Reverse-Maximum Distance Profile Convolutional Codes over the Erasure Channel*

Virtudes Tomás[§], Joachim Rosenthal[†] and Roxana Smarandache[‡]

***Abstract*— The loss of transmitted packets over an erasure channel, such as the Internet, can generate delay of the received information due to retransmission, and this can have adverse effects in real-time applications. Error forward correction is a technique used to avoid this delay. Until now mainly block codes have been used for this purpose and convolutional codes have been much less studied. In this paper we study in detail the use of convolutional codes over this channel and we show that the complexity of decoding is polynomial. We see how maximum distance profile (MDP) convolutional codes can deal with situations which are not possible for a maximum distance separable (MDS) block code and we introduce a new concept: reverse-MDP convolutional codes. Reverse-MDP codes double the potential of MDP convolutional codes since they behave as MDP codes in a forward and a backward sense. Due to this fact, we propose this new kind of codes as very good candidates to improve the decoding process. In addition, we provide a particular construction for reverse-MDP convolutional codes.**

I. INTRODUCTION

The main and most widely used representative of the class of erasure channels is the Internet. To be transmitted through this type of communication channels the information is divided in packets that either arrive correctly to the receiver or do not arrive due to different reasons. One cause is that the packets are usually protected with mechanisms that allow to know if the information is in error and when this is the case, the packet is erased and shown as not received. Another reason is that the physical restrictions of the systems can make certain packets disappear sometimes. For instance, when a buffer is full packets can be dropped out until this situation is restored, and all those would appear as erased as well. The receiver knows which are the received packets and where exactly the erasures happen. The second reason provides the channel with the following special behavior: if a packet is erased at instant t, the probability that the packet received at instant $t+1$ is erased increases. Therefore erasures tend to occur in bursts. This is an important point to take into account when modeling the channel.

* Part of these results where presented in the 2009 IEEE International Symposium on Information Theory in Seoul, Korea [14].

§ Partially supported by Spanish grant MTM2008-06674-C02-01 and a grant of the Vicerectorat d'Investigació, Desenvolupament i Innovació of the Universitat d'Alacant for PhD students during a stay at Zürich Universität on charge to the same program. Department of Statistics and Operations Research, University of Alicante, Alicante, Spain. vtomas@ua.es

† The research of this author is supported in part by the Swiss National Science Foundation under grant no. 126948. Mathematics Institute, University of Zürich, Switzerland. rosenthal@math.uzh.ch

‡ The work of this author is supported by NSF Grants DMS-0708033 and TF-0830608. Department of Mathematics and Statistics, San Diego State University, San Diego, CA 92182-7720, USA. rsmarand@sciences.sdsu.edu

In order to mend this situation, the protocol that the Internet uses is the retransmission of lost packets until they are correctly received. But this delay on the information has damaging effects on real-time communications. Forward error correction is a technique that allows to avoid this retardment. Until now mainly block codes have been used for this purpose, with maximum distance separable (MDS) block codes being the ones achieving the best performance [3]. In this paper we consider the less studied class of convolutional codes [1], [2] to accomplish this task.

Convolutional codes treat the information as a complete sequence [9] and this provides them with what we call the sliding window property, that is, convolutional codes can move along the sequence and frame windows of different sizes where it is convenient. The information is not separated in blocks of a fixed length as in the block code case, i.e., they do not have a fixed grouping of a fixed length. Thanks to this property convolutional codes can adapt the decoding process to the concrete patterns of erasures happening in a sequence.

the fact that only erasures occur over the erasure channel makes the complexity of the decoding process to be polynomial. Using the subclass of maximum distance profile (MDP) convolutional codes [7] which perform like MDS block codes in each window size, we show with some examples how these codes achieve better results in situations where MDS block codes cannot complete the decoding [14].

With the intention of improving the decoding algorithm, we introduce a new type of codes: reverse-MDP convolutional codes. These codes are more powerful than the MDP ones because they behave as MDP codes not only forward, but in a backward direction as well. Thanks to this doubled MDP property of reverse-MDP codes we implement an inverted recovering process that allow to solve situations that neither MDS block codes nor MDP convolutional codes can deal with [15]. Based on these results we propose reverse-MDP convolutional codes as a very good alternative to block codes when transmitting over the erasure channel. In additton, we give a particular construction for this new subclass of codes.

The paper is organized as follows. Section II provides the necessary background on convolutional codes for the development of the paper. In Section III we illustrate our proposed decoding algorithm over the erasure channel. We also provide examples and special concerns to be noticed when comparing MDP convolutional codes with MDS block codes. In Section IV we introduce the idea of backwards

ISBN 978-963-311-370-7

decoding process and we define and prove the existence of reverse-MDP convolutional codes. In Section V we give a method to construct reverse-MDP convolutional codes and finally, we provide some conclusions in Section VI.

II. Convolutional codes

Let $\mathbb{F}$ be a finite field. We view a convolutional code $\mathcal{C}$ of rate k/n as a submodule of $\mathbb{F}[z]^n$ (see [5], [11], [10]) that can be described as

$$\mathcal{C} = \left\{ \boldsymbol{v}(z) \in \mathbb{F}[z]^n \,|\, \boldsymbol{v}(z) = G(z)\boldsymbol{u}(z),\, \boldsymbol{u}(z) \in \mathbb{F}[z]^k \right\}$$

where $G(z)$ is an $n \times k$ full-rank polynomial matrix called a **generator matrix** for $\mathcal{C}$, $\boldsymbol{u}(z)$ is the **information vector** and $\boldsymbol{v}(z)$ is the **code vector** or **codeword**.

