

Locally Repairable Convolutional Codes with Sliding Window Repair

Umberto Martínez-Peñas, *Member, IEEE*, and Diego Napp, ,

Abstract—Locally repairable convolutional codes (LRCCs) for distributed storage systems (DSSs) are introduced in this work. They enable local repair, for a single node erasure (or more generally, $\partial - 1$ erasures per local group), and sliding-window global repair, which can correct erasure patterns with up to $d_j^c - 1$ erasures in every window of $j + 1$ consecutive blocks of n nodes, where d_j^c is the j th column distance of the code. The parameter j can be adjusted, for a fixed LRCC, according to different catastrophic erasure patterns, requiring only to contact $n(j + 1) - d_j^c + 1$ nodes, plus less than μn other nodes, in the storage system, where μ is the memory of the code. A Singleton-type bound is provided for d_j^c . If it attains such a bound, an LRCC can correct the same number of catastrophic erasures in a window of length $n(j + 1)$ as an optimal locally repairable block code of the same rate and locality, and with block length $n(j + 1)$. In addition, the LRCC is able to perform the flexible and somehow local sliding-window repair by adjusting j . Furthermore, by adjusting and/or sliding the window, the LRCC can potentially correct more erasures in the original window of $n(j + 1)$ nodes than an optimal locally repairable block code of the same rate and locality, and length $n(j + 1)$. Finally, the concept of partial maximum distance profile (partial MDP) codes is introduced. Partial MDP codes can correct all information-theoretically correctable erasure patterns for a given locality, local distance and information rate. An explicit construction of partial MDP codes whose column distances attain the provided Singleton-type bound, up to certain parameter $j = L$, is obtained based on known maximum sum-rank distance convolutional codes.

Index Terms—Convolutional Codes, Distributed Storage, Locally Repairable Codes, Locally Repairable Convolutional Codes, Sliding-Window Repair, Sum-Rank Metric.

I. INTRODUCTION

LOCALLY repairable codes (LRCs) [11] are an important class of codes for Distributed Storage Systems (DSSs), since they allow to repair a single node by contacting and downloading the content of a small number (called *locality*) of other nodes (in contrast with MDS codes), while still being able to repair a large number of nodes in case of catastrophic erasures (in contrast with Cartesian products). LRCs are thus natural hybrids between MDS codes and Cartesian products of codes that enjoy both global and local erasure-correction capabilities simultaneously, given by global and local distances,

This work is supported by The Independent Research Fund Denmark (Grant No. DFF-7027-00053B), by the Generalitat Valenciana (Grant No. AICO/2017/128) and the Universitat d'Alacant (Grant No. VIGROB-287).

Parts of this paper have been presented at the 2019 IEEE International Symposium on Information Theory, Paris, France, July, 2019 [25].

U. Martínez-Peñas is with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada. D. Napp is with The Department of Mathematics, University of Alicante, Spain. (e-mail: umberto@ece.utoronto.ca; diego.napp@ua.es).

respectively. Note that *repair* is typically used interchangeably with *erasure correction* in the storage literature. We will use both terms throughout this work.

LRCs have already been implemented in practice (see [13], [33] for instance). *Optimal* LRCs (meaning LRCs attaining optimal global distance for a given locality, local distance and information rate) for general parameters and field sizes that are linear in the code length were first obtained in [34]. LRCs capable of correcting all information-theoretically correctable global erasure patterns, for a given locality, local distance and information rate, were introduced independently in [4], [10] (where they are called *partial MDS* and *maximally recoverable* LRCs, respectively). As expected, maximally recoverable LRCs also attain optimal global distance. However, they can correct strictly more global erasure patterns than general optimal LRCs (see Remark 3 in Section VI) for the same parameters. Constructions of maximally recoverable LRCs with relatively small field sizes for general parameters have been given in [8], [23] (see also the references therein).

On the other hand, it is shown in [35] that maximum distance profile (MDP) convolutional codes provide an interesting alternative to MDS block codes since they admit *sliding-window* erasure correction: They can correct any erasure pattern such that there are no more than $(n - k)(j + 1)$ erasures in any consecutive $j + 1$ blocks of n symbols (that is, $n(j + 1)$ consecutive symbols), where k/n is the rate of the code (see [35, Th. 3.1] or Fig. 1). Recall that an (n, k) MDS code has rate k/n and can correct up to $n - k$ arbitrary erasures in the block of n symbols. Furthermore, erasure correction with an MDP convolutional code is performed somehow locally by sliding recursively the window of $j + 1$ blocks, and the parameter j may vary arbitrarily up to a certain constant L determined by the degree (thus memory) of the convolutional code (see (2)). Therefore MDP convolutional codes already enable certain local and flexible repair, since the window size $n(j + 1)$ can be chosen according to how catastrophic the erasure pattern is. Moreover, by adjusting and/or sliding a window of $j + 1$ blocks (see Fig. 2), an MDP code can potentially correct in a window of size $n(j + 1)$ more erasures than an MDS block code of the same rate and of block length $n(j + 1)$ (i.e., an $(n(j + 1), k(j + 1))$ MDS code). Unfortunately, in case of one single node erasure (most common case), sliding-window repair with $j = 0$ still requires contacting and downloading the content of μn extra symbols, where μ is the memory of the code, due to its convolutional nature.

Motivation

The previous paragraphs describe the advantages of optimal block LRCs and MDP convolutional codes over MDS block codes. Motivated by that discussion, we introduce in this work *locally repairable convolutional codes* (LRCCs). See Section III for the formal definition of LRCCs and for considerations on locality for convolutional codes. The main advantages of optimal LRCCs (with optimal column distances, i.e., attaining (4)) over MDP convolutional codes and over optimal block LRCs are, respectively, as follows:

- 1) In contrast with MDP convolutional codes, optimal LRCCs can repair a single node (or more generally, $\partial-1$ erasures per local group) by contacting $r < n$ other nodes (instead of contacting μn nodes, as is the case for MDP convolutional codes). Simultaneously, they enable sliding-window repair (see Fig. 4), which can be set up flexibly according to different catastrophic erasure patterns (see Figs. 4, 5 and 6), which is the main advantage of MDP convolutional codes over MDS block codes.
- 2) Analogously to the advantage of MDP convolutional codes over MDS block codes, optimal LRCCs can potentially correct in a window of size $n(j+1)$ more erasures than an optimal block LRC of the same rate and locality, and of block length $n(j+1)$ (see Example 1 and Fig. 5), thanks to adjusting the sliding-window size. In addition, optimal LRCCs may adjust the sliding-window size in order to contact less nodes when intermediate erasure patterns occur (see Example 2). Hierarchical LRCs [32] also require contacting less nodes for intermediate erasure patterns, however their global distance is strictly lower than that of optimal LRCs for a given block length (see [32, Th. 2.1]), in contrast with optimal LRCCs. Furthermore, LRCCs also enable encoding and storing an unrestricted sequence of files, while locality remains constant and encoding and sliding-window repair complexities are all bounded (by the memory of the code). Note that LRCCs can easily be turned into block codes by converting them into tail-biting convolutional codes, while the properties described above still hold.

We will illustrate the main advantages of LRCCs over block LRCs with Examples 1 and 2, which are placed in Section IV since we first need to introduce the main definitions and notation in Sections II and III.

Main Contributions

Our main contributions are the following. We define LRCCs (Definition 10) and provide a Singleton-type bound on their column distances (Theorem 2), which measure the global sliding window repair capability of the code. We later define partial MDP codes (Definition 15), which can correct all information-theoretically correctable erasure patterns for the given local constraints and, in particular, attain the previous bound for as long as possible. We provide in Construction 1 a method for finding partial MDP codes based on outer MSRDC convolutional codes (Theorem 4). By plugging in Construction

1 the MSRDC convolutional codes from [21], we obtain an explicit family of partial MDP codes (Corollary 2) for general parameters. Their main disadvantage is their big global field size, although local fields are small. However, this is only an issue in terms of computational complexity, since nodes in DSSs typically store large amounts of data. Furthermore, our construction gives some field size to guarantee the existence of partial MDP codes, but constructions over smaller fields may be possible.

To conclude this subsection, we note that the use of streaming or convolutional codes for storage or as LRCs is not new. Binary tail-biting convolutional codes were proposed as LRCs in [7], [38], but sliding-window repair was not considered. Locality properties of more general (but still binary) convolutional codes were recently considered in [14]. However, LRCCs and sliding-window repair as considered in this work were not treated in [14]. Rateless streaming codes (e.g. Fountain codes [5]) are an interesting alternative to MDS block codes for global repair in DSSs (see [20, Ch. 50]), since they generally achieve low redundancy and enable global erasure correction with complexity of $\mathcal{O}(k \log(k))$ XOR operations (products in \mathbb{F}_2) or even less, for k encoded symbols. Locally repairable Fountain codes were proposed in [3]. However, their locality is of order $\log(k)$ (unbounded), for k encoded symbols, and they do not enable sliding-window global repair.

Organization

The remainder of the paper is organized as follows. In Section II, we collect some preliminaries on convolutional codes. In Section III, we introduce LRCCs and give a Singleton-type bound on their column distances, which determine the sliding-window erasure-correction capability of LRCCs. In Section IV, we provide two motivating concrete examples showing the advantages of optimal LRCCs over optimal block LRCs. In Section V, we show how to obtain LRCCs with arbitrary and small-field local codes and optimal global column distances (in view of the previous bound) based on codes with optimal column sum-rank distances [21]. In Section VI, we introduce *partial MDP* convolutional codes, whose sliding windows can correct analogous erasure patterns as partial MDS block codes [4], [10]. We also provide concrete constructions of partial MDP convolutional codes based on the codes in [21]. Finally, in Section VII, we discuss extending our work to considering LRCCs with unequal localities and local distances, and how to turn our LRCCs to tail-biting convolutional codes.

II. PRELIMINARIES ON CONVOLUTIONAL CODES

In this section, we collect general definitions and results on convolutional codes that we will use throughout the paper.

Let \mathbb{F} be a finite field, and denote by $\mathbb{F}[D]$ the ring of polynomials with coefficients in \mathbb{F} . Fix a positive integer $n \in \mathbb{N}$. We will typically consider and graphically represent a word in $\mathbb{F}[D]^n$ as an unrestricted sequence of vectors of length n , $v(D) = \sum_{j \in \mathbb{N}} v_j D^j \equiv (v_0, v_1, v_2, \dots) \in (\mathbb{F}^n)^\mathbb{N}$, where we use the following terminology. A *block* is each of the n consecutive coordinates in $(\mathbb{F}^n)^\mathbb{N}$ that support each

vector v_0, v_1, \dots , being *the j th block* the block containing the coordinates supporting v_j , for $j \in \mathbb{N}$. A *symbol* is each component of the vectors v_0, v_1, \dots , thus it is an element of \mathbb{F} . Finally, a *node* is the abstraction of the storage device that stores a given symbol. Hence, in this work, each block corresponds to n nodes storing n symbols over \mathbb{F} .

