

# Cyclic and convolutional codes with locality

Zitan Chen

Alexander Barg

**Abstract**—Locally recoverable (LRC) codes and their variants have been extensively studied in recent years. In this paper we focus on cyclic constructions of LRC codes and derive conditions on the zeros of the code that support the property of hierarchical locality. As a result, we obtain a general family of hierarchical LRC codes for a new range of code parameters. We also observe that our approach enables one to represent an LRC code in quasicyclic form, and use this representation to construct tail-biting convolutional LRC codes with locality. Among other results, we extend the general approach to cyclic codes with locality to multidimensional cyclic codes, yielding new families of LRC codes with availability, and construct a family of  $q$ -ary cyclic hierarchical LRC codes of unbounded length.

**Index Terms**—Hierarchical locality, tail-biting codes, sliding window repair, bi-cyclic codes.

## I. INTRODUCTION

### A. The locality problem: main definitions

Locally recoverable (LRC) codes form a family of erasure codes, motivated by applications in distributed storage, that support repair of a failed storage node by contacting a small number of other nodes in the cluster. LRC codes can be constructed in a number of ways. A connection between LRC codes and the well-studied family of Reed-Solomon (RS) codes was put forward in [33], where codes with large distance were constructed as certain subcodes of RS codes. The results of [33] paved the way for using powerful algebraic techniques of coding theory for constructing other families of LRC codes including algebraic geometric codes [3], [23], [24]. In [35] it was observed that a particular class of the codes in [33] can be represented in cyclic form, and the distance and locality properties of cyclic LRC codes were described in terms of the zeros of the code. This established a framework for cyclic LRC codes that was advanced in a number of ways in several recent works [4], [7], [15], [27].

In this paper we focus on several aspects of cyclic LRC codes that have not been previously addressed in the literature. The first of these is codes with hierarchical locality (H-LRC codes), defined in [30], which assume that the code can correct one or more erasures by using a subset whose size depends on the number of erased locations, increasing progressively with their count. In addition to defining the problem and deriving a bound on the parameters of H-LRC codes, the authors of [30] extended the construction of [33] to the hierarchical case. Their

construction was further generalized to algebraic geometric codes in [10].

Another general problem related to cyclic codes that we study addresses construction of LRC convolutional codes. This class of erasure codes was considered in several previous works [8], [17], [40] before being thoroughly analyzed in a recent paper [28]. Here we exploit a classic link between quasicyclic codes and convolutional codes [32] to construct LRC convolutional codes with a particular type of locality whose parameters are controlled by the set of zeros of the underlying (quasi)cyclic code.

We also address the problem of maximum length of optimal LRC codes [14], wherein the main question is constructing such codes of length larger than the size of the code alphabet. This problem has been the subject of a number of recent papers including [2], [4], [24], and we study this question in the hierarchical case. Our techniques here combine the approach to cyclic H-LRC codes developed in this paper with the ideas of [27] aimed at constructing long LRC codes.

We describe the known and the new results in more details after giving the basic definitions. In coding-theoretic terms, the central problem addressed by LRC codes is correcting one or several erasures in the codeword based on the contents of a small number of other coordinates. This problem was first isolated in [12], and it has been actively studied in the last decade.

**Definition 1 (LRC CODES).** A linear code  $\mathcal{C} \subset \mathbb{F}_q^n$  is locally recoverable with locality  $r$  if for every  $i \in \{1, 2, \dots, n\}$  there exists an  $r$ -element subset  $I_i \subset \{1, 2, \dots, n\} \setminus \{i\}$  and a linear function  $\phi_i : \mathbb{F}_q^r \rightarrow \mathbb{F}_q$  such that for every codeword  $c \in \mathcal{C}$  we have  $c_i = \phi_i(c_{j_1}, \dots, c_{j_r})$ , where  $j_1 < j_2 < \dots < j_r$  are the elements of  $I_i$ .

The coordinates in  $I_i$  are called the *recovering set* of  $i$ , and the set  $\{i\} \cup I_i$  is called a *repair group*. Below we refer to a linear LRC code of length  $n$ , dimension  $k$ , and locality  $r$  as an  $(n, k, r)$  LRC code. Since the code is occasionally used to correct a large number of erasures (such as in the event of massive system failure), another parameter of interest is the maximum number of erasures that it can tolerate. This is controlled by the minimum distance  $d(\mathcal{C})$  of the code, for which there are several bounds known in the literature. We will be interested in the generalized Singleton bound of [12] which states that for any LRC code  $\mathcal{C}$ ,

$$d(\mathcal{C}) \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2. \quad (1)$$

While in most situations repairing a single failed node restores the system to the functional state, occasionally there may be a need to recover the data from several concurrent node failures. The following extension of the previous definition is

An extended abstract of this work appears in Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT 2020).

Zitan Chen is with Dept. of ECE and ISR, University of Maryland, College Park, MD 20742. Email: chenztan@umd.edu.

Alexander Barg is with Dept. of ECE and ISR, University of Maryland, College Park, MD 20742 and also with IITP, Russian Academy of Sciences, 127051 Moscow, Russia. Email: abarg@umd.edu.

This research was supported by NSF grants CCF1618603 and CCF1814487.

due to [20].

**Definition 2** ( $(r, \delta)$  LOCALITY). *For any  $\delta \geq 2$  we say that a linear code  $\mathcal{C}$  has  $(r, \delta)$  locality if every coordinate  $i \in \{1, \dots, n\}$  is contained in a subset  $J_i \subset \{1, \dots, n\}$  of size at most  $r + \delta - 1$  such that the restriction  $\mathcal{C}_{J_i}$  to the coordinates in  $J_i$  forms a code of distance at least  $\delta$ .*

Note that in the case of  $\delta = 2$  the codes defined here are exactly the codes of Def. 1 above.

An intermediate situation arises when the code is designed to correct a single erasure by contacting a small number  $r_1$  of helper nodes, while at the same time supporting local recovery of multiple erasures. Extending this idea to multiple levels of local protection, the authors of [31] introduced the concept of hierarchical LRC (H-LRC) codes, which are defined as follows.

**Definition 3** (H-LRC CODES). *Let  $h \geq 1$ ,  $0 < r_1 < r_2 < \dots < r_h < k$ , and  $1 < \delta_1 \leq \dots \leq \delta_h \leq d$  be integers. A linear code  $\mathcal{C} \subset \mathbb{F}_q^n$  is said to have  $h$ -level hierarchical locality  $(r_1, \delta_1), \dots, (r_h, \delta_h)$  if for every  $1 \leq i \leq h$  and every coordinate of the code  $\mathcal{C}$  there is a punctured code  $\mathcal{C}^{(i)}$  such that the coordinate is in the support of  $\mathcal{C}^{(i)}$  and*

- (a)  $\dim(\mathcal{C}^{(i)}) \leq r_i$ ,
- (b)  $d(\mathcal{C}^{(i)}) \geq \delta_i$ ,
- (c) the  $i$ -th local code  $\mathcal{C}^{(i)}$  has  $(i - 1)$ -level hierarchical locality  $(r_1, \delta_1), \dots, (r_{i-1}, \delta_{i-1})$ .

The authors of [31] proved the following extension of the bound (1): The minimum distance of an  $h$ -level H-LRC code with locality satisfies the inequality

$$d \leq n - k + \delta_h - \sum_{i=1}^h \left\lceil \frac{k}{r_i} \right\rceil (\delta_i - \delta_{i-1}), \quad (2)$$

where  $\delta_0 = 1$ . In particular, for  $h = 1$  this gives a version of the bound (1) for the distance of an  $(n, k)$  code with  $(r, \delta)$  locality:

$$d \leq n - k + \delta - \left\lceil \frac{k}{r} \right\rceil (\delta - 1). \quad (3)$$

We call an H-LRC code *optimal* if its distance attains the bound (2). Note that it is possible that the code is optimal while its local codes  $\mathcal{C}^{(i)}$  for some or even all  $i \leq h - 1$  are not. We say that an H-LRC code  $\mathcal{C}$  is *strongly optimal* if in every level  $i$ ,  $1 \leq i \leq h$ , the  $i$ -th local codes are optimal H-LRC codes with  $i - 1$  levels. For  $h = 1$  the distance of the optimal code with  $(r, \delta)$  locality attains the bound (3) with equality.

## B. Earlier results and our contributions

As noted above, the starting point of our constructions is a cyclic version of the RS-like codes with locality designed in [33]. RS codes over  $\mathbb{F}_q$  can be alternatively described in terms of polynomial evaluation and (in the case that the code length  $n$  divides  $q - 1$ ) as cyclic codes of the BCH type. While [33] adopted the former approach, the authors of [35] studied the cyclic case, finding a condition on the zeros of the code that support the locality property.

The approach of [35] was later advanced in a number of ways. In particular, the authors of [4] extended the main

construction of [35] to codes with  $(r, \delta)$  locality, designing a cyclic representation of the polynomial evaluation codes from [33] for the general case of  $\delta \geq 2$ .

Our main results relate to constructions of cyclic LRC codes with hierarchy. Several recent studies presented families of codes with multiple levels of erasure correction [10], [13], [39], not necessarily within the framework of the above definition. As far as H-LRC codes are concerned, the only general family of optimal H-LRC codes that we are aware of was presented in [4], [31]. This construction essentially followed the approach of [33], relying on multivariate polynomials that are constant on the blocks that form the support of the code  $\mathcal{C}^{(i)}$  in Def. 3. The codes of [4], [31] form a family of strongly optimal H-LRC codes which can be constructed for any code length  $n \leq q$ , dimension  $k$ , and any values of  $r_i$ ,  $i = 1, \dots, h$  as long as  $r_i | r_{i+1}$ ,  $i = 1, \dots, h - 1$  and  $r_h | k$ . The divisibility constraint is essential for the constructions discussed, and it limits the possible choices of the code parameters.

In this paper we derive conditions on the zeros of a cyclic code that support several levels of hierarchy. As a result, we construct families of cyclic H-LRC codes that do not rely on the divisibility assumptions, which yields a new range of code parameters. We also derive conditions that are sufficient for our codes to be strongly optimal and give examples of such codes. These results are presented in Sec. I-A.

These results also enable us to connect the construction of LRC cyclic codes and convolutional codes with locality. The recent work [28] focused on the so-called sliding window repair property of convolutional codes. The authors of [28] further observed that certain families of convolutional codes, notably the so-called codes with the maximum distance profile [40], [37], suggest an approach to constructing codes with locality. They also presented a family of LRC convolutional codes with sliding window repair for the case of *column locality*, and they suggested that there may be a connection between H-LRC codes and LRC convolutional codes. We show that this connection indeed leads to fruitful results, designing LRC convolutional codes for the case of *row locality* defined herein. The lower bounds on the column distance of the codes constructed here and in [28] are the same; however the alphabet size of our codes is much smaller than in [28]. We also derive an upper bound on the column distance of LRC convolutional codes (with either type of locality); however, our construction falls short of attaining it. The method that we use relies on the characterization of zeros of cyclic block H-LRC codes described above. We observe that several levels of hierarchy enable one to put our cyclic LRC codes in quasicyclic form, and then use a classic connection between quasicyclic codes and convolutional codes [32] to construct convolutional codes with locality; see Sec. II.

We also examine two other problems for LRC codes that benefit from the cyclic code construction. The first of them is the problem of maximum length of optimal LRC codes put forward in [14]. Answering the challenge of constructing optimal LRC codes of length larger than  $q$ , the authors of [4], [6], [14], [27] constructed several families of optimal cyclic codes of large, and in some cases even unbounded length, and [6] extended these results to the case of several erasures.

Here we follow the lead of [27] and construct an infinite family of H-LRC codes over a given finite field and establish conditions for their optimality in terms of the bound (2). Finally, we consider the problem of *availability* which calls for constructing LRC codes with several disjoint recovering sets for each code coordinate (see the definition in Sec. IV). We note that multidimensional cyclic codes naturally yield several recovering sets for the coordinates. We use a description of bi-cyclic codes in terms of their zeros together with a special version of code concatenation [30] to construct codes with availability and rate higher than the rate of product codes. We note that the known bounds on the code parameters for multiple recovering sets [21], [29], [34], [38] do not support a conclusive picture, and we are not aware of general families of codes with availability whose distance attains one of the known upper limits. Our construction in this paper also falls in the same category.

Although we do not pursue this direction here, let us note that the methods of this paper enable one to construct codes with both hierarchical locality and availability. We remark that constructions of LRC codes that have both properties were presented in [10], where the main tools were fiber products and covering maps of algebraic curves.

### C. Optimal cyclic LRC codes

Let  $\mathcal{C}$  be a cyclic code of length  $n$  with generator polynomial  $g(x)$  and check polynomial  $h(x) = \frac{x^n - 1}{g(x)}$ . The dual code  $\mathcal{C}'$  has generator polynomial  $g_{\mathcal{C}'}(x) = x^{\deg(h(x))} h(x^{-1})$ , and the code  $\tilde{\mathcal{C}}'$  generated by  $h(x)$  is obtained from  $\mathcal{C}'$  by inverting the order of the coordinates. A codeword  $a(x) \in \mathcal{C}'$  of weight  $r+1$  defines a repair group of the code  $\mathcal{C}$ , and so does the reversed codeword  $\tilde{a}(x) \in \tilde{\mathcal{C}}'$ . For this reason below in this section we argue about the code  $\tilde{\mathcal{C}}'$  rather than  $\mathcal{C}'$ , which makes the writing more compact without affecting the conclusions.

