Constructions and Decoding of Cyclic Codes Over b-Symbol Read Channels

Eitan Yaakobi, *Member, IEEE*, Jehoshua Bruck, *Fellow, IEEE*, and Paul H. Siegel, *Fellow, IEEE*

***Abstract*— Symbol-pair read channels, in which the outputs of the read process are pairs of consecutive symbols, were recently studied by Cassuto and Blaum. This new paradigm is motivated by the limitations of the reading process in high density data storage systems. They studied error correction in this new paradigm, specifically, the relationship between the minimum Hamming distance of an error correcting code and the minimum pair distance, which is the minimum Hamming distance between symbol-pair vectors derived from codewords of the code. It was proved that for a linear cyclic code with minimum Hamming distance d_H, the corresponding minimum pair distance is at least d_H+3. In this paper, we show that, for a given linear cyclic code with a minimum Hamming distance d_H, the minimum pair distance is at least $d_H + \lceil d_H/2 \rceil$. We then describe a decoding algorithm, based upon a bounded distance decoder for the cyclic code, whose symbol-pair error correcting capabilities reflect the larger minimum pair distance. Finally, we consider the case where the read channel output is a larger number, $b \geqslant 3$, of consecutive symbols, and we provide extensions of several concepts, results, and code constructions to this setting.**

***Index Terms*— Coding theory, codes for storage media, cyclic codes, symbol pairs.**

I. INTRODUCTION

THE TRADITIONAL approach in information theory to analyzing noisy channels involves parsing a message into individual information units, called symbols. Even though in many works the error correlation and interference between the symbols is studied, the process of writing and reading is usually assumed to be performed on individual symbols. However, in some of today's emerging storage technologies, as well as in some proposed for the future, this is no longer an accurate assumption and symbols can only be written and read in possibly overlapping groups. This brings us to study a model, recently proposed by Cassuto and Blaum [1], for channels whose outputs are overlapping pairs of symbols.

The rapid progress in high density data storage technologies paved the way for high capacity storage with reduced price. However, since the bit size at high densities is small, it is a challenge to successfully read the individual bits recorded on the storage medium; for more details, see [1]. The symbol-pair read channel model studied in [1], and later by Cassuto and Litsyn in [2], mimics the reading process of such storage technologies. In that model, the outputs produced by a sequence of read operations are (possibly corrupted) overlapping pairs of adjacent symbols, called *pair-read symbols*. For example, if the recorded sequence is (010), then in the absence of any noise the output of the symbol-pair read channel would be $[(01), (10), (00)]$. In this new paradigm, the errors are no longer individual symbol errors, but, rather, *symbol-pair errors*, where in a symbol-pair error at least one of the symbols is erroneous. The main task now becomes combating these symbol-pair errors by designing codes with large minimum symbol-pair distance.

The results in [1] and [2] addressed several fundamental questions regarding the pair-metric, as well as construction and decoding of codes with pair-error correction capability. Finite-length and asymptotic bounds on code sizes were also derived. These were extended in [3] and [4], where construction of maximum distance separable codes for the symbol-pair metric was considered, and in [5], where the authors studied syndrome decoding of symbol-pair codes. The paradigm of the symbol-pair channel studied in these prior works can be generalized to b-symbol read channels, where the result of a read operation is a consecutive sequence of $b > 2$ symbols. In essence, we receive b estimates of the same stored sequence. This insight connects the symbol-pair problem to the sequence reconstruction problem, which was first introduced by Levenshtein [8]–[10]. In the sequence reconstruction scenario, the same codeword is transmitted over multiple channels. Then, a decoder receives all channel outputs, which are assumed to be different from each other, and outputs an estimate of the transmitted codeword. The original motivation did not come from data storage but rather from other domains, such as molecular biology and chemistry, where the amount of redundancy in the information is too low and thus the only way to combat errors is by repeatedly transmitting the

Manuscript received April 8, 2015; revised September 29, 2015; accepted January 13, 2016. Date of publication January 27, 2016; date of current version March 16, 2016. This work was supported was supported in part by the ISEF Foundation, the Lester Deutsch Fellowship, the University of California Laboratory Fees Research Program, under Award 09-LR-06-118620-SIEP, in part by the National Science Foundation under Grant CCF-1116739 and Grant CCF-1405119, in part by the Center for Magnetic Recording Research, University of California at San Diego, and in part by the NSF Expeditions in Computing Program under Grant CCF-0832824. E. Yaakobi was supported by the Electrical Engineering Department, California Institute of Technology, Pasadena, CA, USA. P. H. Siegel was supported by Technion, within the Fellowship from the Lady Davis Foundation and by a Viterbi Research Fellowship. This paper was presented at the 2012 IEEE International Symposium on Information Theory, [19].

E. Yaakobi is with the Department of Computer Science, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: yaakobi@cs.technion.ac.il).

J. Bruck is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: bruck@caltech.edu).

P. H. Siegel is with the Center for Magnetic Recording Research, Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: psiegel@ucsd.edu).

Communicated by Y. Mao, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2016.2522434

same message. However, this model is very relevant for the advanced storage technologies mentioned above as well as in any other context where the stored information is read multiple times. Furthermore, we note that the model proposed by Levenshtein was recently studied and generalized, with applications to associative memories [18].

In the channel model described by Levenshtein, all channels are (almost) independent from each other, as it is only guaranteed that the channel outputs are all different. Assuming that the transmitted message $\boldsymbol{c}$ belongs to a code with minimum Hamming distance d_H and the number of errors in every channel can be strictly greater than $\lfloor \frac{d_H-1}{2} \rfloor$, Levenshtein studied the minimum number of channels that are necessary to construct a successful decoder. The corresponding value for the Hamming metric (as well as other distance metrics) was studied in [9]; extensions to distance metrics over permutations, e.g. [6], [7], and error graphs [11] have also been considered. Recently, the analogous problem has been addressed for the Grassmann graph and for permutations under Kendall's τ distance [20], and an information-theoretic study motivated by applications related to DNA sequencing was carried out for a special case of a channel with deletions [12], [13].

More specifically, for the Hamming distance, the following result was proved in [9]. Assume the transmitted word belongs to a code with minimum Hamming distance d_H and the number of errors, t, in every channel is greater than $\lfloor \frac{d_H-1}{2} \rfloor$. Then, in order to construct a successful decoder, the number of channels has to be greater than

$$\sum_{i=0}^{t-\lceil d_H/2 \rceil} \binom{n-d_H}{i} \sum_{k=i+d_H-t}^{t-i} \binom{d_H}{k}.$$

For example, if $t = \lfloor \frac{d_H-1}{2} \rfloor + 1$, i.e., only one more than the error correction capability, then the number of channels has to be at least $\binom{2t}{t} + 1$. Note that if $t > \lfloor \frac{d_H-1}{2} \rfloor + 1$ then this number is at least on the order of the message length n. This disappointing result is a consequence of the arbitrary errors that may occur in every channel. In practice, especially for storage systems, we can take advantage of the fact that the errors are more constrained in number in order to improve the error correction capability.

In the symbol-pair read channel, there are in fact two channels. If the stored information is $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$, then the corresponding *pair-read vector* of $\boldsymbol{x}$ is

$$\pi(\boldsymbol{x}) = [(x_0, x_1), (x_1, x_2), \ldots, (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)],$$

and the goal is to correct a large number of the so-called *symbol-pair errors*. With symbol alphabet Σ, the *pair distance*, $d_p(\boldsymbol{x}, \boldsymbol{y})$, between two pair-read vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is the Hamming distance over the symbol-pair alphabet $(\Sigma \times \Sigma)$ between their respective pair-read vectors, that is, $d_p(\boldsymbol{x}, \boldsymbol{y}) = d_H(\pi(\boldsymbol{x}), \pi(\boldsymbol{y}))$. Accordingly, the *minimum pair distance* of a code $\mathcal{C}$ is defined as $d_p(\mathcal{C}) = \min_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{C}, \boldsymbol{x} \neq \boldsymbol{y}} \{d_p(\boldsymbol{x}, \boldsymbol{y})\}$.

In [1], it was shown that for a linear cyclic code with minimum Hamming distance d_H, the minimum pair distance, d_p, satisfies $d_p \geqslant d_H + 3$. Our main contribution is the stronger result that

$$d_p \geqslant d_H + \left\lceil \frac{d_H}{2} \right\rceil.$$

According to [1], this permits the correction of $\left\lfloor \frac{d_p-1}{2} \right\rfloor$ symbol-pair errors. Thus, in contrast to Levenshtein's results on independent channels, on the symbol-pair read channel we can correct a large number of symbol-pair errors. In order to exploit this potentially much larger minimum pair distance guarantee, we explicitly construct a decoder, based upon a bounded distance decoder of the given linear cyclic code, that can correct a number of symbol-pair errors up to the decoding radius corresponding to this bound.

We then address the general paradigm of channels that sense some prescribed number, $b > 2$, of consecutive symbols on each read. First, some of the results of the symbol-pair read channel are generalized. Next, we study properties of codes for the b-symbol read channel that are constructed by interleaving b component codes. Finally, we examine the b-distance of two specific families of codes, namely the codebooks Σ^n and the linear cyclic Hamming codes.

The rest of the paper is organized as follows. In Section II, we formally review the symbol-pair read channel and some of its basic properties. In Section III, we show that linear cyclic codes can correct a large number of symbol-pair errors and in Section IV, a decoding algorithm for such codes is given. Section V generalizes some of the results on the symbol-pair read channel to b-symbol read channels, where $b > 2$. Finally, Section VI concludes the paper.

II. Definitions and Basic Properties

In this section, we review the symbol-pair read channel model introduced in [1]. If a length-n vector is stored in the memory then its pair-read vector is also a length-n vector in which every entry is a pair of cyclically consecutive symbols in the stored vector. More formally, if $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \Sigma^n$ is a length-n vector over some alphabet Σ, then the *symbol-pair read vector* of $\boldsymbol{x}$, denoted by $\pi(\boldsymbol{x})$, is defined to be

$$\pi(\boldsymbol{x}) = [(x_0, x_1), (x_1, x_2), \ldots, (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)].$$

Note that $\pi(\boldsymbol{x}) \in (\Sigma \times \Sigma)^n$, and for $\boldsymbol{x}, \boldsymbol{y} \in \Sigma$,

$$\pi(\boldsymbol{x} + \boldsymbol{y}) = \pi(\boldsymbol{x}) + \pi(\boldsymbol{y}).$$

We will focus on binary vectors, so $\Sigma = \{0, 1\}$ and unless stated otherwise, all indices are taken modulo n. The all-zeros, all-ones vector is denoted by $\mathbf{0}, \mathbf{1}$, respectively. The Hamming distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted by $d_H(\boldsymbol{x}, \boldsymbol{y})$. Similarly, the Hamming weight of a vector $\boldsymbol{x}$ is denoted by $w_H(\boldsymbol{x})$. The *pair distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted by $d_p(\boldsymbol{x}, \boldsymbol{y})$ and is defined to be

$$d_p(\boldsymbol{x}, \boldsymbol{y}) = d_H(\pi(\boldsymbol{x}), \pi(\boldsymbol{y})).$$

Accordingly, the *pair weight* of $\boldsymbol{x}$ is $w_p(\boldsymbol{x}) = w_H(\pi(\boldsymbol{x}))$. A *symbol-pair error* in the i-th symbol of $\pi(\boldsymbol{x})$ changes at least one of the two symbols (x_i, x_{i+1}). Note that the following connection between the pair distance and pair weight holds.

