Cyclic Lowest-Density MDS Array Codes

Yuval Cassuto, *Member, IEEE,* and Jehoshua Bruck, *Fellow, IEEE*

***Abstract*— Three new families of lowest-density MDS array codes are constructed, which are cyclic or quasi-cyclic. In addition to their optimal redundancy (Maximum Distance Separable) and optimal update complexity (lowest-density), the symmetry offered by the new codes can be utilized for simplified implementation in storage applications. The proof of the code properties has an indirect structure: first MDS codes that are not cyclic are constructed, and then transformed to cyclic codes by a minimum-distance preserving transformation.**

***Index Terms*— Array Codes, Low-Density Parity-Check Codes, MDS Codes, Cyclic Codes, Systematically-Cyclic Codes**

I. Introduction

MDS (Maximum Distance Separable) codes over large symbol alphabets are ubiquitous in data storage applications. Being MDS, they offer the maximum protection against device failures for a given amount of redundancy. Array codes [2] are one type of such codes that is very useful to dynamic high-speed storage applications as they enjoy low-complexity decoding algorithms over small fields, as well as low update complexity when small changes are applied to the stored content. That is in contrast to the family of Reed-Solomon codes [5, Ch.10] that in general has none of these favorable properties.

A particular array-code subclass of interest is *Lowest Density* array codes, those that have the smallest possible update complexity for their parameters. Since the update complexity dictates the access time to the storage array, even in the absence of failures, this parameter of the code is the primary limiting factor of the code implementation in dynamic storage applications. Examples of constructions that yield lowest-density array-codes can be found in [10],[8],[9],[4],[3]. In this paper we propose lowest-density codes that are also *cyclic* or *quasi-cyclic*. Adding regularity in the form of cyclic symmetry to lowest-density MDS array codes makes their implementation simpler and potentially less costly. The benefit of the cyclic symmetry becomes especially significant when the code is implemented in a distributed way on distinct network nodes. In that case, the use of cyclic codes allows a uniform design of the storage nodes and the interfaces between nodes. The code constructions additionally offer a theoretical value by unveiling more of the rich structure of lowest-density MDS array-codes.

Yuval Cassuto was with the Department of Electrical Engineering, California Institute of Technology, and is now with Hitachi Global Storage Technologies, 3403 Yerba Buena Rd., San Jose, CA 95135, U.S.A. (e-mail: yuval.cassuto@hitachigst.com).

Jehoshua Bruck is with the Department of Electrical Engineering, California Institute of Technology, 1200 E California Blvd., Mail Code 136-93, Pasadena, CA 91125, U.S.A. (e-mail: bruck@paradise.caltech.edu).

This work was supported in part by the Caltech Lee Center for Advanced Networking.

As an example, we examine the following code defined on a 2×6 array. The $+$ signs represent the binary Exclusive-OR operation.

a_0	a_1	a_2	a_3	a_4	a_5
$a_2+a_3+a_4$	$a_3+a_4+a_5$	$a_4+a_5+a_0$	$a_5+a_0+a_1$	$a_0+a_1+a_2$	$a_1+a_2+a_3$

This code has 6 information bits $a_0,\dots,a_5$, and 6 parity bits $a_2+a_3+a_4$, $a_3+a_4+a_5$, $a_4+a_5+a_0$, $a_5+a_0+a_1$, $a_0+a_1+a_2$, $a_1+a_2+a_3$. It is easy to see that all 6 information bits can be recovered from *any* set of 3 columns. For example, if we want to recover a_3,a_4,a_5 from the bits of the left 3 columns, we can proceed by $a_3 = (a_3+a_4+a_5)+(a_4+a_5+a_0)+a_0$, then $a_4 = a_2+(a_2+a_3+a_4)+a_3$, and finally $a_5 = (a_3+a_4+a_5)+a_3+a_4$. Since 3 columns have 6 bits in total, the code is Maximum Distance Separable (MDS). Additionally, the code has lowest-density, since updating an information bit requires 3 parity updates - a trivial lower bound for a code that recovers from any 3 erasures. However, the focus of this paper is a different property of this sample code: its cyclicity. To convince oneself that the code is cyclic, we observe that all the indices in a column can be obtained by adding one (modulo 6) to the indices in the column to its (cyclic) left. Thus any shift of the information bits row results in an identical shift in the parity bits row (and hence the code is closed under cyclic shifts of its columns).

The sample code above, as well as all the codes constructed in the paper, belong to a subclass of cyclic array codes: *systematically-cyclic array-codes*. The Appendix of this article contains characterizations of cyclic array codes in general and systematically-cyclic codes in particular. Codes in the systematically-cyclic subclass enjoy greater implementation benefits relative to the general class of cyclic codes. Properties of cyclic and systematically-cyclic array-codes that imply simpler implementation are provided in section V. In particular, these properties manifest simpler updates and encoding, and more efficient erasure and error decoding.

	array dimensions	r	notes
$\kappa_{1\circ}$	$(p-1)/2 \times (p-1)$	2	
$\kappa_{2\circ}$	$(p-1)/r \times (p-1)$	3,4	2 primitive in F_p
$\kappa_{3\circ}$	$(p-1) \times 2(p-1)$	2	2-quasi-cyclic

TABLE I

SUMMARY OF CYCLIC CODE CONSTRUCTIONS

In sections III and IV, three families of lowest-density, systematically-cyclic (or systematically-quasi-cyclic) MDS array-codes are constructed. The families are named $\kappa_{1\circ}$, $\kappa_{2\circ}$ and $\kappa_{3\circ}$, respectively (the $\circ$ qualifier designates a cyclic or quasi-cyclic code), and their properties are summarized in Table I above. For all primes p, $\kappa_{1\circ}$ provides codes on arrays with dimensions $(p-1)/2 \times (p-1)$ and redundancy $r = 2$, over any Abelian group. For all primes p, such that $r|p-1$ and 2 is primitive in F_p, $\kappa_{2\circ}$, which is a generalization of $\kappa_{1\circ}$, provides codes on arrays with dimensions $(p-1)/r \times (p-1)$ and redundancy $r = 3, 4$, over fields of characteristic 2. $\kappa_{2\circ}$ is the first known family of cyclic lowest density MDS array codes with $r > 2$. Finally, for all primes p, $\kappa_{3\circ}$ provides systematically-quasi-cyclic codes on arrays with dimensions $(p-1) \times 2(p-1)$, over any Abelian group. A specific instance of the family $\kappa_{i\circ}$ will be denoted $\kappa_{i\circ}(p)$, for some prime p. Cyclic codes with the same parameters as $\kappa_{1\circ}$ were proposed in [10], but these are not systematically-cyclic and therefore enjoy only part of the properties $\kappa_{1\circ}$ have. Non-cyclic codes with the same parameters as $\kappa_{2\circ}$ are given in [4]. In addition, the existence of codes with the same parameters as $\kappa_{1\circ}$ and $\kappa_{3\circ}$ was shown in [8]. However, using the suggested combinatorial construction tools of [8] gives non-cyclic codes.

The construction technique we use is first constructing non-cyclic lowest-density MDS codes, and then explicitly providing a transformation to their parity check matrices that results in new, non-equivalent, cyclic codes with the same minimum distance and density. For easier reading, a construction of a sample code precedes the general construction method in section III while the construction of section IV works an example after each step.

II. Definitions

A *linear array code* $\mathcal{C}$ of dimensions $b \times n$ over a field $F = F_q$ is a linear subspace of the vector space F^{nb}. The dual code $\mathcal{C}^{\perp}$ is the null-space of $\mathcal{C}$ over F. To define the minimum distance of an array code we regard it as a code over the alphabet F^b, where F^b denotes length b vectors over F. Then the minimum distance is simply the minimum Hamming distance of the length n code over F^b. Note that though the code symbols can be regarded as elements in the finite field F_{q^b}, we do not assume linearity over this field.