The maximum degree of all polynomials in the j-th column of $G(z)$ is called the j-th **column degree** of $G(z)$ and we denote it by δ_j.

We define the **degree** δ of a convolutional code $\mathcal{C}$ as the maximum of the degrees of the determinants of the $k \times k$ sub-matrices of any generator matrix of $\mathcal{C}$. Then we say that $\mathcal{C}$ is an (n, k, δ) convolutional code [9].

The high order coefficient matrix of $G(z)$, G_∞, is the matrix whose j-th column is formed by the coefficients of z^{δ_j} in the j-th column of $G(z)$.

We say that a code $\mathcal{C}$ is **observable** (see, e.g., [13], [10]) if the generator matrix $G(z)$ has a polynomial left inverse and $G(0)$ is full rank. If $\mathcal{C}$ is an observable code then it can be equivalently described through a **parity check matrix**. In other words, there exists in this case an $(n-k) \times n$ full rank polynomial matrix $H(z)$ such that

$$\mathcal{C} = \left\{ \boldsymbol{v}(z) \in \mathbb{F}[z]^n \,|\, H(z)\boldsymbol{v}(z) = \boldsymbol{0} \in \mathbb{F}[z]^{n-k} \right\}.$$

If we write $\boldsymbol{v}(z) = \mathbf{v}_0 + \mathbf{v}_1 z + \ldots + \mathbf{v}_l z^l$ (with $l \geq 0$) and we represent $H(z)$ as a matrix polynomial

$$H(z) = H_0 + H_1 z + \cdots + H_\nu z^\nu$$

we can expand the kernel representation in the following way

$$\begin{bmatrix} H_0 & & & & & \\ \vdots & \ddots & & & & \\ H_\nu & \cdots & H_0 & & & \\ & \ddots & & \ddots & & \\ & & H_\nu & \cdots & H_0 & \\ & & & \ddots & \vdots & \\ & & & & H_\nu & \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_l \end{bmatrix} = \mathbf{0}. \tag{1}$$

An important distance measure for convolutional codes is the **free distance**:

$$d_{\text{free}}(\mathcal{C}) := \min \left\{ \mathrm{wt}(\boldsymbol{v}(z)) \,|\, \boldsymbol{v}(z) \in \mathcal{C},\, \boldsymbol{v}(z) \neq \boldsymbol{0} \right\}.$$

Rosenthal and Smarandache [12] showed that an (n, k, δ) convolutional code has a free distance upper bounded by

$$d_{\text{free}}(\mathcal{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) + \delta + 1. \tag{2}$$

This bound is known as the **generalized Singleton bound** [12] since it generalizes in a natural way the Singleton bound for block codes. Moreover, an (n, k, δ) code is defined to be a **maximum distance separable** convolutional code (MDS) [12] if its free distance equals the generalized Singleton bound.

A more local distance measure, is the **column distance** [8], $d_j^c(\mathcal{C})$, given by the expression

$$d_j^c(\mathcal{C}) = \min \left\{ \mathrm{wt}(\mathbf{v}_{[0,j]}(z)) \,|\, \boldsymbol{v}(z) \in \mathcal{C},\, \mathbf{v}_0 \neq \mathbf{0} \right\}$$

where $\mathbf{v}_{[0,j]}(z) = \mathbf{v}_0 + \mathbf{v}_1 z + \cdots + \mathbf{v}_j z^j$ represents the j-th truncation of the codeword $\boldsymbol{v}(z) \in \mathcal{C}$. It is related with the $d_{\text{free}}(\mathcal{C})$ in the following way

$$d_{\text{free}}(\mathcal{C}) = \lim_{j \to \infty} d_j^c(\mathcal{C}). \tag{3}$$

The j-th column distance is upper bounded [4], [7]

$$d_j^c(\mathcal{C}) \leq (n-k)(j+1) + 1 \tag{4}$$

and the maximality of any of the column distances implies the maximality of all the previous ones, that is, if $d_j^c(\mathcal{C}) = (n-k)(j+1)+1$ for some j, then $d_i^c(\mathcal{C}) = (n-k)(i+1)+1$ for $i \leq j$ (see [4], [7]). The $(m+1)$-tuple $(d_0^c(\mathcal{C}), d_1^c(\mathcal{C}), \ldots, d_m^c(\mathcal{C}))$ is called the **column distance profile** of the code [8].

Since no column distance can achieve a value greater than the generalized Singleton bound, the largest integer j for which that bound (4) can be attained is $j = L$,

$$L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor. \tag{5}$$

An (n, k, δ) convolutional code $\mathcal{C}$ is said to have **maximum distance profile** (MDP) if $d_L^c(\mathcal{C}) = (n-k)(L+1)+1$ (see [4], [7]). In this case, every $d_j^c(\mathcal{C})$ for $j \leq L$ is maximal, so we can say that the column distances of MDP codes increase as rapidly as possible for as long as possible.

The following theorem characterizes MDP codes algebraically. Assume that the parity check matrix is given as $H(z) = \sum_{i=0}^{\nu} H_i z^i$. For each $j > \nu$ let $H_j = 0$ and define:

$$\mathcal{H}_j = \begin{bmatrix} H_0 & & & \\ H_1 & H_0 & & \\ \vdots & \vdots & \ddots & \\ H_j & H_{j-1} & \cdots & H_0 \end{bmatrix} \in \mathbb{F}^{(j+1)(n-k) \times (j+1)n}. \tag{6}$$

Let $G(z) = \sum_{i=0}^{m} G_i z^i$ and let $G_j = O$, $j > m$. We define

$$\mathcal{G}_j = \begin{bmatrix} G_0 & G_1 & \cdots & G_j \\ & G_0 & \cdots & G_{j-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix}. \tag{7}$$

