Correcting Grain-Errors in Magnetic Media

Ryan Gabrys*, Eitan Yaakobi†, and Lara Dolecek*

*University of California, Los Angeles
rgabrys@ucla.edu, dolecek@ee.ucla.edu

†California Institute of Technology
yaakobi@caltech.edu

***Abstract*—This paper studies new bounds and constructions that are applicable to the combinatorial granular channel model previously introduced by Sharov and Roth. The main theme of the paper is that codes capable of correcting grain-errors are related to codes that correct insertions/deletions and codes that correct asymmetric errors. Using this insight, new bounds on the maximum cardinality of a grain-error correcting code are derived and constructions of codes that correct grain-errors are considered. It is also demonstrated that permutations of the classical group codes can correct a single grain-error. In several cases of interest, our results improve upon the currently best known bounds and constructions.**

I. Introduction

Granular media is a promising magnetic recording technology [10]. However, the error models for the granular media channel are different than the classical binary symmetric channel. In [10], a channel model for grains potentially spanning several bits was considered. In this setup, the polarity of a grain is set by the last bit to be written into it. The errors manifest themselves as overwrites (or smears) where the last bit in the grain overwrites the previous values in the grain. Typically, the locations and bit lengths of the grains on the medium are unknown during the encoding and decoding phases. In this work, the focus is on grains of length at most two and so a *grain-error* is an error where the information from one bit overwrites the information stored in an adjacent bit. Moreover, it is not known to the encoder and decoder which grains are of length-2 and which are of length-1.

In [8], Sharov and Roth introduced a granular channel model, known as the *overlapping* grain-error model. The overlapping grain-error model encapsulates the earlier model, known as the *non-overlapping* grain-error model, previously introduced by Mazumdar, Barg, and Kashyap in [6]. In both works, bounds and constructions were considered. In [11], improved upper bounds were derived for the non-overlapping grain-error model using the techniques introduced in [5]. In [4], a related model was studied from an information-theoretic perspective.

The contribution of this work will be to provide bounds and improved constructions for the overlapping grain-error model. Our approach will be to leverage connections between codes that correct grain-errors and codes capable of correcting insertions/deletions and codes that correct asymmetric errors. Then, using some of the classical ideas from [7] and [9], new bounds and constructions will be given. The resulting codes have a similar structure as the codes that correct asymmetric errors in [3]. This connection is exploited in later sections.

The paper is organized as follows. In Section II, the main results of the paper are highlighted. In Section III, the channel model is formally defined and the notation used for the remainder of the paper is introduced. In Section IV, an upper bound on the size of a code that can correct up to t grain-errors is derived using an approach similar to that in [5]. In Section V, code constructions are presented. Section VI concludes the paper. Our codes offer an improvement in code size over earlier works [8], [6].

II. The Main Results

The main contributions of the paper are the following:

1) A non-asymptotic expression for the maximum cardinality of a t grain-error correcting code (that is, a code that can correct up to t grain-errors).
2) Constructions of the largest known grain-error correcting codes for many values of t and n where n is the length of the code.

We include the following table illustrating our contributions for the case where $t = 1$.

TABLE I
Upper and Lower Bounds for Single Grain-Error Correcting Codes

Length	Previous Lower Bound	Previous Upper Bound	Current Lower Bound	Current Upper Bound
3	4 [8]	4 [8]	4	4 [8]
4	6 [8]	6 [8]	6 [8]	6 [8]
5	8 [8]	8 [8]	8	8 [8]
6	16 [8]	16 [8]	16 [8]	16 [8]
7	26 [8]	26 [8]	26 [8]	26 [8]
8	44[8]	44 [8]	44 [8]	44 [8]
9	44 [8]	151[8]	64	114
10	64 [6]	279 [8]	104	205
11	128 [6]	507 [8]	188	373
12	256 [6]	943 [8]	344	683
13	512 [6]	1760 [8]	632	1261
14	1024 [6]	3256 [8]	1172	2341
15	2176 [8]	6149 [8]	2192	4369
16	4096 [6]	11533 [8]	4096	8192
17	4096 [6]	21654 [8]	7712	15420
18	8192[6]	41341 [8]	14592	29127
19	16384 [6]	77792 [8]	27596	55189
20	32768 [6]	147789 [8]	52432	104858

Remark 1. *For $t = 1$, there is no distinction between overlapping and non-overlapping grain-errors and the upper bound result for $t = 1$ in this paper (displayed in Table I) was derived concurrently in [11].*

We note that the previous lower bound and the current lower bound shown in Table I are *constructive* (e.g., the lower bound $\frac{2^n}{n}$ is only constructive when n is a power of 2, see first [6]). The lower bounds marked in red are the result of using the code construction proposed in Section V. The upper bounds shown in red are derived in Section IV.

