Long MDS Codes for Optimal Repair Bandwidth

Zhiying Wang*, Itzhak Tamo*†, and Jehoshua Bruck*
*Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, USA
†Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 84105, Israel
{*zhiying, tamo, bruck*}*@caltech.edu*

***Abstract*—MDS codes are erasure-correcting codes that can correct the maximum number of erasures given the number of redundancy or parity symbols. If an MDS code has r parities and no more than r erasures occur, then by transmitting all the remaining data in the code one can recover the original information. However, it was shown that in order to recover a single symbol erasure, only a fraction of $1/r$ of the information needs to be transmitted. This fraction is called the *repair bandwidth (fraction)*. Explicit code constructions were given in previous works. If we view each symbol in the code as a vector or a column, then the code forms a 2D array and such codes are especially widely used in storage systems. In this paper, we ask the following question: given the length of the column l, can we construct high-rate MDS array codes with optimal repair bandwidth of $1/r$, whose code length is as long as possible? In this paper, we give code constructions such that the code length is $(r+1)\log_r l$.**

I. Introduction

MDS (maximum distance separable) codes are optimal error-correcting codes in the sense that they have the largest minimum distance given the number of parity symbols. If each symbol is a vector or a column, we call such a code an MDS array code (e.g. [2], [7], [11], [22], [23]). In (distributed) storage systems, each column is usually stored in a different disk, and MDS array codes are widely used to protect data against erasures due to their error correction ability and low computational complexity. In this paper, we call each symbol a column or a node, and the column length, or the vector size of a symbol, is denoted by l.

If an MDS code has r parities, then it can correct up to r erasures of entire columns. In this paper, we not only would like to recover any e erasures, $e \leq r$, but also care about the efficiency in recovery: what is the fraction of the remaining data transmitted in order to correct e erasures? We call this fraction the *repair bandwidth (fraction)*. For example, if $e = r$ erasures happen, it is obvious that we have to transmit all of the remaining information, therefore, the fraction is 1. For $e = 1$ erasure it was shown in [8] (which also formulated the repair problem) that this fraction is actually lowered bounded by $1/r$. If $e \leq r$ symbols are erased and we repair them exactly as they were, this fraction is lower bounded by e/r [17]. If this bound is achieved for some code, we say it has optimal repair. Since the repair of information is much more crucial than redundancy, and we study mainly high-rate codes, we will focus on the optimal repair of information or systematic nodes. Moreover, since single erasure is the most common scenario in practice, we assume $e = 1$. For example, in Figure 1, we show an MDS code with 4 systematic nodes, $r = 2$ parity nodes, and column length $l = 2$. One can check that this code can correct any two erasures, therefore it is an MDS code. In order to repair any systematic node, only $1/r = 1/2$ fraction of the remaining information is transmitted. Thus this code has optimal repair.

In [12]–[14], [20], [21] codes achieving the repair bandwidth lower bound were studied where the number of systematic nodes is less than the number of parity nodes (low code rate). For arbitrary code rate, [6], [15] proved that the lower bound is asymptotically achievable when the column length l goes to infinity. And [3]–[5], [9], [10], [16], [17], [19] studied codes with more systematic nodes than parity nodes (high code rate) and finite l, and achieved the lower bound of the repair bandwidth. If we are interested in the *code length*, i.e., the number of systematic nodes given l, low-rate codes have a linear code length $l+1$ [13], [14]; on the other hand, high-rate constructions are relatively short. For example, suppose that we have 2 parity nodes, then the number of systematic nodes is only $\log l$ in all of the constructions, except for [5] it is $2\log l$. In [18] it is shown that an upper bound for the code length is $k \leq 1 + l\binom{l}{l/2}$, but the tightness of this bound is not known. It is obvious that there is a big gap between this upper bound and the constructed codes.

The main contribution of this paper is to construct codes with 2 parity nodes and $3\log l$ systematic nodes. The code uses a finite field of size $1 + 2\log l$. Moreover, we will give a general construction of high-rate codes with $(r+1)\log_r l$ systematic nodes for arbitrary number of parities r. It turns out that this construction is a combination of the code in [5] and also [3], [10], [16].

The rest of the paper is organized as follows: in Section II we will formally introduce the repair bandwidth and the code length problem. In Section III codes with 2 parity nodes are constructed, and we show that the code length is $3\log l$. Generalized code constructions for arbitrary number of parities are given in Section IV and finally we conclude in Section V.

II. Problem Settings

An (n,k,l) MDS array code is an $(n-k)$-erasure-correcting code such that each symbol is a column of length l. The number of systematic symbols is k and the number of parity symbols is $r = n - k$. We call each symbol a column or a node, and k the *code length*. We assume that the code is systematic, hence the first k nodes of the code

N1	N2	N3	N4	P1	P2
a	b	c	d	$a+b+c+d$	$2a+w+2b+3c+d$
w	x	y	z	$w+x+y+z$	$3w+b+3x+2y+z$

Figure 1. (n=6,k=4,l=2) MDS code over finite field $\mathbb{F}_4$ generated by primitive polynomial x^2+x+1. Here 2 is a primitive element of the field. The first 4 nodes are systematic and the last 2 are parities. To repair $N1$ transmit the first row from every remaining node. To repair $N2$ transmit the second row. To repair $N3$ transmit the sum of both rows. And to repair $N4$ transmit the sum of the first row and 2 times the second row from nodes $N1, N2, N3, P1$, and the sum of the first row and 3 times the second row from node $P2$.

are information or systematic nodes, and the last r nodes are parity or redundancy nodes.

Suppose the columns of the code are $C_1, C_2, \ldots, C_n$, each being a column vector in $\mathbb{F}^l$, for some finite field $\mathbb{F}$. We assume that for parity node $k+i$, information node j, the *coding matrix* is $A_{i,j}$ of size $l \times l$, $i \in [r]$, $j \in [k]$. And the parity columns are computed as

$$C_{k+i} = \sum_{j=1}^{k} A_{i,j} C_j,$$

for all $i \in [r]$. For example, in Figure 1, the coding matrices are $A_{1,j} = I$ for all $j \in [k]$ and $A_{2,j}$, $j = 1,2,3,4$ are

$$\begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Here the finite field is $\mathbb{F}_4$ generated by x^2+x+1. In our constructions, we require that $A_{1,j} = I$ for all $j \in [k]$. Hence the first parity is the row sum of the information array. Even though this assumption is not necessarily true for an arbitrary linear MDS array code, it can be shown that any linear code can be equivalently transformed into one with such coding matrices [18].

Suppose a code has optimal repair for any systematic node i, $i \in [k]$, meaning only a fraction of $1/r$ data is transmitted in order to repair it. When a systematic node i is erased, we are going to use size $l/r \times l$ matrices $S_{i,j}$, $j \neq i, j \in [n]$, to repair the node: From a surviving node j, we are going to compute and transmit $S_{i,j}C_j$, which is only $1/r$ of the information in this node.

Notations: In order to simplify the notations, we write $S_{i,j}$ and $S_{i,k+t}A_{t,j}$ both as matrices of size $l/r \times l$ and the subspaces of their row spans.

Optimal repair of a systematic node i is equivalent to the following **subspace property**: There exist matrices $S_{i,j}$, $j \neq i, j \in [n]$, all with size $l/r \times l$, such that for all $j \neq i, j \in [k], t \in [r]$,

$$S_{i,j} = S_{i,k+t}A_{t,j}, \tag{1}$$

where the equality is defined on the row spans instead of the matrices. And

$$\sum_{t=1}^{r} S_{i,k+t}A_{t,i} = \mathbb{F}^l. \tag{2}$$

Here the sum of two subspaces A, B of $\mathbb{F}^l$ is defined as $A+B = \{a+b : a \in A, b \in B\}$. Obviously, the dimension of each subspace $S_{i,k+t}A_{t,i}$ is no more than l/r, and the sum of r such subspaces has dimension no more than l. This means these subspaces intersect only on the zero vector. Therefore, the sum is actually the direct sum of vector spaces. Moreover, we know that each $S_{i,k+t}$ has full rank l/r.