Proposition 1: For all $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^n$, $d_p(\boldsymbol{x}, \boldsymbol{y}) = w_p(\boldsymbol{x} + \boldsymbol{y})$.

Proof: Note that for $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^n$,

$$\begin{aligned} d_p(\boldsymbol{x}, \boldsymbol{y}) &= d_H(\pi(\boldsymbol{x}), \pi(\boldsymbol{y})) = w_H(\pi(\boldsymbol{x}) + \pi(\boldsymbol{y})) \\ &= w_H(\pi(\boldsymbol{x} + \boldsymbol{y})) = w_p(\boldsymbol{x} + \boldsymbol{y}). \end{aligned}$$

■

A first observation on the connection between the Hamming distance and pair distance was proved in [1].

Proposition 2 [1]: Let $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^n$ be such that $0 < d_H(\boldsymbol{x}, \boldsymbol{y}) < n$. Then,

$$d_H(\boldsymbol{x}, \boldsymbol{y}) + 1 \leqslant d_p(\boldsymbol{x}, \boldsymbol{y}) \leqslant 2d_H(\boldsymbol{x}, \boldsymbol{y}).$$

For a code $\mathcal{C}$, we denote its minimum Hamming distance by $d_H(\mathcal{C})$. The *symbol-pair code* of $\mathcal{C}$ is the code

$$\pi(\mathcal{C}) = \{\pi(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}\}.$$

Similarly, the *minimum pair distance* of $\mathcal{C}$, $d_p(\mathcal{C})$, is the minimum Hamming distance of $\pi(\mathcal{C})$, i.e.,

$$d_p(\mathcal{C}) = d_H(\pi(\mathcal{C})).$$

From Proposition 2, if $0 < d_H(\mathcal{C}) < n$ then the following connection between $d_H(\mathcal{C})$ and $d_p(\mathcal{C})$ is established [1]:

$$d_H(\mathcal{C}) + 1 \leqslant d_p(\mathcal{C}) \leqslant 2d_H(\mathcal{C}).$$

Example 1: In this example we choose the code $\mathcal{C}$ to be the single parity-check code of length three, that is,

$$\mathcal{C} = \{(0, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1)\}.$$

Thus, the symbol-pair code of $\mathcal{C}$ is

$$\pi(\mathcal{C}) = \{(00, 00, 00), (11, 10, 01), (10, 01, 11), (01, 11, 10)\}.$$

The minimum Hamming distance of $\mathcal{C}$ is $d_H(\mathcal{C}) = 2$, while the minimum pair distance of $\mathcal{C}$ is $d_p(\mathcal{C}) = 3$.

It was proved in [1] that if a code has minimum pair distance $d_p(\mathcal{C})$ then it can correct $\left\lfloor \frac{d_p(\mathcal{C})-1}{2} \right\rfloor$ symbol-pair errors. Therefore, the goal in constructing codes for the pair-read channel is to achieve high minimum pair distance with respect to the minimum Hamming distance. It was shown in [1] that interleaving two codes with minimum Hamming distance d_H generates a code with the same minimum Hamming distance d_H but with minimum pair distance $2d_H$. This construction generates codes with the largest possible minimum pair distance with respect to the minimum Hamming distance of the component codes. However, it is not particularly attractive as, in general, interleaving produces codes that suffer from a poor Hamming distance relative to their codeword length. In [3] and [4], maximum distance separable codes for the symbol-pair read channel were proposed, and in [5], the decoding of symbol-pair codes by syndrome decoding was studied.

Yet another interesting family of codes analyzed in [1] is the class of linear cyclic codes. For such a code, $\mathcal{C}$, with minimum Hamming distance d_H, it was proved that the minimum pair distance is at least $d_H + 2$. Using the Hartmann-Tzeng bound, this lower bound was improved to $d_H + 3$ when the code length is a prime number. Our main goal in the next section is to derive an improved lower bound on the minimum pair distance of linear cyclic codes.

III. The Pair Distance of Cyclic Codes

The goal of this section is to show that linear cyclic codes provide large minimum pair distance. In order to do so, we first give a method to determine the pair weight of a vector $\boldsymbol{x}$. A similar characterization of the pair weight was proved in [1] (Theorem 2).

The key observation is that if $x_i = 1$, then two symbol-pairs in $\pi(\boldsymbol{x})$, the $(i-1)$-st and i-th symbol-pairs must be non-zero. Of course, the condition $x_{i-1} = 1$ also causes the $(i-1)$-st symbol-pair to be non-zero. Hence, as we increment the index i, we can think of the condition $(x_{i-1}, x_i) = (0, 1)$ as contributing two new non-zero symbols to $\pi(\boldsymbol{x})$, whereas the condition $(x_{i-1}, x_i) = (1, 1)$ contributes only one. Therefore, in order to determine the weight of $\pi(\boldsymbol{x})$, one needs to determine the number of occurrences of the sequence $(x_{i-1}, x_i) = (0, 1)$ in the vector $\boldsymbol{x}$, which we next show how to do.

For $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$, we define

$$\boldsymbol{x}' = (x_0 + x_1, x_1 + x_2, \ldots, x_{n-1} + x_0). \quad (1)$$

The next lemma provides a characterization of the pair weight of a vector $\boldsymbol{x}$.

Lemma 1: For any $\boldsymbol{x} \in \Sigma^n$, $w_p(\boldsymbol{x}) = w_H(\boldsymbol{x}) + w_H(\boldsymbol{x}')/2$.

Proof: Let

$$\begin{aligned} S_0 &= \{i : (x_i, x_{i+1}) \neq (0, 0) \text{ and } x_i = 1\}, \\ S_1 &= \{i : (x_i, x_{i+1}) \neq (0, 0) \text{ and } x_i = 0\}. \end{aligned}$$

Hence, $|S_0| = w_H(\boldsymbol{x})$, $S_0 \cap S_1 = \emptyset$, and $w_p(\boldsymbol{x}) = |S_0| + |S_1|$. For all $0 \leqslant i \leqslant n-1$, $i \in S_1$ if and only if $x_i = 0$ and $x_{i+1} = 1$. In this case, $x'_i = x_i + x_{i+1} = 1$. Thus, we get

$$|S_1| = \left|\{i : x_i = 0 \text{ and } x'_i = 1\}\right|.$$

Note that for any $\boldsymbol{x} \in \Sigma^n$,

$$\left|\{i : x_i = 0 \text{ and } x'_i = 1\}\right| = \left|\{i : x_i = 1 \text{ and } x'_i = 1\}\right|,$$

and the sum of the cardinalities of these two sets is $w_H(\boldsymbol{x}')$. Therefore, $|S_1| = \frac{w_H(\boldsymbol{x}')}{2}$ and

$$w_p(\boldsymbol{x}) = |S_0| + |S_1| = w_H(\boldsymbol{x}) + \frac{w_H(\boldsymbol{x}')}{2}.$$

■

Using Lemma 1, we now derive an improved lower bound on the minimum pair distance of linear cyclic codes.

Theorem 1: Let $\mathcal{C}$ be a linear, cyclic code of dimension greater than one. Then,

$$d_p(\mathcal{C}) \geqslant d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

Proof: Let $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$ be a codeword in $\mathcal{C}$. Assume first that $\boldsymbol{x} \neq \mathbf{1}$. Since the code is cyclic, $(x_1, \ldots, x_{n-1}, x_0) \in \mathcal{C}$ and thus

$$\boldsymbol{x}' = (x_0, \ldots, x_{n-1}) + (x_1, \ldots, x_{n-1}, x_0) \in \mathcal{C}.$$

Note that the weight of $\boldsymbol{x}'$ is even and since $\boldsymbol{x} \neq \mathbf{1}$, we have $\boldsymbol{x}' \neq \mathbf{0}$. Hence $w_H(\boldsymbol{x}') \geqslant 2\lceil d_H(\mathcal{C})/2 \rceil$. Furthermore, $w_H(\boldsymbol{x}) \geqslant d_H(\mathcal{C})$. It follows that

$$w_p(\boldsymbol{x}) = w_H(\boldsymbol{x}) + w_H(\boldsymbol{x}')/2 \geqslant d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

Next, assume that $\boldsymbol{x} = \mathbf{1}$ is a codeword in $\mathcal{C}$, in which case $w_p(\boldsymbol{x}) = n$. We now show that if the dimension of $\mathcal{C}$ is greater than one, then $d_H(\mathcal{C}) \leqslant \lfloor 2n/3 \rfloor$. This will imply that

$$d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil \leqslant \lfloor 2n/3 \rfloor + \lceil n/3 \rceil = n = w_p(\boldsymbol{x})$$

from which the theorem will follow.

Since the dimension of $\mathcal{C}$ exceeds one, it contains at least three distinct non-zero codewords. Choose two of them, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. If it were true that $d_H(\mathcal{C}) > \lfloor 2n/3 \rfloor$, then it would follow that $w_H(\boldsymbol{x}_1), w_H(\boldsymbol{x}_2) \geqslant \lfloor 2n/3 \rfloor + 1$, implying that

$$w_H(\boldsymbol{x}_1 + \boldsymbol{x}_2) \leqslant 2n - (2 \cdot \lfloor 2n/3 \rfloor + 1) \leqslant \lfloor 2n/3 \rfloor,$$

which is a contradiction. Therefore, we conclude that $d_H(\mathcal{C}) \leqslant \lfloor 2n/3 \rfloor$, completing the proof. ■

IV. Decoding

From Theorem 1 we conclude that linear cyclic codes have large minimum pair distance, thereby permitting the correction of a large number of symbol-pair errors. It is therefore of interest to construct efficient decoders for these codes, which is the topic of this next section. First note that since these codes are linear, it was shown in [5] how a modified version of syndrome decoding can be used in order to correct symbol-pair errors within the error correction capability of the codes. However, since syndrome decoding suffers exponential complexity, our goal is to provide more efficient decoders whose complexity is of the same order as classical decoders for cyclic codes.

