A construction of MDS array codes

S. D. Cardell[1] J.-J. Climent[1] & V. Requena[2]
[1] *Departament d'Estadística i Investigació Operativa*
Universitat d'Alacant, Spain
[2] *Departamento de Estadística, Matemática Aplicada e Informática*
Universidad Miguel Hernández de Elche, Spain

Abstract

In this paper a new construction of MDS array codes is introduced. In order to obtain a code with this property, the parity-check matrix is constructed just using a superregular matrix by blocks composed by powers of the companion matrix of a primitive polynomial. Also a decoding algorithm for these codes is introduced.
Keywords: Array code, MDS code, block linear code, finite field, superregular matrix, companion matrix, primitive polynomial

1 Introduction

Array codes are a class of error control codes which have several applications in communication, in storage systems to protect data against erasures, [1, 2, 3] and they have been studied by several authors (see, for example, [1, 4, 5, 6, 7, 8, 9]).

Array codes can be constructed with symbols from a field, a ring or a group, they can have a wide range of parameters (block or constraint length, rate, distance, etc.), and their interest lies in their ability to detect and correct random and/or bursts or clusters of errors. Our motivation to investigate array codes is that they provide a good trade-off between error-control power and complexity of decoding.

These codes are very useful to dynamic high-speed storage applications since they have low-complexity decoding algorithms over small fields and low update complexity when small changes are applied to the stored data [1]. In general, Reed-Solomon codes have none of these properties; thus, they are more efficient than Reed-Solomon codes in computational complexity terms [1, 5]. Furthermore, our goal is to work with maximum distance separable (MDS) codes, those codes whose minimum Hamming distance attains the Singleton bound for a given length and

dimension [10], since they provide the maximum protection against device failure for a given amount of redundancy [11]. It is possible to find some constructions of this kind of codes in [12, 11, 13, 14, 9, 2, 3].

The rest of the paper is organized as follows. In Section 2 we introduce some notation and preliminary results that we need to follow the paper. Moreover, we recall some properties and definitions. In Section 3 we present the construction of an array code using a superregular matrix and the companion matrix of a primitive polynomial which will be part of the parity-check matrix of an MDS array code. We give necessary and sufficient conditions for our codes to be MDS. We also introduce a decoding algorithm in Section 4 for the MDS array codes constructed in Section 3 for the binary case. Finally, we present our main conclusions in Section 5.

2 Preliminaries

Let $\mathbb{F}_q$ be the Galois field of q elements and consider b a positive integer. If $\mathcal{C}$ is a code of length n over $\mathbb{F}_q^b$, we can consider the codewords of $\mathcal{C}$ as codewords of length nb over $\mathbb{F}_q$. Then, a code $\mathcal{C}$ is said to be a **linear array code** (or an $\mathbb{F}_q$**-linear code**) of length n over $\mathbb{F}_q^b$ if it is a linear code of length nb over $\mathbb{F}_q$ (see [14]).

We represent the code $\mathcal{C}$ as $\mathcal{C}_{\mathbb{F}_q^b}$ (respectively, $\mathcal{C}_{\mathbb{F}_q}$) when we consider $\mathcal{C}$ as a code over $\mathbb{F}_q^b$ (respectively, $\mathbb{F}_q$). Note that both $\mathcal{C}_{\mathbb{F}_q}$ and $\mathcal{C}_{\mathbb{F}_q^b}$ refer to the same set of codewords, but considering the alphabets $\mathbb{F}_q$ and $\mathbb{F}_q^b$, respectively. It is worth pointing out that the code symbols of $\mathcal{C}_{\mathbb{F}_q^b}$ can be regarded as elements in the field $\mathbb{F}_{q^b}$. However, linearity over this field is not assumed.

Let $[N, K, D]$ denote the parameters of the code $\mathcal{C}_{\mathbb{F}_q}$ over $\mathbb{F}_q$., i.e.

$$N = \left|\mathcal{C}_{\mathbb{F}_q}\right| = q^K, \quad K = \dim \mathcal{C}_{\mathbb{F}_q}, \quad D = \mathrm{d}\left(\mathcal{C}_{\mathbb{F}_q}\right).$$

Then the Singleton bound (see, for example, [10]) states that

$$D \leq N - K + 1.$$

The linear codes that achieve equality in the Singleton bound are called **maximum distance separable** codes, or MDS codes for short.