$\mathcal{C}$ can be specified by either its parity-check matrix H of size $N_p \times bn$ or its generator matrix G of size $(bn - N_p) \times bn$, both over F. An array S of size $b \times n$ is a codeword of $\mathcal{C}$ if the length bn column vector $\boldsymbol{\sigma}$, obtained by taking the bits of S column after column, satisfies $H\boldsymbol{\sigma} = \mathbf{0}$, where $\mathbf{0}$ is the length N_p all-zero column vector. From practical considerations, array-codes are required to be *systematic*, namely to have a parity-check (or generator) matrix that is systematic, as now defined.

Definition 1. *A parity-check (or generator) matrix is called [weakly]* ***systematic*** *if it has N_p (or $nb - N_p$), not necessarily adjacent, columns that when stacked together form the identity matrix of order N_p (or $nb - N_p$), respectively.*

Given a systematic H matrix or G matrix (one can be easily obtained from the other), the nb symbols of the $b \times n$ array can be partitioned into N_p parity symbols and $nb - N_p$ information symbols. Define the *density* of the code as the average number of non-zeros in a row of G: $\frac{N(G)}{nb-N_p}$, where $N(M)$ is the number of non-zeros in a matrix M. When H is systematic, an alternative expression for the density is $1 + \frac{N(H)-N_p}{nb-N_p}$. The codes proposed in this paper, all have the lowest possible density, as defined below.

Definition 2. *A code $\mathcal{C}$ is called* ***lowest density*** *if its density equals its minimum distance.*

(the minimum distance is an obvious lower bound on the density [3]). If $b|N_p$ and the minimum distance d equals $\frac{N_p}{b} + 1$, then the code is called Maximum Distance Separable (MDS) with redundancy $r = \frac{N_p}{b}$.

Throughout the paper $[s, t]$ denotes the set $\{x \in \mathbb{Z} : \; s \leqslant x \leqslant t\}$. To simplify the presentation of the constructions in the paper, we introduce another structure that defines a code when, as is the situation here, the parity check matrix has elements in $\{0, 1\}$.

Definition 3. *Given a parity check matrix H of a code $\mathcal{C}$, define the* ***index array*** *$\mathsf{A}_{\mathcal{C}}$ to be a $b \times n$ array of subsets of $[0, N_p - 1]$. The set in location i, j of $\mathsf{A}_{\mathcal{C}}$ contains the elements $\{x : \; h_{i+bj}(x) = 1\}$, where h_l denotes the l^{th} column of H and $h_l(x)$ denotes the x^{th} element of h_l, $x \in [0, N_p - 1]$*

Each set in $\mathsf{A}_{\mathcal{C}}$ represents a column of H. If H is systematic, $\mathsf{A}_{\mathcal{C}}$ has N_p sets of size 1, called singletons. Note that $\mathsf{A}_{\mathcal{C}}$ has the same dimensions as the code array. As an example we take a $(n = 6, b = 3, N_p = 6)$ systematic code and provide in Figure 1 a generator matrix G and a parity check matrix H with its index array $\mathsf{A}_{\mathcal{C}}$.

$$G = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc}
0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&0&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&1&0&0&0&0&0\\
1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&0&0\\
0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0\\
1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right] \updownarrow nb - N_p$$

$$\xleftarrow{\qquad} nb \xrightarrow{\qquad}$$

$$H = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc}
1&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&1\\
0&0&1&1&0&0&0&1&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&1&1&0&0&0&1&0&0&1&0&0&0&1\\
0&0&1&0&0&0&0&0&1&1&0&0&0&1&0&0&1&0\\
0&1&0&0&0&1&0&0&0&0&0&1&1&0&0&0&1&0\\
0&1&0&0&1&0&0&0&1&0&0&0&0&0&1&1&0&0
\end{array}\right] \updownarrow N_p$$

$$\xleftarrow{\qquad} n \xrightarrow{\qquad}$$

$\mathsf{A}_{\mathcal{C}} =$ ($\updownarrow b$)

0	1	2	3	4	5
4,5	5,0	0,1	1,2	2,3	3,4
1,3	2,4	3,5	4,0	5,1	0,2

Figure 1. G,H and $\mathsf{A}_{\mathcal{C}}$ for a sample $n = 6, b = 3, N_p = 6$ code $\mathcal{C}$

III. $\kappa_{1\circ}, \kappa_{2\circ}$: Cyclic Lowest-Density MDS codes with $n = p - 1, b = \frac{p-1}{r}$

The constructions of the code families in this paper specify the index arrays of codes with growing dimensions. For two of the code families - $\kappa_{1\circ}, \kappa_{2\circ}$, the construction uses abstract properties of Finite Fields to obtain index-array sets that guarantee cyclic lowest-density MDS codes for all code dimensions. To better understand the construction method of $\kappa_{1\circ}, \kappa_{2\circ}$, the general construction is preceded by the construction of one particular instance of the family: $\kappa_{1\circ}(7)$.

$\kappa_{1\circ}(7)$ is a cyclic MDS array code with dimensions $b = 3, n = 6$ and redundancy $r = 2$. In the finite field with 7 elements, F_7,[1] pick $\alpha = 6$, an element of multiplicative order $r = 2$. Pick $\beta = 3$, an element with multiplicative order $p - 1 = 6$. Using α and β, F_7 is partitioned into the following sets C_i.

$$C_{-1} = \{0\} \quad , \quad C_0 = \{\beta^0, \beta^0\alpha\} = \{1, 6\},$$
$$C_1 = \{\beta^1, \beta^1\alpha\} = \{3, 4\} \quad , \quad C_2 = \{\beta^2, \beta^2\alpha\} = \{2, 5\}$$

The elements of the sets C_{-1}, C_1, C_2 (C_0 is discarded since it contains the element $p - 1 = 6$) are permuted by the permutation $[0, 1, 2, 3, 4, 5] \xrightarrow{\bar{\psi}} [0, 2, 1, 4, 5, 3]$ and the corresponding sets D_j now follow.

$$D_0 = \bar{\psi}(C_{-1}) = \{0\} \quad ,$$
$$D_1 = \bar{\psi}(C_1) = \{4, 5\} \quad , \quad D_2 = \bar{\psi}(C_2) = \{1, 3\}$$

The sets D_0, D_1, D_2 define the first column of the index array of $\kappa_{1\circ}(7)$. Each of the other 5 columns is obtained by adding 1 modulo 6 to the elements of the sets in the column to its left. The final index array of the code $\kappa_{1\circ}(7)$ is now given.

$$\mathsf{A}_{\kappa_{1\circ}(7)} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 4,5 & 5,0 & 0,1 & 1,2 & 2,3 & 3,4 \\ \hline 1,3 & 2,4 & 3,5 & 4,0 & 5,1 & 0,2 \\ \hline \end{array} .$$

It is left as an exercise to verify that $\kappa_{1\circ}(7)$ is cyclic, lowest-density and MDS.

We now provide the general construction of the code families $\kappa_{1\circ}, \kappa_{2\circ}$.

