

Two-Dimensional Cyclic Codes Correcting Known Error Patterns

Sung Whan Yoon and Jaekyun Moon
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, 305-701, Republic of Korea
Email: shyoon8@kaist.ac.kr, jmoon@kaist.edu

Abstract—¹ This paper considers error-correcting codes designed to correct a finite set of known two-dimensional (2D) error patterns that can occur in a 2D array of bits. Obvious applications for this type of codes include storage and display devices. The specific codes designed in this paper are cyclic codes that can correct any single occurrences of dominant known error patterns that can occur anywhere in the 2D array. As example codes, rate-0.994 codes are constructed which target eight known 2D error patterns in a 63×63 bit array.

I. INTRODUCTION

Two-dimensional (2D) errors are of particular interest in storage. 2D intersymbol interference (ISI) is a well-known cause for 2D errors in storage [1] [5]. In traditional one-dimensional ISI channels, errors are often dominated by a few dominant patterns, and codes can be designed to handle such known error patterns [6] [4]. This paper presents code design methods that can detect and correct known error patterns that occur in a 2D array of bits. A theory of 2D cyclic codes has been established by Imai [2] [3]. In this paper, we take advantage of the 2D theory established in [2] [3] and present a new cyclic code capable of detecting and correcting any single occurrence of error patterns specified in a predetermined list.

Our code design method starts with the establishment of a theorem that provides a set of zeros that produces distinct syndrome sets for the targeted 2D error patterns. We then present procedures, by way of example, for switching or adding zeros to make all syndrome sets to have a full period, a condition necessary to guarantee correction of any targeted error pattern anywhere in the designated array. Efforts are also made to minimize the number of required parity bits while meeting this objective. The rates of the proposed codes are extremely high. We specifically construct rate-0.994 codes targeting 8 particular 2D error patterns that can occur anywhere in a 63×63 bit array.

This paper is organized as follows. In Section II, a brief summary of 2D cyclic code theory of Imai [2] [3] is given. We then present a theorem that specifies a sufficient condition for creating distinct syndrome sets for all target error patterns considered. This theorem provides a basic guideline for implementing the desired codes. In Section III, code design procedures are shown by way of example. Section IV draws conclusions.

II. CONSTRUCTION OF ERROR-PATTERN-CORRECTING 2D CYCLIC CODES

A. General Description of Two-Dimensional Cyclic Codes

The following general description of 2D cyclic codes is based on Imai's work of 1973 [2] and 1977 [3]. A 2D cyclic codeword is a binary array with N_x rows and N_y columns. The size of this cyclic code, say, C , is said to be $(N_x \times N_y)$. Define the set

$$\Omega = \{(i, j) \mid 0 \leq i \leq N_x - 1, 0 \leq j \leq N_y - 1\}. \quad (1)$$

Then a cyclic codeword of size $(N_x \times N_y)$ can be represented with a polynomial:

$$c(x, y) = \sum_{(i,j) \in \Omega} c_{i,j} x^i y^j \quad (2)$$

where $c_{i,j}$ are binary coefficients taking the value 1 or 0. Like the one-dimensional (1D) cyclic codes, cyclic shifting (in the 2D sense) of a 2D codeword $c(x, y)$ in code C results in another 2D codeword in C .

Recall that a 1D cyclic code is completely specified by its generator polynomial. The zeros of the generator polynomial naturally becomes zeros of any codeword. On the other hand, a 2D cyclic code is completely specified by a "set of zeros". With C denoting a 2D cyclic code of size $(N_x \times N_y)$, the set of zeros V_c is defined as follows:

$$V_c = \{(\alpha^i, \beta^j) \mid \forall \text{ codewords } c(x, y), c(\alpha^i, \beta^j) = 0\} \quad (3)$$

where α and β are the N_x^{th} and N_y^{th} roots of the equations $X^{N_x} - 1 = 0$ and $Y^{N_y} - 1 = 0$, respectively. For 1D cyclic codes, an encoder is constructed using a feedback shift-register with the feedback coefficients reflecting the given generator polynomial. For 2D cyclic codes, the set of zeros V_c determines the shift-register setting for encoding and syndrome computation. Before discussing the shift-register operation for 2D cyclic codes, let us talk about the parity check bits. For a 1D cyclic code, given the code length n and with the parity bits occupying k positions, the message bits take the remaining $n - k$ bit positions. A 2D cyclic code also has a designated area for the parity check bits. Let the set $\Pi \in \Omega$ correspond to parity bit positions. The remaining area $\Omega - \Pi$ contains the message bits. The set of zeros V_c determines the parity area and the corresponding feedback connections. Recall that in the

¹This work was supported by NRF of Korea under Grant 2011-0029854.

encoding of a 1D cyclic code, the contents of the feedback shift-register become the parity bits once the entire message bits feed through the register. Likewise, the contents of the 2D feedback shift-register eventually become the parity bits in 2D codes. The 2D register allows shifting of their contents in two directions. Like the 1D case, the contents of the register that are being pushed out of the parity check area Π , are fed back to the 2D register in Π . The feedback connection settings are determined uniquely by V_c .