Then MDP convolutional codes are characterized as following:

Theorem 2.1 ([4, Th. 2.4]): Let $\mathcal{G}_j$ and $\mathcal{H}_j$ be like in (7) and (6). Then the following are equivalent:

(a) $d_j^c = (n-k)(j+1)+1$;

(b) every $(j+1)k \times (j+1)k$ full-size minor of $\mathcal{G}_j$ formed from the columns with indices $1 \leq t_1 < \cdots < t_{(j+1)k}$, where $t_{sk+1} > sn$ for $s = 1, 2, \ldots, j$, is nonzero;

(c) every $(j+1)(n-k) \times (j+1)(n-k)$ full-size minor of $\mathcal{H}_j$ formed from the columns with indices $1 \leq r_1 < \cdots < r_{(j+1)(n-k)}$, where $r_{s(n-k)} \leq sn$ for $s = 1, 2, \ldots, j$, is nonzero.

A code satisfying the conditions of Theorem 2.1 is said to have the *MDP property*. MDP convolutional codes are similar to MDS block codes within windows of size $(L+1)n$.

III. Decoding over an erasure channel

Let us suppose that we use an MDP convolutional code $\mathcal{C}$ to transmit over an erasure channel. Then we can state the following result.

Theorem 3.1: Let $\mathcal{C}$ be an (n,k,δ) convolutional code with $d^c_{j_0}$ the j_0-th column distance. If in any sliding window of length $(j_0+1)n$ at most $d^c_{j_0}-1$ erasures occur, then we can completely recover the transmited sequence.

Proof: Assume that we have been able to correctly decode up to an instant $t-1$. Then we have the following homogeneous system:

$$\begin{bmatrix} H_\nu & H_{\nu-1} & \dots & H_0 & & & \\ & H_\nu & \dots & H_1 & H_0 & & \\ & & \ddots & & & \ddots & \\ & & & H_{j_0} & H_{j_0-1} & \dots & H_0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t-\nu} \\ \vdots \\ \mathbf{v}_{t-1} \\ \star \\ \star \\ \vdots \\ \star \end{bmatrix} = 0 \qquad (8)$$

where $\star$ takes the place of a vector that had some of the components erased. Let the positions of the erased field elements be $i_1,\dots,i_e$, $e \le (L+1)(n-k)1$, where $i_1,\dots,i_s$, $s \le n$, are the erasures occurring in the first n-vector erased. We can take the columns of the matrix in equation (8) that correspond to the coefficients of the erased elements to be the coefficients of a new system. The rest of the columns in (8) will help us to compute the independent terms. In this way we get a nonhomogeneous system with $(L+1)n$ equations and e, at most $(L+1)(n-k)$, variables.

We claim that there is an extension

$$\{\tilde{\mathbf{v}}_t,\dots,\tilde{\mathbf{v}}_{t+L}\}$$

such that the vector

$$(\mathbf{v}_{t-\nu},\dots,\mathbf{v}_{t-1},\tilde{\mathbf{v}}_t,\dots,\tilde{\mathbf{v}}_{t+L})$$

is a codeword and such that $\tilde{\mathbf{v}}_t$ is unique.

Indeed, we know that a solution of the system exists since we assumed that only erasures occur. To prove the uniqueness of $\tilde{\mathbf{v}}_t$, or equivalently, of the erased elements $\tilde{v}_{i_1},\dots,\tilde{v}_{i_s}$, let us suppose there exist two such good extensions $\{\tilde{\mathbf{v}}_t,\dots,\tilde{\mathbf{v}}_{t+L}\}$ and $\{\tilde{\tilde{\mathbf{v}}}_t,\dots,\tilde{\tilde{\mathbf{v}}}_{t+L}\}$. Let $\mathbf{h}_{i_1},\dots,\mathbf{h}_{i_e}$, be the column vectors of the sliding parity-check matrix in (8) which correspond to the erasure elements. We have:

$$\tilde{v}_{i_1}\mathbf{h}_{i_1}+\dots+\tilde{v}_{i_s}\mathbf{h}_{i_s}+\dots+\tilde{v}_{i_e}\mathbf{h}_{i_e}=\tilde{\mathbf{b}}$$

and

$$\tilde{\tilde{v}}_{i_1}\mathbf{h}_{i_1}+\dots+\tilde{\tilde{v}}_{i_s}\mathbf{h}_{i_s}+\dots+\tilde{\tilde{v}}_{i_e}\mathbf{h}_{i_e}=\tilde{\tilde{\mathbf{b}}},$$

where the vectors $\tilde{\mathbf{b}}$ and $\tilde{\tilde{\mathbf{b}}}$ correspond to the known part of the system. Subtracting these equations and observing that $\tilde{\mathbf{b}}=\tilde{\tilde{\mathbf{b}}}$, we obtain:

$$(\tilde{v}_{i_1}-\tilde{\tilde{v}}_{i_1})\mathbf{h}_{i_1}+\dots+(\tilde{v}_{i_e}-\tilde{\tilde{v}}_{i_e})\mathbf{h}_{i_e}=0.$$

Using Theorem 2.1 part (c) we obtain that, necessarily,

$$\tilde{v}_{i_1}-\tilde{\tilde{v}}_{i_1}=0,\ \dots,\ \tilde{v}_{i_s}-\tilde{\tilde{v}}_{i_s}=0.$$

In order to find the value of this unique vector, we solve the full column rank system, find a solution and retain the part which is unique. Then we slide n bits to the next $n(L+1)$ window and proceed as above. This concludes the proof of our claim. ∎

As an immediate result we have that the best scenario happens when the convolutional code is an MDP convolutional code.