Let us recall a connection between cyclic codes and LRC codes of [35], which we present in the form close to the earlier works [4], [27]. The following lemma underlies constructions of cyclic LRC codes in this paper and elsewhere, and it represents a mild extension of Lemma 3.3 in [35]. In the statement as well as elsewhere in the paper we do not distinguish between zeros of the code and their exponents in terms of some fixed primitive  $n$ th root of unity in  $\mathbb{F}_q$ .

**Lemma 1.** *Let  $\mathcal{C}$  be a cyclic code over  $\mathbb{F}_q$  of length  $n|(q-1)$  and let  $\alpha$  be a primitive  $n$ th root of unity in  $\mathbb{F}_q$ . Suppose that  $n = \nu m$  for some integers  $\nu, m$ . Then the code  $\tilde{\mathcal{C}}'$  contains a vector*

$$b(x) = \sum_{i=0}^{m-1} x^{i\nu} \alpha^{(m-1-i)\nu u}, \quad u \in \{0, \dots, m-1\} \quad (4)$$

if and only if the set  $\mathcal{L} = \{u + im, i = 0, 1, \dots, \nu-1\}$  is among the zeros of  $\mathcal{C}$ .

*Proof:* Notice that the polynomial  $b(x)$  can be equivalently written as  $b(x) = \frac{x^n - 1}{x^\nu - \alpha^{\nu u}}$ , where

$$x^\nu - \alpha^{\nu u} = \prod_{i=0}^{\nu-1} (x - \alpha^{u+im}) \quad (5)$$

is the annihilator polynomial of the set  $\mathcal{L}$ . Thus,  $b(\alpha^t) \neq 0$  for all  $t \in \mathcal{L}$ . If  $b(x) \in \tilde{\mathcal{C}}'$ , this implies that  $\mathcal{L}$  is a subset of the set of zeros of  $\mathcal{C}$ .

Conversely, let  $g(x) = (x^\nu - \alpha^{\nu u})p(x)$  be the generator polynomial of  $\mathcal{C}$ . Then

$$\left(\frac{x^n - 1}{g(x)}\right)p(x) = \frac{x^n - 1}{x^\nu - \alpha^{\nu u}} = b(x) \in \tilde{\mathcal{C}}'.$$

This lemma immediately yields the cyclic codes from [35] (the cyclic case of the codes from [33]).

**Theorem 2** ([35]). *Let  $(r+1)|n, r|k, n|(q-1)$ . Let  $\alpha \in \mathbb{F}_q$  be a primitive  $n$ -th root of unity, and let  $\mathcal{C}$  be an  $(n, k)$  cyclic code with zeros  $\alpha^i, i \in \mathcal{Z} := \mathcal{L} \cup \mathcal{D}$ , where*

$$\begin{aligned} \mathcal{L} &= \{1 + l(r+1), l = 0, \dots, \frac{n}{r+1} - 1\} \\ \mathcal{D} &= \{1, 2, \dots, n - \frac{k}{r}(r+1) + 1\}. \end{aligned} \quad (6)$$

*Then  $\mathcal{C}$  is an  $(n, k, r)$  optimal LRC code.*

*Proof:* Since  $\mathcal{L} \subset \mathcal{Z}$ , Lemma 1 implies that

$$b(x) = \sum_{i=0}^r \alpha^{i \frac{n}{r+1}} x^{(r-i) \frac{n}{r+1}}$$

is a codeword in  $\tilde{\mathcal{C}}'$ . This codeword is of weight  $r+1$ , and its cyclic shifts give  $\frac{n}{r+1}$  disjoint repair groups, supporting the locality claim. At the same time, the BCH bound implies that  $d(\mathcal{C}) \geq n - \frac{k}{r}(r+1) + 2$ , so the code is optimal by (1) once one observes  $|\mathcal{Z}| = n - k$  and  $\dim(\mathcal{C}) = n - |\mathcal{Z}| = k$ . ■

This construction extends without difficulty to codes with  $(r, \delta)$  locality for any  $\delta \geq 2$ . A family of optimal codes in the sense of the bound (3) was constructed in [33], Construction 8 (see also [4]). The codes in this family are constructed as certain subcodes of Reed-Solomon codes that rely on piecewise-constant polynomials over  $\mathbb{F}_q$ . In the particular case that the code length  $n$  divides  $q-1$  it is possible to represent these codes in cyclic form. For this, we assume that  $r|k$ , take  $m = r + \delta - 1$ , and take the zeros of the code to be

$$\begin{aligned} \mathcal{L} &= \{i + lm \mid l = 0, \dots, \nu-1, i = 1, \dots, \delta-1\} \\ \mathcal{D} &= \{1, 2, \dots, n - k - ((k/r) - 1)(\delta-1)\}. \end{aligned} \quad (7)$$

As will be apparent from the proof of Lemma 3, the condition about zeros given by the set  $\mathcal{L}$  translates into conditions on the dual code that support the locality claim. The distance of the code  $\mathcal{C}$  clearly meets the bound (3) with equality.

### D. Cyclic codes with locality

In the next lemma we present a slightly more general view of the method in Theorem 2 that will be instrumental in the code constructions below in this work. The main element of the construction is code  $\mathcal{C}^{(0)}$  defined in (8), which isolates a repair group in the code  $\mathcal{C}$  and supports local correction of several erasures.

**Lemma 3.** *Let  $n|(q-1), n = \nu m$ . Let  $\alpha$  be a primitive  $n$ -th root of unity in  $\mathbb{F}_q$ , and fix  $\delta \in \{2, \dots, m\}$ . Let  $\mathcal{Z}$  be a subset of size  $Z$  such that*

$$\{1, \dots, \delta-1\} \subset \mathcal{Z} \subset \{0, \dots, m-1\}$$

and let  $\mathcal{L} = \bigcup_{s=0}^{\nu-1} (\mathcal{Z} + sm)$ . Consider a cyclic code  $\mathcal{C} = \langle g(x) \rangle$  of length  $n$ , where

$$g(x) = p(x) \prod_{t \in \mathcal{L}} (x - \alpha^t),$$

and  $p(x) \in \mathbb{F}_q[x]$  is some polynomial. Let

$$\mathcal{C}^{(0)} = \{(c_0, c_\nu, \dots, c_{(m-1)\nu}) \mid (c_0, \dots, c_{n-1}) \in \mathcal{C}\}. \quad (8)$$

Then  $\dim(\mathcal{C}^{(0)}) \leq m - Z$ ,  $d(\mathcal{C}^{(0)}) \geq \delta$ , and thus, the code  $\mathcal{C}$  is an LRC code with  $(m - Z, \delta)$  locality.

Further, if for every  $u \in \{0, \dots, m-1\} \setminus \mathcal{Z}$  there exists  $s \in \{0, 1, \dots, \nu-1\}$  such that  $g(\alpha^{u+sm}) \neq 0$ , then  $\dim(\mathcal{C}^{(0)}) = m - Z$ .

*Proof:* We proceed similarly to Theorem 2. Let

$$l(x) = \prod_{t \in \mathcal{L}} (x - \alpha^t) = \prod_{s=0}^{\nu-1} \prod_{i \in \mathcal{Z}} (x - \alpha^{i+sm}) = \prod_{i \in \mathcal{Z}} l_i(x)$$

where  $l_i(x) := x^\nu - \alpha^{\nu i}$ . Let  $h(x) = \frac{x^n - 1}{g(x)}$  and consider a subset of  $Z$  codewords of  $\mathcal{C}'$  given by

$$b_i(x) := h(x)p(x) \prod_{j \in \mathcal{Z} \setminus \{i\}} l_j(x), \quad i \in \mathcal{Z}.$$

A codeword  $b_i$  has the form

$$b_i(x) = \frac{x^n - 1}{l_i(x)} = \sum_{j=0}^{m-1} \alpha^{(m-1-j)\nu} x^{j\nu},$$

Hamming weight  $m$ , and contains  $\nu - 1$  zero coordinates after every nonzero entry.

To prove the statement about locality, let us form a  $Z \times m$  matrix  $H$  obtained by inverting the order of coordinates in the codewords  $b_i, i \in \mathcal{Z}$ , writing the resulting vectors as rows, and discarding all the zero columns. By construction, every row of  $H$  is a parity-check equation of the code  $\mathcal{C}^{(0)}$ . Any submatrix of  $\delta - 1$  columns of  $H$  has rank  $\delta - 1$  (its first  $\delta - 1$  rows form a Vandermonde determinant), and thus,  $d(\mathcal{C}^{(0)}) \geq \delta$ . Since the rows of  $H$  give  $Z$  independent parity-check equations for the code  $\mathcal{C}^{(0)}$ , we also have  $\dim(\mathcal{C}^{(0)}) \leq m - Z$ . This argument exhibits a local code in the coordinates that are integer multiples of  $\nu$ , and by cyclic shifts we can partition the set  $\{0, 1, \dots, n-1\}$  into supports of disjoint local codes of length  $m$  and distance at least  $\delta$ . Furthermore, we note that the punctured code  $\mathcal{C}^{(0)}$  is itself a cyclic code of length  $m|n|$ . Let  $g_0(x)$  be its generator polynomial. Since each row of  $H$  is a parity-check equation for the code  $\mathcal{C}^{(0)}$ , we have  $g_0(\alpha^{i\nu}) = 0$  for every  $i \in \mathcal{Z}$  and thus  $\deg(g_0(x)) \geq Z$ . Together these arguments prove the claim about  $(m - Z, \delta)$  locality of the code  $\mathcal{C}$ .

Next we show that if  $\deg(g_0(x)) > Z$ , then necessarily there exists  $u \in \{0, \dots, m-1\} \setminus \mathcal{Z}$  such that  $g(\alpha^{u+sm}) = 0$  for all  $s = 0, \dots, \nu-1$ . Suppose that there exists  $u \in \{0, \dots, m-1\} \setminus \mathcal{Z}$  such that  $g_0(\alpha^{u\nu}) = 0$ . By Eq. (5) in the proof Lemma 1 there exists a codeword  $b_u \in \mathcal{C}'$  given by

$$b_u(x) = \sum_{j=0}^{m-1} \alpha^{ju\nu} x^{(m-1-j)\nu} = \frac{x^n - 1}{x^\nu - \alpha^{\nu u}}.$$

On the other hand,  $\mathcal{C}' = \langle h(x) \rangle$ , so  $h(x) | b_u(x)$  and therefore  $(x^\nu - \alpha^{\nu u}) | g(x)$ . Noticing that  $x^\nu - \alpha^{\nu u} = \prod_{s=0}^{\nu-1} (x - \alpha^{u+sm})$  (cf. (5)), we conclude that  $g(x)$  is divisible by  $x - \alpha^{u+sm}$  for all  $s = 0, \dots, \nu-1$ . Hence, if for every  $u \in \{0, \dots, m-1\} \setminus \mathcal{Z}$  there exists  $0 \leq s \leq \nu-1$  such that  $(x - \alpha^{u+sm}) \nmid g(x)$ , i.e.,  $g(\alpha^{u+sm}) \neq 0$ , then  $\deg(g_0(x)) = Z$ , and thus  $\dim(\mathcal{C}^{(0)}) = m - Z$ . ■

## II. CODES WITH HIERARCHICAL LOCALITY

### A. Optimal cyclic codes with hierarchy

In this section we construct a family of H-LRC cyclic codes with  $h \geq 1$  levels of hierarchy and derive sufficient conditions for their optimality. Suppose that  $h$  is fixed and we are given the local dimension  $r_1$  (the dimension of the first, innermost local code), and the designed local distances  $1 = \delta_0 \leq \delta_1 \leq \dots \leq \delta_h \leq \delta_{h+1}$ , where  $\delta_{h+1}$  is the designed distance of the overall code  $\mathcal{C}$ .

We will assume that the first local code is MDS and thus its length is  $n_1 = r_1 + \delta_1 - 1$ . For  $1 \leq i \leq h$ , let  $n_{i+1} = \nu_i n_i$  be the length of the code in the  $(i+1)$ st level of hierarchy, where  $\nu_i > 1$  is an integer. Let  $\mathbb{F}_q$  be a finite field and suppose that  $n_{h+1} | (q-1)$ .

We construct a cyclic H-LRC code  $\mathcal{C}$  over  $\mathbb{F}_q$  of length  $n = n_{h+1}$  and designed (local) distances  $\delta_1, \dots, \delta_{h+1}$  as follows. Let  $\alpha \in \mathbb{F}_q$  be a primitive  $n$ -th root of unity. The code  $\mathcal{C}$  will be given by its defining set of zeros  $\mathcal{Z}$  which we specify via a recursive procedure. Consider the set of exponents  $\mathcal{D}_1 = \{1, \dots, \delta_1 - 1\}$  of the primitive element  $\alpha$ . Further, let  $\mathcal{L}_1 = \emptyset$  and

$$\mathcal{Z}_1 = \mathcal{L}_1 \cup \mathcal{D}_1. \quad (9)$$

Having (6) in mind, for  $1 \leq i \leq h$  let

$$\mathcal{L}_{i+1} = \bigcup_{s=0}^{\nu_i-1} (\mathcal{Z}_i + sn_i), \quad \mathcal{D}_{i+1} = \{1, \dots, \delta_{i+1} - 1\}, \quad (10)$$

$$\mathcal{Z}_{i+1} = \mathcal{L}_{i+1} \cup \mathcal{D}_{i+1}.$$

Finally, put  $\mathcal{Z} = \mathcal{Z}_{h+1}$ .