The number $k = \log_{q^b}\left|\mathcal{C}_{\mathbb{F}_q^b}\right|$ is called the **normalized dimension** (or just dimension) of $\mathcal{C}_{\mathbb{F}_q^b}$. If b divides K, then $k = K/b$ (in what follows, b divides K). Thus, the parameters of the code $\mathcal{C}_{\mathbb{F}_q^b}$ are $[n, k, d]$ over $\mathbb{F}_q^b$, where d is the minimum distance and $n = N/b$. To define the minimum (Hamming) distance of $\mathcal{C}_{\mathbb{F}_q^b}$, we consider it as a code over the alphabet $\mathbb{F}_q^b$. Then, the distance d is measured respect to the symbols of $\mathbb{F}_q^b$ (see [14]). It is not difficult to see that $d \leq D$ and that

$$d \leq n - k + 1.$$

That is, the Singleton bound also holds for linear array codes. Consequently, we call **MDS linear array codes** the linear array codes that achieve equality in the Singleton bound.

It is worth remembering that the code $\mathbb{F}_q^b$ can be specified by either its parity-check matrix H of size $(n-k)b \times nb$ or its generator matrix G of size $kb \times nb$, both over $\mathbb{F}_q$. From practical considerations, array codes are required to be systematic, that is, its parity-check (or generator) matrix has to be systematic. Recall that the matrix H (respectively, G) is said to be **systematic** if it contains the identity matrix of size $(n-k)b \times (n-k)b$ (respectively, $kb \times kb$).

The following theorem is useful to check whether a linear array code is MDS or not, without computing the minimum distance. We quote it here for completeness.
Theorem 1 (Proposition 3.2 of [14]): *Let $H = \left[A\ I_{(n-k)b}\right]$ be an $(n-k)b \times nb$ systematic parity-check matrix of an $\mathbb{F}_q$-linear code $\mathcal{C}_{\mathbb{F}_q^b}$ with parameters $[n,k]$. Assume that $A = \left[A_{ij}\right] \in \mathrm{Mat}_{(n-k)b \times kb}\left(\mathbb{F}_q\right)$, where each A_{ij} is a $b \times b$ matrix. Then $\mathcal{C}_{\mathbb{F}_q^b}$ is MDS if and only if every square submatrix of A consisting of full blocks submatrices A_{ij} is nonsingular.*

Recall that matrix is said to be **superregular** (see [15]) if every square submatrix is nonsingular. Several constructions of MDS block codes based on superregular matrices have been proposed (see, for example, [16, 15]). Our purpose here is to extend these constructions using the characterization given in Theorem 1 in order to obtain linear array codes which are also MDS. The way we show it is by using an special type of matrices called superregular b-block matrices.
Definition 1: A matrix $A \in \mathrm{Mat}_{mb \times tb}(\mathbb{F}_q)$ is said to be a **superregular b-block matrix** if every square submatrix of A consisting of full blocks matrices of size $b \times b$ is nonsigular over $\mathbb{F}_q$.

Finally, recall that if $\alpha \in \mathbb{F}_{q^b}$ is a primitive element in $\mathbb{F}_{q^b}$, then the **Zech logarithm** of $k \in \{0, 1, 2, \ldots, q^b - 3, q^b - 2\}$ in the basis α is the integer $Z(k)$ such that $\alpha^{Z(k)} = 1 + \alpha^k$ (see, for example, [17] for the properties of the Zech's logarithm).

3 Main results

Recall that the companion matrix of the monic polynomial

$$p(\mathsf{x}) = \mathsf{x}^b + p_{b-1}\mathsf{x}^{b-1} + \cdots + p_1\mathsf{x} + p_0 \in \mathbb{F}_q[\mathsf{x}]$$

is the square matrix defined as

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -p_0 \\ 1 & 0 & \cdots & 0 & -p_1 \\ 0 & 1 & \cdots & 0 & -p_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -p_{b-2} \\ 0 & 0 & \cdots & 1 & -p_{b-1} \end{bmatrix}.$$

Moreover, if $p(\mathsf{x})$ is a primitive polynomial, it is well known (see, for example, [18]) that $\mathbb{F}_{q^b} \approx \mathbb{F}_q[C]$. This field isomorphism $\psi : \mathbb{F}_{q^b} \longrightarrow \mathbb{F}_q[C]$, can be defined

as $\psi(\alpha) = C$, where $\alpha \in \mathbb{F}_{q^b}$ is a primitive element, and can be extended to a ring isomorphism

$$\Psi : \mathrm{Mat}_{m\times t}(\mathbb{F}_{q^b}) \longrightarrow \mathrm{Mat}_{m\times t}(\mathbb{F}_q[C]) \tag{1}$$

in the following way: if $A = \left[\alpha_{ij}\right] \in \mathrm{Mat}_{m\times t}(\mathbb{F}_{q^b})$, then $\Psi(A) = \left[\psi(\alpha_{ij})\right] \in \mathrm{Mat}_{m\times t}(\mathbb{F}_q[C])$.