Corollary 3.2: Let $\mathcal{C}$ be an (n,k,δ) MDP convolutional code. If in any sliding window of length $(L+1)n$ at most $(L+1)(n-k)$ erasures occur in a transmited sequence then we can completely recover the sequence in polynomial time in δ.

One must notice that although in Corollary 3.2 we fix the value L, other window sizes can be taken during the process in order to optimize it. The parameter L gives an upper bound on the length of the window that we can take to correct. For every $j \le L$, in a window of size $(j+1)n$ we can recover at most $(j+1)(n-k)$ erasures. This means that we can conveniently choose the size of the window at each step depending on the distribution of the erasures in the sequence. This is an advantage that these codes have over block codes. If we receive a part of sequence with a few errors we do not need to wait until we receive the complete block, we can already decode within very small windows.

This property allows us to recover the erasures in situations where the MDS block codes cannot do it. The following example illustrates this scenario.

Example 3.3: Let us take a $(2,1,50)$ MDP convolutional code to decode over an erasure channel. In this case the decoding can be completed if in any sliding window of length 202 there are not more than 101 erasures; 50% of the erasures can be recovered.

The MDS block code which achieves a comparable performance is a $[202,101]$ MDS block code. In a block of 202 symbols we can recover 101 erasures, that is again 50%.

Assume that we use a $(2,1,50)$ MDP convolutional code and a $[202,101]$ MDS block code to transmit over an erasure channel. Suppose we have been able to correctly decode up to an instant t and then we receive the following pattern of erasures

$$\dots\mathbf{vv}|\overbrace{\star\star\dots\star\star}^{(A)60}\overbrace{\mathbf{vv}\dots\mathbf{v}}^{(B)80}\overbrace{\star\star\dots\star\star}^{(C)60}\mathbf{vv}|\mathbf{vv}\dots,$$

where each $\star$ stands for a component of the vector that has been erased and $\mathbf{v}$ means that the component has been correctly received. In this situation 120 erasures happen in a block of 202 symbols and the MDS block code is not able to recover them. In the block code situation one has to skip the whole window losing that information, and go on with the decoding of the next block.

However, the MDP convolutional code can deal with this situation. Let us frame a 120 symbol length window; in this window we can correct up to 60 erasures. In this way we can recover the first block of erasures.

$$\overbrace{\mathbf{vv}\dots\mathbf{vv}}^{100}|\overbrace{\star\star\dots\star\star}^{(A)60}\overbrace{\mathbf{vv}\dots\mathbf{v}}^{(B)60}$$

Then we can slide through the received sequence and frame the rest of the erasures in the same way.

$$\overbrace{\mathbf{v}\mathbf{v}\ldots\mathbf{v}\mathbf{v}}^{(A+B)100}\overbrace{\star\star\ldots\star\star}^{(C)60}\overbrace{\mathbf{v}\mathbf{v}|\mathbf{v}\mathbf{v}\ldots}^{60}$$

After this we have correctly decoded the sequence. □

Remark 3.4: There are situations in which the pattern of erasures that is mentioned in Theorem 3.1 or in Corollary 3.2 does not happen nor we recover choosing smaller window sizes. Then we get lost in the recovering process.

When using the parity check matrix we know that

$$\begin{bmatrix} H_\nu & H_{\nu-1} & \cdots & H_0 & & & \\ & H_\nu & \cdots & H_1 & H_0 & & \\ & & \ddots & & & \ddots & \\ & & & H_j & H_{j-1} & \cdots & H_0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t-\nu} \\ \vdots \\ \mathbf{v}_t \\ \mathbf{v}_{t+1} \\ \vdots \\ \mathbf{v}_{t+j} \end{bmatrix} = \mathbf{0}.$$

Then, in order to continue our recovering process once we get lost, we need to find a block of νn symbols without erasures $(\mathbf{v}_{t-\nu},\ldots,\mathbf{v}_{t-1})$ preceding a block of $(j+1)n$ symbols $(\mathbf{v}_t,\ldots,\mathbf{v}_{t+j})$ where not more that $(j+1)(n-k)$ erasures occur. In other words, we need to have *clean memory*. □

IV. The backward process and the reverse-MDP convolutional codes

In this section we introduce a new class of codes called reverse-MDP convolutional codes which have the MDP property both forward and backward. This forward and backward flexibility will help us to solve situations for which an MDP convolutional code fails.

In Remark 3.4 we gave necessary conditions to restart recovering once we get lost, i.e., once the number of erasures is too large we need to find a window of clean or "almost" clean memory. However, finding a whole block of clean memory can end up in a long waiting time, together with a loss of a long part of the sequence until a νn block of clean symbols is found. The following example illustrates this situation.

Example 4.1: As previously, assume we use a $(2,1,50)$ MDP convolutional code to transmit over an erasure channel and we are able to recover the sequence up to an instant t.

Suppose then that we receive a part of a sequence with the following pattern

$$\ldots\star\star\overbrace{\star\ldots\star}^{(A)22}\overbrace{\mathbf{v}\mathbf{v}\star\star\mathbf{v}\mathbf{v}\star\star\ldots\mathbf{v}\mathbf{v}\star\star}^{(B)180}|\overbrace{\mathbf{v}\mathbf{v}\ldots\mathbf{v}\mathbf{v}}^{(C)202}|$$

$$|\overbrace{\star\star\ldots\star}^{(D)80}\overbrace{\mathbf{v}\mathbf{v}\ldots\mathbf{v}}^{(E)62}\overbrace{\star\star\ldots\star}^{(F)60}|\overbrace{\mathbf{v}\mathbf{v}\ldots\mathbf{v}}^{(G)202}.$$

In this case we cannot recover the sequence because we either do not have enough clean memory in between the blocks of erasures (we need 100 clean symbols) or, when we have clean memory, we have patterns that surpass the number of erasures allowed per window. Therefore, the previous algorithm would skip over these erasures and lose the information. A $[202,101]$ MDS block code would not be better either since in each block of 202 symbols there are more than 101 erasures. □

In the previous example we saw that, even if sometimes we can find enough clean memory between the bursts of erasures, we cannot decode moving left-to-right along the sequence. However, one can notice that in the places where this clean memory appears we could move right-to-left and we would obtain that the erasures are less accumulated in that block. So reading the patterns right-to-left would provide us with a distribution of erasures with an appropriate density per window to be recovered. So moving along the sequence in the inverse direction, we find a more favorable situation for the recovering and, this way, we would lose less information.