The generator polynomial of the cyclic code  $\mathcal{C}$  of length  $n$  is given by

$$g(x) = \prod_{t \in \mathcal{Z}} (x - \alpha^t). \quad (11)$$

The parameters of the code  $\mathcal{C}$  are estimated as follows.

**Proposition 4.** (i) The dimension of the code  $\mathcal{C}$  is  $n - |\mathcal{Z}|$  and the distance  $d \geq \delta_{h+1}$ . (ii) The code  $\mathcal{C}$  is an  $h$ -level H-LRC code with locality  $(n_i - |\mathcal{Z}_i|, \delta_i), i = 1, \dots, h$ .

*Proof:* (i) The value of the dimension is clear from the construction, and the estimate for the distance comes from the BCH bound.

(ii) The statement about the locality parameters follows by Lemma 3 once we observe that  $g(x)$  is divisible by  $\prod_{t \in \bigcup_{s=0}^{n/n_i-1} (\mathcal{Z}_i + sn_i)} (x - \alpha^t)$  for every  $i = 1, \dots, h$ . ■

Next we examine the conditions that suffice for the distance of  $\mathcal{C}$  to meet the bound (2) with equality. We build up the optimality of our code in an incremental manner in the sense that we first ensure that the first local codes are optimal (i.e., MDS codes), and relying on these optimal local codes we



make sure that the second local codes are optimal (i.e., optimal LRC codes), and so forth until we reach the outermost code.

Let  $r_1 < r_2 < \dots < r_h < r_{h+1} = \dim(\mathcal{C})$  be the chosen values of the dimensions of the local codes. As before, we set  $n_1 = r_1 + \delta_1 - \delta_0$  and let  $n_{i+1} = \nu_i n_i$  for  $1 \leq i \leq h$  where the integer number  $\nu_i$  satisfies  $\nu_i \geq \lceil r_{i+1}/r_i \rceil$ . Note that this assumption does not entail a loss of generality since, assuming the  $i$ th and  $(i+1)$ st local codes are optimal, by (III) we have  $|\mathcal{L}_{i+1}| = (n_i - r_i)\nu_i$ , and  $r_{i+1} = n_{i+1} - |\mathcal{Z}_{i+1}| \leq n_{i+1} - |\mathcal{L}_{i+1}| = r_i \nu_i$ .

To show optimality, we connect the target values of the local distances  $\delta_2, \delta_3, \dots, \delta_{h+1}$  with the dimension values through several auxiliary parameters. For  $1 \leq i \leq h$ , let us write  $r_{i+1} = a_i r_i - b_i$  where  $0 \leq b_i < r_i$ . Further, let  $b_i^{(i)} = b_i$  and for  $j = i, i-1, \dots, 2$ , let

$$b_j^{(i)} = u_{j-1}^{(i)} r_{j-1} + b_{j-1}^{(i)}, \quad 0 \leq b_{j-1}^{(i)} < r_{j-1}.$$

Put  $b_0^{(0)} = 0$  and for  $1 \leq i \leq h$  let

$$b_0^{(i)} = (b_1^{(i)} + b_0^{(i-1)}) \bmod r_1, \quad u_0^{(i)} = \left\lfloor \frac{b_1^{(i)} + b_0^{(i-1)}}{r_1} \right\rfloor. \quad (12)$$

Finally, for  $1 \leq i \leq h$ , let

$$\delta_{i+1} = (\nu_i - a_i)n_i + \delta_i + \sum_{j=1}^{i-1} u_j^{(i)} n_j + u_0^{(i)} n_1 + b_0^{(i)} - b_0^{(i-1)}. \quad (13)$$

The high-level ideas behind these parameters are as follows. By Lemma 3, the quantities  $\delta_1, \dots, \delta_{h+1}$  control the distances via the BCH bound and we would like to make these quantities as large as possible given the target dimensions. We need to make sure that the  $i$ th local code has a run of consecutive zeros of length  $\delta_i - 1$ , and our budget of creating such a run is limited by the dimension. Therefore, we seek to rely on the already present runs of zeros of the  $j$ th local codes,  $j < i$ , and spend the budget frugally on the way to optimality. The quantities  $b_j^{(i)}$  serve the purpose bridging the “distance gap” (the gaps between runs of zeros) between the local codes in levels  $j$  and  $j+1$  on the way to ensure the distance of the  $i$ th code.

As for the dimensions of the local codes, by Lemma 3 they are determined by  $\mathcal{Z}_i, 1 \leq i \leq h+1$ . The cardinality of  $\mathcal{Z}_i$  is established in the next claim.

**Proposition 5.** For  $1 \leq i \leq h+1$ , we have  $|\mathcal{Z}_i| = n_i - r_i$ .

The proof of this proposition proceeds by induction and is given in Appendix A. An examination of the proof also gives a better understanding of the parameters  $a_i, b_i$  introduced above.

On account of Proposition 5, the locality parameters of the code  $\mathcal{C}$  are  $(r_i, \delta_i), 1 \leq i \leq h$ . Furthermore,  $\dim(\mathcal{C}) = r_{h+1}$ , and by the BCH bound  $d(\mathcal{C}) \geq \delta_{h+1}$ .

Sufficient conditions for optimality of the code  $\mathcal{C}$  are given in the following lemma whose proof is given in Appendix B.

**Lemma 6.** Suppose that for  $i \geq 2$  and  $2 \leq s \leq i$  the following conditions are satisfied:

$$\begin{aligned} \left\lfloor \frac{r_{s+1}}{r_s} \right\rfloor \left\lfloor \frac{r_s}{r_1} \right\rfloor - \left\lfloor \frac{r_{s+1}}{r_1} \right\rfloor &= u_0^{(s)} + u_1^{(s)} + \sum_{j=2}^{s-1} u_j^{(s)} \left\lfloor \frac{r_j}{r_1} \right\rfloor \\ \left\lfloor \frac{r_{s+1}}{r_s} \right\rfloor \left\lfloor \frac{r_s}{r_l} \right\rfloor - \left\lfloor \frac{r_{s+1}}{r_l} \right\rfloor &= u_l^{(s)} + \sum_{j=l+1}^{s-1} u_j^{(s)} \left\lfloor \frac{r_j}{r_l} \right\rfloor, \end{aligned} \quad (14)$$

where the first condition holds for  $s \geq 2$  and the second for  $2 \leq l \leq s-1$ . Then for  $1 \leq i \leq h$ , we have

$$\delta_{i+1} = n_{i+1} - r_{i+1} + \delta_i - \sum_{l=1}^i \left\lfloor \frac{r_{i+1}}{r_l} \right\rfloor (\delta_l - \delta_{l-1}).$$

It follows from Lemma 3, Lemma 6, and the bound (2) that the code  $\mathcal{C}$  is an  $(n = n_{h+1}, k = r_{h+1})$  optimal H-LRC code with local parameters  $(r_i, \delta_i), 1 \leq i \leq h$ . Clearly, when  $h = 0$  our construction gives an  $(n_1, r_1)$  MDS code and when  $h = 1$ , it gives an  $(n_2, r_2)$  optimal LRC code of [35]. For  $h = 2$ , conditions (14) take a simpler form:

$$\left\lfloor \frac{r_3}{r_2} \right\rfloor \left\lfloor \frac{r_2}{r_1} \right\rfloor - \left\lfloor \frac{r_3}{r_1} \right\rfloor = \left\lfloor \frac{b_1 + b_2}{r_1} \right\rfloor. \quad (15)$$

We note that the condition of [31, Theorem 2.6] is easily seen to be equivalent to (15). Another known case of optimality, the divisibility conditions  $r_i | r_{i+1}, i = 1, \dots, h$ , is also covered by Lemma 6 (in this case both the left-hand sides and the right-hand sides of (14) are zero).

Let us give a general example of the choice of parameters that ensures optimality. Suppose that  $r_1 \geq 2^h$  and  $r_{i+1} = 2r_i - 1$  for  $1 \leq i \leq h$ . Then conditions (14) are satisfied. Indeed, we have  $r_i = 2^{i-1}(r_1 - 1) + 1$  for  $1 \leq i \leq h+1$ . Therefore, for  $1 \leq j \leq i \leq h+1$  we have

$$\left\lfloor \frac{r_i}{r_j} \right\rfloor = \left\lfloor 2^{i-j} - \frac{2^{i-j} - 1}{2^j(r_1 - 1) + 1} \right\rfloor = 2^{i-j},$$

where the last equality follows because  $r_1 \geq 2^h$ . It follows that the left-hand sides of conditions (14) are zero. On the other hand, we have  $b_i = 1$  for all  $1 \leq i \leq h$ . Since  $r_1 \geq 2^h$ , we have  $u_l^{(s)} = 0$  for all  $1 \leq s \leq h$  and  $0 \leq l \leq s-1$ , and thus, the right-hand sides are also zero, which confirms the optimality claim.

**Proposition 7.** Suppose that the conditions (14) are satisfied, then the code  $\mathcal{C}$  is a strongly optimal H-LRC code in the sense of Sec. 4A.

*Proof:* It suffices to show that the dimension of the  $i$ -th local code equals  $r_i$  for all  $i$ .

By assumption, we have  $r_{i+1} > r_i$  and thus  $a_i \geq 2$  for  $1 \leq i \leq h$ . It is not difficult to verify from (13) that  $\delta_{i+1} \leq (\nu_i - a_i + 1)n_i \leq n_{i+1} - n_i$  for all  $1 \leq i \leq h$ . We claim that  $g(\alpha^{n-n_i+t_i}) \neq 0$  for every  $t_i \in \mathcal{T}_i, 1 \leq i \leq h$  where

$$\mathcal{T}_i = \{n\} \cup (\{1, \dots, n_i - 1\} \setminus \mathcal{Z}_i + n - n_i).$$

Then by the second part of Lemma 3 the dimension of the  $i$ -th local code equals  $r_i$  and the strong optimality follows.

Now let us show  $n - n_i + t_i \notin \mathcal{Z}$ . Observe that the set  $\mathcal{Z}$  contains  $n/n_i$  copies of  $\mathcal{Z}_i$  and the set  $\mathcal{T}_i$  is the complement

to the last one of those copies with respect to  $\{1, \dots, n_i\}$ . Indeed, we have  $n - n_i + t_i = t_i + (n/n_i - 1)n_i \notin \{n\} \cup (\mathcal{Z}_i + (n/n_i - 1)n_i)$ . Now consider the last copy of  $\mathcal{Z}_{i+1}$  contained in  $\mathcal{Z}$ . Obviously, it contains the last copy of  $\mathcal{Z}_i$ . To establish  $n - n_i + t_i \notin \mathcal{Z}$ , it remains to show  $n - n_i + t_i$  is not in the last copy of  $\mathcal{D}_{i+1}$ , namely,  $n - n_i + t_i \notin \mathcal{D}_{i+1} + n - n_{i+1}$ . Since  $\delta_{i+1} \leq n_{i+1} - n_i$  as we observed above and  $n_{i+1} - n_i + t_i \geq n_{i+1} - n_i + \delta_i$ , we have  $n_{i+1} - n_i + t_i \notin \mathcal{D}_{i+1}$ . It follows that  $n - n_i + t_i \notin \mathcal{D}_{i+1} + n - n_{i+1}$ . Therefore,  $n - n_i + t_i \notin \mathcal{Z}$  and  $g(\alpha^{n-n_i+t_i}) \neq 0$  for every  $t_i \in \mathcal{T}_i, 1 \leq i \leq h$ .

In the case that  $r_{i+1} = r_i$  for some  $1 \leq i \leq h$  (although we rule out this trivial case in Definition 5), the code  $\mathcal{C}$  is still strongly optimal if the optimality conditions are satisfied. In fact, if  $r_{i+1} = r_i$  then from (13) we have  $\delta_{i+1} = (\nu_i - a_i)n_i + \delta_i = n_{i+1} - n_i + \delta_i \notin \mathcal{D}_{i+1}$ . By similar arguments as above, we have  $n - n_i + t_i \notin \mathcal{Z}$  for every  $t_i \in \mathcal{T}_i, 1 \leq i \leq h$ , and thus establish the strong optimality of the code. ■

We conclude with a numerical example that shows that the assumptions on the parameters can be simultaneously satisfied for moderate values of the length and alphabet size.

**Example 1.** Consider the case  $h = 3$ . Let  $r_1 = 2$  and  $\delta_1 = 2$ . Then  $n_1 = 3$ . Let  $(n_2, r_2) = (9, 3)$ ,  $(n_3, r_3) = (27, 5)$ , and  $(n_4, r_4) = (81, 7)$ . Then our construction (with designed distances found from (13)) gives rise to a strongly optimal H-LRC code of length  $n = n_4$  and dimension  $k = r_4$  with local distances  $\delta_2 = 6, \delta_3 = 17$  and distance  $d = \delta_4 = 53$  over a finite field  $\mathbb{F}_q$  where  $81|(q-1)$  (for example, we can take  $q = 163$ ).

### B. Hierarchical cyclic codes of unbounded length

In this section we construct a family of H-LRC codes with distance  $d = \delta_h + 1, h \geq 1$  and unbounded length. The construction combines the idea of [27] with H-LRC codes of the previous section.