We claim that (1) (2) are necessary and sufficient conditions for optimal repair. The sketch of the proof is as follows: suppose the code has optimal repair bandwidth, then we need to transmit l/r elements from each surviving column. Suppose we transmit $S_{i,j}C_j$ from a systematic node $j \neq i, j \in [k]$, and $S_{i,k+t}C_{k+t} = \sum_{z=1}^{k} S_{i,k+t}A_{t,z}C_z$ from a parity node $k+t \in [k+1, k+r]$. Our goal is to recover C_i and cancel out all C_j, $j \neq i, j \in [k]$. In order to cancel out C_j, (1) must be satisfied. In order to solve C_i, all equations related to C_i must have full rank l, so (2) is satisfied. One the other hand, if (1) (2) are satisfied, one can transmit $S_{i,j}C_j$ from each node j, $j \neq i, j \in [n]$ and optimally repair the node i. Similar interference alignment technique was first introduced in [6], [15] for the repair problem. Also, [13] was the first to formally prove similar conditions.

It is shown in [18] that we can further simplify our repair strategy of node i and assume $S_{i,j} = S_i$, for all $j \neq i, j \in [n]$ by equivalent transformation of the coding matrices (probably with an exception of the strategy of one node). Then the **subspace property** becomes for any $j \neq i, j \in [k], t \in [r]$,

$$S_i = S_i A_{t,j}. \tag{3}$$

Again the equality means equality of row spans. And the sum of subspaces satisfies

$$\sum_{t=1}^{r} S_i A_{t,i} = \mathbb{F}^l. \tag{4}$$

Notice that if (3) is satisfied, we can say that S_i is an invariant subspace of $A_{t,j}$ (multiplied on the left) for all parity nodes $k+t$ and all information nodes $j \neq i$. If $A_{t,j}$ is diagonalizable and has l linearly independent left eigenvectors, an invariant subspace has a set of basis which are all eigenvectors of $A_{t,j}$. As a result, our goal is to find matrices $A_{t,j}$ and their invariant subspaces. And by using sufficiently large finite field and varying the eigenvalues of the coding matrices, we are able to ensure that the codes are MDS. Therefore, we will first focus on finding eigenvectors of the coding matrices and then discuss about the eigenvalues.

For example, in Figure 1, the matrices S_i, $i = 1,2,3$ are

$$(1,0), (0,1), (1,1).$$

One can check that the subspace property (3)(4) is satisfied for $i \in [3]$. For instance, since $S_3 = (1,1)$ is an eigenvector for $A_{t,j}$, $t = 1,2, j = 1,2,4$, we have $S_3 = S_3 A_{t,j}$. And it is easy to check that $S_3 \oplus S_3 A_{2,3} = \text{span}(1,1) \oplus \text{span}(3,2) = \mathbb{F}^2$. For the node $N4$, the matrices $S_{4,j}$'s are not equal. In fact $S_{4,j} = (1,2)$ for $j = 1,2,3,5$ and $S_{4,6} = (1,3)$.

III. Code Constructions with 2 Parities

In this section, we are going to construct codes with column length $l = 2^m$, $k = 3m$ systematic nodes, and $r = 2$ parity nodes. Here m is some integer. As we showed in the previous section, we can assume the coding matrices are

$$\begin{pmatrix} I & \cdots & I \\ A_1 & \cdots & A_k \end{pmatrix}, \tag{5}$$

where $A_{1,i} = I$ and $A_{2,i} = A_i$ correspond to parity 1 and 2 respectively.

Now we only need to find coding matrices A_i's, and subspaces S_i's. For now we only care about eigenvectors of A_i, not its eigenvalues because eigenvectors determine the repair bandwidth. Later we will show that using a large enough finite field, we can choose the eigenvalues such that the code is indeed MDS. In the following construction, for any $i \in [k]$, A_i has two different eigenvalues $\lambda_{i,0}, \lambda_{i,1}$, each corresponding to $l/2 = 2^{m-1}$ eigenvectors. Denote these eigenvectors as

$$V_{i,0} = \begin{pmatrix} v_{i,1} \\ v_{i,2} \\ \vdots \\ v_{i,l/2} \end{pmatrix}$$

for eigenvalue $\lambda_{i,0}$, and

$$V_{i,1} = \begin{pmatrix} v_{i,l/2+1} \\ v_{i,l/2+2} \\ \vdots \\ v_{i,l} \end{pmatrix}$$

for eigenvalue $\lambda_{i,1}$. Therefore, A_i can be computed as

$$A_i = \begin{pmatrix} V_{i,0} \\ V_{i,1} \end{pmatrix}^{-1} \begin{pmatrix} \lambda_{i,0} I_{\frac{l}{2}\times\frac{l}{2}} & \\ & \lambda_{i,1} I_{\frac{l}{2}\times\frac{l}{2}} \end{pmatrix} \begin{pmatrix} V_{i,0} \\ V_{i,1} \end{pmatrix}.$$

By abuse of notations, we also use $V_{i,0}, V_{i,1}$ to represent the eigenspace corresponding to $\lambda_{i,0}, \lambda_{i,1}$, respectively. Namely, $V_{i,0} = \text{span}\{v_{i,1}, \ldots, v_{i,l/2}\}$ and $V_{i,1} = \text{span}\{v_{i,l/2+1}, \ldots, v_{i,l}\}$.

When a systematic node i is erased, $i \in [k]$, we are going to use S_i to rebuild it. The **subspace property** becomes

$$S_i = S_i A_j, \quad \forall j \neq i, j \in [k], \tag{6}$$

$$S_i + S_i A_i = \mathbb{F}^l. \tag{7}$$

In the following construction, e_a, $a \in [0, l-1]$, are some basis of $\mathbb{F}^l$, for example, one can think of them as the standard basis. The subscript a is represented by its binary expansion, $a = (a_1, a_2, \ldots, a_m)$. For example, if $l = 16, m = 4, a = 5$, then $e_5 = e_{(0,1,0,1)}$ and $a_1 = a_3 = 0, a_2 = a_4 = 1$.

In order to construct the code, we first define 3 sets of vectors for $i \in [m]$:

$$\begin{aligned} P_{i,0} &= \{e_a : a_i = 0\}, \\ P_{i,1} &= \{e_a : a_i = 1\}, \\ Q_i &= \{e_a + e_b : a_i + b_i = 1, a_j = b_j, \forall j \neq i\}. \end{aligned}$$

For example, if $m = 2, i = 1$, then $P_{1,0} = \{e_{(0,0)}, e_{(0,1)}\} = \{e_0, e_1\}$, $P_{1,1} = \{e_{(1,0)}, e_{(1,1)}\} = \{e_2, e_3\}$, and $Q_1 = \{e_{(0,0)} + e_{(1,0)}, e_{(0,1)} + e_{(1,1)}\} = \{e_0 + e_2, e_1 + e_3\}$. **Notation**: The subscript i for sets $P_{i,u}, Q_i$ and a_i (the i-th digit of vector a) is written modulo m. For example, if $i \in [tm+1, (t+1)m]$ for some integer t, then $P_{i,u} := P_{i-tm,u}$.