We now provide a more rigorous description. First, let U denote a set of zeros of a polynomial over $GF(2)$. Then for any elements (or points) (ξ, η) in U ,

$$(\xi^{2^i}, \eta^{2^i}) \quad (4)$$

for $i = 1, 2, \dots, n-1$ are also members of U . Here n is the least positive integer such that $(\xi, \eta) = (\xi^{2^n}, \eta^{2^n})$. The group of all such (ξ^{2^i}, η^{2^i}) is called the conjugate point set of (ξ, η) . The conjugate set is completely represented by (ξ, η) . From this point on, we assume that a set of zeros is the collection of representative points which are delegates of their conjugate point sets. Generally a set of zeros can be written as

$$V_c = \{(\xi_1, \eta_{1,1}), (\xi_1, \eta_{1,2}), \dots, (\xi_1, \eta_{1,t_1}), (\xi_2, \eta_{2,1}), (\xi_2, \eta_{2,2}), \dots, (\xi_2, \eta_{2,t_2}), \dots, (\xi_s, \eta_{s,1}), (\xi_s, \eta_{s,2}), \dots, (\xi_s, \eta_{s,t_s})\}. \quad (5)$$

We can then construct a row vector for each parity bit position $(k, l) \in \Pi$:

$$\mathbf{h}_{k,l} = [(\xi_1^k \eta_{1,1}^l), (\xi_1^k \eta_{1,2}^l) \dots (\xi_1^k \eta_{1,t_1}^l), (\xi_2^k \eta_{2,1}^l), (\xi_2^k \eta_{2,2}^l) \dots (\xi_2^k \eta_{2,t_2}^l) \dots (\xi_s^k \eta_{s,1}^l), (\xi_s^k \eta_{s,2}^l) \dots (\xi_s^k \eta_{s,t_s}^l)] \quad (6)$$

where each element $(\xi_i^k \eta_{i,j}^l)$, $1 \leq i \leq s$ and $1 \leq j \leq t_i$, is a $m_i n_{i,j}$ -tuple. Here m_i is the degree of the minimal polynomial of ξ_i and $n_{i,j}$ is the degree of the monic minimal polynomial of $\eta_{i,j}$ over $GF(2^{m_i})$. The length of the row vector is $\sum_{i=1}^s \sum_{j=1}^{t_i} m_i n_{i,j} \triangleq \sum_{i=1}^s m_i n_i$. The collection of all such vectors corresponding to all positions in Π gives rise to a set of basis vectors. Note that there are a total of $\sum_{i=1}^s m_i n_i$ positions in Π . The corresponding $\sum_{i=1}^s m_i n_i$ basis vectors with length $\sum_{i=1}^s m_i n_i$ are independent of one other. Therefore, a linear combination of these basis vectors can uniquely represent any vector of length $\sum_{i=1}^s m_i n_i$. A vector for position $(i, j) \in \Omega - \Pi$ can also be represented using the form of (6) with $(k, l) \in \Pi$ replaced by $(i, j) \in \Omega - \Pi$. Furthermore, a linear combination of the basis vectors obtained for area Π can also represent each vector in $\Omega - \Pi$:

$$\mathbf{h}_{i,j} = \sum_{(k,l) \in \Pi} h_{k,l}^{(i,j)} \mathbf{h}_{k,l} \quad (7)$$

where $(i, j) \in \Omega - \Pi$. Since $\mathbf{h}_{i,j}$ and $\mathbf{h}_{k,l}$ are all completely specified once V_c is given, the coefficients $h_{k,l}^{(i,j)}$ can be obtained by solving a linear system of equations.

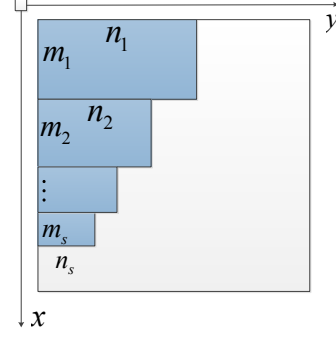


Fig. 1. Encoding for 2D cyclic code

We now specify the feedback connections in the 2D shift-register. First define the border areas between the parity region and the message region:

$$\Pi_{\partial x} = \{(i, j) \in \Pi | (i+1, j) \in \Omega - \Pi\} \quad (8)$$

$$\Pi_{\partial y} = \{(i, j) \in \Pi | (i, j+1) \in \Omega - \Pi\} \quad (9)$$

See Fig. 1. The gray area of a peculiar shape (consisting of a number of m_i by n_i subarrays) in the upper-left corner of the 2D array represents the parity region Π . View the 2D array also as a 2D shift-register with feedback connections given only in the parity region. Imagine the message bits entering the 2D array from the upper-left corner. The shift register is initially cleared. The entered bits are shifted along x and y directions step by step in any order. Once all message bits have entered the array, $\sum_{i=1}^s m_i n_i$ extra zero bits are further inputted to completely fill the array.