Let $\mathcal{C}$ be a linear cyclic code with minimum distance $d_H(\mathcal{C}) = 2t + 1$. Assume there is a decoder for $\mathcal{C}$ that can correct up to t errors. We will show how to use this decoder in the design of a decoder for the code $\pi(\mathcal{C})$ which corrects up to $t_0 = \lfloor \frac{3t+1}{2} \rfloor$ symbol-pair errors.

We assume that the dimension of $\mathcal{C}$ is greater than one. This condition implies, according to the proof of Theorem 1, that $d_H(\mathcal{C}) \leqslant \lfloor 2n/3 \rfloor$; that is, $2t + 1 \leqslant \lfloor 2n/3 \rfloor$, or

$$t \leqslant \left\lfloor \frac{\lfloor 2n/3 \rfloor - 1}{2} \right\rfloor < n/3.$$

It is straightforward to verify that $t_0 < n/2$.

We define the decoder as a mapping $\mathcal{D}_{\mathcal{C}} : \Sigma^n \to \mathcal{C} \cup \{F\}$, where F denotes a decoder failure. For a received word $\boldsymbol{y} \in \Sigma^n$ we write $\mathcal{D}_{\mathcal{C}}(\boldsymbol{y}) = \widehat{\boldsymbol{c}} \in \mathcal{C} \cup \{F\}$. If $\boldsymbol{c} \in \mathcal{C}$ is the transmitted word and $d_H(\boldsymbol{c}, \boldsymbol{y}) \leqslant t$, then it is guaranteed that $\widehat{\boldsymbol{c}} = \boldsymbol{c}$. However, if $d_H(\boldsymbol{c}, \boldsymbol{y}) > t$, then either $\widehat{\boldsymbol{c}}$ is a codeword different from $\boldsymbol{c}$, whose Hamming distance from the received word $\boldsymbol{y}$ is at most t, i.e., $d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}) \leqslant t$, or $\widehat{\boldsymbol{c}} = F$, indicating that no such codeword exists.

We now introduce another code that will play a role in the symbol-pair decoder design. The *double-repetition code* of $\mathcal{C}$ is the code defined by

$$\mathcal{C}_2 = \{(\boldsymbol{c}, \boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}\}.$$

Note that its length is $2n$ and its minimum Hamming distance satisfies $d_H(\mathcal{C}_2) = 2d_H(\mathcal{C})$. The code $\mathcal{C}_2$ can correct up to $2t$ errors and we assume that it has a decoder $\mathcal{D}_{\mathcal{C}_2} : \Sigma^n \times \Sigma^n \to \Sigma^n \cup \{F\}$. Every codeword in $\mathcal{C}_2$ consists of two identical codewords from $\mathcal{C}$ and thus, for simplicity of notation, we have assumed that when $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{c}})$ is a codeword at distance no more than $2t$ from a received word $(\boldsymbol{y}_1, \boldsymbol{y}_2)$, the decoder $\mathcal{D}_{\mathcal{C}_2}$ returns $\widehat{\boldsymbol{c}} \in \mathcal{C}$. We defer the explicit design of the decoder $\mathcal{D}_{\mathcal{C}_2}$ to the end of the section.

Consider a codeword $\boldsymbol{c} \in \mathcal{C}$ and let $\pi(\boldsymbol{c}) \in \pi(\mathcal{C})$ be its corresponding symbol-pair vector. Let $\boldsymbol{y} = \pi(\boldsymbol{c}) + \boldsymbol{e}$ be a received word, where $\boldsymbol{e} \in (\Sigma \times \Sigma)^n$ is the error vector with weight $w_H(\boldsymbol{e}) \leqslant t_0 = \lfloor \frac{3t+1}{2} \rfloor$. We will describe a decoder $\mathcal{D}_\pi : (\Sigma \times \Sigma)^n \to \{0, 1\}^n$ that can correct the error $\boldsymbol{e}$.

The received vector has the form

$$\boldsymbol{y} = \big((y_{0,0}, y_{0,1}), (y_{1,0}, y_{1,1}), \ldots, (y_{n-1,0}, y_{n-1,1})\big).$$

We define three related vectors:

$$\begin{aligned}
\boldsymbol{y}_L &= (y_{0,0}, \ldots, y_{n-1,0}),\\
\boldsymbol{y}_R &= (y_{0,1}, \ldots, y_{n-1,1}),\\
\boldsymbol{y}_S &= \boldsymbol{y}_L + \boldsymbol{y}_R\\
&= \big(y_{0,0} + y_{0,1}, \ldots, y_{n-1,0} + y_{n-1,1}\big).
\end{aligned}$$

Since the vector $\boldsymbol{y}$ suffers at most t_0 symbol-pair errors, the vectors $\boldsymbol{y}_L$ and $\boldsymbol{y}_R$ each have at most t_0 errors, as well. One can think of $\boldsymbol{y}_L$ as a noisy version of $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$, and $\boldsymbol{y}_R$ as a noisy version of a left-cyclic shift of $\boldsymbol{c}$, $(c_1, \ldots, c_{n-1}, c_0)$. Therefore, $\boldsymbol{y}_L$ and the right-cyclic shift of $\boldsymbol{y}_R$, $\boldsymbol{y}_R^{(1)} = (y_{n-1,1}, y_{0,1}, \ldots, y_{n-2,1})$, can be viewed as two noisy versions of the same codeword $\boldsymbol{c}$. Furthermore, the vector $\boldsymbol{y}_S$ has at most t_0 errors with respect to the codeword $\boldsymbol{c}' = (c_0 + c_1, \ldots, c_{n-1} + c_0)$. In general, the codeword $\boldsymbol{c}'$ does not uniquely determine the value of $\boldsymbol{c}$. However, we will now show that, in this setting, it does. This result, which we will make use of in the development of the decoding algorithm $\mathcal{D}_\pi$, is proved in the following lemma.

Lemma 2: In the symbol-pair read channel setting described above, if the codeword $\boldsymbol{c}' \in \mathcal{C}$ is successfully recovered, then we can uniquely determine the codeword $\boldsymbol{c}$.

Proof: Since the word $\boldsymbol{c}'$ is decoded successfully, we know the value of

$$\boldsymbol{c}' = (c'_0, c'_1, \ldots, c'_{n-1}) = (c_0 + c_1, c_1 + c_2, \ldots, c_{n-1} + c_0).$$

The codeword $\boldsymbol{c}$ satisfies $c_i = c_0 + \sum_{j=0}^{i-1} c'_j$. Hence if we define $\widetilde{\boldsymbol{c}} = [\widetilde{c}_0, \ldots, \widetilde{c}_{n-1}]$ by $\widetilde{c}_0 = 0$ and for $1 \leqslant i \leqslant n-1$, $\widetilde{c}_i = \sum_{j=0}^{i-1} c'_j$, then the codeword $\boldsymbol{c}$ is either $\widetilde{\boldsymbol{c}}$ or $\widetilde{\boldsymbol{c}} + \mathbf{1}$, depending on the value of c_0. The distance between $\boldsymbol{y}_L$ and $\boldsymbol{c}$ is at most t_0 and $d_H(\boldsymbol{c}, \boldsymbol{c} + \mathbf{1}) = n$. Recalling that $t_0 < n/2$, we have

$$d_H(\boldsymbol{y}_L, \boldsymbol{c} + 1) = n - t_0 > t_0.$$

Hence, if $d_H(\boldsymbol{y}_L, \widetilde{\boldsymbol{c}}) < d_H(\boldsymbol{y}_L, \widetilde{\boldsymbol{c}} + \mathbf{1})$, then $\boldsymbol{c} = \widetilde{\boldsymbol{c}}$; otherwise, $\boldsymbol{c} = \widetilde{\boldsymbol{c}} + \mathbf{1}$. In either case, we can recover the codeword $\boldsymbol{c}$. ■

For convenience, we denote the codeword $\boldsymbol{c}$ obtained from the codeword $\boldsymbol{c}'$ by the method of Lemma 5 as $\boldsymbol{c}'^*$; That is, $\boldsymbol{c}'^* = \boldsymbol{c}$.

The number of symbol-pair errors in the vector $\boldsymbol{y}$ is at most t_0. Each symbol-pair error corresponds to one or two-bit errors in the symbol-pair. We let E_1 be the number of single-bit symbol-pair errors and E_2 be the number of double-bit symbol-pair errors, where $E_1 + E_2 \leqslant t_0$. Thus, the number of errors in $\boldsymbol{y}_S$ is E_1 and the number of errors in $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$ is $E_1 + 2E_2$.

The following result will also play a role in the validation of the symbol-pair error decoding algorithm.

Lemma 3: If $\boldsymbol{c} \in \mathcal{C}$, $\boldsymbol{y} = \pi(\boldsymbol{c}) + \boldsymbol{e}$, and $w_H(\boldsymbol{e}) \leqslant t_0$, then either $\mathcal{D}_{\mathcal{C}}(\boldsymbol{y}_S) = \boldsymbol{c}'$ or $\mathcal{D}_{\mathcal{C}_2}((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})) = \boldsymbol{c}$.

Proof: If $E_1 \leqslant t$, then the decoder $\mathcal{D}_{\mathcal{C}}(\boldsymbol{y}_S)$ is successful. Otherwise, $E_1 \geqslant t+1$ and $E_2 \leqslant t_0 - (t+1)$, so the number of errors in $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$ satisfies

$$E_1 + 2E_2 \leqslant t_0 + t_0 - (t+1) = 2\left\lfloor \frac{3t+1}{2} \right\rfloor - (t+1) \leqslant 2t.$$

This implies that the decoder $\mathcal{D}_{\mathcal{C}_2}((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)}))$ is successful. ■

From Lemma 3, we know that at least one of the two decoders, $\mathcal{D}_{\mathcal{C}}$ and $\mathcal{D}_{\mathcal{C}_2}$, succeeds. However, it is not obvious which one of them does, and the main task of the algorithm underlying the decoder mapping $\mathcal{D}_\pi$, which we now describe, is to identify the successful decoder.

Given a received vector $\boldsymbol{y}$, the decoder output $\mathcal{D}_\pi(\boldsymbol{y}) = \widehat{\boldsymbol{c}}$ is calculated as follows.