A. Degree and Memory

Recall that, since $\mathbb{F}[D]$ is a principal ideal domain, every $\mathbb{F}[D]$ -submodule of $\mathbb{F}[D]^n$ is free.

Definition 1. An (n, k) convolutional code is a (free) $\mathbb{F}[D]$ -submodule $\mathcal{C} \subseteq \mathbb{F}[D]^n$ of rank k . A *generator matrix* of the code is a full-rank matrix $G(D) \in \mathbb{F}[D]^{k \times n}$ such that

$$\mathcal{C} = \{u(D)G(D) \mid u(D) \in \mathbb{F}[D]^k\}.$$

For a vector $v(D) \in \mathbb{F}[D]^n$, we define its degree as the maximum degree of its components, which are polynomials in $\mathbb{F}[D]$. We say that a generator matrix $G(D)$ of \mathcal{C} is *reduced* if the sum of the row degrees of $G(D)$ is minimum among generator matrices of \mathcal{C} , where by row degrees we mean the degrees of the rows in $G(D)$.

It follows from Theorem A-2, Item 3, in [26] that if $e_1 \leq e_2 \leq \dots \leq e_k$ and $f_1 \leq f_2 \leq \dots \leq f_k$ are the row degrees of a reduced generator matrix $G(D) \in \mathbb{F}[D]^{k \times n}$ and some other generator matrix $\tilde{G}(D) \in \mathbb{F}[D]^{k \times n}$, respectively, of \mathcal{C} , then $e_i \leq f_i$, for $i = 1, 2, \dots, k$. In particular, the set of degrees $\{e_1, e_2, \dots, e_k\}$ of one, thus any, reduced generator matrix is an invariant of the convolutional code \mathcal{C} . Hence the following definition is consistent.

Definition 2. Given an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$, let e_1, e_2, \dots, e_k be the row degrees of one, thus any, of its reduced generator matrices. We define the *degree* and *memory* of \mathcal{C} , respectively, as

$$\delta = \delta(\mathcal{C}) = \sum_{i=1}^k e_i \quad \text{and} \quad \mu = \mu(\mathcal{C}) = \max\{e_1, e_2, \dots, e_k\}.$$

Note that convolutional codes with zero memory (thus zero degree) coincide with (potentially infinite) Cartesian products of a single (n, k) block code $\mathcal{C} \subseteq \mathbb{F}^n$.

B. Non-Catastrophic Codes and Parity-Check Matrices

In most results in this work, although not all, we will require convolutional codes to be *non-catastrophic* or *observable*, which we now define in terms of basic generator matrices.

Definition 3. Given an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$, we say that a generator matrix $G(D)$ of \mathcal{C} is basic if it has a polynomial right inverse, that is, if there exists $F(D) \in \mathbb{F}[D]^{n \times k}$ such that $G(D)F(D) = I_k$. We say that \mathcal{C} is non-catastrophic if it admits a generator matrix that is reduced and basic.

Observe that any reduced and basic generator matrix $G(D) = \sum_{j=0}^{\mu} G_j D^j$ of a convolutional code satisfies that $G_0 \in \mathbb{F}^{k \times n}$ is full-rank. For many results in this work, we will only need this weaker property.

Using Theorem A-1, Item 5, in [26], and using the vector space over $\mathbb{F}(D)$ (the field of fractions of $\mathbb{F}[D]$) generated by a non-catastrophic convolutional code, it is easy to see that it admits a polynomial parity-check matrix. This strong property of non-catastrophic codes is what we will need for sliding-window repair, as described in Subsection II-D.

Lemma 1. If $\mathcal{C} \subseteq \mathbb{F}[D]^n$ is a non-catastrophic (n, k) convolutional code, then there exists a full-rank matrix $H(D) \in \mathbb{F}[D]^{(n-k) \times n}$ such that

$$\mathcal{C} = \{v(D) \in \mathbb{F}[D]^n \mid v(D)H(D)^T = 0\}.$$

We call $H(D)$ a (polynomial) parity-check matrix of \mathcal{C} .

C. Free and Column Distances

We now recall the main notions of minimum distance of convolutional codes. Given $v(D) = \sum_{j \in \mathbb{N}} v_j D^j \in \mathbb{F}[D]^n$, we define its *Hamming weight* as

$$\text{wt}(v(D)) = \sum_{j \in \mathbb{N}} \text{wt}(v_j).$$

The free distance, which we now define, gives the correction capability of a convolutional code when considering whole codewords. In other words, there is no maximum degree j for a codeword considered by the free distance.

Definition 4. Given an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$, we define its *free distance* as

$$d(\mathcal{C}) = \min\{\text{wt}(v(D)) \mid v(D) \in \mathcal{C} \text{ and } v(D) \neq 0\}.$$

We next define column distances, which give the sliding-window correction capability of a non-catastrophic convolutional code (see the next subsection). This will be the type of distance that we will be interested in for global repair in our locally repairable codes.

Definition 5. Given an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$, with memory μ and reduced generator matrix $G(D) = \sum_{h=0}^{\mu} G_h D^h$, define the *j th truncated sliding generator matrix* $G_j^c \in \mathbb{F}[D]^{(j+1)k \times (j+1)n}$ as

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

for $j \in \mathbb{N}$, where $G_h = 0$ if $h > \mu$. Define now the *j th column block code* of \mathcal{C} as

$$\begin{aligned} \mathcal{C}_j^c &= \{(u_0, u_1, \dots, u_j)G_j^c \mid u_0, u_1, \dots, u_j \in \mathbb{F}^k, u_0 \neq 0\} \\ &\stackrel{*}{=} \left\{ (v_0, v_1, \dots, v_j) \mid \sum_{h \in \mathbb{N}} v_h D^h \in \mathcal{C}, v_0 \neq 0 \right\} \subseteq \mathbb{F}^{(j+1)n}, \end{aligned}$$

where the equality $*$ holds if G_0 is full-rank. Finally, define the *j th column distance* of \mathcal{C} as

$$d_j^c(\mathcal{C}) = d(\mathcal{C}_j^c),$$

for $j \in \mathbb{N}$.

The column distances satisfy the following Singleton bound, which was proven in [9, Prop. 2.2].

Proposition 1 ([9]). For an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$ with a generator matrix $G(D) = \sum_{j=0}^{\mu} G_j D^j$ (possibly not reduced) such that G_0 is full-rank, and for $j \in \mathbb{N}$, it holds that

$$d_j^c(\mathcal{C}) \leq (n - k)(j + 1) + 1. \quad (1)$$

Items 1 and 2 in the following proposition follow from [9, Cor. 2.3] and [31, Th. 2.2], respectively.

Proposition 2 ([9], [31]). Given a non-catastrophic (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$ of degree δ , the following hold:

- 1) If $d_{j+1}^c(\mathcal{C}) = (n - k)(j + 2) + 1$, then $d_j^c(\mathcal{C}) = (n - k)(j + 1) + 1$.
- 2) If $d_j^c(\mathcal{C}) = (n - k)(j + 1) + 1$, then

$$j \leq L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n - k} \right\rfloor. \quad (2)$$

The previous proposition motivates the following definition.

Definition 6 ([9]). We say that an (n, k) convolutional code \mathcal{C} is j -MDS if it is non-catastrophic and $d_j^c(\mathcal{C}) = (n - k)(j + 1) + 1$. We say that \mathcal{C} is *maximum distance profile (MDP)* if it is non-catastrophic and $d_L^c(\mathcal{C}) = (n - k)(L + 1) + 1$, where L is as in (2).

D. Sliding-Window (Global) Repair

As shown in [35, Th. 3.1] a non-catastrophic (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$ may correct any erasure pattern if at most $d_j^c(\mathcal{C}) - 1$ erasures occur in any tuple $(v_t, v_{t+1}, \dots, v_{t+j}) \in \mathbb{F}^{n(j+1)}$, for $j \in \mathbb{N}$. Furthermore, it may do so recursively by sliding a window that only involves the symbols in $v_{t-\mu}, v_{t-\mu+1}, \dots, v_{t+j}$, where $\mu = \mu(\mathcal{C})$. The formal statement is as follows. See also Figs. 1 and 2 for a graphical description.

For convenience, we first define erasures formally.

Definition 7. Let \star be a symbol representing an erasure and not belonging to any finite field, and denote $\tilde{\mathbb{F}} = \mathbb{F} \cup \{\star\}$. Given $N \in \mathbb{N}$ and $v \in \mathbb{F}^N$, we say that $v^* \in \tilde{\mathbb{F}}^N$ is the vector v with e erasures, where $0 \leq e \leq N$, if e components of v^* are the symbol \star , and v and v^* coincide in the other $N - e$ components.

We now restate [35, Th. 3.1] and part of its proof.

Theorem 1 ([35]). Let $\mathcal{C} \subseteq \mathbb{F}[D]^n$ be a non-catastrophic (n, k) convolutional code with memory μ , and fix $j \in \mathbb{N}$. Let $v(D) = \sum_{h \in \mathbb{N}} v_h D^h \in \mathcal{C}$ and let $v_0^*, v_1^*, v_2^*, \dots \in \tilde{\mathbb{F}}^n$ be such that $(v_t^*, v_{t+1}^*, \dots, v_{t+j}^*) \in \tilde{\mathbb{F}}^{n(j+1)}$ is the vector $(v_t, v_{t+1}, \dots, v_{t+j}) \in \mathbb{F}^{n(j+1)}$ with at most $d_j^c(\mathcal{C}) - 1$ erasures, for all $t \in \mathbb{N}$. Then, for each $t = 0, 1, 2, \dots$, the vector $v_t \in \mathbb{F}^n$ can be recursively and uniquely recovered from the tuple

$$(v_{t-\mu}, v_{t-\mu+1}, \dots, v_{t-1}, v_t^*, v_{t+1}^*, \dots, v_{t+j}^*) \in \tilde{\mathbb{F}}^{n(\mu+j+1)} \quad (3)$$

by solving a system of non-homogeneous equations, whose coefficients are given by a parity-check matrix of \mathcal{C} (Lemma 1), the symbols in $v_{t-\mu}, v_{t-\mu+1}, \dots, v_{t-1}$, and the symbols such

that $v_{u,i}^* = v_{u,i}$, for $u = t, t+1, \dots, t+j$, and whose unknowns are $x_{u,i}$, for i such that $v_{u,i}^* = \star$, for $u = t, t+1, \dots, t+j$. To recover v_{t+1} , we “slide” the window (3) one position to the right (see Figs. 1 and 2).

In the previous theorem, we implicitly assume that $v_j = 0$ for all $j = -1, -2, \dots, -\mu$.