Let the contents of the 2D shift-register corresponding to the parity region at some point in the encoding process be written as

$$\sigma(x, y) = \sum_{(k,l) \in \Pi} \sigma_{k,l} x^k y^l. \quad (10)$$

Assuming the message bits that are being entered have already arrived at the left (or the upper) side of the border line between the parity and message areas, a further shift in y (or x) direction will trigger feedback connection. Specifically, the new content $\sigma'_{k,l}$ for position (k, l) is determined by

$$\sigma'_{k,l} = \sigma_{k-1,l} + \sum_{(i,j) \in \Pi_{\partial x}} h_{k,l}^{(i+1,j)} \sigma_{i,j} \quad (11)$$

after a step along x -direction. For shifting one step along y -direction, the new content $\sigma'_{k,l}$ is similarly determined by

$$\sigma'_{k,l} = \sigma_{k,l-1} + \sum_{(i,j) \in \Pi_{\partial y}} h_{k,l}^{(i,j+1)} \sigma_{i,j}. \quad (12)$$

As the last tail zero bit enters the array, the message area is now filled with the message bits and the parity check area is occupied by the final parity bits.

Consider two operators T_x and T_y such that $T_x \sigma(x, y)$ and $T_y \sigma(x, y)$ represent the contents of the shift register corresponding to one-step shifting of $\sigma(x, y)$ along the x and y directions, respectively. Then, after shifting $\sigma(x, y)$ by

TABLE I
ERROR EVENTS FOR 2D PR1 CHANNEL IN POLYNOMIAL FORM

pattern no. i	$e_i(x, y)$	pattern no. i	$e_i(x, y)$
0	1	4	$1 + x + x^2$
1	$1 + y$	5	$1 + x + y + xy$
2	$1 + x$	6	$1 + xy$
3	$1 + y + y^2$	7	$x + y$

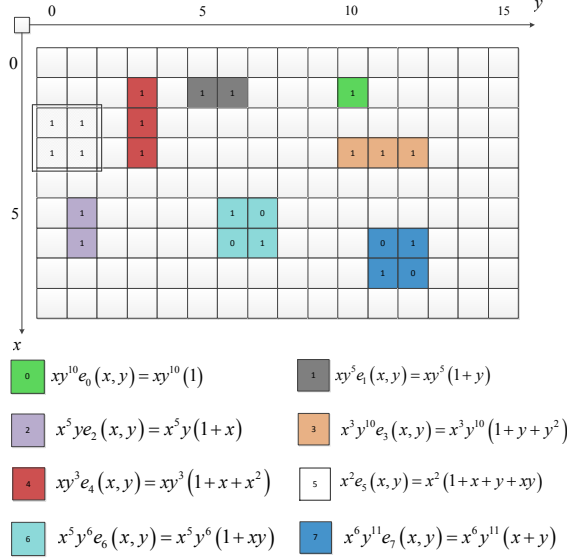


Fig. 2. Examples of 2D Error Events

k positions in the x direction and by l positions in the y direction, we get $T_x^k T_y^l \sigma(x, y)$. Further defining $m(T_x, T_y)$ as the final content of the parity check area at the end of the encoding process, we can write the corresponding codeword as $c(x, y) = m(T_x, T_y) + m(x, y)$. The syndrome polynomial for an error polynomial $e(x, y)$ can be written as $e(T_x, T_y) = \sigma^{(0,0)}(x, y)$. The syndrome $e(T_x, T_y)$ can be viewed as the content of the 2D feedback shift-register corresponding to the parity check area after the $N_x \times N_y$ word $e(x, y)$ completely enters the 2D array.

B. 2D Cyclic Codes Tailored to a Set of Target Error Patterns

Consider the 2D error patterns listed in Table I. These are dominant error events in the 2D Partial Response 1 channel with additive white Gaussian noise [5]. Fig. 2 shows examples of these error patterns occurring in some arbitrary positions in a 9×16 array.

Let $e_i(x, y)$, $0 \leq i \leq L - 1$, be the L dominant error patterns of interest. Write the corresponding syndrome polynomials as $e_i(T_x, T_y) = \sigma_i^{(0,0)}(x, y)$. Assume that the i^{th} error pattern has occurred at position (k, l) . The corresponding error polynomial is $x^k y^l e_i(x, y)$. The corresponding syndrome polynomial is $T_x^k T_y^l e_i(T_x, T_y)$ and we can write $T_x^k T_y^l e_i(T_x, T_y) = \sigma_i^{(k,l)}(x, y)$. Considering all possible locations for the i^{th} error pattern, there are $N_x N_y$ syndrome

