Coding for Combined Block–Symbol Error Correction

Ron M. Roth[*]
Computer Science Department
Technion, Haifa 32000, Israel
Email: ronny@cs.technion.ac.il

Pascal O. Vontobel
Hewlett–Packard Laboratories
Palo Alto, CA 94304, USA
Email: pascal.vontobel@ieee.org

***Abstract*—We design low-complexity error correction coding schemes for channels that introduce different types of errors and erasures: on the one hand, the proposed schemes can successfully deal with symbol errors and erasures, and, on the other hand, they can also successfully handle phased burst errors and erasures.**

I. Introduction

Many data transmission and storage systems suffer from different types of errors at the same time. For example, in some data storage systems the state of a memory cell might be altered by an alpha particle that hits this memory cell. On the other hand, an entire block of memory cells might become unreliable because of hardware wear-out. Such data transmission and storage systems can be modeled by channels that introduce symbol errors and block (*i.e.*, phased burst) errors, where block errors encompass several contiguous symbols. Moreover, if some side information is available, say based on previously observed erroneous behavior of a single or of multiple memory cells, this can be modeled as symbol erasures and block erasures.

In this paper, we design novel error correction coding schemes that can deal with both symbol and block errors and both symbol and block erasures for a setup as in Figure 1. Every small square corresponds to a symbol in $F = \mathrm{GF}(q)$, where q is an arbitrary prime power. (In applications, q is typically a small power of 2.) All small squares are arranged in the shape of an $m \times n$ rectangular array. We say that a *symbol error* happens if the content of a small square is altered. We say that a *block error* happens if one or several small squares in a column of the array are altered. Similarly, we say that a *symbol erasure* happens if the content of a small square is erased and we say that a *block erasure* happens if all small squares in a column of the array are erased.

We can correct such errors and erasures by imposing that the symbols in such an array constitute a codeword in some suitably chosen code $\mathbb{C}$ of length mn over F. The two main ingredients of the code $\mathbb{C}$ that is proposed in this paper are, on the one hand, a matrix H_{in} of size $m \times (mn)$ over F, and, on the other hand, a code $\mathcal{C}$ of length n over F. Namely, an array forms a codeword in $\mathbb{C}$ if and only if every row of the array is a codeword in $\mathcal{C}$ once the n columns have been transformed by n different bijective mappings $F^m \to F^m$ derived from the matrix H_{in}. The resulting error-correcting coding scheme has the following salient aspects:

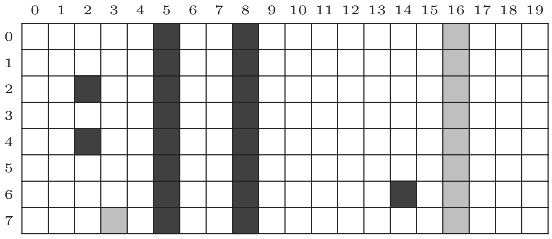

Fig. 1. Array of size $m \times n$ with symbol block errors and erasures. Here, $m = 8$, $n = 20$, and there are symbol errors at positions $(2,2)$, $(4,2)$, and $(6,14)$, a symbol erasure at position $(7,3)$, block errors in columns 5 and 8, and a block erasure in column 16.

- It can be seen as a concatenated coding scheme, however with two distinctive features. First, multiple inner codes are used (one for every column encoding), and, second, all these inner codes have rate one (*i.e.*, the encoders of these inner codes can be considered to be scramblers).
- One can identify a range of code parameters for $\mathbb{C}$ for which (to the best of our knowledge) the resulting redundancy improves upon the best known.
- One can devise efficient decoders for combinations of symbol and block errors and erasures most relevant in practical applications. In particular, these decoders are more efficient than a corresponding decoder for a suitably chosen generalized Reed–Solomon (GRS) code of length mn over F, assuming such a GRS code exists in the first place. (Finding efficient decoders for the general case is still an open problem.)

