

Automorphism Groups of Error-Correcting Codes and Reed-Muller Codes

Notes for Error-Correcting Codes
Henry D. Pfister

February 19th, 2019

1 Introduction

The symmetry groups of error-correcting codes have now been studied for quite some time. In some cases, e.g., the Golay code, they give rise to groups that are mathematically interesting in their own right. In other cases, they have been used to help calculate weight enumerators of codes. Recently, symmetry has also been used to prove that a sequence of deterministic codes achieves capacity on the BEC. This handout also introduces Reed-Muller codes, which are a class of binary codes with large symmetry groups. Much of the material in this handout can be found in [1].

2 Permutation Automorphisms

Let \mathcal{X} be a finite alphabet and $\mathcal{C} \subseteq \mathcal{X}^n$ be a code. Let S_n denote the symmetric group on n elements so that $\sigma \in S_n$ is a bijective function $\sigma: [n] \rightarrow [n]$. For $\sigma \in S_n$ and $\underline{x} \in \mathcal{X}^n$, let $\underline{x}_\sigma = (x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)})$. Similarly, for a matrix $A \in \mathcal{X}^{m \times n}$ with columns $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n$, let $A_\sigma = [\underline{a}_{\sigma(1)}, \underline{a}_{\sigma(2)}, \dots, \underline{a}_{\sigma(n)}]$.

Definition 2.1. A code $\mathcal{C}' \subseteq \mathcal{X}^n$ is called *equivalent* to \mathcal{C} if there is a permutation $\sigma \in S_n$ such that $\underline{x}_\sigma = (x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)}) \in \mathcal{C}'$ for all $\underline{x} \in \mathcal{C}$.

Definition 2.2. The *permutation automorphism group* of \mathcal{C} is defined to be

$$\text{Per}(\mathcal{C}) \triangleq \{ \sigma \in S_n \mid (x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)}) \in \mathcal{C} \forall \underline{x} \in \mathcal{C} \}.$$

These permutations form a subgroup of S_n because, if two permutations separately preserve the set of codewords, then their composition also preserves the set of codewords. Thus, the set of permutations in $\text{Per}(\mathcal{C})$ is closed under composition and must form a subgroup.

Example 2.3. Consider the $(4, 2, 2)$ linear code with codewords

$$\mathcal{C} = \left\{ \begin{array}{l} 0000 \\ 1100 \\ 0011 \\ 1111 \end{array} \right\}.$$

Observe that swapping $x_1 \leftrightarrow x_2$ or $x_3 \leftrightarrow x_4$ leaves the list of codewords unchanged. Also, swapping $x_1x_2 \leftrightarrow x_3x_4$ changes the list but this can be undone by swapping the 2nd and 3rd codewords. Thus, all of these permutations are automorphisms.

Remark 2.4. The terms homomorphism, isomorphism, and automorphism are used throughout mathematics to describe mappings that preserve some mathematical structure. For standard algebraic objects (such as groups, fields, and vector spaces), the definitions of these terms are well known. But, what key structure of the code is preserved by the permutation automorphism? Though this question does

not have a unique answer, the right answer is the Hamming distance. Thus, a code is treated as metric space when discussing morphisms. This means the full automorphism group may also include non-permutation mappings such as the pointwise relabeling of the codeword symbols. For example, $\rho(\underline{x}) \triangleq (\rho(x_1), \rho(x_2), \dots, \rho(x_n))$ could apply the bijection $\rho: \mathcal{X} \rightarrow \mathcal{X}$ to each coordinate. For linear codes, these morphisms must also preserve linearity.

For linear codes over \mathbb{F}_q , the permutation automorphism group satisfies a number of properties. For example, we will show that $\text{Per}(\mathcal{C}) = \text{Per}(\mathcal{C}^\perp)$, where \mathcal{C}^\perp is the dual code.

Definition 2.5. For a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$, the *dual code* is defined by

$$\mathcal{C}^\perp \triangleq \left\{ \underline{y} \in \mathbb{F}_q^n \mid \sum_{i=1}^n x_i y_i = 0 \forall \underline{x} \in \mathcal{C} \right\}.$$

The dual code is equal to the set of parity checks that all codewords satisfy. Thus, it is generated by the parity-check matrix of the original.

