Efficient Systematic Deletions/Insertions of 0's Error Control Codes*

Luca G. Tallini†, Nawaf Alqwaifly‡,§ and Bella Bose‡

†Facoltà di Scienze della Comunicazione, Università degli Studi di Teramo, Teramo, Italy. E-mail: ltallini@unite.it
‡School of EECS, Oregon State University, Corvallis, OR, USA. E-mail: alqwaifn@oregonstate.edu, bose@eecs.orst.edu
§College of Engineering, Qassim University, Saudi Arabia.

2022 IEEE Information Theory Workshop (ITW) | 978-1-6654-8341-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITW54588.2022.9965892

***Abstract*—This paper gives some theory and efficient design of binary block codes capable of controlling the deletions of the symbol "0" (referred to as 0-deletions) and/or the insertions of the symbol "0" (referred to as 0-insertions). This problem of controlling 0-deletions and/or 0-insertions (referred to as 0-errors) is shown to be equivalent to the efficient design of L_1 metric asymmetric error control codes over the natural alphabet, $\mathbb{N}$. Optimal systematic code designs are given. In particular, *for all* $t, k \in \mathbb{N}$, a recursive method is presented to encode k information bits into efficient systematic t Symmetric 0-Error Correcting, $(t+1)$ Symmetric 0-Error Detecting and All Unidirectional 0-Error Detecting (t-Sy0EC/$(t+1)$-Sy0ED/AU0ED) codes of length**

$$n \leq k + t \log_2 k + o(t \log n)$$

as $n \in \mathbb{N}$ increases. Decoding can be efficiently performed by algebraic means using the Extended Euclidean Algorithm (EEA).

***Index Terms*—deletion/insertion of zero errors, repetition/sticky errors, L_1 distance, asymmetric distance, elementary symmetric functions, constant weight codes.**

I. Introduction

In communication and magnetic recording systems, the channel may cause two types of synchronization errors. The first one is not receiving a transmitted symbol (a deletion error), and the second one is receiving a spurious symbol (an insertion error). The propagation of these errors will significantly reduce the performance of the systems.

The general problem of designing efficient codes capable of correcting t insertion and deletion of symbols is still an open research problem even though some results have been reported in these research papers [1], [5]–[10], [14]–[16], [18], [20]–[24] (also please see the references in these papers). However, some efficient code designs for correcting insertion/deletion of some fixed symbol, say 0, are given in [4], [11]–[13], [15], [17], [27], [32]. In the present paper, some systematic codes capable of correcting t insertion and deletion of the symbol 0 are given which are superior to the codes given in [17] and [27] in terms of redundancy and reliability.

Let $\mathbb{Z}_2^*$ be the set of all finite length binary sequences where $\mathbb{Z}_2 \stackrel{\text{def}}{=} \{0, 1\}$. In this paper, we are interested in the efficient design of binary block codes capable of correcting $t \in \mathbb{N}$ or less deletions and/or insertions of a fixed binary symbol, say, $0 \in \mathbb{Z}_2$. In this error model, if

$$X = 0100101000101110 \in \mathbb{Z}_2^{16} \tag{1}$$

is a transmitted binary sequence of length $n = 16$, then

$$\begin{aligned} Y &= 0\mathbf{0}10\lambda1\lambda100\mathbf{0}01\lambda11\mathbf{00}\mathbf{1}0\mathbf{0} \\ &= 001011000011100100 \qquad \in \mathbb{Z}_2^{18} \end{aligned} \tag{2}$$

is the received word obtained from X due to 3 deletions (λ represents the empty symbol) and 5 insertions of the symbol 0. The problem of designing efficient codes to control these types of 0 deletion and/or insertion errors (briefly, 0-errors) is an open research problem introduced by Levenshtein in [15] which is important for at least two reasons. From the application perspective, through the Gray mapping, correcting t deletions or insertions of 0's is equivalent to correcting t repetition (or, sticky) errors which occur in high speed communication and data storage systems due to synchronization loss [4], [17], [27]. From the theoretical perspective, the design problem of t deletion and/or insertion of 0's Error Correcting (i. e., t-Symmetric 0-Error Correcting (t-Sy0EC)) codes is important because it is a particular instance of the general problem also introduced by Levenshtein in [16].

With regard to the 0-error problem, for all $X, Y \in \mathbb{Z}_2^*$, let

$$d_{0\text{-}D/I}(X, Y) \stackrel{\text{def}}{=} \text{the minimum number of deletions and/or insertions of 0's needed to transform the binary word } X \text{ to } Y. \tag{3}$$

For example, if X and Y are the words given in (1) and (2) respectively, then $d_{0\text{-}D/I}(X, Y) = 8$. The above function introduced in [15] is a distance (called here the deletion/insertion of 0's distance). In fact, it is a graph distance defined in the graph (N, E) where the set of nodes $N \stackrel{\text{def}}{=} \mathbb{Z}_2^*$ and the set of edges $E \stackrel{\text{def}}{=} \{(X, Y) \in N^2 : d_{0\text{-}D/I}(X, Y) = 1\}$. Synchronization errors due to 0-errors can be controlled by inserting a marker or synchronization sequence between consecutive codewords in the sequences that are sent [5], [15], [19]. Thus, we assume no synchronization errors due to erroneous receptions of sequences of codewords (i. e., we assume that the receiver knows the length of the received word). In this case, since 1-errors are forbidden in our error model,

$$w_H(X) \neq w_H(Y) \iff d_{0\text{-}D/I}(X, Y) = \infty; \tag{4}$$

where $w_H(Z)$ denotes the Hamming weight of $Z \in \mathbb{Z}_2^*$. In this way, the metric space $(\mathbb{Z}_2^*, d_{0\text{-}D/I})$ or its associated graph (N, E) remains partitioned into many distinct connected components, one for each possible Hamming weight, $w = w_H(X) \in \mathbb{N}$, of words $X \in \mathbb{Z}_2^*$.

