Erasures repair for decreasing monomial-Cartesian and augmented Reed-Muller codes of high rate

Hiram H. López, Gretchen L. Matthews, *Senior Member, IEEE,* Daniel Valvo

***Abstract*—In this work, we present linear exact repair schemes for one or two erasures in decreasing monomial-Cartesian codes DM-CC, a family of codes which provides a framework for polar codes. In the case of two erasures, the positions of the erasures should satisfy a certain restriction. We present families of augmented Reed-Muller (ARM) and augmented Cartesian codes (ACar) which are families of evaluation codes obtained by strategically adding vectors to Reed-Muller and Cartesian codes, respectively. We develop repair schemes for one or two erasures for these families of augmented codes. Unlike the repair scheme for two erasures of DM-CC, the repair scheme for two erasures for the augmented codes has no restrictions on the positions of the erasures. When the dimension and base field are fixed, we give examples where ARM and ACar codes provide a lower bandwidth (resp., bitwidth) in comparison with Reed-Solomon (resp., Hermitian) codes. When the length and base field are fixed, we give examples where ACar codes provide a lower bandwidth in comparison with ARM. Finally, we analyze the asymptotic behavior when the augmented codes achieve the maximum rate.**

***Index Terms*—Reed-Muller codes, codes with high rate, Cartesian codes, monomial codes, monomial-Cartesian codes. 2010 Mathematics Subject Classification. Primary 11T71; Secondary 14G50.**

I. Introduction

The design of linear exact repair schemes for evaluation codes began with the foundational work of Guruswami and Wootters in which they developed a repair scheme (GW-scheme) to efficiently repair an erasure in a Reed-Solomon (RS) code [8]. This work served as motivation for linear exact repair schemes for algebraic geometry codes [11] and Reed-Muller codes [3]. In each of these instances, codes are considered over an extension field whose elements may be represented using subsymbols, meaning elements of a smaller base field. Erasure recovery is accomplished using subsymbols rather than the symbols themselves. Under certain conditions, these new schemes require less information than standard approaches to repair. In the distributed storage setting, this allows the information on a failed node to be recovered with the information stored on the remaining nodes. In particular, a codeword is stored so that each node stores a symbol and recovering a failed node exactly is equivalent to fixing an erasure in the codeword [5], [6].

An *evaluation code* [10] may be defined by sets of evaluation points and polynomials. Every codeword coordinate of an evaluation code depends of one of the evaluation points. *Monomial-Cartesian codes*(M-CC) [12] are evaluation codes that allow for more finely-tuned polynomial sets than RS and RM codes which employ polynomials of restricted degrees. *Decreasing monomial-Cartesian codes* (DM-CC) are a particular case of M-CC that satisfy the property that the polynomials sets are closed under divisibility. Recently, it was shown in [2] that families of polar codes with multiple kernels can be viewed as decreasing monomial-Cartesian codes and that any symmetric over the field channel polarizes utilizing multikernel polar codes constructed from decreasing monomial-Cartesian codes. This more general setting provides the opportunity to design high rate evaluation codes that admits a repair scheme, complementing the work done for Reed-Muller codes [3]. We will see these new codes compare favorably with existing families.

In particular, we introduce *augmented Reed-Muller* (ARM) codes and *augmented Cartesian* (ACar) codes via monomial-Cartesian codes. These augmented codes are evaluation codes obtained when certain vectors are added to a RM code and a Cartesian code, respectively. Thus the dimension is increased. We develop repair schemes for one or two erasures for these families of augmented codes. Unlike the repair scheme for two erasures of DM-CC, the repair scheme for two erasures for the augmented codes has no restrictions about the positions of the erasures. Because the GW-scheme repairs a RS code provided the code satisfies a restriction on the dimension, there are codes and parameters for which the GW-scheme does not apply. In this paper, we fill some of those gaps using ARM codes. When the dimension and base field are fixed, there are instances where ARM codes provide a lower bandwidth in comparison with RS codes and a lower bitwidth versus Hermitian codes. When the length and base field are fixed, we give examples where ACar codes provide a lower bandwidth in comparison with ARM. The repair scheme designed for ACar1 is strongly inspired by that in [3]. The challenge is adapting the framework to support repair when the dual code has fewer defining polynomials (than in the Reed-Muller case) while still being able to construct valid repair polynomials. We note that the repair polynomials utilized in [3] are products which lead to a necessarily higher degree which makes them applicable in

The work of Hiram H. López was supported in part by an AMS–Simons Travel Grant. The work of Gretchen L. Matthews was supported in part by NSF under Grant DMS-1855136 and in part by the Commonwealth Cyber Initiative. The portion of this work on augmented Reed-Muller codes was presented at ISIT 2021. *(Corresponding author: Hiram H. López.)*

H. H. López is with the Department of Mathematics and Statistics, Cleveland State University, Cleveland, OH, USA, e-mail: h.lopezvaldez@csuohio.edu.

G. L. Matthews and Daniel Valvo are with the Department of Mathematics, Virginia Tech, Blacksburg, VA, USA, e-mail addresses: gmatthews@vt.edu and vdaniel1@vt.edu.

a lower-rate setting. We circumvent this by defining the repair polynomials differently.

In Section II, we provide notation and definitions needed for the rest of the work. This section includes the necessary background on the families and the main properties of codes for which we develop repair schemes: decreasing monomial-Cartesian, augmented Reed-Muller, and augmented Cartesian codes which are introduced in Section III. In Sections IV and V, we develop repair schemes for one and two erasures, respectively, on the families DM-CC (with some restrictions on the positions of the erasures), ARM, and ACar. In Section VI, we explain some circumstances where a particular family may be preferable to others. Section VII concludes the paper with a summary of the main ideas and results of the work.

II. Preliminaries

Let q be a power of a prime p, $\mathbb{F}_q$ denote the finite field with q elements, and $K=\mathbb{F}_{q^t}$ be an extension field of $\mathbb{F}_q$ of degree $t=[K:\mathbb{F}_q]$. Given a linear code C of length n over K, elements of the field extension K are called *symbols* and the elements of the base field $\mathbb{F}_q$ are called *subsymbols*. As K is an $\mathbb{F}_q$-vector space, every coordinate for every vector $c \in C$ depends of t subsymbols. A *repair scheme* is an algorithm that recovers any component c_i of the vector $c\in C$ using other components. The *bandwidth* b is the number of subsymbols that the scheme needs to download to recover an erased entry c_i. The *normalized bandwidth*, $\left\lceil \frac{b}{t} \right\rceil$, can be interpreted as the number of symbols needed to recover the erased entry c_i. As a vector $c\in K^n$ is composed of nt subsymbols, the *bandwidth rate* $\dfrac{b}{nt}$ represents the fraction of the vector c that the repair scheme uses to recover the erased entry c_i. The *bitwidth* $b\log_2(q)$ represents the number of bits that the scheme needs to download to recover the erased entry c_i.

The *field trace* is defined as the polynomial $\mathrm{Tr}_{K/\mathbb{F}_q}(x)\in K[x]$ given by

$$\mathrm{Tr}_{K/\mathbb{F}_q}(x)=x^{q^{t-1}}+\cdots+x^{q^0}.$$

For the sake of convenience, we will often refer to $\mathrm{Tr}_{K/\mathbb{F}_q}(x)$ as simply $\mathrm{Tr}(x)$ when the extension being used is obvious from the context. Given $a\in K$, the field trace $\mathrm{Tr}(a)\in\mathbb{F}_q$. Additionally, $\mathrm{Tr}:K\to\mathbb{F}_q$ is an $\mathbb{F}_q$-linear map. More useful properties of the trace function are found in Remarks 2.1 and 2.2 below. They will be necessary for the repair schemes for decreasing and augmented codes.

Remark 2.1. *[17, Definition 2.30 and Theorem 2.40] Let $\mathcal{B}=\{z_1,\ldots,z_t\}$ be a basis of K over $\mathbb{F}_q$. Then there exists a basis $\{z_1',\ldots,z_t'\}$ of K over $\mathbb{F}_q$, called the dual basis of $\mathcal{B}$ such that $Tr(z_iz_j')=\delta_{ij}$ is an indicator function. For $\alpha\in K$,*

$$\alpha=\sum_{i=1}^{t} Tr(\alpha z_i)z_i'.$$

Thus, determining α is equivalent to finding $Tr(\alpha z_i)$, for $i\in\{1,\ldots,t\}$.

The next observation follows directly from the Rank-Nullity Theorem.

Remark 2.2. *Given $\alpha\in K\setminus\{0\}$, consider $Tr(\alpha x)$ as a function of x. Then $\ker(Tr(\alpha x))$ has dimension $t-1$ as an $\mathbb{F}_q$-vector space.*

Next, we review decreasing monomial-Cartesian codes, setting the foundation for the augmented codes. Let $R=K[x_1,\ldots,x_m]$ be the set of polynomials in m variables over K. For a lattice point $\boldsymbol{a}=(a_1,\ldots,a_m)\in\mathbb{Z}^m_{\geq 0}$, $\boldsymbol{x}^{\boldsymbol{a}}$ denotes the monomial $x_1^{a_1}\cdots x_m^{a_m}\in R$. For $\ell\in\mathbb{Z}_{\geq 0}$, $[\ell]:=\{1,\ldots,\ell\}$. Given a finite set $\mathcal{A}\subset\mathbb{Z}^m_{\geq 0}$, the subspace of polynomials of R that are K-linear combinations of monomials $\boldsymbol{x}^{\boldsymbol{a}}$, where $\boldsymbol{a}\in\mathcal{A}$, is

$$\mathcal{L}(\mathcal{A})=\mathrm{Span}_K\{\boldsymbol{x}^{\boldsymbol{a}}:\boldsymbol{a}\in\mathcal{A}\}\subseteq R.$$

Let $\mathcal{S}=S_1\times\cdots\times S_m\subseteq K^m$ be a Cartesian product, where every $S_i\subseteq K$ has $n_i:=|S_i|>0$, and $n:=|\mathcal{S}|$. We will assume that $n_1\leq\cdots\leq n_m$. Enumerate the elements of $\mathcal{S}$: $\boldsymbol{s}_1,\ldots,\boldsymbol{s}_n$. The *monomial-Cartesian code* associated with $\mathcal{S}$ and $\mathcal{A}$ is given by

$$\mathcal{C}(\mathcal{S},\mathcal{A})=\{\mathrm{ev}_{\mathcal{S}}(f):f\in\mathcal{L}(\mathcal{A})\}\subseteq K^n,$$

where $\mathrm{ev}_{\mathcal{S}}(f)=(f(\boldsymbol{s}_1),\ldots,f(\boldsymbol{s}_n))$. From now on, we assume that the degree of each monomial $M\in\mathcal{L}(\mathcal{A})$ in x_i is less than n_i. Then the length and rate of the monomial-Cartesian code $\mathcal{C}(\mathcal{S},\mathcal{A})$ are given by $|\mathcal{S}|$ and $\frac{|\mathcal{A}|}{|\mathcal{S}|}$, respectively [12, Proposition 2.1].

The dual of $\mathcal{C}(\mathcal{S},\mathcal{A})$, denoted by $\mathcal{C}(\mathcal{S},\mathcal{A})^\perp$, is the set of all $\boldsymbol{\alpha}\in K^n$ such that $\boldsymbol{\alpha}\cdot\boldsymbol{\beta}=0$ for all $\boldsymbol{\beta}\in\mathcal{C}(\mathcal{S},\mathcal{A})$, where $\boldsymbol{\alpha}\cdot\boldsymbol{\beta}$ is the ordinary inner product in K^n. The dual code $\mathcal{C}(\mathcal{S},\mathcal{A})^\perp$ was studied in [12] in terms of the vanishing ideal of $\mathcal{S}$ and in [15] in terms of the indicator functions of $\mathcal{S}$.