A. Paper Overview

The paper starts in Section II by considering a simplified version of the above error and erasure scenario and of the above-mentioned code construction. Namely, in this section we consider only block errors and erasures, *i.e.*, no symbol errors or erasures. Moreover, an $m \times n$ array forms a codeword if and only if every row is a codeword in some code $\mathcal{C}$ of length n (*i.e.*, there are no bijective mappings applied to the columns); in other words, the array code considered is simply an m-level interleaving of $\mathcal{C}$. Our main purpose of Section II is laying out

[*] This work was done in part while visiting Hewlett–Packard Laboratories, Palo Alto, CA 94304, USA. This work was supported in part by Grants 1280/08 and 1092/12 from the Israel Science Foundation.

some of the ideas and tools that will be used in subsequent parts of the paper; in particular, it is shown how one can take advantage of the *rank* of the error array in order to increase the correction capability of the array code.

We then move on to Section III, which is the heart of the paper and which gives all the details of the construction of the code $\mathbb{C}$, and presents an outline for an efficient decoding algorithm for it.

Due to space limitations, many details and proofs are omitted from this abstract. They can be found in the full version of this work [15].

B. Related Work

The idea of exploiting the rank of the error array when decoding interleaved codes was presented by Metzner and Kapturowski in [14] and by Haslach and Vinck in [9], [10]. Therein, the code $\mathcal{C}$ is chosen to be a linear $[n,k,d]$ code over F, and, clearly, any combination of block errors can be corrected as long as their number does not exceed $(d-1)/2$. In [14] and [9], it was further assumed that the set of nonzero columns in the (additive) $m \times n$ error array E over F is linearly independent over F; namely, the rank of E (as a matrix over F) equals the number of block errors. It was then shown that under this additional assumption, it is possible to correct (efficiently) any pattern of up to $d-2$ block errors. Essentially, the linear independence allows to easily locate the nonzero columns in E, and from that point onward, the problem reduces to that of erasure decoding. A generalization to the case where the nonzero columns in E are not necessarily full-rank was discussed in [10]; we will recall the latter result in more detail in Section II-A.

The case where the constituent code $\mathcal{C}$ is a GRS code has been studied in quite a few papers, primarily in the context where the contents of each block error is assumed to be uniformly drawn from F^m. In [2], Bleichenbacher *et al.* identified a threshold, $(m/(m+1))(d-1)$, on the number of block errors, below which the decoding failure probability approaches 0 as d goes to infinity and n/q goes to 0. A better bound on the decoding error probability was obtained by Kurzweil *et al.* [13] and Schmidt *et al.* [16]. See also [4], [5], [11], [12], and [17].

Turning to the main coding problem studied in this paper—namely, handling combinations of symbol errors and block errors—a general solution was given by Zinov'ev [18] and Zinov'ev and Zyablov [19], using concatenated codes and their generalizations.

Recently, Blaum *et al.* [1] have proposed new erasure-correcting codes for combined block–symbol error patterns. The advantage of their scheme is having the smallest possible redundancy (equaling the largest total number of symbols that can be erased) and an efficient *erasure* decoding algorithm. However, the parameters of their constructions are rather strongly limited: first, the array size is typically much smaller than q (and, in one application, must in fact be smaller than $\log_2 q$), and, secondly, verifying whether the construction actually works for given parameters becomes intractable, unless the number of block erasures or the number of symbol erasures is very small.

In [7], Gabrys *et al.* presented a coding scheme which is targeted mainly at applications for flash memories. In their setting, an erroneous column may have at most a prescribed number ℓ of symbol errors; and in addition to limiting the total number of erroneous columns, a further restriction is assumed on the number of columns with at most a prescribed number ℓ' $(< \ell)$ of symbol errors.