Let r be a divisor of $p - 1$, and p an odd prime. Let α be an element in F_p of order r and β be an element in F_p of order $p - 1$. The order of an element x in F_p is defined as the smallest non-zero integer i such that $x^i = 1 \pmod{p}$. α and β define a partition of F_p to $\frac{p-1}{r} + 1$ sets. These sets are the $\frac{p-1}{r}$ cosets of the multiplicative subgroup of order r of F_p, plus a set that contains only the zero element. Except for the zero set, all sets are of cardinality r.

$$C_{-1} = \{0\} \qquad\qquad C_i = \{\beta^i, \beta^i\alpha, \ldots, \beta^i\alpha^{r-1}\} \qquad (1)$$

where $0 \leqslant i < \frac{p-1}{r}$. The sets C_i are used in [4] to construct (non-cyclic) lowest density MDS codes with redundancy $r = 3, 4$. The same construction, only with $r = 2$, provides (non-cyclic) lowest density MDS codes by applying the perfect 1-factorization of complete graphs with $p + 1$ vertices by Anderson [1], to the construction of [8]. Shortened versions of the non-cyclic constructions of [8] and [4] are used in the proofs of the constructions of this paper, and are denoted κ_1 and κ_2, respectively. As shown by [4], κ_2 provides lowest density MDS codes for a wide range of parameters. When F has characteristic 2, MDS codes are obtained for $r = 3$ and $r = 4$, whenever 2 is primitive in F_p. For larger characteristics, codes with additional r values were shown to be MDS. For $r = 2$, κ_1 provides MDS codes over any Abelian group [8].

Since $\kappa_{1\circ}, \kappa_{2\circ}$ follow the same construction (only with different r), in the forthcoming discussion we treat them as one family (denoted $\kappa_{1\circ,2\circ}$). Following the presentation of the $\kappa_{1\circ,2\circ}$ construction, we explicitly present the construction for the non-cyclic MDS codes $\kappa_{1,2}$. This is done for the benefit of proving the MDS property of $\kappa_{1\circ,2\circ}$ - through a minimum-distance preserving transformation from the parity-check matrix of $\kappa_{1,2}$ to that of $\kappa_{1\circ,2\circ}$.

Better readability in mind and with a slight abuse of notation, operations on sets denote element-wise operations on the elements of the sets. Specifically, if $\langle x + l \rangle_z$ is used to denote $x + l \pmod{z}$, then $\langle S + l \rangle_z$ denotes the set that is obtained by adding l to the elements of S modulo z. Similarly, permutations and arithmetic operations on sets represent the corresponding operations on their elements.

We now turn to show how the sets C_i of equation (1) are used to construct the cyclic lowest-density MDS codes $\kappa_{1\circ}, \kappa_{2\circ}$. Define $I_0 = \{i : \forall x \in C_i \ , \ 0 \leqslant x < p - 1\}$. I_0 is the set of all indices i, except for the unique index i' for which $C_{i'}$ contains the element $p - 1$. Clearly $|I_0| = \frac{p-1}{r}$. Denote the j^{th} element of I_0 by $I_0(j)$, $j \in [0, \frac{p-1}{r} - 1]$, where indices in I_0 are ordered lexicographically. The permutation $\psi : [0, p-2] \to [0, p-2]$ is defined to be $\psi(x) = \beta^x - 1 \pmod{p}$. We also define the inverse of ψ, $\bar{\psi}(y) = \log_\beta(y + 1)$. The constructing sets D_j are now defined using C_i and the permutation ψ.

$$D_j = \bar{\psi}(C_{I_0(j)}), \text{ for } j \in [0, \frac{p-1}{r} - 1].$$

The construction of $\kappa_{1\circ,2\circ}$ is now provided by specification of the index array $\mathsf{A}_{\kappa_{1\circ,2\circ}}$.

In $\mathsf{A}_{\kappa_{1\circ,2\circ}}$, the set at location

$$(j, l) \in [0, \frac{p-1}{r} - 1] \times [0, p-2]$$

is

$$\langle D_j + l \rangle_{p-1}.$$

The codes $\kappa_{1\circ,2\circ}$ are systematically-cyclic by Definition A6 (in the Appendix) since the top row ($j = 0$) contains sets of size 1, and for every l, translations of the same sets D_j are taken.

As for the codes $\kappa_{1,2}$, for every $0 \leqslant m < p - 1$ define $I_m = \{i : \forall x \in \langle C_i + m \rangle_p \ , \ 0 \leqslant x < p - 1\}$ (I_m is the set of all indices i, except for the unique index i' for which $\langle C_i + m \rangle_p$ contains the element $p - 1$). It is obvious that for every m, $|I_m| = \frac{p-1}{r}$ since for every translation m of the sets C_i, only one set contains the element $p - 1$. Denote the j^{th} element of I_m by $I_m(j)$, $j \in [0, \frac{p-1}{r} - 1]$, where indices in I_m are ordered lexicographically. The code $\kappa_{1,2}$ is defined via an index array $\mathsf{A}_{\kappa_{1,2}}$.

[1] F_p used for the code construction should not be confused with F, the code alphabet

In $\mathsf{A}_{\kappa_{1,2}}$, the set at location

$$(j,m) \in [0, \frac{p-1}{r} - 1] \times [0, p-2]$$

is

$$\langle C_i + m \rangle_p \ , \quad i = I_m(j).$$

Note that because of the restriction $i \in I_m$, $\kappa_{1,2}$ provides *non-*cyclic codes.

The known MDS property of $\kappa_{1,2}$ is next used to prove the MDS property of $\kappa_{1\circ,2\circ}$.

Theorem 4. *$\kappa_{1\circ,2\circ}$ and $\kappa_{1,2}$ have the same redundancy, minimum distance and density.*

Proof: We explicitly show an invertible transformation from $\mathsf{A}_{\kappa_{1\circ,2\circ}}$ to $\mathsf{A}_{\kappa_{1,2}}$ that preserves the code redundancy, density, and minimum distance. To refer to an element x in the set at location (j,l) in an index array $\mathsf{A}_{\mathcal{C}}$, we use the tuple $(x,j,l,\mathcal{C})$. The aforementioned transformation is given by showing that $\mathsf{A}_{\kappa_{1,2}}$ is obtained from $\mathsf{A}_{\kappa_{1\circ,2\circ}}$ by a mapping $(x,j,l,\kappa_{1\circ,2\circ}) \leftrightarrow (\psi(x), j', m, \kappa_{1,2})$. The mapping $x \leftrightarrow \psi(x)$ represents permuting the rows of the parity check matrix and the mapping $(j,l) \leftrightarrow (j',m)$ represents permuting columns of the parity check matrix (which for array codes, in general, does not preserve the minimum distance). As will soon be proved, the mapping $(j,l) \leftrightarrow (j',m)$ has a special property that it only reorders columns of the index array and reorders sets *within* its columns (m is a function of l, independent of j, and j' is a function of both j,l.). Hence, all operations preserve the redundancy of the code, its minimum distance and its density. More concretely, we need to show that for every $l \in [0, p-2]$ there exists an $m \in [0, p-2]$ such that every j has a corresponding $t = I_m(j')$ that together satisfy

$$\psi[\langle D_j + l \rangle_{p-1}] = \langle C_t + m \rangle_p$$

Since $\langle D_0 + l \rangle_{p-1}$ consists of the single element l and $\langle C_{-1} + m \rangle_p$ consists of the single element m, the integers l and m have to satisfy $m = \psi(l)$. Then, for the remainder of the sets ($j > 0$), we rewrite the above condition as

$$\psi[\langle D_j + l \rangle_{p-1}] = \langle C_t + \psi(l) \rangle_p$$

Define $i = I_0(j)$, we can now prove the above statement

$$\psi[\langle D_j + l \rangle_{p-1}] = \psi[\langle \bar{\psi}[C_i] + l \rangle_{p-1}] =$$

$$\langle \beta^{\log_\beta(C_i+1)+l} - 1 \rangle_p = \langle \beta^l C_i + \beta^l - 1 \rangle_p =$$

$$\langle C_{\langle i+l \rangle_{\frac{p-1}{r}}} + \psi(l) \rangle_p$$

and the required transformation is $(x,j,l,\kappa_{1\circ,2\circ}) \ \leftrightarrow \ (\psi(x), j', \psi(l), \kappa_{1,2})$, where j' satisfies $I_{\psi(l)}(j') = \langle I_0(j) + l \rangle_{(p-1)/r}$ for $j > 0$, and $j' = j = 0$ for $j = 0$.