Construction 1 *The $(n = 3m+2, k = 3m, l = 2^m)$ code has coding matrices $A_i, i \in [k]$, each with two distinct eigenvalues, and eigenvectors $V_{i,0}, V_{i,1}$. When node i is erased, we are going to use S_i to rebuild. We construct the code as follows:*

1) *For $i \in [m]$, $V_{i,0} = \text{span}(Q_i)$, $V_{i,1} = \text{span}(P_{i,1})$, $S_i = \text{span}(P_{i,0})$.*
2) *For $i \in [m+1, 2m]$, $V_{i,0} = \text{span}(P_{i,0})$, $V_{i,1} = \text{span}(Q_i)$, $S_i = \text{span}(P_{i,1})$.*
3) *For $i \in [2m+1, 3m]$, $V_{i,0} = \text{span}(P_{i,0})$, $V_{i,1} = \text{span}(P_{i,1})$, $S_i = \text{span}(Q_i)$.*

Example 1 *Deleting the node N4, Figure 1 is a code using Construction 1 and $l = 2$. Another example of $l = 4$ is shown in Figure 2. One can check (6) holds. For instance, $S_1 = \text{span}\{e_0, e_1\} = \text{span}\{e_0 + e_1, e_1\}$ is an invariant subspace of A_2. So $S_1 = S_1 A_2$. If the two eigenvalues of A_i are distinct, it is easy to show that $S_i \oplus S_i A_i = \mathbb{F}^4, \forall i \in [6]$.*

The above example shows that for $m = 1, 2$, the constructed code has optimal repair. It is true in general, as the following theorem suggests.

Theorem 2 *Construction 1 is a code with optimal repair bandwidth $1/2$ for rebuilding any systematic node.*

Proof: By symmetry of the first two cases in the construction, we are only going to show that the rebuilding of node i, $i \in [m] \cup [2m+1, 3m]$ is optimal. Namely, the subspace property (6)(7) is satisfied. Recall that $S_i A_j = S_i$ is equivalent to S_i being an invariant subspace of A_j.

Case 1: $i \in [m]$.

- When $j \in [tm+1, (t+1)m], j - tm \neq i, t \in \{0, 1\}$, define $B = \{e_a : a_j = 1 - t, a_i = 0\} \cup \{e_a + e_b : a_j + b_j = 1, a_i = b_i = 0, a_z = b_z, \forall z \neq i, j\}$. Then it is easy to see that $S_i = \text{span}(P_{i,0}) = \text{span}(B)$. Moreover, each vector in set B is an eigenvector of A_j, therefore S_i is an invariant subspace of A_j.
- When $j - m = i$, $S_i = V_{j,0} = \text{span}(P_{i,0})$, so S_i is an eigenspace of A_j.
- When $j \in [2m+1, 3m]$, we can see that every vector in $P_{i,0}$ is a vector in $V_{j,0} = \text{span}(P_{j,0})$ or in $V_{j,1} = \text{span}(P_{j,1})$, hence it is an eigenvector of A_j.
- When $j = i$, consider a vector $e_a \in P_{i,0}$, then $a_i = 0$. And $e_a = (e_a + e_b) - e_b$ where $b_i = 1, b_j = a_j$ for all $j \neq i$. Here both $e_a + e_b$ and e_b are eigenvectors of A_i.

$$\begin{aligned} e_a A_i &= (e_a + e_b) A_i - e_b A_i \\ &= \lambda_{i,0}(e_a + e_b) - \lambda_{i,1} e_b \\ &= (\lambda_{i,0} - \lambda_{i,1}) e_b + \lambda_{i,0} e_a. \end{aligned}$$

Because $\lambda_{i,0} \neq \lambda_{i,1}$, we get $\text{span}\{e_a A_i, e_a\} = \text{span}(e_a, e_b)$. Hence $S_i A_i + S_i = \text{span}\{e_a, e_b : a_i = 0, b_i = 1, a_j = b_j, \forall j \neq i\} = \mathbb{F}^l$.

	N1	N2	N3	N4	N5	N6
1st eigenspace of A_i	e_0+e_2 e_1+e_3	e_0+e_1 e_2+e_3	e_0 e_1	e_0 e_2	e_0 e_1	e_0 e_2
2nd eigenspace of A_i	e_2 e_3	e_1 e_3	e_0+e_2 e_1+e_3	e_0+e_1 e_2+e_3	e_2 e_3	e_1 e_3
S_i	e_0 e_1	e_0 e_2	e_2 e_3	e_1 e_3	e_0+e_2 e_1+e_3	e_0+e_1 e_2+e_3

Figure 2. (n=8,k=6,l=4) code. The first parity node is assumed to be the row sum, and the second parity is computed using coding matrices A_i. In order to rebuild node i, S_i is multiplied to each surviving node. The first $2m=4$ nodes have optimal access, and the last $m=2$ nodes have optimal update.

Case 2: $i \in [2m+1, 3m]$.

- When $j = i-m$ or $j = i-2m$, $S_i = \text{span}(Q_i)$ is an eigenspace of A_j.
- When $j \in [tm+1, (t+1)m]$, and $j \neq i-tm$ for $t \in \{0,1\}$, define $D = \{e_a+e_b : a_j = b_j = 1-t, a_i+b_i = 1, a_z = b_z, \forall z \neq i,j\} \cup \{e_a+e_b+e_c+e_d : a_j = b_j = 0, c_j = d_j = 1, a_i+b_i = 1, c_i+d_i = 1, a_z = b_z = c_z = d_z, \forall z \neq i,j\}$. We can see that $S_i = \text{span}(Q_i) = \text{span}(D)$ and every vector in D is an eigenvector of A_j.
- When $j \in [2m+1, 3m], j \neq i$. We can see that $Q_i = \{e_a+e_b : a_j = b_j = 0, a_i+b_i = 1, a_z = b_z, \forall z \neq i,j\} \cup \{e_a+e_b : a_j = b_j = 1, a_i+b_i = 1, a_z = b_z, \forall z \neq i,j\}$. Apparently, every vector in Q_i is a sum of two vectors in $P_{j,0}$ or two vectors in $P_{j,1}$. So $S_i = \text{span}(Q_i)$ is an invariant subspace of A_j.
- When $j = i$, consider any $e_a+e_b \in Q_i$, where $a_i = 1, b_i = 0, a_z = b_z, \forall z \neq i$. We have

$$(e_a+e_b)A_i = \lambda_{i,1}e_a + \lambda_{i,0}e_b.$$

  Because $\lambda_{i,0} \neq \lambda_{i,1}$, we get $\text{span}\{(e_a+e_b)A_i, e_a+e_b\} = \text{span}\{e_a, e_b\}$. Thus $S_iA_i + S_i = \text{span}\{e_a, e_b : a_i = 1, b_i = 0, a_z = b_z, \forall z \neq i\} = \mathbb{F}^l$.

■

It should be noted that if we shorten the code and keep only the first $2m$ systematic nodes in the code, then it is actually equivalent to the code in [5]. The repairing of the first $2m$ nodes does not require computation within each remaining node, since only standard bases are multiplied to the surviving columns (e.g. Figure 2). We call such repair *optimal access*. It is shown in [18] that if a code has optimal access, then the code has no more than $2m$ nodes. On the other hand, the shortened code with the last m systematic nodes in the above construction is equivalent to that of [3], [10], [16]. Since the coding matrices A_i, $i \in [2m+1, 3m]$ are all diagonal, every information entry is included in only $r+1$ entries in the code. We say such a code has *optimal update*. In [18] it is proven that an optimal-update code with diagonal coding matrices has no more than m nodes. Therefore, our code is a combination of the longest optimal-access code and the longest optimal-update code, which provides tradeoff among access, update, and the code length. The shortening technique was also used in [13] in order to get optimal-repair code with different code rates.