C. Notation

For integers a and b with $0 \le a < b$, we denote by $\langle a, b \rangle$ the set of integers $\{a, a+1, a+2, \ldots, b-1\}$, and $\langle b \rangle$ will be used as a shorthand notation for $\langle 0, b \rangle$. Entries of vectors will be indexed starting at 0, and so will be the rows and columns of matrices. For a vector $\boldsymbol{u} \in F^n$ and a subset $W \subseteq \langle n \rangle$, we let $(\boldsymbol{u})_W$ be the sub-vector (in $F^{|W|}$) of $\boldsymbol{u}$ that is indexed by W. The support of $\boldsymbol{u}$ will be denoted by $\mathsf{supp}(\boldsymbol{u})$. We extend these definitions to any $m \times n$ matrix E over F, with $(E)_W$ denoting the $m \times |W|$ sub-matrix of E that is formed by the columns that are indexed by W. Column j of E will be denoted by E_j, and $\mathsf{supp}(E)$ will stand for the column support of E, namely, the set of indexes j for which $E_j \neq \mathbf{0}$. The linear subspace of F^m that is spanned by the columns of E will be denoted by $\mathsf{colspan}(E)$.

With any $m \times n$ matrix $E = (e_{h,j})_{h \in \langle m \rangle, j \in \langle n \rangle}$ over F, we associate the bivariate polynomial

$$E(y,x) = \sum_{h \in \langle m \rangle, j \in \langle n \rangle} e_{h,j} y^h x^j$$

(namely, the powers of y index the rows and the powers of x index the columns). With each column j of E we associate the univariate polynomial $E_j(y) = \sum_{h \in \langle m \rangle} e_{h,j} y^h$; thus, $E(y,x) = \sum_{j \in \langle n \rangle} E_j(y) x^j$.

II. Simplified Code Construction

In this section we consider the simplified scenario mentioned in Section I-A. Namely, we consider only block errors and erasures, *i.e.*, no symbol errors or erasures. Moreover, an $m \times n$ array forms a codeword of length mn if and only if every row is a codeword in some code $\mathcal{C}$ with parameters $[n,k,d]$ (*i.e.*, there are no bijective mappings applied to the columns); equivalently, the array code considered is simply an m-level interleaving of $\mathcal{C}$. If $\mathcal{C}$ is specified by an $(n-k) \times n$ parity-check matrix H, then the syndrome matrix S is defined to be the $m \times (n-k)$ matrix $S = YH^\mathsf{T}$, where the $m \times n$ matrix

$$Y = \Gamma + E$$

over F represents the read out (or received) message, where the $m \times n$ matrix Γ over F represents the stored (or transmitted) codeword, and where the $m \times n$ matrix E over F represents the alterations that happen to Γ over time (or during transmission). Note that our formalism treats erasures like errors, with the side information $K \subseteq \langle n \rangle$ telling us their location.

The subsections of this section are structured as follows. In Section II-A we study the error correction capabilities of the interleaved array code, where $\mathcal{C}$ is any linear $[n, k, d]$ code over F. Then, in Section II-B, we present an efficient decoder for the special case where $\mathcal{C}$ is a GRS code. (All the details, along with an application of this decoder for the probabilistic decoding of the array code under the assumption that the block errors are uniformly distributed over F^m, can be found in [15].)

A. Block Errors and Erasures with Rank Constraints

We start with Theorem 1 below that generalizes the results of [14] and [9] to the case where the set of nonzero columns of the $m \times n$ error array E are not necessarily linearly independent. Note that this theorem was already stated (without proof) in the one-page abstract [10] for the error-only case (*i.e.*, no block erasures).

Theorem 1 *Let $\mathcal{C}$ be a linear $[n, k, d]$ code over F and let H be an $(n{-}k) \times n$ parity-check matrix of $\mathcal{C}$ over F. Fix K to be a subset of $\langle n \rangle$ of size r. Given any $m \times (n{-}k)$ (syndrome) matrix S over F, there exists at most one $m \times n$ matrix E over F that has the following properties:*

(i) $S = EH^{\mathsf{T}}$ *, and—*

(ii) *writing $\overline{K} = \langle n \rangle \setminus K$, the values $t = |\mathsf{supp}\,((E)_{\overline{K}})|$ and $\mu = \mathsf{rank}\big((E)_{\overline{K}}\big)$ satisfy*

$$2t + r \leq d + \mu - 2 \; . \qquad (1)$$

■

(Compare with the classical case where the rank μ is ignored: Eq. (1) should then be replaced by $2t + r \leq d - 1$.)