This isomorphism allows us to introduce the following result.

Theorem 2: *If $A \in \mathrm{Mat}_{(n-k)\times k}(\mathbb{F}_{q^b})$ is a superrregular matrix, then*

$$H = \left[\begin{array}{cc} \Psi(A) & I_{(n-k)b} \end{array}\right]$$

is the parity check-matrix of an $[n, k, n-k+1]$ MDS array code $\mathcal{C}_{\mathbb{F}_q^b}$.

PROOF: Since A is a superregular matrix over $\mathbb{F}_{q^b}$, we can say that $\Psi(A)$ is a superregular b-block matrix. So, according to Theorem 1 and Definition 1, the array code $\mathcal{C}_{\mathbb{F}_q^b}$ is MDS.

The following example helps us to understand this construction.

Example 1: Consider the primitive polynomial $p(\mathsf{x}) = \mathsf{x}^3 + \mathsf{x} + 1 \in \mathbb{F}_2[\mathsf{x}]$ whose companion matrix is

$$C = \left[\begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right].$$

Let $\alpha \in \mathbb{F}_{2^3}$ be a primitive element such that $\alpha^3+\alpha+1 = 0$. It is easy to check that $A = \left[\begin{array}{cc} 1 & \alpha \\ 1 & \alpha^3 \end{array}\right]$ is a superregular matrix over $\mathbb{F}_{2^3}$. So, according to Theorem 2, the matrix

$$H = \left[\begin{array}{cc:c} I_3 & C & I_6 \\ I_3 & C^3 & \end{array}\right] = \left[\begin{array}{ccc:ccc:cccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

is the parity-check matrix of a $[4, 2, 3]$ array code $\mathcal{C}_{\mathbb{F}_2^3}$. Thus, the array code is MDS. Nevertheless, it is not an MDS code over $\mathbb{F}_2$.

Superregular matrices with entries in a finite field can be obtained from Cauchy matrices or Vandermonde matrices (see, for example, [19, 20, 16, 15]).

4 Decoding algorithm

For a prime number p, Blaum and some of his coauthors [12, 11, 21], introduce a binary $[p+2, p, 3]$ MDS array code and provide a decoding algorithm based on the

corresponding parity-check matrix. In this section we present a similar algorithm for the codes proposed in Section 3, for the binary case, by considering two specific cases. That is, we assume that $\mathcal{C}_{\mathbb{F}_2^b}$ is an $[n, k, n-k+1]$ MDS array linear code and that

$$H = \left[\begin{array}{cccc:c} A_{11} & A_{12} & \cdots & A_{1k} & \\ A_{21} & A_{22} & \cdots & A_{2k} & \\ \vdots & \vdots & & \vdots & I_{(n-k)b} \\ A_{(n-k)1} & A_{(n-k)2} & \cdots & A_{(n-k)k} & \end{array}\right], \tag{2}$$

is the corresponding parity-check matrix. Therefore $A_{ij} = C^{\sigma(i,j)}$ where $C \in \mathrm{Mat}_{b\times b}(\mathbb{F}_2)$ is the companion matrix of a primitive polynomial $p(\mathsf{x}) \in \mathbb{F}_2[\mathsf{x}]$ of degree b.

Assume that $\boldsymbol{c}$ is a codeword, $\boldsymbol{v}$ is the error-corrupted word, and $\boldsymbol{e} = \boldsymbol{v} - \boldsymbol{c}$ is the error vector. Then

$$\begin{aligned} \boldsymbol{c} &= \left[\begin{array}{ccccccc} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_k & \boldsymbol{c}_{k+1} & \cdots & \boldsymbol{c}_n \end{array}\right], \\ \boldsymbol{v} &= \left[\begin{array}{ccccccc} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_k & \boldsymbol{v}_{k+1} & \cdots & \boldsymbol{v}_n \end{array}\right], \\ \boldsymbol{e} &= \left[\begin{array}{ccccccc} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_k & \boldsymbol{e}_{k+1} & \cdots & \boldsymbol{e}_n \end{array}\right], \end{aligned}$$

with $\boldsymbol{c}_\ell, \boldsymbol{v}_\ell, \boldsymbol{e}_\ell \in \mathbb{F}_2^b$ for $\ell = 1, 2, \ldots, k, k+1, \ldots, n$. Then the syndrome $\boldsymbol{s}$ of $\boldsymbol{v}$, defined by $\boldsymbol{s}^T = H\boldsymbol{v}^T$, can be computed, for $i = 1, 2, \ldots, n-k$, as

$$\boldsymbol{s}_i^T = \sum_{\ell=1}^{t} A_{i\ell}\boldsymbol{v}_\ell^T + \boldsymbol{v}_{t+i}^T = \sum_{\ell=1}^{t} A_{i\ell}\boldsymbol{e}_\ell^T + \boldsymbol{e}_{t+i}^T, \tag{3}$$

where $\boldsymbol{s} = \left[\begin{array}{cccc} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_{n-k} \end{array}\right]$.