From now on we will refer to this inverted recovering process as *backward decoding* and to the normal left-to-right process as *forward decoding*.

We will show how this forward and backward flexibility of convolutional codes allow to recover patterns of erasures that block codes cannot recover. In order to do so we recall the following results.

Proposition 4.2 ([6, Prop. 2.9]): Let $\mathcal{C}$ be an (n,k,δ)-code with minimal generator matrix $G(z)$. Let $\overline{G}(z)$ be the matrix obtained by replacing each entry $g_{ij}(z)$ of $G(z)$ by $\overline{g_{ij}}(z) := z^{\delta_j} g_{ij}(z^{-1})$, where δ_j is the j-th column degree of $G(z)$. Then, $\overline{G}(z)$ is a minimal generator matrix of an (n,k,δ)-code $\overline{\mathcal{C}}$, having the characterization

$$\mathbf{v}_0+\mathbf{v}_1 z+\cdots+\mathbf{v}_{s-1}z^{s-1}+\mathbf{v}_s z^s \in \mathcal{C}$$

if and only if

$$\mathbf{v}_s+\mathbf{v}_{s-1} z+\cdots+\mathbf{v}_1 z^{s-1}+\mathbf{v}_0 z^s \in \overline{\mathcal{C}}.$$

We call $\overline{\mathcal{C}}$ the **reverse code** of $\mathcal{C}$. Similarly, we denote by $\overline{H}(z)=\sum_{i=0}^{\nu}\overline{H}_i z^i$ the parity check matrix of $\overline{\mathcal{C}}$. The reason for introducing $\overline{\mathcal{C}}$ is that it allows us to invert the time of our sequence allowing us to describe the backward process we are looking for.

$\mathcal{C}$ and $\overline{\mathcal{C}}$ have the same d_{free}. However, their column distances may be different since the truncations of the code words $\boldsymbol{v}(z)=\sum_{i=0}^{s}\mathbf{v}_i z^i$ and $\overline{\boldsymbol{v}}(z)=\sum_{i=0}^{s}\mathbf{v}_{s-i}z^i$ do not involve the same coefficients:

$$\begin{aligned} d_j^c(\mathcal{C}) &= \min\left\{wt(\mathbf{v}_{[0,j]}(z)) \,|\, \boldsymbol{v}(z)\in\mathcal{C},\, \mathbf{v}_0\neq\mathbf{0}\right\} \\ &= \min\left\{\sum_{i=0}^{j} wt(\mathbf{v}_i) \,|\, \boldsymbol{v}(z)\in\mathcal{C},\, \mathbf{v}_0\neq\mathbf{0}\right\} \\ d_j^c(\overline{\mathcal{C}}) &= \min\left\{wt(\overline{\mathbf{v}}_{[0,j]}(z)) \,|\, \overline{\boldsymbol{v}}(z)\in\overline{\mathcal{C}},\, \overline{\mathbf{v}}_0\neq\mathbf{0}\right\} \\ &= \min\left\{\sum_{i=0}^{s-j+1} wt(\mathbf{v}_{s-i}) \,|\, \boldsymbol{v}(z)\in\mathcal{C},\, \mathbf{v}_s\neq\mathbf{0}\right\}. \end{aligned}$$

Similar to the forward decoding process, in order to achieve maximum recovering capability when recovering using backward decoding we need the column distances of $\overline{\mathcal{C}}$ to be maximal up to a point. This leads to the following definition.

Definition 4.3: Let $\mathcal{C}$ be an MDP (n,k,δ) convolutional code. We say that $\mathcal{C}$ is a reverse-MDP convolutional code if the reverse code $\overline{\mathcal{C}}$ of $\mathcal{C}$ is an MDP code as well.

As previously explained, reverse-MDP convolutional codes are better candidates than MDP convolutional codes for recovering over the erasure channel. The following

theorem shows that the existence of this class of codes is guaranteed over fields with enough number of elements.

Theorem 4.4: Let k, n and δ be positive integers. An (n, k, δ) reverse-MDP convolutional code exists over a sufficiently large field.

Here we present a sketch of the proof. For further details we refer the reader to [15].

SKETCH OF PROOF: It was shown [7] that the condition for a convolutional code to be MDP code is an open condition, in other words, MDP convolutional codes form a generic set, that is, a nonempty Zariski open subset in the quasi-projective variety of all (n, k, δ) convolutional codes.

The intersection of two such sets is still a nonempty Zariski open set. A reverse-MDP convolutional code is an element of the intersection of the set given by the conditions on $\mathcal{C}$ to be MDP and the set given by the conditions on $\overline{\mathcal{C}}$ to be MDP. This intersection is still a nonempty Zariski open set, i.e., a generic set. Thus, reverse-MDP convolutional codes exist over large enough fields since they form a generic set. ■

Using reverse-MDP convolutional codes one can solve the situation in example 4.1.