The rest of the paper is organized as follows. In Section II, it is shown that the 0-insertion/deletion error control problem is equivalent to some L_1 error control problem [32]. In addition, the non-systematic codes given by us in [32], which is the basis for the proposed systematic code design, is briefly described. In Section III, for fixed $t \in \mathbb{N}$, new efficient recursive code designs are presented to encode $k \in \mathbb{N}$ information bits into systematic t Symmetric 0-Error Correcting, $(t+1)$

*This work is supported by the NSF grants CCF-2006571.

Symmetric 0-Error Detecting and All Unidirectional 0-Error Detecting (t-Sy0EC/$(t+1)$-Sy0ED/AU0ED) codes of length $n = k + t\log_2 k + O(\log\log k)$ and these codes improve the codes in [17] and [27] in terms of redundancy and reliability.

II. 0-DELETION/INSERTION ERRORS AND THE L_1 METRIC

In this section, it is shown that the design problem of t-Sy0EC codes is equivalent to the design problem of some L_1 metric asymmetric error control codes over the natural alphabet, $\mathbb{N}$. Before describing this result, some background materials are given first.

Let $x, y \in \mathbb{Z}_m \subseteq \mathbb{N} \stackrel{\text{def}}{=} \mathbb{Z}_\infty$, where $m \in \mathbb{N}$. Define $x \dot{-} y$ = max $\{0,\ x - y\}$. For example, if $x = 2$ and $y = 0$ then $x \dot{-} y$ = 2 and $y \dot{-} x$ = 0. Given any two words $X, Y \in \mathbb{Z}_m^n$, the operations $X \cap Y \in \mathbb{Z}_m^n$, $X \cup Y \in \mathbb{Z}_m^n$, $X + Y \in \mathbb{N}^n$, and $X \dot{-} Y \in \mathbb{Z}_m^n$ are defined as the digit by digit $\min$, $\max$, integer addition and $\dot{-}$ operation between X and Y, respectively. For example, if $m = 3$, $n = 9$, $X = (012012012)$ and $Y = (000111222)$ then $X \cap Y = (000011012)$, $X \cup Y = (012112222)$, $X + Y = (012123234)$, $X \dot{-} Y = (012001000)$ and $Y \dot{-} X = (000100210)$. In addition, the support of a word $X = x_1 x_2 \ldots x_n \in \mathbb{Z}_m^n$ is $\partial X = s_1 s_2 \ldots s_n \in \mathbb{Z}_2^n$ where $s_i = 1$ if $x_i \neq 0$ and $s_i = 0$ otherwise. For example $\partial(42101) = (11101)$.

To better describe the error control properties of codes for the L_1 metric, the following distances between m-ary words $X, Y \in \mathbb{Z}_m^n$ are considered in [28], [30] (the "+" sign below indicates an integer sum).

$$\begin{array}{ll} \text{symmetric } L_1\text{:} & d_{L_1}^{sy}(X,Y) \stackrel{\text{def}}{=} |Y \dot{-} X| + |X \dot{-} Y|, \\ \text{asymmetric } L_1\text{:} & d_{L_1}^{as}(X,Y) \stackrel{\text{def}}{=} \max\{|Y \dot{-} X|, |X \dot{-} Y|\}, \\ \text{Hamming:} & d_H(X,Y) \stackrel{\text{def}}{=} |\partial(Y \dot{-} X)| + |\partial(X \dot{-} Y)|. \end{array} \quad (5)$$

For example, if $m = 5$, $n = 5$, $X = (01423)$, $Y = (43213)$ then $|X \dot{-} Y| = 3$, $|Y \dot{-} X| = 6$, $|\partial(X \dot{-} Y)| = 2$, $|\partial(Y \dot{-} X)| = 2$ and $d_{L_1}^{sy}(X,Y) = 3 + 6 = 9$, $d_{L_1}^{as}(X,Y) = \max\{6,3\} = 6$ and $d_H(X,Y) = 2 + 2 = 4$. From the error control perspective, if X is the transmitted word and Y is the received word then $Y \dot{-} X$ and $X \dot{-} Y$ give the increasing and decreasing error vectors, respectively. Thus,

$$X = Y - (Y \dot{-} X) + (X \dot{-} Y).$$

Note that,

$$\text{for all } X, Y \in \mathbb{Z}_m^n, \quad d_H(X,Y) \leq d_{L_1}^{sy}(X,Y) \quad (6)$$

because $|\partial X| \leq |X|$, for all $X \in \mathbb{Z}_m^n$.

Constant weight codes play an important role in what follows. Thus, given $n, w \in \mathbb{N}$ and any numeric set $A \subseteq \mathbb{N}$ as alphabet, let

$$\mathcal{S}(A,n,w) \stackrel{\text{def}}{=} \{X \in A^n :\ w_{L_1}(X) = |X| = w\} \quad (7)$$

be the set of all word over A of length n and constant weight w. We readily note, from (7), that

$$\mathcal{S}(A,n,w) = \bigcup_{x \in A} \mathcal{S}(A, n-1, w-x)x; \quad (8)$$

where the above union is a disjoint union of sets. Hence, the general recurring formula,

$$|\mathcal{S}(A,n,w)| = \sum_{x \in A} |\mathcal{S}(A, n-1, w-x)|, \quad (9)$$

holds for, say, the "A-nominal coefficient n choose w", $|\mathcal{S}(A,n,w)|$. If $A = \mathbb{Z}_m$ then the cardinality of the above set is the m-nominal coefficient n choose w

$$|\mathcal{S}(\mathbb{Z}_m,n,w)| = \binom{n}{w}_m = \sum_{v=0}^{m-1} \binom{n-1}{w-v}_m, \quad (10)$$

for all integers $m \in \mathbb{N}$.