Decoder $\mathcal{D}_\pi$:

Step 1. $\boldsymbol{c}_1 = \mathcal{D}_{\mathcal{C}}(\boldsymbol{y}_S)$, $e_1 = d_H(\boldsymbol{c}_1, \boldsymbol{y}_S)$.

Step 2. $\boldsymbol{c}_2 = \mathcal{D}_{\mathcal{C}_2}\big((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})\big)$, $e_2 = d_H((\boldsymbol{c}_2, \boldsymbol{c}_2), (\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)}))$.

Step 3. If $\boldsymbol{c}_1 = F$ or $w_H(\boldsymbol{c}_1)$ is odd then $\widehat{\boldsymbol{c}} = \boldsymbol{c}_2$.

Step 4. If $e_1 \leqslant \lfloor \frac{t+2}{2} \rfloor$, then $\widehat{\boldsymbol{c}} = \boldsymbol{c}_1^*$.

Step 5. If $e_1 > \lfloor \frac{t+2}{2} \rfloor$, let $e_1 = \lfloor \frac{t+2}{2} \rfloor + a$, $(1 \leqslant a \leqslant \lceil \frac{t}{2} \rceil - 1)$

- **a)** If $e_2 \leqslant t_0 + a$ then $\widehat{\boldsymbol{c}} = \boldsymbol{c}_2$,
- **b)** Otherwise, $\widehat{\boldsymbol{c}} = \boldsymbol{c}_1^*$.

The correctness of the decoder is proved in the next theorem.

Theorem 2: The decoder output satisfies $\mathcal{D}_\pi(\boldsymbol{y}) = \widehat{\boldsymbol{c}} = \boldsymbol{c}$.

Proof: According to Lemma 3, at least one of the two decoders in Steps 1 and 2 succeeds. Steps 3–5 help to determine which of the two decoders succeeds.

Step 3: Since $\boldsymbol{y}_S$ is a noisy version of the codeword $\boldsymbol{c}'$, the decoding operation in Step 1 attempts to decode $\boldsymbol{c}'$, which, we recall, has even weight. If either $\boldsymbol{c}_1 = F$ or the Hamming weight of $\boldsymbol{c}_1$ is odd, then this decoding operation necessarily fails, implying that the decoding operation in Step 2 was successful. If we reach Steps 4 and 5 then $w_H(\boldsymbol{c}_1)$ must be even.

Step 4: We now show that if $e_1 \leqslant \lfloor \frac{t+2}{2} \rfloor$, then $E_1 \leqslant \lfloor \frac{t+2}{2} \rfloor$ as well, and therefore the decoding operation in Step 1 succeeded. In order to see this, first note that the minimum Hamming distance of the code $\mathcal{C}$ is $2t+1$. If there is a miscorrection in Step 1, then the word $\boldsymbol{y}_S$ is miscorrected to some codeword of even weight. The weight of the error vector found in Step 1, e_1, is at most $\lfloor \frac{t+2}{2} \rfloor$. Since the minimum Hamming distance of the code $\mathcal{C}$ is $2t+1$, the number E_1 of actual errors in $\boldsymbol{y}_S$ satisfies

$$E_1 \geqslant 2t + 2 - \left\lfloor \frac{t+2}{2} \right\rfloor = \left\lceil \frac{3t}{2} \right\rceil + 1 > t_0,$$

contradicting the fact that the number of errors in $\boldsymbol{y}_S$ is at most t_0. Therefore, the condition on e_1 implies that the decoding operation $\boldsymbol{c}_1 = \mathcal{D}_{\mathcal{C}}(\boldsymbol{y}_S) = \boldsymbol{c}'$ succeeds, and according to Lemma 2, we can conclude that

$$\widehat{\boldsymbol{c}} = \boldsymbol{c}_1^* = \boldsymbol{c}'^* = \boldsymbol{c}.$$

Step 5: We are left with the case where $e_1 > \lfloor \frac{t+2}{2} \rfloor$. Since $e_1 \leqslant t$, we can write $e_1 = \lfloor \frac{t+2}{2} \rfloor + a$, where $1 \leqslant a \leqslant \lceil \frac{t}{2} \rceil - 1$.

Assume the decoding in Step 2 fails. Then, according to Lemma 3, the decoding operation $\boldsymbol{c}_1 = \mathcal{D}_{\mathcal{C}}(\boldsymbol{y}_S)$ succeeds, implying that

$$E_1 = e_1 = \left\lfloor \frac{t+2}{2} \right\rfloor + a.$$

The value of E_2 then satisfies

$$E_2 \leqslant t_0 - \left(\left\lfloor \frac{t+2}{2} \right\rfloor + a\right) = \left\lfloor \frac{3t+1}{2} \right\rfloor - \left\lfloor \frac{t+2}{2} \right\rfloor - a \leqslant t - a.$$

The total number of errors in $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$ is

$$E_1 + 2E_2 \leqslant t_0 + t - a = \left\lfloor \frac{5t+1}{2} \right\rfloor - a.$$

Since the decoder $\mathcal{D}_{\mathcal{C}_2}((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)}))$ fails and the minimum distance of $\mathcal{C}_2$ is $4t+2$, it follows that the weight of the error vector in Step 2, e_2, must satisfy

$$e_2 \geqslant 4t + 2 - (E_1 + 2E_2) \geqslant 4t + 2 - \left(\left\lfloor \frac{5t+1}{2} \right\rfloor - a\right)$$
$$\geqslant \left\lfloor \frac{3t+1}{2} \right\rfloor + a + 1 = t_0 + a + 1.$$

Hence, we conclude that if $e_2 \leqslant t_0 + a$, then the decoder in Step 2 must succeed.

Alternatively, assume the decoding in Step 1 fails. As in Step 4, this means that the number of errors E_1 in $\boldsymbol{y}_S$ is at least

$$E_1 \geqslant 2t + 2 - \left(\left\lfloor \frac{t+2}{2} \right\rfloor + a\right) = \left\lceil \frac{3t}{2} \right\rceil - (a-1) = t_0 - (a-1).$$

Since $E_1 + E_2 \leqslant t_0$, it follows that E_2 satisfies $0 \leqslant E_2 \leqslant a-1$, and $E_1 + 2E_2$, the total number of errors in $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$, satisfies

$$t_0 - (a-1) \leqslant E_1 + 2E_2 = (E_1 + E_2) + E_2 \leqslant t_0 + a - 1.$$

Thus, the decoding operation $\mathcal{D}_{\mathcal{C}_2}((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)}))$ succeeds, and

$$t_0 - (a-1) \leqslant e_2 \leqslant t_0 + a - 1.$$

Hence, if $e_2 > t_0 + a$, then the decoder in Step 1 must succeed. That completes the explanation of the assignments in a) and b) of Step 5. ■

We demonstrate the decoding algorithm in the following example.

Example 2: In this example we choose the code $\mathcal{C}$ to be the cyclic binary triple-error correcting BCH code of length 15, so $d_H(\mathcal{C}) = 7$, $d_p(\mathcal{C}) = 7 + \lceil \frac{7}{2} \rceil = 11$, $t = 3$, and $t_0 = 5$. Thus the code can correct 5 symbol-pair errors. Assume the stored word is the all zeros codeword $\mathbf{0}$.

Let $\boldsymbol{y}$ be the received vector

$$\boldsymbol{y} = (00, 11, 10, 00, 00, 00, 11, 00, 10, 00, 00, 11, 00, 00, 00).$$

Then,

$$\begin{aligned} \boldsymbol{y}_L &= (0,1,1,0,0,0,1,0,1,0,0,1,0,0,0), \\ \boldsymbol{y}_R &= (0,1,0,0,0,0,1,0,0,0,0,1,0,0,0), \\ \boldsymbol{y}_S &= (0,0,1,0,0,0,0,0,1,0,0,0,0,0,0), \end{aligned}$$

and

$$\boldsymbol{y}_R^{(1)} = (0,0,1,0,0,0,0,1,0,0,0,0,1,0,0).$$

In Step 1 of the decoding algorithm we calculate

$$\boldsymbol{c}_1 = \mathcal{D}_{\mathcal{C}}(\boldsymbol{y}_S), e_1 = d_H(\boldsymbol{c}_1, \boldsymbol{y}_S).$$

Since $\boldsymbol{y}_S$ suffered two errors and the decoder $\mathcal{D}_{\mathcal{C}}$ can decode at most three errors, we get $\boldsymbol{c}_1 = \mathbf{0}$ and $e_1 = 2$. In Step 2 we calculate

$$\boldsymbol{c}_2 = \mathcal{D}_{\mathcal{C}_2}\big((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})\big), e_2 = d_H((\boldsymbol{c}_2, \boldsymbol{c}_2), (\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})).$$

The word $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$ suffered eight errors and since the code $\mathcal{C}_2$ has minimum distance 14, the decoder $\mathcal{D}_{\mathcal{C}_2}$ can successfully correct at most six errors. Hence, the output $\boldsymbol{c}_2$ is either F, indicating there is no codeword of distance at most six from $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$, or some codeword $\boldsymbol{c}_2$ such that $e_2 = 6$. The condition in Step 3 fails, but the condition in Step 4 holds because $e_1 = 2 = \lfloor \frac{3+2}{2} \rfloor = 2$. Therefore, we conclude that the decoder in Step 1 succeeds and we can decode the codeword as $\hat{\boldsymbol{c}} = \mathbf{0}^* = \mathbf{0}$. Note that the operation $\mathbf{0}^*$ can result in either $\mathbf{0}$ or $\mathbf{1}$, but we can eliminate the word $\mathbf{1}$ as its distance from the received word is too large.

As another example, consider the received vector $\boldsymbol{y}$ given by

$$\boldsymbol{y} = (10,00,01,00,10,00,00,00,00,10,00,00,00,10,00),$$

with associated vectors

$$\begin{aligned} \boldsymbol{y}_L &= (1,0,0,0,0,0,0,0,0,1,0,0,0,1,0), \\ \boldsymbol{y}_R &= (0,0,1,0,1,0,0,0,0,0,0,0,0,0,0), \\ \boldsymbol{y}_S &= (1,0,1,0,1,0,0,0,0,1,0,0,0,1,0), \\ \boldsymbol{y}_R^{(1)} &= (0,0,0,1,0,1,0,0,0,0,0,0,0,0,0). \end{aligned}$$

The word $\boldsymbol{y}_S$ suffered five errors, so in Step 1 the decoded word $\boldsymbol{c}_1$ is either the failure symbol F or some codeword of weight seven or eight with distance either two or three from $\boldsymbol{y}_S$, respectively. Let us assume for this example that $\boldsymbol{c}_1$ is a codeword of weight eight and $e_1 = 3$.