polynomials. Let the collection of these syndrome polynomials be

$$S_i = \{\sigma_i^{(k,l)}(x, y) \mid (k, l) \in \Omega\}. \quad (13)$$

Note that while we only consider single occurrences of the target error patterns in this work, a certain error pattern can be viewed as multiple occurrences of some other error patterns, and in this sense the proposed method is also capable of correcting multiple error occurrences for certain simple error pattern types. For instance, when error pattern $e_0(x, y) = 1$ occurs at positions $(k, l) \in \Omega$ and $(k + 1, l + 1) \in \Omega$ simultaneously, then the corresponding double occurrence can also be viewed as a single occurrence of $e_6(x, y) = 1 + xy$ at position $(k, l) \in \Omega$. Likewise, when two error patterns $e_6(x, y) = 1 + xy$ and $e_7(x, y) = x + y$ occur at position $(k, l) \in \Omega$ simultaneously, then the resulting overall error pattern coincides with $e_5(x, y) = 1 + x + y + xy$ at $(k, l) \in \Omega$.

Shifting the syndrome polynomial $\sigma_i^{(k,l)}(x, y)$ of $x^k y^l e_i(x, y)$ ($N_x - k$) and ($N_y - l$) steps along x and y directions, respectively, leads to the ‘‘initial’’ syndrome polynomial $\sigma_i^{(0,0)}(x, y)$. Therefore, by counting the number of shifts to get to the initial syndrome polynomial, we can determine the location of the error pattern. All initial syndrome polynomials need be stored for the decoding process.

For perfect correction of any single occurrence of the targeted error patterns, all syndrome sets must be distinct. The sufficient condition for having all distinct syndrome sets with no common elements are given by the following theorem.

Theorem 1: Let V_c be the set of zeros of a 2D cyclic code. Define E_i as the set $E_i = \{(\alpha^k, \beta^l) \mid e_i(\alpha^k, \beta^l) = 0, (\alpha^k, \beta^l) \in V_c\}$, $0 \leq i \leq L - 1$. Further let C_i be the intersection of E_i and V_c . If $C_i \neq C_j$ for all $i \neq j$, $0 \leq i, j \leq L - 1$, then the syndrome sets S_i and S_j of the respective error polynomials $e_i(x, y)$ and $e_j(x, y)$ are distinct.

We do not include the proof in this paper due to the space constraint.

III. EXAMPLE CODE CONSTRUCTION

Let us target the eight error patterns of Table I. Consider constructing a 2D cyclic code with size $(2^6 - 1, 2^6 - 1) = (63, 63)$. Let α be the 63^{th} root, i.e., $\alpha^{63} = 1$. For a given $e_i(x, y)$, a set of zeros of the error polynomial is

$$E_i = \{(\alpha^k, \alpha^l) \mid e_i(\alpha^k, \alpha^l) = 0, (k, l) \in \Omega\}. \quad (14)$$

Fig. 3 shows all eight sets of zeros E_i , $0 \leq i \leq 7$. Naturally only the representatives for conjugate point sets are shown. In the figure, $(21, 0)$ means the point (α^{21}, α^0) , for example.

A. Code with a Minimum Number of Parity Bits

Theorem 1 allows more than one choice for selecting V_c . However, the number of required parity check bits of each choice is different. Since we need to distinguish eight sets C_i , $0 \leq i \leq 7$, we need at least three elements in V_c . However, in our collection of E_i , a set of three members that can separate all C_i does not exist. To distinguish eight different