III. Preliminaries

In this section, we describe in detail the structure of grain-errors. Afterwards, key notation that will be used for the remainder of the paper is introduced. This section is organized into subsections containing relevant definitions and preliminary results that are used in sections later in the paper.

A. Grain-errors

In this subsection, we formally introduce the definition of a grain-error and provide an elementary example. Some simple lemmas are derived that will be used later for counting the number of grain-errors. A grain-error can occur when two bits of data are being stored within the same grain. A grain-error then causes the two bits in the same grain to either both be 0 or both be 1; the error operation can be interpreted as a smearing. Following the setup of [6], we assume that the first bit smears the second. The problem of interest is how to correct grain-errors when the locations and lengths of the grains are unknown to the encoder and decoder.

Throughout the analysis, we note that the alphabet of interest is binary. Before continuing, we provide a formal definition of a t grain-error pattern $\epsilon_{t,n}$.

Definition 1. *Let $t \geq 1$ be an integer and let $\epsilon_{t,n} = (\epsilon_{1,t,n}, \ldots, \epsilon_{n,t,n})$ where $\epsilon_{t,n} \in GF(2)^n$. Then, $\epsilon_{t,n}$ is a t grain-error pattern for a vector $\boldsymbol{x} \in GF(2)^n$ if the following holds:*

1) *$wt(\epsilon_{t,n}) \leq t$ and $\epsilon_{1,t,n} = 0$,*
2) *For $2 \leq i \leq n$, if $\epsilon_{i,t,n} \neq 0$, then $x_i \neq x_{i-1}$.*

For shorthand we refer to the vector $\epsilon_{t,n} + \boldsymbol{x}$ as $\epsilon_{t,n}(\boldsymbol{x})$.

Definition 2. *A t grain-error correcting code is a code that can correct any t grain-error pattern.*

For a given vector $\boldsymbol{x}$, let $\mathcal{B}_{t,n}(\boldsymbol{x})$ denote the set of all possible vectors that can be received, assuming that $\boldsymbol{x}$ was transmitted and t grain-errors may occur. For notational convenience, we let $b_{t,n}(\boldsymbol{x}) = |\mathcal{B}_{t,n}(\boldsymbol{x})|$.

Definition 2 coincides with the *overlapping* grain-error model discussed in [8]. We briefly note that since the original model of *non-overlapping* grain-errors [6] is a special case of the more general overlapping grain-error model, many of the results (and in particular the code constructions) in this paper apply to both models.

Example 1. *Suppose $\boldsymbol{x} = (0\ 0\ 0\ 1\ 0)$ was transmitted. Then, the set $\mathcal{B}_{1,5}(\boldsymbol{x})$ is $\{(0\ 0\ 0\ 1\ 0), (0\ 0\ 0\ 1\ 1), (0\ 0\ 0\ 0\ 0)\}$.*

In Section IV, we derive upper bounds on the maximum number of codewords for grain-error correcting codes. Before continuing, we introduce some notation and basic results that are used in the remainder of the paper.

A *run* is a maximal substring of one or more consecutive identical symbols. We denote the number of runs in a vector $\boldsymbol{x}$ as $r(\boldsymbol{x})$. The following ideas will be used for deriving upper bounds in the next section.

Lemma 1. *For any vector $\boldsymbol{x}$, $b_{t,n}(\boldsymbol{x}) = \sum_{j=0}^{\min\{t, r(\boldsymbol{x})-1\}} \binom{r(\boldsymbol{x})-1}{j}$.*

Proof: Suppose a vector $\boldsymbol{x}$ was transmitted and that it consists of $v = r(\boldsymbol{x})$ runs. By Definition 1, a grain-error can occur only at the boundaries between runs. If there are exactly $v \geq t+1$ runs, there are $v-1$ transitions between runs and therefore $b_{t,n}(\boldsymbol{x}) = \sum_{j=0}^{t} \binom{v-1}{j}$. If there are t or fewer runs (i.e., $v \leq t$), then at most $\sum_{j=0}^{v-1} \binom{v-1}{j}$ places an error can occur. ■

The following Lemma is a consequence of the smearing effect of a grain-error.