Example 4.1b. Assume that the $(2, 1, 50)$ code we use to transmit in Example 4.1 is a reverse-MDP convolutional code. The reverse code $\overline{\mathcal{C}}$ has the same recovering capability as $\mathcal{C}$.

We are not able to recover the sequence with the left-to-right process and the mentioned pattern, but we can do it recovering backward.

Once we have received 100 symbols of C we can recover part of the past erasures. If we take the following window

$$\overbrace{\mathbf{vv}\star\star\mathbf{vv}\star\star\ldots\mathbf{vv}\star\star}^{(B)180}\,|\,\overbrace{\mathbf{vv}\ldots\mathbf{v}}^{(C)100}$$

and we use the reverse code $\overline{\mathcal{C}}$ to solve the inverted system, then we can recover the erasures in B. Framing the following window

$$\overbrace{\mathbf{vv}\ldots\mathbf{v}}^{(E)60}\overbrace{\star\star\ldots\star}^{(F)60}|\overbrace{\mathbf{vv}\ldots\mathbf{v}}^{(G)100}$$

we can in the same way recover block F. Like this we recovered 150 erasures, this is more than 59% of the erasures that happened in that concrete part of the sequence. □

In the previous example we showed how reverse-MDP convolutional codes and the backward process make possible to recover information that would already be considered as lost by an MDS block code, because it is happening in a previous block, or by an MDP code, because it cannot recover previous erasures. We use a space of clean memory, not only to recover the next burst of erasures, but additionally to recover the previous one. We can do this as soon as we received enough clean symbols and we do not need to wait until we receive a whole new block.

If we would allow this backward process to be complete, that is, to go from the end of the sequence up to the beginning, we would recover much more information. We do not consider this situation since it would imply that we need to wait until the whole sequence was received in order to start recovering right-to-left, and this would not give better results than the retransmission of lost packets.

V. CONSTRUCTION OF REVERSE-MDP CONVOLUTIONAL CODES

As we showed previously, reverse-MDP convolutional codes exist over sufficiently large fields and they give a good performance when decoding over the erasure channel. In this section we construct reverse-MDP codes for the case when $(n-k) \mid \delta$ and $k > \delta$ —when computing the parity check matrix— (or $k \mid \delta$ and $(n-k) > \delta$ —when constructing the generator matrix—).

For this construction we need to define a special type of matrices.

Definition 5.1: Let A be an $r \times r$ lower triangular Toeplitz matrix

$$A = \begin{bmatrix} a_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_r & \cdots & a_1 & a_0 \end{bmatrix}.$$

Let $s \in \{1, 2, \ldots, r\}$. Suppose that $I = \{i_1, \ldots, i_s\}$ is a set of row indices of A, $J = \{j_1, \ldots, j_s\}$ is a set of column indices of A, and that the elements of each set are ordered increasingly. We denote by A_J^I the submatrix of A with columns indexed by J and rows indexed by I. A submatrix of A is said to be **proper** if, for each $t \in \{1, 2, \ldots, s\}$, the inequality $j_t \leq i_t$ holds. The matrix A is said to be **superregular** if every proper submatrix of A has a nonzero determinant.

Definition 5.2: We say a superregular matrix A is **reverse-superregular** if the matrix

$$A_{\text{rev}} = \begin{bmatrix} a_r & 0 & \cdots & 0 \\ a_{r-1} & a_r & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_0 & \cdots & a_{r-1} & a_r \end{bmatrix}$$

is superregular, too.

Example 5.3: Let $\mathbb{F} = \mathbb{F}_8$ with $\alpha^3 + \alpha^2 + 1 = 0$. The matrices

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha^4 & 1 & 0 & 0 \\ \alpha^6 & \alpha^4 & 1 & 0 \\ \alpha^3 & \alpha^6 & \alpha^4 & 1 \end{bmatrix},\ C_{\text{rev}} = \begin{bmatrix} \alpha^3 & 0 & 0 & 0 \\ \alpha^6 & \alpha^3 & 0 & 0 \\ \alpha^4 & \alpha^6 & \alpha^3 & 0 \\ 1 & \alpha^4 & \alpha^6 & \alpha^3 \end{bmatrix}$$

are superregular. Therefore, C is a reverse-superregular matrix. □

If we let $(n-k) \mid \delta$ and $k > \delta$ and we extract appropriately certain columns and rows from a reverse-superregular matrix, we can obtain the parity check matrix of a reverse-MDP code $\mathcal{C}$, i.e., $\mathcal{C}$ and $\overline{\mathcal{C}}$ satisfying the MDP property.

Theorem 5.4: Let A be an $r \times r$ reverse-superregular matrix with $r = (L+1)(2n-k-1)$. For $j = 0, 1, \ldots, L$ let I_j be sets of row indices

$$I_j = \{(j+1)n + j(n-k-1),\ (j+1)n + j(n-k-1) + 1, \ldots, (j+1)(2n-k-1)\}$$

and J_j the sets of column indices

$$J_j = \{jn+j(n-k-1)+1, \\ jn+j(n-k-1)+2, \ldots, (j+1)n+j(n-k-1)\}$$

and let I and J be the union of these sets

$$I = \bigcup_{j=0}^{L} I_j, \qquad J = \bigcup_{j=0}^{L} J_j.$$

Let $\widetilde{A}$ be the $(L+1)(n-k)\times(L+1)n$ lower block triangular submatrix obtained from A as $\widetilde{A} = A_J^I$. Then every $(L+1)(n-k)\times(L+1)(n-k)$ full size minor of $\widetilde{A}$ formed from the columns with indices $1 \leq i_1 < \cdots < i_{(L+1)(n-k)}$, where $i_{s(n-k)} \leq sn$ for $s = 1, 2, \ldots, L$, is nonzero. Moreover, the same property holds for $\widetilde{A}_{rev}$.