■

A. Example: $\kappa_{1\circ}(7)$ revisited - the transformation from $\kappa_1(7)$

To construct $\kappa_1(7)$, the sets

$$C_{-1} = \{0\}, \ C_0 = \{1,6\}, \ C_1 = \{3,4\}, \ C_2 = \{2,5\}$$

are used by taking the sets $\langle C_i + m \rangle_7$ to be the sets of $\mathsf{A}_{\kappa_1(7)}$ in column m, leaving out the particular set in that column that contains the element 6.

$\mathsf{A}_{\kappa_1(7)} =$

0	1	2	3	4	5
3,4	2,0	3,1	4,2	5,3	1,2
2,5	4,5	4,0	5,1	0,1	0,3

The permutations ψ and $\bar{\psi}$ written explicitly are $[0,1,2,3,4,5] \xrightarrow{\psi} [0,2,1,5,3,4]$ and $[0,1,2,3,4,5] \xrightarrow{\bar{\psi}} [0,2,1,4,5,3]$. $\bar{\psi}$ acting on the array $\mathsf{A}_{\kappa_1(7)}$ yields

$\bar{\psi}(\mathsf{A}_{\kappa_1(7)}) =$

0	2	1	4	5	3
4,5	1,0	4,2	5,1	3,4	2,1
1,3	5,3	5,0	3,2	0,2	0,4

which after reordering of columns and sets within columns results in the systematically-cyclic code $\kappa_{1\circ}(7)$.

$\mathsf{A}_{\kappa_{1\circ}(7)} =$

0	1	2	3	4	5
4,5	5,0	0,1	1,2	2,3	3,4
1,3	2,4	3,5	4,0	5,1	0,2

IV. $\kappa_{3\circ}$: Quasi-Cyclic Lowest-Density MDS Codes with $n = 2(p-1), b = p-1, r = 2$

Before constructing the 2-quasi-cyclic code $\kappa_{3\circ}$, we discuss quasi-cyclic array-codes in general. The definitions and characterizations provided for cyclic array-codes in the Appendix can be generalized to quasi-cyclic array-codes.

Definition 5. *The code $\mathcal{C}$ over F^b is **T-quasi-cyclic** if*

$$s = (s_0, s_1, \ldots, s_{n-2}, s_{n-1}) \in \mathcal{C} \Rightarrow s' = (s_T, s_{T+1}, \ldots, s_{n-1}, s_0, \ldots, s_{T-1}) \in \mathcal{C}$$

and $s_i \in F^b$.

A generalization of Theorem A3 to quasi-cyclic array-codes is now provided.

Theorem 6. *A code $\mathcal{C}$ on $b \times n$ arrays and $N_p = \rho n$, ρ an integer, is T-quasi-cyclic ($n = \lambda T$) if it has a parity check matrix of the form*

$$H = \begin{bmatrix} \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{\lambda-1} \\ \mathbf{Q}_{\lambda-1} & \mathbf{Q}_0 & \cdots & \mathbf{Q}_{\lambda-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_0 \end{bmatrix}$$

where $\mathbf{Q}_i$ are arbitrary matrices of size $T\rho \times Tb$.

Systematically-quasi-cyclic codes are now defined through their index arrays as a generalization of systematically-cyclic codes defined in Definition A6.

Definition 7. *A code $\mathcal{C}$ on $b \times n$ arrays and $N_p = \rho n$, ρ an integer, is **systematically-T-quasi-cyclic** if it has an index array representation $\mathsf{A}_{\mathcal{C}}$, in which N_p of the sets are singletons and adding $T\rho$ to all set elements modulo ρn, results in a T-cyclic shift of $\mathsf{A}_{\mathcal{C}}$.*

A. Construction of the $\kappa_{3_\circ}$ Codes

The code $\kappa_{3_\circ}$ is defined over arrays of size $(p-1)\times 2(p-1)$. Since it is a systematically quasi-cyclic code ($T=2$), we denote the $N_p = 2(p-1)$ parity constraints in the index array $\mathsf{A}_{\kappa_{3_\circ}}$ by $a_0, b_0, a_1, b_1, \ldots, a_{p-2}, b_{p-2}$. The $n = 2(p-1)$ columns of the array will be marked by the same labels. The construction to follow, specifies the contents of "a columns" (a_l) and "b columns" (b_l) of $\mathsf{A}_{\kappa_{3_\circ}}$ separately.

Let p be an odd prime and β be a primitive element in F_p. The permutation $\psi : [0, p-2] \rightarrow [0, p-2]$ is defined, as in section III, to be $\psi(x) = \beta^x - 1 \pmod p$. The inverse permutation $\bar{\psi}$ is then $\bar{\psi}(y) = \log_\beta(y+1)$. For any permutation ϕ, $\phi(a_i)$,$\phi(b_i)$ denote, respectively, $a_{\phi(i)}$, $b_{\phi(i)}$. Also $a_i + l$,$b_i + l$ are used for a_{i+l},b_{i+l}, respectively, and $[a_s, a_t]$,$[b_s, b_t]$ are used for $\{a_s, a_{s+1}, \ldots, a_t\}$ and $\{b_s, b_{s+1}, \ldots, b_t\}$, respectively.

1) a Columns: Define the sets Γ_i, $i \in [0, p-2]$ to be

$$\Gamma_i = \left\{ a_i, b_{\langle i-1\rangle_p} \right\} \tag{2}$$

Define the sets Δ_j, $j \in [1, p-2]$ to be

$$\Delta_j = \left\{ \bar{\psi}(a_j), \bar{\psi}(b_{\langle j-1\rangle_p}) \right\} \tag{3}$$

The a columns of $\mathsf{A}_{\kappa_{3_\circ}}$ are now defined. The set in location $(0, a_l)$, $a_l \in [a_0, a_{p-2}]$ is $\{a_l\}$ and the set in location $(j, a_l) \in [1, p-2] \times [a_0, a_{p-2}]$ is $\langle \Delta_j + l \rangle_{p-1}$.