It is useful to focus on the case where the monomial set $\mathcal{L}(\mathcal{A})$ is *closed under divisibility*, meaning $\mathcal{L}(\mathcal{A})$ satisfies the property that if $M\in\mathcal{L}(\mathcal{A})$ and M' divides M, then $M'\in\mathcal{L}(\mathcal{A})$. In this case, the code $\mathcal{C}(\mathcal{S},\mathcal{A})$ is called a *decreasing monomial-Cartesian* code.

According to [2, Theorem 3.3], the dual of a decreasing monomial-Cartesian code is monomially equivalent to a decreasing monomial-Cartesian code. For completeness, we now give a proof. The key step is to observe that if the monomial set $\mathcal{L}(\mathcal{A})$ is closed under divisibility, then its complement in $\mathcal{S}$, the monomial set $\mathcal{L}\left(\mathcal{A}_{\mathcal{S}}^{\complement}\right)$, is also closed under divisibility, where $\mathcal{A}_{\mathcal{S}}^{\complement}=\left\{(n_1-1-a_1,\ldots,n_m-1-a_m)\in\mathbb{Z}^m_{\geq 0}:\boldsymbol{a}\notin\mathcal{A}\right\}$. Indeed, consider M a monomial in $\mathcal{L}(\mathcal{A}_{\mathcal{S}}^{\complement})$ and M' a monomial dividing M. Then, $M=\boldsymbol{x}^{\boldsymbol{\alpha}}$ and $M'=\boldsymbol{x}^{\boldsymbol{\alpha'}}$ for some $\boldsymbol{\alpha}=(n_1-1-a_1,\ldots,n_m-1-a_m)$ and $\boldsymbol{\alpha'}=(n_1-1-a_1',\ldots,n_m-1-a_m')$ where $\boldsymbol{a}=(a_1,\ldots,a_m)\notin\mathcal{A}$ and $\boldsymbol{a'}=(a_1',\ldots,a_m')$ is such that $a_1'\geq a_1,\ldots,a_m'\geq a_m$. Thus $\boldsymbol{x}^{\boldsymbol{a}}$ divides $\boldsymbol{x}^{\boldsymbol{a'}}$. Then, because $\mathcal{L}(\mathcal{A})$ is closed under divisibility, if $\boldsymbol{a'}\in\mathcal{A}$ that would imply $\boldsymbol{a}\in\mathcal{A}$ as well, a contradiction. Hence, $\boldsymbol{a'}\notin\mathcal{A}$, implying $\boldsymbol{\alpha'}\in\mathcal{A}_{\mathcal{S}}^{\complement}$. Therefore, $M'=\boldsymbol{x}^{\boldsymbol{\alpha'}}\in\mathcal{L}(\mathcal{A}_{\mathcal{S}}^{\complement})$, so $\mathcal{L}(\mathcal{A}_{\mathcal{S}}^{\complement})$ is closed under divisibility.

Now, given $\boldsymbol{s}\in\mathcal{S}$, define the nonzero element

$$\lambda_{\boldsymbol{s}}=\left(\prod_{i=1}^{m}\prod_{s_i'\in S_i\setminus\{s_i\}}(s_i-s_i')\right)^{-1}\in K.$$

For f in $K[x_1, \ldots, x_m]$, take the *residues vector*

$$\operatorname{Res}_{\mathcal{S}}(f) = (\lambda_{\boldsymbol{s}_1} f(\boldsymbol{s}_1), \ldots, \lambda_{\boldsymbol{s}_n} f(\boldsymbol{s}_n)) .$$

As $\mathcal{L}(\mathcal{A})$ is closed under divisibility, by [12, Theorem 2.7] we have that

$$\begin{aligned} C(\mathcal{S}, \mathcal{A})^{\perp} &= \operatorname{Span}_K \left\{ \operatorname{Res}_{\mathcal{S}}(\boldsymbol{x}^{\boldsymbol{a}}) : \boldsymbol{a} \in \mathcal{A}_{\mathcal{S}}^{\complement} \right\} \\ &= \{(\lambda_{\boldsymbol{s_1}} f(\boldsymbol{s_1}), \ldots, \lambda_{\boldsymbol{s_n}} f(\boldsymbol{s_n})) \ : \ f \in \mathcal{L}(\mathcal{A}_{\mathcal{S}}^{\complement})\}. \end{aligned}$$

Taking $\mathrm{D}_{\mathcal{S}} = \operatorname{diag}(\lambda_{\boldsymbol{s}_1}, \ldots, \lambda_{\boldsymbol{s}_n})$ to be the $n \times n$ diagonal matrix with $\lambda_{\boldsymbol{s}_i}$ in position (i, i) and 0 in any other position, the last equation implies that

$$C(\mathcal{S}, \mathcal{A})^{\perp} = \mathrm{D}_{\mathcal{S}}\, C(\mathcal{S}, \mathcal{A}_{\mathcal{S}}^{\complement}), \tag{1}$$

which means the dual of a decreasing monomial-Cartesian code is monomially equivalent to a decreasing monomial-Cartesian code.

A *Cartesian code*, introduced in [7] and then independently in [14], is defined by

$$\operatorname{Car}(\mathcal{S}, k) = \mathcal{C}(\mathcal{S}, \mathcal{A}_{Car}(k)),$$

where $\mathcal{A}_{Car}(k) = \{\boldsymbol{a} \in \mathbb{Z}_{\geq 0}^m : a_i \leq n_i - 1, \sum_{i=1}^m a_i \leq k\}$. By Equation (1), the dual of the Cartesian code $\operatorname{Car}(\mathcal{S}, k)$ is given by $\operatorname{Car}(\mathcal{S}, k)^{\perp} = \mathrm{D}_{\mathcal{S}} \operatorname{Car}(\mathcal{S}, k^{\perp})$, where $k^{\perp} = \sum_{i=1}^m (n_i - 1) - k - 1$.

Observe that if $\mathcal{S} = K^m$, the Cartesian code $\operatorname{Car}(\mathcal{S}, k)$ is the Reed-Muller code $\mathrm{RM}(K^m, k)$. The dual code $\mathrm{RM}(K^m, k)^{\perp}$ has been extensively studied in the literature. See for instance [3], [4], [9], where it is shown that the dual of a RM code is another RM code.

III. Augmented Codes

In this section, we define the augmented Cartesian codes for which we will provide repair schemes in the following section. Augmented Cartesian codes generalize the augmented Reed-Muller codes considered in [13]. Keeping the notation from the previous sections, we describe two families below.

A. Augmented Cartesian codes 1

An *augmented Cartesian code 1* (ACar1 code) over $K = \mathbb{F}_{q^t}$ is defined by

$$\mathrm{ACar1}(\mathcal{S}, \boldsymbol{k}) = \mathcal{C}(\mathcal{S}, \mathcal{A}_{Car1}(\boldsymbol{k})),$$

where $\boldsymbol{k} = (k_1, \ldots, k_m)$, with $0 \leq k_i \leq n_i - q^{t-1}$, and

$$\mathcal{A}_{Car1}(\boldsymbol{k}) = \prod_{i=1}^{m} \{0, \ldots, n_i - 1\} \setminus \prod_{i=1}^{m} \{k_i, \ldots, n_i - 1\} .$$

An augmented Cartesian code 1 is shown in Example 3.1. The condition that $0 \leq k_i \leq n_i - q^{t-1}$ for all $i \in \{1, m\}$ will be utilized later to guarantee that certain polynomials are in the appropriate dual codes.

Example 3.1. Take $K = \mathbb{F}_{17}$. Let $S_1, S_2 \subseteq K$ with $n_1 = |S_1| = 6$ and $n_2 = |S_2| = 7$. The code $\mathrm{ACar1}(S_1 \times S_2, (2, 2))$ is generated by the vectors $\mathrm{ev}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in Figure 1 (a). The dual code $\mathrm{ACar1}(S_1 \times S_2, (2, 2))^{\perp}$ is generated by the vectors $\operatorname{Res}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in Figure 1 (b).

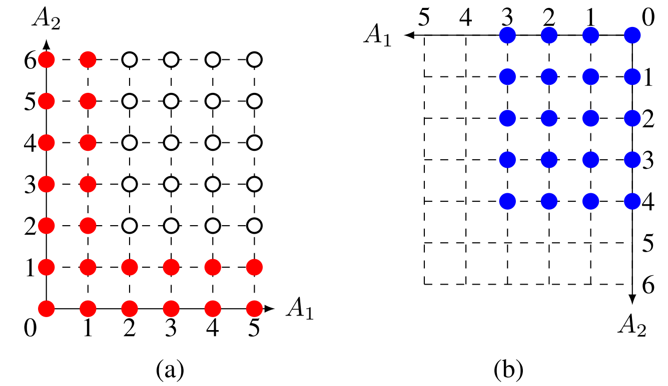


Fig. 1. The code $\mathrm{ACar1}(S_1 \times S_2, (2, 2))$ in Example 3.1 with $K = \mathbb{F}_{17}$ is generated by vectors of the form $\mathrm{ev}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in (a). Points corresponding to monomials with exponents in $\prod_{i=1}^m \{k_i, \ldots, n_i - 1\}$ are indicated by $\circ$. The dual code $\mathrm{ACar1}(S_1 \times S_2, (2, 2))^{\perp}$ is generated by the vectors of the form $\operatorname{Res}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in (b).

When $k_i = k \leq q^t - q^{t-1}$ for all $i \in [m]$ and $\mathcal{S} = K^m$, the augmented Cartesian code 1 $\mathrm{ACar1}(\mathcal{S}, \boldsymbol{k})$ is called an *augmented Reed-Muller code 1*, which is denoted by $\mathrm{ARM1}(K^m, k)$. An augmented Reed-Muller code 1 is shown in Example 3.2.

Example 3.2. Take $K = \mathbb{F}_7$. The code $\mathrm{ARM1}(K^2, 2)$ is generated by the vectors $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent is a point in Figure 2 (a). The dual $\mathrm{ARM1}(K^2, 2)^{\perp}$ is generated by the vectors $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent is a point in Figure 2 (b).

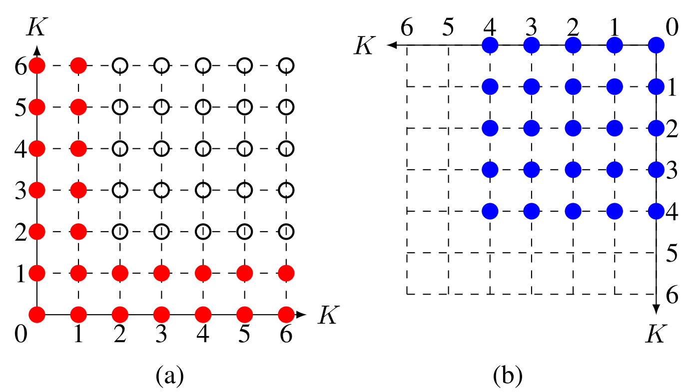


Fig. 2. The $\mathrm{ARM1}(K^2, 2)$ code in Example 3.2 with $K = \mathbb{F}_7$ is generated by the vectors of the form $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent corresponds to a point in (a). Points corresponding to monomials with exponents in $\prod_{i=1}^m \{k_i, \ldots, n_i - 1\}$ are indicated by $\circ$. $\mathrm{ARM1}(K^2, 2)^{\perp}$ is generated by the vectors of the form $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent corresponds to a point in (b).

In Figure 2, the monomials that define $\mathrm{RM}(K^2, 2)$ may be seen as those under the diagonal in $\mathrm{ARM1}(K^2, 2)$. The monomial diagram for any Reed-Muller code will restrict the allowable monomials under some diagonal excluding many monomials along or near the edges, resulting in codes with lower dimensions and rates. This explains why ARM1 codes have higher rates than their associated Reed-Muller codes.

The next result is relevant for developing the repair scheme for $\mathrm{ACar1}(\mathcal{S}, \boldsymbol{k})$.