The condition $(n-k) \mid \delta$ and $k > \delta$ ensures that $L = \nu = \frac{\delta}{(n-k)}$. In this way $H_\nu = H_L$ and all the matrices of the expansion of $H(z)$ appear in $\mathcal{H}_L$ and we can describe $H(z)$.

Let μ_j be the maximum degree of all polynomials in the j-th row of $H(z)$ and let H_∞ be the matrix whose j-th row is formed by the coefficients of z^{μ_j} in the j-th row of $H(z)$. One can note that in general $H_\infty \neq H_\nu$, but since $(n-k) \mid \delta$, H_ν has full rank then both matrices coincide. $\overline{H}_i = H_{\nu-i}$ for $i = 0, \ldots, \nu$ and the expression for the parity check matrix of $\overline{\mathcal{C}}$ is $\overline{H}(z) = H_\nu + H_{\nu-1}z + \cdots + H_1 z^{\nu-1} + H_0 z^\nu$.

In this way, the blocks of $\widetilde{A}$ are the appropriate matrices H_i and we can construct $H(z)$ and $\overline{H}(z)$. We illustrate the process with the following example.

Example 5.5: We can construct the parity check matrix of a $(3, 2, 1)$ reverse-MDP convolutional code $\mathcal{C}$ over $\mathbb{F}_{32}$ using a 6×6 reverse-superregular matrix. Assume $\mu^5+\mu^2+1 = 0$ and we have the reverse-superregular matrix

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \mu^{15} & 1 & 0 & 0 & 0 & 0 \\ \mu^{21} & \mu^{15} & 1 & 0 & 0 & 0 \\ \mu^{23} & \mu^{21} & \mu^{15} & 1 & 0 & 0 \\ \mu^{21} & \mu^{23} & \mu^{21} & \mu^{15} & 1 & 0 \\ \mu^{10} & \mu^{21} & \mu^{23} & \mu^{21} & \mu^{15} & 1 \end{bmatrix}.$$

Applying Theorem 5.4 we can extract the corresponding blocks and obtain the matrix

$$\mathcal{H}_L = \begin{bmatrix} H_0 & O \\ H_1 & H_0 \end{bmatrix} = \begin{bmatrix} \mu^{21} & \mu^{15} & 1 & 0 & 0 & 0 \\ \mu^{10} & \mu^{21} & \mu^{23} & \mu^{21} & \mu^{15} & 1 \end{bmatrix}$$

so the parity check matrix of our code is

$$H(z) = \begin{bmatrix} \mu^{21}+\mu^{10}z & \mu^{15}+\mu^{21}z & 1+\mu^{23}z \end{bmatrix}.$$

The expression for

$$\overline{H}(z) = \sum_{i=0}^{1} \overline{H}_i z^i = \sum_{i=0}^{1} H_{1-i} z^i$$

is now

$$\overline{H}(z) = \begin{bmatrix} \mu^{10}+\mu^{21}z & \mu^{21}+\mu^{15}z & \mu^{23}+z \end{bmatrix},$$

where

$$\overline{H}_L = \begin{bmatrix} \mu^{10} & \mu^{21} & \mu^{23} & 0 & 0 & 0 \\ \mu^{21} & \mu^{15} & 1 & \mu^{10} & \mu^{21} & \mu^{23} \end{bmatrix}.$$

□

The same kind of construction can be applied in order to obtain the generator matrix of a code. Transposing the reverse-superregular matrix and changing the sizes for the extraction of rows and columns we obtain the following theorem similar to Theorem 5.4.

Theorem 5.6: Let B be the transpose of an $r \times r$ reverse-superregular matrix with $r = (L+1)(n+k-1)$. For $j = 0, 1, \ldots, L$, let I_j be sets of row indices

$$I_j = \{jn+j(k-1)+1, \\ jn+j(k-1)+2, \ldots, (j+1)n+j(k-1)\}$$

and J_j the sets of column indices

$$J_j = \{(j+1)n+j(k-1), \\ (j+1)n+j(k-1)+1, \ldots, (j+1)(n+vk-1)\}$$

and let I and J be the union of these sets

$$I = \bigcup_{j=0}^{L} I_j, \qquad J = \bigcup_{j=0}^{L} J_j.$$

Let $\widetilde{B}$ be the $(L+1)n \times (L+1)k$ upper block triangular submatrix obtained from B as $\widetilde{B} = B_J^I$. Then every $(L+1)k\times(L+1)k$ full size minor of $\widetilde{B}$ formed from the columns with indices $1 \leq i_1 < \cdots < i_{(L+1)k}$, where $i_{sk+1} > sn$ for $s = 1, 2, \ldots, L$, is nonzero. Moreover, the same property holds for $\widetilde{B}_{rev}$.

In this case we take $k \mid \delta$ and $(n-k) > \delta$ so that $L = m = \frac{\delta}{k}$. Then all the matrices in the expansion of matrix $G(z)$ appear in $\mathcal{G}_L$. Like this we can completely define $G(z)$. As in the parity check matrix case, in general $G_\infty \neq G_m$, but with $k \mid \delta$, G_m has full rank and $G_\infty = G_m$. Now $\overline{G}_i = G_{m-i}$ for $i = 0, \ldots, m$, and the generator matrix of $\overline{\mathcal{C}}$ looks like $\overline{G}(z) = G_m + G_{m-1}z + \cdots + G_1 z^{m-1} + G_0 z^m$.

Extracting the blocks from $\widetilde{B}$ we can construct the generator matrix of the code as shown in the example below.