Given a matrix S and assuming that $|\mathsf{supp}(E)| - \mathsf{rank}(E)$ takes on a small value, there exists an efficient algorithm for finding E with the properties stated in Theorem 1. However, when the difference $|\mathsf{supp}(E)| - \mathsf{rank}(E)$ becomes large, we do not know how to find E efficiently, even if $\mathcal{C}$ can be decoded at no computational cost. Nevertheless, when the code $\mathcal{C}$ is suitable chosen, we can formulate an efficient decoder, as shown in the next subsection.

B. The GRS Case

Figure 2 presents an efficient decoder for finding the error matrix E under the conditions of Theorem 1, for the special case where $\mathcal{C}$ is an $[n, k, d{=}n{-}k{+}1]$ GRS code $\mathcal{C}_{\mathrm{GRS}}$ over F with a parity-check matrix $H_{\mathrm{GRS}} = \big(\alpha_j^i\big)_{i \in \langle d-1 \rangle, j \in \langle n \rangle}$, where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ are distinct nonzero elements of F. Note that the decoding algorithm in Figure 2 applies also to the more general class of alternant codes over F, with d now standing for the designed minimum distance of the code. The proof of correctness of the algorithm can be found in [15].

III. Main Code Construction

We come now to the main code construction of this paper, namely the code construction $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ that was outlined in Section I. Section III-A gives all the details of the channel model and the code construction; Section III-B presents the correction capabilities of the code; Section III-C discusses a variety of examples based on specific choices for the code $\mathcal{C}$ and the matrix H_{in}; and, finally, Section III-D contains an outline of a decoding algorithm for $\mathbb{C}$, for the case where $\mathcal{C}$ is a GRS code.

Input:

- Array Y of size $m \times n$ over F.
- Set K of size r of indexes of column erasures.

Steps:

1) Compute the $m \times (d{-}1)$ syndrome array
$$S = Y H_{\mathrm{GRS}}^{\mathsf{T}} \; .$$

2) Compute the modified syndrome array to be the unique $m \times (d{-}1)$ matrix σ that satisfies the congruence:
$$\sigma(y, x) \equiv S(y, x)\, \mathrm{M}(x) \pmod{x^{d-1}} \; ,$$
where
$$\mathrm{M}(x) = \prod_{j \in K} (1 - \alpha_j x) \; .$$
Let μ be the rank of the $m \times (d{-}1{-}r)$ matrix $\tilde{S}$ formed by the columns of σ that are indexed by $\langle r, d{-}1 \rangle$.

3) Using the Feng–Tzeng algorithm [6], compute a polynomial $\lambda(x)$ of (smallest) degree $\Delta \leq (d{+}\mu{-}r)/2$ such that the following congruence is satisfied for some polynomial $\omega(y, x)$ with $\deg_x \omega(y, x) < r + \Delta$:
$$\sigma(y, x)\lambda(x) \equiv \omega(y, x) \pmod{x^{d-1}} \; .$$
If no such $\lambda(x)$ exists or the computed $\lambda(x)$ does not divide $\prod_{j \in \langle n \rangle} (1 - \alpha_j x)$ then declare decoding failure and **Stop**.

4) Compute the $m \times n$ error array E by
$$E_j(y) = \begin{cases} \frac{-\alpha_j \cdot \omega(y, \alpha_j^{-1})}{\lambda'(\alpha_j^{-1}) \cdot \mathrm{M}(\alpha_j^{-1})} & \text{if } \lambda(\alpha_j^{-1}) = 0 \\ \frac{-\alpha_j \cdot \omega(y, \alpha_j^{-1})}{\lambda(\alpha_j^{-1}) \cdot \mathrm{M}'(\alpha_j^{-1})} & \text{if } j \in K \\ 0 & \text{otherwise} \end{cases} \; ,$$
where $(\cdot)'$ denotes formal differentiation.

Output:

- Decoded array $Y - E$ of size $m \times n$.