In addition, if we try to extend an optimal-access code $\mathcal{C}$ with length $2m$ to a code $\mathcal{D}$ with length k, so that $\mathcal{C}$ is a shortened code of $\mathcal{D}$, then the following theorem shows that $k = 3m$ is largest code length. Therefore, our construction is longest in the sense of extending $\mathcal{C}$.

Theorem 3 *Any extended code of an optimal-access code of length $2m$ will have no more than $3m$ systematic nodes.*

Proof: Let $\mathcal{C}$ be an optimal-access code of length $2m$. Let $\mathcal{D}$ be an extended code of $\mathcal{C}$. By equivalently transforming the coding matrices (see [18]), we can always assume the coding matrices of the parities in $\mathcal{D}$ are

$$\begin{pmatrix} I & \cdots & I & I & \cdots & I \\ A_1 & \cdots & A_{2m} & A_{2m+1} & \cdots & A_k \end{pmatrix}.$$

Here the first $2m$ column blocks corresponds to the coding matrices of $\mathcal{C}$. First consider the code $\mathcal{C}$, that is, the first $2m$ nodes. If $\mathcal{C}$ has optimal access, then S_i is the span of $l/2$ standard basis, for $i \in [2m]$. Since there are $2m$ systematic nodes, on average each e_z appears $2m \times \frac{l}{2} \times \frac{1}{l} = m$ times, for $z \in [0, l-1]$. We claim that each e_z appears exactly m times. Otherwise, there exists one e_z that appears in $\{S_i : i \in I\}$, for some $|I| > m, I \subset [2m]$. So $|\cap_{i\in I} S_i| \geq 1$. However, by [18] we know when $|I| > m$, $|\cap_{i\in I} S_i| = 0$. So every e_z, $z \in [0, l-1]$, must appear in m of the S_i's, say $e_z \in S_i$, $\forall i \in J$, $|J| = m, J \subset [2m]$. Again by [18] when $|J| = m$, we have $|\cap_{i\in J} S_i| = 1$, so $\cap_{i\in J} S_i = e_z$. So these m subspaces intersect only on e_z.

Now consider the extended code $\mathcal{D}$. Since every S_i, $i \in J$, is an invariant subspace of A_j, $j \in [2m+1, k]$ by the subspace property, we know their intersection, e_z is also an invariant subspace of A_j. In other words, e_z is an eigenvector of A_j. This result is true for all $z \in [0, l-1]$. Hence, we know the standard basis are all the eigenvectors of A_j, $j \in [2m+1, k]$. Equivalently, A_j are all diagonal. So the last $k-2m$ nodes in $\mathcal{D}$ are optimal update. By [18], there are only m nodes that are all optimal update. So $k \leq 3m$. ■

Next let us discuss about the finite field size of the code. In order to make the code MDS, it is equivalent that we should be able to recover from any two column erasures. In other words, any 1×1 or 2×2 submatrices of the matrix (5) should be invertible. Therefore, all eigenvalues $\lambda_{i,s}$ should be nonzero, $i \in [k], s \in \{0,1\}$. Moreover, the following matrix should be invertible for all $i \neq j$:

$$\begin{bmatrix} I & I \\ A_i & A_j \end{bmatrix}.$$

Or equivalently, $A_i - A_j$ should be invertible.

Let us first look at an example. Suppose $m = 2, i = 1, j = 2$ (see Figure 2), then $A_1 - A_2$ is

$$\begin{bmatrix} \lambda_{1,0}-\lambda_{2,0} & \lambda_{2,1}-\lambda_{2,0} & \lambda_{1,0}-\lambda_{1,1} & 0 \\ 0 & \lambda_{1,0}-\lambda_{2,1} & 0 & \lambda_{1,0}-\lambda_{1,1} \\ 0 & 0 & \lambda_{1,1}-\lambda_{2,0} & \lambda_{2,1}-\lambda_{2,0} \\ 0 & 0 & 0 & \lambda_{1,1}-\lambda_{2,1} \end{bmatrix} \tag{8}$$

We can simply compute the determinant by expanding along the first column and the last row. The remaining 2×2 submatrix in the middle is diagonal:

$$\begin{bmatrix} \lambda_{1,0}-\lambda_{2,1} & 0 \\ 0 & \lambda_{1,1}-\lambda_{2,0} \end{bmatrix} \tag{9}$$

Hence, the determinant $\det(A_1 - A_2)$ is

$$(\lambda_{1,0}-\lambda_{2,0})(\lambda_{1,0}-\lambda_{2,1})(\lambda_{1,1}-\lambda_{2,0})(\lambda_{1,1}-\lambda_{2,1}).$$

For another example, let $m = 2, i = 1, j = 3$, then $A_1 - A_3$ is

$$\begin{bmatrix} \lambda_{1,0}-\lambda_{3,0} & 0 & \lambda_{1,0}-\lambda_{1,1} & 0 \\ 0 & \lambda_{1,0}-\lambda_{3,0} & 0 & \lambda_{1,0}-\lambda_{1,1} \\ \lambda_{3,0}-\lambda_{3,1} & 0 & \lambda_{1,1}-\lambda_{3,1} & 0 \\ 0 & \lambda_{3,0}-\lambda_{3,1} & 0 & \lambda_{1,1}-\lambda_{3,1} \end{bmatrix} \tag{10}$$

Since we can permutate rows and columns of a matrix and not change its rank, the above matrix can be changed into:

$$\begin{bmatrix} \lambda_{1,0}-\lambda_{3,0} & \lambda_{1,0}-\lambda_{1,1} & 0 & 0 \\ \lambda_{3,0}-\lambda_{3,1} & \lambda_{1,1}-\lambda_{3,1} & 0 & 0 \\ 0 & 0 & \lambda_{1,0}-\lambda_{3,0} & \lambda_{1,0}-\lambda_{1,1} \\ 0 & 0 & \lambda_{3,0}-\lambda_{3,1} & \lambda_{1,1}-\lambda_{3,1} \end{bmatrix}. \tag{11}$$

And its determinant is

$$\det(A_1 - A_3) = (\lambda_{1,0}-\lambda_{3,1})^2(\lambda_{3,0}-\lambda_{1,1})^2.$$

Now let us discuss in general the finite field size of the code.

Construction 2 *Let the elements of the code be over $\mathbb{F}_q$, with $q \geq 2m+1$. Let c be a primitive element in $\mathbb{F}_q$ and write $< i >:= i \mod m$. Assign the eigenvalues of the coding matrices to be*

$$\lambda_{i,s} = \begin{cases} c^{<i>+sm}, & i \in [2m] \\ c^{<i>+(1-s)m}, & i \in [2m+1, 3m] \end{cases} \tag{12}$$

If we have an extra systematic column with $A_{3m+1} = I$ (see column $N4$ in Figure 1), we can use a field of size $2m+2$ and simply modify the above construction by

$$\lambda_{i,s} = \begin{cases} c^{<i>+sm+1}, & i \in [2m] \\ c^{<i>+(1-s)m+1}, & i \in [2m+1, 3m] \end{cases}$$

For example, when $m = 1$, the coefficients in Figure 1 are assigned using the above formula, where the field size is 4 and $c = 2$. For another example, if $m = 2$, we can use finite field $\mathbb{F}_5$ and $c = 2$, then assign the eigenvalues to be

$$(\lambda_{1,0}, \ldots, \lambda_{6,0}) = (1, 2, 1, 2, 4, 3),$$
$$(\lambda_{1,1}, \ldots, \lambda_{6,1}) = (4, 3, 4, 3, 1, 2).$$

Theorem 4 *The above construction guarantees that the constructed code is MDS and has optimal repair bandwidth. The finite field size is $q \geq 2m+1$.*

Proof: We claim that if we check any two indices $i \neq j \in [3m]$, then the following conditions are necessary and sufficient for $A_i - A_j$ to be invertible. Assume $r, s \in \{0, 1\}$.