Lemma 2.6. $\text{Per}(\mathcal{C}) = \text{Per}(\mathcal{C}^\perp)$

Proof. For all $\sigma \in \text{Per}(\mathcal{C})$, $\underline{x} \in \mathcal{C}$, and $\underline{y} \in \mathcal{C}^\perp$, Definitions 2.2 and 2.5 imply that

$$\sum_{i=1}^n x_{\sigma(i)} y_i = 0.$$

Changing the variable of summation to $j = \sigma(i)$ shows that

$$\sum_{j=1}^n x_j y_{\sigma^{-1}(j)} = 0.$$

Thus, $\sigma^{-1} \in \text{Per}(\mathcal{C}^\perp)$ because $\underline{y}_{\sigma^{-1}} \in \mathcal{C}^\perp$ for all $\sigma \in \text{Per}(\mathcal{C})$ and $\underline{y} \in \mathcal{C}^\perp$. Since every element in a group has a unique inverse in the group, this implies that $\text{Per}(\mathcal{C}) \subseteq \text{Per}(\mathcal{C}^\perp)$. Also, a symmetric argument holds starting from $\sigma \in \text{Per}(\mathcal{C}^\perp)$, so we see that $\text{Per}(\mathcal{C}^\perp) \subseteq \text{Per}(\mathcal{C})$. Hence, the two groups are equal. \square

Theorem 2.7. For an (n, k) linear code \mathcal{C} over \mathbb{F}_q , we have $\sigma \in \text{Per}(\mathcal{C})$ if and only if there is an invertible $k \times k$ matrix L over \mathbb{F}_q such that $LG_\sigma = G$, where G is a generator matrix for \mathcal{C} . Similarly, we have $\sigma \in \text{Per}(\mathcal{C})$ if and only if there is an invertible $(n - k) \times (n - k)$ matrix L' over \mathbb{F}_q such that $L'H_\sigma = H$, where H is a parity-check matrix for \mathcal{C} .

Proof. Since a linear code \mathcal{C} over \mathbb{F}_q is a k -dimensional subspace of \mathbb{F}_q^n , a matrix G' is a generator for \mathcal{C} if and only if its rows form a basis for \mathcal{C} (i.e., it can be transformed by left-multiplication into a known generator for \mathcal{C}). Thus, the matrix G_σ is a generator for \mathcal{C} if and only if there is an invertible $k \times k$ matrix L satisfying $LG_\sigma = G$.

For the case of parity-check matrices, Definition 2.5 notes that the parity-check matrix generates the dual code. Thus, for any $\sigma \in \text{Per}(\mathcal{C}^\perp)$, there is an invertible $(n - k) \times (n - k)$ matrix L satisfying $LH_\sigma = H$. Since Lemma 2.6 shows that $\text{Per}(\mathcal{C}) = \text{Per}(\mathcal{C}^\perp)$, the same result also hold for all $\sigma \in \text{Per}(\mathcal{C})$. \square

Corollary 2.8. For a linear code \mathcal{C} , the permutation group $\text{Per}(\mathcal{C})$ is isomorphic to some subgroup of $GL_m(\mathbb{F}_q)$ (i.e., the general linear group of invertible $m \times m$ matrices over \mathbb{F}_q) where $m = \min\{k, n - k\}$.

Proof. Without loss of generality, assume that $k \leq n - k$. For each $\sigma \in \text{Per}(\mathcal{C})$, there is an invertible $m \times m$ matrix L that induces the permutation σ on the columns of the generator G . Thus, we can represent the composition of elements $\sigma, \sigma' \in \text{Per}(\mathcal{C})$ by the product of invertible $m \times m$ matrices L, L' over \mathbb{F}_q . In particular, we can read off the permutation $\sigma'(\sigma(\cdot))$ associated with $L'L$ by computing $G' = L'LG$ and then matching the columns with G . If all columns are unique (i.e., the minimum distance is at least 3), then the permutation will be specified uniquely. Otherwise, there will be trivial automorphisms that interchange identical columns and leave the information sequence unchanged. \square

