pp.1386–1397, Oct. 1998. (in Japanese)

[13] H. E. Andersen, O. Geil, "Evaluation codes from order domain theory," *Finite Fields Appl.,* vol.14, no.1, pp.92–123, Jan. 2008

[14] R. Matsumoto, S. Miura, "On the Feng–Rao bound for the $\mathcal{L}$-construction of algebraic geometry codes," *IEICE Trans. Fundam. Electron. Commun. Comput.,* E83-A, no.5, pp.923–926, May 2000.

[15] H. Matsui, "Lemma for linear feedback shift registers and DFTs applied to affine variety codes," *IEEE Trans. Inf. Theory,* vol.60, no.5, pp.2751–2769, May 2014.

[16] S. Sakata, "Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array," *J. Symb. Comput.,* vol.5, no.3, pp.321–337, June 1988.

[17] S. Sakata, "Extension of the Berlekamp–Massey algorithm to N dimensions," *Inform. Comput.,* vol.84, no.2, pp.207–239, Feb. 1990.

[18] S. Sakata, H. E. Jensen, T. Høholdt, "Generalized Berlekamp–Massey decoding of algebraic-geometric codes up to half the Feng–Rao bound," *IEEE Trans. Inf. Theory,* vol.41, no.6, pp.1762–1768, Nov. 1995.

[19] S. Sakata, D. A. Leonard, H. E. Jensen, T. Høholdt, "Fast erasure-and-error decoding of algebraic geometry codes up to the Feng–Rao bound," *IEEE Trans. Inf. Theory,* vol.44, no.4, pp.1558–1564, July 1998.

[20] I. F. Blake and R. C. Mullin, *The Mathematical Theory of Coding*, New York: Academic Press, 1975.

[21] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms,* 2nd ed., Springer, Berlin, 1997.

[22] D. Cox, J. Little, D. O'Shea, "The Berlekamp–Massey–Sakata algorithm," *Using Algebraic Geometry,* 2nd ed., Chapter 10, pp.494–532, Springer, Berlin, 2004.

[23] K. Saints, C. Heegard, "Algebraic–geometric codes and multidimensional cyclic codes: Theory and algorithms for decoding using Gröbner bases," *IEEE Trans. Inf. Theory,* vol.41, no.6, pp.1733–1751, Nov. 1995.

[24] J. Fitzgerald, R. F. Lax, "Decoding affine variety codes using Gröbner bases," *Designs Codes Cryptogr.,* vol.13, no.2, pp.147–158, Feb. 1998

[25] M. Bras-Amorós, M. E. O'Sullivan, "The correction capability of the Berlekamp–Massey–Sakata algorithm with majority voting," *Appl. Algebr. Eng. Commun. Comput.*, vol.17, no.5 pp.315–335, May 2006.

[26] P. Heijnen, R. Pellikaan, "Generalized Hamming weights of q-ary Reed–Muller codes," *IEEE Trans. Inf. Theory,* vol.44, no.1, pp.181–196, Jan. 1998.

[27] N. Nakashima, H. Matsui, "Correction of errors for projective RM codes by decomposing projective space into affine spaces," *in 37th Symposium on Information Theory and its Applications (SITA2014),* pp.89–94, Dec. 2014.