The input word $(\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})$ suffered five errors; therefore, in Step 2 the decoder operation $\mathcal{D}_{\mathcal{C}_2}\big((\boldsymbol{y}_L, \boldsymbol{y}_R^{(1)})\big)$ succeeds, so $\boldsymbol{c}_2 = \mathbf{0}$ and $e_2 = 5$. Now the conditions in Steps 3 and 4 do not hold, so Step 5 will determine which decoder succeeds. First, we see that $a = 1$ and so $e_2 = 5 < 5 + 1 = t_0 + a$. Hence, the condition in Step 5a) holds and we conclude that the second decoder succeeds, i.e., $\hat{\boldsymbol{c}} = \mathbf{0}$.

To complete the decoder presentation, we return to the construction of the decoder $\mathcal{D}_{\mathcal{C}_2}$. This decoder receives two vectors, $\boldsymbol{y}_1 = (y_{1,0}, \ldots, y_{1,n-1})$ and $\boldsymbol{y}_2 = (y_{2,0}, \ldots, y_{2,n-1})$. Each is a noisy version of some codeword $\boldsymbol{c} \in \mathcal{C}$, and the goal is to correct a total of $2t$ errors in the two vectors. We define the vector $\widetilde{\boldsymbol{y}} = (\widetilde{y}_0, \ldots, \widetilde{y}_{n-1})$ such that for all $0 \leqslant i \leqslant n-1$, $\widetilde{y}_i = y_{1,i}$ if $y_{1,i} = y_{2,i}$, and otherwise $\widetilde{y}_i = ?$ to indicate an erasure. If the number of errors in $\widetilde{\boldsymbol{y}}$ is τ and the number of erasures is ρ, then we have $2\tau + \rho \leqslant 2t = d(\mathcal{C}) - 1$, which is within the error and erasure correcting capability of $\mathcal{C}$. We are left only with the problem of defining a decoder that corrects errors and erasures for cyclic codes. For that, we refer the reader to, for example, [14], [16]. Alternatively, we can treat the code $\mathcal{C}_2$ as a concatenated code where the inner code is simply the repetition code of length two. A general technique for decoding concatenated codes is described in [15, Ch. 12].

A symbol-pair error can change the value of either a single bit or both bits in a pair-read symbol. However, the knowledge on the maximum number of symbol-pair errors from each kind may be known in advance. For example, a symbol-pair error which corrupts only a single bit can be a result of a bit that was written erroneously to the media, while the two bits may be in error as a result of a reading noise. Thus, we consider codes that distinguish between these two types of errors. Specifically, we say that a code is a (t_1, t_2) *symbol-pair error-correcting code* if it can correct up to t_1 single-bit symbol-pair errors and up to t_2 double-bit symbol-pair errors.

Our discussion of (t_1, t_2) symbol-pair error-correcting codes uses as a point of departure a given binary cyclic linear code $\mathcal{C}$ with minimum Hamming distance $d_H(\mathcal{C})$ and a decoder $\mathcal{D}_{\mathcal{C}}$. The next theorem provides a condition on $d_H(\mathcal{C})$ that implies the code $\mathcal{C}$ is a (t_1, t_2) symbol-pair error-correcting code.

Theorem 3: A binary linear cyclic code $\mathcal{C}$ of length n and minimum distance $d_H(\mathcal{C})$ is a (t_1, t_2) symbol-pair error-correcting code if

$$d_H(\mathcal{C}) \geqslant \min\{t_1 + 2t_2 + 1, 2t_1 + 1\},$$

and $t_1 + t_2 < n/2$.

Proof: We now examine two different approaches to correct such errors.

1) The first approach uses the decoder $\mathcal{D}_{\mathcal{C}_2}$ of the double-repetition code of $\mathcal{C}$, as introduced earlier in this section. This decoder will need to correct $t_1 + 2t_2$ errors and therefore

$$d_H(\mathcal{C}_2) = 2d_H(\mathcal{C}) \geqslant 2(t_1 + 2t_2) + 1,$$

or

$$d_H(\mathcal{C}) \geqslant t_1 + 2t_2 + 1.$$

2) The second approach uses the decoder $\mathcal{D}_{\mathcal{C}}$ that we applied to the vector $\boldsymbol{y}_S$, also previously described. This decoder is required to correct t_1 errors and thus $d_H(\mathcal{C}) = 2t_1 + 1$. Note that according to Lemma 2, the condition $t_1 + t_2 < n/2$ guarantees a successful recovery of the stored codeword $\boldsymbol{c}$ based upon the decoding of $\boldsymbol{c}'$.

We conclude that if the condition in the theorem statement holds, we can choose either of these two approaches as the basis for a successful decoder. ■

From Theorem 3, we conclude that knowing the values of t_1 and t_2 significantly simplifies the decoder operation since we know which of the two decoders to apply in order to correct the symbol-pair errors. However, this knowledge may also reduce the lower bound on the required minimum Hamming distance of the code $\mathcal{C}$, and thus the required code redundancy. To see this, assume that we construct a (t_1, t_2) symbol-pair error-correcting code using a linear cyclic code $\mathcal{C}$ which corrects t_1+t_2 symbol-pair errors. Then, according to Theorem 1, its minimum Hamming distance $d_H(\mathcal{C})$ is required to satisfy

$$d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil \geqslant 2(t_1 + t_2) + 1. \quad (2)$$

On the other hand, according to Theorem 3, $d_H(\mathcal{C})$ must be greater than or equal to $\min\{t_1 + 2t_2 + 1, 2t_1 + 1\}$. It is straightforward to verify that for any two non-negative integers t_1 and t_2, this lower bound on $d_H(\mathcal{C})$ is no greater than the lower bound on $d_H(\mathcal{C})$ implied by (2).

To conclude the discussion in this section, we note that the presented decoder $\mathcal{D}_\pi$ uses two decoders: the first one is $\mathcal{D}_\mathcal{C}$ for the code $\mathcal{C}$ and the second is $\mathcal{D}_{\mathcal{C}_2}$ for the code $\mathcal{C}_2$. Since there is no specific requirement for these decoders besides their error and erasure correction capability, we can use any of the existing decoders for cyclic codes. Hence, the complexity of the decoder $\mathcal{D}_\pi$ will be of the same order as that of the best decoders for cyclic codes. This provides a significant improvement upon the syndrome-type decoder for linear codes which was given in [5].

Finally, we note that a decoding approach similar to the one discussed above was presented in [17]. The goal in [17] was to use binary linear cyclic error-correcting codes in order to detect and correct multiple binary bursts whose total weight is limited. The decoder design exploited the property that, in a linear cyclic code, if $\boldsymbol{c}$ is a codeword then so is $\boldsymbol{c}'$. Thus, decoding of at least $\boldsymbol{c}$ and $\boldsymbol{c}'$ has to succeed.

In the next section, we extend some of our results on the symbol-pair read channel to b-symbol read channels, with $b \geqslant 3$.

V. Extensions to b-Symbol Read Channel

In this section, we study the b-symbol read channel, $3 \leqslant b < n$, where b symbols are sensed in each read operation. First, we formally define the b-symbol read channel model and introduce the b-symbol distance function. We then prove some of their basic properties, along the lines of those of the symbol-pair read channel model and symbol-pair distance. Next, we turn to constructions of codes for the b-symbol read channel. Generalizing results in [1], we analyze the b-symbol distance properties of codes obtained by interleaving b component codes, and then describe a decoding algorithm that decodes up to the decoding radius. Finally, we study b-symbol properties of two specific families of codes, namely the codes corresponding to the entire space of codewords, Σ^n, and the linear cyclic Hamming codes of length $n = 2^m - 1$, $m \geqslant 3$.

A. Basic Properties

For $b \geqslant 3$, the *b-symbol read vector* corresponding to the vector $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1}) \in \Sigma^n$ is defined as

$$\pi_b(\boldsymbol{x}) = [(x_0, \ldots, x_{b-1}), \ldots, (x_{n-1}, x_0, \ldots, x_{b-2})] \in \left(\Sigma^b\right)^n.$$

We refer to the elements of $\pi_b(\boldsymbol{x})$ as *b-symbols*. The *b-symbol distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted by $d_b(\boldsymbol{x}, \boldsymbol{y})$, is defined as

$$d_b(\boldsymbol{x}, \boldsymbol{y}) = d_H(\pi_b(\boldsymbol{x}), \pi_b(\boldsymbol{y})).$$

For simplicity, we sometimes refer to this as the *b-distance*. Similarly, we define the *b-weight* of the vector $\boldsymbol{x}$ as $w_H(\pi_b(\boldsymbol{x}))$. As was the case for the symbol-pair distance, it is easy to see that $d_b(\boldsymbol{x}, \boldsymbol{y}) = w_b(\boldsymbol{x} + \boldsymbol{y})$, for all $\boldsymbol{x}, \boldsymbol{y}$. For convenience of notation, if the i-th symbol $\pi_b(\boldsymbol{x})_i$ of $\pi_b(\boldsymbol{x})$ is not the all-zero b-tuple, we will write $\pi_b(\boldsymbol{x})_i \neq 0$.

The following proposition is a natural generalization of Proposition 2.

Proposition 3: Let $\boldsymbol{x} \in \Sigma^n$ be such that $0 < w_H(\boldsymbol{x}) \leqslant n - (b-1)$. Then,

$$w_H(\boldsymbol{x}) + b - 1 \leqslant w_b(\boldsymbol{x}) \leqslant b \cdot w_H(\boldsymbol{x}).$$

Proof: Let $S = \{i : x_i \neq 0\}$, so $|S| = w_H(\boldsymbol{x})$. Since every symbol $x_i \in \boldsymbol{x}$ appears in b elements of $\pi_b(\boldsymbol{x})$, the number of non-zero b-symbols in $\pi_b(\boldsymbol{x})$ is at most $|S| \cdot b = w_H(\boldsymbol{x}) \cdot b$, implying the upper bound $w_b(\boldsymbol{x}) \leqslant b \cdot w_H(\boldsymbol{x})$.