Proposition 3.3. *The following holds for the augmented Cartesian code 1.*

(a) *The dimension is*

$$\dim ACar1(\mathcal{S}, \boldsymbol{k}) = \prod_{j=1}^{m} n_j - \prod_{j=1}^{m} (n_j - k_j).$$

(b) *The dual is* $ACar1(\mathcal{S}, \boldsymbol{k})^{\perp} = \mathrm{D}_{\mathcal{S}}\, \mathcal{C}(\mathcal{S}, \mathcal{A}^{\perp}_{Car1}(\boldsymbol{k}))$, *where* $\mathcal{A}^{\perp}_{Car1}(\boldsymbol{k}) = \prod_{i=1}^{m} \{0, \ldots, n_i - k_i - 1\}$.

Proof. (a) The statement follows immediately, because

$$\begin{aligned}|\mathcal{A}_{Car1}(\boldsymbol{k})| &= |\prod_{j=1}^{m}\{0, \ldots, n_j - 1\} \setminus \prod_{j=1}^{m}\{k_j, \ldots, n_j - 1\}| \\ &= \prod_{j=1}^{m} n_j - \prod_{j=1}^{m}(n_j - k_j).\end{aligned}$$

(b) Observe that $\mathcal{A}_{Car1}(\boldsymbol{k})^{\complement}_{\mathcal{S}} = \mathcal{A}^{\perp}_{Car1}(\boldsymbol{k})$. Indeed,

$$(n_1 - 1 - a_1, \ldots, n_m - 1 - a_m) \in \mathcal{A}_{Car1}(\boldsymbol{k})^{\complement}_{\mathcal{S}}$$

if and only if

$$(a_1, \ldots, a_m) \in \prod_{i=1}^{m} \{0, \ldots, n_i - 1\} \setminus \mathcal{A}_{Car1}(\boldsymbol{k}),$$

which happens if and only if

$$(n_1 - 1, \ldots, n_m - 1) - (a_1, \ldots, a_m) \in \mathcal{A}^{\perp}_{Car1}(\boldsymbol{k}).$$

Thus, the result follows from Equation (1). $\square$

B. Augmented Cartesian codes 2

We next define a second family of high-rate Cartesian codes. The *augmented Cartesian code 2* (ACar2 code) is defined by

$$\mathrm{ACar2}(\mathcal{S}, \boldsymbol{k}) = \mathcal{C}(\mathcal{S}, \mathcal{A}_{Car2}(\boldsymbol{k})),$$

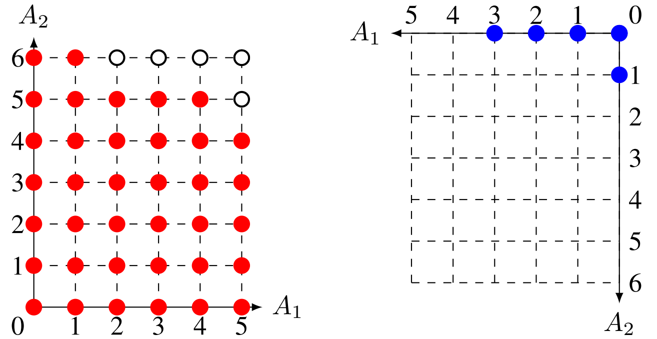
where $\boldsymbol{k} = (k_1, \ldots, k_m)$, with $0 \le k_i \le n_i - q^{t-1}$, and $\mathcal{A}_{Car2}(\boldsymbol{k}) = \prod_{j=1}^{m} \{0, \ldots, n_j - 1\} \setminus \bigcup_{j=1}^{m} L_j$, with

$$L_j = \{\boldsymbol{a} : k_j \le a_j \le n_j - 1 \ ,\ a_i = n_i - 1 \text{ for all } i \neq j\}.$$

An augmented Cartesian code 2 is shown in Example 3.4.

Example 3.4. Take $K = \mathbb{F}_{17}$. Let S_1 and S_2 be subsets of K with $n_1 = |S_1| = 6$ and $n_2 = |S_2| = 7$. The code $\mathrm{ACar2}(S_1 \times S_2, (2, 5))$ is generated by the vectors $\mathrm{ev}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in Figure 3 (a). The dual code $\mathrm{ACar2}(S_1 \times S_2, (2, 5))^{\perp}$ is generated by the vectors $\mathrm{Res}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in Figure 3 (b).

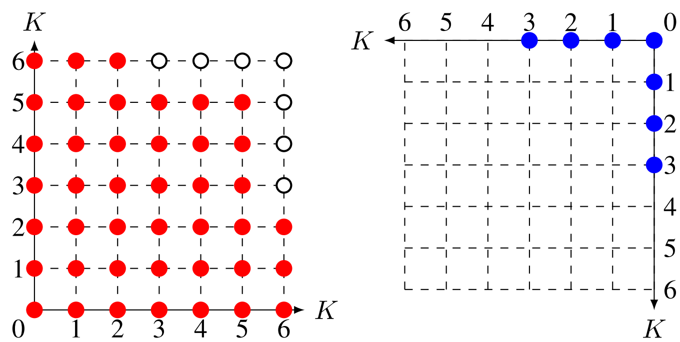
When $k_i = k \le q^t - q^{t-1}$ for all $i \in [m]$ and $\mathcal{S} = K^m$, the augmented Cartesian code 2 $\mathrm{ACar2}(\mathcal{S}, \boldsymbol{k})$ is called an *augmented Reed-Muller code* 2, which is denoted by $\mathrm{ARM2}(K^m, k)$. An augmented Reed-Muller code 2 is shown in Example 3.5.



(a) (b)

Fig. 3. The code $\mathrm{ACar2}(S_1 \times S_2, (2, 5))$ in Example 3.4 with $K = \mathbb{F}_{17}$ is generated by the vectors of the form $\mathrm{ev}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in (a). Elements of $L_1 \cup L_2$ are indicated by $\circ$. The dual $\mathrm{ACar2}(S_1 \times S_2, (2, 5))^{\perp}$ is generated by the vectors of the form $\mathrm{Res}_{S_1 \times S_2}(M)$, where M is a monomial whose exponent is a point in (b).

Example 3.5. Take $K = \mathbb{F}_7$. The code $\mathrm{ARM2}(K^2, 3)$ is generated by the vectors $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent is a point in Figure 4 (a). The dual $\mathrm{ARM2}(K^2, 3)^{\perp}$ is generated by the vectors $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent is a point in Figure 4 (b).



(a) (b)

Fig. 4. The code $\mathrm{ARM2}(K^2, 3)$ in Example 3.5 with $K = \mathbb{F}_7$ is generated by the vectors of the form $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent corresponds to a point in (a). Elements of $L_1 \cup L_2$ are indicated by $\circ$. The dual $\mathrm{ARM2}(K^2, 3)^{\perp}$ is generated by the vectors of the form $\mathrm{ev}_{K^2}(M)$, where M is a monomial whose exponent corresponds to a point in (b).

The next result is relevant for developing the repair scheme for $\mathrm{ACar2}(\mathcal{S}, \boldsymbol{k})$.

Proposition 3.6. *The following holds for the augmented Cartesian code 2.*

(a) *The dimension is*

$$\dim ACar2(\mathcal{S}, \boldsymbol{k}) = \prod_{i=1}^{m} n_i - \sum_{i=1}^{m} (n_i - k_i - 1) - 1.$$

(b) *The dual is* $ACar2(\mathcal{S}, \boldsymbol{k})^{\perp} = \mathrm{D}_{\mathcal{S}}\, \mathcal{C}(\mathcal{S}, \mathcal{A}^{\perp}_{Car2}(\boldsymbol{k}))$, *where* $\mathcal{A}^{\perp}_{Car2}(\boldsymbol{k}) = \bigcup_{j=1}^{m} L'_j$ *and*

$$L'_j = \{\boldsymbol{a} \ :\ 0 \le a_j \le n_j - k_j - 1 \ ,\ a_i = 0 \text{ for all } i \neq j\}.$$

Proof. (a) We have that

$$\begin{aligned}|\mathcal{A}_{Car2}(\boldsymbol{k})| &= |\prod_{i=1}^{m}\{0,\dots,n_i-1\}\setminus\bigcup_{i=1}^{m}L_i| \\ &= \prod_{i=1}^{m}n_i - |\bigcup_{i=1}^{m}L_i|.\end{aligned}$$

As $\bigcap_{i=1}^{m} L_i = \{\boldsymbol{a}\}$, where $\boldsymbol{a} = (n_1-1,\dots,n_m-1)$, and $(L_i\setminus\{\boldsymbol{a}\})\bigcap(L_j\setminus\{\boldsymbol{a}\}) = \emptyset$ for all $i\neq j$, then $|\bigcup_{i=1}^{m}L_i| = \sum_{i=1}^{m}|L_i\setminus\{\boldsymbol{a}\}|+1 = \sum_{i=1}^{m}(n_i-k_i-1)+1$. Thus $|\mathcal{A}_{Car2}(\boldsymbol{k})| = \prod_{i=1}^{m}n_i - \sum_{i=1}^{m}(n_i-k_i-1)-1$.

(b) In a similar way to the proof of Proposition 3.3, it is not difficult to check that $\mathcal{A}_{Car2}(\boldsymbol{k})^{\complement}_{K^m} = \mathcal{A}^{\perp}_{Car2}(\boldsymbol{k})$. Thus, the result follows from Equation (1). □

IV. Single Erasure Repair Schemes

In this section, we develop a repair scheme that repairs a single erasure of a decreasing monomial-Cartesian code $\mathcal{C}(\mathcal{S},\mathcal{A})$ that satisfies the property that $\mathcal{A}\cap L_j=\emptyset$ for some j, where $L_j = \{\boldsymbol{a} : n_j - q^{t-1} \le a_j < n_j, a_i = n_i-1 \text{ for } i\neq j\}$. As a consequence, we obtain repair schemes for single erasures of augmented Cartesian and Reed-Muller codes.

Theorem 4.1. *Let $\mathcal{C}(\mathcal{S},\mathcal{A})$ be a decreasing monomial-Cartesian code of length n such that there is $j\in[n]$ with $\mathcal{A}\cap L_j=\emptyset$. Then there exists a repair scheme for one erasure with bandwidth at most*

$$b = n-1+(t-1)\left(\frac{n}{n_j}-1\right).$$

Proof. Let $\boldsymbol{s}^* = (s_1^*,\dots,s_m^*)\in\mathcal{S}$ and assume that the entry $f(\boldsymbol{s}^*)$ of the codeword $(f(\boldsymbol{s}_1),\dots,f(\boldsymbol{s}_n))\in\mathcal{C}(\mathcal{S},\mathcal{A})$ has been erased. Let $\{z_1,\dots,z_t\}$ be a basis for K over $\mathbb{F}_q$. For $i\in[t]$, define the following polynomials

$$\begin{aligned}p_i(\boldsymbol{x}) &= \frac{\operatorname{Tr}(z_i(x_j-s_j^*))}{(x_j-s_j^*)} \\ &= z_i + z_i^q(x_j-s_j^*)^{q-1}+\cdots+z_i^{q^{t-1}}(x_j-s_j^*)^{q^{t-1}-1}.\end{aligned}$$

As $\mathcal{A}\cap L_j=\emptyset$, $(L_j)^{\complement}_{\mathcal{S}} = \{(0,\dots,0,a) : 0\le a<q^{t-1}\}\subseteq \mathcal{A}^{\complement}_{\mathcal{S}}$. Thus, for $i\in[t]$, every polynomial $p_i(\boldsymbol{x})\in\mathcal{L}((L_j)^{\complement}_{\mathcal{S}})\subseteq\mathcal{L}(\mathcal{A}^{\complement}_{\mathcal{S}})$ defines an element in $\mathcal{C}(\mathcal{S},\mathcal{A})^{\perp} = \mathrm{D}_{\mathcal{S}}\,C(\mathcal{S},\mathcal{A}^{\complement}_{\mathcal{S}})$. Therefore, we obtain the t equations

$$\lambda_{\boldsymbol{s}^*}p_i(\boldsymbol{s}^*)f(\boldsymbol{s}^*) = -\sum_{\mathcal{S}\setminus\{\boldsymbol{s}^*\}}\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s}),\quad i\in[t]. \tag{2}$$