This type of erasure correction may already be considered as local repair, since j may be small. Furthermore, the window size is not necessarily restricted, since j may be arbitrary. However, setting $j = 0$, we see that correcting one erasure in a single block $v_t \in \mathbb{F}^n$ requires contacting another μn nodes and downloading their symbols, corresponding to $(v_{t-\mu}, v_{t-\mu+1}, \dots, v_{t-1}) \in \mathbb{F}^{\mu n}$, in order to set up the necessary system of linear equations. Thus, although sliding-window repair enjoys certain local nature, it admits considerable room for improvement. Adding locality inside each block v_t optimally will be our main objective in the rest of the paper.

In addition, if a convolutional code has optimal column distances as in (1) for a given j (which can be chosen as $j = L$ for MDP codes as in Definition 6), then it may correct strictly more erasures than an MDS block of the same rate for a block of $n(j + 1)$ symbols, as shown in Fig. 2. Note that this superior correction capability is not hurt globally by the fact that we need the previous μ blocks to be correct, since in the first window we have that $v_{-1} = v_{-2} = \dots = v_{-\mu} = 0$, hence these first μ blocks are always assumed to be correct.

III. LOCALITY IN CONVOLUTIONAL CODES

In this section, we formulate locality for convolutional codes. For this purpose, we define the following two types of restrictions for a convolutional code. The first type consists in considering one generic block $v_j \in \mathbb{F}^n$ for arbitrary codewords $v(D)$ in the convolutional code.

Definition 8. Given an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$ with reduced generator matrix $G(D) = \sum_{j=0}^{\mu} G_j D^j$, where μ is the memory of the code, we define its *associated block code* as

$$\begin{aligned} \mathcal{C}^0 &= \left\{ \sum_{j=0}^{\mu} u_j G_j \mid u_j \in \mathbb{F}^k, j = 0, 1, \dots, \mu \right\} \\ &= \left\{ v_\mu \in \mathbb{F}^n \mid v(D) = \sum_{j \in \mathbb{N}} v_j D^j \in \mathcal{C} \right\} \subseteq \mathbb{F}^n. \end{aligned}$$

Note that by the second equality, the definition of \mathcal{C}^0 does not depend on the generator matrix of \mathcal{C} . We now give the second type of restriction, which consists in restricting each block of the convolutional code to some subset of coordinates $\Gamma \subseteq [n]$. Here, we use the notation $[n] = \{1, 2, \dots, n\}$.

Definition 9. Given an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$ and given a non-empty subset $\Gamma \subseteq [n]$, we define the *restriction* of \mathcal{C} to Γ as the convolutional code

$$\mathcal{C}_\Gamma = \{v(D)_\Gamma \mid v(D) \in \mathcal{C}\} \subseteq \mathbb{F}[D]^{|\Gamma|}.$$

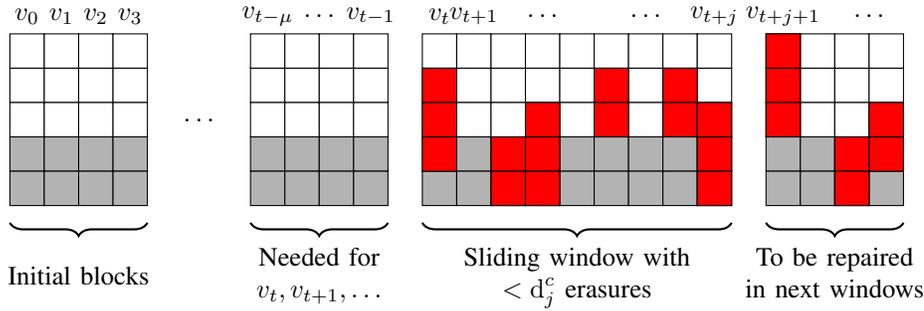


Fig. 1. Sliding-window erasure correction as described in Theorem 1 for a $(5, 3)$ convolutional code. Here, parity-check symbols are depicted in grey, and erasures are depicted in red.

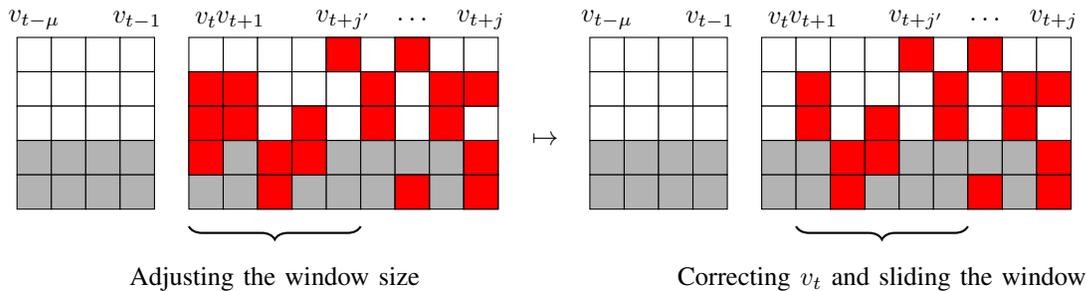


Fig. 2. An example of a $(5, 3)$ convolutional code, say non-catastrophic. If the j th column distance of such a code attains (1) , for $j = 8$, then it can correct erasure patterns with up to $(n - k)(j + 1) = 18$ erasures per window of $j + 1 = 9$ consecutive blocks of $n = 5$ coordinates. The window consisting of the $j + 1 = 9$ blocks $v_t, v_{t+1}, \dots, v_{t+j}$ contains however 19 erasures. Hence it cannot be corrected by a convolutional code with optimal j th column distance, by considering that window, nor by an $(n(j + 1), k(j + 1)) = (45, 27)$ MDS block code for such a window. However, the convolutional code also has optimal j' th column distance for $j' = 4$ (Proposition 2), thus it can correct erasure patterns with up to 10 erasures per window of $j' + 1 = 5$ consecutive blocks of $n = 5$ coordinates. Thus the convolutional code may correct the erasure pattern by reducing the window length to $j' + 1 = 5$ blocks.

Here, if $v \in \mathbb{F}^n$, then $v_\Gamma \in \mathbb{F}^{|\Gamma|}$ denotes the projection of v onto the coordinates in Γ . Then if $v(D) = \sum_{j \in \mathbb{N}} v_j D^j \in \mathbb{F}[D]^n$, we use the notation $v(D)_\Gamma = \sum_{j \in \mathbb{N}} (v_j)_\Gamma D^j \in \mathbb{F}[D]^{|\Gamma|}$.

For a matrix $G(D) \in \mathbb{F}[D]^{k \times n}$, we denote by $G(D)_\Gamma \in \mathbb{F}[D]^{k \times |\Gamma|}$ the matrix whose rows are the rows of $G(D)$ restricted to Γ .

Observe that if $G(D) \in \mathbb{F}[D]^{k \times n}$ is a generator matrix of \mathcal{C} , then the rows of $G(D)_\Gamma \in \mathbb{F}^{k \times |\Gamma|}$ generate \mathcal{C}_Γ , although they may not be $\mathbb{F}[D]$ -linearly independent.

We may now extend the definition of (r, ∂) -locality for block codes from [16, Def. 1] to convolutional codes.

Definition 10. We say that an (n, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}[D]^n$ has (r, ∂) -locality if there exist non-empty sets Γ_i , for $i = 1, 2, \dots, g$, such that $[n] = \bigcup_{i=1}^g \Gamma_i$, and

- 1) $|\Gamma_i| \leq r + \partial - 1$,
- 2) $d(\mathcal{C}_{\Gamma_i}^0) \geq \partial$,

for $i = 1, 2, \dots, g$. Here, we write $\mathcal{C}_{\Gamma_i}^0$ instead of $(\mathcal{C}_{\Gamma_i})^0 = (\mathcal{C}^0)_{\Gamma_i}$ for simplicity. Thus, $\mathcal{C}_{\Gamma_i}^0$ denotes the block code associated (Definition 8) to the restriction (Definition 9) of \mathcal{C} on Γ_i .

We say then that \mathcal{C} is an (n, k, r, ∂) locally repairable convolutional code, or LRCC for short. The set Γ_i is called the i th local group, for $i = 1, 2, \dots, g$, and r and ∂ are called the locality and local distance of \mathcal{C} , respectively.

In other words, we consider local groups in each block of n symbols, corresponding to terms $v_j \in \mathbb{F}^n$ in a codeword $v(D) = \sum_{j \in \mathbb{N}} v_j D^j \in \mathcal{C}$. See Fig. 3 for a graphical example of a $(6, 3, 2, 2)$ LRCC with 2 local groups. In contrast to block codes, local repair with only one local group ($g = 1$) per block already outperforms sliding-window repair even when $j = 0$, in terms of total contacted nodes, see Fig. 4.

We state now the local erasure-correction capability of LRCCs. Definition 10 is given so that the following result holds. The proof is straightforward.

Proposition 3. Let $\mathcal{C} \subseteq \mathbb{F}[D]^n$ be an (n, k, r, ∂) LRCC with local groups Γ_i , for $i = 1, 2, \dots, g$. Fix $j \in \mathbb{N}$ and $i = 1, 2, \dots, g$. For all $v(D) = \sum_{j \in \mathbb{N}} v_j D^j \in \mathcal{C}$, if $v^* \in \mathbb{F}^{|\Gamma_i|}$ is the vector $(v_j)_{\Gamma_i} \in \mathbb{F}^{|\Gamma_i|}$ with at most $\partial - 1$ erasures (see Definition 7), then we may uniquely recover the vector $(v_j)_{\Gamma_i}$ from v^* by using the restricted block code $\mathcal{C}_{\Gamma_i}^0 \subseteq \mathbb{F}^{|\Gamma_i|}$, without contacting nodes or reading symbols outside of Γ_i in the j th block of the convolutional code.

As it was the case for locally repairable block codes, the main goal, given the parameters n, k, r and ∂ (and now also δ and μ), is to obtain a corresponding LRCC with maximum global distance properties, which would allow for global erasure correction in case of catastrophic failures. In this work, we consider column distances for “global correction”, since we will focus on sliding-window erasure correction as in Theorem 1. See Fig. 4 for a graphical description of local

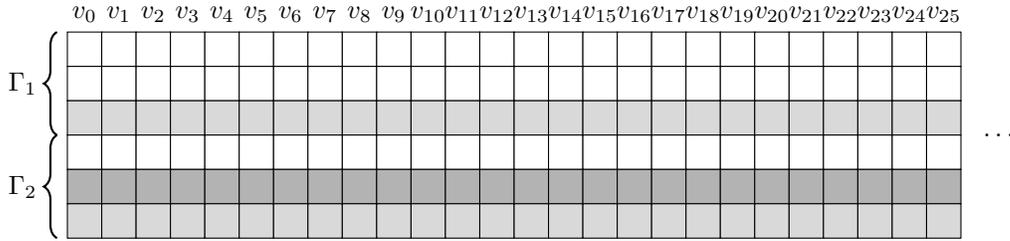


Fig. 3. Graphical description of an $(n, k, r, \delta) = (6, 3, 2, 2)$ LRCC with 2 local groups, Γ_1 and Γ_2 , each of size 3 and constituting the two halves of each column. White, light grey and dark grey boxes depict information symbols, local parities and global parities, respectively. Typically in this case ($\delta = 2$), local parities may be chosen as the XOR of the other symbols in the local group (see Construction 1).