Problem 2.9. Consider the (1,4) binary Reed-Muller code \mathcal{C} of length $N = 16$ with the following generator matrix:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

If π is a permutation on the $N = 16$ bits and P_π is the associated permutation matrix, then the action of π on the code is defined by the matrix $A = GP_\pi$. Recall that if and only if there exists a matrix $K \in GL_5(\mathbb{F}_2)$ such that $KG = A$ (i.e. A is also a generator for \mathcal{C}). For the following two permutations, if $\pi \in \text{Per}(\mathcal{C})$ calculate K . Otherwise, show that such a K does not exist and $\pi \notin \text{Per}(\mathcal{C})$.

1. The permutation $\pi_1 = (2, 4)$ which that swaps bits 2 and 4 (with bit indexing starting with 0) and fixes all other bits,
2. The permutation π_2 defined by

$$\pi_2 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 7 & 2 & 5 & 4 & 3 & 6 & 1 & 12 & 11 & 14 & 9 & 8 & 15 & 10 & 13 \end{pmatrix}.$$

Hint: Find a full rank 5×5 submatrix of G and its mapping in A to solve for K and check if it exists.

Definition 2.10. A permutation group $\mathcal{G} \subseteq S_n$ is called *transitive* if, for all $i, j \in [n]$, there is a permutation $\sigma \in \mathcal{G}$ such that $\sigma(i) = j$. It is called *doubly transitive* if, for all $i, j, k, l \in [n]$ such that $i \neq j$ and $k \neq l$, there is a permutation $\sigma \in \mathcal{G}$ such that $\sigma(i) = k$ and $\sigma(j) = l$.

Remark 2.11. For a code \mathcal{C} on a memoryless channel where $\text{Per}(\mathcal{C})$ is transitive, the probability of symbol error under ML decoding is the same for all symbols. For a code \mathcal{C} where $\text{Per}(\mathcal{C})$ is doubly transitive, all pairs of symbols have the same joint distribution of errors.

Example 2.12. Consider the (7,4) binary Hamming code \mathcal{C} defined by the parity-check matrix

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{1}$$

The columns of H consist of all non-zero binary vectors of length 3. If $H' = L^{-1}H$ for some invertible 3×3 matrix L , then this statement remains true for H' because the minimum distance of the code is preserved. Thus, for each invertible 3×3 matrix L , we can read off a permutation $\sigma \in \text{Per}(\mathcal{C})$ by matching the columns of H and H' . Since the columns of H are in the natural binary order, it follows that $\sigma(i)$ equals the decimal number associated with i -th column of H' . In particular,

$$H' = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} = H_\sigma$$

for $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 6 & 5 & 4 & 7 & 2 & 1 \end{pmatrix}$. Combined with Corollary 2.2, this shows that $\text{Per}(\mathcal{C}) \simeq GL_3(\mathbb{F}_2)$.

In addition, the set of permutations in $\text{Per}(\mathcal{C})$ is defined by the action of $GL_3(\mathbb{F}_2)$ on the set of non-zero binary vectors of length 3. Thus, $\text{Per}(\mathcal{C})$ is doubly transitive because, for any pair of distinct binary vectors of length 3, there is an invertible matrix $A \in GL_3(\mathbb{F}_2)$ that maps them to any other pair of distinct binary vectors.

Definition 2.13. Let $\mathcal{F}(q, m, r)$ denote the set of m -variate polynomials over \mathbb{F}_q with degree at most r . This means that each $f \in \mathcal{F}(q, m, r)$ is a mapping $f: \mathbb{F}_q^m \rightarrow \mathbb{F}_q$. Mathematically, we can write

$$\mathcal{F}(q, m, r) \triangleq \left\{ \sum_{\underline{i} \in \{0, 1, \dots, q-1\}^m: |\underline{i}| \leq r} a_{\underline{i}} \left(\prod_{j=1}^m x_j^{i_j} \right) \in \mathbb{F}_q[x_1, x_2, \dots, x_m] \mid a_{\underline{i}} \in \mathbb{F}_q \right\},$$

where $|\underline{i}| \triangleq \sum_{j=1}^m i_j$. We note that powers greater than $q-1$ are not included because $x^{q-1+i} = x^i$ for all $x \in \mathbb{F}_q$.