4.1 Correcting one symbol in error

For the codes constructed in Theorem 2 with parameters $[2+k, k, 3]$ over $\mathbb{F}_2^b$, with $k \in \mathbb{N}$, we can correct one error.

Assume that

$$\boldsymbol{e} = \left[\begin{array}{ccccccc:cc} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{e}_j & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right],$$

then, by expression (3), $\boldsymbol{s}_1^T = A_{1j}\boldsymbol{e}_j^T$ and $\boldsymbol{s}_2^T = A_{2j}\boldsymbol{e}_j^T$, and consequently,

$$\boldsymbol{s}_2^T = A_{2j}A_{1j}^{-1}\boldsymbol{s}_1^T = C^{\sigma(2,j)-\sigma(1,j)}\boldsymbol{s}_1^T.$$

The location in error is given by the integer j satisfying the above expression and can be computed as

$$\boldsymbol{e}_j^T = C^{-\sigma(1,j)}\boldsymbol{s}_1^T = C^{-\sigma(2,j)}\boldsymbol{s}_2^T.$$

If no such j exists and one of the block syndromes is nonzero, then there is an error in the corresponding parity-check block $\boldsymbol{e}_{k+1} = \boldsymbol{s}_1$ or $\boldsymbol{e}_{k+2} = \boldsymbol{s}_2$. Otherwise there are more than one error and we cannot correct.

4.2 Correcting two symbols in error

For the codes constructed in Theorem 2 with parameters $[4+k, k, 5]$ over $\mathbb{F}_2^b$, with $k \in \mathbb{N}$, we can correct two errors.

The following algorithm uses some ideas from [11] and the properties of the Zech logarithm.

Algorithm 1: Assume we know the syndrome $\boldsymbol{s} = \left[\boldsymbol{s}_1\ \boldsymbol{s}_2\ \boldsymbol{s}_3\ \boldsymbol{s}_4\right]$.

1. If at least two of the block syndromes $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\boldsymbol{s}_3$, $\boldsymbol{s}_4$ are zero, then there are no errors in the information symbols and the algorithm stops. Otherwise set $\ell_1 = 0$.
2. Set $\ell_1 = \ell_1 + 1$. If $\ell_1 = k$, then the algorithm stops and we declare there are more than two errors. Otherwise, go to next step.
3. Compute the following vectors

$$\begin{aligned} \boldsymbol{y}_1^T &= \boldsymbol{s}_1^T + A_{1\ell_1} A_{4\ell_1}^{-1} \boldsymbol{s}_4^T, \qquad & \boldsymbol{y}_2^T &= \boldsymbol{s}_2^T + A_{2\ell_1} A_{1\ell_1}^{-1} \boldsymbol{s}_1^T, \\ \boldsymbol{y}_3^T &= \boldsymbol{s}_3^T + A_{3\ell_1} A_{2\ell_1}^{-1} \boldsymbol{s}_2^T, \qquad & \boldsymbol{y}_4^T &= \boldsymbol{s}_4^T + A_{4\ell_1} A_{3\ell_1}^{-1} \boldsymbol{s}_3^T. \end{aligned}$$

4. If $(\boldsymbol{y}_1, \boldsymbol{y}_2) = (\boldsymbol{0}, \boldsymbol{0})$, or $(\boldsymbol{y}_2, \boldsymbol{y}_3) = (\boldsymbol{0}, \boldsymbol{0})$, or $(\boldsymbol{y}_3, \boldsymbol{y}_4) = (\boldsymbol{0}, \boldsymbol{0})$, or $(\boldsymbol{y}_4, \boldsymbol{y}_1) = (\boldsymbol{0}, \boldsymbol{0})$, then there is one single error in the information symbols in the position ℓ_1 given by $\boldsymbol{e}_{\ell_1}^T = A_{t\ell_1}^{-1} \boldsymbol{s}_t^T$ with $t = 2, 3, 4, 1$, and the algorithm stops. Otherwise, go to next step.
5. If $\boldsymbol{y}_3^T = C^{r_1} \boldsymbol{y}_2^T$ with

$$\begin{aligned} r_1 = \sigma(3,\ell_2) - \sigma(2,\ell_2) + Z(\sigma(3,\ell_1) - \sigma(3,\ell_2) - \sigma(2,\ell_1) + \sigma(2,\ell_2)) \\ - Z(\sigma(2,\ell_1) - \sigma(2,\ell_2) - \sigma(1,\ell_1) + \sigma(1,\ell_2)) \end{aligned}$$