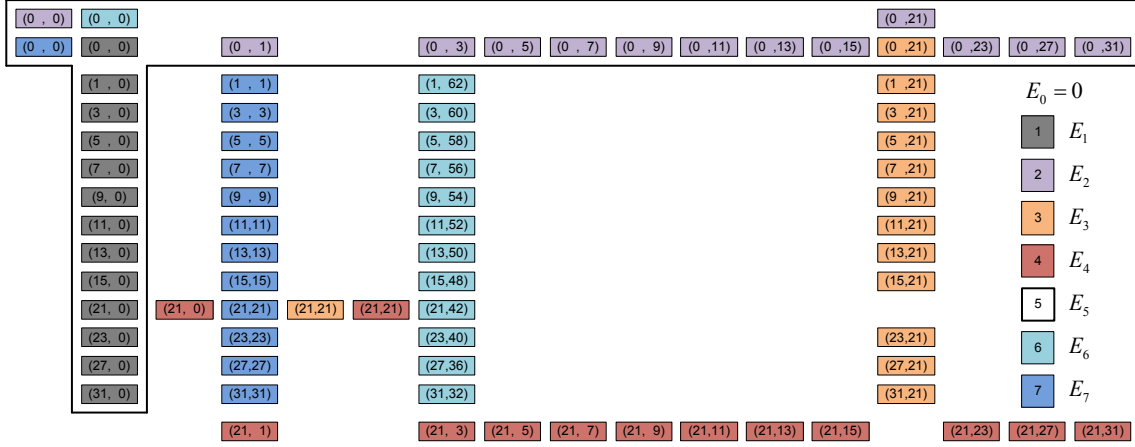


Fig. 3. Collection of sets of zeros E_i , $0 \leq i \leq 7$

subsets of a set, say, $A = \{a_1, a_2, a_3\}$, three elements should appear in four different subsets. However, according to Fig. 3, the only element that appears in more than four groups is (α^0, α^0) . Therefore, the minimum size of our set of zeros should be 4. We can easily see that the point (α^0, α^0) costs only one bit for inclusion in V_c . Moreover, only the four zeros (α^{21}, α^0) , $(\alpha^{21}, \alpha^{21})$, $(\alpha^{21}, \alpha^{42})$ and (α^0, α^{21}) cost two bits each for inclusion in V_c . Therefore, (α^0, α^0) plus a selection of three zeros among (α^{21}, α^0) , $(\alpha^{21}, \alpha^{21})$, $(\alpha^{21}, \alpha^{42})$ and (α^0, α^{21}) would give the smallest number of parity check bits, which is exactly 7. Among four such combinations, only the following two combinations satisfy Theorem 1.

$$\begin{aligned} A &= \{(\alpha^0, \alpha^0), (\alpha^{21}, \alpha^0), (\alpha^{21}, \alpha^{42}), (\alpha^0, \alpha^{21})\} \\ B &= \{(\alpha^0, \alpha^0), (\alpha^{21}, \alpha^0), (\alpha^{21}, \alpha^{21}), (\alpha^0, \alpha^{21})\} \end{aligned} \quad (15)$$

Between these two combinations, implementing a code with A is more complex than a code with B due to the presence of α^{42} in A . Thus, B is preferred over A . Taking set B as V_c , the resulting code would require the minimum number of parity check bits, 7. Accordingly, each syndrome polynomial can be represented as a 7-tuple. In Table II and III, the members of eight syndrome sets corresponding to the eight error patterns are shown. All syndrome polynomials $\sigma_i^{(k,l)}(x, y)$ where $0 \leq k \leq 62$ and $0 \leq l \leq 62$ are given as decimal numbers between 0 and $2^7 - 1$. Not all syndrome values are shown, but all syndrome sets are distinct with no common syndrome polynomials.

The next issue to consider is the period of syndrome sets. Take the syndrome polynomials for $e_0(x, y) = 1$. They repeat in period of +3 along the y direction ($\sigma_0^{(0,0)} = 1$ and $\sigma_0^{(0,3)} = 1 \dots$ in the row of $e_0(x, y) = 1$ in Table II). In addition, it is seen that $\sigma_0^{(0,0)} = 1$ and $\sigma_0^{(3,0)} = 1 \dots$ from the row of $e_0(x, y) = 1$ in Table II and III, indicating that the same syndrome polynomials reappear with shifts of +3 along the x direction. We say that the period of the syndrome set for $e_0(x, y)$ is (+3,+3). This would mean that the location of

error pattern $e_0(x, y)$ cannot be determined completely by computing the syndrome polynomial.

It is easy to see that other error patterns, $e_1(x, y) = 1 + y$, $e_2(x, y) = 1 + x$, $e_6(x, y) = 1 + xy$ and $e_7(x, y) = x + y$, also show a period of (+3,+3) for their syndrome sets. For $e_3(x, y) = 1 + y + y^2$, the syndrome set gives the same syndrome polynomial after shifting +3 along x ($\sigma_3^{(0,0)} = 7$ and $\sigma_3^{(3,0)} = 7 \dots$ in the row of $e_3(x, y) = 1 + y + y^2$ in Table II). Along the y direction ($\sigma_3^{(0,0)} = 7$ and $\sigma_3^{(0,1)} = 7 \dots$ in the row of $e_3(x, y) = 1 + y + y^2$ in Table II), it can be seen that the syndrome set has a period of 1. Accordingly, we say $e_3(x, y) = 1 + y + y^2$ has a period (+3,+1) for its syndrome set. Along the same line, $e_4(x, y) = 1 + x + x^2$ shows a (+1,+3) period. For $e_5(x, y) = 1 + x + y + xy$, its syndrome polynomial $\sigma_5^{(0,0)} = 27$ appears again at positions $(3k - (l \bmod 3), l)$, where $1 \leq k \leq 21$ and $0 \leq l \leq 62$. For $\sigma_5^{(1,0)} = 120$, it appears again at positions $(3k - (l \bmod 3) + 1, l)$, where $1 \leq k \leq 21$. Therefore the period is $(3k - (l \bmod 3), l)$, $1 \leq k \leq 21$ and $0 \leq l \leq 62$, for the syndrome set for $e_5(x, y) = 1 + x + y + xy$. We say that the period in this case is $(3k - (l \bmod 3) \text{ AND } l)$ with “AND” indicating simultaneous shifting along both x and y directions. Table IV summarizes the periods for different syndrome sets.