Let  $1 < r_1 < r_2 < \dots < r_h$  be integers. Let  $1 = \delta_0 < \delta_1$  and let  $\delta_2, \delta_3, \dots, \delta_h$  be as in (13). Again, we put  $n_1 = r_1 + \delta_1 - \delta_0$  and let  $n_{i+1} = \nu_i n_i$  for  $1 \leq i \leq h-1$ , where  $\nu_i \geq \lceil r_{i+1}/r_i \rceil$  is an integer. Let  $\mathbb{F}_{q^m}, m \geq 1$  be a finite field and let  $n_h|(q-1)$ . Let  $n = q^m - 1$  and observe that  $n_h|n$ . Let  $\alpha \in \mathbb{F}_{q^m}$  be a primitive  $n$ -th root of unity. Let  $\mathcal{Z}_h$  be constructed by the procedure in (9) and (10). Finally, define

$$\mathcal{L} = \bigcup_{s=0}^{n/n_h-1} (\mathcal{Z}_h + s n_h), \quad \mathcal{Z} = \mathcal{L} \cup \{0\}. \quad (16)$$

Consider a cyclic code  $\mathcal{C}$  with generator polynomial

$$g(x) = \prod_{t \in \mathcal{Z}} (x - \alpha^t). \quad (17)$$

As is easily seen,  $g(x) \in \mathbb{F}_q[x]$ . Indeed,

$$\begin{aligned} g(x) &= (x-1) \prod_{t \in \mathcal{L}} (x - \alpha^t) \\ &= (x-1) \prod_{s=0}^{n/n_h-1} \prod_{t \in \mathcal{Z}_h} (x - \alpha^{sn_h+t}) \\ &= (x-1) \prod_{t \in \mathcal{Z}_h} (x^{n/n_h} - \alpha^{nt/n_h}). \end{aligned}$$

For the last equality we note that  $x^{n/n_h} - \alpha^{nt/n_h} = \prod_{s=0}^{n/n_h-1} (x - \alpha^{sn_h+t})$ . Observe that for  $t \in \mathcal{Z}_h$  we have  $(\alpha^{nt/n_h})^{q-1} = 1$  since  $n_h|(q-1)$ . It follows that  $\alpha^{nt/n_h} \in \mathbb{F}_q$  for  $t \in \mathcal{Z}_h$  and thus  $g(x) \in \mathbb{F}_q[x]$ .

**Proposition 8.** Let  $\mathcal{C} = \langle g(x) \rangle \in \mathbb{F}_q[x]/(x^n - 1)$  be a cyclic code. Then  $\dim(\mathcal{C}) = nr_h/n_h - 1$ ,  $d(\mathcal{C}) \geq \delta_h + 1$ , and the locality parameters are  $(r_i, \delta_i)$  for  $1 \leq i \leq h$ .

*Proof:* The dimension of  $\mathcal{C}$  is found as

$$k = n - \deg(g(x)) = n - 1 - \frac{|\mathcal{Z}_h|n}{n_h} = \frac{nr_h}{n_h} - 1,$$

where the last equality follows by Proposition 5. The distance of the code  $\mathcal{C}$  is  $d \geq \delta_h + 1$  since  $g(x)$  has consecutive roots  $\alpha^t, t = 0, \dots, \delta_h - 1$ .

The locality parameters of the code  $\mathcal{C}$  follow immediately by Lemma 3 and Proposition 5. ■

The next lemma provides the conditions when the code  $\mathcal{C}$  is optimal. Its proof amounts to a calculation based on Lemma 6 and Proposition 8.

**Lemma 9.** Suppose that for  $1 \leq i \leq h-1$ , conditions (14) are satisfied. Further, suppose that

$$\frac{n}{n_h} \left\lceil \frac{r_h}{r_l} \right\rceil = \left\lceil \frac{k}{r_l} \right\rceil, \quad 1 \leq l \leq h-1. \quad (18)$$

Then the code  $\mathcal{C}$  is optimal.

*Proof:* By Lemma 5, we have

$$\delta_h = n_h - r_h + \delta_{h-1} - \sum_{l=1}^{h-1} \left\lceil \frac{r_h}{r_l} \right\rceil (\delta_l - \delta_{l-1}). \quad (19)$$

By Proposition 8, we have  $k = nr_h/n_h - 1$ . Using the bound (20), the distance of the code cannot exceed

$$\begin{aligned} n - k + \delta_h - \sum_{l=1}^h \left\lceil \frac{k}{r_l} \right\rceil (\delta_l - \delta_{l-1}) \\ = n - \frac{nr_h}{n_h} + 1 + \delta_h - \frac{n}{n_h} (\delta_h - \delta_{h-1}) \\ - \sum_{l=1}^{h-1} \left\lceil \frac{k}{r_l} \right\rceil (\delta_l - \delta_{l-1}) \end{aligned} \quad (20)$$

$$\begin{aligned} = 1 + \delta_h + \frac{n}{n_h} \sum_{l=1}^{h-1} \left\lceil \frac{r_h}{r_l} \right\rceil (\delta_l - \delta_{l-1}) \\ - \sum_{l=1}^{h-1} \left\lceil \frac{k}{r_l} \right\rceil (\delta_l - \delta_{l-1}) \end{aligned} \quad (21)$$

$$= 1 + \delta_h, \quad (22)$$

where (20) follows since  $r_h > 1$  implies  $\lceil k/r_h \rceil = n/n_h$ , in (21) we used (19), and (22) follows by (18). Hence, the code  $\mathcal{C}$  has the largest possible distance  $d = \delta_h + 1$ . ■

In particular, the conditions in Lemma 9 are satisfied when  $r_i|r_{i+1}, i = 1, \dots, h-1$  and  $r_h|k$ .

As in Sec. II-A, the code  $\mathcal{C}$  constructed above in this section has the strong optimality property if the optimality conditions in Lemma 9 are satisfied. Specifically, the main difference between the construction in this section and the one in the previous section is in the final step of constructing the defining set  $\mathcal{Z}$ , which also includes element 0 (i.e.,  $\alpha^0$ ). By an argument

similar to Sec. II-A, one can show  $n - n_i + t_i \notin \mathbb{Z}$  for every  $t_i \in (\{0, \dots, n_i - 1\} \setminus \mathbb{Z}_i) + (n - n_i)$ ,  $1 \leq i \leq h$  and so strong optimality follows.

**Example 2.** Consider the case  $h = 3$ . Let  $r_1 = 2$  and  $\delta_1 = 2$ . Then  $n_1 = 3$ . Let  $(n_2, r_2) = (9, 3)$  and  $(n_3, r_3) = (27, 5)$ . Let  $m \geq 1$  be an arbitrary integer and  $q = 163$ . Then our construction (with designed local distances given by (13)) gives rise to a strongly optimal H-LRC code of length  $n = 163^m - 1$  and dimension  $k = 2(163^m - 1)/9$  with local distances  $\delta_2 = 6$ ,  $\delta_3 = 17$  and distance  $d = 18$  over  $\mathbb{F}_q$ .

Recall that [4] shows that the length of an optimal LRC code in the general case cannot be greater than a certain power of the alphabet size  $q$ . Using similar arguments, it might be possible to derive upper bounds on the length of optimal H-LRC codes in the general case; however already in the case of  $(r, \delta)$  locality addressed in [6] (with just a single level of hierarchy), following this route requires cumbersome calculations.

### III. CONVOLUTIONAL CODES WITH LOCALITY

It has been recognized a long while ago that quasi-cyclic codes can be encoded convolutionally, and multiple papers constructed families of convolutional codes from their quasi-cyclic counterparts [9], [19], [36]. In this section, we present a family of convolutional codes with locality by relying on the tailbiting version of convolutional codes [32]. We single out this approach because it enables us to establish the locality properties of convolutional codes based on the properties of cyclic H-LRC codes constructed above in this paper.

We begin with a brief reminder of the basic notions for convolutional codes [18]. Let  $D$  be an indeterminate and let  $\mathbb{F}_q(D)$  be the field of rational functions of one variable over  $\mathbb{F}_q$ . A  $q$ -ary  $(n, k)$  convolutional code  $\mathcal{C}$  is a linear  $k$ -dimensional subspace of  $\mathbb{F}_q(D)^n$ . A generator matrix  $G(D) = (g_{ij}(D))$  of the code  $\mathcal{C}$  is a  $k \times n$  matrix with entries in  $\mathbb{F}_q(D)$  whose rows form a basis of  $\mathcal{C}$ . Thus, the code  $\mathcal{C}$  is a linear space  $\{u(D)G(D) \mid u(D) \in \mathbb{F}_q(D)^k\}$ . The matrix  $G(D)$  can be transformed to the polynomial form by multiplying every element by the least common denominator of its entries. The transformed matrix generates the same code  $\mathcal{C}$ , and so in the sequel we will consider only *polynomial generator matrices*. Below we will assume that the generator matrix  $G(D)$  is a  $k \times n$  matrix with entries in  $\mathbb{F}_q[D]$ , where  $\mathbb{F}_q[D]$  is the ring of polynomials over  $\mathbb{F}_q$ .

For  $1 \leq i \leq k$ , the degree  $m_i$  of the  $i$ -th row of  $G(D)$  is the maximum degree of the entries in row  $i$ , namely,  $m_i = \max_{1 \leq j \leq n} \deg(g_{ij}(D))$ . As with linear block codes, the encoding of a convolutional code depends on the choice of a generator matrix. The maximum degree  $M := \max_{1 \leq i \leq k} m_i$  is called the *memory* of the encoder. The generator matrix of the code  $\mathcal{C}$  can also be written in the form

$$G = \begin{bmatrix} G_0 & G_1 & \dots & G_M \\ & G_0 & G_1 & \dots & G_M \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix}, \quad (23)$$

where each  $G_i$  is a  $k \times n$  matrix over  $\mathbb{F}_q$ . The codeword of the code  $\mathcal{C}$  is obtained as a product  $uG$ , where  $u$  is a semi-infinite input sequence of symbols of  $\mathbb{F}_q$ .

With a given convolutional code  $\mathcal{C}$  one can associate a multitude of distance measures. In direct analogy with block codes, one defines the *free distance* of the code  $\mathcal{C}$  as the minimum Hamming weight of the Laurent expansions of the nonzero codewords.

Another distance measure of interest is the so-called *column distance* of the code [18, p.162]. To define it, let  $\mathcal{C}_{[0,j]}$  be the truncated code of  $\mathcal{C}$  at the  $j$ -th time instant,  $j \geq 0$ , namely,

$$\mathcal{C}_{[0,j]} = \left\{ c_{[0,j]}(D) = \sum_{i=0}^j c_i D^i \mid c(D) = \sum_{i \geq 0} c_i D^i \in \mathcal{C} \right\}.$$

This is a linear block code of length  $n(j+1)$ , and by (23) its generator matrix can be written in the form

$$G_j^c = \begin{bmatrix} G_0 & G_1 & \dots & G_j \\ & G_0 & \dots & G_{j-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix} \quad (24)$$

where we put  $G_l = 0$  for  $l > M$ . Clearly, the code  $\mathcal{C}_{[0,j]}$  is obtained by truncating the code  $\mathcal{C}$  to its first  $j+1$  entries. A codeword of  $\mathcal{C}_{[0,j]}$  has the form  $(c_0, c_1, \dots, c_j)$ , where for  $j \leq M$  and each  $l = 0, 1, \dots, j$

$$c_l = \sum_{i=0}^l u_i G_{l-i}, \quad (25)$$

where  $c_l = (c_l^{(1)}, \dots, c_l^{(n)})$  for each  $l$ .

We assume throughout that  $G_0$  has full rank, so the mapping  $\mathbb{F}_q^k \xrightarrow{G_0} \mathbb{F}_q^n$  given by  $u_0 G_0 \mapsto c_0$  is injective.

**Definition 4.** For  $j \geq 0$  the  $j$ -th column distance of  $\mathcal{C}$  is given by

$$d_j^c = \min\{\text{wt}(c_{[0,j]}(D)) \mid c_{[0,j]}(D) \in \mathcal{C}_{[0,j]}, c_0 \neq 0\}.$$

Clearly, the value of  $d_j^c$  is at least the minimum distance of the truncated code  $\mathcal{C}_{[0,j]}$ . This follows because for the column distance we seek the minimum of pairwise distances of codewords that differ in the first coordinate, while the standard minimum distance computation does not involve this assumption. In many cases the column distance is in fact strictly greater. This remark is important for the sliding window repair which enables one to correct more erasures than would be possible for block codes relying on their minimum distance.

Convolutional codes support several forms of erasure repair. One of them, called the sliding window repair [28], [37], is based on the column distance and is used to correct erasures in streaming applications [37]. We illustrate the idea of sliding window repair in Fig. 1, representing a code sequence of the code  $\mathcal{C}$  as a semi-infinite matrix whose columns are length  $n$  vectors  $c_l$ ,  $l \geq 0$ , and whose row  $c^{(i)}$ ,  $i = 1, \dots, n$  represents the stream formed by the  $i$ th coordinates of the symbols  $c_l$ ,  $l = 0, 1, \dots$ . We begin with fixing  $j$  based on the value of the column distance  $d_j^c$  of the code. The box in the figure shown with dashed lines represents the window of length  $j+1$  that

contains the truncated code at time  $l \geq j$ . The erasures within the sliding window can clearly be repaired as long as their number at no point exceeds  $d_j^c - 1$ .