As an example we write the a columns of $\mathsf{A}_{\kappa_{3_\circ}}(5)$. For $p = 5$ the sets Γ_i are

$$\Gamma_0 = \{a_0, b_4\},\ \Gamma_1 = \{a_1, b_0\},\ \Gamma_2 = \{a_2, b_1\},\ \Gamma_3 = \{a_3, b_2\}$$

For $\beta = 2$, the permutation $\bar{\psi}$ is $[0,1,2,3] \xrightarrow{\bar{\psi}} [0,1,3,2]$. The sets Δ_j, defined through the permutation $\bar{\psi}$, are

$$\Delta_1 = \{a_1, b_0\},\ \Delta_2 = \{a_3, b_1\},\ \Delta_3 = \{a_2, b_3\}$$

Finally, the a columns of $\mathsf{A}_{\kappa_{3_\circ}(5)}$ are provided.

a_0		a_1		a_2		a_3	
a_1, b_0		a_2, b_1		a_3, b_2		a_0, b_3	
a_3, b_1		a_0, b_2		a_1, b_3		a_2, b_0	
a_2, b_3		a_3, b_0		a_0, b_1		a_1, b_2	

2) b Columns: Define the following p sets

$$\begin{aligned} &\{b_0, b_{p-1}\}, \{b_1, b_{p-2}\}, \ldots, \{b_{(p-3)/2}, b_{(p+1)/2}\} \\ &\{a_0, a_{p-1}\}, \{a_1, a_{p-2}\}, \ldots, \{a_{(p-3)/2}, a_{(p+1)/2}\} \\ &\qquad\qquad\qquad\qquad\quad, \{a_{(p-1)/2}, b_{(p-1)/2}\}. \end{aligned}$$

The indices of every set sum to $p-1$. From the sets above define the following $p-1$ sets

$$\begin{aligned} &\{\ \ b_0\ \ \}, \{b_1, b_{p-2}\}, \ldots, \{b_{(p-3)/2}, b_{(p+1)/2}\} \\ &\qquad\quad, \{a_1, a_{p-2}\}, \ldots, \{a_{(p-3)/2}, a_{(p+1)/2}\} \\ &\qquad\qquad\qquad\qquad\quad, \{a_{(p-1)/2}, b_{(p-1)/2}\}. \end{aligned}$$

The element b_{p-1} was removed from the set $\{b_0, b_{p-1}\}$ and the set $\{a_0, a_{p-1}\}$ was removed altogether. After modifying the sets listed above, the resulting sets contain distinct elements from the sets $[a_0, a_{p-2}]$ and $[b_0, b_{p-2}]$. The sets $\nabla_0, \ldots, \nabla_{p-2}$ are obtained by permuting the sets above using $\bar{\psi}$,

$$\begin{aligned} &\{\bar{\psi}(b_0)\}, \{\bar{\psi}(b_1), \bar{\psi}(b_{p-2})\}, \ldots, \{\bar{\psi}(b_{(p-3)/2}), \bar{\psi}(b_{(p+1)/2})\} \\ &\qquad\quad, \{\bar{\psi}(a_1), \bar{\psi}(a_{p-2})\}, \ldots, \{\bar{\psi}(a_{(p-3)/2}), \bar{\psi}(a_{(p+1)/2})\} \\ &\qquad\qquad\qquad\qquad\quad, \{\bar{\psi}(a_{(p-1)/2}), \bar{\psi}(b_{(p-1)/2})\}. \end{aligned}$$

The b columns of $\mathsf{A}_{\kappa_{3_\circ}}$ are now defined. The set in location $(j, b_l) \in [0, p-2] \times [b_0, b_{p-2}]$ is $\langle \nabla_j + l \rangle_{p-1}$.

As an example we write the b columns of $\mathsf{A}_{\kappa_{3_\circ}}(5)$. For $p = 5$, the $p-1$ sets, before operating the $\bar{\psi}$ permutation are

$$\begin{aligned} &\{\ \ b_0\ \ \}, \{b_1, b_3\} \\ &\qquad\quad, \{a_1, a_3\} \\ &\qquad\quad, \{a_2, b_2\}. \end{aligned}$$

After applying the $\bar{\psi}$ permutation, the sets $\nabla_0, \nabla_1, \nabla_2, \nabla_3$ are obtained

$$\begin{aligned} &\{\ \ b_0\ \ \}, \{b_1, b_2\} \\ &\qquad\quad, \{a_1, a_2\} \\ &\qquad\quad, \{a_3, b_3\}. \end{aligned}$$

Finally, the b columns of $\mathsf{A}_{\kappa_{3_\circ}(5)}$ are provided.

	b_0		b_1		b_2		b_3
	b_1, b_2		b_2, b_3		b_3, b_0		b_0, b_1
	a_1, a_2		a_2, a_3		a_3, a_0		a_0, a_1
	a_3, b_3		a_0, b_0		a_1, b_1		a_2, b_2

By mapping the indices $(a_0, b_0, \ldots, a_{p-2}, b_{p-2})$ to the integer indices $(0, 1, \ldots, 2p-3)$, the code $\kappa_{3_\circ}$ clearly satisfies the requirements of Definition 7, hence

Proposition 8. *The code $\kappa_{3_\circ}$ is systematically 2-quasi-cyclic.*

The rest of this section is devoted to proving that $\kappa_{3_\circ}$ is an MDS code.

B. Proof of the MDS Property

To prove the MDS property of the codes $\kappa_{3_\circ}$, a two step proof will be carried out. First we define a different, non-quasi-cyclic code κ_3, and show that it is MDS. Then we show a distance preserving mapping from the rows and columns of the parity-check matrix of κ_3 to those of $\kappa_{3_\circ}$. κ_3 is now defined. The definition only specifies the sets of each column of A_{κ_3}, without specifying the set locations within a column. This definition suffices for the MDS proof and for the mapping provided later. The array dimensions and code parameters of κ_3 are identical to those of $\kappa_{3_\circ}$.

Definition 9. *The columns $a_0, b_0, a_1, b_1, \ldots, a_{p-2}, b_{p-2}$ of the code κ_3 are defined as follows.*
1) An a column $a_l \in [a_0, a_{p-2}]$ of A_{κ_3} contains the set $\{a_l\}$ and all sets $\{a_m, b_{m'}\}$ such that $m - m' = l + 1 \pmod p$. Only the $p-2$ such sets with $(m, m') \in [0, p-2]^2$ are taken.
2) A b column $b_l \in [b_0, b_{p-2}]$ of A_{κ_3} contains the set $\{b_l\}$, the set $\{a_{(l-1)/2}, b_{(l-1)/2}\}$, and all sets $\{a_m, a_{m'}\}$ and $\{b_m, b_{m'}\}$ such that $m + m' = l - 1 \pmod p$. Here too, only the $p-3$ sets with $(m, m') \in [0, p-2]^2$ are taken.

To prove the MDS property of κ_3, we define and use a graphical interpretation of index arrays. This interpretation can be applied when the index array $\mathsf{A}_{\mathcal{C}}$, of a binary parity-check matrix, has only sets of sizes two or less. Given an index array whose union of sets is $\{0, 1, \ldots, \mathrm{R}-1\}$, denote by $\mathsf{K}_{\mathrm{R}+1}$ the complete graph on the $\mathrm{R}+1$ vertices labeled $\{0, 1, \ldots, \mathrm{R}-1, \infty\}$. Each set of size two, $\{x, y\}$, defines a subgraph of $\mathsf{K}_{\mathrm{R}+1}$, called set-subgraph, that has the vertices x, y and an edge connecting them. Each set of size one, $\{x\}$, defines a set-subgraph of $\mathsf{K}_{\mathrm{R}+1}$ that has the vertices x, ∞ and an edge connecting them. A bit [2] assignment to an array corresponds to the union of set-subgraphs in locations with non-zero entries. The following is a simple but useful observation.

Proposition 10. *A bit assignment to an array is a codeword of $\mathcal{C}$ if and only if all vertices have even degrees in its $\mathsf{A}_{\mathcal{C}}$ set-subgraph union (the subgraph is a cycle or a union of edge-disjoint cycles, with possibly some isolated vertices).*

The above graphical interpretation is now explained with an example.