This code requires only 7 parity bits among $63 \times 63 = 3969$ bits, and the rate of the code is approximately 0.998. This code can detect all 8 targeted error patterns but cannot correct them as the position information is not complete. For complete identification of the error event position and thus a successful correction, each syndrome set must have a full period, i.e., the period matching the size of the array. For the given array size, we need to have (+63, +63) as the period for each and every syndrome set. We next seek a modification of V_c to achieve the full period while compromising the code rate only slightly.

B. 2D Code with Full Periods

Recall that for each position $(k, l) \in \Omega$ there exists an associated vector $\mathbf{h}_{k,l}$. For a general set of zeros V_c in (5), the

TABLE II
SYNDROME SETS (SYNDROME POLYNOMIALS IN DECIMALS) FOR THE MINIMUM-PARITY 2D CODE

Target error polynomial	Syndrome set in decimals												
	$\sigma^{(0,0)}$	$\sigma^{(1,0)}$	$\sigma^{(2,0)}$	$\sigma^{(3,0)}$...	$\sigma^{(62,0)}$	$\sigma^{(0,1)}$	$\sigma^{(1,1)}$	$\sigma^{(2,1)}$	$\sigma^{(3,1)}$...	$\sigma^{(62,1)}$...
1	1	8	32	1	...	32	2	16	64	2	...	64	...
1 + y	3	24	96	3	...	96	6	126	29	6	...	29	...
1 + x	9	40	33	9	...	33	18	80	66	18	...	66	...
1 + y + y ²	7	118	61	7	...	61	7	118	61	7	...	61	...
1 + x + x ²	41	41	41	41	...	41	82	82	82	82	...	82	...
1 + x + y + xy	27	120	99	27	...	99	120	99	27	120	...	27	...
1 + xy	17	72	34	17	...	34	108	77	68	108	...	68	...
x + y	10	48	65	10	...	65	20	46	95	20	...	95	...

TABLE III
SYNDROME SETS (SYNDROME POLYNOMIALS IN DECIMALS) FOR THE MINIMUM-PARITY 2D CODE

Target error polynomial	Syndrome set in decimals													
	$\sigma^{(0,3)}$	$\sigma^{(1,3)}$	$\sigma^{(2,3)}$	$\sigma^{(3,3)}$...	$\sigma^{(62,3)}$...	$\sigma^{(0,62)}$	$\sigma^{(1,62)}$	$\sigma^{(2,62)}$	$\sigma^{(3,62)}$...	$\sigma^{(62,62)}$	
1	1	8	32	1	...	32	...	4	110	93	4	...	93	
1 + y	3	24	96	3	...	96	...	5	1025	125	5	...	125	
1 + x	9	40	33	9	...	33	...	106	51	89	106	...	89	
1 + y + y ²	7	118	61	7	...	61	...	7	118	61	7	...	61	
1 + x + x ²	41	41	41	41	...	41	...	55	55	55	55	...	55	
1 + x + y + xy	27	120	99	27	...	99	...	99	27	120	99	...	120	
1 + xy	17	72	34	17	...	34	...	12	78	92	12	...	92	
x + y	10	48	65	10	...	65	...	111	85	36	111	...	36	

TABLE IV
PERIODS OF THE SYNDROME SETS FOR THE MINIMUM-PARITY 2D CODE

Target error polynomial	Period
1	(+3, +3)
1 + y	(+3, +3)
1 + x	(+3, +3)
1 + y + y ²	(+3, +1)
1 + x + x ²	(+1, +3)
1 + x + y + xy	(3k - (l mod 3) AND l)
1 + xy	(+3, +3)
x + y	(+3, +3)

vector $\mathbf{h}_{k,l}$ for position $(k, l) \in \Omega$ is given by (6). Examination of the periodic structure of this vector provides useful insight into the period properties of the syndrome sets.

First take the simplest error pattern $e_0(x, y) = 1$. It is already known that the initial syndrome polynomial for $e_0(x, y) = 1$ is $\sigma_0^{(0,0)}(x, y) = 1$. Therefore, if the period associated with $e_0(x, y)$ is (P_x, P_y) , then we have $\mathbf{h}_{k,l} = \mathbf{h}_{k+P_x, l+P_y}$. Since the current set of zeros, B , consists only of the terms α^{21} and α^0 , we can write

$$\begin{aligned} \mathbf{h}_{k,l} &= [\alpha^0 \alpha^{21k} \alpha^{21(k+l)} \alpha^{21l}] \\ &= [\alpha^0 \alpha^{21(k+P_x)} \alpha^{21(k+l+P_x+P_y)} \alpha^{21(l+P_y)}] \\ &= \mathbf{h}_{k+P_x, l+P_y}. \end{aligned} \quad (16)$$

We can easily see that when P_x and P_y are multiples of 3, (16) holds, which is consistent with the earlier observation that the period for the syndrome set associated with $e_0(x, y) = 1$ is (+3,+3).