for some $\ell_2 \in \{\ell_1 + 1, \ell_1 + 2, \ldots, k\}$, go to next step. Otherwise, go to step 2.
6. If $\boldsymbol{y}_4^T = C^{r_2} \boldsymbol{y}_3^T$ with

$$\begin{aligned} r_2 = \sigma(4,\ell_2) - \sigma(3,\ell_2) + Z(\sigma(4,\ell_1) - \sigma(4,\ell_2) - \sigma(3,\ell_1) + \sigma(3,\ell_2)) \\ - Z(\sigma(3,\ell_1) - \sigma(3,\ell_2) - \sigma(2,\ell_1) + \sigma(2,\ell_2)) \end{aligned}$$

we declare there are errors in positions ℓ_1 and ℓ_2. The algorithm stops. In order to obtain the errors $\boldsymbol{e}_{\ell_1}$ and $\boldsymbol{e}_{\ell_2}$, we solve the linear system

$$\left.\begin{array}{rcrcl} A_{1\ell_1} \boldsymbol{e}_{\ell_1}^T & + & A_{1\ell_2} \boldsymbol{e}_{\ell_2}^T & = & \boldsymbol{s}_1^T \\ A_{2\ell_1} \boldsymbol{e}_{\ell_1}^T & + & A_{2\ell_2} \boldsymbol{e}_{\ell_2}^T & = & \boldsymbol{s}_2^T \end{array}\right\}$$

Otherwise, go to step 2.

The following theorem shows us that Algorithm 1 can correct up to two errors.

Theorem 3: *If $\mathcal{C}_{\mathbb{F}_2^b}$ is a linear array code over $\mathbb{F}_2^b$ with parameters $[4+k, k, 5]$, with $k \in \mathbb{N}$ and the parity-check matrix of the code is given by expression (2) for $n - k = 4$, then Algorithm 1 corrects up to two errors.*

PROOF: We check every possible case and we see that in every case, the algorithm corrects the errors.

Case 1: We have one or two errors in the parity symbols. In this case, three or two syndromes are zero, respectively. Then, we would stop in step 1, declaring no errors in the information symbols.

Case 2: We have one single error in the information symbols in the l_1th position. The syndromes are given by

$$\boldsymbol{s}_t^T = A_{t\ell_1}\boldsymbol{e}_{\ell_1}^T, \quad \text{for } t = 1, 2, 3, 4.$$

It is easy to check that vectors $\boldsymbol{y}_t = \mathbf{0}$, for $t = 1, 2, 3, 4$. Then, the algorithm would run for symbols $1, 2, \ldots, \ell_1$ and would stop in step 4, declaring one error in position ℓ_1.

Case 3: We have one error in the information symbols in the ℓ_1th position and one single error in the parity symbols. Without loss of generality we suppose the error in the parity symbol is in the $(k+1)$th position. The syndromes are given by

$$\begin{aligned}\boldsymbol{s}_1^T &= A_{1\ell_1}\boldsymbol{e}_{\ell_1}^T + \boldsymbol{e}_{k+1},\\ \boldsymbol{s}_t^T &= A_{t\ell_1}\boldsymbol{e}_{\ell_1}^T, \quad \text{for } t = 2, 3, 4.\end{aligned}$$

Now, it is possible to check that $\boldsymbol{y}_1 \neq \mathbf{0}$, $\boldsymbol{y}_2 \neq \mathbf{0}$, and $\boldsymbol{y}_3 = \boldsymbol{y}_4 = \mathbf{0}$. Then, the algorithm would run for symbols $1, 2, \ldots, \ell_1$ and would stop in step 4, declaring one error in position ℓ_1 and another error in a parity symbol.

Case 4: We have two errors in the information symbols in positions ℓ_1 and ℓ_2. The syndromes are given by

$$\boldsymbol{s}_t^T = A_{t\ell_1}\boldsymbol{e}_{\ell_1}^T + A_{t\ell_2}\boldsymbol{e}_{\ell_2}^T, \quad \text{for } t = 1, 2, 3, 4.$$