1) $\lambda_{i,s} \neq \lambda_{j,r}$, for any $i \neq j \mod m$.
2) $\lambda_{i,s} \neq \lambda_{j,1-s}$, for $i \in [m], j = i + m$.
3) $\lambda_{i,s} \neq \lambda_{j,s}$, for $i \in [2m], j \in [2m+1, 3m]$, $i = j \mod m$.

If we have an extra systematic column with $A_{3m+1} = I$, then $A_i - I$ is invertible iff

4) $\lambda_{i,s} \neq 1$.

By the proof of Theorem 2 we already know that optimal repair bandwidth is equivalent to

5) $\lambda_{i,0} \neq \lambda_{i,1}$.

It can be easily checked that the above conditions are satisfied by Construction 2. Here we only prove condition 1 for $i, j \in [m]$ and condition 2. The rest cases all follow similar ideas. Without loss of generality we can assume $\{e_i\}$ is standard basis, because the basis will not change the value of $\det(A_i - A_j)$.

When $i, j \in [m]$, $V_{i,0} = Q_i, V_{i,1} = P_{i,1}$, and $V_{j,0} = Q_j, V_{j,1} = P_{i,1}$. So $V_{i,1}, V_{j,1}$ share the same eigenvectors $B = \{e_a : a_i = a_j = 1\}$. If we view each element in B as an integer in $[0, 2^m - 1]$ (each vector in B is the binary representation of an integer), we can say A_i, A_j both have only one nonzero element in each row in B. On the other hand, columns of V_i^{-1}, V_j^{-1} correspond to the right eigenvectors of A_i, A_j, respectively. And it is easy to show that they share the right eigenvectors $C = \{e_a^T : a_i = a_j = 0\}$, where the superscript T means transpose. Hence, A_i, A_j both have only one nonzero element in each column in C. To compute the determinant of $A_i - A_j$, we can expand along rows B and columns C. The remaining submatrix will be diagonal since we already eliminated all the non-diagonal elements. Then it is easy to verify condition 1. See (8)(9) for an example.

When $i \in [m], j = i + m$, $V_{i,0} = Q_i, V_{i,1} = P_{i,1}$ and $V_{j,0} = P_{i,0}, V_{j,1} = Q_i$. Therefore both A_i, A_j have nonzero elements at the diagonal locations. Also A_i has nonzero elements at row $P_{i,0}$ and column $P_{i,1}$. Similarly A_j has nonzero elements at row $P_{i,1}$ and column $P_{i,0}$. Let $a = (0, \ldots, 0, 1, 0, \ldots, 0)$ be a binary vector of length m and the only '1' is at location i. And let us view e_0, e_a as the corresponding integers $0, 2^{m-i}$. Then we can see that rows $\{e_0, e_a\}$ and columns $\{e_0, e_a\}$ have only four nonzero elements. We can permutate the rows/columns of a matrix and not change its rank. Therefore move these two rows/columns to rows/columns $0, 1$, and we get a block diagonal matrix. Following the same procedure, we will get block diagonal matrix, where each block is of size 2×2. And the determinant is simple to compute. See (10)(11) for an example. ∎

We can see that the field size q is about $2/3$ of the number of systematic nodes and is not a constant. Also the code has

parameters $(n = 3m+2, k = 3m, l = 2^m)$. On the other hand, the $(n = m+3, k = m+1, l = 2^m)$ code in [17] has constant field of size $q = 3$. So the proposed code has longer k but longer (actual) column length $l \log q$ as well. Nonetheless, it may be possible to alter the structure of A_i's a bit (for example, do not require A_i to be diagonalizable) and obtain a constant field size. And this will be one of our future work directions.

IV. Codes with Arbitrary Number of Parities

In this section, we will give constructions of codes with arbitrary number of parity nodes. Our code will have $l = r^m$ rows, $k = (r+1)m$ systematic nodes, and r parity nodes, for any $r \geq 2$, $m \geq 1$.

Suppose $A_{s,i}$ is the coding matrix for parity node $k+s$ and information node i. From Section II, we assume $A_{1,i} = I$ for all i. In our construction, we are going to add the following assumptions. Every $A_{s,i}$ has r distinct eigenvalues, each corresponding to $l/r = r^{m-1}$ linearly independent eigenvectors, for $s \in [2, r]$. Moreover, given an information node $i \in [k]$, all matrices $A_{s,i}$, $s \in [2, r]$, share the same eigenspaces $V_{i,0}, V_{i,1}, \ldots, V_{i,r-1}$. If these eigenspaces correspond to eigenvalues $\lambda_{i,0}, \lambda_{i,1}, \ldots, \lambda_{i,r-1}$ for $A_{2,i}$, then we assume they correspond to eigenvalues $\lambda_{i,0}^{s-1}, \lambda_{i,1}^{s-1}, \ldots, \lambda_{i,r-1}^{s-1}$ for $A_{s,i}$. By abuse of notations, $V_{i,u}$ represents both the eigenspace and the $l/r \times l$ matrix containing l/r independent eigenvectors. Under these assumptions, it is easy to see that if we write $A_{s,i}$ as

$$\begin{pmatrix} V_{i,0} \\ \vdots \\ V_{i,r-1} \end{pmatrix}^{-1} \begin{pmatrix} \lambda_{i,0}^{s-1} I & & \\ & \ddots & \\ & & \lambda_{i,r-1}^{s-1} I \end{pmatrix} \begin{pmatrix} V_{i,0} \\ \vdots \\ V_{i,r-1} \end{pmatrix},$$

where the identity matrices are of size $\frac{l}{r} \times \frac{l}{r}$, then $A_{s,i} = A_{2,i}^{s-1}$, for all $s \in [r]$. Hence, we are going to write $A_i = A_{2,i}$, thus $A_{s,i} = A_i^{s-1}$, and our construction will only focus on the matrix A_i. As a result, the **subspace property** becomes

$$S_i = S_i A_j, \forall j \neq i, j \in [k] \quad (13)$$

$$S_i + S_i A_i + S_i A_i^2 + \cdots + S_i A_i^{r-1} = \mathbb{F}^l \quad (14)$$

Note that such choice of eigenvalues is not the unique way to construct the matrices, but it guarantees that the code has optimal repair bandwidth. Also, when the finite field size is large enough, we can find appropriate values of $\lambda_{i,u}$'s such that the code is MDS. At last, since each $V_{i,u}$ has dimension l/r and corresponds to l/r independent eigenvectors, we know that any vector in the subspace $V_{i,u}$ is an eigenvector of A_i.