Fig. 2. Decoding of interleaved GRS codes. (See Section II-B.)

A. Channel Model and Code Definition

We consider the following channel model. An $m \times n$ array Γ over F is stored (or transmitted), and Γ is subject to the following error and erasure types (see Figure 1):

(T1) *Block errors:* a subset of columns in Γ that are indexed by $\mathcal{J} \subseteq \langle n \rangle$ can be erroneous.

(T2) *Block erasures:* a subset of columns in Γ that are indexed by $\mathcal{K} \subseteq \langle n \rangle \setminus \mathcal{J}$ can be erased.

(T3) *Symbol errors:* a subset of entries in Γ that are indexed by $\mathcal{L} \subseteq \langle m \rangle \times (\langle n \rangle \setminus (\mathcal{K} \cup \mathcal{J}))$ can be erroneous.

(T4) *Symbol erasures:* a subset of entries in Γ that are indexed by $\mathcal{R} \subseteq \big(\langle m \rangle \times (\langle n \rangle \setminus \mathcal{K})\big) \setminus \mathcal{L}$ can be erased.

Let the $m \times n$ matrix $\mathcal{E}$ over F represent the alterations that happen to Γ over time (or during transmission). Then the read out (or received) message is given by the $m \times n$ matrix

$$\Upsilon = \Gamma + \mathcal{E} \qquad (2)$$

over F. The sets $\mathcal{K}$ are $\mathcal{R}$ known to the decoder.

Write $\tau = |\mathcal{J}|$, $\rho = |\mathcal{K}|$, $\vartheta = |\mathcal{L}|$, and $\varrho = |\mathcal{R}|$. The total number of symbol errors (resulting from error types (T1) and (T3)) is at most $m\tau + \vartheta$ and the total number of symbol erasures (resulting from erasure types (T2) and (T4)) is at most $m\rho + \varrho$; hence, we should be able to correct all error and erasure types (T1)–(T4) (when occurring simultaneously) while using a code of length mn over F with minimum distance at least $m(2\tau + \rho) + 2\vartheta + \varrho + 1$. However, such a strategy does not take into account the fact that errors of type (T1)–(T2) are aligned across the m rows of the $m \times n$ array Γ. The next construction is designed to take advantage of such an alignment.

Definition 2 *Let $\mathcal{C}$ be a linear $[n, k, d]$ code over F, and let H_{in} be an $m \times (mn)$ matrix over F that satisfies the following two properties for some positive integer δ:*

(a) *H_{in} is a parity-check matrix of a linear code over F of length mn and minimum distance at least δ, and—*

(b) *writing*

$$H_{\mathrm{in}} = \left(\; H_0 \; \middle| \; H_1 \; \middle| \; \ldots \; \middle| \; H_{n-1} \; \right) ,$$

with $H_0, H_1, \ldots, H_{n-1}$ being $m \times m$ sub-matrices of H_{in}, each H_j is invertible over F.

Given $\mathcal{C}$ and H_{in}, we define $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ to be the linear $[mn, mk]$ code over F which consists of all $m \times n$ matrices

$$\Gamma = \left(\; \Gamma_0 \; \middle| \; \Gamma_1 \; \middle| \; \ldots \; \middle| \; \Gamma_{n-1} \; \right)$$

over F (where Γ_j stands for column j of Γ) such that each row in

$$Z = \left(\; H_0\Gamma_0 \; \middle| \; H_1\Gamma_1 \; \middle| \; \ldots \; \middle| \; H_{n-1}\Gamma_{n-1} \; \right)$$

is a codeword of $\mathcal{C}$.

One can view the code $\mathbb{C}$ as a (generalized) concatenated code, where the outer code is an m-level interleaving of $\mathcal{C}$, such that an $m \times n$ matrix $Z = \left(\; Z_0 \; \middle| \; Z_1 \; \middle| \; \ldots \; \middle| \; Z_{n-1} \; \right)$ over F is an outer codeword if and only if each row in Z belongs to $\mathcal{C}$. Each column in Z then undergoes encoding by an inner encoder of rate one, where the encoder of column j is given by the bijective mapping $Z_j \mapsto H_j^{-1} Z_j$.