Having in mind streaming applications, one may argue that a more efficient way of repairing erasures is to rely either on the symbols at a fixed time instant, or on a small group of symbols contained within the same stream  $i$ . Accordingly, in the next two subsections we define two types of locality for convolutional codes, calling them the column and row localities.

#### A. Convolutional codes with column locality

Column locality was introduced in [28]. First let us define the  $i$ th column code  $\mathcal{C}_i, i \geq 0$  of a convolutional code  $\mathcal{C}$  as a block code of length  $n$  given by

$$\mathcal{C}_i = \{c_i \mid c(D) \in \mathcal{C}\}.$$

We say that a convolutional code  $\mathcal{C}$  has  $(r, \delta)$  column locality if for all  $i$  the codes  $\mathcal{C}_i$  have the  $(r, \delta)$  locality property.

The results in [28] are based on a version of this definition that requires that *only the code*  $\mathcal{C}_M$  support  $(r, \delta)$  locality. This restriction may seem too narrow until one realizes that if locality is present in the code  $\mathcal{C}_M$ , then every code  $\mathcal{C}_i, i \geq 0$  has the  $(r, \delta)$  locality property. This follows immediately from (25) and the definition of  $\mathcal{C}_i$  because  $\mathcal{C}_i = \mathcal{C}_M$  for  $i > M$  and  $\mathcal{C}_i$  forms a linear subcode of  $\mathcal{C}_M$  otherwise. The only difference between this definition and the one given above is that under the approach of [28], every code  $\mathcal{C}_i$  has similarly aligned repair groups which are propagated from the repair groups of  $\mathcal{C}_M$ , while our definition allows differently aligned repair groups for different values of  $i$ .

To enable local repair, we simply assume that every column of the codeword forms a block code with  $(r, \delta)$  locality. An example is given in Fig 2, demonstrating sliding window repair combined with column locality.

#### B. Convolutional codes with row locality

In this section we introduce and study another notion of locality for convolutional codes. Given a convolutional code  $\mathcal{C}$ , define the  $i$ th row code  $\mathcal{C}_{[0,j]}^{(i)}, 1 \leq i \leq n$  truncated at  $j$ th time instant,  $j \geq 0$ , as follows:

$$\mathcal{C}_{[0,j]}^{(i)} = \{c_{[0,j]}^{(i)} = (c_0^{(i)}, \dots, c_j^{(i)}) \mid c \in \mathcal{C}\}. \quad (26)$$

**Definition 5.** We say that  $\mathcal{C}$  has  $(r, \delta)$  row locality if for all  $t \geq 0$  the codes  $\mathcal{C}_{[t,t+j]}^{(i)}, 1 \leq i \leq n$  have the  $(r, \delta)$  locality property, where  $j \geq 0$  is fixed.

In the case of tailbiting codes, it is more convenient to give this definition in the following form, which will also be used in our constructions below.

**Definition 6.** Let  $j \geq 0$  be fixed. A convolutional code  $\mathcal{C}$  has  $(r, \delta)$  row locality at time  $j$  if every code  $\mathcal{C}_{[0,j]}^{(i)}, 1 \leq i \leq n$  has  $(r, \delta)$  locality.

We give two examples of repair with row locality. Namely, Figure 3 illustrates Def. 5 while Figure 4 applies to the case of tailbiting codes and Def. 6. Let us stress that whenever local

repair by rows is not possible, we fall back on sliding window repair relying on the column distance of the truncated code.

The problem that we address is to construct convolutional codes with locality and large column distance. This is similar to the problem studied in [28] and also to the case of block codes with locality and large minimum distance. We begin with deriving an upper bound on the column distance of the truncated code with either column or row locality property.

**Proposition 10.** (a) Let  $\mathcal{C}$  be an  $(n, k)$  convolutional code with  $(r, \delta)$  (column or row) locality. Then for any  $j \geq 0$ , the  $j$ -th column distance satisfies

$$d_j^c \leq (n - k)(j + 1) + \delta - \left\lceil \frac{k}{r} \right\rceil (\delta - 1). \quad (27)$$

(b) Equality in (27) implies that for all  $i \leq j$ , the  $i$ -th column distance satisfies

$$d_i^c = (n - k)(i + 1) + \delta - \left\lceil \frac{k}{r} \right\rceil (\delta - 1). \quad (28)$$

The proof is given in Appendix Q.

Part (b) of this proposition is similar to the propagation of the column distance optimality property in the case of general convolutional codes proved in [11]. Namely, the Singleton bound implies that the column distance for all  $j$  satisfies

$$d_j^c \leq (n - k)(j + 1) + 1, \quad (29)$$

and equality for a given  $j$  implies that all the other distances  $d_i^c, i \leq j$  also attain their versions of the Singleton bound with equality.

#### C. Convolutional codes and quasicyclic codes

A transformation between these two code families was constructed in [32], and it has led to a broader family of convolutional codes and trellises now known as *tailbiting codes* (tailbiting trellises) [18]. An  $(n(m + 1), k(m + 1))$  quasicyclic code can be defined by a generator matrix

$$G = (G_{ij}), \quad i = 0, \dots, k - 1, j = 0, \dots, n - 1$$

where each  $G_{ij}$  is an  $(m + 1) \times (m + 1)$  circulant matrix. With a given matrix  $G_{ij}$  we associate a polynomial  $g_{ij}(D) = \sum_{l=0}^m g_l D^l$ , where  $(g_0, \dots, g_m)$  is the first row of the matrix. Then the  $k \times n$  generator matrix  $G(D) = (g_{ij}(D))$  defines an  $(n, k)$  convolutional code. The authors of [32] showed that if one takes the input sequences of the convolutional code in the form

$$u(D) = \sum_{l=-M}^m u_l D^l \text{ such that } u_{-s} = u_{m+1-s} \text{ for } s = 1, \dots, M, \quad (30)$$

then the convolutional code is equivalent to the quasicyclic code defined above. In other words, the quasicyclic code can be encoded convolutionally, and the convolutional code with “symmetric” input sequences as in (30) is exactly the quasicyclic code.



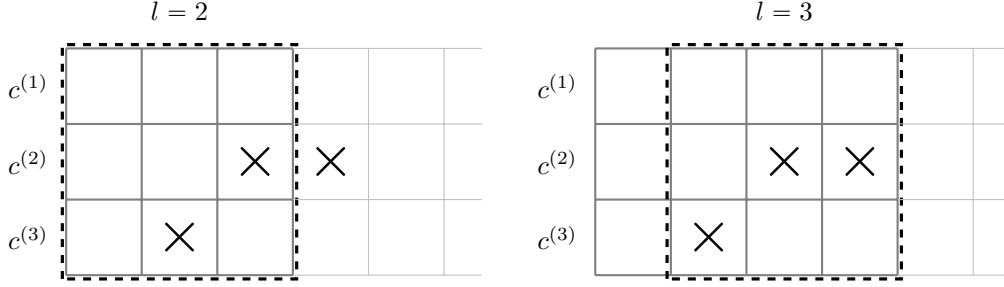


Fig. 1: SLIDING WINDOW REPAIR. Suppose  $\mathcal{C}$  is an  $(3, 2)$  convolutional code with  $d_2^c = 4$ . Let  $(c^{(1)}, c^{(2)}, c^{(3)}) \in \mathcal{C}$  be a codeword, where the crosses denote erasures. At time instant  $l = 2$ , there are two erasures in the window of length three (dashed box in the left figure), which is less than the 2nd column distance. However, neither of the two erasures are in the first column of the window, and thus their recovery is postponed until later. At time  $l = 3$  (right figure), the sliding window contains 3 erasures, of which one is in the first column. This erasure can be recovered from the other symbols in the window. The remaining erasures are corrected in the next steps as long as the number of erasures in the window does not exceed  $d_2^c - 1$ .

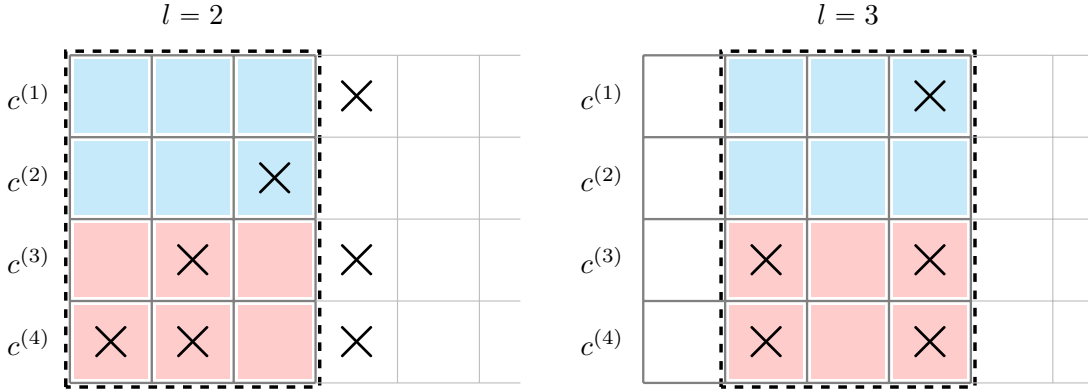


Fig. 2: SLIDING WINDOW REPAIR WITH COLUMN LOCALITY. Suppose  $\mathcal{C}$  is a  $(4, 2)$  convolutional code with  $(1, 2)$  column locality and  $d_2^c = 6$ . Let  $(c^{(1)}, c^{(2)}, c^{(3)}, c^{(4)}) \in \mathcal{C}$  be a codeword, where the crosses denote erasures, and different repair groups in the columns are shown in different colors. At time  $l = 2$ , the window of length  $j + 1 = 3$  contains 4 erasures. Of these, the symbols  $c_0^{(4)}, c_2^{(2)}$  can be recovered within their repair groups. Then for  $l = 3$  the window contains 5 erasures, of which two in the first column can be repaired from the other symbols in the window, while the symbol  $c_3^{(1)}$  can be recovered locally.

#### D. A family of tailbiting convolutional codes with row locality

We come to the main result of this section, which is a construction of a family of convolutional codes with  $(r, \delta)$  row locality. This family of codes has the largest possible minimum distance for the truncated code  $\mathcal{C}_{[0, j]}$ , however we stop short of showing that the  $j$ -th column distance attains (27) with equality. The construction is achieved by exploiting the connection between quasi-cyclic codes and convolutional codes discussed above. In high-level terms, our plan is to construct a cyclic code from its set of zeros chosen according to the procedure in Sec. III, writing it in a quasicyclic form (via a circulant generator matrix) and to construct a convolutional code using the technique discussed above. We note that the lower bounds on the column distance of the codes constructed here and in [28] coincide. At the same time, the field size needed for our construction is linear in the output length  $n$  of the code at each time instant whereas the construction in [28] requires exponentially sized alphabet.

Let us first construct an  $(n(j + 1), k(j + 1))$  cyclic LRC codes with  $(r, \delta)$  locality. We proceed similarly to Sec. III-A. We will need a few assumptions regarding the parameters of the code. Let  $j \geq 0$  be such that  $k \leq j + 1 \leq n$  and that

$j + 1 = (r + \delta - 1)\nu$  where  $\nu \geq 1$ . Let  $\mathbb{F}_q$  be a finite field such that  $n(j + 1) \mid (q - 1)$  and let  $\alpha \in \mathbb{F}_q$  be a primitive root of unity of order  $n(j + 1)$  in  $\mathbb{F}_q$ .

The set of zeros of the code is obtained as follows. let  $\mathcal{Z}_1 = \{1, \dots, \delta - 1\}$ . Using (III) and setting  $\mathcal{D}_2 = \mathcal{Z}_1$ , we have

$$\mathcal{Z}_2 = \bigcup_{l=0}^{\nu-1} (\mathcal{Z}_1 + l(r + \delta - 1)). \quad (31)$$

Further, let  $\mathcal{L}_3 = \bigcup_{l=0}^{n-1} (\mathcal{Z}_2 + l(j + 1))$  and let  $\mathcal{D}_3 = \{1, \dots, \delta_3 - 1\}$ , where

$$\delta_3 = (n - k)(j + 1) + \delta - \left\lceil \frac{k(j + 1)}{r} \right\rceil (\delta - 1). \quad (32)$$

Finally, put  $\mathcal{Z} = \mathcal{L}_3 \cup \mathcal{D}_3$  and let  $\mathcal{B}$  be the cyclic code with generator polynomial  $g_{\mathcal{B}}(x) = \prod_{t \in \mathcal{Z}} (x - \alpha^t)$ . Note for future use that the complement of the set  $\mathcal{D}_3$  in the set of exponents of  $\alpha$  has cardinality

$$\begin{aligned} |\bar{\mathcal{D}}_3| &= n(j + 1) - (\delta_3 - 1) \\ &= k(j + 1) + \left( \left\lceil \frac{k(j + 1)}{r} \right\rceil - 1 \right) (\delta - 1). \end{aligned} \quad (33)$$



let  $l = 0$ . By Lemma 3, Eq. (8), the code  $\mathcal{B}^{(0)}$  given by

$$\mathcal{B}^{(0)} = \{(c_0, c_{n\nu}, \dots, c_{(r+\delta-2)n\nu}) \mid (c_0, \dots, c_{n(j+1)-1}) \in \mathcal{B}\}$$

has dimension at most  $r$  and minimum distance at least  $\delta$ . This implies that the coordinates that are multiples of  $n\nu$  isolate a repair group of the code  $\mathcal{B}_0$ . By shifting this set of coordinates to the right by  $n$  positions, we obtain another repair group of  $\mathcal{B}_0$ , which is disjoint from the first one. After several more shifts we will reach the set of coordinates  $\{n(\nu-1), n(\nu-1) + n\nu, \dots, n(\nu-1) + (r+\delta-2)n\nu\}$ . The collection of the sets constructed along the way forms a partition of the support of  $\mathcal{B}_0$  into disjoint repair groups. The same argument works for every code  $\mathcal{B}_l, 1 \leq l \leq n-1$  whose repair groups are formed by shifting the repair groups of  $\mathcal{B}_0$  to the right by  $l$  positions. This concludes the proof of Part (a).