Let $\{e_0, e_1, \ldots, e_{r^m-1}\}$ be the standard basis of $\mathbb{F}^l$. And we are going to use the r-ary expansion to represent the index of a base. An index $a \in [0, r^m - 1]$ is written as $a = (a_1, a_2, \ldots, a_m)$, where a_i is its i-th digit. For example, when $r = 3, m = 4$, we have $e_5 = e_{(0,0,1,2)}$. Define for $i \in [k], u \in [0, r-1]$ the following sets of vectors:

$$\begin{aligned} P_{i,u} &= \{e_a : a_i = u\}, \\ Q_i &= \{\sum_{a_i=0}^{r-1} e_a : a_j \in [0, r-1], j \neq i\}. \end{aligned}$$

i	$P_{i,0}$	$P_{i,1}$	$P_{i,2}$	Q_i
1	e_0	e_3	e_6	$e_0+e_3+e_6$
	e_1	e_4	e_7	$e_1+e_4+e_7$
	e_2	e_5	e_8	$e_2+e_5+e_8$
2	e_0	e_1	e_2	$e_0+e_1+e_2$
	e_3	e_4	e_5	$e_3+e_4+e_5$
	e_6	e_7	e_8	$e_6+e_7+e_8$

Figure 3. Sets of vectors used to construct a code with $r = 3$ parities and column length $l = 3^2 = 9$.

So $P_{i,u}$ is the set of bases whose index is u in the i-th digit. The sum in Q_i is over all e_a such that the j-th digit of a is some fixed value for all $j \neq i$, and the i-th digit varies in $[0, r-1]$. In other words, a vector in Q_i is the summation of the corresponding bases in $P_{i,u}$, $\forall u$. For example, when $r = 3, m = 2$, $P_{1,0} = \{e_{(0,0)}, e_{(0,1)}, e_{(0,2)}\} = \{e_0, e_1, e_2\}$, $P_{1,1} = \{e_3, e_4, e_5\}$, $P_{1,2} = \{e_6, e_7, e_8\}$, and $Q_1 = \{e_0 + e_3 + e_6, e_1 + e_4 + e_7, e_2 + e_5 + e_8\}$.

Notations: If $a = (a_1, a_2, \ldots, a_m)$ is an r-ary vector, denote by $a_i(u) = (a_1, \ldots, a_{i-1}, u, a_{i+1}, \ldots, a_m)$ the vector that is the same as a except digit i, $u \in [0, r-1]$. In the following, all of the subscript i for sets $P_{i,u}, Q_i$ and for digit a_i are computed modulo m. For example, if $i \in [tm+1, (t+1)m]$ for some integer t, then $Q_i := Q_{i-tm}$.

Construction 3 *The $(n = (r+1)m + r, k = (r+1)m, l = r^m)$ code is constructed as follows. For information node $i \in [tm+1, (t+1)m]$, $t \in [0, r-1]$, the u-th eigenspace ($u \in [0, r-1]$) of coding matrix A_i and the rebuilding subspace S_i are defined as*

$$\begin{aligned} V_{i,u} &= \text{span}(P_{i,u}), \forall u \neq t, \\ V_{i,t} &= \text{span}(Q_i), \\ S_i &= \text{span}(P_{i,t}). \end{aligned}$$

For information node $i \in [rm+1, (r+1)m]$, the eigenspaces and rebuilding subspaces are

$$\begin{aligned} V_{i,u} &= \text{span}(P_{i,u}), \forall u \in [0, r-1] \\ S_i &= \text{span}(Q_i). \end{aligned}$$

Example 5 *Figure 3 illustrated the subspaces $P_{i,u}, Q_i$ for $r = 3$ parities and column length $l = 9$. Figure 4 is a code constructed from these subspaces and has 8 systematic nodes. One can see that if a node is erased, one can transmit only a subspace of dimension 3 to rebuild, which corresponds to only $1/3$ repair bandwidth fraction. The three coding matrices for systematic node i are I, A_i, A_i^2, for $i \in [8]$.*

The following theorem shows that the code indeed has optimal repair bandwidth $1/r$.

Theorem 6 *Construction 3 has optimal repair bandwidth $1/r$ when rebuilding one systematic node.*

Proof: By symmetry of the construction, we are only going to show that the subspace property (13)(14) is satisfied for $i \in [1, m] \cup [rm+1, (r+1)m]$. Also $S_i A_j = S_i$ implies that S_i has a basis that are all eigenvectors of A_j.

i	1	2	3	4	5	6	7	8
$V_{i,0}$	Q_1	Q_2	$P_{1,0}$	$P_{2,0}$	$P_{1,0}$	$P_{2,0}$	$P_{1,0}$	$P_{2,0}$
$V_{i,1}$	$P_{1,1}$	$P_{2,1}$	Q_1	Q_2	$P_{1,1}$	$P_{2,1}$	$P_{1,1}$	$P_{2,1}$
$V_{i,2}$	$P_{1,2}$	$P_{2,2}$	$P_{1,2}$	$P_{2,2}$	Q_1	Q_2	$P_{1,2}$	$P_{2,2}$
S_i	$P_{1,0}$	$P_{2,0}$	$P_{1,1}$	$P_{2,1}$	$P_{1,2}$	$P_{2,2}$	Q_1	Q_2

Figure 4. An $(n = 11, k = 8, l = 9)$ code. Sets $P_{i,u}$ and Q_i are listed in Figure 3. $V_{i,u}$ is the u-th eigenspace of the coding matrix A_i. S_i is the subspace used to rebuild systematic node i.

Case 1: $i \in [1, m]$. Before we begin to explore the different cases, let us define the following sets of vectors

$$\begin{aligned} B_u &= \{e_a : a_i = 0, a_j = u\}, u \in [0, r-1], \\ C_t &= \{\sum_{a_j=0}^{r-1} e_a : a_i = 0, a_z \in [0, r-1], z \neq i, j\}. \end{aligned}$$

In the definition of C_t, the sum is over all e_a such that the i-th digit of a is 0, the the z-th digit is some fixed value, $z \neq i, j$, and the j-th digit varies in $[0, r-1]$. Then one can see that

$$B_u \subset P_{j,u}, C_t \subset Q_j.$$

- $j \in [tm+1, (t+1)m]$, for some $t \in [0, r-1]$ and $j - tm \neq i$. Then the eigenspaces of A_j are $V_{j,u} = \text{span}(P_{j,u})$, $u \neq t$, and $V_{j,t} = \text{span}(Q_j)$. Then it is clear that $S_i = \text{span}(P_{i,0}) = \text{span}(\{B_u : u \neq t\} \cup C_t)$. Also every vector of $B_u, u \neq t$ and C_t is an eigenvector of A_j.
- $j \in [rm+1, (r+1)m]$, $j - rm \neq i$. The eigenspaces of A_j are $V_{j,u} = \text{span}(P_{j,u})$, $u \in [0, r-1]$. And $S_i = \text{span}(P_{i,0}) = \text{span}\{B_u : u \in [0, r-1]\}$ and every vector in $B_u, \forall u$ is an eigenvector of A_j.
- $j - tm = i$, $t \in [1, r]$. Then the first eigenspace of A_j is $V_{j,0} = \text{span}(P_{i,0}) = S_i$.
- $j = i$. In this case we want to check (14) in the subspace property. Suppose the distinct eigenvalues of A_i are $\lambda_0, \lambda_1, \ldots, \lambda_{r-1}$. Then the eigenvalues for A_i^s will be $\lambda_0^s, \lambda_1^s, \ldots, \lambda_{r-1}^s$, for $s \in [0, r-1]$. Notice that $S_i = \text{span}(P_{i,0}) = \text{span}\{e_{a_i(0)} : \forall a \in \mathbb{Z}_r^m\}$ and

$$\begin{aligned} & e_{a_i(0)} A_i^s \\ = & (\sum_{u=0}^{r-1} e_{a_i(u)} - e_{a_i(1)} - \cdots - e_{a_i(r-1)}) A_i \\ = & \lambda_0^s \sum_{u=0}^{r-1} e_{a_i(u)} - \lambda_1^s e_{a_i(1)} - \cdots - \lambda_{r-1}^s e_{a_i(r-1)} \\ = & \lambda_0^s e_{a_i(0)} + \sum_{u=1}^{r-1} (\lambda_0^s - \lambda_u^s) e_{a_i(u)}. \end{aligned}$$