Example 5.7: We can construct the generator matrix of a $(3, 1, 1)$ code over $\mathbb{F}_{32}$. For this we use the transpose of a 6×6 reverse-superregular matrix. Let $\gamma^5+\gamma^4+\gamma^3+\gamma^2+1 = 0$. We can apply Theorem 5.6 to the matrix

$$S = \begin{bmatrix} 1 & \gamma^{19} & \gamma^{16} & \gamma^{20} & \gamma^5 & \gamma^{16} \\ 0 & 1 & \gamma^{19} & \gamma^{16} & \gamma^{20} & \gamma^5 \\ 0 & 0 & 1 & \gamma^{19} & \gamma^{16} & \gamma^{20} \\ 0 & 0 & 0 & 1 & \gamma^{19} & \gamma^{16} \\ 0 & 0 & 0 & 0 & 1 & \gamma^{19} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

to obtain

$$\mathcal{G}_L = \begin{bmatrix} G_0 & G_1 \\ O & G_0 \end{bmatrix} = \begin{bmatrix} \gamma^{16} & \gamma^{16} \\ \gamma^{19} & \gamma^5 \\ 1 & \gamma^{20} \\ 0 & \gamma^{16} \\ 0 & \gamma^{19} \\ 0 & 1 \end{bmatrix}.$$

The generator matrices of $\mathcal{C}$ and $\overline{\mathcal{C}}$ are

$$G(z) = \begin{bmatrix} \gamma^{16}+\gamma^{16}z \\ \gamma^{19}+\gamma^5 z \\ 1+\gamma^{20}z \end{bmatrix}, \quad \overline{G}(z) = \begin{bmatrix} \gamma^{16}+\gamma^{16}z \\ \gamma^5+\gamma^{19}z \\ \gamma^{20}+z \end{bmatrix}.$$

□

VI. Conclusions

In this paper, we introduce reverse-MDP convolutional codes as an alternative to MDS block codes when decoding over an erasure channel. We have seen that the MDS behavior of the MDP codes withing windows of size $(j+1)n$, doubled by the reverse MDP capacity of reverse-MDP convolutional codes, lets us recover a larger number of erasures in a sequence dealing with situations that for MDS block codes are impossible. Even over large field sizes the complexity of decoding is polynomial for a fixed window size since the decoding algorithm requires only the solving of some linear systems. Moreover, the sliding window property allows us to adapt the decoding process to the distribution of the erasures in the sequence. We have shown how the possibility of taking smaller windows lets us recover erasures that MDS block codes cannot recover.

We prove the existence of reverse-MDP convolutional codes and give a particular construction. These codes give a better performance than MDP codes when working over the erasure channel.

References

[1] M. Arai, A. Yamamoto, A. Yamaguchi, S. Fukumoto, and K. Iwasaki. Analysis of using convolutional codes to recover packet losses over burst erasure channels. In *PRDC '01: Proceedings of the 2001 Pacific Rim International Symposium on Dependable Computing*, page 258, Washington, DC, USA, 2001. IEEE Computer Society.

[2] M. A. Epstein. Algebraic decoding for a binary erasure vhannel. Technical Report 340, Massachusetts Institute of Technology, March 1958. Reprinted from the 1958 IRE National Convention Record, Part 4.

[3] S. Fashandi, S.O. Gharan, and A.K. Khandani. Coding over an erasure channel with a large alphabet size. In *IEEE ISIT '08: Proceedings of the 2008 IEEE International Symposium on Information Theory*, pages 1053 –1057, 2008.

[4] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache. Strongly MDS convolutional codes. *IEEE Trans. Inform. Theory*, 52(2):584–598, 2006.

[5] H. Gluesing-Luerssen and F.-L. Tsang. A matrix ring description for cyclic convolutional codes. *Adv. Math. Commun.*, 2(1):55–81, 2008.

[6] R. Hutchinson. The existence of strongly MDS convolutional codes. *SIAM J. Control Optim.*, 47(6):2812–2826, 2008.

[7] R. Hutchinson, J. Rosenthal, and R. Smarandache. Convolutional codes with maximum distance profile. *Systems & Control Letters*, 54(1):53–63, 2005.

[8] R. Johannesson and K. Sh. Zigangirov. *Fundamentals of Convolutional Coding*. IEEE Press, New York, 1999.

[9] R. J. McEliece. The algebraic theory of convolutional codes. In V. Pless and W.C. Huffman, editors, *Handbook of Coding Theory*, volume 1, pages 1065–1138. Elsevier Science Publishers, Amsterdam, The Netherlands, 1998.

[10] J. Rosenthal. Connections between linear systems and convolutional codes. In B. Marcus and J. Rosenthal, editors, *Codes, Systems and Graphical Models*, IMA Vol. 123, pages 39–66. Springer-Verlag, 2001.

[11] J. Rosenthal, J. M. Schumacher, and E. V. York. On behaviors and convolutional codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1881–1891, 1996.

[12] J. Rosenthal and R. Smarandache. Maximum distance separable convolutional codes. *Appl. Algebra Engrg. Comm. Comput.*, 10(1):15–32, 1999.

[13] J. Rosenthal and E. V. York. BCH convolutional codes. *IEEE Trans. Inform. Theory*, 45(6):1833–1844, 1999.

[14] V. Tomás, J. Rosenthal, and R. Smarandache. Decoding of MDP Convolutional Codes over the Erasure Channel. In *Proceedings of the 2009 IEEE International Symposium on Information Theory*, pages 556–560, Seoul, South Korea, 2009.

[15] V. Tomás, J. Rosenthal, and R. Smarandache. Convolutional Codes Decoding over the Erasure Channel. In preparation.