As $p_i(\boldsymbol{s}^*)=z_i$, applying the trace function to both sides of previous equations and employing the linearity of the trace function, we obtain

$$\operatorname{Tr}(z_i\lambda_{\boldsymbol{s}^*}f(\boldsymbol{s}^*)) = -\sum_{\mathcal{S}\setminus\{\boldsymbol{s}^*\}}\operatorname{Tr}(\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s})),\quad i\in[t].$$

Define the set $\Gamma = \{(s_1,\dots,s_m)\in\mathcal{S} : s_j = s_j^*\}$. For $\boldsymbol{s}\in\Gamma$, we have that $p_i(\boldsymbol{s}) = z_i$. For $\boldsymbol{s}\in\mathcal{S}\setminus\Gamma$, $p_i(\boldsymbol{s}) = \frac{\operatorname{Tr}(z_i(s_j-s_j^*))}{(s_j-s_j^*)}$. Therefore, we obtain that for $i\in[t]$,

$$\begin{aligned}&\sum_{\mathcal{S}\setminus\{\boldsymbol{s}^*\}}\operatorname{Tr}(\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s})) \\ &= \sum_{\Gamma\setminus\{\boldsymbol{s}^*\}}\operatorname{Tr}(\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s})) + \sum_{\mathcal{S}\setminus\Gamma}\operatorname{Tr}(\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s})) \\ &= \sum_{\Gamma\setminus\{\boldsymbol{s}^*\}}\operatorname{Tr}(\lambda_{\boldsymbol{s}}z_if(\boldsymbol{s})) + \sum_{\mathcal{S}\setminus\Gamma}\operatorname{Tr}\left(\lambda_{\boldsymbol{s}}\frac{\operatorname{Tr}(z_i(s_j-s_j^*))}{(s_j-s_j^*)}f(\boldsymbol{s})\right) \\ &= \sum_{\Gamma\setminus\{\boldsymbol{s}^*\}}\operatorname{Tr}(\lambda_{\boldsymbol{s}}z_if(\boldsymbol{s})) + \sum_{\mathcal{S}\setminus\Gamma}\operatorname{Tr}(z_i(s_j-s_j^*))\operatorname{Tr}\left(\frac{\lambda_{\boldsymbol{s}}f(\boldsymbol{s})}{(s_j-s_j^*)}\right).\end{aligned}$$

By Remark 2.1, $\lambda_{\boldsymbol{s}^*}f(\boldsymbol{s}^*)$, and as a consequence, $f(\boldsymbol{s}^*)$, can be recovered from its t independent traces $\operatorname{Tr}(z_i\lambda_{\boldsymbol{s}^*}f(\boldsymbol{s}^*))$, which can be obtained by downloading:

- t subsymbols $\operatorname{Tr}(\lambda_{\boldsymbol{s}}z_1f(\boldsymbol{s})),\dots,\operatorname{Tr}(\lambda_{\boldsymbol{s}}z_tf(\boldsymbol{s}))$, for each $\boldsymbol{s}\in\Gamma\setminus\{\boldsymbol{s}^*\}$, and
- one subsymbol $\operatorname{Tr}\left(\frac{\lambda_{\boldsymbol{s}}f(\boldsymbol{s})}{(s_j-s_j^*)}\right)$, for each $\boldsymbol{s}\in\mathcal{S}\setminus\Gamma$.

Hence, the bandwidth is $b = t(|\Gamma|-1)+|\mathcal{S}\setminus\Gamma| = (t-1)\left(\frac{n}{n_j}-1\right)+n-1$. □

Corollary 4.2. *There exist repair schemes for one erasure of ACar1$(\mathcal{S},\boldsymbol{k})$ and ACar2$(\mathcal{S},\boldsymbol{k})$, each with bandwidth at most*

$$b = \prod_{i=1}^{m}n_i - 1 + (t-1)\left(\prod_{i=1}^{m-1}n_i-1\right).$$

Proof. Since $k_i\le n_i-q^{t-1}$ for $i\in[m]$, $\mathcal{A}_{Car1}(\boldsymbol{k})\cap L_m=\emptyset$ and $\mathcal{A}_{Car2}(\boldsymbol{k})\cap L_m=\emptyset$, where $L_m = \{(n_1-1,\dots,n_{m-1}-1,a) : n_m-q^{t-1}\le a<n_m\}$. Thus, the result follows from Theorem 4.1. □

As another consequence from Theorem 4.1, by taking $\mathcal{S}=K^m$, we obtain a repair scheme for augmented Reed-Muller codes, whose family was first introduced in [13, Theorem 2.5].

Corollary 4.3. *There exists a repair scheme for one erasure for ARM1(K^m,k) and for ARM2(K^m,k), each with bandwidth*

$$b = |K|^m - 1 + (t-1)(|K|^{m-1}-1).$$

Remark 4.4. The bandwidth of the repair scheme developed in Corollary 4.3 for augmented Reed-Muller codes is less than the one developed in [13, Theorems 2.5 and 3.4]. This is due to the fact that the repair polynomials used in the proofs of [13, Theorems 2.5 and 3.4] have more zeros over $\mathcal{S}$ than the repair polynomials of the proof of Corollary 4.3. Thus, the number of subsymbols that are needed to repair an erasure is less when we use Corollary 4.3.

V. Two Erasures Repair Schemes

In this section, we keep the same notation as in previous sections and develop a repair scheme that repairs two simultaneous erasures $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ of $\mathcal{C}(\mathcal{S},\mathcal{A})$ provided the erasure positions satisfy the property that $s^*_j \neq s'_j$. Then we give a repair scheme that repairs two simultaneous erasures of the augmented Cartesian and Reed-Muller codes that does not require that the position vectors $\boldsymbol{s'}$ and $\boldsymbol{s}^*$ are different on a specific component.

Theorem 5.1. *Let $\mathcal{C}(\mathcal{S},\mathcal{A})$ be a decreasing monomial-Cartesian code of length n such that there exists $j \in [n]$ with $\mathcal{A} \cap L_j = \emptyset$. Let $\boldsymbol{s}^* = (s^*_1,\ldots,s^*_m), \boldsymbol{s'} = (s'_1,\ldots,s'_m) \in \mathcal{S}$ such that $s^*_j \neq s'_j$. There exists a repair scheme for the two simultaneous erasures $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ with bandwidth at most*

$$b = 2\left[n-2+(t-1)\left(\frac{n}{n_j}-2\right)\right].$$

Proof. Assume that the entries $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ of the codeword $(f(\boldsymbol{s}_1),\ldots,f(\boldsymbol{s}_n)) \in \mathcal{C}(\mathcal{S},\mathcal{A})$ have been erased. By Remark 2.2, $\Delta_j := \left\{\alpha \in K : \mathrm{Tr}\left(\alpha(s'_j - s^*_j\right) = 0\right\}$ has dimension $t-1$ as $\mathbb{F}_q$-vector space. Let $\{z_1,\ldots,z_{t-1}\}$ be an $\mathbb{F}_q$-basis for Δ_j and z_t an element in K such that $\{z_1,\ldots,z_{t-1},z_t\}$ is an $\mathbb{F}_q$-basis for K. Finally, let τ be an element of $\ker(\tau)$. We are ready to define the repair polynomials. Take

$$p_i(\boldsymbol{x}) = \tau\frac{\mathrm{Tr}\left(z_i(x_j - s^*_j)\right)}{\left(x_j - s^*_j\right)}$$

and

$$q_i(\boldsymbol{x}) = \frac{\mathrm{Tr}\left(z_i(x_j - s'_j)\right)}{(x_j - s'_j)}, \quad i \in [t].$$

As $\mathcal{A} \cap L_j = \emptyset$, the polynomials $p_i(\boldsymbol{x})$ and $q_i(\boldsymbol{x})$ define elements in the dual code $\mathcal{C}(\mathcal{S},\mathcal{A})^{\perp}$. Therefore, in a similar way to the proof of Theorem 4.1, we obtain the following $2t$ equations for all $i \in [t]$:

$$\begin{aligned}&\lambda_{\boldsymbol{s}^*}p_i(\boldsymbol{s}^*)f(\boldsymbol{s}^*) + \lambda_{\boldsymbol{s'}}p_i(\boldsymbol{s'})f(\boldsymbol{s'})\\ &= -\sum_{\boldsymbol{s}\in\mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}}\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s}),\end{aligned} \tag{3}$$

and

$$\begin{aligned}&\lambda_{\boldsymbol{s}^*}q_i(\boldsymbol{s}^*)f(\boldsymbol{s}^*) + \lambda_{\boldsymbol{s'}}q_i(\boldsymbol{s'})f(\boldsymbol{s'})\\ &= -\sum_{\boldsymbol{s}\in\mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}}\lambda_{\boldsymbol{s}}q_i(\boldsymbol{s})f(\boldsymbol{s}).\end{aligned} \tag{4}$$

By definition of the p_i's and q_i's, $p_i(\boldsymbol{s}^*) = \tau z_i$ and $q_i(\boldsymbol{s'}) = z_i$ for $i \in [t]$. As $\{z_1,\ldots,z_{t-1}\}$ is an $\mathbb{F}_q$-basis for Δ_j, $p_i(\boldsymbol{s'}) = q_i(\boldsymbol{s}^*) = 0$ for $i \in [t-1]$, thus Equations 3 and 4 become

$$\lambda_{\boldsymbol{s}^*}\tau z_i f(\boldsymbol{s}^*) = -\sum_{\boldsymbol{s}\in\mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}}\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s}), \quad i \in [t-1], \tag{5}$$

$$\lambda_{\boldsymbol{s}^*}\tau z_t f(\boldsymbol{s}^*) + \lambda_{\boldsymbol{s'}}\,p_t(\boldsymbol{s'})f(\boldsymbol{s'}) = -\sum_{\boldsymbol{s}\in\mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}}\lambda_{\boldsymbol{s}}p_t(\boldsymbol{s})f(\boldsymbol{s}), \tag{6}$$

$$\lambda_{\boldsymbol{s'}}\,z_i f(\boldsymbol{s'}) = -\sum_{\boldsymbol{s}\in\mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}}\lambda_{\boldsymbol{s}}q_i(\boldsymbol{s})f(\boldsymbol{s}), \quad i \in [t-1], \tag{7}$$

$$\lambda_{\boldsymbol{s}^*}q_t(\boldsymbol{s}^*)f(\boldsymbol{s}^*) + \lambda_{\boldsymbol{s'}}\,z_t f(\boldsymbol{s'}) = -\sum_{\boldsymbol{s}\in\mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}}\lambda_{\boldsymbol{s}}q_t(\boldsymbol{s})f(\boldsymbol{s}). \tag{8}$$

Observe that

$$\begin{aligned}&\mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\,p_t(\boldsymbol{s'})f(\boldsymbol{s'})\right)\\ &= \mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\,\tau\frac{\mathrm{Tr}\left(z_t(s'_j - s^*_j)\right)}{(s'_j - s^*_j)}f(\boldsymbol{s'})\right)\\ &= \mathrm{Tr}\left(z_t(s'_j - s^*_j)\right)\mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\,\frac{\tau}{(s'_j - s^*_j)}f(\boldsymbol{s'})\right).\end{aligned}$$

As $\dfrac{\tau}{(s'_j - s^*_j)} \in \Delta_j$, whose $\mathbb{F}_q$-basis is $\{z_1,\ldots,z_{t-1}\}$, there exist $\alpha_1,\ldots,\alpha_{t-1} \in \mathbb{F}_q$ such that previous equations imply that

$$\mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\,p_t(\boldsymbol{s'})f(\boldsymbol{s'})\right) = \mathrm{Tr}\left(z_t(s'_j - s^*_j)\right)\sum_{i=1}^{t-1}\alpha_i\mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\,z_i f(\boldsymbol{s'})\right).$$

By Remark 2.1, the element $f(\boldsymbol{s}^*)$ can be recovered from the t traces $\mathrm{Tr}(\lambda_{\boldsymbol{s}^*}\tau z_i f(\boldsymbol{s}^*))$. Thus, from last equation, and applying the trace function to both sides of Equations 5, 6 and 7, we get that the traces $\mathrm{Tr}(\lambda_{\boldsymbol{s}^*}\tau z_i f(\boldsymbol{s}^*))$, for $i \in [t]$, can be obtained by downloading for every $\boldsymbol{s} \in \mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}$, the elements $\mathrm{Tr}(\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s}))$ for $i \in [t]$, and $\mathrm{Tr}(\lambda_{\boldsymbol{s}}q_i(\boldsymbol{s})f(\boldsymbol{s}))$ for $i \in [t-1]$. Finally, as $f(\boldsymbol{s}^*)$ has been already recovered, from Equation 8, we can obtain $\mathrm{Tr}(\lambda_{\boldsymbol{s'}}\,z_t f(\boldsymbol{s'}))$, and as a consequence $f(\boldsymbol{s'})$, by downloading for every $\boldsymbol{s} \in \mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}$, the elements $\mathrm{Tr}(\lambda_{\boldsymbol{s}}q_t(\boldsymbol{s})f(\boldsymbol{s}))$.