B. Error Correction Capabilities

This subsection discusses what combinations of errors and erasures of the types (T1)–(T4) can be handled by the code $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ that was specified in Definition 2.

Theorem 3 *There exists a decoder for the code $\mathbb{C}$ that correctly recovers the transmitted array in the presence of errors of types (T1)–(T4) (which may occur simultaneously), whenever τ $(= |\mathcal{J}|)$, ρ $(= |\mathcal{K}|)$, ϑ $(= |\mathcal{L}|)$, and ϱ $(= |\mathcal{R}|)$ satisfy*

$$2\tau + \rho \leq d - 2 \; ,$$
$$2\vartheta + \varrho \leq \delta - 1 \; .$$

■

The proof of Theorem 3 (which we omit) is based on analyzing the rank of the following matrix

$$E = \left(\, H_0\mathcal{E}_0 \, \middle| \, H_1\mathcal{E}_1 \, \middle| \, \ldots \, \middle| \, H_{n-1}\mathcal{E}_{n-1} \, \right) , \qquad (3)$$

which is obtained by left-multiplying the columns of the error array $\mathcal{E}$ in (2) by the $m \times m$ matrices H_j. These multiplications transform the symbol errors and erasures (of type (T3) and (T4)) into column vectors that potentially increase the rank of the error array, allowing us to reduce to a setting akin to that of Theorem 1.

Remark 4 We draw the attention of the reader to the condition on τ and ρ in Theorem 3, namely, that the expression $2\tau + \rho$ be at most $d - 2$, rather than the (more common) requirement that it be at most $d - 1$. It is this slightly stronger condition that, implicitly, provides the required redundancy for correcting the (additional) symbol errors and erasures. □

C. Examples

In the following examples, we look at special choices for $\mathcal{C}$ and H_{in}, and demonstrate some of the properties of the resulting code $\mathbb{C}$.

Example 5 Consider the special case where $mn \leq q + 1$. Here, we can take $\mathcal{C}$ to be an MDS code (e.g., a GRS code) over F and H_{in} to be a parity-check matrix of an MDS code over F. Under such circumstances we have $d = n - k + 1$ and $\delta = m + 1$, which means that it suffices that the sizes τ, ρ, ϑ, and ϱ satisfy

$$2\tau + \rho \leq n - k - 1 \quad \text{and} \quad 2\vartheta + \varrho \leq m \; .$$

The redundancy of $\mathbb{C}$, being $m(n - k)$, is then the smallest possible for this correction capability: since the total number of symbol errors is $m\tau + \vartheta$ and the total number of symbol erasures is $m\rho + \varrho$, then, by the Reiger bound, we need a redundancy of at least $m(2\tau + \rho) + 2\vartheta + \varrho$ symbols over F in order to be able to correct all error types (T1)–(T4). Admittedly, the same performance of correction capability versus redundancy can be achieved also by a single linear $[mn, mk]$ MDS code C over F (which exists under the assumption that $mn \leq q+1$). However, as pointed out earlier, the use of such a code C does not take into account the alignment of error types (T1) and (T2) across the rows of the received $m \times n$ array. It is this alignment that allows $\mathbb{C}$ to achieve the same correction capability using a code $\mathcal{C}$, which is m times shorter than C. While we still need for $\mathbb{C}$ the parity-check matrix H_{in} of an MDS code of length mn, the redundancy of the latter code needs to be only m, rather than $m(n - k)$. □

Example 6 Given τ, ϑ, and $F = \mathrm{GF}(q)$ such that $0 < \tau \leq q/2 - 1$, we assume that $(2\tau + 2 \leq)\; n \leq q$ and we take H_{in} to be a parity-check matrix of a (possibly extended) shortened BCH code of length mn over F, where m is determined by ϑ, n, and q to satisfy the equality

$$m = 1 + \lceil (1 - (1/q)) \cdot (2\vartheta - 1) \rceil \cdot \lceil \log_q (mn) \rceil$$