Example 2.14. Let $\mathcal{M} \triangleq \mathcal{F}(q, 1, k-1) = \{f \in \mathbb{F}_q[x] \mid \deg(f(x)) \leq k-1\}$ be the set of polynomials with coefficients in \mathbb{F}_q and degree at most $k-1$. For $n = q$, let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be the $(n, k, n-k+1)$ Reed-Solomon code defined by evaluating all polynomials in \mathcal{M} on all points in the field. In particular, we let $\theta: [q] \rightarrow \mathbb{F}_q$ be a bijection and define

$$\mathcal{C} = \{\underline{c} \in \mathbb{F}_q^n \mid f \in \mathcal{M}, c_i = f(\theta(\ell)), \ell \in [n]\}.$$

For $a, b \in \mathbb{F}_q$ with $a \neq 0$, the affine mapping $ax + b$ defines a bijection on \mathbb{F}_q because $ax_1 + b = ax_2 + b$ if and only if $x_1 = x_2$. This mapping also preserves degree because

$$f(ax + b) = \sum_{i=0}^{k-1} (ax + b)^i f_i = \sum_{j=0}^{k-1} x^j \sum_{i=0}^{k-1} f_i \binom{i}{j} a^j b^{i-j}$$

satisfies $\deg(f(ax + b)) = \deg(f(x))$. Thus, for each $f \in \mathcal{M}$, we have $f(ax + b) \in \mathcal{M}$ and defining $\sigma_{a,b}(\ell) \triangleq \theta^{-1}(a\theta(\ell) + b)$ shows that $\sigma_{a,b} \in \text{Per}(\mathcal{C})$ because $f(a\theta(\ell) + b) = f(\theta(\sigma_{a,b}(\ell)))$ for all $a, b \in \mathbb{F}_q$ and $a \neq 0$. Finally, the group $\text{Per}(\mathcal{C})$ is doubly transitive because, for any $w, x, y, z \in \mathbb{F}_q$ such that $w \neq x$ and $y \neq z$, one can fit a line to find $a, b \in \mathbb{F}_q$ such that $a \neq 0$, $y = aw + b$ and $z = ax + b$.

Example 2.15. Let $\mathcal{M}' \triangleq \mathcal{F}(2, m, r) = \{f \in \mathbb{F}_2[x_1, x_2, \dots, x_m] \mid \deg(f(x)) \leq r\}$ be the set of m -variate polynomials with coefficients in \mathbb{F}_2 and degree at most r . For $n = 2^m$, let $\mathcal{R}(r, m) \subseteq \mathbb{F}_2^n$ be the length- $n = 2^m$ Reed-Muller code defined by evaluating all polynomials in \mathcal{M}' on all points in $\underline{x} \in \mathbb{F}_2^m$. In particular, we let $\theta: [n] \rightarrow \mathbb{F}_2^m$ be a bijection and define

$$\mathcal{C} = \{\underline{c} \in \mathbb{F}_2^n \mid f \in \mathcal{M}', c_\ell = f(\theta(\ell)), \ell \in [n]\}.$$

For an invertible matrix $A \in GL_m(\mathbb{F}_q)$ and column vector $\underline{b} \in \mathbb{F}_q^m$, the affine mapping $A\underline{x} + \underline{b}$ defines a bijection on \mathbb{F}_q^m because $A\underline{x} + \underline{b} = A\underline{y} + \underline{b}$ if and only if $\underline{x} = \underline{y}$. This mapping also preserves degree because, for $|\underline{i}| \triangleq \sum_{j=1}^m i_j$, we have

$$f(A\underline{x} + \underline{b}) = \sum_{\underline{i} \in \{0, 1\}^m: |\underline{i}| \leq r} f_{\underline{i}} \prod_{j=1}^m \left(\sum_{k=1}^m A_{j,k} x_k + b_j \right)^{i_j}$$