Therefore, both erasures $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ can be recovered by downloading for every $\boldsymbol{s} \in \mathcal{S}\setminus\{\boldsymbol{s}^*,\boldsymbol{s'}\}$, the elements $\mathrm{Tr}(\lambda_{\boldsymbol{s}}p_i(\boldsymbol{s})f(\boldsymbol{s}))$ and $\mathrm{Tr}(\lambda_{\boldsymbol{s}}q_i(\boldsymbol{s})f(\boldsymbol{s}))$ for $i \in [t]$. The bandwidth is a consequence of the proof of Theorem 4.1, considering that now we need to download twice the information about $n-2$ elements, instead of only $n-1$ as in Theorem 4.1. □

Theorem 5.2. *There exists a repair scheme for ACar1$(\mathcal{S},\boldsymbol{k})$ that repairs two simultaneous erasures $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ with bandwidth at most*

$$b = 2\left[\prod_{i=1}^{m}n_i - 1 + (t-1)\left(\prod_{i=1}^{m-1}n_i - 1\right)\right].$$

Proof. As $\boldsymbol{s}^* \neq \boldsymbol{s'}$, there is $j \in [m]$ such that $s^*_j \neq s'_j$. The condition $k_i \leq n_i - q^{t-1}$ on the definition of augmented Cartesian code 1 implies that $\mathcal{A}_{Car1}(\boldsymbol{k}) \cap L_j = \emptyset$, where $L_j = \{(a_1,\ldots,a_m) : n_j - q^{t-1} \leq a_j < n_j, a_i = n_{i-1} - 1 \text{ for } i \neq j\}$. Thus, the result follows from the proof of Theorem 5.1 and the fact that the length of the augmented Cartesian code 1, $n = \prod_{i=1}^{m}n_i$, is given by the cardinality of the Cartesian set $\mathcal{S}$. □

Theorem 5.3. *There exists a repair scheme for ACar2$(\mathcal{S},\boldsymbol{k})$ that repairs two simultaneous erasures $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ with bandwidth at most*

$$b = 2\left[\prod_{i=1}^{m}n_i - 1 + (t-1)\left(\prod_{i=1}^{m-1}n_i - 1\right)\right].$$

Proof. As $\boldsymbol{s}^* \neq \boldsymbol{s'}$, there is $j \in [m]$ such that $s^*_j \neq s'_j$. By Remark 2.2, $\ker(\mathrm{Tr})$ has dimension $t-1$ as $\mathbb{F}_q$-vector space. Let $\{z_1, \ldots, z_{t-1}\}$ be an $\mathbb{F}_q$-basis for $\ker(\mathrm{Tr})$ and z_t an element in K such that $\{z_1, \ldots, z_{t-1}, z_t\}$ is an $\mathbb{F}_q$-basis for K. Then we define the repair polynomials

$$p_i(\boldsymbol{x}) = z_1 \frac{\mathrm{Tr}\left(\dfrac{z_i(x_j - s_j^*)}{s'_j - s_j^*}\right)}{\dfrac{x_j - s_j^*}{s'_j - s_j^*}}$$

and

$$q_i(\boldsymbol{x}) = \frac{\mathrm{Tr}\left(\dfrac{z_i(x_j - s'_j)}{s_j^* - s'_j}\right)}{\dfrac{x_j - s'_j}{s_j^* - s'_j}}, \quad i \in [t].$$

By definition of augmented Cartesian code 2, $k_i \leq n_i - q^{t-1}$, for $i \in [m]$, thus the polynomials $p_i(\boldsymbol{x})$ and $q_i(\boldsymbol{x})$ define elements in the dual code $\mathrm{ACar2}(\mathcal{S}, \boldsymbol{k})^\perp$. Observe that the polynomials p_i's and q_i's have the property that $p_i(\boldsymbol{s}^*) = z_1 z_i$ and $q_i(\boldsymbol{s'}) = z_i$ for $i \in [t]$. By definition of the z_i's, $p_i(\boldsymbol{s'}) = z_1 \mathrm{Tr}(z_i) = 0$ and $q_i(\boldsymbol{s}^*) = \mathrm{Tr}(z_i) = 0$, for $i \in [t-1]$. In addition, observe that

$$\begin{aligned}\mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\, p_t(\boldsymbol{s'}) f(\boldsymbol{s'})\right) &= \mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\, z_1 \mathrm{Tr}(z_t) f(\boldsymbol{s'})\right) \\ &= \mathrm{Tr}\left(z_t\right) \mathrm{Tr}\left(\lambda_{\boldsymbol{s'}}\, z_1 f(\boldsymbol{s'})\right).\end{aligned}$$

Following the lines of the proof of Theorem 5.1, we obtain that both erasures $f(\boldsymbol{s'})$ and $f(\boldsymbol{s}^*)$ can be recovered by downloading for every $\boldsymbol{s} \in \mathcal{S} \setminus \{\boldsymbol{s}^*, \boldsymbol{s'}\}$, the elements $\mathrm{Tr}(\lambda_{\boldsymbol{s}} p_i(\boldsymbol{s}) f(\boldsymbol{s}))$ and $\mathrm{Tr}(\lambda_{\boldsymbol{s}} q_i(\boldsymbol{s}) f(\boldsymbol{s}))$ for $i \in [t]$. Therefore, the result follows from the proof of Theorem 5.1 and the fact that the length of the augmented Cartesian code 2, $n = \prod_{i=1}^m n_i$, is given by the cardinality of the Cartesian set $\mathcal{S}$. ☐

Remark 5.4. *Given certain circumstances, it is possible to extend the repair scheme described above for two erasures to three erasures and beyond. This extension may be seen as analogous to the extension to three erasures in the Reed-Solomon case developed in [16]. In particular, such a repair scheme for three erasures which all differ on the same coordinate j, $\boldsymbol{s}^*, \boldsymbol{s'}$, and $\tilde{\boldsymbol{s}}$ begins with finding the kernels of the following maps:*

$$Tr(z(s'_j - s_j^*)), Tr(z(s'_j - \tilde{s_j})), Tr(z(s_j^* - \tilde{s}_j)).$$

Then, similar to the two erasure case, repair polynomials $\{p_1, \ldots, p_t\}, \{q_1, \ldots, q_t\}$, and $\{r_1, \ldots, r_t\}$ can be constructed which each evaluate to a basis element $\{z_1, \ldots, z_t\}$ at the associated erased coordinates $\boldsymbol{s}^*, \boldsymbol{s'}$, and $\tilde{\boldsymbol{s}}$ respectively. The basis chosen will be an extension of the basis for intersection of the three kernels. This choice of basis combined with the properties of the trace function will guarantee that each repair polynomial will evaluate to 0 at their non-associated erased coordinates, on all but two i. However, on these remaining i, the repair polynomials will evaluate to an element in the span of the outputs of other repair polynomials. This will create a system of equations which can be solved given the output of two polynomials on these remaining i. For example, $p_{t-1}(\boldsymbol{s'})$ and $p_t(\tilde{\boldsymbol{s}})$ would be enough given the appropriate repair polynomial definitions. Under particular circumstances, such as $t \mid char(K)$, these two outputs can be determined from the remaining nodes, and therefore produce $Tr(p_i(s)f(s))$, $Tr(q_i(s)f(s))$, and $Tr(r_i(s)f(s))$ at each erased coordinate for all i's. Then, a typical linear exact repair scheme can proceed from there to fix all three erasures.

VI. Comparisons and examples

The GW-scheme [8, Theorem 1] has the following parameters on the Reed-Solomon code $\mathrm{RS}(K, k)$: length $n \leq |K|$, dimension k and bandwidth $n-1$. Recall that the bandwidth represents the number of subsymbols needed to repair a symbol. We can also note its normalized bandwidth is $\lceil \frac{n-1}{t} \rceil$. Proposition 3.3 and Corollary 4.3 give the following parameters for the repair scheme on the augmented Reed-Muller code 1 (ARM1-scheme): length $n = |K|^m$, dimension $|K|^m - (q^t - k)^m$ and bandwidth $|K|^m - 1 + (t-1)(|K|^{m-1} - 1)$. We summarize this information, along with the parameters of the augmented Cartesian codes and Hermitian codes from [11, Theorem 13], in Table I.

It is clear that in general, the bandwidth and normalized bandwidth of the ARM1-scheme may be much larger than their counterparts in the GW-scheme, but the dimension and the length are also much larger. We now compare both schemes when the dimension and the base field $\mathbb{F}_q$ are the same.

Assume m divides t and $t = mt^*$. The GW-scheme and the ARM1-scheme repair the codes $\mathrm{RS}(\mathbb{F}_{q^t}, k)$ and $\mathrm{ARM1}(\mathbb{F}^m_{q^{t^*}}, k)$ when the dimensions are at most $q^t - q^{t-1}$ and $q^t - q^{t-m}$, respectively. An advantage of the $\mathrm{ARM1}(\mathbb{F}^m_{q^{t^*}}, k)$ comes when a code with dimension k^* between $q^t - q^{t-1}$ and $q^t - q^{t-m}$ is required. The restriction on the dimension of the GW-scheme implies that to employ an RS code, it must utilize an alphabet of size q^{t+1} to achieve dimension k^*. However, as the dimension of the code $\mathrm{ARM1}(\mathbb{F}^m_{q^{t^*}}, k)$ can be up to $q^t - q^{t-m}$, there are values between $q^t - q^{t-1}$ and q^t where we can still use $\mathrm{ARM1}(\mathbb{F}^m_{q^{t^*}}, k)$, whose bandwidth can be lower. We show this in the following example.

Example 6.1. Assume that a code of dimension $k^* = 648$ over a field of characteristic 3 is required. Observe that $3^6 - 3^5 = 486 < k^* < 3^6 = 729$. Over the field of size 3^6, there is a Reed-Solomon code with dimension 648, but the GW-scheme is not applicable. Indeed, the requirement that the dimension is at most $n - q^{t-1} = 486$ is not satisfied. To resolve this, a larger field such as one of size $3^7 = 2187$ may be used. Given that the GW-scheme requires the dimension to be at most $n - q^{t-1}$, the RS code length must then be bounded below by $648 + q^{t-1} = 1377$, meaning the bandwidth is at least 1376. The code $\mathrm{ARM1}(\mathbb{F}^2_{3^3}, 18)$ has dimension $k^* = 648$ and according to Corollary 4.3, bandwidth $b = |K|^m - 1 + (t-1)(|K|^{m-1} - 1) = 27^2 - 1 + (2)(27-1) = 780$. As a consequence, when a code of dimension 648 is required, the specifications for each family are as in Table 6.1.