(so mn may be larger than q). The code $\mathcal{C}$ is taken as a (possibly extended) $[n, k, d]$ GRS code over F where $d = 2\tau + 2$. The overall redundancy of $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ can be verified to be

$$2\tau m + 1 + \lceil (1 - (1/q)) \cdot (2\vartheta - 1) \rceil \cdot \lceil \log_q(mn) \rceil \ . \qquad (4)$$

The first term, $2\tau m$, is the smallest redundancy possible if one is to correct any τ block errors of length m. The second term therein is the redundancy (or an upper bound thereof) of a BCH code that corrects any ϑ symbol errors over F. In comparison, a shortened BCH code of length mn over F that corrects any $\tau m + \vartheta$ symbol errors may have redundancy as large as

$$1 + \left\lceil (1 - (1/q)) \cdot \big(2(\tau m + \vartheta) - 1\big) \right\rceil \cdot \lceil \log_q(mn) \rceil \ .$$

It can be verified that the latter expression is larger than (4) when $mn > q \geq 4$. □

In the full version [15], we demonstrate that in many cases, the redundancy of $\mathbb{C}$ is smaller even than that of the generalized concatenated (GC) code construction of [3], [18] and [19]. For example, given τ, ϑ, and $F = \mathrm{GF}(q)$ such that $n \leq q$, we show that the redundancy of the GC construction is at least

$$2\tau m + \Big[(2\vartheta + 1) \ln \vartheta + 2\gamma \cdot \vartheta + O(1) \Big] \ , \qquad (5)$$

where γ is Euler's constant (approximately 0.5772). The redundancy of $\mathbb{C}$ can therefore be made smaller than that of GC codes (with the same correction capabilities) whenever the bracketed term in (5) exceeds m.

D. Decoding Algorithm for the Main Code Construction

The next theorem states that when the constituent code $\mathcal{C}$ in $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ is a GRS code then, under certain conditions on the parameters of $\mathbb{C}$, all error pattern that satisfy the conditions of Theorem 3 can be efficiently decoded.

Theorem 7 *For $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ such that $\mathcal{C}$ is a GRS code over F, the decoder guaranteed by Theorem 3 can be implemented by a polynomial-time algorithm, whenever*

$$d - \rho - \varrho \geq 2\sqrt{\delta(n - \rho - \varrho)} - \delta \qquad (6)$$

(or, simply, whenever $d \geq 2\sqrt{\delta n} - \delta$ in case $\rho = \varrho = 0$). ■

To obtain the decoder claimed in Theorem 7, we look at the m-level interleaving of $\mathcal{C}$ as a GRS code over the extension field $\mathrm{GF}(q^m)$, and condition (6) then guarantees that such a code has a polynomial-time *list decoder* [8] that can correct any pattern containing $\rho + \varrho$ (column) erasures and up to $(d+\delta-\rho-\varrho-3)/2$ (column) errors. This, in turn, allows us to efficiently compute from the received array $\Upsilon = \Gamma + \mathcal{E}$, polynomially many candidates for the array E as in the left-hand side of (3). For each such E, we can then compute from (3) the respective error array $\mathcal{E}$ and select the one that satisfies the conditions of Theorem 3. The latter theorem guarantees that the solution will be unique. More details can be found in the full version [15].

We point out that while the above algorithm outline has made essential use of an efficient list decoder for the m-level interleaving of $\mathcal{C}$, nothing is assumed about H_{in} (other than the requirements in Definition 2(a)). In particular, nothing is assumed about the decoding complexity of the code with the parity-check matrix H_{in}.

In [15], we present another decoder for the case where $\mathcal{C}$ is a GRS code and H_{in} is a parity-check matrix of a GRS code, under some restrictions on the patterns of errors of type (T3). That decoder, which, *inter alia*, uses the algorithm in Figure 2 as a building block, is more efficient than the one outlined above, and is generally more efficient than the GRS decoder that we would need if we use a GRS code C to encode the whole $m \times n$ array (see Example 5).