Example 11. *Let the array code $\mathcal{C}$ be defined by the following index array.*

$$\mathsf{A}_{\mathcal{C}} = \begin{array}{|c|c|c|c|c|c|}\hline 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 4,5 & 5,0 & 0,1 & 1,2 & 2,3 & 3,4 \\ \hline 1,3 & 2,4 & 3,5 & 4,0 & 5,1 & 0,2 \\ \hline \end{array}$$

The word

$$V_1 = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

has the set-subgraph union in Figure 2(a). Vertices $4, 5$ have odd degrees of 1, and thus the word V_1 is not a codeword of $\mathcal{C}$. On the other hand, the word

$$V_2 = \begin{array}{|c|c|c|c|c|c|}\hline 0 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

has the set-subgraph union in Figure 2(b). All vertices have even degrees and thus V_2 is a codeword of $\mathcal{C}$.

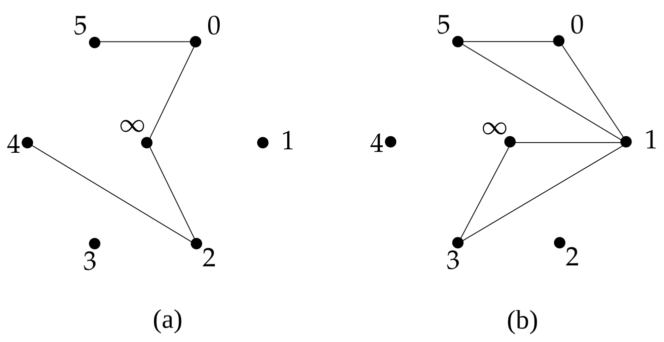

Figure 2. Set-subgraph unions of the code $\mathcal{C}$. (a) for the word V_1. (b) for the codeword V_2.

[2] A similar interpretation works for array symbols from any Abelian group

The next Lemma establishes the MDS property of κ_3 by showing that there are no codewords of column weight smaller than 3.

Lemma 12. *For any two columns from $\{a_0, b_0, a_1, b_1, \ldots, a_{p-2}, b_{p-2}\}$, there are no non-zero codewords of κ_3 that are all zero outside these two columns.*

Proof: For each pair of columns, the proof will show that no subgraph of the set subgraph corresponding to these two columns, can contain a cycle. Hence there are no non-zero codewords with column weight 2 or less. We distinguish between three cases. A similar proof, but for a different combinatorial construct (which does not yield quasi-cyclic codes) appears in [1].

Case 1: Two a columns contain all non-zero locations.
For columns a_l and a_{l+v} such that $0 \leqslant l < l+v \leqslant p-2$, the set-subgraph is given in Figure 3. A solid edge comes from a set in column a_l and a dashed edge comes from a set in column a_{l+v}. Note that the edges satisfy the constraints of 1 in Definition 9. To have a cycle as a subgraph, there must exist

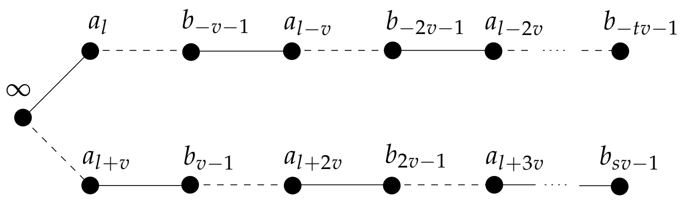

Figure 3. Set-Subgraph of Columns a_l,a_{l+v}

two integers s, t such that $s+t < p$ and either $l - tv \equiv l + sv \pmod{p}$ or $-tv - 1 \equiv sv - 1 \pmod{p}$. The first condition refers to the case when an index of a from the upper chain is identical to an index of a from the lower chain (and thus a cycle is created). The second condition refers to the case when an index of b from the upper chain is identical to an index of b from the lower chain. Each of the conditions requires $(s+t)v \equiv 0 \pmod{p}$, which is a contradiction for a prime p.

Case 2: Two b columns contain all non-zero locations.
For columns b_l and b_{l+v} such that $0 \leqslant l < l+v \leqslant p-2$, the set-subgraph is given in Figure 4. The edges satisfy the constraints of 2 in Definition 9. Cycles with an odd number of edges are not possible since elements appear at most once in every column (any vertex has one solid edge and one dashed edge incident on it). To have a cycle with an even number of edges, the same contradictory conditions of Case 1 apply.

Case 3: One a column and one b column contain all non-zero locations.
Denote the non-zero columns by a_l and b_ℓ. A solid edge comes from a set in column a_l and a dashed edge comes from a set in column b_ℓ. Assume first that the cycle does not contain the edge that corresponds to the special set $\{a_{(\ell-1)/2}, b_{(\ell-1)/2}\}$. Then the number of edges in the cycle is a multiple of 4 (because of the $a \dashrightarrow a \rightarrow b \dashrightarrow b \rightarrow a$ structure), and it has the structure of Figure 5. For each path of length 4 of the pattern $a \dashrightarrow a \rightarrow b \dashrightarrow b \rightarrow a$, the index of the final a vertex is greater by $2l+2$ modulo p than the index of the initial a vertex. Therefore, as seen at the top vertex in Figure 5,

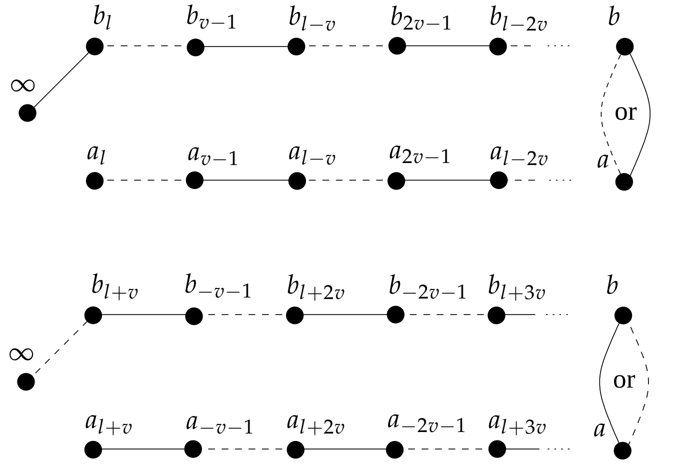

Figure 4. Set-Subgraph of Columns b_l,b_{l+v}

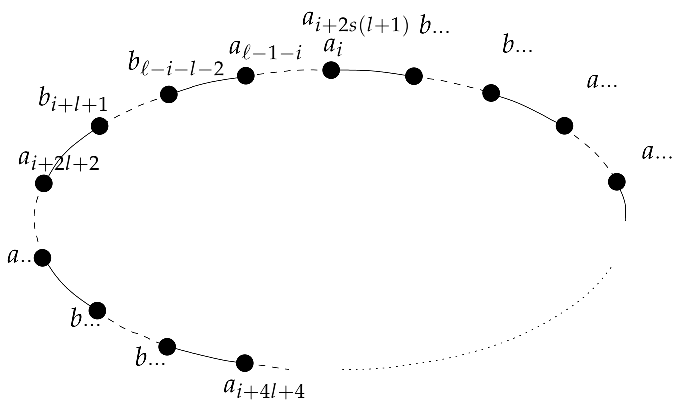

Figure 5. A cycle from Columns a_l,b_ℓ

the existence of such a cycle depends on the condition that $i \equiv i + 2s(l + 1) \pmod p$, for some $s < (p - 1)/2$. This is a contradiction for a prime p and $l < p - 1$. Now assume that there exists a cycle that does contain the edge $\{a_{(\ell-1)/2}, b_{(\ell-1)/2}\}$. In that case there exists a path from a_ℓ to b_{-l-2} (the only two vertices with degree 1), which does *not* include the edge $\{a_{(\ell-1)/2}, b_{(\ell-1)/2}\}$, with the structure of Figure 6. For each path of length 4 of the pattern $b \longrightarrow a \dashrightarrow$