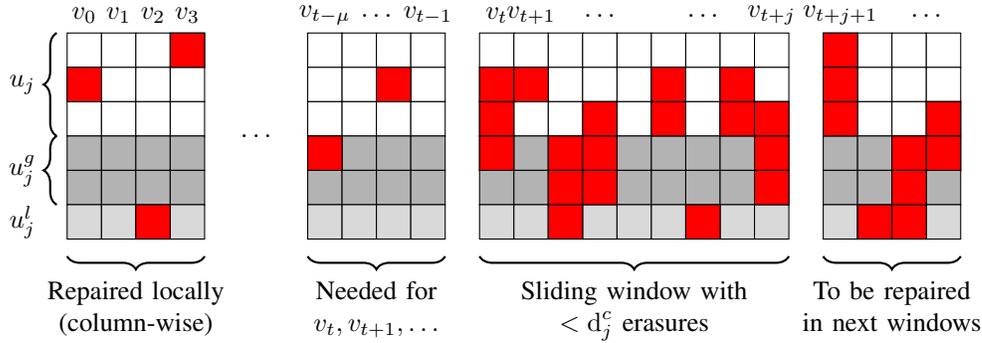


Fig. 4. Sliding-window repair combined with local repair. Here, an $(n, k, r, \delta) = (6, 3, 5, 2)$ LRCC, with one local group ($g = 1$) per block of n symbols, is depicted. Each column depicts a systematic encoded block $v_j = (u_j, u_j^g, u_j^l)$. White, light grey and dark grey boxes denote information symbols u_j , global parities u_j^g and local parities u_j^l , respectively. The local parities can be invoked block-wise to correct one erasure per block (since $\delta = 2$), requiring the other $r = 5$ symbols in the block for repair. If catastrophic erasures occur, with $< d_j^c$ erasures in each window $(v_t, v_{t+1}, \dots, v_{t+j})$, then sliding-window repair is invoked, as described in Theorem 1. Observe that, although sliding-window repair is somehow local, by setting $j = 0$, we see that we still need to contact and download the previous μn symbols from $v_{t-\mu}, v_{t-\mu+1}, \dots, v_{t-1}$ in order to repair one symbol in v_t , hence cannot compete with the considered type of locality even for relatively small memory μ .

repair combined with sliding-window global repair.

Before giving a Singleton bound on column distances of LRCCs, we need the following lemma.

Lemma 2. *Let $\mathcal{C} \subseteq \mathbb{F}[D]^n$ be an (n, k, r, δ) LRCC with local groups Γ_i , for $i = 1, 2, \dots, g$, as in Definition 10. Let $\Delta_i \subseteq \Gamma_i$ be such that $|\Gamma_i \setminus \Delta_i| \leq \delta - 1$, for $i = 1, 2, \dots, g$, and define $\Delta = \bigcup_{i=1}^g \Delta_i$ and $N = |\Delta|$. Then the restricted code $\mathcal{C}_\Delta \subseteq \mathbb{F}[D]^N$ has rank k , or in other words, it is an (N, k) convolutional code.*

In addition, if $G(D) = \sum_{j=0}^{\mu} G_j D^j$ is a reduced generator matrix of \mathcal{C} such that G_0 is full-rank, then $(G_0)_\Delta$ is also full-rank.

Proof. Let $G(D) \in \mathbb{F}[D]^{k \times n}$ be a generator matrix of \mathcal{C} . It suffices to prove that the rows of $G(D)_\Delta \in \mathbb{F}[D]^{k \times N}$ are $\mathbb{F}[D]$ -linearly independent.

Assume that there exists $x(D) \in \mathbb{F}[D]^k$ such that $x(D)G(D)_\Delta = 0$. If $v(D) = x(D)G(D) \in \mathcal{C}$, then we have that $v(D)_\Delta = x(D)G(D)_\Delta = 0$. Write $v(D) = \sum_{j \in \mathbb{N}} v_j D^j$ and fix $j \in \mathbb{N}$. We then deduce that $(v_j)_\Delta = 0$, and therefore $(v_j)_{\Delta_i} = 0$, for $i = 1, 2, \dots, g$. Since $d(\mathcal{C}_{\Gamma_i}^0) \geq \delta$ and $|\Gamma_i \setminus \Delta_i| \leq \delta - 1$, we deduce that $(v_j)_{\Gamma_i} = 0$, for $i = 1, 2, \dots, g$. Now, because $[n] = \bigcup_{i=1}^g \Gamma_i$, we conclude that $v_j = 0$.

Thus we have proven that $x(D)G(D) = 0$. Since $G(D)$ has full rank, we conclude that $x(D) = 0$, and we are done.

The statement regarding G_0 and $(G_0)_\Delta$ is proven following the same lines. \square

We may now prove the above mentioned Singleton bound on column distances of LRCCs.

Theorem 2. *Given an (n, k, r, δ) LRCC $\mathcal{C} \subseteq \mathbb{F}[D]^n$ with a reduced generator matrix $G(D) = \sum_{h=0}^{\mu} G_h D^h$ such that G_0 is full-rank, it holds that*

$$d_j^c(\mathcal{C}) \leq (n-k)(j+1) - \left(\left\lceil \frac{k(j+1)}{r} \right\rceil - 1 \right) (\delta-1) + 1, \quad (4)$$

for all $j \in \mathbb{N}$.

Proof. Fix $j \in \mathbb{N}$. Observe that the column block code $\mathcal{C}_j^c \subseteq \mathbb{F}^{n(j+1)}$ is an (r, δ) -locally repairable block code as in [16, Def. 1]. However, it is not linear, thus [16, Th. 2.1] cannot be directly applied. Furthermore, we have that $|\mathcal{C}_j^c| < |\mathbb{F}|^{k(j+1)}$, hence [30, Th. 21], which works also for non-linear codes, cannot be applied either.

To prove our bound, we need to show that there is a pattern of

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

erasures in the block $[n(j+1)]$ of coordinates that the code \mathcal{C}_j^c cannot correct.

We assume for brevity that $k(j+1) = \ell r$, for some integer $\ell \in \mathbb{N}$. The case where r does not divide $k(j+1)$ is proven similarly, although the notation is more cumbersome. Furthermore, we may assume that $|\Gamma_i| = r + \partial - 1$, for all $i = 1, 2, \dots, g$, simply by adding zeros to codewords in \mathcal{C}_j^c . For convenience in the notation, we also identify Γ_i in the h th block of n coordinates with the set $\tilde{\Gamma}_{gh+i} = [(gh+i-1)(r+\partial-1)+1, (gh+i)(r+\partial-1)]$, for $i = 1, 2, \dots, g$ and $h = 0, 1, 2, \dots, j$ (keep in mind that $|\Gamma_i| = |\tilde{\Gamma}_{gh+i}|$, but $\Gamma_i \subseteq [n]$, while $\tilde{\Gamma}_{gh+i} \subseteq [n(j+1)]$).

For an erasure pattern $\mathcal{E} \subseteq [n(j+1)]$ with $e = |\mathcal{E}|$, define $\mathcal{E}_u = \mathcal{E} \cap \tilde{\Gamma}_u$ and $\mathcal{R}_u = \tilde{\Gamma}_u \setminus \mathcal{E}_u$, for $u = 1, 2, \dots, g(j+1)$. Note that we may decompose

$$e = n(j+1) - (r + \partial - 1)\ell + \partial.$$

Therefore we may choose the erasure pattern \mathcal{E} such that $\mathcal{R}_u = \tilde{\Gamma}_u$, for $u = 1, 2, \dots, \ell - 1$, $|\mathcal{R}_\ell| = r - 1$, and $\mathcal{R}_u = \emptyset$, for $u = \ell + 1, \ell + 2, \dots, n(j+1)$.

Choose subsets $\Delta_i \subseteq \Gamma_i$ such that $|\Delta_i| = r$, for $i = 1, 2, \dots, g$, and $\mathcal{R}_\ell \subseteq \tilde{\Delta}_\ell$ (with the previous identification). In particular, $\mathcal{C}_\Delta \subseteq \mathbb{F}[D]^N$ is an (N, k) convolutional code by Lemma 2, where $N = |\Delta|$ and $\Delta = \bigcup_{i=1}^g \Delta_i$. Note that $(G_0)_\Delta \in \mathbb{F}^{k \times |\Delta|}$ is full-rank, for a given reduced generator matrix $G(D) = \sum_{h=0}^{\mu} G_h D^h$ of \mathcal{C} such that G_0 is full-rank, by Lemma 2. Hence Proposition 1 can be applied to \mathcal{C}_Δ .

Next, because it holds that

$$N(j+1) - |\mathcal{E} \cap \tilde{\Delta}| = \sum_{u=1}^{g(j+1)} |\mathcal{R}_u \cap \tilde{\Delta}_u| = (\ell-1)r + (r-1) = k(j+1) -$$

and $d_j^c(\mathcal{C}_\Delta) \leq (N-k)(j+1) + 1$ by Proposition 1, there exist $d_1, d_2 \in (\mathcal{C}_\Delta)_j^c$ such that $d_1 \neq d_2$ and $(d_1)_{\mathcal{R} \cap \tilde{\Delta}} = (d_2)_{\mathcal{R} \cap \tilde{\Delta}}$, where $\mathcal{R} = [n(j+1)] \setminus \mathcal{E}$.

Since $(\mathcal{C}_\Delta)_j^c \subseteq (\mathcal{C}_j^c)_{\tilde{\Delta}}$ (actually equality holds), there exist codewords $c_1, c_2 \in \mathcal{C}_j^c$ such that $d_1 = (c_1)_{\tilde{\Delta}}$ and $d_2 = (c_2)_{\tilde{\Delta}}$. Since $d_1 \neq d_2$, it must hold that $c_1 \neq c_2$. Finally, since $(d_1)_{\mathcal{R}_u \cap \tilde{\Delta}_u} = (d_2)_{\mathcal{R}_u \cap \tilde{\Delta}_u}$, it must hold that $(c_1)_{\mathcal{R}_u} = (c_2)_{\mathcal{R}_u}$, for $u = 1, 2, \dots, n(j+1)$. This is due to

- 1) Item 2 in Definition 10 if $u < \ell$, since $\mathcal{R}_u = \tilde{\Gamma}_u$ and $|\mathcal{R}_u \setminus \tilde{\Delta}_u| \leq \partial - 1$,
- 2) the fact that $\mathcal{R}_\ell \subseteq \tilde{\Delta}_\ell$ if $u = \ell$, or
- 3) the fact that $\mathcal{R}_u = \emptyset$ if $u > \ell$.