Then, if we substitute in the vectors $\boldsymbol{y}_k$ given in step 3, we obtain

$$\begin{aligned}\boldsymbol{y}_2^T &= A_{2\ell_1}\boldsymbol{e}_{\ell_1}^T + A_{2\ell_2}\boldsymbol{e}_{\ell_2}^T + A_{2\ell_1}A_{1\ell_1}^{-1}\left(A_{1\ell_1}\boldsymbol{e}_{\ell_1}^T + A_{1\ell_2}\boldsymbol{e}_{\ell_2}^T\right)\\ &= A_{2\ell_2}\boldsymbol{e}_{\ell_2}^T + A_{2\ell_1}A_{1\ell_1}^{-1}A_{1\ell_2}\boldsymbol{e}_{\ell_2}^T\\ &= \left(C^{\sigma(2,\ell_2)} + C^{\sigma(2,\ell_1)-\sigma(1,\ell_1)+\sigma(1,\ell_2)}\right)\boldsymbol{e}_{\ell_2}^T\end{aligned}$$

and using the properties of the Zech's logarithms shown in [17] we obtain

$$\boldsymbol{y}_2^T = C^{\sigma(2,\ell_2)+Z(\sigma(2,\ell_1)-\sigma(1,\ell_1)+\sigma(1,\ell_2)-\sigma(2,\ell_2)}\boldsymbol{e}_{\ell_2}^T.$$

We do the same for $\boldsymbol{y}_3$ and obtain

$$\boldsymbol{y}_3^T = C^{\sigma(3,\ell_2)+Z(\sigma(3,\ell_1)-\sigma(2,\ell_1)+\sigma(2,\ell_2)-\sigma(3,\ell_2)}\boldsymbol{e}_{\ell_2}^T.$$

As a consequence $\boldsymbol{y}_3^T = C^{r_1}\boldsymbol{y}_2^T$, where r_1 is given in step 5.

In the same way, we can obtain $\boldsymbol{y}_4^T = C^{r_2}\boldsymbol{y}_3^T$, where r_2 is given in step 6.

The algorithm would run for symbols $1, 2, \ldots, \ell_1$ and we would have to check for ℓ_1 and for the rest of the information symbols if the expressions in steps 5 and

6 hold. They would hold for ℓ_2. We declare two information errors in positions ℓ_1 and ℓ_2.

The next example allows us to understand the previous algorithm.

Example 2: We consider the primitive polynomial $p(\mathsf{x}) = \mathsf{x}^4 + \mathsf{x} + 1 \in \mathbb{F}_2[\mathsf{x}]$ whose companion matrix is

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

Let $\alpha \in \mathbb{F}_{2^4}$ be a primitive element such that $\alpha^4 + \alpha + 1 = 0$. It is easy to check that

$$A = \begin{bmatrix} \alpha^{14} & 1 & \alpha^5 & \alpha^8 \\ \alpha^5 & \alpha^{13} & \alpha^{14} & \alpha^4 \\ \alpha^2 & \alpha^4 & \alpha^{12} & \alpha^{13} \\ \alpha^6 & \alpha & \alpha^3 & \alpha^{11} \end{bmatrix}$$

is a superregular matrix. Then

$$H = \left[\begin{array}{cccc:c} C^{14} & I_4 & C^5 & C^8 & \\ C^5 & C^{13} & C^{14} & C^4 & \\ C^2 & C^4 & C^{12} & C^{13} & I_{16} \\ C^6 & C & C^3 & C^{11} & \end{array}\right],$$

is the parity-check matrix of an MDS array linear code with parameters $[8, 4, 5]$ over $\mathbb{F}_2^4$.

Assume we receive the word

$$\boldsymbol{v} = \left[\begin{array}{c:c:c:c:c:c:c:c} 0001 & 0000 & 0000 & 0000 & 0001 & 0101 & 0011 & 1110 \end{array}\right].$$

The syndrome vector $\boldsymbol{s} = \left[\begin{array}{cccc} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \boldsymbol{s}_3 & \boldsymbol{s}_4 \end{array}\right]$ is given by

$$\boldsymbol{s}_1^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_2^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_3^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{s}_4^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

All are different from zero. So, we start the algorithm with $\ell_1 = 1$ and we compute the vectors of step 3

$$\boldsymbol{y}_1^T = s_1^T + A_{11}A_{41}^{-1}\boldsymbol{s}_4^T = \boldsymbol{0}^T, \qquad \boldsymbol{y}_2^T = s_2^T + A_{21}A_{11}^{-1}\boldsymbol{s}_1^T = \boldsymbol{0}^T,$$

$$\boldsymbol{y}_3^T = s_3^T + A_{31}A_{21}^{-1}\boldsymbol{s}_2^T = \mathbf{0}^T, \qquad \boldsymbol{y}_4^T = s_4^T + A_{41}A_{31}^{-1}\boldsymbol{s}_3^T = \mathbf{0}^T.$$

Since all of them are zero, that means we have one error in position $\ell_1 = 1$ given by

$$\boldsymbol{e}_1^T = A_{11}^{-1}\boldsymbol{s}_1^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