Let us prove Part (b). Recalling the discussion in the beginning of Sec. 4.1, it is possible to represent the generator matrix  $G$  of the code to have rows of the form

$$((\alpha^t)^{n(j+1)-1}, (\alpha^t)^{n(j+1)-2}, \dots, 1), \\ t \in \{0, \dots, n(j+1)-1\} \setminus \mathcal{Z}, \quad (34)$$

(note the inverse order of the exponents). Let

$$\mathcal{J} = \{0, n/k, \dots, n(j+1) - n/k\}$$

be a subset of coordinates. As before, we label the columns of  $G$  by the exponents of  $\alpha$  from 0 to  $n(j+1)-1$  and consider a square  $k(j+1) \times k(j+1)$  submatrix  $G_{\mathcal{J}}$  formed of the columns with indices in  $\mathcal{J}$ . We claim that  $G_{\mathcal{J}}$  is invertible. Indeed, the rows of  $G_{\mathcal{J}}$  have the form

$$\alpha^{-t}((\alpha^{tn/k})^{k(j+1)}, (\alpha^{tn/k})^{k(j+1)-1}, \dots, \alpha^{tn/k}), \\ t \in \{0, \dots, n(j+1)-1\} \setminus \mathcal{Z},$$

and thus it forms a Vandermonde matrix generated by the set  $(\alpha^{tn/k})$  for all  $t$  outside the set of zeros  $\mathcal{Z}$ . We will assume that the submatrix  $G_{\mathcal{J}} = I_{k(j+1)}$  (the identity matrix) and continue to use the notation  $G$  for the resulting generator matrix of the code.

Let  $g_{i,0}, g_{i,1}, \dots, g_{i,n(j+1)-1}$  be the  $i$ th row of  $G$ , where  $i \in \{0, \dots, k-1\}$ . With an outlook of constructing convolutional codes later in this section, define the polynomials

$$g_i(D) = \sum_{s=0}^{n(j+1)-1} g_{i,s} D^s \\ g_{i,l}(D) = \sum_{s=0}^j g_{i,ns+i} D^s, \quad l = 0, 1, \dots, n-1. \quad (35)$$

Then we have

$$g_i(D) = \sum_{l=0}^{n-1} D^l \sum_{s=0}^j g_{i,ns+l} D^{ns} = \sum_{l=0}^{n-1} D^l g_{i,l}(D^n),$$

Since  $G_{\mathcal{J}}$  is the identity matrix, we have  $g_{i,in/k}(D) = 1$  and  $g_{i,l}(D) = 0$  for  $l \in \{0, n/k, \dots, n - n/k\} \setminus \{in/k\}$ .

To write the generator matrix in the circulant form given in the statement, we need to form the matrices  $G_{il}$ . This is accomplished by writing the coefficients of  $g_{i,j}(D)$  as the first row of  $G_{il}$  and filling the rest of this matrix by consecutive

cyclic shifts to the right. This yields the following  $(j+1) \times n(j+1)$  matrix

$$(G_{i,0} \quad G_{i,1} \quad \dots \quad G_{i,n-1}). \quad (36)$$

Note that each row in this matrix is a codeword of the code (equivalent to)  $\mathcal{B}$ . Finally, the matrix

$$G = \begin{pmatrix} G_{0,0} & G_{0,1} & \dots & G_{0,n-1} \\ G_{1,0} & G_{1,1} & \dots & G_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ G_{k-1,0} & G_{k-1,1} & \dots & G_{k-1,n-1} \end{pmatrix}. \quad (37)$$

formed of the rows (36) for  $i = 0, \dots, k-1$  generates a code equivalent to the code  $\mathcal{B}$ . ■

This theorem gives an explicit representation of  $\mathcal{B}$  as a quasisyclic code, and we can use this representation to construct a convolutional code following the method in Sec. 4.1. Let  $G(D) = (g_{i,l}(D))$  be a  $k \times n$  generator matrix where  $g_{i,l}(D)$  is defined in (35). Since  $\deg(g_{i,l}(D)) \leq j$ , we conclude that the memory of the generator matrix  $G(D)$  is  $M \leq j$ . Having (30) in mind, define an  $(n, k)$  tailbiting convolutional code over  $\mathcal{C} = \mathcal{C}_{[0,j]} \in \mathbb{F}_q[D]$  as a set of sequences

$$\mathcal{C} = \{c(D) \mid c(D) = u(D)G(D); u(D) = \sum_{s=-M}^j u_s D^s; \\ u_{-s} = u_{j+1-s}, s = 1, \dots, M\}.$$

Here  $j, k-1 \leq j \leq n-1$  is any integer such that  $(r+\delta-1) \mid (j+1)$  and  $k \mid n$ .

The next theorem states the main properties our construction.

**Theorem 12.** *The code  $\mathcal{C}$  has  $(r, \delta)$  row locality. When viewed as a block code, the minimum distance of  $\mathcal{C}$  attains the bound (3).*

*Proof:* For  $l = 0, \dots, n-1$ , we have

$$c^{(l)}(D) = \sum_{i=0}^{k-1} u^{(i)}(D) g_{i,l}(D).$$

Furthermore, since  $u_{-s} = u_{j+1-s}$  for  $s = 1, \dots, M$ , we have the following relation

$$c^{(l)}(D) = \sum_{i=0}^{k-1} u^{(i)}(D) g_{i,l}(D) \mod (D^{j+1} - 1).$$

In other words, we have

$$(c^{(0)} c^{(1)} \dots c^{(n-1)}) = (u^{(0)} u^{(1)} \dots u^{(k-1)}) G,$$

where the matrix  $G$  is defined in (37). This implies that the codes  $\mathcal{C}^{(l)}$  are exactly the codes  $\mathcal{B}_l$  defined in Theorem 4(a), viz.,  $\mathcal{C}^{(l)} = \mathcal{B}_l$  for  $l = 0, \dots, n-1$ . Since the code  $\mathcal{B}_l$  has  $(r, \delta)$  locality for  $l = 0, \dots, n-1$ , we conclude that the code  $\mathcal{C}$  has  $(r, \delta)$  row locality. Concluding, we have established that the convolutional code  $\mathcal{C}$  has  $(r, \delta)$  row locality.

As a block code,  $\mathcal{C}$  is equivalent to the code  $\mathcal{B}$ , which proves the last claim of the theorem. ■

The large minimum distance of the code  $\mathcal{C}$  is related to the performance of the (hard decision) Viterbi decoding of the code  $\mathcal{C}$ , and is therefore of interest in applications.

As remarked earlier, the constructed codes stop short of attaining the bound (27), and thus cannot be claimed to be optimal. Of course, as observed after Def. 4, the  $j$ th column distance of the code  $\mathcal{C}$  is at least the minimum distance of the code  $\mathcal{B}$ , given by (32), but a more precise estimate remains an open question. Nevertheless, we believe that extension of the basic construction of LRC codes to the case of convolutional codes carries potential for future research into their structure. In particular, the algebraic machinery of quasicyclic codes of [22], [25] could lead to new constructions, and it may also be possible to further extend these studies to codes over ring alphabets [26].

#### IV. BI-CYCLIC CODES WITH AVAILABILITY

The H-LRC codes constructed in Sec. I rely on  $h$  embedded recovering sets for each code coordinate, which are not disjoint. In this section we consider LRC codes with  $t$  disjoint recovering sets for each code coordinate, i.e., LRC codes with availability  $t$  (here we do not pursue a hierarchy of locality). This arises when the data is simultaneously requested by a large number of users, which suggests that the erased coordinate afford recovery from several nonoverlapping recovering sets in order to increase data availability. LRC codes with this property are defined as follows [29], [38].

**Definition 7** (LRC CODES WITH AVAILABILITY). *Let  $t \geq 1$  and  $(r_1, \delta_1), \dots, (r_t, \delta_t)$  be integers. A code  $\mathcal{C} \subset \mathbb{F}_q^n$  is said to have availability  $t$  with locality  $(r_1, \delta_1), \dots, (r_t, \delta_t)$  if for every coordinate  $j, 1 \leq j \leq n$  of the code  $\mathcal{C}$  there are  $t$  punctured codes  $\mathcal{C}^{(i)}, 1 \leq i \leq t$  such that  $j \in \text{supp}(\mathcal{C}^{(i)})$ ,  $i = 1, \dots, t$  and*

- (a)  $\dim(\mathcal{C}^{(i)}) \leq r_i$ ,
- (b)  $d(\mathcal{C}^{(i)}) \geq \delta_i$ ,
- (c)  $\bigcap_{i=1}^t \text{supp}(\mathcal{C}^{(i)}) = \{j\}$ .

Below we limit ourselves to the case  $t = 2$ . We say that two partitions  $\mathcal{P}_1, \mathcal{P}_2$  of the coordinates of the code are *orthogonal* (or *transversal*) if  $|P_1 \cap P_2| \leq 1$  for any  $P_1 \in \mathcal{P}_1, P_2 \in \mathcal{P}_2$  and every coordinate is contained in a pair of subsets  $X \in \mathcal{P}_1, Y \in \mathcal{P}_2$ . Orthogonal partitions enable multiple disjoint recovering sets and were used in [33] to construct codes with availability. A simple observation made in [33] is that product codes naturally yield orthogonal partitions, and it is possible to use products of one-dimensional cyclic codes to support this structure. A drawback of this approach is that product codes result in rather poor parameters of LRC codes with availability, in particular the rate of the resulting codes is low (although the alphabet is small compared to the code length [16], [33]). It is well known that the rate of product codes can be increased with no loss to the distance by passing to generalized concatenations of codes [5]. In this section we use a particular case of this construction given in [30] and sometimes called *hyperbolic codes*. The resulting LRC codes with availability have the same distance guarantee as simple product codes while having a much higher rate. As above, our starting point is the general method of Theorem 4, and we proceed similarly to Eq. (10).

We start with a finite field  $\mathbb{F}_q$  and assume that the code length  $n$  divides  $q - 1$ . We further choose the size of the

repair groups to be  $r_1, r_2$  and suppose that  $0 < r_1 \leq r_2$  and  $(r_1 + 1)|n$  and  $(r_2 + 1)|n$ . Further, let  $\nu_1 = n/(r_1 + 1)$  and  $\nu_2 = n/(r_2 + 1)$ . Let  $\alpha \in \mathbb{F}_q$  be a primitive  $n$ -th root of unity. To simplify the expressions below, we will construct codes with  $\delta_1 = \delta_2 = 2$ . To construct the defining set  $\mathcal{Z}$  of our code, let

$$\mathcal{L}_1 = \emptyset, \quad \mathcal{D}_1 = \{(0, 0)\}, \quad \mathcal{Z}_1 = \mathcal{L}_1 \cup \mathcal{D}_1. \quad (38)$$

Let us fix the designed distance of the code  $\mathcal{C}$  to be  $\delta \geq 2$ . Define

$$\mathcal{L}_{2,1} = \bigcup_{l_1=0}^{\nu_1-1} \bigcup_{j=0}^{n-1} (\mathcal{Z}_1 + (l_1(r_1 + 1), j)),$$

$$\mathcal{L}_{2,2} = \bigcup_{i=0}^{n-1} \bigcup_{l_2=0}^{\nu_2-1} (\mathcal{Z}_1 + (i, l_2(r_2 + 1))),$$

$$\mathcal{L}_2 = \mathcal{L}_{2,1} \cup \mathcal{L}_{2,2},$$

$$\mathcal{D}_2 = \{(i, j) \mid (i + 1)(j + 1) < \delta\}, \quad \mathcal{Z}_2 = \mathcal{L}_2 \cup \mathcal{D}_2.$$

Note that zeros are now indexed by pairs of exponents, and pairs are added element-wise. Finally, put  $\mathcal{Z} = \mathcal{Z}_2$ .

Consider a two-dimensional cyclic code  $\mathcal{C} = \langle g(x, y) \rangle$  of length  $n^2$ , where

$$g(x, y) = \prod_{(i,j) \in \mathcal{Z}} (x - \alpha^i)(y - \alpha^j). \quad (39)$$

**Lemma 13.** *The code  $\mathcal{C}$  has two disjoint recovering sets of size  $r_1$  and  $r_2$  for every coordinate.*

*Proof:* The proof relies on Lemma 3. For  $c \in \mathcal{C}$ , let us write  $c = (c_{i,j})$  where  $0 \leq i \leq n-1, 0 \leq j \leq n-1$ . Fix  $j$  and let  $\mathcal{C}^{(1)} = \{(c_{0,j}, c_{\nu_1,j}, \dots, c_{r_1\nu_1,j}) \mid c \in \mathcal{C}\}$  and let  $\mathcal{L}_{2,1}^j = \bigcup_{l_1=0}^{\nu_1-1} (\mathcal{Z}_1 + (l_1(r_1 + 1), j))$ . The generator polynomial for the punctured code  $\{(c_{0,j}, c_{1,j}, \dots, c_{n-1,j}) \mid c \in \mathcal{C}\}$  is given by  $y^j \prod_{(i,j) \in \mathcal{Z}} (x - \alpha^i)$ , which is divisible by  $\prod_{(i,j) \in \mathcal{L}_{2,1}^j} (x - \alpha^i)$ . Then, by arguments similar to Lemma 3 we conclude that the code  $\mathcal{C}^{(1)}$  has dimension at most  $r_1$  and distance at least 2. Since the code is cyclic in both dimensions, we also claim that every symbol of the code  $\mathcal{C}$  has a recovering set of size at most  $r_1$  for one erasure.