Now, if $X \in \mathbb{Z}_2^*$ then X can be uniquely written as [15], [32],

$$X = 0^{v_1}10^{v_2}10\ldots010^{v_w}10^{v_{w+1}} \quad (11)$$

where $l = l(X) \in \mathbb{N}$ is the length of X, $w = w_H(X) \in [0, l(X)]$ is the Hamming weight of X and, for all integers $i \in [1, w+1]$,$v_i \stackrel{\text{def}}{=} v_i(X) \in \mathbb{Z}_{l-w+1} \subseteq \mathbb{N}$ is the i-th run length of 0's in the word X. Note that

$$v_{w+1} = (l(X) - w(X)) - \sum_{i=1}^{w} v_i. \quad (12)$$

Given the above representation, consider the following bijective function (which we call the bucket of 0's mapping)

$$V : \mathbb{Z}_2^* \to \mathbb{Z}_\infty^* = \mathbb{N}^* \quad (13)$$

which associates any $X \in \mathbb{Z}_2^*$ represented as in (11) with $V(X) \stackrel{\text{def}}{=} (v_1, v_2, \ldots, v_w, v_{w+1}) \in \mathbb{N}^*$. For example, if $X = 01\,001\,01\,0001\,01\,1\,1\,0000000 \in \mathbb{Z}_2^*$ then $V(X) = (1,2,1,3,1,0,0,7) \in \mathbb{N}^*$. The mapping V in (13), already considered by Levensthein in [15], defines a bijection from the set of all binary words of any finite length $n \in \mathbb{N}$ and Hamming weight w (= number of 1's of the binary words) into the words over $\mathbb{N}$ of length $w+1$ (= number of buckets defined by the w 1's of the binary words) and L_1 weight $n-w$ (= number of 0's of the binary words). Except for the rightmost "1" which is dropped, the function $V^{-1} : \mathbb{Z}_\infty^* = \mathbb{N}^* \to \mathbb{Z}_2^*$ is nothing but the prefix free unary representation of a sequence of integer numbers. Hence, both V and V^{-1} are one-to-one mappings such that $V(\mathcal{S}(\mathbb{Z}_2,n,w)) = \mathcal{S}(\mathbb{N}, w+1, n-w)$, and $\mathcal{S}(\mathbb{Z}_2,n,w) = V^{-1}(\mathcal{S}(\mathbb{N}, w+1, n-w))$. For example, for $n = 4$, the mapping V acts on $\mathbb{Z}_2^4$ is as reported in Table I. Let

$$\hat{V} : \mathbb{Z}_2^* \to \mathbb{N}^* \quad (14)$$

be the function obtained from V by dropping the last component; $\hat{V}$ associates any $X \in \mathbb{Z}_2^*$ represented as in (11) with $\hat{V}(X) \stackrel{\text{def}}{=} (v_1, v_2, \ldots, v_w) \in \mathbb{N}^*$. Obviously, since V is a one-to-one function, it is possible to reconstruct X from $V(X)$; likewise, even though $\hat{V}$ is not one-to-one (for example, $\hat{V}(0110) = \hat{V}(011000) = (1,0)$), it is possible to reconstruct X from $\hat{V}(X)$ and $n = l(X)$ because of (12). In this case, v_{w+1} can be considered as a parity digit which makes the L_1 weight $w_{L_1}(V(X)) = n - w$. Both functions V

TABLE I
THE MAPPING V ACTING ON $\mathbb{Z}_2^4$.
In the table, $v_{w(X)+1}$ is in boldface and $l(X)$ is the length of any $X \in A^*$.

$l(X)=n$	$w(X)$	X	$V(X)=\hat{V}(X)\boldsymbol{v_{w+1}}$	$l(V(X))$	$w(V(X))$
4	0	0000	**4**	1	4
4	1	0001	3**0**	2	3
		0010	2**1**		
		0100	1**2**		
		1000	0**3**		
4	2	0011	2**00**	3	2
		0101	1**10**		
		0110	1**01**		
		1001	0**20**		
		1010	0**11**		
		1100	0**02**		
4	3	0111	1**000**	4	1
		1011	0**100**		
		1101	0**010**		
		1110	0**001**		
4	4	1111	0000**0**	5	0

and $\hat{V}$ play important roles in our code designs and analysis. Consider the following example words

$$\begin{aligned} X &= 01\,001\,01\,0001\,01\,1\,1\,0 && \in \mathbb{Z}_2^{16}, \\ Y &= 001\,001\,1\,00001\,1\,1\,001\,00 && \in \mathbb{Z}_2^{19}, \\ Y' &= 001\,001\,01\,0001\,01\,00 && \in \mathbb{Z}_2^{16}; \end{aligned}$$

and their associated V values are

$$\begin{aligned} V(X) &= (1,2,1,3,1,0,0,\mathbf{1}) && \in \mathbb{N}^8, \\ V(Y) &= (2,2,0,4,0,0,2,\mathbf{2}) && \in \mathbb{N}^8, \\ V(Y') &= (2,2,1,3,1,\mathbf{2}) && \in \mathbb{N}^6. \end{aligned}$$

Note that if X is sent, Y' can never be received because $7 = w(X) \neq w(Y') = 5$ and 1-errors are forbidden in our channel model; whereas, Y can erroneously be received and the number of 0-deletions ($=2$) plus the number of 0-insertions ($=5$) from X to Y is equal to the L_1 distance between $V(X)$ and $V(Y)$, $d_{L_1}^{sy}(V(X),V(Y)) = 2+5 = 7$. In fact, in general, a sequence $Y \in \mathbb{Z}_2^*$ is obtained from the sequence $X \in \mathbb{Z}_2^*$ due to t_- deletions and t_+ insertions of the symbol 0 if, and only if, $w(Y) = w(X)$ and $d_{L_1}^{sy}(V(Y),V(X)) = t_- + t_+$; that is, $V(Y)$ is obtained from $V(X)$ due to a negative error pattern of magnitude t_- and a positive error pattern of magnitude t_+. Hence, the bucket of 0's mapping $X \to V(X)$ reduces the t_- 0-deletion and t_+ 0-insertion error correction problem into the t_- negative and t_+ positive error correction problem for the L_1 distance over $\mathbb{N}$.