Lemma 2. *For any two vectors $\boldsymbol{x}, \boldsymbol{y}$ where $\boldsymbol{y} \in B_{t,n}(\boldsymbol{x})$, $r(\boldsymbol{y}) \leq r(\boldsymbol{x})$ and $b_{t,n}(\boldsymbol{y}) \leq b_{t,n}(\boldsymbol{x})$.*

B. Tools for computing upper bounds

In this subsection, we briefly review some of the tools used in Section IV for computing a non-asymptotic upper bound on the cardinality of grain-error correcting codes. We begin by revisiting some of the notation and results from [5].

Definition 3. *A hypergraph $\mathcal{H}$ is a pair $(\mathcal{X}, \mathcal{E})$, where $\mathcal{X}$ is a finite set and $\mathcal{E}$ is a collection of nonempty subsets of $\mathcal{X}$ such that $\cup_{E \in \mathcal{E}} E = \mathcal{X}$. The elements of the set $\mathcal{E}$ are called hyperedges.*

Definition 4. *A matching of a hypergraph $\mathcal{H} = (\mathcal{X}, \mathcal{E})$ is a collection of disjoint hyperedges $E_1, \ldots, E_j \in \mathcal{E}$. The matching number of $\mathcal{H}$, denoted $\nu(\mathcal{H})$, is the largest j for which a matching exists.*

As will be described shortly, the following can be interpreted as the dual of the matching of a hypergraph.

Definition 5. *A transversal of a hypergraph $\mathcal{H} = (\mathcal{X}, \mathcal{E})$ is a subset $T \in \mathcal{X}$ that intersects every hyperedge in $\mathcal{E}$. The transversal number of $\mathcal{H}$, denoted by $\tau(\mathcal{H})$, is the smallest size of a transversal.*

Let $\mathcal{H}$ be a hypergraph with vertices $x_1, \ldots, x_n$ and hyperedges $E_1, \ldots, E_m$. The relationships contained within $\mathcal{H}$ can be interpreted through the matrix $A \in \{0,1\}^{n \times m}$, where

$$A(i,j) = \begin{cases} 1 & \text{if } x_i \in E_j, \\ 0 & \text{otherwise,} \end{cases}$$

for $1 \leq i \leq n, 1 \leq j \leq m$. Cast in this light, the matching number and the transversal number can be derived using linear optimization techniques.

Lemma 3. *(cf. [5]) The matching number and the transversal number are solutions of the integer linear programs:*

$$\nu(\mathcal{H}) = \max\{\mathbf{1^T} \boldsymbol{z} | A\boldsymbol{z} \leq \mathbf{1}, z_j \in \{0,1\}, 1 \leq j \leq m\}, \text{ and} \quad (1)$$

$$\tau(\mathcal{H}) = \min\{\mathbf{1^T} \boldsymbol{w} | A^T \boldsymbol{w} \geq \mathbf{1}, w_i \in \{0,1\}, 1 \leq i \leq n\}, \quad (2)$$

where $\mathbf{1}$ *denotes a column vector of all* 1s *of the appropriate dimension.*

Relaxing the condition that the solutions to the programming problem are comprised of 0s and 1s, we have the following problems:

$$\nu^*(\mathcal{H}) = \max\{\mathbf{1}^{\mathbf{T}}\boldsymbol{z} | A\boldsymbol{z} \leq \mathbf{1}, \boldsymbol{z} \geq 0\}, \text{ and} \quad (3)$$

$$\tau^*(\mathcal{H}) = \min\{\mathbf{1}^{\mathbf{T}}\boldsymbol{w} | A^T\boldsymbol{w} \geq \mathbf{1}, \boldsymbol{w} \geq 0\}. \quad (4)$$

Clearly $\nu(\mathcal{H}) \leq \nu^*(\mathcal{H})$ and $\tau(\mathcal{H}) \geq \tau^*(\mathcal{H})$. Since (3), (4) are linear programs they satisfy strong duality and $\nu^*(\mathcal{H}) = \tau^*(\mathcal{H})$. Thus, combining these inequalities leads us to $\nu(\mathcal{H}) \leq \tau^*(\mathcal{H})$ [5].