has degree at most r . Thus, for each $f \in \mathcal{M}'$, we have $f(A\underline{x} + \underline{b}) \in \mathcal{M}'$ and defining $\sigma_{A,\underline{b}}(\ell) \triangleq \theta^{-1}(A\theta(\ell) + \underline{b})$ shows that $\sigma_{A,\underline{b}} \in \text{Per}(\mathcal{C})$ because $f(A\theta(\ell) + \underline{b}) = f(\theta(\sigma_{A,\underline{b}}(\ell)))$ for all \underline{b} if $A \in \mathbb{F}_2^{m \times m}$ is invertible. Finally, the group $\text{Per}(\mathcal{C})$ is doubly transitive because, for any $\underline{w}, \underline{x}, \underline{y}, \underline{z} \in \mathbb{F}_2^m$ such that $\underline{w} \neq \underline{x}$ and $\underline{y} \neq \underline{z}$, there is an $A \in GL_m(\mathbb{F}_q)$ and a $\underline{b} \in \mathbb{F}_2^m$ such that $\underline{y} = A\underline{w} + \underline{b}$ and $\underline{z} = A\underline{x} + \underline{b}$.

Theorem 2.16 ([2]). *Let \mathcal{C}_m be a sequence of binary codes with increasing blocklength whose rates converge to $R \in (0, 1)$. If these codes are transmitted over a BEC(ϵ), with $\epsilon < 1 - r$, and $\text{Per}(\mathcal{C}_m)$ is doubly transitive for all m , then the bit erasure rate of optimal decoding converges to 0 as $m \rightarrow \infty$. Thus, this code sequence achieves capacity.*

Corollary 2.17 ([2]). *A sequence $\mathcal{C}_m = \mathcal{R}(r_m, m)$ of Reed-Muller codes achieves capacity on the BEC if the implied rate sequence R_m converges to some $R \in (0, 1)$.*

$\prod_{j=1}^m z_j^{x_j}$ is given by $f(\underline{x})$. This gives

$$\begin{aligned}
g(\underline{z}) &= \sum_{\underline{x} \in \{0,1\}^m} f(\underline{x}) \left(\prod_{j=1}^m z_j^{x_j} \right) \\
&= \left(\sum_{\underline{i} \in \{0,1\}^m} a_{\underline{i}} \left(\prod_{j=1}^m x_j^{i_j} \right) \right) \left(\prod_{j=1}^m z_j^{x_j} \right) \\
&= \sum_{\underline{i} \in \{0,1\}^m} a_{\underline{i}} \sum_{\underline{x} \in \{0,1\}^m} \left(\prod_{j=1}^m x_j^{i_j} \right) \left(\prod_{j=1}^m z_j^{x_j} \right) \\
&= \sum_{\underline{i} \in \{0,1\}^m} a_{\underline{i}} \prod_{j=1}^m \sum_{x_j \in \{0,1\}} x_j^{i_j} z_j^{x_j} \\
&\stackrel{(a)}{=} \sum_{\underline{i} \in \{0,1\}^m} a_{\underline{i}} \prod_{j=1}^m \delta_{i_j, z_j} \\
&= a_{\underline{z}},
\end{aligned}$$

where (a) follows from $0^i z^0 + 1^i z = \delta_{i,z}$ with the convention that $0^0 = 1$. From this, it follows that

$$\dim(\mathcal{R}(r, m)) = \dim(\mathcal{F}(2, m, r)) = \sum_{i=0}^r \binom{m}{i}.$$

To see that $\mathcal{R}(m-1, m)$ equals the single parity-check code \mathcal{C}_0 of length $n = 2^m$, we first observe that it is spanned by the codewords associated with monomials of degree at most $m-1$. Lemma 3.3 shows that the codeword associated with a monomial of weight r has weight 2^{m-r} and we note that these weights are even for $r \in \{0, 1, \dots, m-1\}$. Thus, all codewords have even weight because all linear combinations of even weight vectors have even weight. This implies that $\mathcal{R}(m-1, m)$ is a subset of \mathcal{C}_0 . To see that it equals \mathcal{C}_0 , we note that $\dim(\mathcal{R}(m-1, m)) = n-1$ equals the dimension of \mathcal{C}_0 . \square