***Theorem** 1 (isometry between $(\mathbb{Z}_2^*, d_{0\text{-}D/I})$ and $(\mathbb{N}^*, d_{L_1}^{sy})$):* For all $X, Y \in \mathbb{Z}_2^*$,

$$d_{0\text{-}D/I}(X,Y) = \begin{cases} d_{L_1}^{sy}(V(X),V(Y)) & \text{if } w(X)=w(Y), \\ \infty & \text{if } w(X)\neq w(Y). \end{cases} \tag{15}$$

Note that (15) implies that $d_{0\text{-}D/I}(X,Y) < \infty$ if, and only if, $w(X) = w(Y)$. So, if we extend the domain of $d_{L_1}^{sy}$ from $\mathbb{N}^l \times \mathbb{N}^l$, $l \in \mathbb{N}$, to $\mathbb{N}^* \times \mathbb{N}^*$ by letting $d_{L_1}^{sy}(U,V) = \infty$ whenever $l(U) \neq l(V)$ then,

$$\text{for all } X, Y \in \mathbb{Z}_2^*, \quad d_{0\text{-}D/I}(X,Y) = d_{L_1}^{sy}(V(X),V(Y)).$$

This implies that the mapping V in (13) is an isometry between the metric spaces $(\mathbb{Z}_2^*, d_{0\text{-}D/I})$ and $(\mathbb{N}^*, d_{L_1}^{sy})$.

Because of Theorem 1, the proposed code is the union of block (i. e., constant) length $n \in \mathbb{N}$ constant weight $w \in [0,n]$ codes, where the union is over w. Under the "bucket of zeros" mapping, for all $w \in [0,n]$, a word $X \in \mathcal{S}(\mathbb{Z}_2, n, w) \subseteq \mathbb{Z}_2^n$ is transferred to a word $V(X) = (v_1, v_2, \ldots, v_{w+1}) \in \mathbb{Z}_{n-w+1}^{w+1}$. Note that, knowing $\hat{V}(X) = (v_1, v_2, \ldots, v_w) \in \mathbb{Z}_{n-w+1}^{w}$ it is possible to calculate v_{w+1}. So, in our code design method, we design L_1 asymmetric distance $t+1$ codes using only the first $w = w_H(X)$ components of $V(X)$. Note that, if $U, V \in \mathbb{Z}_{n-w+1}^{w+1}$ are two constant weight codewords of asymmetric L_1 distance $t+1$ then the symmetric L_1 distance between them is $2(t+1)$ because $|U \dot{-} V| = |V \dot{-} U| = t+1$. So, for any $\mathcal{C}_w \subseteq \mathcal{S}(\mathbb{Z}_2, n, w)$, if the minimum code distance $d_{L_1}^{as}(\hat{V}(\mathcal{C}_w)) > t$ then $d_{L_1}^{as}(V(\mathcal{C}_w)) > t$, and so, $d_{L_1}^{sy}(V(\mathcal{C}_w)) > 2t+1$. Thus, $V(\mathcal{C}_w)$ can correct t-symmetric errors, detect $t+1$ symmetric errors and detect all unidirectional errors under the L_1 distance metric [34]. This implies that $\mathcal{C}_w$ can correct t-symmetric 0-errors, detect $t+1$ symmetric 0-errors and detect all unidirectional 0-errors under the $d_{0\text{-}D/I}$ distance metric. Moreover, any union over w of $\mathcal{C}_w$'s is t-Sy0EC/$(t+1)$-Sy0ED/AU0ED because of (4). Thus, in general, any L_1 distance error control property of codes over $\mathbb{N}$ reflects into the analogous $d_{0\text{-}D/I}$ distance error control property of codes over $\mathbb{Z}_2$ because of Theorem 1. So, from the L_1 metric asymmetric/unidirectional coding theory [2], [3], [28]–[32], [34] and Theorem 1, the following theorem holds.

***Theorem** 2 (Decomposition Theorem):* Let $t, t_-, t_+, \tau \in \mathbb{N}$ be given such that $t_- + t_+ = t$ and $\tau \in [0,t]$. If

$$\mathcal{C} = \bigcup_{w \in [0,n]} \mathcal{C}_w \subseteq \mathbb{Z}_2^n$$

is a binary code of length $n \in \mathbb{N}$ and $\mathcal{C}_w \stackrel{\text{def}}{=} \mathcal{C} \cap \mathcal{S}(\mathbb{Z}_2, n, w)$, for all integer $w \in [0,n]$, then the following statements are equivalent:

1) $\mathcal{C}$ is a t-Sy0EC code;
2) $d_{0\text{-}D/I}(\mathcal{C}) > 2t$;
3) $d_{0\text{-}D/I}(\mathcal{C}) \geq 2t+2$;
4) $\mathcal{C}$ is a (t_-, t_+)-0EC code;
5) $\mathcal{C}$ is a t-Sy0EC/$(t+1)$-Sy0ED/AU0ED code;
6) $\mathcal{C}$ is a τ-Sy0EC/$(2t-\tau+1)$-Sy0ED/AU0ED code;
7) for all $w \in [0,n]$, $d_{L_1}^{as}(\hat{V}(\mathcal{C}_w)) \geq t+1$;
8) for all $w \in [0,n]$, $d_{L_1}^{sy}(V(\mathcal{C}_w)) \geq 2(t+1)$;
9) for all $w \in [0,n]$, $\mathcal{C}_w$ is a t-Sy0EC code.

A. Non Systematic Code Design

The L_1 metric t-SyEC codes over $\mathbb{Z}_m$ are designed based on the σ-codes defined in [25]–[32]. The σ-code theory is based on the sigma polynomials of a word defined below. Let $m \in \mathbb{N} \cup \{\infty\}$, $\mathbb{F}$ be **any** field and $\partial S \subseteq \mathbb{F}$ be a set of $n \in \mathbb{N}$ distinct elements in $\mathbb{F}$. The σ-polynomial associated with a word $X \stackrel{\text{def}}{=} (x_a)_{a \in \partial S} \in \mathbb{Z}_m^n$ is defined as [28],

$$\sigma_X(z) \stackrel{\text{def}}{=} z^{x_0} \prod_{a \in \partial S - \{0\}} (1-az)^{x_a} = z^{x_0}\left(1 + \sigma_1(X)z + \sigma_2(X)z^2 + \ldots\right) \in \mathbb{F}[z]. \tag{16}$$

For example, if $n = 7$, $\partial S = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\} \subseteq \mathbb{F} - \{0\}$ and $X = (3021000) = \{a_0, a_0, a_0, a_2, a_2, a_3\}$ then

$$\sigma_X(z) = (1 - a_0 z)^3 (1 - a_2 z)^2 (1 - a_3 z) = 1 - (3a_0 + 2a_2 + a_3)z + (3a_0^2 + 6a_0 a_2 + 3a_0 a_3 + a_2^2 + 2a_2 a_3)z^2 + \ldots - (a_0^3 a_2^2 a_3) z^7.$$