Repeating the above arguments for a fixed index  $i$ , we isolate another recovering of size  $r_2$  for every coordinate. Furthermore, the two recovering sets are disjoint by construction. ■

To estimate the distance, recall the following result about bicyclic codes.

**Lemma 14** (HYPERBOLIC BOUND [30]). *Suppose that the defining set of zeros of an  $n \times n$  bicyclic code contains a subset given by  $\mathcal{Z} = \{(i, j) : (i + 1)(j + 1) < d\}$ . Then the distance of the code is at least  $d$ .*

Thus, the distance of the code  $\mathcal{C}$  constructed above is at least  $\delta$ , and its dimension  $\dim(\mathcal{C}) = n^2 - |\mathcal{Z}|$ . Let us estimate the dimension from below. We have

$$\begin{aligned} |\mathcal{D}_2| &= \sum_{j=0}^{\delta-2} \left\lfloor \frac{\delta}{j+1} \right\rfloor \leq \delta \sum_{j=1}^{\delta-1} \frac{1}{j} \\ &\leq \delta \left( 1 + \int_1^{\delta-1} \ln(x-1) dx \right) \end{aligned}$$



$$\leq \delta(1 + \ln(\delta - 1)).$$

Therefore, we have

$$\begin{aligned} n^2 - |\mathcal{Z}| &\geq n^2 - |\mathcal{L}_2| - |\mathcal{D}_2| \\ &= n^2 - (r_1 + r_2 + 1)\nu_1\nu_2 - \delta(1 + \ln(\delta - 1)) \\ &= \frac{n^2 r_1 r_2}{(r_1 + 1)(r_2 + 1)} - \delta(1 + \ln(\delta - 1)). \end{aligned} \quad (40)$$

To compare this estimate of the dimension with product codes, let  $\mathcal{C}'$  be a direct product of two cyclic LRC codes of length  $n$  with locality  $r$  and distance  $\sqrt{\delta}$ . The defining set of this code are given by  $\mathcal{Z}' = \mathcal{L}' \cup \mathcal{D}'$ , where

$$\begin{aligned} \mathcal{L}' &= \{(i, j) \mid i, j = 1 \bmod (r + 1)\}, \\ \mathcal{D}' &= \{(i, j) \mid i, j = 1, \dots, \sqrt{\delta} - 1\}. \end{aligned}$$

Choosing  $r_1 = r_2 = r$  in our construction, we have  $\mathcal{L}_2 + (1, 1) = \mathcal{L}'$ . It is also easy to see that  $|\mathcal{Z}| \leq |\mathcal{Z}'|$  and thus,  $\dim(C) \geq \dim(\mathcal{C}')$ .

Let us write out an estimate for the rate of the constructed codes. Putting  $r_1 = r_2 = r$  in (40), we obtain for the rate of the code  $\mathcal{C}$  the following estimate in terms of the relative distance  $\theta = \delta/n^2$ :

$$R \geq 1 - \frac{2}{r+1} + \frac{1}{(r+1)^2} - \theta(1 + \ln(\delta - 1)).$$

The best known upper bound on the rate of LRC codes with locality  $r$ , availability  $t$ , and relative distance  $\theta$  [21] has the form

$$R \leq \frac{r-1}{r}(1 - \theta) - o(1).$$

The gap between this bound and the lower estimate implied by our construction is roughly  $O(\frac{1}{r} + \theta \ln \delta)$ .

**Example 3.** Let  $r_1 = 2$ ,  $r_2 = 6$ , and  $\delta = 9$ . Our construction gives rise to an LRC code with availability two, of length  $n^2 = 441$ , dimension  $k = 246$ , and distance  $d \geq 9$ , over a finite field  $\mathbb{F}_q$  such that  $21 \mid (q - 1)$  (for example, we can take  $q = 64$ ). To compare these codes with the product construction, let us choose the column codes and row codes of length 21 and distance 3, and let us take the maximum dimension of codes as given by the bound (III) for locality 2 and 6. This gives  $k_1 = 13, k_2 = 17$  and the overall dimension  $k = 221$ , lower than above.

An extension of the construction in this section to the case of  $t \geq 2$  can be easily obtained via  $t$ -dimensional cyclic codes and the general hyperbolic bound. Furthermore, using procedures similar to those used in (9) and (III), our construction can be generalized to  $h \geq 2$  levels of hierarchy such that the local codes in each of the  $h$  levels have availability  $t \geq 2$ .

#### APPENDIX A PROOF OF PROPOSITION 5

Let us first prove a technical claim.

**Claim 15.** For  $1 \leq i \leq h+1$ , we have  $\delta_i = (\delta_i - b_0^{(i-1)}) \bmod n_1$ .

*Proof:* We prove the claim by induction. Clearly, for  $i = 1$  we have  $\delta_1 = (\delta_1 - b_0^{(0)}) \bmod n_1$ . Next suppose that for a

given  $i$ ,  $1 \leq i < h+1$  we have  $\delta_i = (\delta_i - b_0^{(i-1)}) \bmod n_1$ . Reducing (II3) modulo  $n_1$ , we obtain the equality  $(\delta_i - b_0^{(i-1)}) = (\delta_{i+1} - b_0^{(i)}) \bmod n_1$ . Thus also  $\delta_i = (\delta_{i+1} - b_0^{(i)}) \bmod n_1$ , and this completes the proof. ■

Now we are ready to prove Proposition 5. Clearly, for  $i = 1$  we have  $|\mathcal{Z}_1| = \delta_1 - 1 = n_1 - r_1$ . Suppose for  $1 \leq i' \leq i$  we have  $|\mathcal{Z}_{i'}| = n_{i'} - r_{i'}$ , where  $1 \leq i < h+1$ . Let us establish the induction step.

By the definition of  $\mathcal{Z}_{i+1}$  and (II3), we have  $|\mathcal{Z}_{i+1}| \geq (\nu_i - a_i)n_i$ . Consider the set

$$\begin{aligned} \mathcal{A}_{i+1} &= (\nu_i - a_i)n_i + \{\delta_i - b_0^{(i-1)} - \delta_1 + 1, \\ &\quad \delta_i - b_0^{(i-1)} - \delta_1 + 2, \dots, n_i\} \end{aligned}$$

formed by adding  $(\nu_i - a_i)n_i$  to every element of the subset above, and let  $\mathcal{B}_{i+1} = \mathcal{A}_{i+1} \cap \mathcal{D}_{i+1}$ . Since  $\bigcup_{l=0}^{\nu_i - a_i - 1} (\mathcal{Z}_i + l n_i) \subseteq \mathcal{D}_{i+1}$ , we have

$$\begin{aligned} |\mathcal{Z}_{i+1}| &= (\nu_i - a_i)n_i + |\mathcal{B}_{i+1} \setminus (\mathcal{Z}_i + (\nu_i - a_i)n_i)| \\ &\quad + \left| \bigcup_{l=\nu_i - a_i}^{\nu_i - 1} (\mathcal{Z}_i + l n_i) \right|. \end{aligned}$$

Next, we would like to determine the cardinality of the set  $\mathcal{B}_{i+1} \setminus (\mathcal{Z}_i + (\nu_i - a_i)n_i)$ . By Claim 15, we have  $\delta_1 = (\delta_1 - b_0^{(i-1)}) \bmod n_1$ . Therefore,  $|\mathcal{A}_{i+1}|$  is divisible by  $n_1$ . Note that  $\mathcal{B}_{i+1} \subseteq \mathcal{A}_{i+1}$ . Moreover, among the first  $u_{i-1}^{(i)} n_{i-1}$  elements of  $\mathcal{A}_{i+1}$  there are  $u_{i-1}^{(i)} r_{i-1} - b_0^{(i-1)}$  elements that are in  $\mathcal{B}_{i+1}$  but not in  $\mathcal{Z}_i + (\nu_i - a_i)n_i$ . For the next  $u_{i-2}^{(i)} n_{i-2}$  elements in  $\mathcal{A}_{i+1}$  there are  $u_{i-2}^{(i)} r_{i-2}$  elements that are in  $\mathcal{B}_{i+1}$  but not in  $\mathcal{Z}_i + (\nu_i - a_i)n_i$ , and so forth. Hence, we have

$$\begin{aligned} &|\mathcal{B}_{i+1} \setminus (\mathcal{Z}_i + (\nu_i - a_i)n_i)| \\ &= \sum_{j=1}^{i-1} u_j^{(i)} r_j + u_0^{(i)} r_1 + b_0^{(i)} - b_0^{(i-1)} \\ &= b_i. \end{aligned}$$

It follows that

$$\begin{aligned} |\mathcal{Z}_{i+1}| &= (\nu_i - a_i)n_i + b_i + \left| \bigcup_{l=\nu_i - a_i}^{\nu_i - 1} (\mathcal{Z}_i + l n_i) \right| \\ &= (\nu_i - a_i)n_i + b_i + a_i |\mathcal{Z}_i| \\ &= n_{i+1} - a_i n_i + b_i + a_i (n_i - r_i) \\ &= n_{i+1} - (a_i r_i - b_i) \\ &= n_{i+1} - r_{i+1}, \end{aligned} \quad (41)$$

where (41) uses  $|\mathcal{Z}_i| = n_i - r_i$ , which is the induction hypothesis. This completes the proof.

#### APPENDIX B PROOF OF LEMMA 6

We again argue by induction on  $i$ . For  $i = 1$ , by (II3) we have

$$\begin{aligned} \delta_2 &= (\nu_1 - a_1)n_1 + \delta_1 + u_0^{(1)} n_1 + b_0^{(1)} - b_0^{(0)} \\ &= n_2 - a_1 n_1 + \delta_1 + u_0^{(1)} (n_1 - r_1) + b_1 \\ &= n_2 - a_1 (r_1 + \delta_1 - \delta_0) + \delta_1 + u_0^{(1)} (n_1 - r_1) + b_1 \end{aligned} \quad (42)$$

$$\begin{aligned}
&= n_2 - r_2 + \delta_1 - a_1(\delta_1 - \delta_0) + u_0^{(1)}(\delta_1 - \delta_0) \quad (43) \\
&= n_2 - r_2 + \delta_1 - (a_1 - u_0^{(1)})(\delta_1 - \delta_0) \\
&= n_2 - r_2 + \delta_1 - \left\lceil \frac{r_2}{r_1} \right\rceil (\delta_1 - \delta_0),
\end{aligned}$$

where (42) follows from  $n_2 = n_1\nu_1$  and (12), in (43) we used  $a_1r_1 = r_2 + b_1$ , and the last equality follows from  $a_1 = \lceil r_2/r_1 \rceil$  and  $u_0^{(1)} = \lfloor (b_1^{(1)} + b_0^{(0)})/r_1 \rfloor = 0$ .