So, the correct codeword is then

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{v} - \boldsymbol{e} \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} 0001 & 0000 & 0000 & 0000 & 0001 & 0101 & 0011 & 1110 \end{array}\right] \\ &\quad - \left[\begin{array}{c|c|c|c|c|c|c|c} 1101 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} 1100 & 0000 & 0000 & 0000 & 0001 & 0101 & 0011 & 1110 \end{array}\right] \end{aligned}$$

Assume now that we receive another word, for example,

$$\boldsymbol{v} = \left[\begin{array}{c|c|c|c|c|c|c|c} 0000 & 1000 & 0000 & 0000 & 0001 & 0101 & 0011 & 1110 \end{array}\right].$$

The syndrome vector $\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \boldsymbol{s}_3 & \boldsymbol{s}_4 \end{bmatrix}$ is given by

$$\boldsymbol{s}_1^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{s}_2^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}_3^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{s}_4^T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

All are different from zero. Then, we start the algorithm with $\ell_1 = 1$ and we compute the polynomial given in step 3,

$$\boldsymbol{y}_1^T = s_1^T + A_{11}A_{41}^{-1}\boldsymbol{s}_4^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{y}_2^T = s_2^T + A_{21}A_{11}^{-1}\boldsymbol{s}_1^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\boldsymbol{y}_3^T = s_3^T + A_{31}A_{21}^{-1}\boldsymbol{s}_2^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{y}_4^T = s_4^T + A_{41}A_{31}^{-1}\boldsymbol{s}_3^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

None of them are zero, that means we could have an error in $\ell_1 = 1$, but there should be another error in another position. We have to try whether the conditions in steps 5 and 6 hold for any $\ell_2 \in \{2, 3, 4\}$. We start with $\ell_2 = 2$ and compute

$$\begin{aligned} r_1 &= \sigma(3,\ell_2) - \sigma(2,\ell_2) + Z(\sigma(3,\ell_1) - \sigma(3,\ell_2) - \sigma(2,\ell_1) + \sigma(2,\ell_2)) \\ &\quad - Z(\sigma(2,\ell_1) - \sigma(2,\ell_2) - \sigma(1,\ell_1) + \sigma(1,\ell_2)) \\ &= 4 - 13 + Z(6) - Z(-22) = 2. \\ r_2 &= \sigma(4,\ell_2) - \sigma(3,\ell_2) + Z(\sigma(4,\ell_1) - \sigma(4,\ell_2) - \sigma(3,\ell_1) + \sigma(3,\ell_2)) \\ &\quad - Z(\sigma(3,\ell_1) - \sigma(3,\ell_2) - \sigma(2,\ell_1) + \sigma(2,\ell_2)) \\ &= 1 - 4 + Z(7) - Z(6) = -7. \end{aligned}$$

Then

$$C^{r_1}\boldsymbol{y}_2^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \boldsymbol{y}_3^T \quad \text{and} \quad C^{r_2}\boldsymbol{y}_3^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \boldsymbol{y}_4^T.$$

Therefore, there are errors in positions $\ell_1 = 1$ and $\ell_2 = 2$.

In order to obtain the errors $\boldsymbol{e}_{\ell_1}$ and $\boldsymbol{e}_{\ell_2}$, we solve the linear system

$$\left.\begin{array}{rcrcl} A_{1\ell_1}\boldsymbol{e}_{\ell_1}^T & + & A_{1\ell_2}\boldsymbol{e}_{\ell_2}^T & = & \boldsymbol{s}_1^T \\ A_{2\ell_1}\boldsymbol{e}_{\ell_1}^T & + & A_{2\ell_2}\boldsymbol{e}_{\ell_2}^T & = & \boldsymbol{s}_2^T \end{array}\right\}$$

and we obtain the errors

$$\boldsymbol{e}_1^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{e}_2^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The correct codeword is then

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{v} - \boldsymbol{e} \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} 0000 & 1000 & 0000 & 0000 & 0001 & 0101 & 0011 & 1110 \end{array}\right] \\ &\quad - \left[\begin{array}{c|c|c|c|c|c|c|c} 1100 & 1000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} 1100 & 0000 & 0000 & 0000 & 0001 & 0101 & 0011 & 1110 \end{array}\right]. \end{aligned}$$

It is possible to extend this idea for decoding codes with higher length. However, the decoding of such schemes grows exponentially with the length.

5 Conclusions

In this paper a construction of MDS linear array codes based on superregular matrices has been introduced. The main idea is to replace the elements of an $(n-k) \times k$ superregular matrix by powers of the companion matrix of a primitive polynomial of degree b. The resultant matrix allows us to construct the parity-check matrix of an $[n, k, n-k+1]$ MDS linear array code. Also, a decoding algorithm has been introduced that can correct up to $\left\lfloor \frac{n-k}{2} \right\rfloor$ symbols in error for the cases $n-k=2$ and $n-k=4$.