C. Distance metrics and group codes

In this subsection some distance metrics are introduced that will be used in Section V to construct grain-error correcting codes. In addition, we define group codes.

Definition 6. *Suppose* $\boldsymbol{x}, \boldsymbol{y} \in GF(2)^n$. *Their Hamming distance is denoted* $d_H(\boldsymbol{x}, \boldsymbol{y}) = |\{i : x_i \neq y_i\}|$.

Definition 7. *Suppose* $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{y} = (y_1, \ldots, y_n) \in GF(2)^n$. *For* $1 \leq i \leq n$, $N(\boldsymbol{x}, \boldsymbol{y}) = |\{i : x_i > y_i\}|$.

Definition 8. *Suppose* $\boldsymbol{x}, \boldsymbol{y}$ *are two vectors in* $GF(2)^n$. *Their asymmetric distance is denoted* $d_A(\boldsymbol{x}, \boldsymbol{y}) = \max\{N(\boldsymbol{x}, \boldsymbol{y}), N(\boldsymbol{y}, \boldsymbol{x})\}$.

The *minimum distance* of a code $\mathcal{C}$ is $d_{H_{\min}}$ if $d_{H_{\min}}$ is the smallest Hamming distance between any two distinct codewords in $\mathcal{C}$. Similarly we say the *minimum asymmetric distance* of a code $\mathcal{C}$ is $d_{A_{\min}}$ if $d_{A_{\min}}$ is the smallest asymmetric distance between any two distinct codewords in $\mathcal{C}$.

Suppose $\mathcal{G}$ is an Abelian group of size n and suppose $(g_1, \ldots, g_{n-1})$ is a sequence consisting of the non-zero elements of $\mathcal{G}$. We define a *group code* $\mathcal{C}_{\mathcal{G}}$ as the set of binary vectors $\boldsymbol{x} = (x_1, \ldots, x_{n-1}) \in GF(2)^{n-1}$ such that $\sum_{k=1}^{n-1} x_k g_k = 0$. Such a construction was shown in [1] to have $d_{A_{\min}} = d_{H_{\min}} = 2$. We assume, without loss of generality, that the Abelian groups referred to in this paper are additive and the group operation is addition.

IV. Upper Bounds on Grain-Error Codes

In this section, linear programming methods are used to produce a closed-form upper bound on the cardinality of a t grain-error correcting code. The approach will be analogous to that found in [5] where upper bounds were computed for the deletion channel.

The approach is the following. First, the vector space from which codewords are chosen, is projected onto a hypergraph. Then, an approximate solution to a matching problem is derived. For shorthand, the maximum size of a t grain-error correcting code of length-n will be referred to as $M(n, t)$.

Let $\mathcal{H}_{t,n}$ denote the hypergraph for a t grain-error correcting code. More formally, let

$$\mathcal{H}_{t,n} = (GF(2)^n, \{\mathcal{B}_{t,n}(\boldsymbol{x}) | \boldsymbol{x} \in GF(2)^n\}).$$

In this graph, the vertices represent candidate codewords and the hyperedges represent vectors that result when t or fewer grain-errors occur in any of the candidate codewords.

As in [5], the upper bound $\nu(\mathcal{H}_{t,n})$ will be derived by considering the dual problem defined in equation (4). Since the vector $\boldsymbol{w}$ in (4) should assign weights for every vertex in $\mathcal{H}_{t,n}$, we define it as $\boldsymbol{w} = (w(\boldsymbol{x}))_{\boldsymbol{x} \in GF(2)^n}$. The problem is to find

$$\tau^*(\mathcal{H}_{t,n}) = \min_{\boldsymbol{w}}\{\sum_{\boldsymbol{x} \in GF(2)^n} w(\boldsymbol{x})\}$$

subject to

$$\sum_{\boldsymbol{x} \in \mathcal{B}_{t,n}(\boldsymbol{y})} w(\boldsymbol{x}) \geq 1, \forall \boldsymbol{y} \in GF(2)^n \quad (5)$$

and $w(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x} \in GF(2)^n$.

We are now ready to state the main result of the section.