Lemma 3.6. $\mathcal{R}(r, m)^\perp = \mathcal{R}(m-r-1, m)$

Proof. For any codeword $\underline{a} \in \mathcal{R}(r, m)$, there is a polynomial $f \in \mathcal{F}(2, m, r)$ that generates \underline{a} via the evaluation map. Similarly, for any codeword $\underline{b} \in \mathcal{R}(m-r-1, m)$, there is a polynomial $g \in \mathcal{F}(2, m, m-r-1)$ that generates \underline{b} . Since $h = fg$ has degree at most $m-1$, it follows that $h \in \mathcal{F}(2, m, m-1)$ and applying the evaluation map to $h(\underline{x}) = f(\underline{x})g(\underline{x})$ generates

$$\underline{c} = (a_1 b_1, a_2 b_2, \dots, a_n b_n) \in \mathcal{R}(m-1, m).$$

Lemma 3.5 shows that $\mathcal{R}(m-1, m)$ is the single parity-check code and thus we find that

$$\sum_{i=1}^n a_i b_i = 0.$$

It follows that $\underline{b} \in \mathcal{R}(r, m)^\perp$ and hence $\mathcal{R}(m-r-1, m) \subseteq \mathcal{R}(r, m)^\perp$. To see that $\mathcal{R}(m-r-1, m)$ is not larger than $\mathcal{R}(r, m)^\perp$, we observe that they have the same dimension,

$$\dim(\mathcal{R}(m-r-1, m)) = \sum_{i=0}^{m-r-1} \binom{m}{i} = n - \sum_{i=0}^r \binom{m}{i} = \dim(\mathcal{R}(r, m)^\perp).$$

This completes the proof. \square

4 Plotkin's $(u, u + v)$ Construction

For $i \in \{1, 2\}$, let \mathcal{C}_i be an (n, k_i, d_i) linear code with generator matrix G_i and parity-check matrix H_i . Then, Plotkin's $(u, u + v)$ construction defines a new $(2n, k, d)$ linear code,

$$\mathcal{C} = \{(\underline{u}, \underline{u} + \underline{v}) \mid \underline{u} \in \mathcal{C}_1, \underline{v} \in \mathcal{C}_2\},$$

with generator and parity-check matrices given by

$$G = \begin{bmatrix} G_1 & G_1 \\ 0 & G_2 \end{bmatrix} \quad H = \begin{bmatrix} H_1 & 0 \\ -H_2 & H_2 \end{bmatrix}. \quad (2)$$

To see that G is a generator, we note that

$$[\underline{m}_1 \ \underline{m}_2]G = [\underline{m}_1 G_1, \underline{m}_1 G_1 + \underline{m}_2 G_2] = [\underline{u}, \underline{u} + \underline{v}], \quad \underline{u} \in \mathcal{C}_1, \underline{v} \in \mathcal{C}_2.$$

The existence of a generator also implies that \mathcal{C} is linear. Subtracting the 1st column from the 2nd column also shows that $k = \text{rank}(G) = \text{rank}(G_1) + \text{rank}(G_2) = k_1 + k_2$. One can verify that H is a parity-check matrix by computing GH^T and noting that $2n - k = \text{rank}(H) = \text{rank}(H_1) + \text{rank}(H_2) = 2n - k_1 - k_2$.

Next, we observe how this can be used recursively to construct Reed-Muller codes.