Therefore $(c_1)_{\mathcal{R}} = (c_2)_{\mathcal{R}}$ while $c_1 \neq c_2$, hence the code \mathcal{C}_j^c cannot correct the erasure pattern \mathcal{E} correctly, and we are done. \square

In the next section, we show how to explicitly construct a non-catastrophic LRCC attaining the previous bound, for all $j = 0, 1, 2, \dots, L$, where L is as in (2), for fields of any characteristic but sufficiently large.

Remark 1. Recall that, by Proposition 2, a convolutional code that is j -MDS is also h -MDS, for all $h = 0, 1, 2, \dots, j$. However, it is not clear whether a code attaining the bound (4) for some j implies attaining the bound for $h < j$. We leave this as an open problem. In any case, Construction 1 below based on a j -MSRD convolutional code attains the bound (4) for all $h = 0, 1, 2, \dots, j$.

Remark 2. Observe that, in the proof of Theorem 2, we make use of Proposition 1 on the restricted code \mathcal{C}_Δ . Note that \mathcal{C}_Δ may be catastrophic even if \mathcal{C} is non-catastrophic. However, \mathcal{C}_Δ still satisfies the weaker assumption in Proposition 1.

IV. COMPARISON BETWEEN LRCCS AND LRCs

In this section, we revisit the advantages of optimal LRCCs (those whose column distances attain (4)) over optimal block LRCs. As mentioned in Section I,

- 1) optimal LRCCs can correct in a window of size $n(j+1)$ more erasures than an optimal block LRC of the same rate and locality, and of block length $n(j+1)$ (see Example 1 and Fig. 5 below), thanks to adjusting the sliding-window size, and
- 2) optimal LRCCs may adjust the sliding-window size in order to contact less nodes when intermediate erasure patterns occur (see Example 2 below). Hierarchical LRCs [32] also require contacting less nodes for intermediate erasure patterns, however their global distance is strictly lower than that of optimal LRCs for a given block length (see [32, Th. 2.1]), in contrast with optimal LRCCs.

We now illustrate these two advantages with Examples 1 and 2, respectively.

Example 1. Consider a $(6, 3)$ convolutional code that encodes a stream of file vectors over a finite field \mathbb{F} , each of length $k = 3$, into a stream of encoded vectors, each of length $n = 6$. Assume a node in the storage system stores a symbol over \mathbb{F} , and call *block* each set of $n = 6$ coordinates supporting each encoded vector. In this example, each block forms a *local group*. If the code has locality $r = 5$ and local distance $\partial = 2$ (Section III), it means that a single node erasure ($\partial - 1$ node erasures) in each block may be repaired by only contacting the other 5 nodes in that block.

The code can correct erasure patterns with up to $d_j^c - 1$ erasures in every window of $j + 1$ consecutive blocks of n symbols, where $j = 0, 1, 2, \dots$ can be adjusted. Assume that an erasure pattern as in Fig. 5 occurs, with 21 erasures in a given window of $j + 1 = 9$ blocks. If the LRCC has optimal j th column distance d_j^c (as in Corollaries 1 and 2), then $d_j^c = 18$, for $j = 8$, and the code cannot correct such erasures considering such a window. Furthermore, an optimal block LRC with block length $(j + 1)n = 54$, dimension $(j + 1)k = 27$, locality $r = 5$ and local distance $\partial = 2$, also has global distance 18 (see [11, Eq. (1)]), hence it cannot correct that erasure pattern either.

However, we may adjust the window for the LRCC. Consider instead windows of length $j' + 1 = 5$, as in Fig. 5. If the LRCC also has optimal j' th column distance (as in Corollaries 1 and 2), then $d_{j'}^c = 12$ for $j' = 4$. Observe that now every window of $j' + 1 = 5$ consecutive blocks of n symbols contains at most 11 erasures. Therefore, the LRCC may correct such an erasure pattern by sliding the new adjusted window of length $j' + 1 = 5$, whereas the optimal block LRC as in the previous paragraph cannot.

The disadvantage of the LRCC is that, in order to perform such an erasure correction, we need to read the content (which

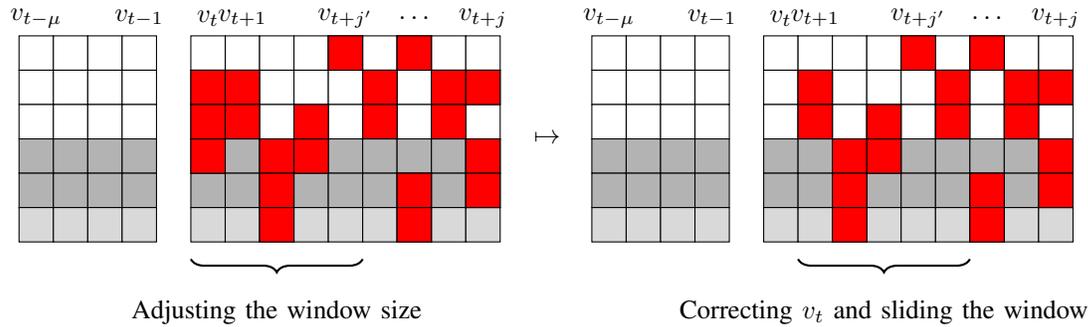


Fig. 5. Illustration of Example 1. The window consisting of the $j + 1 = 9$ blocks $v_t, v_{t+1}, \dots, v_{t+j}$ contains 21 erasures. Hence it cannot be corrected by an LRCC with optimal j th column distance, which is 18, by considering that window. Furthermore, it could not be corrected either by a block LRC of length $(j + 1)n = 54$, dimension $(j + 1)k = 27$, locality $r = 5$ and local distance $\partial - 1 = 2$, since its distance is also 18. Assume the LRCC also has optimal j' th column distance for $j' = 4$, which would be 12. Then the LRCC may correct the erasure pattern by reducing the window length to $j' + 1 = 5$ blocks.

needs to be correct) of the μ blocks of n symbols previous to such a window (see Fig. 5), where μ is the memory of the LRCC. Observe that, even if we require the previous μ blocks to be correct, the average number of erasures that the LRCC can correct is still strictly larger than that correctable by an optimal block LRC of the same rate. This is because, for $t = 0$, it holds that $v_{-1} = v_{-2} = \dots = v_{-\mu} = 0$, hence they are always considered to be correct.

Example 2. Consider now a $(6, 4)$ LRCC with locality $r = 5$ and local distance $\partial = 2$ (Section III). Assume also that the code has memory $\mu = 5$ and degree $\delta = 20$ (Subsection II-A) and that it is an optimal LRCC (as in Corollaries 1 and 2). As in Example 1, such an LRCC can correct the same number of erasures in any window consisting of $L + 1 = 26$ consecutive blocks ($n(L + 1) = 156$ symbols) as an optimal block LRC of the same rate ($2/3$), same locality ($r = 5$), same local distance ($\partial = 2$) and total length $n(L + 1) = 156$. In both cases, such a number of erasures is $32 = (n - k)(L + 1) - \left\lceil \frac{k(L+1)}{r} \right\rceil + 1$ (see Theorem 2 and [11, Eq. (1)], respectively).

However, the LRCC may correct any erasure pattern with up to $(n - k)(j + 1) - \left\lceil \frac{k(j+1)}{r} \right\rceil + 1$ erasures in any window of $n(j + 1)$ consecutive nodes, for all $j = 0, 1, \dots, L = 25$. This only requires reading and downloading the content of the remaining nodes in that window (that is, $k(j+1) + \left\lceil \frac{k(j+1)}{r} \right\rceil - 1$ symbols), plus another $\mu N = \mu(n - 1) = 25$ previous nodes (see Fig. 6). For instance, for $j = 2$, we may repair any erasure pattern with up to 4 erasures in any consecutive 18 nodes (3 blocks of 6 nodes), by contacting 14 nodes in that window, plus another $\mu N = 25$ previous nodes. The optimal LRC, in contrast, would always require contacting $124 = k(L + 1) + \left\lceil \frac{k(L+1)}{r} \right\rceil - 1$ other nodes in order to repair any 4 erasures in 3 blocks of 6 nodes, since at least one of these blocks contains 2 erasures, which cannot be repaired locally.

Therefore, adjusting the window size when using an LRCC also reduces the number of nodes that need to be contacted. Thus sliding-window erasure correction of LRCCs provides a type of erasure correction in between local and global erasure correction. Hierarchical block LRCs [32] also provide a type of erasure correction in between local and global. However,

the number of global erasures that they may correct is strictly less than $32 = (n - k)(L + 1) - \left\lceil \frac{k(L+1)}{r} \right\rceil + 1$, if they have at least 2 levels of locality, rate k/n , locality r (the smaller locality) and block length $n(L + 1)$ as in this example (see [32, Th. 2.1]).

V. LRCCS BASED ON SUM-RANK CONVOLUTIONAL CODES

In this section, we will show how to construct non-catastrophic LRCCs attaining the bound in Theorem 2, for $j = 0, 1, \dots, L$, using a j -MSRD convolutional code (see Definition 13 below). To that end, we will use the notion of *sum-rank weight* on each block of a convolutional code. Sum-rank weights were first defined in [29] for error correction in multishot network coding (see also [21], [22], [24], [27], [36] and the references therein). They were implicitly considered earlier in the space-time coding literature (see [19, Sec. III]), and they have been first used for locally repairable block codes in [23].

Throughout this section, we will fix a prime power q and a positive integer m , and we will assume that $\mathbb{F} = \mathbb{F}_{q^m}$. Fix an ordered basis $\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ of \mathbb{F}_{q^m} over \mathbb{F}_q . For any positive integer s , we denote by $M_{\mathcal{A}} : \mathbb{F}_{q^m}^s \rightarrow \mathbb{F}_q^{m \times s}$ the corresponding *matrix representation map*, given by

$$M_{\mathcal{A}}(c) = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1s} \\ c_{21} & c_{22} & \dots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{ms} \end{pmatrix} \in \mathbb{F}_q^{m \times s}, \quad (5)$$

where $c = \sum_{i=1}^m \alpha_i(c_{i,1}, c_{i,2}, \dots, c_{i,s}) \in \mathbb{F}_{q^m}^s$ and $c_{i,j} \in \mathbb{F}_q$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, s$.

Throughout this section, we will also fix a number of local groups g , a locality r , and the *sum-rank length decomposition* $N = gr$. The following definition is given in [29].