Note that $\sigma_X(z)$ is a polynomial of degree $\deg(\sigma_X) = w_{L_1}(X) = |X|$ having $w_H(X) = |\partial X|$ distinct roots in $\mathbb{F}$, each with multiplicity x_a, for $a \in \partial S \subseteq \mathbb{F}$. In particular, X coincides with the multiset of all the inverses of the roots of $\sigma_X(z)$, where we let $1/0 \stackrel{\text{def}}{=} 0$. Hence, its coefficient sequence is given by the elementary symmetric functions, 1, $\sigma_1(X)$, $\sigma_2(X)$, $\ldots \in \mathbb{F}$, of the elements in the multiset $X - \{0\}$ ordered in increasing order of their degrees, and eventually right shifted by $x_0 \in \mathbb{Z}_m \subseteq \mathbb{N}$ if $0 \in \partial S \subseteq \mathbb{F}$. At this point the general definition of σ-code is the following. For all polynomials $g(z), \sigma(z) \in \mathbb{F}[z]$, the m-ary σ-code of length n associated with g and σ is defined as $\mathcal{C}_{g,\sigma}(\mathbb{Z}_m, n) \stackrel{\text{def}}{=}$

$$\left\{ X \in \mathbb{Z}_m^n \;\middle|\; \begin{array}{l} \sigma_X(z) = c_X \sigma(z) \bmod g(z), \\ \text{with } c_X \in \mathbb{F} - \{0\} \end{array} \right\}. \tag{17}$$

For simplicity, we can choose $g(z) = z^{t+1}$.

To define a t-Sy0EC code $\mathcal{C} \subseteq \mathbb{Z}_2^n$, the σ-codes are used in the function $\hat{V}$ codomain; where $\hat{V}$ is given in (14). So, $X \in \mathcal{C}$ if, and only if $\sigma_{\hat{V}(X)}(z) = \sigma(z) \bmod z^{t+1}$, where $\sigma(z)$ is a monic polynomial of degree t. Note that under the mapping $X \to \sigma_{\hat{V}(X)}(z) \bmod z^{t+1}$, the set of constant weight w vectors of length n over $\mathbb{Z}_2$ (and in fact, the set $\mathcal{S}(\mathbb{N}, w+1, n-w)$) is partitioned into $|\mathbb{F}|^t$ classes, $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{\mathbb{F}^t}$, where, X and Y are in $\mathcal{D}_i$ if, and only if, $\sigma_{\hat{V}(X)}(z) = \sigma_{\hat{V}(Y)}(z) \bmod z^{t+1}$. Now, we prove that each of the $\hat{V}(\mathcal{D}_i)$'s is an asymmetric L_1 distance $t+1$ code. Suppose $X, Y \in \mathcal{D}_i$, let $\hat{V} \stackrel{\text{def}}{=} \hat{V}(X)$ and $\hat{U} \stackrel{\text{def}}{=} \hat{V}(Y)$. Then, $\sigma_{\hat{V}}(z) = \sigma_{\hat{U}}(z) \bmod z^{t+1}$ and this implies $\sigma_{\hat{V} \dot{-} \hat{U}}(z) = \sigma_{\hat{U} \dot{-} \hat{V}}(z) \bmod z^{t+1}$ because

$$\text{for all } A, B \in \mathbb{N}^n,\ \sigma_A(z)\sigma_{B \dot{-} A}(z) = \sigma_B(z)\sigma_{A \dot{-} B}(z). \tag{18}$$

Now, if the asymmetric L_1 distance between $\hat{V}$ and $\hat{U}$ is $s < t+1$ then the degrees of $\sigma_{\hat{V} \dot{-} \hat{U}}(z)$ and $\sigma_{\hat{U} \dot{-} \hat{V}}(z)$ are $s < t+1$ and so, $\sigma_{\hat{V} \dot{-} \hat{U}}(z) = \sigma_{\hat{U} \dot{-} \hat{V}}(z)$. This means, $\sigma_{\hat{V} \dot{-} \hat{U}}(z)$ has $2s$ roots (i. e., the s roots of $\sigma_{\hat{V} \dot{-} \hat{U}}(z)$ and the s roots of $\sigma_{\hat{U} \dot{-} \hat{V}}(z)$), which gives a contradiction. Therefore, the minimum asymmetric L_1 distance of the code is at least $t+1$. So, under the $X \to \sigma_{\hat{V}(X)}(z) \bmod z^{t+1}$ mapping the set $\mathcal{S}(\mathbb{Z}_2, n, w)$ is partitioned into the $|\mathbb{F}|^t$ classes $\mathcal{D}_i$'s. Thus, by pigeon-hole principle, one of the classes, say $\tilde{\mathcal{D}}(\mathbb{F}; n, w)$ should have at least $\binom{n}{w}/|\mathbb{F}|^t$ codewords. From equivalence 7) of Theorem 2, the t-Sy0EC code, $\mathcal{C}$, can be simply defined by letting for all $w \in [0, w]$, $\mathcal{C}_w \stackrel{\text{def}}{=} \tilde{\mathcal{D}}(\mathbb{F}; n, w) \subseteq \mathcal{S}(\mathbb{Z}_2, n, w)$; where, to maximize $|\mathcal{C}|$, the algebraic structure $\mathbb{F}$ is chosen to be the smallest possible field if $t > 1$ or the smallest group if $t = 1$. In this way, the number of codewords is

$$|\mathcal{C}| \geq \sum_{w=0}^{n} \left\lceil \binom{n}{w} \Big/ |\mathbb{F}_w|^t \right\rceil. \tag{19}$$

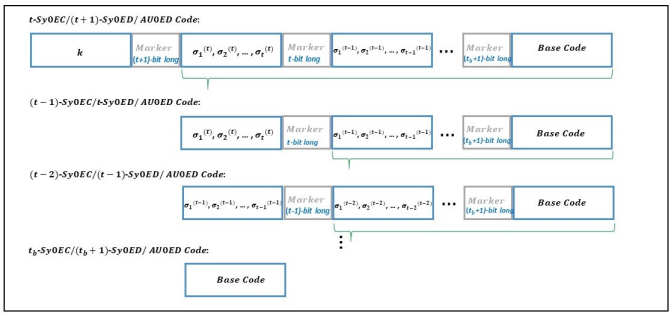

Fig. 1. Proposed recursive code construction.

where $\mathbb{F}_w$ is the smallest field, $\mathbb{F}$, whose cardinality is $|\mathbb{F}| > w$, when $t > 1$ and $\mathbb{F}_w = (\mathbb{Z}_{w+1}, +_{\bmod (w+1)})$ when $t = 1$. Note that if $t = 1$, then $|\mathbb{F}_w| = w + 1$.