Writing the equations for all $s \in [0, r-1]$ in a matrix, we get

$$\begin{pmatrix} e_{a_i(0)} \\ e_{a_i(0)} A_i \\ e_{a_i(0)} A_i^2 \\ \vdots \\ e_{a_i(0)} A_i^{r-1} \end{pmatrix} = M \begin{pmatrix} e_{a_i(0)} \\ e_{a_{i(1)}} \\ \vdots \\ e_{a_i(r-1)} \end{pmatrix},$$

with

$$M = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \lambda_0 & \lambda_0 - \lambda_1 & \cdots & \lambda_0 - \lambda_{r-1} \\ \lambda_0^2 & \lambda_0^2 - \lambda_1^2 & \cdots & \lambda_0^2 - \lambda_{r-1}^2 \\ \vdots & \vdots & & \vdots \\ \lambda_0^{r-1} & \lambda_0^{r-1} - \lambda_1^{r-1} & \cdots & \lambda_0^{r-1} - \lambda_{r-1}^{r-1} \end{pmatrix}.$$

After a sequence of elementary column operations, M becomes the following Vandermonde matrix

$$M' = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_0 & \lambda_1 & \cdots & \lambda_{r-1} \\ \lambda_0^2 & \lambda_1^2 & \cdots & \lambda_{r-1}^2 \\ \vdots & \vdots & & \vdots \\ \lambda_0^{r-1} & \lambda_1^{r-1} & \cdots & \lambda_{r-1}^{r-1} \end{pmatrix}.$$

Since λ_i's are distinct, we know M' and hence M is non-singular. Therefore, $\text{span}\{e_{a_i(0)}, e_{a_i(0)} A_i, \ldots, e_{a_i(0)} A_i^{r-1}\} = \text{span}\{e_{a_i(0)}, e_{a_i(1)}, \ldots, e_{a_i(r-1)}\}$. Since S_i contains $e_{a_i(0)}$ for all r-ary vector a, we know $S_i + S_i A_i + \cdots + S_i A_i^{r-1} = \mathbb{F}^l$.

Case 2: $i \in [rm, (r+1)m]$. Again, we first define some sets of vectors to help with our arguments.

$$\begin{aligned} B'_u &= \{\sum_{a_i=0}^{r-1} e_a : a_j = u, a_z \in [0, r-1], z \neq i, j\} \\ C'_t &= \{\sum_{a_i=0}^{r-1} \sum_{a_j=0}^{r-1} e_a : a_z \in [0, r-1], z \neq i, j\}. \end{aligned}$$

Here the sum in B'_u has fixed values of $a_j = u$ and a_z, $z \neq i, j$, and the i-th digit varies in $[0, r-1]$. The sum in C'_t has fixed values of a_z, $z \neq i, j$, and the i-th and j-th digit both vary in $[0, r-1]$. Then one can check that

$$B'_u \subset \text{span}(P_{j,u}), C'_t \subset \text{span}(Q_j).$$

- $j \in [tm+1, (t+1)m]$, $t \in [0, r-1]$, and $j - tm \neq i - rm$. The eigenspaces of A_j are $\text{span}(P_{j,u})$, $u \neq t$ and $\text{span}(Q_j)$. And $S_i = \text{span}(Q_i) = \text{span}(\{B'_u : u \neq t\} \cup C'_u)$. We can see that every vector in $B'_u, u \neq t$ and C'_t is an eigenvector of A_j.
- $j \in [rm+1, (r+1)m]$, $j \neq i$. The eigenspaces of A_j are $P_{j,u}$, $u \in [0, r-1]$. And $S_i = \text{span}(Q_i) = \text{span}\{B'_u : u \in [0, r-1]\}$. We can see that every vector of B'_u is an eigenvector of A_j.
- $j - tm = i - rm$, $t \in [0, r-1]$. Then the t-th eigenspace of A_j is $\text{span}(Q_i)$, which is equal to S_i.

- $j = i$. Take $\sum_{u=0}^{r-1} e_{a_i(u)} \in S_i$ for arbitrary a, then

$$\sum_{u=0}^{r-1} e_{a_i(u)} A_i^s = \sum_{u=0}^{r-1} \lambda_u^s e_{a_i(u)}.$$

Written in a matrix form, we have

$$\begin{pmatrix} e_{a_i(0)} \\ e_{a_i(0)}A_i \\ e_{a_i(0)}A_i^2 \\ \vdots \\ e_{a_i(0)}A_i^{r-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_0 & \lambda_1 & \cdots & \lambda_{r-1} \\ \lambda_0^2 & \lambda_1^2 & \cdots & \lambda_{r-1}^2 \\ \vdots & \vdots & & \vdots \\ \lambda_0^{r-1} & \lambda_1^{r-1} & \cdots & \lambda_{r-1}^{r-1} \end{pmatrix} \begin{pmatrix} e_{a_i(0)} \\ e_{a_{i(1)}} \\ \vdots \\ e_{a_i(r-1)} \end{pmatrix}.$$

So similar to Case 1, we know $S_i + S_iA_i + \ldots S_iA_i^{r-1}$ spans the entire space $\mathbb{F}^l$.

■

Again, this construction can be shortened to an optimal-access code of length rm [5] and an optimal-update code of length m [3], [10], [16].

The finite field size of this code can be bounded by the following theorem. In the following, we do not assume that the eigenvalue of $A_{s,i}$ is the s-th power of $A_{2,i}$, and $A_{1,i}$ is not necessarily identity. Hence, we only assume that $A_{1,i}, \ldots, A_{r,i}$ share the same eigenspaces for all i.

Theorem 7 *A finite field of size $k^{r-1}r^{m-1}+1$ suffices for the code to be MDS and optimal repair bandwidth. Here $k = (r+1)m$.*

Proof: Let $\{\lambda_{i,j}^{(s)}\}$ be the j-th eigenvalue of $A_{s,i}$, $i \in [k], j \in [0, r-1], s \in [r]$. In order to show that the code is MDS, we need to check if all $x \times x$ submatrices of the following matrix are invertible, for all $x \in [1, r]$.

$$\begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,k} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,k} \\ \vdots & \vdots & & \vdots \\ A_{r,1} & A_{r,2} & \cdots & A_{r,k} \end{bmatrix}$$

Note that each $A_{s,i}$ can be written as $V_i^{-1}\Lambda_{s,i}V_i$ for some diagonal matrix $\Lambda_{s,i}$, where the rows of V_i are eigenvectors and the diagonal of $\Lambda_{s,i}$ are eigenvalues. Since $A_{1,i}, \ldots, A_{r,i}$ share the same eigenvectors V_i, we can multiply V_i^{-1} on the right of the i-th block column, and not change the rank of the above matrix:

$$M = \begin{bmatrix} V_1^{-1}\Lambda_{1,1} & V_2^{-1}\Lambda_{1,2} & \cdots & V_k^{-1}\Lambda_{1,k} \\ V_1^{-1}\Lambda_{2,1} & V_2^{-1}\Lambda_{2,2} & \cdots & V_k^{-1}\Lambda_{2,k} \\ \vdots & \vdots & & \vdots \\ V_1^{-1}\Lambda_{r,1} & V_2^{-1}\Lambda_{r,2} & \cdots & V_k^{-1}\Lambda_{r,k} \end{bmatrix}.$$

Here all $\lambda_{i,j}^{(s)}$ are unknowns in the finite field $\mathbb{F}_q$. We are going to show that if we write the determinants of each $x \times x$ submatrix as a polynomial, and take the product of all these polynomials, then it is an nonzero polynomial. Moreover, by Combinatorial Nullstellensatz [1] we can find assignments of the unknowns over a large enough finite field, such that this polynomial is not zero. Then we are guaranteed to have all the $x \times x$ submatrices invertible. In [1] it is proved that if the degree of a polynomial $f(x_1, \ldots, x_s)$ is $\deg(f) = \sum_{i=1}^s t_i$, and the coefficient of $\prod_{i=1}^s x_i^{t_i}$ is nonzero, then a finite field of size $\max_i\{t_i\}$ is sufficient for an assignment $c_1, \ldots, c_s$ such that $f(c_1, \ldots, c_s) \neq 0$.