Lemma 4.1. *The Reed-Muller code $\mathcal{R}(r+1, m+1)$ can be constructed by applying the Plotkin $(u, u + v)$ construction to the Reed-Muller codes $\mathcal{R}(r+1, m)$ and $\mathcal{R}(r, m)$. More precisely, we have*

$$\mathcal{R}(r+1, m+1) = \{(\underline{u}, \underline{u} + \underline{v}) \mid \underline{u} \in \mathcal{R}(r+1, m), \underline{v} \in \mathcal{R}(r, m)\}.$$

Let $G(r, m)$ and $H(r, m)$ denote generator and parity-check matrices for $\mathcal{R}(r, m)$. Then, this recursive construction also implies that

$$G(r+1, m+1) = \begin{bmatrix} G(r+1, m) & G(r+1, m) \\ 0 & G(r, m) \end{bmatrix} \quad H(r+1, m+1) = \begin{bmatrix} H(r+1, m) & 0 \\ H(r, m) & H(r, m) \end{bmatrix}.$$

Moreover, this implies that

$$\text{rank}(G(r, m)) = \sum_{i=0}^r \binom{m}{i}.$$

Proof. Using Definition 3.1, we observe that each codeword $\underline{c} \in \mathcal{R}(r+1, m+1)$ is formed by evaluating a polynomial $f \in \mathcal{F}(2, m+1, r+1)$. For any such f , there are unique polynomials $g \in \mathcal{F}(2, m, r+1)$ and $h \in \mathcal{F}(2, m, r)$ such that

$$f(x_1, \dots, x_{m+1}) = g(x_1, \dots, x_m) + x_{m+1}h(x_1, \dots, x_m). \quad (3)$$

The polynomial g consists of all monomials in f that do not contain x_{m+1} while the polynomial h consists of $\frac{1}{x_{m+1}}$ times all monomials in f that do contain x_{m+1} . The mapping from f to the pair (g, h) is also bijective because any pair $(g, h) \in \mathcal{F}(2, m, r+1) \times \mathcal{F}(2, m, r)$ defines an $f \in \mathcal{F}(2, m+1, r+1)$ via (3).

Next, we observe that

$$\theta_{m+1}(\ell) = \begin{cases} (\theta_m(\ell), 0) & \text{if } \ell \in [2^m] \\ (\theta_m(\ell - 2^m), 1) & \text{if } \ell \in \{2^m + 1, \dots, 2^{m+1}\}. \end{cases}$$

Notice that $\underline{b} = \theta_{m+1}(\ell)$ satisfies $b_{m+1} = 0$ if and only if $\ell \in [2^m]$. From this, we see that

$$f(\theta_{m+1}(\ell)) = \begin{cases} g(\theta_m(\ell)) + 0 \cdot h(\theta_m(\ell)) & \text{if } \ell \in [2^m] \\ g(\theta_m(\ell - 2^m)) + 1 \cdot h(\theta_m(\ell - 2^m)) & \text{if } \ell \in \{2^m + 1, \dots, 2^{m+1}\}. \end{cases} \quad (4)$$

If we define \underline{u} via $u_\ell = g(\theta_m(\ell))$ and \underline{v} via $v_\ell = h(\theta_m(\ell))$, then $\underline{u} \in \mathcal{R}(r+1, m)$ and $\underline{v} \in \mathcal{R}(r, m)$. Using this and 4, we define \underline{c} via $c_\ell = f(\theta_{m+1}(\ell))$ and observe that $\underline{c} = (\underline{u}, \underline{u} + \underline{v})$.

To verify the rank formula, we start by noting that $\text{rank}(G(0, 1)) = 1$ and $\text{rank}(G(1, 1)) = 2$ gives the rank $(G(r, 1))$ formula for $r \in \{0, 1\}$. Proceeding by induction on m , we observe that the $(u, u + v)$ construction implies

$$\begin{aligned} \text{rank}(G(r + 1, m + 1)) &= \text{rank}(G(r + 1, m)) + \text{rank}(G(r, m)) \\ &= \binom{m}{0} + \sum_{i=0}^r \binom{m}{i+1} + \sum_{i=0}^r \binom{m}{i} \\ &= \binom{m}{0} + \sum_{i=0}^r \left[\binom{m}{i+1} + \binom{m}{i} \right] \\ &= \binom{m}{0} + \sum_{i=0}^r \binom{m+1}{i+1} \\ &= \sum_{i=0}^{r+1} \binom{m+1}{i} \end{aligned}$$

for $r \in \{0, 1, \dots, m-1\}$. The remaining cases of $\text{rank}(G(0, m+1)) = 1$ and $\text{rank}(G(m+1, m+1)) = m+1$ follow from basic arguments. This completes the proof. \square

This construction also provides a lower bound on the minimum distance of \mathcal{C} .