III. Systematic recursive code design

In the proposed non-systematic t-Sy0EC code design in Section II, any given word $X \in \mathbb{Z}_2^k$ is mapped to

$$\sigma_{\hat{V}(X)0^*}(z) = 1 + \sigma_1\left(\hat{V}_X\right) z + \ldots + \sigma_t\left(\hat{V}_X\right) z^t \bmod z^{t+1};$$

where

$$\hat{V}_X \stackrel{\text{def}}{=} (v_1, v_2, \ldots, v_w, 0, 0, \ldots, 0) \in \mathbb{Z}_k^k \equiv \mathbb{F}^k. \tag{20}$$

Now, all input words mapping into the same $\sigma_1, \sigma_2, \ldots, \sigma_t \in \mathbb{F}_{w(X)}$ form a t-Sy0EC. To design the code, if, for simplicity, we use the same field $\mathbb{F} \stackrel{\text{def}}{=} \mathbb{F}_k$, for all possible weights $w = w(X) \in [0, k]$, then the set of input vectors is partitioned into $|\mathbb{F}|^t$ classes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{\mathbb{F}^t}$, and each of these classes is a t-Sy0EC (or, equivalently, a t-Sy0EC/$(t+1)$-Sy0ED/AU0ED) code. In the proposed systematic t-Sy0EC recursive code design given in this section, for the given information word $X \in \mathbb{Z}_2^k$, we first find the values of its $\sigma_i \stackrel{\text{def}}{=} \sigma_i\left(\hat{V}_X\right)$'s, $i = 1, 2, \ldots, t$, and append them as check. Then, assuming these σ_i's as an information word, we encode them with a $(t-1)$-Sy0EC (or, $(t-1)$-Sy0EC/t-Sy0ED/AU0ED) code. This process continues until a base code is used.

We now explain why this code gives distance $d_{0\text{-}D/I}(C) \geq 2t+2$. If two information words X and Y map to the same value then the $d_{0\text{-}D/I}(X, Y) \geq 2t+2$. On the other hand if they map to different values, by our construction, the checks will have $d_{0\text{-}D/I} \geq 2t$. Since X and Y are constant weight words, $d_{0\text{-}D/I}(X, Y) \geq 2$ and so the distance between these two codewords is at least $2t+2$. However, for the reasons given below, we need to insert a marker between the successive words generated in these recursive iterations. This code design is shown in Figure 1. For t-Sy0EC/$(t+1)$-Sy0ED/AU0ED error control decoding, note that if the receiver knows the sent information word weight and the check symbols (i. e., the σ_i's) then the sent information word can be recovered by first applying the V mapping to the information part and then applying any L_1 metric t-SyEC/$(t+1)$-SyED/AUED decoding algorithm for constant weight codes to this information part. Based on (18), efficient L_1 metric t-SyEC/$(t+1)$-SyED/AUED error control algorithm can be defined for constant weight σ-codes which are based on the EEA (Extended Euclidean Algorithm). Thus, in the entire decoding process, once the

correct parsing of the received information word is done, we sequentially decode the remaining received check part starting from the base code, all the way up to the first iteration.

Since the proposed efficient code designs rely on the concatenation of some codewords, we need to be aware of the following unexpected behavior of the $d_{0\text{-}D/I}$ distance. Unlike usual metrics, the metric function $d_{0\text{-}D/I}$ is not additive with respect to concatenation. For example, if $X_1 = 010$, $X_2 = 010$, $Y_1 = 0001$, $Y_2 = 001$ then

$$d_{0\text{-}D/I}(X_1X_2, Y_1Y_2) = d_{0\text{-}D/I}(010\,010, 0001\,001) = 3 \neq d_{0\text{-}D/I}(010, 0001) + d_{0\text{-}D/I}(010, 001) = d_{0\text{-}D/I}(X_1, Y_1) + d_{0\text{-}D/I}(X_2, Y_2) = 3 + 2 = 5.$$

In general, the $d_{0\text{-}D/I}$ metric is not additive with respect to the concatenation. In order to avoid this problem, similar to the ideas proposed in [5], [15], [19], we insert 'markers' as shown in the figure.

Next, how the base codes are designed is described. Depending on the value of t/k, different base code designs can be defined, each of which gives better information rate than the others. Because of the space limitation, here we only give a code design especially suited when t/k is big.

A. Base Code Design for t/k Big

Given $t, k \in \mathbb{N}$, the basic idea of this code construction is a generalization of the following. Divide k information bits into $\lceil k/b \rceil$ bytes of b bits. Each of these b-bit bytes can be considered as an element in a field $\mathbb{F}$, where $\max\{2^b, \lceil k/b \rceil + t\} \leq |\mathbb{F}|$. Design a distance $t+1$ Reed-Solomon code with these bytes as the information digits. Note that this RS code generates t check digits. The next step is to map each codeword digit of the generated RS code to a balanced code. In general, we use a $(\tau-1)$-Sy0EC constant weight codes. Finally, to separate the bytes, insert a 1 after each byte for synchronization. The following example explains this base code design.