Definition 11 ([29]). Let $c = (c^{(1)}, c^{(2)}, \dots, c^{(g)}) \in \mathbb{F}_{q^m}^N$, where $c^{(i)} \in \mathbb{F}_{q^m}^r$, for $i = 1, 2, \dots, g$. We define the *sum-rank weight* of c as

$$\text{wt}_{SR}(c) = \sum_{i=1}^g \text{Rk}(M_{\mathcal{A}}(c^{(i)})).$$

We extend sum-rank weights to convolutional codes as follows.

Definition 12. Given $v(D) = \sum_{j \in \mathbb{N}} v_j D^j \in \mathbb{F}_{q^m}[D]^N$, we define its *sum-rank weight* as

$$\text{wt}_{SR}(v(D)) = \sum_{j \in \mathbb{N}} \text{wt}_{SR}(v_j).$$

Given an (N, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^N$, we define its *sum-rank free distance* as

$$d_{SR}(\mathcal{C}) = \min\{\text{wt}_{SR}(v(D)) \mid v(D) \in \mathcal{C} \text{ and } v(D) \neq 0\}.$$

Finally, we define the j th *sum-rank column distance* of \mathcal{C} as

$$d_{SR,j}^c(\mathcal{C}) = d_{SR}(\mathcal{C}_j^c),$$

where \mathcal{C}_j^c is as in Definition 5, for $j \in \mathbb{N}$.

Observe that, for any $c = (c^{(1)}, c^{(2)}, \dots, c^{(g)}) \in \mathbb{F}_{q^m}^N$, where $c^{(i)} \in \mathbb{F}_{q^m}^r$, for $i = 1, 2, \dots, g$, it holds that

$$\text{wt}_{SR}(c) = \sum_{i=1}^g \text{Rk}(M_{\mathcal{A}}(c^{(i)})) \leq \sum_{i=1}^g \text{wt}(c^{(i)}) = \text{wt}(c),$$

since the rank of a matrix is at most the number of its non-zero columns. Hence, the following result follows immediately from its Hamming-metric counterpart (Propositions 1 and 2).

Proposition 4. Given a non-catastrophic (N, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^N$, it holds that

$$d_{SR,j}^c(\mathcal{C}) \leq (N - k)(j + 1) + 1, \quad (6)$$

for all $j \in \mathbb{N}$. Furthermore, the following hold.

- 1) If $d_{SR,j+1}^c(\mathcal{C}) = (N - k)(j + 2) + 1$, then $d_{SR,j}^c(\mathcal{C}) = (N - k)(j + 1) + 1$, for $j \in \mathbb{N}$.
- 2) If $d_{SR,j}^c(\mathcal{C}) = (N - k)(j + 1) + 1$, then $j \leq L$, where L is as in (2).

The previous proposition motivates the following definition.

Definition 13. We say that an (N, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^N$ is j -maximum-sum-rank-distance, or j -MSRD for short, if it is non-catastrophic and $d_{SR,j}^c(\mathcal{C}) = (N - k)(j + 1) + 1$.

We now describe how to construct LRCCs from sum-rank convolutional codes. This construction is inspired by [30, Const. I]. The idea is to split the construction of the global code into an outer code and local codes. The local codes can trivially be chosen as short MDS block codes. The difficult part is finding a suitable outer code.

Construction 1. Assume that $q \geq r + \partial - 1$, and choose:

- 1) *Outer code:* An (N, k) convolutional code $\mathcal{C}_{out} \subseteq \mathbb{F}_{q^m}[D]^N$.
- 2) *Local codes:* An MDS $(r + \partial - 1, r)$ block code $\mathcal{C}_{loc} \subseteq \mathbb{F}_q^{r + \partial - 1}$ with generator matrix $A \in \mathbb{F}_q^{r \times (r + \partial - 1)}$.
- 3) *Global code:* We define the global code $\mathcal{C}_{glob} \subseteq \mathbb{F}_{q^m}[D]^n$, with $n = (r + \partial - 1)g = N + (\partial - 1)g$, as the (n, k) convolutional code given by

$$\mathcal{C}_{glob} = \left\{ \sum_{j \in \mathbb{N}} (v_j \text{Diag}_g(A)) D^j \mid \sum_{j \in \mathbb{N}} v_j D^j \in \mathcal{C}_{out} \right\},$$

where $\text{Diag}_g(A)$ is defined as a block-diagonal matrix with $A \in \mathbb{F}_q^{r \times (r + \partial - 1)}$ repeated g times (recall that $N = gr$ and $n = g(r + \partial - 1)$):

$$\text{Diag}_g(A) = \text{Diag}(A, A, \dots, A) = \begin{pmatrix} A & 0 & \dots & 0 \\ 0 & A & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A \end{pmatrix} \in \mathbb{F}_q^{N \times n}.$$

Observe that if $G_{out}(D) = \sum_{j=0}^{\mu} G_{out,j} D^j \in \mathbb{F}_{q^m}[D]^{k \times N}$ is a generator matrix of $\mathcal{C}_{out} \subseteq \mathbb{F}_{q^m}[D]^N$, then a generator matrix of $\mathcal{C}_{glob} \subseteq \mathbb{F}_{q^m}[D]^n$ is simply given by

$$G_{glob}(D) = \sum_{j=0}^{\mu} G_{glob,j} D^j = \sum_{j=0}^{\mu} (G_{out,j} \text{Diag}_g(A)) D^j \in \mathbb{F}_{q^m}[D]^{k \times n}.$$

In addition, note that multiplying a vector $v(D) \in \mathbb{F}[D]^N$ on the right by a rank- N constant matrix $C \in \mathbb{F}^{N \times n}$ preserves the degree of $v(D)$. Hence if $G_{out}(D)$ is reduced, then so is $G_{glob}(D)$. It also follows easily that if $G_{out}(D)$ is basic, then so is $G_{glob}(D)$. Thus we deduce the following.

Lemma 3. In Construction 1, if \mathcal{C}_{out} is non-catastrophic, then so are \mathcal{C}_{glob} and $(\mathcal{C}_{glob})_{\Delta}$, for any $\Delta = \bigcup_{i=1}^g \Delta_i \subseteq [n]$ such that $\Delta_i \subseteq \Gamma_i$ and $|\Delta_i| \geq r$, for $i = 1, 2, \dots, g$. Here, we denote $\Gamma_i = [(r + \partial - 1)(i - 1) + 1, (r + \partial - 1)i] \subseteq [n]$, for $i = 1, 2, \dots, g$.

As it was the case for locally repairable block codes (see [23, Lemma 1]), any LRCC whose local codes are all encoded by the same linear MDS code over the subfield \mathbb{F}_q , are necessarily of the form of Construction 1. For this reason, Construction 1 not only is natural, but it is somehow unavoidable.

The main reason why the sum-rank metric gives the erasure-correction capability of the global code in Construction 1 is due to the following lemma. The direct implication was proven in [29, Th. 1], whereas the converse was given in [24, Th. 1].

Lemma 4 ([24], [29]). Recall that $N = gr$. Given a (linear or non-linear) block code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^N$ and matrices $B_i \in \mathbb{F}_q^{r \times r}$, for $i = 1, 2, \dots, g$, there exists a decoder

$$D : \mathcal{C} \text{Diag}(B_1, B_2, \dots, B_g) \longrightarrow \mathcal{C}$$

(depending on the matrices B_1, B_2, \dots, B_g), such that $D(c \text{Diag}(B_1, B_2, \dots, B_g)) = c$, for all $c \in \mathcal{C}$, if and only if,

$$N - \sum_{i=1}^g \text{Rk}(B_i) < d_{SR}(\mathcal{C}).$$

We may now prove the main result of this section, which states that \mathcal{C}_{glob} in Construction 1 has maximum h th sum-rank column distance among all non-catastrophic (n, k, r, ∂) LRCC, for $h = 0, 1, 2, \dots, j$, if \mathcal{C}_{out} is j -MSRD.

Theorem 3. In Construction 1, \mathcal{C}_{glob} is an (n, k, r, ∂) LRCC. Furthermore, if $j \in \mathbb{N}$ and \mathcal{C}_{out} is j -MSRD (thus non-catastrophic), then \mathcal{C}_{glob} is non-catastrophic and

$$d_h^c(\mathcal{C}_{glob}) = (n - k)(h + 1) - \left(\left\lceil \frac{k(h + 1)}{r} \right\rceil - 1 \right) (\partial - 1) + 1, \quad (7)$$

for all $h = 0, 1, 2, \dots, j$.

Proof. First, it follows easily from the definitions and Construction 1 that \mathcal{C}_{glob} is an (n, k, r, ∂) LRCC. The non-catastrophic property is part of Lemma 3. Therefore, we only need to show that (7) holds for $h = j$, since if \mathcal{C}_{out} is j -MSRD, then \mathcal{C}_{out} is h -MSRD, for all $h = 0, 1, 2, \dots, j$, by Proposition 4.

The proof now follows the lines of the proof of Theorem 2, for a given $j \in \mathbb{N}$. Now we need to show that the block code $(\mathcal{C}_{glob})_j^c \subseteq \mathbb{F}_{q^m}^{n(j+1)}$ can correct any pattern of

$$e = (n - k)(j + 1) - \left(\left\lceil \frac{k(j + 1)}{r} \right\rceil - 1 \right) (\partial - 1)$$

erasures in the block $[n(j + 1)]$ of coordinates.

Let $c \in (\mathcal{C}_{glob})_j^c$ be the truncated global codeword. Observe that $c \in \mathbb{F}_{q^m}^{n(j+1)}$ is a block codeword, and by construction, there exists $x \in (\mathcal{C}_{out})_j^c$ such that $c = x \text{Diag}_{g(j+1)}(A)$.

Let $\mathcal{E}_{gh+i} \subseteq [r + \partial - 1]$ be the erasure pattern in the i th local group in the h th block of n coordinates, and define $\mathcal{R}_{gh+i} = [r + \partial - 1] \setminus \mathcal{E}_{gh+i}$, for $i = 1, 2, \dots, g$ and $h = 0, 1, 2, \dots, j$. The truncated global codeword with such an erasure pattern is

$$c^* = x \text{Diag}(A|_{\mathcal{R}_1}, A|_{\mathcal{R}_2}, \dots, A|_{\mathcal{R}_{g(j+1)}}) \in \mathbb{F}_{q^m}^{n(j+1)-e}, \quad (8)$$

where $e = \sum_{u=1}^{g(j+1)} |\mathcal{E}_u| = n(j + 1) - \sum_{u=1}^{g(j+1)} |\mathcal{R}_u|$.