The first strategy to consider in modifying V_c is to replace its member. Noticing that α^0 has no role in determining the period

in (16), it is natural to consider replacing the zero (α^0, α^0) in attempting a change in the period structure of the code. With the removal of (α^0, α^0) from B in (15), two sets $C_0 = E_0 \cap V_c = \phi$ and $C_6 = E_6 \cap V_c$ become identical. Therefore, we should add one of the elements of E_6 to V_c for distinguishing C_6 from $C_0 = \phi$. From Fig. 3, all elements of E_6 take the form $(\alpha^a, \alpha^{63-a})$ with $a = 0, 1, 3, 5, 7, 9, 11, 13, 15, 21, 23, 27, 31$. Imagine replacing (α^0, α^0) with $(\alpha^a, \alpha^{63-a})$ and consider the new period for $(\alpha^a, \alpha^{63-a})$. To let $\alpha^{ak} \alpha^{(63-a)l}$ be equal to $\alpha^{a(k+P_x)} \alpha^{(63-a)(l+P_y)}$, we need $P_x = P_y$ for the cases $a = 1, 5, 11, 13, 23, 31$. For the cases $a = 0, 3, 7, 9, 15, 27$, period properties are worse. For example, for $a = 3$, the vector element repeats when $P_x = P_y$ and additionally when $|P_x - P_y| = 21$ or 42. Therefore, it makes sense to choose a among 1, 5, 11, 13, 23, 31. We choose $a = 1$ for simplicity and this means adding the zero (α^1, α^{62}) . The resulting code has a set of zeros:

$$V_c = \{(\alpha^1, \alpha^{62}), (\alpha^{21}, \alpha^0), (\alpha^{21}, \alpha^{21}), (\alpha^0, \alpha^{21})\} \quad (17)$$

and the syndrome set for $e_0(x, y) = 1$ now has period (+3k AND +3k).

To make the period full, we need additional modification. Let us consider further replacement of zeros. Leaving alone (α^1, α^{62}) which has just been replaced, we consider replacing one of the remaining three zeros: $(\alpha^{21}, \alpha^{21})$, (α^0, α^{21}) and (α^{21}, α^0) .

Candidate 1: If we remove $(\alpha^{21}, \alpha^{21})$, then $C_0 = C_7 = \phi$, $C_1 = C_4 = (\alpha^{21}, \alpha^0)$ and $C_2 = C_3 = (\alpha^0, \alpha^{21})$. Therefore, to satisfy Theorem 1, we should add one of the members of E_7 to our set of zeros. However, this would not be good enough

because the new zero cannot resolve the other two conflicts $C_1 = C_4$ and $C_2 = C_3$. Therefore, at least two additional zeros should be added; one to distinguish C_1 and C_4 , and the other to distinguish C_2 and C_3 . In all, if we remove $(\alpha^{21}, \alpha^{21})$, three additional zeros have to be added to the set of zeros to satisfy Theorem 1.

Candidate 2 : If we remove (α^0, α^{21}) , then $C_0 = C_2 = \phi$, $C_3 = C_7 = (\alpha^{21}, \alpha^{21})$ and $C_1 = C_5 = (\alpha^{21}, \alpha^0)$. Therefore, to satisfy Theorem 1, we should add one of the members of E_2 . Then this zero also resolves the conflict between C_1 and C_5 . But an additional zero for distinguishing C_3 and C_7 is needed. Overall, two zeros should be added if we were to remove (α^0, α^{21}) . This is apparently better than removing the first candidate.

Candidate 3 : If we remove (α^{21}, α^0) , then $C_0 = C_1 = \phi$, $C_2 = C_5 = (\alpha^0, \alpha^{21})$ and $C_4 = C_7 = (\alpha^{21}, \alpha^{21})$. Therefore, to satisfy Theorem 1, we should add a member from E_1 to resolve the conflict between C_0 and C_1 . Then, fortunately this zero will also resolve the conflict between C_2 and C_5 . An additional zero for distinguishing C_4 and C_7 is needed, and like in the second case above, two zeros need to be added if we remove (α^{21}, α^0) . This is also better than removing the first candidate.

We can easily see that replacing (α^0, α^{21}) or (α^{21}, α^0) is better than replacing $(\alpha^{21}, \alpha^{21})$. Let us compare Candidates 2 and 3.

For Candidate 3, the members of E_2 are in the form of (α^a, α^0) where $a = 0, 1, 3, 5, 7, 9, 11, 13, 15, 21, 23, 27, 31$. Consider the periods of the members of E_2 . For the cases where $a = 1, 5, 11, 13, 23, 31$, (α^a, α^0) does not repeat itself until a +63 shift along x . Among them, (α^1, α^0) is the best choice because it is the simplest. We require an additional subarray of column length 6 for the parity check area when we introduce (α^1, α^0) .