***Example** 1:* Suppose we are given $k = 9$ information bits and we want to design a 4-Sy0EC code. Choose $b = 3$ and so the field $\mathbb{F} = GF(2^3)$ can be used for the code design because $\max\{2^b, \lceil k/b \rceil + t\} = \max\{2^3, 9/3+4\} = 2^3 \leq |\mathbb{F}|$. Assume the given information word is

$$x_1x_2x_3\,x_4x_5x_6\,x_7x_8x_9 \stackrel{\text{def}}{=} X_1\,X_2\,X_3 \in \left(\mathbb{Z}_2^3\right)^3 \equiv \mathbb{Z}_2^9.$$

Each of the byte, $X_i \in \mathbb{Z}_2^3$, $i = 1, 2, 3$, can be considered as an element in the field $\mathbb{F} = GF(2^3)$. The $(\mathbb{F}; 7, 3, 5)$ RS code of length 7 and minimum distance $t+1 = 5$ can be designed by taking the generator polynomial $g(z) = (z-\alpha^0)(z-\alpha^1)(z-\alpha^2)(z-\alpha^3)$ where α is a root of the primitive polynomial $z^3 + z + 1$. Thus, $\alpha^0 \equiv 001$, $\alpha^1 \equiv 010$, $\alpha^2 \equiv 100$, $\alpha^3 \equiv 011$, $\alpha^4 \equiv 110$, $\alpha^5 \equiv 111$, $\alpha^6 \equiv 101$ and $\mathbf{0} \equiv 000$. For simplicity, assume the given information word is $000\,000\,000 \equiv \mathbf{0\,0\,0}$ so that its associated RS codeword is

$$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \in \mathbb{F}^7. \tag{21}$$

Now we need to design a one-to-one mapping of the symbols in $\mathbb{F} = GF(2^3)$ to the codewords of a $(\tau-1)$-Sy0EC constant weight code. For this example, assume $\tau = 1$ and so we can use 2-out-of-5 words for this mapping since $\binom{5}{2} = 10 \geq 8 = |\mathbb{F}|$. One of these mappings is $\mathbf{0} \equiv 000 \to 00011$, $\alpha^0 \equiv 001 \to 00101$, $\alpha^1 \equiv 010 \to 00110$, $\alpha^2 \equiv 100 \to 01001$, $\alpha^3 \equiv 011 \to 01010$, $\alpha^4 \equiv 110 \to 01100$, $\alpha^5 \equiv 111 \to 10001$, $\alpha^6 \equiv 101 \to 10010$. Thus, for the all 0 RS codeword, after this mapping and also adding an additional 1 at the end of each byte, the codeword is

$$000111\,000111\,000111\,000111\,000111\,000111\,000111.$$

Suppose A and B are two codewords. Since the Hamming distance between them is at least five and each symbol is mapped into a balanced 2-out-5 codeword, the $D_{0-D/I} \geq 2 \cdot 5 = 10$. Thus the code can correct 4-Sy0EC/5-Sy0ED/AU0ED code.

Now we explain how the $t = 4$ 0-error correction is done. This is based on, as explained later, $e \in [0, t]$ erasures error correction, $\theta \stackrel{\text{def}}{=} \lfloor (t-e)/2 \rfloor$ error correction and $\delta \stackrel{\text{def}}{=} \lceil (t-e)/2 \rceil$ error detection (e-EEC/θ-EC/δ-ED) for this code. In particular, the e-EEC/θ-EC/δ-ED error control algorithm for Reed-Solomon code is used to simulate the $(e+2\theta)$-Sy0EC/$(e+2\theta+1)$-Sy0ED part of the control algorithm for this code. Since a 1 is inserted at the end of each byte, by reading from left to right of the received word, the bytes can be parsed correctly even with some 0-errors. In general, if the number of 0 insertion errors is not equal to the number of 0 deletion errors in a byte, then this byte can be identified as erroneous and, hence, set equal to an erasure byte. On the other hand, if the number of 0 insertion errors is equal to the number of 0 deletion errors in a byte, then that byte is an erroneous byte which, a priori, can not be identified as erroneous. For example, suppose the received word is

$$001011\,0001011\,0000111\,000111\,000111\,000111\,000111.$$

By counting the number of 1's from left to right, it can be noticed that the balanced encoding of the second and third bytes are 6 bit long (excluding the synchronizing bit 1) and so these can be set as erasure bytes by the receiver. After inverse mapping from 2-out-of-5 codewords to $GF(2^3)$ the received word is

$$(\alpha^0 \equiv 001, *, *, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}) \in \{\mathbb{F} \cup \{*\}\}^7;$$

where "$*$" stands for an erasure symbol. Using 2-EEC/1-EC decoding algorithm, the receiver can correct and obtain the sent codeword in (21). ▲

IV. Concluding Remarks

Some theory and efficient design of binary block codes capable of controlling the deletions and/or insertions of the symbol "0" (i. e., the 0-errors) are given. It is shown that the design of codes for insertion and/or deletion of zeros is equivalent to the design of the L_1 metric error control codes. Some asymptotically optimal non-systematic and systematic codes for correcting these errors are described and their encoding method is also explained. Because of the space limitation, the decoding methods are not given here. However, the decoding can be done efficiently using the Extended Euclidean Algorithm (EEA).

REFERENCES

[1] K. A. S. Abdel-Ghaffar, F. Paluncic, H. C. Ferreira and W. A. Clarke, "On Helberg's Generalization of the Levenshtein Code for Multiple Deletion/Insertion Error Correction", *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1804-1808, March 2012.

[2] M. Blaum, *Codes for Detecting and Correcting Unidirectional Errors*. IEEE Computer Society Press, Washington, DC, USA, 1993.

[3] B. Bose and T. R. N. Rao, "Theory of undirectional error correcting/detecting codes", *IEEE Trans. on Comput.*, vol. 31, pp. 521–530, June 1982.

[4] L. Dolecek and V. Anantharam, "Repetition error correcting sets: Explicit constructions and prefixing methods", *SIAM Journal on Discrete Mathematics*, vol. 23, no. 4, pp. 2120–2146, 2010.

[5] H. C. Ferreira, W. A. Clarke, A. S. J. Helberg, K. A. S. Abdel-Ghaffar, and A. J. Han Vinck, "Insertion/Deletion Correction with Spectral Nulls", *IEEE Trans. on Inform. Theory*, vol. 43, no. 2, pp. 722–732, Mar 1997.