By the symmetry of the $\lambda_{i,j}^{(s)}$, we consider only the degree of $\lambda := \lambda_{1,0}^{(1)}$. We will find its maximum degree in the polynomial of determinants. This unknown variable only appears in the matrix

$$\Lambda_{1,1} = \begin{bmatrix} \lambda_{1,0}^{(1)} I & & \\ & \ddots & \\ & & \lambda_{1,r-1}^{(1)} I \end{bmatrix},$$

where I is the identity matrix of size $r^{m-1} \times r^{m-1}$. Let $B = V_1^{-1}\Lambda_{1,1}$. Then we know that λ appears only in the first r^{m-1} columns of B. For the determinant of any $x \times x$ submatrix of M, only the ones containing B needs to be considered, because we are only interested in the degree of λ. Therefore, there are $\binom{k-1}{x-1}\binom{r-1}{x-1}$ submatrices of size $x \times x$ that has λ in its determinant, and its degree is r^{m-1} for each submatrix. So the total degree of λ is

$$r^{m-1} \sum_{x=1}^{r} \binom{k-1}{x-1}\binom{r-1}{x-1}.$$

Moreover, we know from the proof of Theorem 6 that optimal repair bandwidth is achieved for the first systematic node iff the following matrix is invertible

$$\begin{pmatrix} \lambda_{1,0}^{(1)} & \cdots & \lambda_{1,r-1}^{(1)} \\ \vdots & & \vdots \\ \lambda_{1,0}^{(r)} & \cdots & \lambda_{1,r-1}^{(r)} \end{pmatrix}$$

Hence, we need to multiply its determinant to our polynomial. The total degree of λ is

$$\begin{aligned} & 1 + r^{m-1} \sum_{x=1}^{r} \binom{k-1}{x-1}\binom{r-1}{x-1} \\ = \; & 1 + r^{m-1} \sum_{x=0}^{r-1} \binom{k-1}{x}\binom{r-1}{x} \\ < \; & 1 + r^{m-1} \sum_{x=0}^{r-1} (k-1)^x \binom{r-1}{x} \\ = \; & 1 + r^{m-1}k^{r-1}. \end{aligned}$$

Hence the proof is completed. ■

We can see that in the above theorem, for high-rate codes the field size is expositional in the number of systematic nodes.

But we believe that there is still a large space to improve this bound.

V. Conclusions

In this paper, we presented a family of codes with parameters $(n = (r+1)m + r, k = (r+1)m, l = r^m)$ and they are so far the longest high-rate MDS code with optimal repair. The codes were constructed using eigenspaces of the coding matrices, such that they satisfy the subspace property. This property gives more insights on the structure of the codes, and simplifies the proof of optimal repair.

If we require that the code rate approaches 1, i.e., r being a constant and m goes to infinity, then the column length l is *exponential* in the code length k. However, if we require the code rate to be roughly a constant fraction, i.e., m being a constant and r goes to infinity, then l is *polynomial* in k. Therefore, depending on the application, we can see a tradeoff between the code rate and the code length.

It is still an open problem what is the longest optimal-repair code one can build given the column length l. Also, the bound of the finite field size used for the codes may not be tight enough. Unlike the constructions in this paper, the field size may be reduced when we assume that the coding matrices do not have eigenvalues or eigenvectors (are not diagonalizable). These are our future work directions.

References

[1] N. Alon, "Combinatorial nullstellensatz," *Combinatorics Probability and Computing*, vol. 8, no. 1-2, pp. 7–29, Jan 1999.

[2] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[3] V. R. Cadambe, C. Huang, and J. Li,"Permutation code: optimal exact-repair of a single failed node in MDS code based distributed storage systems," in *ISIT*, 2011.

[4] V. R. Cadambe, C. Huang, S. A. Jafar, and J. Li, "Optimal repair of MDS codes in distributed storage via subspace interference alignment", Tech. Rep. arXiv:1106.1250, 2011.

[5] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra," Polynomial length MDS codes with optimal repair in distributed storage systems", in *Proceedings of 45th Asilomar Conference on Signals Systems and Computing*, Nov 2011.

[6] V. R. Cadambe, S. A. Jafar, and H. Maleki, "Minimum repair bandwidth for exact regeneration in distributed storage," in *Wireless Network Coding Conference (WiNC)*, Jun 2010.

[7] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *Proc. of the 3rd USENIX Symposium on File and Storage Technologies (FAST 04)*, 2004.

[8] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[9] D. S. Papailiopoulos, and A.G. Dimakis, "Distributed storage codes through Hadamard designs, " in *ISIT*, 2011.

[10] D. S. Papailiopoulos, A.G. Dimakis, and V. R. Cadambe, " Repair Optimal Erasure Codes through Hadamard Designs," in *Allerton Conference on Control, Computing, and Communication, Urbana-Champaign, IL*, 2011.

[11] J. S. Plank, "The RAID-6 liberation codes," *The International Journal of High Performance Computing and Applications*, vol. 23, pp. 242-251, Aug. 2009.

[12] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage," in *Allerton Conference on Control, Computing, and Communication, Urbana-Champaign, IL*, 2009.

[13] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: necessity and code constructions," *IEEE Trans. on Information Theory*, vol 56, no. 4, pp. 2134–2158, 2012.

[14] C. Suh and K. Ramchandran, "Exact-Repair MDS Code Construction Using Interference Alignment," *IEEE Trans. on Information Theory*, vol 57, no. 3, pp. 1425–1442, 2011.

[15] C. Suh and K. Ramchandran, "On the existence of optimal exact-repair MDS codes for distributed storage," Tech. Rep. arXiv:1004.4663, 2010.

[16] I. Tamo, Z. Wang, and J. Bruck, "MDS array codes with optimal rebuilding," in *ISIT*, 2011.

[17] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," Tech. Rep. arXiv:1112.0371, 2011.

[18] I. Tamo, Z. Wang, and J. Bruck, "Access vs. bandwidth in codes for storage," submitted to *ISIT*, 2012. Available at $http://paradise.caltech.edu/etr.html$.

[19] Z. Wang, I. Tamo, and J. Bruck, " On codes for optimal rebuilding access," in *Allerton Conference on Control, Computing, and Communication, Urbana-Champaign, IL*, 2011.

[20] Y. Wu and A. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *ISIT*, 2009.

[21] Y. Wu, R. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Allerton Conference on Control, Computing, and Communication, Urbana-Champaign, IL*, 2007.

[22] L. Xu, V. Bohossian, J. Bruck, and D. Wagner, "Low-density MDS codes and factors of complete graphs," *IEEE Trans. on Information Theory*, vol. 45, no. 6, pp. 1817–1826, Sep. 1999.

[23] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *IEEE Trans. on Information Theory*, vol. 45, no. 1, pp. 272–276, 1999.