Observe that the normalized bandwidth, which may be viewed as the number of symbols needed to repair a symbol, is smaller for the RS code. Indeed, every symbol is heavier in the RS case than in the ARM1 case, meaning a symbol equates to 7 subsymbols in the RS setting rather than 1 symbol

Code over $\mathbb{F}_{q^t}$ of length n	Restrictions	Dimension	Bandwidth
RS $\left(\mathbb{F}_{q^t}, k\right)$ [8, Theorem 1]	$k \leq n \leq q^t$ $k < q^t - q^{t-1}$	k	$n-1$
RM$(\mathbb{F}_{q^t}^m, k)$ [3, Theorem III.1]	$n = q^{tm}$ $k < q^t - 1$	$\binom{m+k}{k}$	$\left(\sqrt[m]{n}-1\right)\left(t - \lfloor \log_q(\sqrt[m]{n}-k-1)\rfloor\right)$
ARM1$(\mathbb{F}_{q^t}^m, k)$ Corollary 4.3	$n = q^{tm}$ $k < q^t - q^{t-1}$	$n - (\sqrt[m]{n}-k-1)^m$	$n-1+(t-1)(n^{(1-1/t)}-1)$
ARM2$(\mathbb{F}_{q^t}^m, k)$ Corollary 4.3	$n = q^{tm}$ $k < q^t - q^{t-1}$ $m < t, (m,p) = 1$	$n - m(\sqrt[m]{n}-k-2)-1$	$n-1+(t-1)(n^{(1-1/t)}-1)$
ACar1$(\mathcal{S}, \boldsymbol{k})$ Corollary 4.2	$n_1 \leq \cdots \leq n_m \leq q^t$ $n = \prod_{i=1}^{m} n_i \leq q^{tm}$ $0 \leq k_i \leq n_i - q^{t-1}$	$n - \prod_{j=1}^{m}(n_j - k_j - 1)$	$n-1+(t-1)\left(\frac{n}{n_m}-1\right)$
ACar2$(\mathcal{S}, \boldsymbol{k})$ Corollary 4.2	$n_1 \leq \cdots \leq n_m \leq q^t$ $n = \prod_{i=1}^{m} n_i \leq q^{tm}$ $0 \leq k_i \leq n_i - q^{t-1}$	$n - \sum_{i=1}^{m}(n_i - k_i - 2)$	$n-1+(t-1)\left(\frac{n}{n_m}-1\right)$
Hermitian [11, Theorem 13]	$q^t = r^2$, some r $n = r^3$ $m \leq n + r(r-1) - 2$ $-(q^\ell - 1)(r+1)$	$m - \frac{r(r-1)}{2}$	$(n-1)\left(\log q^t - \ell \log q\right)\frac{1}{\log q}$

TABLE I
EACH CODE IS DEFINED OVER $\mathbb{F}_{q^t}$ WITH BASE FIELD $\mathbb{F}_q$. FOR THE HERMITIAN CASE, THE TERM $\frac{1}{\log q}$ APPEARS HERE AND NOT IN [11, THEOREM 13] BECAUSE THE BITWIDTH FOR THIS PAPER, IS CALLED BANDWIDTH IN THAT PAPER.

Family	Scheme	Base field	Code over the field	Length	Bandwidth	Normalized bandwith
RS	GW	$\mathbb{F}_3$	$\mathbb{F}_{3^7}$	1377	1376	197
ARM1	Corollary 4.3	$\mathbb{F}_3$	$\mathbb{F}_{3^3}$	729	780	260

TABLE II
SPECIFICATIONS AND COMPARISONS OF CODES WITH DIMENSION 648 OVER FIELDS OF CHARACTERISTIC 3 AS DESCRIBED IN EXAMPLE 6.1

corresponding to 3 subsymbols in the analogous ARM1 one. Hence, we see that the bandwidth (number of subsymbols needed to repair a symbol) is lower in the ARM1 case whereas the normalized bandwidth is not.

Notice that applying the result in Corollary 4.3 to repair an erasure of ARM1$(\mathbb{F}_{3^3}^2, 18)$ gives a bandwidth of 780 whereas using [13, Theorem 2.5], the bandwidth is 837 [13, Example 4.1].

We can go further. As the following example shows, there are some values between $q^t - q^{t-1}$ and q^t where an augmented Cartesian code may have better parameters than an augmented Reed-Muller code.

Example 6.2. Assume that a code of dimension $k^* = 621$ over a field of characteristic 3 is required. Observe that $3^6 - 3^5 = 486 < k^* < 3^6 = 729$. As we explained in Example 6.1, we can use a Reed-Solomon code and the GW-scheme, but the RS code length must then be bounded below by $648 + q^{t-1} = 1377$, meaning the bandwidth is 1376 and the normalized bandwidth is 197.

Next, we consider whether we can use an augmented Reed-Muller code. Following the lines of Example 6.1, the code ARM1$(\mathbb{F}_{3^3}^2, 17)$ has dimension 629, bandwidth 780, and normalized bandwidth 260. If we decrease the parameter k from 17 to 16, we will obtain a code of dimension less than 621. If we increase t or m, the bandwidth will increase. Decreasing m from 2 to 1 yields a RS code, so the only possible option is to reduce t. Over $\mathbb{F}_{3^2}$, in order to have dimension 621, we need $m = 3$. In this case, according to Proposition 3.3, the dimension of ARM1$(\mathbb{F}_{3^2}^3, 5) = 665$. By Corollary 4.3, the bandwidth is 808 and the normalized bandwidth is 404.

Now take $q = 3, t = 3, m = 2, S_1 = \mathbb{F}_{3^3}$ and $S_2 = \mathbb{F}_{3^3}^* = S_1 \setminus \{0\}$. By Proposition 3.3, the dimension of ACar1$(S_1 \times S_2, (17, 18)) = (26)(27) - (9)(9) = 621$. Using Corollary 4.2, we obtain that the bandwidth of ACar1$(S_1 \times S_2, (17, 18))$ is $(26)(27) - 1 + (2)(26 - 1) = 621 - 1 + 2(25) = 670$ and thus the normalized bandwidth is 224. As a summary, when a code of dimension 621 is required, the specifications for each family are as in Table 6.2. Observe that the normalized bandwidth,

Family	Scheme	Base field	Code over the field	Length	Bandwidth	Normalized bandwith
RS	GW	$\mathbb{F}_3$	$\mathbb{F}_{3^7}$	1377	1376	197
ARM1	Corollary 4.3	$\mathbb{F}_3$	$\mathbb{F}_{3^3}$	729	780	260
ACar1	Corollary 4.2	$\mathbb{F}_3$	$\mathbb{F}_{3^3}$	702	670	224

TABLE III
SPECIFICATIONS AND COMPARISONS OF CODES WITH DIMENSION 621 OVER FIELDS OF CHARACTERISTIC 3 AS DESCRIBED IN EXAMPLE 6.2

which can be seen as the number of symbols needed to repair a symbol, is smaller for the RS codes. Just as in Example 6.1, every symbol is heavier in the RS case than in the ACar1 case, with a symbol representing 7 subsymbols in the RS case as opposed to 3 subsymbols in the ACar1 setting). Hence, despite the normalized bandwidth being smaller for the RS code, the bandwidth (meaning number of subsymbols needed to repair a symbol) is smaller for the ACar1 code.

The ARM1-scheme may be compared with other repair schemes in the literature, such as the repair scheme for algebraic geometry codes [11].

Example 6.3. By Corollary 4.3, the augmented code $\mathrm{ARM1}(\mathbb{F}_{2^3}^3, 3)$ has length 512, dimension 448 and bitwidth $\log_2 q(b) = |K|^m - 1 + (t-1)(|K|^{m-1} - 1) = 8^3 - 1 + (2)(8^2 - 1) = 637$. Whereas using the Hermitian code and the RS code in [11, Example 14], we the observe the comparisons given in Table 6.3.

Code	Base field	Code over the field	Length	Dimension	Bitwidth
Hermitian [11, Example 14]	$\mathbb{F}_{2^3}$	$\mathbb{F}_{2^6}$	512	448	$3(511) = 1533$
RS [11, Example 14]	$\mathbb{F}_{2^3}$	$\mathbb{F}_{2^9}$	512	448	$3(511) = 1533$
ARM1 $(\mathbb{F}_{2^3}^3, 3)$ Corollary 4.3	$\mathbb{F}_2$	$\mathbb{F}_{2^3}$	512	448	637

TABLE IV
SPECIFICATIONS AND COMPARISONS OF CODES CONSIDERED IN EXAMPLE 6.3

Note that using Corollary 4.3 to repair an erasure of $\mathrm{ARM1}(\mathbb{F}_{2^3}^3, 3)$ gives a bandwidth of 637 whereas using [13, Theorem 2.5] provides the bitwidth is 847 [13, Example 4.2].

We now compare augmented Reed-Muller and Cartesian codes when the length and the field $\mathbb{F}_{q^t}$ are both fixed.

Example 6.4. Assume that an augmented code of length $n > 8$ over the field $\mathbb{F}_{2^3}$ is required. The smallest augmented Reed-Muller code with length greater than 8 is the code $\mathrm{ARM1}(\mathbb{F}_{2^3}^2, k)$, where $0 \leq k \leq 4$. The bandwidth is $b = |K|^m - 1 + (t-1)(|K|^{m-1} - 1) = 8^2 - 1 + (2)(8-1) = 77$ and the normalized bandwidth is $\lceil 77/3 \rceil = 26$. The smallest augmented Cartesian code with length greater than 8 is the code $\mathrm{ACar1}(S_1 \times S_2, (k_1, k_2))$, where $n_1 = |S_1| = 4$, $n_2 = |S_2| = 8$, $k_1 = 0$ and $0 \leq k_2 \leq 4$. The bandwidth is $b = \prod_{i=1}^m n_i - 1 + (t-1)\left(\prod_{i=1}^{m-1} n_i - 1\right) = (4)(8) - 1 + (2)(4-1) = 37$ and the normalized bandwidth is $\lceil 37/3 \rceil = 13$.

The advantage of the larger dimension allowed by augmented codes can play an important role when a fixed rate is desired. By Table I, the maximum k for which $\mathrm{RS}(K, k)$ admits the repair scheme given in [8, Theorem 1] is $k^* = q^t - q^{t-1}$, which is the same than the dimension. As the length is q^t, the rate is upper bounded by $1 - \frac{1}{q}$. In a similar way, by Table I, the maximum k for which $\mathrm{ARM1}(K^m, k)$ admits the repair scheme given in Corollary 4.3 is $k^* = q^t - q^{t-1}$. In this case, the dimension is $q^{tm} - q^{(t-1)m}$. As the length is q^{tm}, the rate is upper bounded by $1 - \frac{1}{q^m}$. This means that with the augmented codes, we can achieve rates and it is not possible with the RS codes. See the following example.

Example 6.5. Assume that a code over the base field $\mathbb{F}_2$ and rate at least 0.875 is required. The maximum k for which $\mathrm{RS}(\mathbb{F}_{2^t}, k)$ admits the repair scheme given in [8, Theorem 1] is $k^* = 2^t - 2^{t-1}$, which is the same than the dimension. As the length is 2^t, the maximum rate is $\frac{2^t - 2^{t-1}}{2^t} = 1 - \frac{1}{2}$, which is less than 0.875. However, the augmented code $\mathrm{ARM1}(\mathbb{F}_{2^2}^3, 2)$ code has length $2^6 = 64$ and dimension $64 - = 2^6 - (2^2 - 2)^3 = 2^6 - 2^3 = 56$ yielding a rate 0.875. Further, since $2 \leq 2^2 - 2$, the augmented code $\mathrm{ARM1}(\mathbb{F}_{2^2}^3, 2)$ will admit a linear exact repair scheme with bandwidth 78 and normalized bandwidth 39.