Theorem 1. *A* t *grain-error correcting code of length-*n *has at most* $2\sum_{r=0}^{n-1} \binom{n-1}{r} \frac{1}{\sum_{j=0}^{\min\{t,r\}} \binom{r}{j}}$ *codewords.*

Proof: In order to prove the result, we assign values $w(\boldsymbol{x})$ for $\boldsymbol{x} \in GF(2)^n$ such that the constraint in (5) is satisfied. Let $w(\boldsymbol{x}) = \frac{1}{b_{t,n}(\boldsymbol{x})}$ where $b_{t,n}(\boldsymbol{x})$ is computed as in Lemma 1. Note that

$$\sum_{\boldsymbol{x} \in \mathcal{B}_{t,n}(\boldsymbol{y})} w(\boldsymbol{x}) = \sum_{\boldsymbol{x} \in \mathcal{B}_{t,n}(\boldsymbol{y})} \frac{1}{b_{t,n}(\boldsymbol{x})} \geq \frac{b_{t,n}(\boldsymbol{y})}{b_{t,n}(\boldsymbol{y})} = 1.$$

The inequality holds since for every $\boldsymbol{x} \in \mathcal{B}_{t,n}(\boldsymbol{y})$, from Lemma 2, $b_{t,n}(\boldsymbol{x}) \leq b_{t,n}(\boldsymbol{y})$. The Theorem statement now follows from the expression $\sum_{\boldsymbol{x} \in GF(2)^n} w(\boldsymbol{x})$. Since the number of length-n vectors with r runs is $2\binom{n-1}{r-1}$ and $b_{t,n} = \sum_{j=0}^{\min\{t,r-1\}} \binom{r-1}{j}$ from Lemma 1, we have $M(n,t) \leq 2\sum_{r=1}^{n} \binom{n-1}{r-1} \frac{1}{\sum_{j=0}^{\min\{t,r-1\}} \binom{r-1}{j}}$, which, after reindexing the parameter r, is the statement in the theorem.

■

As a result of Theorem 1, we derive non-asymptotic bounds for $t = 1, 2$.

Lemma 4. *For* $n \geq 1$,

$$M(n, 1) \leq \frac{2^{n+1} - 2}{n},$$

and for $n \geq 17$,

$$M(n, 2) \leq \frac{2^{n+2}(2 + \frac{2}{n-6})}{n(n-3)} + 3n.$$

In general, it is difficult to compare our bounds to those in [8] since the bounds in [8] require solving for a ρ parameter where ρ is the largest integer satisfying $\sum_{r=1}^{\rho}\binom{n-1}{r-1}\sum_{j=0}^{\min(t,r)}\binom{r}{j}\leq 2^{n-1}$. The bounds for $t=1$ and small n were explicitly derived using the formula in [8] and for all values of $8<n\leq 20$ our bound is tighter (as can be partially seen from Table I). Our bounds also have the advantage of being explicit. Finally, we note that the same approach was used in [11] to compute upper bounds for the non-overlapping grain-error model.

Recall the simple construction in [6] that offered $M(n,1)\geq\frac{2^n}{n}$ when n is a power of 2. Thus, in the first part of the next section, the codes of interest should have cardinalities between $\frac{2^n}{n}$ and $\frac{2^{n+1}-2}{n}$. The main idea of next section will be to leverage group codes to construct grain-error correcting codes.

V. Grain-Error Correcting Codes

This section is divided into two subsections. In the first subsection, we consider a construction for single grain-error correcting codes. In the following subsection, the ideas are extended to correct multiple grain-errors.

Recall from Definition 1, if the vector $\boldsymbol{x}$ was transmitted and $\boldsymbol{x}$ experienced t or fewer grain-errors, we refer to the resulting vector (after t or less grain-errors) as $\epsilon_{t,n}(\boldsymbol{x})$. The grain-error pattern is denoted as $\epsilon_{t,n}$ where $\epsilon_{t,n}=\boldsymbol{x}+\epsilon_{t,n}(\boldsymbol{x})$.

A. Single Grain-Error Correcting Codes

We begin by proving some sufficient conditions for a code to correct a single grain-error. Then, a code construction that satisfies these conditions is given. The codes presented in this section provide the largest known cardinalities for nearly all code lengths.

In the following, we say that two vectors $\boldsymbol{x},\boldsymbol{y}$ are t-confusable if there exist two error patterns $\epsilon^1_{t,n},\epsilon^2_{t,n}$ such that $\epsilon^1_{t,n}(\boldsymbol{x})=\epsilon^2_{t,n}(\boldsymbol{y})$. Similarly, two vectors $\boldsymbol{x},\boldsymbol{y}$ are not t-confusable if for every $\epsilon^1_{t,n},\epsilon^2_{t,n}$, $\epsilon^1_{t,n}(\boldsymbol{x})\neq\epsilon^2_{t,n}(\boldsymbol{y})$.