As in Theorem 2, we assume for simplicity that $k(j + 1) = \ell r$, for some integer $\ell \in \mathbb{N}$. As in that theorem, note that we may decompose

$$e = n(j + 1) - (r + \partial - 1)\ell + \partial - 1. \quad (9)$$

As discussed in the proof of [30, Th. 24], the worst-case erasure pattern is obtained when erasures concentrate in the smallest number of local groups. Thus by (9), in the worst case we have without loss of generality that $\mathcal{R}_u = [r + \partial - 1]$, for $u = 1, 2, \dots, \ell - 1$, $|\mathcal{R}_\ell| = r$, and $\mathcal{R}_u = \emptyset$, for $u = \ell + 1, \ell + 2, \dots, n(j + 1)$. Since \mathcal{C}_{loc} is an $(r + \partial - 1, r)$ MDS code, we have that $\text{Rk}(A|_{\mathcal{R}_u}) = r$, for $u = 1, 2, \dots, \ell$. Therefore, in the worst case, c^* is a codeword of $(\mathcal{C}_{out})_j^c$ with

$$\sum_{u=1}^{n(j+1)} \text{Rk}(A|_{\mathcal{R}_u}) = \ell r = k(j + 1)$$

sum-rank erasures. Therefore, by Lemma 4 and the fact that $d_{SR}((\mathcal{C}_{out})_j^c) = d_{SR,j}^c(\mathcal{C}_{out}) = (N - k)(j + 1) + 1$, we conclude that $(\mathcal{C}_{out})_j^c$ can correct such pattern of erasures, and hence so can $(\mathcal{C}_{glob})_j^c$ by local repair, and we are done. \square

We conclude by plugging in Construction 1 the MSRD convolutional codes from [21] (see Appendix A) as outer codes, and applying the previous theorem.

Corollary 1. *If $N = gr$, $(N - k)|\delta$, $M = \max\{N - k, k\}$, $L = \lfloor \frac{\delta}{k} \rfloor + \delta / (N - k)$, $q \geq r + \partial - 1$ and $m \geq q^{M(L+2)-1}$, then there exists a non-catastrophic (n, k, r, ∂) LRCC $\mathcal{C}_{glob} \subseteq \mathbb{F}_{q^m}[D]^n$, of degree δ , satisfying (4) with equality, for $j = 0, 1, 2, \dots, L$, given as in Construction 1, and where $\mathcal{C}_{out} \subseteq \mathbb{F}_{q^m}[D]^N$ is the non-catastrophic L -MSRD convolutional code in Appendix A.*

Corollary 1 not only shows that the upper bound given in (4) is sharp, but also provides an explicit class of codes that achieves such a bound. Moreover, these codes exist for any characteristic (in particular, when $2|q$), and the local code may be arbitrary and with local fields of size $q \approx r + \partial - 1$, which are small. We may also choose $q = 2$ if $\partial = 2$ and local repair would simply consist in XORing. Their main disadvantage is the huge exponent m , which is in turn exponential in the degree δ and in $\max\{N - k, k\}$. However, the bound on m in the corollary is only a bound, and there are cases when m can be chosen much smaller (see Table I in [21]).

VI. PARTIAL j -MDS AND PARTIAL MDP CONVOLUTIONAL CODES

In this section, we introduce partial MDP convolutional codes, analogous to the concept of *partial MDS codes*, or LRC with *maximal recoverability* (MR), introduced in [4], [10]. We will conclude by showing that the codes in Corollary 1 are partial MDP. The main motivation to consider these codes is that they can correct erasure patterns analogous to those correctable by partial MDS codes (as shown in Fig. 6 below), which are known to be all the information-theoretically correctable patterns for a given locality, local distance and information rate. Also analogously to the case of partial MDS codes, it holds that partial MDP convolutional codes have optimal j th column distance up to $j = L$, as in Proposition 2. Hence they have at least the erasure correction capability of optimal LRCCs, although we were not able to prove whether the reversed implication holds (see Remark 3).

Definition 14. With notation as in Definition 10, and for $j \in \mathbb{N}$, we say that an (n, k, r, ∂) LRCC $\mathcal{C} \subseteq \mathbb{F}[D]^n$ is *partial j -MDS* if the following holds: For all $\Delta_i \subseteq \Gamma_i$ such that $|\Gamma_i \setminus \Delta_i| = \partial - 1$, for $i = 1, 2, \dots, g$, the restricted (N, k) convolutional code $\mathcal{C}_\Delta \subseteq \mathbb{F}[D]^N$ is non-catastrophic and j -MDS (Definition 6), where $\Delta = \bigcup_{i=1}^g \Delta_i$ and $N = |\Delta|$.

Some explanations about Definition 14 are in order.

First, recall that the restricted convolutional code $\mathcal{C}_\Delta \subseteq \mathbb{F}[D]^N$ in the previous definition has rank k by Lemma 2, thus the definition is consistent.

Similar to the case of block codes (replacing j -MDS by MDS), the term partial j -MDS is motivated by the fact that the column distances attain the bound (4) (see Proposition 5 below), thus they have smaller column distances than those of j -MDS codes (this is the price to pay for locality). However, partial j -MDS codes as in Definition 14 can be seen as j -MDS codes that can be *added* locality in some optimal sense: We can recover some other j -MDS code after removing *any* (maximal) collection of local parities, not only the added ones. Due to this reason, we gain a considerable flexibility in the erasure patterns that can be corrected (see Fig. 6).

In the block case, partial MDS codes can be equivalently defined as follows: A locally repairable block code is partial MDS if it can correct all erasure patterns that are information-theoretically correctable for the given local constraints r and ∂ and the given dimension k and length n . Obviously, if there are no local constraints ($\partial = 1$ for instance), then being

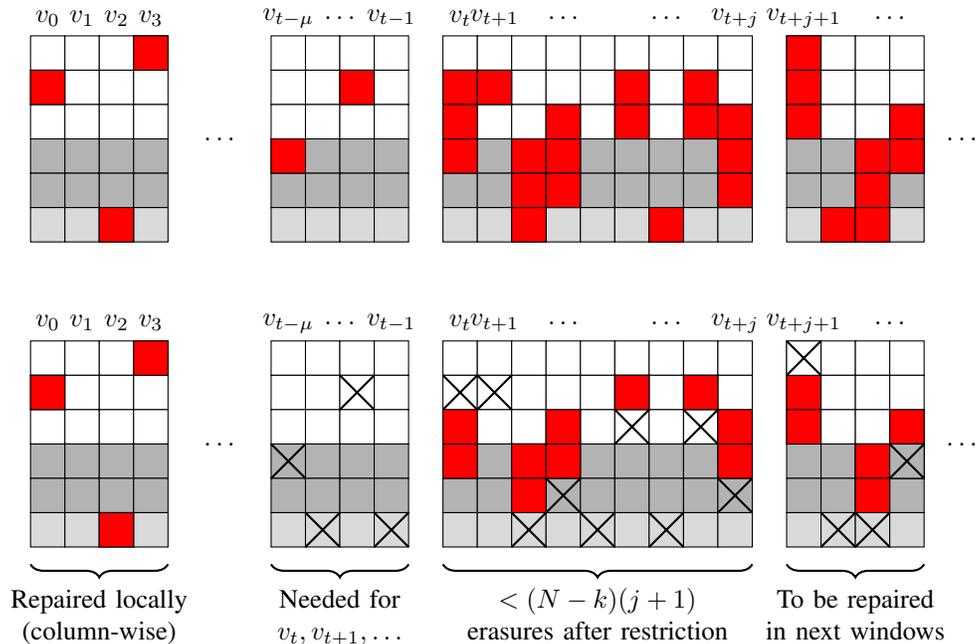


Fig. 6. Sliding-window repair in a partial j -MDS code, with parameters as in Fig. 4 (thus $N = 5$). Following Definition 14 and assuming an erasure pattern as in Fig. 4 (upper figure), we may remove one local parity in each block (depicted as a crossed box) before proceeding with the sliding-window repair for the restricted j -MDS code, which would be performed as in Theorem 1 and may correct up to $(N - k)(j + 1)$ erasures after removing *arbitrary* local parities. Thus a local parity to be removed should be chosen as one of the erased symbols in case there are erased symbols in the corresponding block. Otherwise, we may remove any local parity. After the catastrophic erasures are corrected, the local parities are added again locally (column-wise) in case they were erased.

able to correct all information-theoretically correctable erasure patterns is equivalent to being MDS.

See Fig. 6 for a graphical description of sliding-window repair combined with local repair in a partial j -MDS convolutional code.

We now show that partial j -MDS codes attain the bound (4), hence being optimal LRCCs in terms of column distances. We need a preliminary lemma, which is of interest by itself and which follows directly from Definition 14 and Proposition 2.

Lemma 5. *If an LRCC is partial j -MDS, then it is partial h -MDS, for all $h = 0, 1, 2, \dots, j$.*

Proposition 5. *If an (n, k, r, ∂) LRCC is partial j -MDS for some $j \in \mathbb{N}$, then its column distances attain the bound (4), for all $h = 0, 1, 2, \dots, j$.*

Proof. By the previous lemma, we only need to prove the result for $h = j$. For such a case, the proof follows exactly the same lines as the proof of Theorem 3, and is left to the reader. \square

Remark 3. *In the block case, the converse is not true. For instance, Tamo-Barg codes [34] are locally repairable codes with optimal global distance, but cannot always be maximally recoverable (partial MDS) by the field-size bound in [12, Eq. (2)]. We conjecture, but do not prove or disprove, that not every LRCC attaining the bound (4), for some $j \in \mathbb{N}$, is a partial j -MDS convolutional code.*

Our next goal is to define partial MDP convolutional codes, which are partial j -MDS for the maximum value of j . We

first need the following lemma, which follows directly from the definitions and Proposition 2.

Lemma 6. *Let $\mathcal{C} \subseteq \mathbb{F}[D]^n$ be an (n, k) convolutional code. For any $\Delta \subseteq [n]$, it holds that*

$$\delta(\mathcal{C}_\Delta) \leq \delta(\mathcal{C}).$$

In particular, if \mathcal{C}_Δ is j -MDS, then $j \leq \lfloor \frac{\delta}{k} \rfloor + \lfloor \frac{\delta}{N-k} \rfloor$, for $N = |\Delta|$ and $\delta = \delta(\mathcal{C})$.

We may now define partial MDP convolutional codes.

Definition 15. We say that an (n, k, r, ∂) LRCC $\mathcal{C} \subseteq \mathbb{F}[D]^n$ is *partial MDP* if it is partial L -MDS for $L = \lfloor \frac{\delta}{k} \rfloor + \lfloor \frac{\delta}{N-k} \rfloor$, where $N = n - g(\partial - 1)$ and $\delta = \delta(\mathcal{C})$.

The main purpose of this section is to show that the global code in Construction 1 based on an MSR-D outer code (for instance, that in Appendix A) is partial MDP. In particular, we will show the existence of partial MDP codes for general parameters, over any characteristic, for sufficiently large fields.

We first need the following lemma, which is [23, Th. 1]. Observe that we will make use of this lemma in the non-linear case.