The ARM codes will have greater repair bandwidth than the RM codes as q increases. However, the expression of the bandwidth makes it difficult to immediately appreciate the improvement in rate gained by implementing the ARM codes. Figure 5 illustrates the rate versus the repair bandwidth of the repair schemes of $\mathrm{RM}(\mathbb{F}_{5^4}^3, k)$, $\mathrm{ARM1}(\mathbb{F}_{5^4}^3, k)$, and $\mathrm{ARM2}(\mathbb{F}_{5^4}^3, k)$, for all values of k where the repair schemes developed in [8, Theorem 1] and Corollary 4.3 can be applied. The same figure demonstrates that RM codes admit repair schemes with much lower bandwidth than the ARM. However, it also reveals that the ARM codes have significantly higher rates, increasing from at most 0.2 to more than 0.99. Actual values can be found in Examples 6.6 and 6.7.

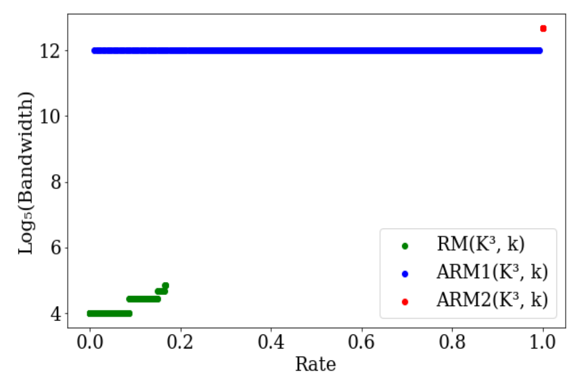


Fig. 5. Rate versus the repair bandwidth of the repair schemes of $\mathrm{RM}(\mathbb{F}_{5^4}^3, k)$, $\mathrm{ARM1}(\mathbb{F}_{5^4}^3, k)$, and $\mathrm{ARM2}(\mathbb{F}_{5^4}^3, k)$, for all values of k where the repair schemes developed in [8, Theorem 1] and Corollary 4.3 can be applied.

Example 6.6. Let $q = 5$, $t = 4$, and $m = 3$. The maximum k for which $\mathrm{RM}(K^m, k)$ admits the repair scheme given in [3, Theorem III.1] is 623. The maximum k for which $\mathrm{ARM1}(K^m, k)$ and $\mathrm{ARM2}(K^m, k)$ admit the repair scheme given in Corollary 4.3 is 499. The code $\mathrm{RM}(K^m, 623)$ has rate 0.167, bandwidth 2496, and normalized bandwidth 624. The code $\mathrm{ARM1}(K^m, 499)$ has rate 0.992, bandwidth 245312496, and normalized bandwidth 61328124. The code $\mathrm{ARM2}(K^m, 499)$ has rate 0.999998468, bandwidth 245312496, and normalized bandwidth 61328124.

Example 6.7. The maximum k for which $\mathrm{RM}(\mathbb{F}_{2^7}^5, k)$ admits the repair scheme given in [3, Theorem III.1] is 126. The maximum k for which $\mathrm{ARM1}(\mathbb{F}_{2^7}^5, k)$ and $\mathrm{ARM2}(\mathbb{F}_{2^7}^5, k)$ admit the repair scheme given in Corollary 4.3 is 63. The code $\mathrm{RM}(\mathbb{F}_{2^7}^5, 126)$ has rate 0.009002376, bandwidth 889, and normalized bandwidth 127. The code $\mathrm{ARM1}(\mathbb{F}_{2^7}^5, 62)$ has rate 0.96975, bandwidth 35970351097, and normalized bandwidth 5138621586. The code $\mathrm{ARM2}(\mathbb{F}_{2^7}^5, k)$ has rate 0.999999991, bandwidth 35970351097, and normalized bandwidth 5138621586.

The previous examples support the same conclusion. Reed-Muller codes admit repair schemes with superior bandwidth but have massively inferior rates when compared with the augmented codes.

Remark 6.8. *Observe that we can also compare with a classical repair scheme using the dual code. Let C be a code and assume $C^{\perp}$ has minimum distance $\delta^{\perp}$. Let $c = (c_1, \ldots, c_n)$ be an element in C and $d = (d_1, \ldots, d_n)$ in $C^{\perp}$. The equation $\sum_{j=1}^{n} c_j d_j = 0$ implies that if the entry c_i is lost, the entry d_i is nonzero, and the weight of d is $\delta^{\perp}$, then the entry c_i can be recovered by downloading $(\delta^{\perp} - 1)$ symbols. In other words, the bandwidth of the classical scheme is $(\delta^{\perp} - 1)t$.*

For instance, we can focus now on the RS codes. By Table I, the maximum k for which the $RS(\mathbb{F}_{q^t}, k)$ code of length $n = q^t$ admits the repair scheme given in [8, Theorem 1] is $k^* = q^t - q^{t-1}$. Thus the minimum distance of the dual code $RS(\mathbb{F}_{q^t}, k^*)^{\perp}$ is $\delta^{\perp} = q^t - q^{t-1} + 1$. As $RS(\mathbb{F}_{q^t}, k^*)$ is an MDS code, for every $i \in [n]$, there is an element $d = (d_1, \ldots, d_n)$ in $RS(\mathbb{F}_{q^t}, k^*)^{\perp}$ such that d_i is nonzero, and the weight of d is $\delta^{\perp}$. Thus, the classical repair scheme can be applied, and the bandwidth is $(\delta^{\perp} - 1)t = (q^t - q^{t-1})t$. It is not difficult to check that this number is larger than or equal to $q^t - 1$, the bandwidth of the GW repair scheme, if and only if $t > 1$.

As a consequence, for a $RS(\mathbb{F}_{q^t}, k)$, as long as $t > 1$, the repair scheme given in [8, Theorem 1] has a better bandwidth than the classical repair scheme using the dual code.

For the augmented codes, a deeper analysis is needed. First, the minimum distance $\delta^{\perp}$ of the dual code is required. As the dimension of an augmented code is large, it may not be the case that for every $i \in [n]$, there exists an element b in the dual such that b_i is nonzero and the weight of b is $\delta^{\perp}$. Assuming that such element exists, a comparison of the bandwidths may be undertaken.

For instance, by Example 6.1, the code $ARM1(\mathbb{F}_{3^3}^2, 18)$ has bandwidth 780. Using the Coding Theory Package for Macaulay2 [1], the minimum distance of the dual code $ARM1(\mathbb{F}_{3^3}^2, 18)^{\perp}$ is 324. Thus, the bandwidth of the classical repair scheme using the dual code would be $(323)3 = 969$.

A. Maximum rates and asymptotic behavior

Focusing on the improved rate, here we study the asymptotic behavior of the rate and the bandwidth rate $\dfrac{b}{nt}$, which represents the fraction of the codeword that is needed by the repair scheme to recover the erased symbol. We continue with the notation $K = \mathbb{F}_{q^t}$.

Reed-Solomon. The maximum k for which $\mathrm{RS}(K, k)$ admits the repair scheme given in [8, Theorem 1] is $k^* = q^t - q^{t-1}$. In this case, $\dim_K \mathrm{RS}(K, k) = q^t - q^{t-1}$ and the bandwidth at k^* is $b^* = (q^t - 1)$. Thus,

$$\begin{aligned}\lim_{t\to\infty} \frac{\dim_K \mathrm{RS}(K, k)}{n} &= \lim_{t\to\infty} \frac{q^t - q^{t-1}}{q^t} \\ &= \lim_{t\to\infty} \frac{q^t(1 - \frac{1}{q})}{q^t} = 1 - \frac{1}{q},\end{aligned}$$

and

$$\lim_{t\to\infty} \frac{\text{Bandwidth}}{tn} = \lim_{t\to\infty} \frac{q^t - 1}{tq^t} = \lim_{t\to\infty} \frac{1}{t} = 0.$$

Reed-Muller. The maximum k for which $\mathrm{RM}(K^m, k)$ admits the repair scheme given in [3, Theorem III.1] is $k^* = q^t - 2$. In this case, $\dim_K \mathrm{RM}(K^m, k^*) = \dbinom{m + q^t - 2}{q^t - 2}$ and bandwidth at k^* is $b^* = (q^t - 1)t$. Thus,

$$\begin{aligned}\lim_{t\to\infty} \frac{\dim_K \mathrm{RM}(K^m, k^*)}{n} &= \lim_{t\to\infty} \frac{\dbinom{m + q^t - 2}{q^t - 2}}{q^{tm}} \\ &= \lim_{t\to\infty} \frac{(m + q^t - 2)!}{(q^t - 2)!\ m!\ q^{tm}} \\ &= \lim_{t\to\infty} \frac{(q^t - 2 + 1) \cdots (q^t - 2 + m)}{m!\ q^{tm}} \\ &= \frac{1}{m!},\end{aligned}$$

and

$$\lim_{t\to\infty} \frac{\text{Bandwidth}}{tn} = \lim_{t\to\infty} \frac{(q^t - 1)t}{tq^{tm}} = \lim_{t\to\infty} \frac{q^t - 1}{q^{tm}} = 0.$$

Augmented Reed-Muller 1. The maximum k for which $\mathrm{ARM1}(K^m, k)$ admits the repair scheme given in Corollary 4.3 is $k^* = q^t - q^{t-1}$. In this case, $\dim_K \mathrm{ARM1}(K^m, k^*) = q^{tm} - q^{(t-1)m}$ and bandwidth at k^* is $b^* = |K|^m - 1 + (t - 1)(|K|^{m-1} - 1)$. Thus,

$$\lim_{t\to\infty} \frac{\dim_K \mathrm{ARM1}(K^m, k^*)}{n} = \lim_{t\to\infty} \frac{q^{tm} - q^{(t-1)m}}{q^{tm}} = 1 - \frac{1}{q^m},$$

and

$$\begin{aligned}\lim_{t\to\infty} \frac{\text{Bandwidth}}{nt} &= \lim_{t\to\infty} \frac{q^{tm} - 1 + (t - 1)(q^{t(m-1)} - 1)}{tq^{tm}} \\ &= \lim_{t\to\infty} \left[\frac{q^{tm} - 1}{tq^{tm}} + \frac{(t - 1)(q^{t(m-1)} - 1)}{tq^{tm}}\right] \\ &= 0.\end{aligned}$$

Augmented Reed-Muller 2. The maximum k for which $\mathrm{ARM2}(K^m, k)$ admits the repair scheme given in Corollary 4.3 is $k^* = q^t - q^{t-1}$. In this case, $\dim_K \mathrm{ARM2}(K^m, k^*) = q^{tm} - m(q^{t-1} - 1) - 1$ and bandwidth at k^* is $b^* = |K|^m - 1 + (t - 1)(|K|^{m-1} - 1)$. Thus,

$$\begin{aligned}&\lim_{t\to\infty} \frac{\dim_K \mathrm{ARM2}(K^m, k^*)}{n} \\ &= \lim_{t\to\infty} \frac{q^{tm} - mq^{t-1} + m - 1}{q^{tm}} \\ &= \lim_{t\to\infty} \left[1 - \frac{m}{q^{t(m-1)+1}} + \frac{m - 1}{q^{tm}}\right] = 1,\end{aligned}$$

and

$$\begin{aligned}
&\lim_{t\to\infty} \frac{\text{Bandwidth}}{nt} \\
&= \lim_{t\to\infty} \frac{q^{tm} - 1 + (t-1)(q^{t(m-1)} - 1)}{tq^{tm}} \\
&= \lim_{t\to\infty} \left[\frac{q^{tm}-1}{tq^{tm}} + \frac{(t-1)(q^{t(m-1)}-1)}{tq^{tm}}\right] = 0.
\end{aligned}$$

Augmented Cartesian Codes. The maximum $\boldsymbol{k}^*$ for which $\mathrm{ACar1}(\mathcal{S}, \boldsymbol{k}^*)$ admits the repair scheme given in Corollary 4.2 is $k_i^* = n_i - q^{t-1}$. In this case, $\dim \mathrm{ACar1}(\mathcal{S}, \boldsymbol{k}^*) = \prod_{j=1}^m n_j - \prod_{j=1}^m (n_j - k_j)$ and bandwidth at $\boldsymbol{k}^*$ is $b^* = \prod_{i=1}^m n_i - 1 + (t-1)\left(\prod_{i=1}^{m-1} n_i - 1\right)$. Thus,

$$\begin{aligned}
&\lim_{t\to\infty} \frac{\text{Bandwidth}}{nt} \\
&= \lim_{t\to\infty} \frac{\prod_{i=1}^m n_i - 1 + (t-1)(\prod_{i=1}^{m-1} n_i - 1)}{t\prod_{i=1}^m n_i} \\
&= \lim_{t\to\infty} \frac{\prod_{i=1}^m n_i - 1}{t\prod_{i=1}^m n_i} + \frac{t-1}{t}\left(\frac{\prod_{i=1}^{m-1} n_i}{\prod_{i=1}^m n_i} - \frac{1}{\prod_{i=1}^m n_i}\right) \\
&= \lim_{t\to\infty} \frac{\prod_{i=1}^m n_i - 1}{t\prod_{i=1}^m n_i} + \frac{t-1}{t}\left(\frac{1}{n_m} - \frac{1}{\prod_{i=1}^m n_i}\right).
\end{aligned}$$

In the case where $n_m = \mathcal{O}(t)$, we have that this limit is 0.