Acknowledgements

The work of the first and the second authors was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Economía y Competitividad of the Gobierno de España. The work of first author was also partially supported by a grant for research students from the Generalitat Valenciana with reference BFPI/2008/138. The work of the third author was partially supported by the research project UMH-Bancaja with reference IPZS01.

References

[1] Blaum, M., Farrell, P.G. & van Tilborg, H.C.A., Array codes. *Handbook of Coding Theory*, eds. V.S. Pless & W.C. Huffman, Elsevier: North-Holland, pp. 1855–1909, 1998.

[2] Xu, L., Bohossian, V., Bruck, J. & Wagner, D.G., Low-density MDS codes and factors of complete graphs. *IEEE Transactions on Information Theory*, **45(6)**, pp. 1817–1826, 1999.

[3] Xu, L. & Bruck, J., X-code: MDS array codes with optimal encoding. *IEEE Transactions on Information Theory*, **45(1)**, pp. 272–276, 1999.

[4] Cassuto, Y. & Bruck, J., Cyclic low-density MDS array codes. *Proceedings of the 2006 IEEE International Symposium on Information Theory (ISIT 2006)*, IEEE: Seattle, WA, USA, pp. 2794–2798, 2006.

[5] Cassuto, Y. & Bruck, J., Cyclic lowest density MDS array codes. *IEEE Transactions on Information Theory*, **55(4)**, pp. 1721–1728, 2009.

[6] Esmaeili, M. & Amoshahy, M.J., On the stopping distance of array code parity-check matrices. *IEEE Transactions on Information Theory*, **55(8)**, pp. 3488–3493, 2009.

[7] Fan, J.L., Array codes as low-desity parity-check codes. *Proceedings of the 2nd International Symposium on Turbo Codes*, Brest, France, pp. 543–546, 2000.

[8] Haslach, C. & Han Vinck, A.J., A decoding algorithm with restrictions for array codes. *IEEE Transactions on Information Theory*, **45(7)**, pp. 2339–2344, 1999.

[9] Louidor, E. & Roth, R.M., Lowest density MDS codes over extension alphabets. *IEEE Transactions on Information Theory*, **52(7)**, pp. 46–59, 2006.

[10] MacWilliams, F.J. & Sloane, N.J.A., *The Theory of Error-Correcting Codes*. North-Holland: Amsterdam, 6th edition, 1988.

[11] Blaum, M., Bruck, J. & Vardy, A., MDS array codes with independent parity symbols. *IEEE Transactions on Information Theory*, **42(2)**, pp. 529–542, 1996.

[12] Blaum, M., Brady, J., Bruck, J. & Menon, J., EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. *IEEE Transactions on Computers*, **42(2)**, pp. 192–202, 1995.

[13] Blaum, M. & Roth, R.M., New array codes for multiple phased burst correction. *IEEE Transactions on Information Theory*, **39(1)**, pp. 66–77, 1993.

[14] Blaum, M. & Roth, R.M., On lowest density MDS codes. *IEEE Transactions on Information Theory*, **45(1)**, pp. 46–59, 1999.

[15] Roth, R.M. & Lempel, A., On MDS codes via Cauchy matrices. *IEEE Transactions on Information Theory*, **35(6)**, pp. 1314–1319, 1989.

[16] Roth, R.M. & Seroussi, G., On generator matrices of MDS codes. *IEEE Transactions on Information Theory*, **31(6)**, pp. 826–830, 1985.

[17] Huber, K., Some comments on Zech's logarithms. *IEEE Transactions on Information Theory*, **36(4)**, pp. 946–950, 1990.

[18] Lidl, R. & Niederreiter, H., *Introduction to Finite Fields and Their Applications*. Cambridge University Press: New York, NY, 1994.

[19] Kéri, G., Types of superregular matrices and the number of n-arcs and complete n-arcs in $PG(r, q)$. *Journal of Combinatorial Designs*, **14(5)**, pp. 363–390, 2006.

[20] Lacan, J. & Fimes, J., A construction of matrices with no singular square submatrices. *Finite Fields and Applications*, eds. G.L. Mullen, A. Poli & H. Stichtenoth, Springer-Verlag: Berlin, volume 2948 of *Lecture Notes in Computer Science*, pp. 145–147, 2003.

[21] Blaum, M., Fan, J.L. & Xu, L., Soft decoding of several classes of array codes. *Proceedings of the 2002 IEEE International Symposium on Information Theory (ISIT 2002)*, IEEE: Lausanne, Switzerland, p. 368, 2002.