Lemma 7 ([23]). *Recall that $N = gr$. Given a (linear or non-linear) block code $\mathcal{C} \subseteq \mathbb{F}_q^N$, it holds that*

$$d_{SR}(\mathcal{C}) = \min \{d(\mathcal{C} \text{Diag}(B_1, B_2, \dots, B_g)) \mid B_i \in \mathbb{F}_q^{r \times r} \text{ invertible}, i = 1, \dots, g\}$$

We may now prove the main result of this section.

Theorem 4. *In Construction 1, the following hold:*

APPENDIX A KNOWN CONSTRUCTION OF MSRD CONVOLUTIONAL CODES

In this appendix, we revisit the construction of non-catastrophic MSRD convolutional codes from [21], which is based on the superregular matrices introduced in [1]. To the best of our knowledge, this is the only known construction of MSRD convolutional codes. In addition, they admit general parameters, except that they usually require impractically large field sizes. Acceptable field sizes can be achieved for certain parameters. See Table I in [21] for a few instances.

Fix $1 \leq k \leq N$. As in [1] (see also [2]), we will restrict ourselves to (N, k) convolutional codes whose degree δ satisfies that $(N - k) \mid \delta$. For general parameters, see [28]. Define $M = \max\{N - k, k\}$ and $L = \lfloor \frac{\delta}{k} \rfloor + \delta/(N - k)$, as in (2). Let q be any prime power and assume that

$$m \geq q^{M(L+2)-1}. \quad (11)$$

The field will be then $\mathbb{F} = \mathbb{F}_{q^m}$. Let $\alpha \in \mathbb{F}_{q^m}$ be a *primitive normal element* over \mathbb{F}_q , that is, a primitive element of \mathbb{F}_{q^m} such that $\alpha, \alpha^q, \dots, \alpha^{q^{m-1}}$ form a basis of \mathbb{F}_{q^m} over \mathbb{F}_q . Such element exists for any finite field extension $\mathbb{F}_q \subseteq \mathbb{F}_{q^m}$ (see [18]). Define the matrix

$$T_j = \begin{pmatrix} \alpha^{[Mj]} & \alpha^{[Mj+1]} & \dots & \alpha^{[M(j+1)-1]} \\ \alpha^{[Mj+1]} & \alpha^{[Mj+2]} & \dots & \alpha^{[M(j+1)]} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{[M(j+1)-1]} & \alpha^{[M(j+1)]} & \dots & \alpha^{[M(j+2)-2]} \end{pmatrix} \in \mathbb{F}_{q^m}^{M \times M} \quad (12)$$

for $j = 0, 1, 2, \dots, L$, where $\alpha^{[i]} = \alpha^{q^i}$, for $i \in \mathbb{N}$. Finally, define the non-catastrophic (N, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^N$ as that with polynomial parity-check matrix

$$H = (A, B) \in \mathbb{F}_{q^m}[D]^{(N-k) \times N},$$

$$A = \sum_{j=0}^{\nu} A_j D^j \in \mathbb{F}_{q^m}[D]^{(N-k) \times (N-k)} \quad \text{and} \quad B = \sum_{j=0}^{\nu} B_j D^j \in \mathbb{F}_{q^m}[D]^{(N-k) \times k},$$

where $\nu = \delta/(N - k)$, $A_0 = I_{N-k}$, and B can be given from A by the rule

$$A^{-1}B = \sum_{j=0}^{\infty} T_j D^j \in \mathbb{F}_{q^m}((D))^{(N-k) \times k}.$$

The following theorem combines [9, Th. 3.1] with [21, Th. 5].

Theorem 5. *The (N, k) convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^N$ described above is non-catastrophic, has degree δ and is L -MSRD for any sum-rank length decomposition of N .*

ACKNOWLEDGEMENT

The authors are grateful for the anonymous reviewers' comments, which helped improve this manuscript.

REFERENCES

- [1] P. Almeida, D. Napp, and R. Pinto, "A new class of superregular matrices and MDP convolutional codes," *Linear Algebra and its Applications*, vol. 439, no. 7, pp. 2145–2157, 2013.
- [2] —, "Superregular matrices and applications to convolutional codes," *Linear Algebra and its Applications*, vol. 499, pp. 1–25, 2016.
- [3] M. Asteris and A. G. Dimakis, "Repairable fountain codes," *IEEE J. Select. Areas Comm.*, vol. 32, no. 5, pp. 1037–1047, May 2014.
- [4] M. Blaum, J. L. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," *IEEE Trans. Info. Theory*, vol. 59, no. 7, pp. 4510–4519, July 2013.
- [5] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain approach to reliable distribution of bulk data," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 56–67, Oct. 1998.
- [6] B. Chen, S. T. Xia, and J. Hao, "Locally repairable codes with multiple (r_i, δ_i) -localities," in *Proc. IEEE Int. Symp. Info. Theory*, June 2017, pp. 2038–2042.
- [7] A. Datta, "Locally repairable rapidRAID systematic codes — one simple convoluted way to get it all," in *Proc. IEEE Info. Theory Workshop*, Nov 2014, pp. 60–64.
- [8] R. Gabrys, E. Yaakobi, M. Blaum, and P. H. Siegel, "Constructions of partial MDS codes over small fields," *IEEE Trans. Info. Theory*, pp. 1–1, 2018.
- [9] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, "Strongly-MDS convolutional codes," *IEEE Trans. Info. Theory*, vol. 52, no. 2, pp. 584–598, Feb 2006.
- [10] P. Gopalan, C. Huang, B. Jenkins, and S. Yekhanin, "Explicit maximally recoverable codes with locality," *IEEE Trans. Info. Theory*, vol. 60, no. 9, pp. 5245–5256, Sept 2014.
- [11] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Info. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov 2012.
- [12] S. Gopi, V. Guruswami, and S. Yekhanin, "On maximally recoverable local reconstruction codes," *Electr. Colloq. Comp. Complexity (ECCC)*, vol. 24, no. 183, 2017.
- [13] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure storage," in *2012 USENIX Annual Technical Conference*, Boston, MA, 2012, pp. 15–26.
- [14] F. Ivanov, A. Kreshchuk, and V. Zyablov, "On the local erasure correction capacity of convolutional codes," in *2018 Int. Symp. Info. Theory and Applications (ISITA)*, Oct 2018, pp. 296–300.
- [15] S. Kadhe and A. Sprintson, "Codes with unequal locality," in *Proc. IEEE Int. Symp. Info. Theory*, July 2016, pp. 435–439.
- [16] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration and erasure correction," *IEEE Trans. Info. Theory*, vol. 60, no. 8, pp. 4637–4660, Aug 2014.
- [17] G. Kim and J. Lee, "Locally repairable codes with unequal local erasure correction," *IEEE Trans. Info. Theory*, vol. 64, no. 11, pp. 7137–7152, Nov 2018.
- [18] H. W. Lenstra and R. J. Schoof, "Primitive normal bases for finite fields," *Mathematics of Computation*, vol. 48, no. 177, pp. 217–231, 1987.
- [19] H.-F. Lu and P. V. Kumar, "A unified construction of space-time codes with optimal rate-diversity tradeoff," *IEEE Trans. Info. Theory*, vol. 51, no. 5, pp. 1709–1730, May 2005.
- [20] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.
- [21] R. Mahmood, A. Badr, and A. Khisti, "Convolutional codes with maximum column sum rank for network streaming," *IEEE Trans. Info. Theory*, vol. 62, no. 6, pp. 3039–3052, 2016.
- [22] U. Martínez-Peñas, "Skew and linearized Reed-Solomon codes and maximum sum rank distance codes over any division ring," *J. Algebra*, vol. 504, pp. 587–612, 2018.
- [23] U. Martínez-Peñas and F. R. Kschischang, "Universal and dynamic locally repairable codes with maximal recoverability via sum-rank codes," in *Proc. Allerton Conf. Comm. Control Comp.*, 2018, pp. 1–8, extended version: <https://arxiv.org/abs/1809.11158>.
- [24] —, "Reliable and secure multishot network coding using linearized Reed-Solomon codes," *IEEE Trans. Info. Theory*, pp. 1–19, 2019.
- [25] U. Martínez-Peñas and D. Napp, "Locally repairable convolutional codes with sliding window repair," in *Proc. IEEE Int. Symp. Info. Theory*, July 2019, pp. 2838–2842.
- [26] R. J. McEliece and R. P. Stanley, "The general theory of convolutional codes," *The Telecommunications and Data Acquisition Report*, vol. 42, no. 113, pp. 89–98, 1993.

- [27] D. Napp, R. Pinto, and V. Sidorenko, "Concatenation of convolutional codes and rank metric codes for multi-shot network coding," *Des., Codes, Crypto.*, vol. 86, no. 2, pp. 303–318, Feb. 2018.
- [28] D. Napp and R. Smarandache, "Constructing strongly MDS convolutional codes with maximum distance profile," *Advances in Mathematics of Communications*, vol. 10, no. 2, pp. 275–290, 2016.
- [29] R. W. Nóbrega and B. F. Uchôa-Filho, "Multishot codes for network coding using rank-metric codes," in *Proc. 2010 Third IEEE Int. Workshop on Wireless Network Coding*, 2010, pp. 1–6.
- [30] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Info. Theory*, vol. 60, no. 1, pp. 212–236, 2014.
- [31] J. Rosenthal and R. Smarandache, "Maximum Distance Separable Convolutional Codes," *Applicable Algebra in Engineering, Communication and Computing*, vol. 10, no. 1, pp. 15–32, Aug 1999. [Online]. Available: <https://doi.org/10.1007/s002000050120>
- [32] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "Codes with hierarchical locality," in *Proc. IEEE Int. Symp. Info. Theory*, June 2015, pp. 1257–1261.
- [33] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: novel erasure codes for big data," in *Proc. 39th int. conf. Very Large Data Bases*, ser. PVLDB'13, 2013, pp. 325–336.
- [34] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Info. Theory*, vol. 60, no. 8, pp. 4661–4676, Aug 2014.
- [35] V. Tomas, J. Rosenthal, and R. Smarandache, "Decoding of convolutional codes over the erasure channel," *IEEE Trans. Info. Theory*, vol. 58, no. 1, pp. 90–108, Jan 2012.
- [36] A. Wachter-Zeh, M. Stinner, and V. Sidorenko, "Convolutional codes in rank metric with application to random network coding," *IEEE Trans. Info. Theory*, vol. 61, no. 6, pp. 3199–3213, 2015.
- [37] A. Zeh and E. Yaakobi, "Bounds and constructions of codes with multiple localities," in *Proc. IEEE Int. Symp. Info. Theory*, July 2016, pp. 640–644.
- [38] B. Zhu, X. Li, H. Li, and K. W. Shum, "Replicated convolutional codes: A design framework for repair-efficient distributed storage codes," in *Proc. Allerton Conf. Comm. Control Comp.*, Sept 2016, pp. 1018–1024.