The ideas of Lemma 5 are a generalization of the ideas used in the construction from [6] that corrects any number of grain-errors in a vector of length n.

Lemma 5. *For any two length-n vectors $\boldsymbol{x},\boldsymbol{y}$, suppose $\boldsymbol{x}+\boldsymbol{y}$ is a vector such that the following holds for some index $1\leq i<n$:*

1) *The vector $\boldsymbol{x}+\boldsymbol{y}$ contains two consecutive 1s starting at index i, and*
2) *$(x_i,x_{i+1})=(0,0),(y_i,y_{i+1})=(1,1)$ or $(x_i,x_{i+1})=(1,1),(y_i,y_{i+1})=(0,0)$.*

Then, for any $\epsilon^1_{t,n},\epsilon^2_{t,n}$ (where $t\geq 1$) $\epsilon^1_{t,n}(\boldsymbol{x})\neq\epsilon^2_{t,n}(\boldsymbol{y})$.

Proof: Since $\boldsymbol{x}$ and $\boldsymbol{y}$ differ at position $i+1$ then in order for $\epsilon^1_{t,n}(\boldsymbol{x})=\epsilon^2_{t,n}(\boldsymbol{y})$ an error must occur at position $i+1$ in either $\boldsymbol{x}$ or $\boldsymbol{y}$. However, a grain-error can never change the information at position $i+1$ in either $\boldsymbol{x}$ or $\boldsymbol{y}$ since both $\boldsymbol{x}$ and $\boldsymbol{y}$ store the same information in positions i and $i+1$ by condition 2) in the statement of the Lemma. ∎

Combining the previous lemma with the ideas underpinning the single grain-error correcting code in [6], the following lemma enumerates sufficient conditions for a code $\mathcal{C}$ to correct any single grain-error.

Lemma 6. *A code $\mathcal{C}$ can correct any single grain-error if for every pair of distinct codewords $\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}$ one of the following holds:*

1) *$d_H(\boldsymbol{x},\boldsymbol{y})=1$ and $x_1\neq y_1$.*
2) *$d_H(\boldsymbol{x},\boldsymbol{y})=d_A(\boldsymbol{x},\boldsymbol{y})=2$ where $\boldsymbol{x}+\boldsymbol{y}$ is a vector with a single run of two 1s (and possibly one or two runs of 0s).*
3) *$d_H(\boldsymbol{x},\boldsymbol{y})\geq 3$.*

We are now ready to state our code construction. For any group referred to in the subsequent discussion, the identity element will be denoted as 0 and will be referred to as the zero element.

Construction A. *Let $\mathcal{G}$ represent an Abelian group of size n. Suppose the sequence $\mathcal{S}=(g_1,g_2,\ldots,g_n)$ contains the elements of $\mathcal{G}$. The elements are ordered so that $g_1=0$ and for any $1<i\leq n$, the elements g_i and g_i^{-1}, if g_i^{-1} exists, are adjacent. Elements without an inverse can be placed arbitrarily in the ordering (as long as they do not come between an element and its inverse). The codewords of $\mathcal{C}$ are the vectors $\boldsymbol{x}\in GF(2)^n$ such that $\sum_{k=1}^{n}x_kg_k=0$.*

Theorem 2. *A code $\mathcal{C}$ created with Construction A can correct any single grain-error.*

Proof: We will show that $\mathcal{C}$ is a single grain-error correcting code by demonstrating that the conditions listed in Lemma 6 hold for any pair of distinct codewords $\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}$. Recall that a group code as defined in Section III-C has $d_{H_{\min}}=d_{A_{\min}}=2$. Let $\mathcal{C}_G$ be the group code created by using the same group as in $\mathcal{C}$ so that $\mathcal{C}_G$ has length $n-1$ and $\mathcal{C}$ has length n. The code $\mathcal{C}_G$ is obtained by shortening the codewords of $\mathcal{C}$ on the first bit.

Suppose $d_H(\boldsymbol{x},\boldsymbol{y})=1$. Then, since $d_{H_{\min}}=2$ for the group code $\mathcal{C}_{\mathcal{G}}$, it follows that if $d_H(\boldsymbol{x},\boldsymbol{y})=1$, then $\boldsymbol{x}$ and $\boldsymbol{y}$ must differ only in the first bit.