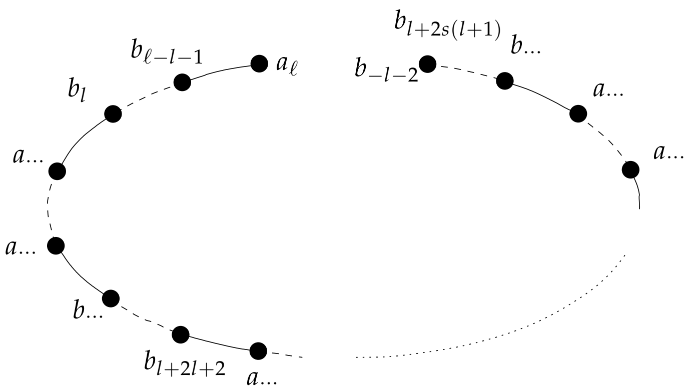

Figure 6. A path from Columns a_l,b_ℓ

$a \longrightarrow b \dashrightarrow b$, the index of the final b vertex is greater by $2l + 2$ modulo p than the index of the initial b vertex. Therefore, as seen at the top right vertex in Figure 6, the existence of such a path depends on the condition that $-l - 2 \equiv l + 2s(l + 1) \pmod p$, or equivalently $2(s + 1)(l + 1) \equiv 0 \pmod p$, for some $s < (p - 1)/2$. This is again a contradiction for a prime p and $l < p - 1$. ■

Lemma 13. $\mathsf{A}_{\kappa_{3\circ}}$ *can be obtained from* A_{κ_3} *by a minimum-distance preserving transformation.*

Proof: We show that by permuting the indices of $\mathsf{A}_{\kappa_{3\circ}}$, its columns and sets within its columns, A_{κ_3} can be obtained. All these operations preserve the redundancy, minimum distance of the code and its density. We provide the transformation and prove its aforementioned property for a and b columns separately.

1) a Columns: Recall that the set in location $(j, a_l) \in [1, p - 2] \times [a_0, a_{p-2}]$ of $\mathsf{A}_{\kappa_{3\circ}}$ is

$$\{\langle\bar{\psi}(a_j) + l\rangle_{p-1}\ ,\ \langle\bar{\psi}(b_{j-1}) + l\rangle_{p-1}\}.$$

To show the transformation we look at the difference between the a index and the b index above

$$\langle\bar{\psi}(j) + l\rangle_{p-1} - \langle\bar{\psi}(j - 1) + l\rangle_{p-1},$$

and permute each summand using ψ to get

$$\psi\left[\langle\bar{\psi}(j) + l\rangle_{p-1}\right] - \psi\left[\langle\bar{\psi}(j - 1) + l\rangle_{p-1}\right] =$$

substituting the permutations $\psi, \bar{\psi}$ we write

$$= \beta^{\log_\beta(j+1)+l} - 1 - \beta^{\log_\beta(j)+l} + 1 =$$

$$= \beta^l(j + 1 - j) = \beta^l - 1 + 1 = \psi(l) + 1.$$

In words, pairs of a, b indices of $\mathsf{A}_{\kappa_{3\circ}}$, after permutation, have the same relation as the pairs of indices of A_{κ_3} (as defined in 1 of Definition 9), with columns permuted by the same permutation. Since all elements in the sets of column l of $\mathsf{A}_{\kappa_{3\circ}}$ are distinct, permuting the indices and columns using ψ results in the same sets that form A_{κ_3}.

2) b Columns: We proceed similarly to the previous case but this time look at the sum

$$\psi\left[\langle\bar{\psi}(j) + l\rangle_{p-1}\right] + \psi\left[\langle\bar{\psi}(p - 1 - j) + l\rangle_{p-1}\right] =$$

and substitute $\psi, \bar{\psi}$ to get

$$= \beta^{\log_\beta(j+1)+l} - 1 + \beta^{\log_\beta(p-j)+l} - 1 =$$

$$= \beta^l(j + 1 - j) - 2 = \beta^l - 1 - 1 = \psi(l) - 1.$$

For b columns too, permuting the indices and columns of $\mathsf{A}_{\kappa_{3\circ}}$ results in the sets of A_{κ_3} (as defined in 2 of Definition 9). ■
Lemma 12 and Lemma 13 together prove the main theorem of the section.

Theorem 14. *For every prime* p, $\kappa_{3\circ}(p)$ *has minimum column distance* 3*, and thus it is an MDS code.*

V. Implementation Benefits of Cyclic and Quasi-Cyclic Array-Codes

Cyclic and Quasi-Cyclic array-codes possess a more regular structure relative to general array codes. Regular structures often simplify the realization of error-correcting codes in complexity limited systems. In particular, when the array code is implemented in a distributed fashion, as practiced in storage and network storage applications, the cyclic symmetry of the codes allows using a single uniform design for all nodes, contrary to non-cyclic codes in which each node needs to perform different operations. Though the exact advantage of cyclic codes depends on the qualities and constraints of particular implementations, we next attempt to motivate their use in general, by illustrating some of their properties. The properties are given for cyclic codes only, but quasi-cyclic codes enjoy similar properties with a slightly reduced symmetry.

A. Encoding and Updates

Property 1 *In a systematically-cyclic array-code (see Definition A4 in the Appendix), if updating an information symbol at array location* (j, l) *requires updating parity symbols at array locations* $\{(j_1, l_1), \ldots, (j_r, l_r)\}$*, then updating an information symbol at array location* $(j, l + s)$ *requires the same parity updates at array locations* $\{(j_1, l_1 + s), \ldots, (j_r, l_r + s)\}$*, where all* $+$ *operations are modulo* n.

This property, established directly from the parity-check matrix structure of systematically-cyclic array-codes, simplifies the circuitry needed for bit updates, an operation that is invoked at a very high rate in a typical dynamic storage application. In cylindrical storage arrays, it also allows to update a group of array symbols without absolute angular synchronization. Cyclic codes that are not systematically cyclic do not enjoy the same property, in general.

B. Syndrome Calculation

The syndrome $\boldsymbol{s}$ of a word R with dimensions $b \times n$ is obtained by first converting it, by column stacking its elements, to a length bn column vector $\boldsymbol{r}$. Then it is defined as $\boldsymbol{s} = H\boldsymbol{r}$. Computing the syndrome is a first step in error and erasure decoding of array codes. A more economic calculation of syndrome symbols is achieved for cyclic array-codes thanks to the following property.

Property 2 *In a cyclic array code, if symbol* i *of the syndrome is a function* f *of the symbols in the following array locations* $f[(j_1, l_1), (j_2, l_2), \ldots]$*, then symbol* $i + \rho s$ *of the syndrome is the function* $f[(j_1, l_1 + s), (j_2, l_2 + s), \ldots]$*, indices taken modulo* n.

C. Erasure and Error Decoding

Property 3 *If in a cyclic array-code, a set of erased columns* $\Lambda = \{i_1, \ldots, i_t\}$ *is recovered by a matrix vector product* $H_\Lambda^{-1}\boldsymbol{s}$*, where* $\boldsymbol{s}$ *is the syndrome of the codeword with missing symbols set to zero, then the set of erased columns* $\Lambda_s = \{i_1 + s, \ldots, i_t + s\}$ *(indices modulo* n*) is recovered by* $H_\Lambda^{-1} U_s \boldsymbol{s}$*, where* U_s *is the sparse matrix that cyclically shifts the syndrome* ρs *locations upward.*

This property relies on the fact that for cyclic codes, $H_{\Lambda_s} = D_s H_\Lambda$, where D_s is the sparse matrix that cyclically shifts the rows of H_Λ, ρs locations downward. Taking the inverse results in $H_{\Lambda_s}^{-1} = H_\Lambda^{-1} D_s^{-1} = H_\Lambda^{-1} U_s$. The benefit of that property is that many of the decoding matrices are cyclically equivalent, and therefore only a $1/n$ portion of decoding matrices needs to be stored, compared to non-cyclic array-codes with the same parameters. A similar advantage exists for error decoding, where the cyclic equivalence of syndromes allows a simpler error location.