REFERENCES

[1] M. Blaum, J.L. Hafner, and S. Hetzler, *Partial-MDS codes and their application to RAID type of architectures*, IBM Res. Rep. No. RJ10498 (February 2012).

[2] D. Bleichenbacher, A. Kiayas, and M. Yung, *Decoding interleaved Reed–Solomon codes over noisy channels, Theor. Comput. Sci.,* 379 (2007), 348–360.

[3] E.L. Blokh and V.V. Zyablov, *Coding of generalized concatenated codes, Probl. Inf. Transm.,* 10 (1974), 218–222.

[4] A. Brown, L. Minder, and A. Shokrollahi, *Probabilistic decoding of interleaved RS-codes of the Q-ary symmetric channel, Proc. IEEE Int. Symp. Inf. Theory,* Chicago, IL, 2004, pp. 326.

[5] D. Coppersmith and M. Sudan, *Reconstructing curves in three (and higher) dimensional space from noisy data, Proc. 35th Annual ACM Symp. Theory of Computing,* San Diego, CA, 2003, pp. 136-142.

[6] G.-L. Feng and K.K. Tzeng, *A generalization of the Berlekamp–Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes, IEEE Trans. Inf. Theory,* 37 (1991), 1274–1287.

[7] R. Gabrys, E. Yaakobi, and L. Dolecek, *Graded bit error-correcting codes with applications to flash memory, IEEE Trans. Inf. Theory,* 59 (2013), 2315–2327.

[8] V. Guruswami and M. Sudan, *Improved decoding of Reed–Solomon and algebraic–geometry codes, IEEE Trans. Inf. Theory,* 45 (1999), 1757–1767.

[9] C. Haslach and A.J.H. Vinck, *A decoding algorithm with restrictions for array codes, IEEE Trans. Inf. Theory,* 45 (1999), 2339–2344 (and correction in the same publication, 47 (2001), 479).

[10] C. Haslach and A.J.H. Vinck, *Efficient decoding of interleaved linear block codes, Proc. IEEE Int. Symp. Inf. Theory,* Sorrento, Italy, 2000, p. 149.

[11] J. Justesen, C. Thommesen, and T. Høholdt, *Decoding of concatenated codes with interleaved outer codes, Proc. IEEE Int. Symp. Inform. Theory,* Chicago, IL, 2004, p. 328.

[12] V.Yu. Krachkovsky, and Y.X. Lee, *Decoding of parallel Reed-Solomon codes with applications to product and concatenated codes, Proc. IEEE Int. Symp. Inf. Theory,* Cambridge, MA, 1998, p. 55.

[13] H. Kurzweil, M. Seidl, and J.B. Huber, *Reduced-complexity collaborative decoding of interleaved Reed–Solomon and Gabidulin codes, Proc. IEEE Int. Symp. Inf. Theory,* St. Petersburg, Russia, 2011, pp. 2557–2561.

[14] J.J. Metzner and E.J. Kapturowski, *A general decoding technique applicable to replicated file disagreement location and concatenated code decoding, IEEE Trans. Inf. Theory,* 36 (1990), 1274–1287.

[15] R.M. Roth and P.O. Vontobel, *Coding for combined block–symbol error correction, submitted to IEEE Trans. Inf. Theory,* 2013, online available at `http://arxiv.org/abs/1302.1931`.

[16] G. Schmidt, V.R. Sidorenko, and M. Bossert, *Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs, IEEE Trans. Inf. Theory,* 55 (2009), 2991–3011.

[17] A. Wachter–Zeh, A. Zeh, and M. Bossert, *Decoding interleaved Reed-Solomon codes beyond their joint error-correcting capability, Des. Codes Cryptogr.,* to appear.

[18] V.A. Zinov'ev, *Generalized concatenated codes for channels with error bursts and independent errors, Probl. Inf. Transm.,* 17 (1981), 254–260.

[19] V.A. Zinov'ev and V. Zyablov, *Correction of error bursts and independent errors using generalized cascade codes, Probl. Inf. Transm.,* 15 (1979), 125–134.