To prove the lower bound, we consider two cases. First, assume that $\boldsymbol{x}$ has no set of b or more consecutive zeros. Then, for every index i, we have $\pi_b(\boldsymbol{x})_i \neq 0$, and thus $w_b(\boldsymbol{x}) = n$. This implies that

$$w_H(\boldsymbol{x}) + b - 1 \leqslant n - (b-1) + (b-1) = n = w_b(\boldsymbol{x}).$$

Next, consider the case where $\boldsymbol{x}$ does contain a sequence of at least b consecutive zeros. Since, by assumption, $w_H(\boldsymbol{x}) > 0$, we can find an index k such that $x_k = x_{k+1} = \cdots = x_{k+b-1} = 0$ and $x_{k+b} = 1$. Therefore, for all $k+1 \leqslant i \leqslant k+b-1$, we have $\pi_b(\boldsymbol{x})_i \neq 0$. Furthermore, for all i such that $x_i = 1$, we also have $\pi_b(\boldsymbol{x})_i \neq 0$. It follows that

$$w_b(\boldsymbol{x}) \geqslant w_H(\boldsymbol{x}) + b - 1.$$

■

Our next goal is to suitably generalize Lemma 1, giving a useful characterization of $w_b(\boldsymbol{x})$ for arbitrary $b \geqslant 3$. In order to do this, we introduce for every $\boldsymbol{x} \in \Sigma^n$ an auxiliary vector $\widehat{\boldsymbol{x}} \in \Sigma^n$, obtained from $\boldsymbol{x}$ by inverting every sequence of $b-2$ or fewer consecutive zeros in $\boldsymbol{x}$. More formally, we define $\widehat{\boldsymbol{x}}$ as follows. If $(x_i, x_{i+1}, \ldots, x_{i+k}, x_{i+k+1}) = (1, 0, \ldots, 0, 1)$ for some $0 \leqslant i \leqslant n-1$ and $k \leqslant b-2$, then $\widehat{x}_j = 1 - x_j = 1$ for $i+1 \leqslant j \leqslant i+k$. For all other values of j, $\widehat{x}_j = x_j$.

Example 3: Assume $b = 4$ and let $\boldsymbol{x} = (0, 1, 1, 0, 0, 0, 1, 0)$. Then

$$\widehat{\boldsymbol{x}} = (1, 1, 1, 0, 0, 0, 1, 1).$$

Note that the sequence of consecutive zeros beginning at position 3 has length $b - 1 = 3$ and therefore these zeros remain unchanged in $\widehat{\boldsymbol{x}}$, while the sequence of cyclically consecutive zeros beginning at position 7 has length $2 < b - 1 = 3$ and therefore these zeros are changed to ones.

Now, we state and prove the generalization of Lemma 1 for $b \geqslant 3$.

Lemma 4: For any $\boldsymbol{x} \in \Sigma^n$ and positive integer $b \geqslant 3$,

$$w_b(\boldsymbol{x}) = w_H(\widehat{\boldsymbol{x}}) + (b-1) \cdot \frac{w_H(\widehat{\boldsymbol{x}}')}{2}.$$

Proof: Let us first show that $w_b(\boldsymbol{x}) = w_b(\widehat{\boldsymbol{x}})$. The only positions j for which $\pi_b(\boldsymbol{x})_j$ and $\pi_b(\widehat{\boldsymbol{x}})_j$ differ are those for which $\pi_b(\boldsymbol{x})_j = (x_j, \ldots, x_{j+b-1})$ contains a zero bit in a length-$(k+2)$ sequence of the form

$$(x_i, x_{i+1}, \ldots, x_{i+k}, x_{i+k+1}) = (1, 0, \ldots, 0, 1)$$

where $k \leqslant b-2$. The zero bits $x_{i+1}, \ldots, x_{i+k}$ appear in the j-th symbol of $\pi_b(\boldsymbol{x})$ for $i-b+2 \leqslant j \leqslant i+k$. However, since $x_i = x_{i+k+1} = 1$, in this range of values of j, we see that both $\pi_b(\boldsymbol{x})_j \neq 0$ and $\pi_b(\widehat{\boldsymbol{x}})_j \neq 0$. For all other positions j, the corresponding bits of $\pi_b(\boldsymbol{x})_j$ and $\pi_b(\widehat{\boldsymbol{x}})_j$ are the same and thus $\pi_b(\boldsymbol{x})_j \neq 0$ if and only if $\pi_b(\widehat{\boldsymbol{x}})_j \neq 0$.

Next, we determine the value of $w_b(\widehat{\boldsymbol{x}})$, making use of the fact that any sequence of consecutive zeros in $\widehat{\boldsymbol{x}}$ has length at least $b-1$. Let

$$\begin{aligned}
S_0 &= \{i : \pi_b(\widehat{\boldsymbol{x}})_i \neq 0, \widehat{x}_i = 1\}, \\
S_1 &= \{i : \pi_b(\widehat{\boldsymbol{x}})_i \neq 0, \widehat{x}_i = 0, \widehat{x}_{i+1} = 1\}, \\
&\vdots \\
S_{b-2} &= \{i : \pi_b(\widehat{\boldsymbol{x}})_i \neq 0, \widehat{x}_i = \cdots = \widehat{x}_{i+b-3} = 0, \widehat{x}_{i+b-2} = 1\}, \\
S_{b-1} &= \{i : \pi_b(\widehat{\boldsymbol{x}})_i \neq 0, \widehat{x}_i = \cdots = \widehat{x}_{i+b-2} = 0, \widehat{x}_{i+b-1} = 1\}.
\end{aligned}$$

Clearly, $w_H(\widehat{\boldsymbol{x}}) = |S_0|$ and, since $S_j \cap S_\ell = \emptyset$, for all $0 \leqslant j < \ell \leqslant b-1$, we also have

$$w_b(\widehat{\boldsymbol{x}}) = |\cup_{i=0}^{b-1} S_i| = \sum_{i=0}^{b-1} |S_i|.$$

Let us now show that for all $2 \leqslant \ell \leqslant b-1$, $|S_1| = |S_\ell|$. If $i \in S_1$ then $(\widehat{x}_i, \widehat{x}_{i+1}) = (0, 1)$. Since there is no sequence of less than $b-1$ consecutive zeros

$$(\widehat{x}_{i-(b-2)}, \ldots, \widehat{x}_i, \widehat{x}_{i+1}) = (0, \ldots, 0, 1)$$

and thus $i - (\ell - 1) \in S_\ell$. Hence, $|S_\ell| \geqslant |S_1|$. For the opposite inequality, note that if $i \in S_\ell$, $\ell \geqslant 1$, then

$$(\widehat{x}_i, \widehat{x}_{i+1}, \ldots, \widehat{x}_{i+\ell-1}, \widehat{x}_{i+\ell}) = (0, \ldots, 0, 1).$$

Therefore $(\widehat{x}_{i+\ell-1}, \widehat{x}_{i+\ell}) = (0, 1)$, so $i + \ell - 1 \in S_1$, implying that $|S_1| \geqslant |S_\ell|$. Hence, $|S_1| = |S_\ell|$ for all $2 \leqslant \ell \leqslant b-1$. Remember that the vector $\widehat{\boldsymbol{x}}'$ is defined according to (1) to be the vector

$$\widehat{\boldsymbol{x}}' = (\widehat{x}_0 + \widehat{x}_1, \widehat{x}_1 + \widehat{x}_2, \ldots, \widehat{x}_{n-1} + \widehat{x}_0),$$

and hence as in the proof of Lemma 1, $|S_1| = \frac{w_H(\widehat{\boldsymbol{x}}')}{2}$, so we can conclude that

$$w_b(\widehat{\boldsymbol{x}}) = \sum_{\ell=0}^{b-1} |S_i| = w_H(\widehat{\boldsymbol{x}}) + (b-1) \cdot \frac{w_H(\widehat{\boldsymbol{x}}')}{2}.$$

■

Example 4: Assume $b = 4$ and let $\boldsymbol{x} = (1, 0, 0, 0, 1, 0, 0, 1)$. Then $w_H(\boldsymbol{x}) = 3$,

$$\pi_4(\boldsymbol{x}) = [1000, 0001, 0010, 0100, 1001, 0011, 0110, 1100],$$

and, therefore, $w_4(\boldsymbol{x}) = 8$. It is easy to verify that the inequalities in Proposition 3 hold.

We also see that

$$\widehat{\boldsymbol{x}} = [1, 0, 0, 0, 1, 1, 1, 1]$$

and

$$\widehat{\boldsymbol{x}}' = [1, 0, 0, 1, 0, 0, 0, 0],$$

so $w_H(\widehat{\boldsymbol{x}}) = 5$ and $w_H(\widehat{\boldsymbol{x}}') = 2$. The relationship in Lemma 4 is clearly seen to hold.

In analogy to the definition of symbol-pair codes, we define the *b-symbol read code* of a code $\mathcal{C}$ over Σ to be the code $\pi_b(\mathcal{C}) = \{\pi_b(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}\}$ over Σ^b. The *minimum b-distance* of $\mathcal{C}$, $d_b(\mathcal{C})$, is given by $d_b(\mathcal{C}) = d_H(\pi_b(\mathcal{C}))$, where the Hamming distance is over the alphabet Σ^b. Referring to Proposition 3, we see that if $0 < d_H(\mathcal{C}) \leqslant n - (b-1)$, then

$$d_H(\mathcal{C}) + b - 1 \leqslant d_b(\mathcal{C}) \leqslant b \cdot d_H(\mathcal{C}). \tag{3}$$

A *b-symbol error* in the i-th symbol of $\pi_b(\boldsymbol{c})$ changes at least one of the b symbols in the vector $\pi_b(\boldsymbol{c})_i = (x_i, \ldots, x_{i+b-1})$. Using the same reasoning as in the proof of Proposition 3 in [1], we see that a code $\mathcal{C}$ can correct any t b-symbol errors if $d_b(\mathcal{C}) \geqslant 2t + 1$.

In the next section, we study the b-distance of a code constructed by interleaving. We note that an extension of Theorem 1 is not straightforward to derive in this case. Namely, if $\boldsymbol{c}$ is a codeword in a linear cyclic code $\mathcal{C}$, then the vectors $\widehat{\boldsymbol{c}}$ and $\widehat{\boldsymbol{c}}'$ do not necessarily belong to the code $\mathcal{C}$, and hence it is not possible to use Lemma 4 to derive a bound on the minimum b-distance of a binary cyclic code.