Now we will discuss the limit of the rate of an Augmented Cartesian Code 1 as the extension degree t approaches infinity through examples. We will find that varying the Cartesian evaluation set will result in augmented Cartesian codes with rate limits varying between 0 and $1 - \frac{1}{q^m}$, even when taking the maximum allowable values for the k_j.

Example 6.9. Suppose we are in the case when the evaluation set $\mathcal{S} = S_1 \times \cdots \times S_m$ is such that $n_j = q^{t-1}+1$ for all $j \in [m]$. Consider the augmented Cartesian 1 code $\mathrm{ACar1}(\mathcal{S}, \boldsymbol{k}^*)$ with maximum rate. This happens when $\boldsymbol{k}^* = \boldsymbol{1}$. The limit of the rate of this code as t approaches infinity is

$$\begin{aligned}
\lim_{t\to\infty} \frac{\dim_K \mathrm{ACar1}(\mathcal{S}, \boldsymbol{1})}{n} &= \lim_{t\to\infty} \frac{\prod_{i=1}^m n_i - \prod_{i=1}^m (n_i - k_i)}{\prod_{i=1}^m n_i} \\
&= \lim_{t\to\infty} \frac{(q^{t-1}+1)^m - (q^{t-1})^m}{(q^{t-1}+1)^m} = 0.
\end{aligned}$$

Example 6.10. Suppose we are in the case when the evaluation set $\mathcal{S} = S_1 \times \cdots \times S_m$ is such that $n_i = q^{t-1}$ for $i \in [m-1]$ and $n_m = 2q^{t-1}$. Consider the augmented Cartesian 1 code $\mathrm{ACar1}(\mathcal{S}, \boldsymbol{k}^*)$ with maximum rate. This happens when $k_i^* = 0$ for $i \in [m-1]$ and $k_m^* = q^{t-1}$. The limit of the rate of this code as t approaches infinity is

$$\begin{aligned}
\lim_{t\to\infty} \frac{\dim_K \mathrm{ACar1}(\mathcal{S}, \boldsymbol{k})}{n} &= \lim_{t\to\infty} \frac{\prod_{i=1}^m n_i - \prod_{i=1}^m (n_i - k_i)}{\prod_{i=1}^m n_i} \\
&= \left(\lim_{t\to\infty} 1 - \frac{\prod_{i=1}^m q^{t-1}}{2\prod_{i=1}^m q^{t-1}}\right) \\
&= 1 - \frac{1}{2} = \frac{1}{2}.
\end{aligned}$$

Example 6.11. Lastly, consider the case when $|S| = K^m$. As this is an augmented Reed-Muller code, we obtain

$$\lim_{t\to\infty} \frac{\dim_K \mathrm{ACar1}(\mathcal{S}, \boldsymbol{k})}{n} = \lim_{t\to\infty} \frac{q^{tm} - q^{(t-1)m}}{q^{tm}} = 1 - \frac{1}{q^m}.$$

A similar situation happens with the augmented Cartesian codes 2. We summarize these findings in Table V. Note each code in Table V will have identical bandwidth rate ($\frac{b}{nt}$) as the field extension t gets large. In particular, for each code $\frac{b}{nt}$ tends toward 0 as t approaches infinity. Importantly, this means that the fraction of subsymbols in an entire codeword that need to be transmitted to repair one erased symbol tends to 0 even in the newly introduced ACar1, ACar2, ARM1, and ARM2 codes.

Code	Dimension	$\lim_{t\to\infty}$ Rate	$\frac{b^*}{t}$
$\mathrm{RS}(K, max)$	$q^t - q^{t-1}$	$1 - \frac{1}{q}$	$\frac{q^t-1}{t}$
$\mathrm{RM}(K^m, max)$	$\binom{m+q^t-2}{q^t-2}$	$\frac{1}{m!}$	$q^t - 1$
$\mathrm{ARM1}(K^m, max)$	$q^{tm} - q^{(t-1)m}$	$1 - \frac{1}{q^m}$	$q^{t(m-1)}(\frac{q^t-1}{t} + 1) - 1$
$\mathrm{ARM2}(K^m, max)$	$q^{tm} - m(q^{t-1}-1) - 1$	1	$q^{t(m-1)}(\frac{q^t-1}{t} + 1) - 1$
$\mathrm{ACar1}(\mathcal{S}, max)$	$\prod_{j=1}^m n_j - q^{m(t-1)}$	$[0,1]$	$\prod_{i=1}^{m-1} n_i(\frac{n_m-1}{t} + 1) - 1$
$\mathrm{ACar2}(\mathcal{S}, max)$	$\prod_{i=1}^m n_i - m(q^{t-1}-1) - 1$	$[0,1]$	$\prod_{i=1}^{m-1} n_i(\frac{n_m-1}{t} + 1) - 1$

TABLE V

ASYMPTOTIC BEHAVIOR OF THE RS, RM, ARM1 AND ARM2, WHEN EACH ACHIEVES THE MAXIMUM DIMENSION SO THE ASSOCIATED REPAIR SCHEME CAN BE APPLIED. OBSERVE THAT BY THE DISCUSSION ABOVE, FOR EACH FAMILY OF CODES, $\lim_{t\to\infty} \frac{b^*}{nt} \to 0$, WHICH MEANS THAT THE FRACTION OF A CODEWORD THAT NEED TO BE TRANSMITTED TO REPAIR ONE ERASED SYMBOL TENDS TO 0, EVEN FOR THE AUGMENTED CODES.

As expected, the augmented codes, which were designed to maximize the rate of the code, have a higher repair bandwidth as well, due to the trade-off between the rate of a code and the bandwidth of its associated repair scheme. In the end, neither of these schemes is objectively better than the other. Any potential user should opt to use the scheme that best deals with the parameter most important to their application, whether that be one that requires high rate codes or one that requires low bandwidth recovery.

VII. Conclusions

In this paper, we introduce a new family of evaluation codes, called augmented Cartesian codes, along with repair schemes for single and certain multiple erasures. They can be designed to have higher rate than their traditional counterparts and include as a special case augmented Reed-Muller codes. In some circumstances, these repair schemes may have lower bandwidth and bitwidth than comparable algebraic geometry codes (such as Reed-Solomon or Hermitian codes). There are parameter ranges in which repairing Reed-Solomon codes may not be available, such as dimension between $q^t - q^{t-1}$ and q^t over $\mathbb{F}_{q^t}$. In some cases, augmented Reed-Muller codes may be designed along with repair schemes for single or pairs of erasures. More generally, we can use augmented Cartesian codes to provide high-rate codes with repair schemes for single erasures and certain pairs of erasures in those settings where the augmented Reed-Muller codes are not.

References

[1] T. Ball, E. Camps, H. Chimal-Dzul, D. Jaramillo-Velez, H. López, N. Nichols, M. Perkins, I. Soprunov, G. Vera and G. Whieldon, Coding Theory Package for *Macaulay2*, Journal of Software for Algebra and Geometry, (2021), to appear.

[2] E. Camps, H. H. López, G. L. Matthews and E. Sarmiento, Polar Decreasing Monomial-Cartesian Codes, IEEE Transactions on Information Theory, **67** (2021), no. 6, 3664–3674.
[3] T. Chen and X. Zhang, Repairing Generalized Reed-Muller Codes, https://arxiv.org/pdf/1906.10310.pdf.
[4] P. Delsarte, J. M. Goethals and F. J. Mac Williams, On generalized Reed-Muller codes and their relatives, Information and control, **16** (1970), no. 5, 403–442.
[5] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright and K. Ramchandran, Network coding for distributed storage systems, IEEE Transactions on Information Theory, **56** (2010), no. 9, 4539–4551.
[6] A. Dimakis, K. Ramchandran, Y. Wu and C. Suh, A survey on network codes for distributed storage, Proceedings of the IEEE, **99** (2011), no. 3, 476–489.
[7] O. Geil, and C. Thomsen, Weighted Reed-Muller codes revisited, Designs Codes and Cryptography, **66** (2013), 195–220.
[8] V. Guruswami and M. Wootters, Repairing Reed-Solomon Codes, IEEE Transactions on Information Theory, **63** (2017), no. 9, 5684–5698.
[9] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*, Cambridge University Press, Cambridge, 2003.
[10] D. Jaramillo, M. Vaz Pinto, and R. H. Villarreal, Evaluation codes and their basic parameters, Designs Codes and Cryptography, **89** (2021), 269–300.
[11] L. Jin, Y. Luo and C. Xing, Repairing Algebraic Geometry Codes, IEEE Transactions on Information Theory, **64** (2018), no. 2, 900–908.
[12] H. H. López, G. L. Matthews and I. Soprunov, Monomial-Cartesian codes and their duals, with applications to LCD codes, quantum codes, and locally recoverable codes, Designs Codes and Cryptography, **88** (2020), 1673–1685.
[13] H. H. López, G. L. Matthews and D. Valvo, Augmented Reed-Muller Codes of High Rate and Erasure Repair, Proceedings of the IEEE, (2021), to appear.
[14] H. H. López, C. Rentería-Márquez and R. H. Villarreal, Affine Cartesian codes, Designs, Codes and Cryptography **71** (2014), no. 1, 5–19.
[15] H. H. López, I. Soprunov and R. H. Villarreal, The dual of an evaluation code, Designs, Codes and Cryptography, (2021).
[16] H. Dau, I. Duursma, H. Kiah and O. Milenkovic, Repairing Reed-Solomon Codes With Multiple Erasures, IEEE Transactions on Information Theory, **64** (2018), no. 10, 6567–6582.
[17] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press (1994).

Hiram H. López is an Assistant Professor at Cleveland State University. He earned the B.S. degree in applied mathematics from Autonomous University of Aguascalientes in 2008 and the Ph.D. degree in mathematics from CINVESTAV-IPN in 2016. After a postdoctoral position at Clemson University from 2016 to 2018 he joined Cleveland State University in 2019. His research interests include coding theory, commutative algebraic and image processing.

Gretchen L. Matthews is a Professor in the Department of Mathematics at Virginia Tech. She earned the B.S. degree in mathematics from Oklahoma State University in 1995 and the Ph.D. degree in mathematics from Louisiana State University in 1999. Following postdoctoral work at the University of Tennessee, she was on the faculty at Clemson University before joining Virginia Tech in 2018. Her research interests include algebraic geometry and combinatorics and their applications to coding theory and cryptography.

Daniel Valvo is a Ph.D. student in mathematics at Virginia Tech. He earned the B.S. and M.S. degrees in 2018 and 2020, respectively, both in mathematics at Virginia Tech.