VI. Conclusion

Beyond the practical benefit of the constructed cyclic codes, these codes and their relationship to known non-cyclic codes raise interesting theoretical questions. The indirect proof technique used for all three code families is a distinctive property of the code constructions. It is curious that a direct MDS proof of the more structured cyclic codes, seems hard to come by. Such a proof may reveal more about the structure of these codes and possibly allow finding new code families. This optimistic view is supported by computer searches that find cyclic lowest-density MDS codes with parameters that are not covered by the known families of non-cyclic codes.

Appendix: cyclic array codes

The codes constructed in this paper are codes of length n over F^b which are cyclic but *not* linear. In this section we wish to discuss such codes in general, providing conditions for a code to be cyclic. One way to characterize cyclic array codes is as cyclic group codes over the direct-product group of the additive group of F. Another is to view them as length nb linear *b-quasi-cyclic* codes. For the most part, the latter view will prove more useful since the constructions in the paper are not explicit group theoretic ones. In fact, the description of array codes using index arrays we chose here was used in [7] to describe quasi-cyclic code constructions. We start off with the basic definition of cyclic codes.

Definition A1 *The code* $\mathcal{C}$ *over* F^b *is* ***cyclic*** *if*

$$s = (s_0, s_1, \ldots, s_{n-2}, s_{n-1}) \in \mathcal{C} \Rightarrow s' = (s_1, s_2, \ldots, s_{n-1}, s_0) \in \mathcal{C}$$

and $s_i \in F^b$.

Cyclic codes over F^b are related to quasi-cyclic codes over F in the following manner.

Proposition A2 *An array code* $\mathcal{C}$ *of length* n *over* F^b *is cyclic if and only if the code* $\mathcal{C}_{1D}$ *of length* bn *over* F*, that has the same parity check matrix, is quasi-cyclic with basic block length* b.

This equivalence allows us to use the characterization of quasi-cyclic codes from [6, pp.257], to determine the cyclicity of an array-code.

Theorem A3 *A code $\mathcal{C}$ on $b \times n$ arrays and $N_p = \rho n$, ρ an integer, is cyclic if it has a parity check matrix of the form*

$$H = \left[\begin{array}{c|c|c|c} \mathbf{Q}_0 & \mathbf{Q}_1 & \dots & \mathbf{Q}_{n-1} \\ \mathbf{Q}_{n-1} & \mathbf{Q}_0 & \dots & \mathbf{Q}_{n-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \dots & \mathbf{Q}_0 \end{array}\right]$$

where $\mathbf{Q}_i$ are arbitrary matrices of size $\rho \times b$.

Note that if H is *not* required to have full rank of ρn, then Theorem A3 captures the most general cyclic array-codes (the *if* statement can be replaced with an *if and only if* one.). However, there exist cyclic array-codes that do not have full rank matrices H, of the form given above ($H = \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right]$ has the following words as codewords

$$\left\{ \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array}\,,\ \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 0 & 0 \\ \hline \end{array}\,,\ \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 1 & 1 \\ \hline \end{array}\,,\ \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} \right\}$$

and hence it is cyclic. However, there is no 2×4 parity check matrix for this code that admits the structure of $\left[\begin{array}{c|c} \mathbf{Q}_0 & \mathbf{Q}_1 \\ \mathbf{Q}_1 & \mathbf{Q}_0 \end{array}\right]$.
).

A sub-class of the cyclic codes characterized above, *systematically-cyclic array-codes*, is next defined. These are cyclic array codes in which each column has ρ parity symbols, at the same locations for all columns.

Definition A4 *A code $\mathcal{C}$ on $b \times n$ arrays and $N_p = \rho n$, ρ an integer, is* ***systematically-cyclic*** *if it has a parity check matrix of the form*

$$H = \left[\begin{array}{c|c|c|c} \mathbf{Q}_0 & \mathbf{O}|\mathbf{P}_1 & \dots & \mathbf{O}|\mathbf{P}_{n-1} \\ \mathbf{O}|\mathbf{P}_{n-1} & \mathbf{Q}_0 & \dots & \mathbf{O}|\mathbf{P}_{n-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{O}|\mathbf{P}_1 & \mathbf{O}|\mathbf{P}_2 & \dots & \mathbf{Q}_0 \end{array}\right]$$

where $\mathbf{O}$ represents the all-zero matrix of order ρ and $\mathbf{Q}_0$ has the identity matrix of order ρ as a sub-matrix. $\mathbf{P}_i$ are arbitrary matrices of size $\rho \times (b - \rho)$.

An equivalent characterization can be obtained using the index array $\mathsf{A}_{\mathcal{C}}$ of the code $\mathcal{C}$. Corollary A5 to Theorem A3 and Definition A6 provide this characterization.

Corollary A5 *A code $\mathcal{C}$ on $b \times n$ arrays and $N_p = \rho n$, ρ an integer, is cyclic if it has an index array representation $\mathsf{A}_{\mathcal{C}}$, in which adding ρ to all set elements modulo ρn results in a cyclic shift of $\mathsf{A}_{\mathcal{C}}$.*

Definition A6 *A code $\mathcal{C}$ on $b \times n$ arrays and $N_p = \rho n$, ρ an integer, is systematically-cyclic if it has an index array representation $\mathsf{A}_{\mathcal{C}}$, in which N_p of the sets are singletons and adding ρ to all set elements modulo ρn results in a cyclic shift of $\mathsf{A}_{\mathcal{C}}$.*

References

[1] B. Anderson, “Finite topologies and Hamilton paths,” *J. Combinatorial Theory (B)*, vol. 14, pp. 87–93, 1973.

[2] M. Blaum, P. Farrell, and H. van Tilborg, “Array codes,” *Handbook of Coding Theory, V. Pless and W. Huffman (Eds.), Elsevier Science B.V*, 1998.

[3] M. Blaum and R. Roth, “On lowest density MDS codes,” *IEEE-Trans-IT*, vol. 45, no. 1, pp. 46–59, 1999.

[4] E. Louidor and R. Roth, “Lowest-density MDS codes over extension alphabets,” *IEEE-Trans-IT*, vol. 52, no. 7, pp. 3186–3197, 2006.

[5] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[6] W. Peterson and E. Weldon, *Error-Correcting Codes*. Cambridge, MA: MIT Press, 1972.

[7] R. Townsend and E. Weldon, “Self-orthogonal quasi-cyclic codes,” *IEEE-Trans-IT*, vol. 13, no. 2, pp. 183–195, 1967.

[8] L. Xu, V. Bohossian, J. Bruck, and D. Wagner, “Low-density MDS codes and factors of complete graphs,” *IEEE-Trans-IT*, vol. 45, no. 6, pp. 1817–1826, 1999.

[9] L. Xu and J. Bruck, “X-code: MDS array codes with optimal encoding,” *IEEE-Trans-IT*, vol. 45, no. 1, pp. 272–276, 1999.

[10] G. Zaitsev, V. Zinov’ev, and N. Semakov, “Minimum-check-density codes for correcting bytes of errors, erasures, or defects,” *Problems Inform. Transm.*, vol. 19, pp. 197–204, 1981.