Suppose $d_H(\boldsymbol{x},\boldsymbol{y})=2$. Since $\mathcal{C}_G$ has $d_{H_{\min}}=2$, $\boldsymbol{x}$ and $\boldsymbol{y}$ do not differ in the first position. Furthermore, since $d_{A_{\min}}=2$ for $\mathcal{C}_G$, then there are two distinct indices i,j $(2\leq i,j\leq n)$ where $x_i\neq y_i$ and $x_j\neq y_j$. Since $d_{A_{\min}}=2$, then suppose, without loss of generality, that $N(\boldsymbol{x},\boldsymbol{y})=2$ so that $x_i=x_j=1$ and $y_i=y_j=0$. Therefore, $g_i+g_j=0$, or $g_j=g_i^{-1}$. However, by Construction A, we have $|j-i|=1$ and so $\boldsymbol{x}+\boldsymbol{y}$ is a vector with a single run of two 1s.

If $d_H(\boldsymbol{x},\boldsymbol{y})$ is not equal to 1 or 2 then $d_H(\boldsymbol{x},\boldsymbol{y})\geq 3$ and so by Lemma 6 the result holds. ∎

Note that since Construction A concatenates an arbitrary bit with a group code $\mathcal{C}_G$, it follows that if the underlying group code has cardinality $|\mathcal{C}_G|$, then a code $\mathcal{C}$ created using the previous construction has $|\mathcal{C}|$ codewords where $|\mathcal{C}|=2|\mathcal{C}_{\mathcal{G}}|\geq\frac{2^n}{n}$. From [7], if n is not a power of 2 then $|\mathcal{C}|>\frac{2^n}{n}$. The values for $|\mathcal{C}_G|$ are elegantly derived using discrete Fourier analysis in [7]. From [1], when the underlying group is the elementary Abelian group of size p^q, then a code $\mathcal{C}$ created from Construction A has the property $|\mathcal{C}| = \frac{2^{p^q} - 2^{p^{q-1}}}{p^q} + 2^{p^{q-1}}$. The results displayed in Table I are the result of using the same group codes found in ([1], Table I).

B. Correcting multiple grain-errors

In this subsection, grain-error correcting codes are constructed that correct more than a single error. Since the constructions in this subsection are extensions of the ideas from the previous subsection, the proofs are omitted. A more rigorous evaluation of the code cardinalities is left for an extended version of the paper ([2]). Generalizing the ideas from Lemma 6, the following Lemma characterizes a code capable of correcting t grain-errors.

Lemma 7. *A code $\mathcal{C}$ can correct any t grain-errors if the following holds for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, $\boldsymbol{x} \neq \boldsymbol{y}$:*

1) $d_H(\boldsymbol{x}, \boldsymbol{y}) \geq 2t + 1$,
2) $d_H(\boldsymbol{x}, \boldsymbol{y}) < 2t + 1$ *then either:*
   a) $x_1 \neq y_1$, *or*
   b) $\boldsymbol{x} + \boldsymbol{y}$ *contains at least one burst of length-*2 *spanning the indices* $i, i+1$, $1 \leq i \leq n-1$ *where* $((x_i, x_{i+1}), (y_i, y_{i+1})) = ((0,0),(1,1))$ *or* $((x_i, x_{i+1}), (y_i, y_{i+1})) = ((1,1),(0,0))$.

Now we turn our attention to code constructions that satisfy the conditions in Lemma 7. Suppose a group code is produced using Construction A. As recently noted in [3], a group code has an alternative interpretation.

Consider the map $\Gamma : \{0,1\}^2 \rightarrow GF(3)$. The map is defined as follows: $(0,0) \rightarrow 0, (0,1) \rightarrow 1, (1,0) \rightarrow 2, (1,1) \rightarrow 0$. Note that the map is not one-to-one since both $(0,0)$ and $(1,1)$ are mapped to 0. If the map Γ is applied to a binary vector of even length then it is simply applied to each pair of consecutive elements at a time (i.e., $\Gamma(0,0,0,0) \rightarrow (0,0)$). Furthermore, if the map Γ is applied to a set of vectors it returns a set of ternary vectors that are the result of applying the map to each vector in the set. The following Lemma was proven in [3].