For the induction step, let us fix  $i, 1 \leq i < h$  and suppose that

$$\delta_{i+1} = n_{i+1} - r_{i+1} + \delta_i - \sum_{l=1}^i \left\lceil \frac{r_{i+1}}{r_l} \right\rceil (\delta_l - \delta_{l-1}), \quad (44)$$

provided that conditions (14) are satisfied. Observe that

$$\begin{aligned}
\delta_{i+2} &= (\nu_{i+1} - a_{i+1})n_{i+1} + \delta_{i+1} \\
&\quad + \sum_{j=1}^i u_j^{(i+1)}n_j + u_0^{(i+1)}n_1 + b_0^{(i+1)} - b_0^{(i)} \\
&= n_{i+2} - a_{i+1}n_{i+1} + \delta_{i+1} \\
&\quad + \sum_{j=1}^i u_j^{(i+1)}(n_j - r_j) + u_0^{(i+1)}(n_1 - r_1) + b_{i+1}
\end{aligned} \quad (45)$$

Substituting  $n_{i+1}$  from (44), we obtain

$$\begin{aligned}
a_{i+1}n_{i+1} &= a_{i+1} \left( r_{i+1} + \delta_{i+1} - \delta_i \right. \\
&\quad \left. + \sum_{l=1}^i \left\lceil \frac{r_{i+1}}{r_l} \right\rceil (\delta_l - \delta_{l-1}) \right). \quad (46)
\end{aligned}$$

In addition, also by the induction hypothesis, we have

$$\begin{aligned}
u_0^{(i+1)}(n_1 - r_1) &= u_0^{(i+1)}(\delta_1 - \delta_0), \\
u_j^{(i+1)}(n_j - r_j) &= u_j^{(i+1)}(\delta_j - \delta_{j-1}) \\
&\quad + \sum_{l=1}^{j-1} u_j^{(i+1)} \left\lceil \frac{r_j}{r_l} \right\rceil (\delta_l - \delta_{l-1}), \\
&\quad 1 \leq j \leq i.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\sum_{j=1}^i u_j^{(i+1)}(n_j - r_j) + u_0^{(i+1)}(n_1 - r_1) \\
&= \sum_{l=1}^i \left( u_l^{(i+1)} + \sum_{j=l+1}^i u_j^{(i+1)} \left\lceil \frac{r_j}{r_l} \right\rceil \right) (\delta_l - \delta_{l-1}) \\
&\quad + u_0^{(i+1)}(\delta_1 - \delta_0). \quad (47)
\end{aligned}$$

Substituting (46) and (47) into (45), we obtain

$$\begin{aligned}
\delta_{i+2} &= n_{i+2} - r_{i+2} + \delta_{i+1} - a_{i+1}(\delta_{i+1} - \delta_i) \\
&\quad + \sum_{l=2}^i \left( u_l^{(i+1)} + \sum_{j=l+1}^i u_j^{(i+1)} \left\lceil \frac{r_j}{r_l} \right\rceil \right. \\
&\quad \left. - a_{i+1} \left\lceil \frac{r_{i+1}}{r_l} \right\rceil \right) (\delta_l - \delta_{l-1}) \\
&\quad + \left( u_0^{(i+1)} + u_1^{(i+1)} + \sum_{j=2}^i u_j^{(i+1)} \left\lceil \frac{r_j}{r_1} \right\rceil \right)
\end{aligned}$$

$$- a_{i+1} \left\lceil \frac{r_{i+1}}{r_1} \right\rceil (\delta_1 - \delta_0).$$

Thus, if the corresponding conditions of (14) are satisfied, then we have

$$\delta_{i+2} = n_{i+2} - r_{i+2} + \delta_{i+1} - \sum_{l=1}^{i+1} \left\lceil \frac{r_{i+2}}{r_l} \right\rceil (\delta_{i+1} - \delta_i).$$

This completes the induction step.

## APPENDIX C PROOF OF PROPOSITION 10

Part (a): The generator matrix of the truncated code  $\mathcal{C}_{[0,j]}$  is given in (24), where  $\text{rank}(G_0) = k$ .

Since  $G_0$  has full rank, by Definition 9, the  $j$ -th column distance of  $\mathcal{C}$  is equal to

$$d_j^c = \min\{\text{wt}(u_{[0,j]}G_j^c) \mid u_0 \neq 0\},$$

where  $u_{[0,j]} = (u_0, \dots, u_j) \in \mathbb{F}_q^{k(j+1)}$  is an input sequence truncated at the  $j$ -th time instant. Obviously,

$$d_j^c \leq \min\{\text{wt}(u_{[0,j]}G_j^c) \mid u_0 \neq 0, u_1 = u_2 = \dots = u_j = 0\}. \quad (48)$$

Consider the  $k \times n(j+1)$  submatrix  $[G_0, G_1, \dots, G_j]$  that forms the first  $k$  rows of  $G_j^c$ . By elementary row operations it is possible to make some, say first,  $k$  columns of  $G_1$  be all-zero, and the same is true for some  $k$  columns of the matrices  $G_i, i = 2, \dots, j$ . (Note that to accomplish this, we use all the rows of  $G_j^c$  and not just the rows of the submatrix  $[G_0, G_1, \dots, G_j]$ .) As a result, the vector  $b(u_0) := u_0[G_0, G_1, \dots, G_j]$  will contain at least  $kj$  zero coordinates for any choice of  $u_0 \in \mathbb{F}_q^k$ , and so the effective length of the set of vectors  $\{b(u_0)\}$  is  $n(j+1) - kj$ . The set of vectors  $\{b(u_0) \mid u_0 \in \mathbb{F}_q^k\}$  is a subset of the code  $\mathcal{C}_{[0,j]}$  and it forms a linear code  $\mathcal{K}$  of effective length  $n(j+1) - kj$  with  $(r, \delta)$  locality if the original convolutional code  $\mathcal{C}$  has (column or row)  $(r, \delta)$  locality. The distance of the code  $\mathcal{K}$  gives an upper bound on  $d_j^c$ , and is itself bounded above as in (8). Substituting the parameters of the code  $\mathcal{K}$ , we obtain the bound (27) for the distance  $d_j^c$ . Part (a) is proved.

*Remark:* Without the locality assumption the above argument proves the Singleton bound (29) for the column distance of the code  $\mathcal{C}$ .

Part (b): Recall the following observation (e.g., [10]):

**Proposition 16.** Let  $j \geq 0$ . Let  $H_j^c$  be the parity check matrix for the truncated convolutional code  $\mathcal{C}_{[0,j]}$ . Then the following properties are equivalent:

- 1)  $d_j^c = d$ ;
- 2) none of the first  $n$  columns of  $H_j^c$  is contained in the span of any other  $d-2$  columns and one of the first  $n$  columns of  $H_j^c$  is in the span of some other  $d-1$  columns of  $H_j^c$ .

Now suppose that  $d_j^c$  attains (27) with equality while  $d_{j-1}^c < (n-k)j + \delta - \lceil \frac{k}{r} \rceil (\delta - 1)$ . By Proposition 16, there exists a column among the first  $n$  columns of  $H_{j-1}^c$  such that it is in

the span of some other  $d_{j-1}^c - 1$  columns of  $H_{j-1}^c$ . Note that  $H_{j-1}^c$  is a submatrix of  $H_j^c$ . Specifically, we have

$$H_j^c = \begin{pmatrix} & & & 0 \\ & H_{j-1}^c & & \vdots \\ & & & 0 \\ H_j & H_{j-1} & \cdots & H_1 & H_0 \end{pmatrix},$$

where  $H_i, 1 \leq i \leq j$  are  $(n-k) \times n$  matrices and  $\text{rank}(H_0) = n-k$ . Because of this, there exists a column among the first  $n$  columns of  $H_j^c$  such that it is in the span of some other  $d_{j-1}^c - 1 + n - k < d_j^c - 1$  columns of  $H_j^c$ , which by Proposition 16 contradicts our assumption about  $d_j^c$ . Hence, it follows that the optimality of the  $j$ -th column distance implies the optimality of the  $i$ -th column distance for all  $i \leq j$  for convolutional codes with (column or row) locality.

## REFERENCES

- [1] S. Ballentine, A. Barg, and S. Vlăduț, "Codes with hierarchical locality from covering maps of curves," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6056–6071, 2019.
- [2] A. Barg, K. Haymaker, E. Howe, G. Matthews, and A. Várilly-Alvarado, "Locally recoverable codes from algebraic curves and surfaces," in *Algebraic Geometry for Coding Theory and Cryptography*, E. Howe, K. Lauter, and J. Walker, Eds. Springer, 2017, pp. 95–126.
- [3] A. Barg, I. Tamo, and S. Vlăduț, "Locally recoverable codes on algebraic curves," *IEEE Trans. Inform. Theory*, vol. 63, no. 8, pp. 4928–4939, 2017.
- [4] A. Beemer, R. Coatney, V. Guruswami, H. H. Lopez, and F. Pinero, "Explicit optimal-length locally repairable codes of distance 5," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018, pp. 800–804.
- [5] E. Blokh and V. Zyablov, "Coding of generalized cascade codes," *Probl. Inform. Trans.*, vol. 10, no. 3, pp. 218–222, 1974.
- [6] H. Cai, Y. Miao, M. Schwartz, and X. Tang, "On optimal locally repairable codes with super-linear length," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4853–4868, 2020.
- [7] B. Chen, S.-T. Xia, J. Hao, and F.-W. Fu, "Constructions of optimal  $(r, \delta)$  locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2499–2511, 2018.
- [8] A. Datta, "Locally repairable rapidRAID systematic codes—one simple convoluted way to get it all," in *2014 IEEE Information Theory Workshop (ITW 2014)*, 2014, pp. 60–64.
- [9] M. Esmaili, T. A. Gulliver, N. P. Secord, and S. A. Mahmoud, "A link between quasi-cyclic codes and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 1, pp. 431–435, 1998.
- [10] R. Freij-Hollanti, T. Westerbäck, and C. Hollanti, "Locally repairable codes with availability and hierarchy: Matroid theory via examples," in *24th International Zürich Seminar on Communications (IZS), Zurich, Switzerland, March 2-4, 2016*. ETH-Zürich, 2016, pp. 45–49, <https://doi.org/10.3929/ethz-a-010645448>.
- [11] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, "Strongly-MDS convolutional codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 584–598, 2006.
- [12] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, 2012.
- [13] M. Grezet and C. Hollanti, "The complete hierarchical locality of the punctured simplex code," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2833–2837.
- [14] V. Guruswami, C. Xing, and C. Yuan, "How long can optimal locally repairable codes be?" *IEEE Trans. Inf. Theory*, vol. 6, no. 6, pp. 3662–3670, 2019.
- [15] L. Holzbaur, R. Freij-Hollanti, and A. Wachter-Zeh, "Cyclic codes with locality and availability," 2018, arXiv preprint arXiv:1812.06897.
- [16] P. Huang, E. Yaakobi, and P. H. Siegel, "Multi-erasure locally recoverable codes over small fields: A tensor product approach," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 2609–2624, 2020.
- [17] F. Ivanov, A. Kreshchuk, and V. Zyablov, "On the local erasure correction capacity of convolutional codes," in *2018 International Symposium on Information Theory and Its Applications (ISITA), Singapore*, 2018, pp. 296–300.
- [18] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*, 2nd ed. Hoboken, NJ: J. Wiley & Sons, Inc., 2015.
- [19] J. Justesen, E. Paaske, and M. Ballan, "Quasi-cyclic unit memory convolutional codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 3, pp. 540–547, 1990.
- [20] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration and erasure correction," *IEEE Trans. Inform. Theory*, vol. 60, no. 8, pp. 4637–4660, 2014.
- [21] S. Kruglik, K. Nazirkhanova, and A. Frolov, "New bounds and generalizations of locally recoverable codes with availability," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4156–4166, 2019.
- [22] K. Lally and P. Fitzpatrick, "Algebraic structure of quasicyclic codes," *Discrete Applied Mathematics*, vol. 111, pp. 157–175, 2001.
- [23] X. Li, L. Ma, and C. Xing, "Construction of asymptotically good locally repairable codes via automorphism groups of function fields," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7087–7094, 2019.
- [24] —, "Optimal locally repairable codes via elliptic curves," *IEEE Trans. Inform. Theory*, vol. 65, no. 1, pp. 108–117, 2019.
- [25] S. Ling and P. Solé, "On the algebraic structure of quasi-cyclic codes, I," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2751–2760, 2001.
- [26] —, "On the algebraic structure of quasi-cyclic codes II: Chain rings," *Designs, Codes and Cryptography*, vol. 30, pp. 113–130, 2003.
- [27] Y. Luo, C. Xing, and C. Yuan, "Optimal locally repairable codes of distance 3 and 4 via cyclic codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1048–1053, 2018.
- [28] U. Martínez-Peñas and D. Napp, "Locally repairable convolutional codes with sliding window repair," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4935–4947, 2020.
- [29] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4481–4493, 2016.
- [30] K. Saints and C. Heegard, "On hyperbolic cascaded Reed-Solomon codes," in *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*. Springer, 1993, pp. 291–303.
- [31] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "Codes with hierarchical locality," in *2015 IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 1257–1261, expanded version in arXiv preprint arXiv:1501.06683.
- [32] G. Solomon and H. C. A. van Tilborg, "A connection between block and convolutional codes," *SIAM J. Appl. Math.*, vol. 37, no. 2, pp. 358–369, 1979.
- [33] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661–4676, 2014.
- [34] I. Tamo, A. Barg, and A. Frolov, "Bounds on the parameters of locally recoverable codes," *IEEE Trans. inf. theory*, vol. 62, no. 6, pp. 3070–3083, 2016.
- [35] I. Tamo, A. Barg, S. Goparaju, and R. Calderbank, "Cyclic LRC codes, binary LRC codes, and upper bounds on the distance of cyclic codes," *International Journal of Information and Coding Theory*, vol. 3, no. 4, pp. 345–364, 2016.
- [36] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, 2004.
- [37] V. Tomás, J. Rosenthal, and R. Smarandache, "Decoding of convolutional codes over the erasure channel," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 90–108, 2012.
- [38] A. Wang and Z. Zhang, "Repair locality with multiple erasure tolerance," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6979–6987, 2014.
- [39] S. Yang, A. Hareedy, R. Calderbank, and L. Dolecek, "Hierarchical coding to enable scalability and flexibility in heterogeneous cloud storage," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6, expanded version in arXiv:2009.09146.
- [40] B. Zhu, X. Li, H. Li, and K. W. Shum, "Replicated convolutional codes: A design framework for repair-efficient distributed storage codes," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 1018–1024.

**Zitan Chen** received the B.Eng. degree in information engineering from the Chinese University of Hong Kong, Hong Kong, in 2015. He is pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD, USA.

**Alexander Barg** is a professor in the Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, MD. He is broadly interested in information and coding theory, applied probability, and algebraic combinatorics, and has published about a hundred research papers. He received the 2015 Information Theory Society Paper Award, was a plenary speaker at the 2016 IEEE International Symposium on Information Theory (Barcelona, Spain), and in 2018-2019 served Editor-in-Chief of this TRANSACTIONS.