Lemma 4.2. *The minimum distance $d = \min\{2d_1, d_2\}$ of \mathcal{C} follows from*

$$d = \min_{\underline{u} \in \mathcal{C}_1, \underline{v} \in \mathcal{C}_2: (\underline{u}, \underline{v}) \neq \underline{0}} w_H((\underline{u}, \underline{u} + \underline{v})) = \min\{2d_1, d_2\}.$$

Proof. Let $(\underline{u}^*, \underline{v}^*)$ achieve the minimum and recall that $w_H(\underline{u} + \underline{v}) \geq w_H(\underline{v}) - w_H(\underline{u})$ with equality if $\underline{u} = \underline{0}$. If $\underline{v}^* = \underline{0}$, then it follows easily that $d = 2d_1$. If $\underline{v}^* \neq \underline{0}$, then

$$\begin{aligned} d &= \min_{\underline{u} \in \mathcal{C}_1, \underline{v} \in \mathcal{C}_2: \underline{v} \neq \underline{0}} w_H((\underline{u}, \underline{u} + \underline{v})) \\ &= \min_{\underline{u} \in \mathcal{C}_1, \underline{v} \in \mathcal{C}_2: \underline{v} \neq \underline{0}} [w_H(\underline{u}) + w_H(\underline{u} + \underline{v})] \\ &\stackrel{(a)}{\geq} \min_{\underline{u} \in \mathcal{C}_1, \underline{v} \in \mathcal{C}_2: \underline{v} \neq \underline{0}} [w_H(\underline{u}) + w_H(\underline{v}) - w_H(\underline{u})] \\ &= \min_{\underline{v} \in \mathcal{C}_2: \underline{v} \neq \underline{0}} w_H(\underline{v}) \\ &= d_2, \end{aligned}$$

where (a) holds with equality because the the minimum is achieved by $\underline{u} = \underline{0}$. Thus, $d = \min\{2d_1, d_2\}$. \square

Example 4.3. Let $\mathcal{C}_1 = \{00, 01, 10, 11\}$ and $\mathcal{C}_2 = \{00, 11\}$. Then,

$$\mathcal{C} = \{0000, 0101, 1010, 1111, 0011, 0110, 1001, 1100\}$$

has $k = k_1 + k_2 = 2 + 1 = 3$ and $d = \min(2, 2) = 2$.

Lemma 4.4. *The minimum distance of $\mathcal{R}(r, m)$ is given by $d(r, m) = 2^{m-r}$.*

Proof. First, it is easy to verify that $\mathcal{R}(0, m)$ is the repeat by 2^m code with $d(0, m) = 2^m$ and $\mathcal{R}(m, m)$ is the ‘‘uncode’’ with $d(m, m) = 1$. This completes the proof for $m = 1$. Next, we proceed by induction on m . Since

$$\mathcal{R}(r + 1, m + 1) = \{(\underline{u}, \underline{u} + \underline{v}) \mid \underline{u} \in \mathcal{R}(r + 1, m), \underline{v} \in \mathcal{R}(r, m)\}$$

by Lemma 4.1, applying Lemma 4.2 shows that $d(r + 1, m + 1) = \min\{2d(r + 1, m), d(r, m)\} = 2^{m-r}$ for $r \in \{0, 1, \dots, m-1\}$. The values of $d(r, m + 1)$ for $r = 0$ and $r = m + 1$ are treated separately using the first argument. This completes the induction. \square

References

- [1] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1977.
- [2] S. Kudekar, S. Kumar, M. Mondelli, H. D. Pfister, E. Şaşođlu, and R. Urbanke, “Reed-Muller codes achieve capacity on erasure channels,” *IEEE Trans. Inform. Theory*, vol. 63, no. 7, pp. 4298–4316, 2017.