For Candidate 2, the members of E_1 are in the form of (α^0, α^a) where $a = 0, 1, 3, 5, 7, 9, 11, 13, 15, 21, 23, 27, 31$. The situation is very similar to the case of Candidate 3, and the best choice is (α^0, α^1) . In this case, we will need an additional subarray of row length 6 for the parity check area as we include (α^0, α^1) in V_c .

In choosing between Candidates 2 and 3, we notice that (α^1, α^{62}) in V_c already requires a parity subarray of column length 6. This would mean that the shape of the parity check area would be more compact with Candidate 3 which requires a parity subarray of column length 6 as discussed above. Therefore, it is better to choose Candidate 3.

Next we consider an additional zero for distinguishing C_4 and C_7 . Let us consider E_4 first. Elements in E_4 are given in the form (α^{21}, α^a) with $a = 0, 1, 3, 5, 7, 9, 11, 13, 15, 23, 27, 31$. For making the period of the syndrome set for $e_0(x, y) = 1$ full, we should only consider elements (α^{21}, α^a) with $a = 1, 5, 11, 13, 23, 31$. Among them, (α^{21}, α^1) is the simplest in terms of implementation complexity. Next, the elements of E_7 are in the form (α^a, α^a) where $a = 0, 1, 3, 5, 7, 9, 11, 13, 15, 23, 27, 31$. We should only consider elements (α^a, α^a) where $a = 1, 5, 11, 13, 23, 31$.

Among them, (α^1, α^1) is the simplest one. Overall the following two sets of zeros are possible options:

$$V_{c1} = \{(\alpha^1, \alpha^0), (\alpha^1, \alpha^{62}), (\alpha^{21}, \alpha^1), (\alpha^{21}, \alpha^{21}), (\alpha^0, \alpha^{21})\} \quad (18)$$

$$V_{c2} = \{(\alpha^1, \alpha^0), (\alpha^1, \alpha^1), (\alpha^1, \alpha^{62}), (\alpha^{21}, \alpha^{21}), (\alpha^0, \alpha^{21})\} \quad (19)$$

Both codes corresponding to the sets (18) and (19) require 22 bits for the parity check bits, giving rise to a code rate of approximately 0.994. It is confirmed that the period of the syndrome set for $e_0(x, y) = 1$ is full for both codes. In addition, it is also seen that the periods of all other syndrome sets are full for both codes. In summary, both codes corresponding to sets (18) and (19) can correct all single occurrences of any of the 8 target error patterns listed in Table I anywhere in the 63×63 array. A general procedure can be established based on the arguments used in constructing the example codes, and it involves examining the V_c -dependent period properties for each target error polynomial for which the period has not yet been made full.

The encoding process involves a simple 2D feedback shift register operation, and the decoding process requires a similar feedback shift-register operation to identify the syndrome pattern and subsequent association with a target error pattern. The codeword structure takes the form shown in Fig 1. It basically is a 63×63 array of bits with 22 parity bits occupying the upper-left area while the message bits filling the remaining area. The exact shapes of the parity check areas Π will be different for the two final codes considered above.

IV. CONCLUSION

In this paper, 2D cyclic codes were designed to correct any single occurrence of known error patterns in a 2D binary array. For constructing these codes, we started with a quick review of 2D cyclic code theory established by Imai. We then extended the error-pattern correcting code idea of Park and Moon to 2D that targeted known dominant error patterns based on the notion of distinct syndrome sets. The code construction relies on selecting a set of zeros that gives rise to a distinct set of syndromes with full periods and a minimal number of parity bits. The resulting codes have extremely high rates and the encoder and decoder processes are based on simple successive shifting of a 2D feedback shift-register.

REFERENCES

- [1] M. Marrow, "Detection and Modelling of 2-Dimensional Signals", Ph. D. dissertation, University of California, San Diego, La Jolla, 2004.
- [2] Hideki Imai, "Two-Dimensional Fire Codes", IEEE Trans. Inf. Theory, vol. 19, no. 6, pp. 796-806, Nov. 1973.
- [3] Hideki Imai, "A Theory of Two-Dimensional Cyclic Codes", Academic Press, Inc. Inform. and control, vol. 34, no. 1, pp. 1-21, May 1977.
- [4] Jihoon Park and Jaekyun Moon, "Error-Pattern-Correcting Cyclic Codes Tailored to a Prescribed Set of Error Cluster Patterns", IEEE Trans. Inf. Theory, vol. 55, no. 4, pp. 1747-1765, Apr. 2009.
- [5] Ismail Demirkan, Paul H. Siegel and Jack K. Wolf, "Error Event Characterization on 2-D ISI Channels", IEEE Trans. Inf. Theory, vol. 55, no. 3, pp. 1146-1152, Mar. 2009.
- [6] Hakim AlHussien, Jihoon Park and Jaekyun Moon, "Iterative Decoding Based on Error-Pattern Correction", IEEE Trans. Magn., vol. 44, no. 1, pp. 181-186, Jan. 2008.