[6] V. Guruswami and J. Håstad. "Explicit two-deletion codes with redundancy matching the existential bound", *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 21–32, Jan. 2021.

[7] A. S. J. Helberg and H. C. Ferreira, "On Multiple Insertion/Deletion Correcting Codes", *IEEE Trans. on Inform. Theory*, vol. 48, no. 1, pp. 305–308, Jan. 2002.

[8] A. S. J. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes", *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 305–308, Jan. 2002.

[9] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Duplication-Correcting Codes for Data Storage in the DNA of Living Organisms", *IEEE Transactions on Information Theory*, vol. 63, no. 8, pp. 4996–5010, Aug. 2017.

[10] W. Kautz, "Fibonacci codes for synchronization control", *IEEE Transactions on Information Theory*, vol. 11, no.2, pp. 284–292, April 1965.

[11] M. Kovačević and V. Y. F. Tan, "Asymptotically Optimal Codes Correcting Fixed-Length Duplication Errors in DNA Storage Systems", *IEEE Communications Letters*, vol. 22, pp. 2194-2197, Nov. 2018.

[12] M. Kovačević, "Runlength-Limited Sequences and Shift-Correcting Codes: Asymptotic Analysis", *IEEE Trans. on Inform. Theory*, Vol. 65, pp. 4804–4814, August 2019.

[13] A. A. Kulkarni, "Insertion and deletion errors with a forbidden symbol", *2014 IEEE Information Theory Workshop (ITW 2014)*, pp. 596–600, November 2014.

[14] A. A. Kulkarni and N. Kiyavash, "Non-asymptotic Upper Bounds for Deletion Correcting Codes", *IEEE Trans. on Inform. Theory*, Vol 59, No 8, pp 5115–5130 (Aug. 2013).

[15] V. I. Levenshtein, "Binary codes with correction for deletions and insertions of the symbol 1", *Probl. Peredachi Inf.*, vol. 1, n. 1, pp. 12–25, 1965 (in Russian). An english translation can be found in, "Binary codes capable of correcting spurious insertions and deletions of ones", *Problems of Information Transmission*, vol. 1, pp. 8–17, 1965.

[16] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals", *Sov. Phys. Dokl.*, vol. 10, no. 8, pp. 707–710, 1966.

[17] H. Mahdavifar and A. Vardy, "Asymptotically optimal sticky-insertion-correcting codes with efficient encoding and decoding", *2017 IEEE ISIT*, pp. 2683–2687, June 2017.

[18] F. Palunčić, K. A. S. Abdel-Ghaffar, H. C. Ferreira, and W. A. Clarke, "A Multiple Insertion/Deletion Correcting Code for Run-Length Limited Sequences", *IEEE Trans. on Inform. Theory*, Vol. 58, pp. 1809–1824, March 2012.

[19] F. Sellers, "Bit loss and gain correction code", in *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 35-38, Jan. 1962.

[20] J. Sima and J. Bruck, "Optimal k-Deletion Correcting Codes", *2019 IEEE ISIT*, pp. 847-851, July 2019.

[21] J. Sima, N. Raviv and J. Bruck, "Two Deletion Correcting Codes From Indicator Vectors", *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2375–2391, April 2020.

[22] J. Sima, R. Gabrys and J. Bruck, "Optimal Systematic t-Deletion Correcting Codes", *Proceedings 2020 IEEE International Symposium on Information Theory*, Los Angeles, CA, USA, 2020, pp. 769–774.

[23] J. Sima, R. Gabrys and J. Bruck, "Optimal Codes for the q-ary Deletion Channel", *Proceedings 2020 IEEE International Symposium on Information Theory*, Los Angeles, CA, USA, 2020, pp. 740-745.

[24] N. J. A. Sloane, "On single-deletion-correcting codes", in *Codes and Designs*, Ohio State University (Ray-Chaudhuri Festschrift), pp. 273–291, 2000. Online: https://arxiv.org/abs/math/0207197.

[25] L. G. Tallini and B. Bose, "On a new class of error control codes and symmetric functions", *2008 IEEE ISIT*, pp. 980-984, July 2008.

[26] L. G. Tallini and B. Bose, "On decoding some error control codes using the elementary symmetric functions". In *Trends in Incidence and Galois Geometries: a Tribute to Giuseppe Tallini - Quaderni di Matematica*, F. Mazzocca, N. Melone and D. Olanda Ed.. vol. 19, p. 265-297, Caserta, Dipartimento di Matematica, Seconda Università di Napoli, 2010.

[27] L. G. Tallini, N. Elarief and B. Bose, "On efficient repetition error correcting codes", *2010 IEEE ISIT*, pp. 1012–1016, June 2010.

[28] L. G. Tallini and B. Bose, "On L_1-distance error control codes", *2011 IEEE ISIT*, pp. 1026–1030, July/Aug. 2011.

[29] L. G. Tallini, B. Bose, "On symmetric L_1 distance error control codes and elementary symmetric functions", *2012 IEEE ISIT*, pp. 741–745, July 2012.

[30] L. G. Tallini and B. Bose, "On L_1 metric asymmetric/unidirectional error control codes, constrained weight codes and σ-codes", *2013 IEEE ISIT*, pp. 694–698, July 2013.

[31] L. G. Tallini and B. Bose, "On Some New $\mathbb{ZZ}_m$ Linear Codes Based on Elementary Symmetric Functions", *2018 IEEE ISIT*, pp. 1665–1669, June 2018.

[32] L. G. Tallini, N. Alqwaifly and B. Bose, "On Deletions and Insertions of the Symbol "0" and Asymmetric/Unidirectional Error Control Codes", *2019 IEEE ISIT*, pp. 2384–2388, July 2019.

[33] R. R. Varshamov and G. M. Tenengolts, "Correcting code for single asymmetric errors", *Avtomatika i Telemekhanika* (in Russian), vol. 26, no. 2, pp. 228–292, 1965.

[34] J. H. Weber, C. de Vroedt, D. E. Boekee, "Necessary and sufficient conditions on block codes correcting/detecting errors of various types", *IEEE Trans. on Comput.*, vol. 41, pp. 1189–1193, Sept. 1992.