B. Code Construction by Interleaving

The interleaving scheme studied in [1] generates codes $\mathcal{C}$ that satisfy $d_p(\mathcal{C}) = 2d_H(\mathcal{C})$. We will next show how this construction can be generalized for arbitrary $b \geqslant 3$, so we generate codes which satisfy $d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C})$. This result approves also that the upper bound on the minimum b-distance stated in (3) is tight. The standard notation of (n, M, d) will be used to denote the parameters of a binary code of length n, size M, and minimum distance d. Given a collection of b codes $\mathcal{C}_0, \ldots, \mathcal{C}_{b-1}$, where for $0 \leqslant i \leqslant b-1$ the code $\mathcal{C}_i$ has parameters (n, M_i, d_i), their interleaved code $\mathcal{C}$ is a $(bn, \Pi_{i=0}^{b-1} M_i, \min_{0 \leqslant i \leqslant b-1}\{d_i\})$ code defined as follows:

$$\begin{aligned}
\mathcal{C} = \{&(c_{0,0}, \ldots, c_{b-1,0}, c_{0,1}, \ldots, c_{b-1,1}, \ldots, c_{0,n-1}, \ldots, \\
&c_{b-1,n-1}) : \boldsymbol{c}_i = (c_{i,0}, \ldots, c_{i,n-1}) \in \mathcal{C}_i, \text{ for } 0 \leqslant i \leqslant b-1\}.
\end{aligned}$$

Theorem 4: Let $\mathcal{C}_0, \ldots, \mathcal{C}_{b-1}$ be a set of b binary codes with respective parameters (n, M_i, d_i), for $0 \leqslant i \leqslant b-1$. Then their interleaved code $\mathcal{C}$ satisfies

$$d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C}) = b \cdot \min_{0 \leqslant i \leqslant b-1} \{d_i\}.$$

Proof: Every codeword in $\boldsymbol{c} \in \mathcal{C}$ has the form

$$\begin{aligned}
\boldsymbol{c} = (&c_{0,0}, \ldots, c_{b-1,0}, c_{0,1}, \ldots, c_{b-1,1}, \ldots, \\
&c_{0,n-1}, \ldots, c_{b-1,n-1}),
\end{aligned}$$

obtained by interleaving codewords

$$\boldsymbol{c}_i = (c_{i,0}, \ldots, c_{i,n-1}) \in \mathcal{C}_i, 0 \leqslant i \leqslant b-1.$$

If $\boldsymbol{c} \neq \boldsymbol{0}$, then $\boldsymbol{c}_i \neq \boldsymbol{0}$ for some $0 \leqslant i \leqslant b-1$. The symbols in $\boldsymbol{c}_i$ are separated by at least b positions from one another in $\boldsymbol{c}$ and, therefore, there are at least $b \cdot w_H(\boldsymbol{c}_i)$ symbols in $\pi_b(\boldsymbol{c})$ that are non-zero. That is,

$$w_b(\boldsymbol{c}) \geqslant b \cdot w_H(\boldsymbol{c}_i) \geqslant b \cdot \min_{0 \leqslant i \leqslant b-1} \{d_i\} = b \cdot d_H(\mathcal{C}).$$

The opposite inequality follows from the fact that there is a codeword $\boldsymbol{c} \in \mathcal{C}$ such that $w_H(\boldsymbol{c}) = d_H(\mathcal{C})$ and, according to Proposition 3,

$$w_b(\boldsymbol{c}) \leqslant b \cdot w_H(\boldsymbol{c}) = b \cdot d_H(\mathcal{C}).$$

We conclude that $d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C})$, as claimed. ■

According to the remark at the end of the previous subsection, Theorem 4 implies that the interleaved code $\mathcal{C}$ can correct any $\lfloor (d_b(\mathcal{C})-1)/2 \rfloor$ b-symbol errors. We now describe a decoding algorithm for $\mathcal{C}$ that achieves this decoding radius.

For a length-n vector $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ we define the length-bn vector

$$(\boldsymbol{c})_b = (c_0, \ldots, c_0, \ldots, c_{n-1}, \ldots, c_{n-1})$$

obtained by repeating b times each bit in $\boldsymbol{c}$. Now, for each component code $\mathcal{C}_i$, $0 \leqslant i \leqslant b-1$, of the interleaved code $\mathcal{C}$, we let $(\mathcal{C}_i)_b$ be the length-bn code

$$(\mathcal{C}_i)_b = \{(\boldsymbol{c})_b \ : \boldsymbol{c} \in \mathcal{C}_i\}.$$

If $\mathcal{C}_i$ is an (n, M_i, d_i) code, then $(\mathcal{C}_i)_b$ is a (bn, M_i, bd_i) code. We assume that the code $\mathcal{C}_i$ has a bounded distance decoder $\mathcal{D}_i$ that corrects errors of weight up to the decoding radius. Since the code $(\mathcal{C}_i)_b$ can be interpreted as a concatenation of an outer code $\mathcal{C}_i$ and an inner b-repetition code, we can also assume that $(\mathcal{C}_i)_b$ has a decoder $(\mathcal{D}_i)_b$ that can correct up to $\lfloor \frac{bd_i-1}{2} \rfloor$ errors; for more details on constructing the decoder $(\mathcal{D}_i)_b$ from decoder $\mathcal{D}_i$, we again refer the reader to [15, Ch. 12].

Assume that the stored codeword $\boldsymbol{c} \in \mathcal{C}$ is

$$\boldsymbol{c} = (c_{0,0}, \ldots, c_{b-1,0}, c_{0,1}, \ldots, c_{b-1,1}, \ldots, c_{0,n-1}, \ldots, c_{b-1,n-1}),$$

where $\boldsymbol{c}_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,n-1}) \in \mathcal{C}_i$ for all $0 \leqslant i \leqslant b-1$. Let $\pi_b(\mathcal{C})$ be the b-symbol read vector of $\boldsymbol{c}$, and let $\boldsymbol{y}$ be the length-bn received vector. We represent $\boldsymbol{y}$ as

$$\boldsymbol{y} = (\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{bn-1})$$

where $\boldsymbol{y}_j = (y_{j,0}, \ldots, y_{j,b-1})$, for $0 \leqslant j \leqslant bn-1$, and we make the assumption that

$$d_H(\boldsymbol{y}, \pi_b(\boldsymbol{c})) \leqslant \lfloor (d_b(\mathcal{C})-1)/2 \rfloor.$$

If we index the bit positions in $\boldsymbol{c}$ from 0 to $bn-1$, the bit $c_{i,j}$ in $\boldsymbol{c}$ lies in position $(jb+i)$, for $0 \leqslant i \leqslant b-1$, $0 \leqslant j \leqslant n-1$. Each bit $c_{i,j}$ is read b times, corresponding to the components $y_{jb+i-(b-1),b-1}, \ldots, y_{jb+i-1,1}$, and $y_{jb+i,0}$ that belong to the b-symbols $\boldsymbol{y}_{jb+i-(b-1)}, \ldots, \boldsymbol{y}_{jb+i-1}$, and $\boldsymbol{y}_{jb+i}$, respectively.

Next, we combine these estimates of $c_{i,j}$ into a binary vector $\overline{\boldsymbol{y}}_{i,j}$ for $0 \leqslant i \leqslant b-1$, $0 \leqslant j \leqslant n-1$, where

$$\overline{\boldsymbol{y}}_{i,j} = (y_{jb+i-(b-1),b-1}, \ldots, y_{jb+i-1,1}, y_{jb+i,0}).$$

Finally, the vector $\boldsymbol{y}$, treated as a binary vector, is partitioned into the following b vectors, each of length bn bits,

$$\overline{\boldsymbol{y}}_i = (\overline{\boldsymbol{y}}_{i,0}, \overline{\boldsymbol{y}}_{i,1}, \ldots, \overline{\boldsymbol{y}}_{i,n-1}),$$

for $0 \leqslant i \leqslant b-1$.

Lemma 5: For $0 \leqslant i \leqslant b-1$,

$$d_H\big(\overline{\boldsymbol{y}}_i, (\boldsymbol{c}_i)_b\big) \leqslant \lfloor (d_b(\mathcal{C})-1)/2 \rfloor$$

where d_H denotes the Hamming distance over the binary alphabet Σ.

Proof: First, note that, for $0 \leqslant i \leqslant b-1$, every $\overline{\boldsymbol{y}}_i$ is a noisy version of the codeword

$$(\boldsymbol{c}_i)_b = (c_{i,0}, \ldots, c_{i,0}, \ldots, c_{i,n-1}, \ldots, c_{i,n-1}) \in (\mathcal{C}_i)_b.$$

Furthermore, every b-symbol error in $\boldsymbol{y}$ can change at most one of the bits in every $\overline{\boldsymbol{y}}_i$ and thus the binary Hamming distance between $(\boldsymbol{c}_i)_b$ and $\overline{\boldsymbol{y}}_i$ is at most $\lfloor (d_b(\mathcal{C})-1)/2 \rfloor$, that is,

$$d_H\big(\overline{\boldsymbol{y}}_i, (\boldsymbol{c}_i)_b\big) \leqslant \lfloor (d_b(\mathcal{C})-1)/2 \rfloor.$$

■

Finally, from Theorem 4, we have

$$\lfloor (d_b(\mathcal{C})-1)/2 \rfloor = \lfloor (b \cdot d_H(\mathcal{C})-1)/2 \rfloor \leqslant \lfloor (b \cdot d_i - 1)/2 \rfloor,$$

for all i with $1 \leqslant i \leqslant b$, which implies that the decoder $(\mathcal{D}_i)_b$ can successfully decode the word $\overline{\boldsymbol{y}}_i$. Thus, for $0 \leqslant i \leqslant b-1$, the codeword $\boldsymbol{c}_i$ is successfully decoded and, therefore, so is the codeword $\boldsymbol{c}$.

C. Codes With Small Minimum Hamming Distance

In this section we study the minimum b-distance of two special classes of codes. The first class corresponds to the "complete" codebooks Σ^n and the second to the linear cyclic Hamming codes.

Given a length-n vector over Σ^b, the *majority decoder* outputs for each b-symbol the majority value among its b constituent bits, or ? if b is even and the number of zeros and ones is equal. The following lemma provides the minimum b-distance of the code Σ^n and proves that the majority decoder can be used to decode up to the b-distance decoding radius.

Lemma 6: Let $\mathcal{C} = \Sigma^n$. For all $b \geqslant 3$, the minimum b-distance satisfies $d_b(\mathcal{C}) = b$ and the majority decoder can correct $\lfloor \frac{b-1}{2} \rfloor$ b-symbol errors.