Lemma 8. *(cf. [3]) Let $\mathcal{C}_3$ be a ternary code with minimum distance* 3. *Let $\mathcal{C}_2$ be the set of length-$2n$ binary vectors such that $\boldsymbol{c}_2 \in \mathcal{C}_2$ if $\Gamma(\boldsymbol{c}_2) \in \mathcal{C}_3$. Then, the code $\mathcal{C}_2$ has minimum asymmetric distance* 2.

Observe that increasing the strength of the ternary code $\mathcal{C}_3$ referred to in the previous Lemma, will increase the minimum distance of the binary code $\mathcal{C}_2$ (where $\Gamma(\mathcal{C}_2) = \mathcal{C}_3$) except in cases when the 2 codewords in $\mathcal{C}_2$ of interest satisfy the second condition enumerated in Lemma 7. This naturally leads to the following construction of a multiple grain-error correcting code $\mathcal{C}$.

Construction B. *Let n be a positive odd number. Suppose $\mathcal{C}_3$ is a t symmetric error correcting code over $GF(3)$ of length-$\frac{n-1}{2}$. Then, a vector $\boldsymbol{x} \in \{0,1\}^n$ where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a codeword in $\mathcal{C}$ if the following holds*

$$\Gamma((x_2, \ldots, x_n)) \in \mathcal{C}_3. \quad (6)$$

Remark 2. *It is interesting to observe that when the code $\mathcal{C}_3$ consists of only the all-zeros codeword, this construction coincides with the construction in [6] that can correct any number of grain-errors.*

The following theorem is straightforward.

Theorem 3. *A code $\mathcal{C}$ created with Construction B can correct any t grain-error.*

VI. Conclusion

In this work, new bounds and constructions were derived for the overlapping grain-error model. The main theme of the paper was that codes capable of correcting grain-errors are related to codes that correct insertions/deletions and codes that correct asymmetric errors. Using this basic insight new non-asymptotic upper bounds were derived. In addition, code constructions were provided that in many cases improved the state of the art. Reference [2] contains an extended version of this paper.

Acknowledgements

The authors would like to thank Artyom Sharov for his helpful discussions about code cardinalities and Enoch Andersson for his assistance with simulations. Research supported in part by SMART scholarship, ISEF, and NSF grants CCF-1029030 and CCF-1150212.

References

[1] S. D. Constantin, T. R. M. Rao, "On the theory of binary asymmetric error correcting codes," *Information and Control*, vol. 40, pp. 20-36, 1979.
[2] R. Gabrys, E. Yaakobi, L. Dolecek, "On correcting grain-errors," Technical Report, available at *http://www.loris.ee.ucla.edu/home/pubs*.
[3] M. Grassl, P. Shor, G. Smith, J. Smolin, B. Zeng, "New constructions of codes for asymmetric channels via concatenation," *Proc. IEEE ISIT*, 2012.
[4] A.R. Iyengar, P.H. Siegel, J.K. Wolf, "Write channel model for bit-patterned media recording," *IEEE Trans. Magn.*, vol. 47, no. 1, pp. 35-45, 2011.
[5] N. Kashyap, G. Zemor, "Upper bounds on the size of grain-correcting codes," available at *http://arxiv.org/abs/1302.6154*, 2013.
[6] A.A. Kulkarni, N. Kiyavash, "Non-asymptotic upper bounds for deletion correcting codes," available at *http://arxiv.org/abs/1211.3128*, 2012.
[7] A. Mazumdar, A. Barg, N. Kashyap, "Coding for high-density recording on a 1-d granular magnetic medium," *IEEE Trans. Inform. Theory*, vol. 57, no. 11, pp. 7403-7417, 2011.
[8] R. J. McEliece, E. R. Rodemich, "The Constantin-Rao construction for binary asymmetric error-correcting codes," *The Deep Space Network*, pp. 124-129, ID 19790016933, 1979.
[9] A. Sharov, R.M. Roth, "Bounds and constructions for granular media coding," *Proc. IEEE ISIT*, 2011.
[10] R. R. Varshamov, G. M. Tenengolts, "Correcting code for single asymmetric errors," *Avtomatika i Telemekhanika*, vol. 26, no. 2, pp. 288-292, 1965.
[11] R. Wood, M. Williams, A. Kavcic, J. Miles, "The feasibility of magnetic recording at 10 terabits per square inch on conventional media," *IEEE Trans. Magn.*, vol. 45, no. 2, pp. 917-923, 2009.