Proof: Assume $\boldsymbol{x}$ is a non-zero vector, and let $\widehat{\boldsymbol{x}}$ be as defined in Section II. Then $\widehat{\boldsymbol{x}} \neq \boldsymbol{0}$ and therefore $w_H(\widehat{\boldsymbol{x}}) \geqslant 1$. If $\boldsymbol{x} \neq \boldsymbol{1}$, then $w_H(\widehat{\boldsymbol{x}}') \geqslant 2$. Lemma 4 then implies that $w_b(\boldsymbol{x}) \geqslant 1 + (b-1) \cdot \frac{2}{2} = b$. If $\boldsymbol{x} = \boldsymbol{1}$, then $w_b(\boldsymbol{x}) = n$, and the inequality follows from the fact that $b < n$. We conclude that $d_b(\mathcal{C}) = b$ by noting that the codewords in $\pi_b(\mathcal{C})$ corresponding to the b-symbol read vector of a weight-1 word in $\mathcal{C}$ has b-weight b.

If there are $\lfloor \frac{b-1}{2} \rfloor$ symbol errors in the received version of $\pi_b(\boldsymbol{x})$, then for every $0 \leqslant i \leqslant n-1$, the bit x_i of the vector $\boldsymbol{x}$ is in error in at most $\lfloor \frac{b-1}{2} \rfloor$ of the b symbols that provide an estimate of that bit. Thus, the majority decoder can be used to recover every bit x_i and, therefore, the vector $\boldsymbol{x}$. ■

The following lemma considers the b-distance of linear cyclic Hamming codes, providing a generalization of a result in [1].

Lemma 7: If $\mathcal{C}$ is the linear cyclic Hamming code of length $n = 2^m - 1$ and $b + 2 \leqslant m$, then $d_b(\mathcal{C}) = 2b + 1$.

Proof: Let $\boldsymbol{x} \in \mathcal{C}$ be a non-zero codeword. We first show that $w_b(\widehat{\boldsymbol{x}}) \geqslant 2b+1$. Clearly, $w_H(\widehat{\boldsymbol{x}}) \geqslant w_H(\boldsymbol{x}) \geqslant 3$. Assume that $\widehat{\boldsymbol{x}} \neq \boldsymbol{1}$, so $w_H(\widehat{\boldsymbol{x}}')$ is a positive even integer. If $w_H(\widehat{\boldsymbol{x}}') \geqslant 4$, then according to Lemma 4, we have $w_b(\boldsymbol{x}) \geqslant 3 + (b-1) \cdot \frac{4}{2} = 2b+1$. If $w_H(\widehat{\boldsymbol{x}}') = 2$, then $\widehat{\boldsymbol{x}}$ contains a single sequence of consecutive ones, whose length ℓ must satisfy $\ell \geqslant m$. Otherwise, if $\ell < m$, the non-zero entries of the codeword $\boldsymbol{x}$ would be confined to at most $m-1$ locations. If $g(x)$ is a generator polynomial of degree m for the code, this would mean there exists a non-zero polynomial of degree at most $m-1$ which is a multiple of $g(x)$, which is not possible. Therefore, Lemma 4 implies that $w_b(\widehat{\boldsymbol{x}}) \geqslant m + (b-1) \cdot \frac{2}{2} = m + b - 1 \geqslant 2b + 1$. Finally, we note that if $\widehat{\boldsymbol{x}} = \mathbf{1}$, then $w_b(\widehat{\boldsymbol{x}}) = n \geqslant 2b + 1$ for $b \geqslant 3$ and $m \geqslant b + 2$.

To show that $d_b(\mathcal{C}) = 2b + 1$, we note that $\mathcal{C}$ contains a weight-3 codeword $\boldsymbol{x}$ with two consecutive ones. For such a codeword, the b-weight is exactly $2b + 1$. ■

VI. Conclusion

In this paper, we studied the symbol-pair read channel and developed an improved lower bound on the minimum pair distance of linear cyclic codes. We then developed an efficient, bounded distance decoding algorithm that corrects a number of symbol-pair errors up to the decoding radius corresponding to this bound. Finally, we studied properties of the b-symbol read channel model, for $b \geqslant 3$, and the associated b-distance. We then considered several constructions of codes for the b-symbol read channel, computed their minimum b-distance, and in several cases proposed effective bounded distance decoding algorithms.

Clearly, the results in this paper do not solve all problems related to b-symbol read channels. In particular, it is not clear whether the lower bound on the minimum pair-distance of a linear cyclic code in Theorem 1 can be improved, and possible improvement in the b-distance case remains open, as well. There is also a need for additional code constructions beyond interleaving and those based on codes with small minimum Hamming distance. Simplified decoding algorithms for specific codes in the b-symbol error correction setting are also of interest. Finally, several other topics relating to b-symbol error-correcting codes have not been explored, such as the existence of perfect codes, tight upper bounds on code cardinalities, and optimal code constructions with some prescribed minimum b-distance.

Acknowledgement

The authors thank Yuval Cassuto for helpful discussions on the symbol-pair read channel and Ryan Gabrys for bringing reference [17] to their attention. They also thank two anonymous reviewers and the Associate Editor Prof. Yongyi Mao for their valuable comments and suggestions.

References

[1] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 8011–8020, Dec. 2011.

[2] Y. Cassuto and S. Litsyn, "Symbol-pair codes: Algebraic constructions and asymptotic bounds," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul./Aug. 2011, pp. 2348–2352.

[3] Y. M. Chee, H. M. Kiah, and C. Wang, "Maximum distance separable symbol-pair codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2886–2890.

[4] Y. M. Chee, L. Ji, H. M. Kiah, C. Wang, and J. Yin, "Maximum distance separable codes for symbol-pair read channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7259–7267, Nov. 2013.

[5] M. Hirotomo, M. Takita, and M. Morii, "Syndrome decoding of symbol-pair codes," in *Proc. IEEE Inf. Theory Workshop*, Hobart, TAS, Australia, Nov. 2014, pp. 162–166.

[6] E. Konstantinova, "Reconstruction of signed permutations from their distorted patterns," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, SA, Australia, Sep. 2005, pp. 474–477.

[7] E. Konstantinova, V. I. Levenshtein, and J. Siemons. (Feb. 2007). "Reconstruction of permutations distorted by single transposition errors." [Online]. Available: http://arxiv.org/abs/math/0702191

[8] V. I. Levenshtein, "Reconstruction of objects from a minimum number of distorted patterns," *Doklady Math.*, vol. 55, no. 3, pp. 417–420, 1997.

[9] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 2–22, Jan. 2001.

[10] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences" *J. Combinat. Theory A*, vol. 93, no. 2, pp. 310–332, 2001.

[11] V. I. Levenshtein and J. Siemons, "Error graphs and the reconstruction of elements in groups," *J. Combinat. Theory A*, vol. 116, no. 4, pp. 795–815, 2009.

[12] S. Motahari, G. Bresler, and D. Tse. (Feb. 2013). "Information theory of DNA shotgun sequencing." [Online]. Available: http://arxiv.org/abs/1203.6233

[13] S. Motahari, G. Bresler, and D. Tse, "Information theory for DNA sequencing: Part I: A basic model," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2741–2745.

[14] E. Orsini and M. Sala, "Correcting errors and erasures via the syndrome variety," *J. Pure Appl. Algebra*, vol. 200, nos. 1–2, pp. 191–226, 2005.

[15] R. M. Roth, *Introduction to Coding Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[16] H. Shahri and K. K. Tzeng, "On error-and-erasure decoding of cyclic codes," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 489–496, Mar. 1992.

[17] S. E. Tavares and S. G. S. Shiva, "Detecting and correcting multiple bursts for binary cyclic codes (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 16, no. 5, pp. 643–644, Sep. 1970.

[18] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 106–110.

[19] E. Yaakobi, J. Bruck, and P. H. Siegel, "Decoding of cyclic codes over symbol-pair read channels," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2891–2895.

[20] E. Yaakobi, M. Schwartz, M. Langberg, and J. Bruck, "Sequence reconstruction for Grassmann graphs and permutations," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 874–878.

Eitan Yaakobi (S'07–M'12) is an Assistant Professor at the Computer Science Department at the Technion — Israel Institute of Technology. He received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. Between 2011-2013, he was a postdoctoral researcher in the department of Electrical Engineering at the California Institute of Technology. His research interests include information and coding theory with applications to non-volatile memories, associative memories, data storage and retrieval, and voting theory. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010-2011.

Jehoshua Bruck (S'86–M'89–SM'93–F'01) is the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology (Caltech). His current research interests include information theory and systems and the theory of computation in nature.

Dr. Bruck received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University, in 1989.

His industrial and entrepreneurial experiences include working with IBM Research where he participated in the design and implementation of the first IBM parallel computer; cofounding and serving as Chairman of Rainfinity (acquired in 2005 by EMC), a spin-off company from Caltech that created the first virtualization solution for Network Attached Storage; as well as cofounding and serving as Chairman of XtremIO (acquired in 2012 by EMC), a start-up company that created the first scalable all-flash enterprise storage system.

Dr. Bruck is a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award and the IBM Outstanding Technical Achievement Award.

Paul H. Siegel (M'82–SM'90–F'97) received the S.B. and Ph.D. degrees in mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, in 1975 and 1979, respectively. He held a Chaim Weizmann Postdoctoral Fellowship at the Courant Institute, New York University. He was with the IBM Research Division in San Jose, CA, from 1980 to 1995. He joined the faculty at the University of California, San Diego in July 1995, where he is currently Professor of Electrical and Computer Engineering in the Jacobs School of Engineering. He is affiliated with the Center for Magnetic Recording Research where he holds an endowed chair and served as Director from 2000 to 2011. His primary research interests lie in the areas of information theory and communications, particularly coding and modulation techniques, with applications to digital data storage and transmission. Prof. Siegel was a member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996 and again from 2009 to 2014. He served as Co-Guest Editor of the May 1991 Special Issue on "Coding for Storage Devices" of the IEEE TRANSACTIONS ON INFORMATION THEORY. He served the same TRANSACTIONS as Associate Editor for Coding Techniques from 1992 to 1995, and as Editor-in-Chief from July 2001 to July 2004. He was also Co-Guest Editor of the May/September 2001 two-part issue on "The Turbo Principle: From Theory to Practice" and the February 2016 issue on "Recent Advances in Capacity Approaching Codes" of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. Prof. Siegel was co-recipient, with R. Karabed, of the 1992 IEEE Information Theory Society Paper Award and shared the 1993 IEEE Communications Society Leonard G. Abraham Prize Paper Award with B. H. Marcus and J.K. Wolf. With J. B. Soriaga and H. D. Pfister, he received the 2007 Best Paper Award in Signal Processing and Coding for Data Storage from the Data Storage Technical Committee of the IEEE Communications Society. He was the 2015 Padovani Lecturer of the IEEE Information Theory Society. Prof. Siegel is an IEEE Fellow and a member of the National Academy of Engineering.