Generalized Integrated Interleaved Codes

Kwame Ackah Bohulu

July 18, 2018

Abstract - Generalized integrated interleaved codes refer to two-level Reed-Solomon codes, such that each code of the nested layer belongs to different subcode of the first-layer code. In this paper, we first devise an efficient decoding algorithm by ignoring first-layer miscorrection and by intelligently reusing preceding results during each iteration of a decoding attempt. Neglecting first-layer miscorrection also enables to explicitly and neatly formulate the decoding failure probability. We next derive an erasure correcting algorithm for redundant arrays of independent disks systems. We further construct an algebraic systematic encoding algorithm, which had been open. Analogously, we propose a novel generalized integrated interleaving scheme over binary BoseChaudhuri-Hocquenghem codes, reveal a lower bound on the minimum distance, and derive a similar encoding and decoding algorithm as those of ReedSolomon codes.

1 Introduction

In [1] two-level scheme which was introduced and subsequently generalized in [2] with the aim to achieve better protection over an array of interleaves (a component word, $\mathbf{c}_i \in \mathbf{c}_0 (0 \leq i < m)$ within a single block. it provides nonuniform redundancy by sharing extra check symbols with all the interleaves and the extra check symbols are used by the interleaves with errors beyond their decoding distance.

The downside to this construction is that it does not provide protection to the shared redundancies which means another code is required to protect these shared check symbols from errors.

An improvement in the form of the integrated interleaved (II) coding scheme [3], [4] creates shared redundancy that is protected by the first-layer code. Specifically, the II coding scheme [3], [4] nests a set of m equally protected interleaves with $v (v < m)$ more powerful codewords in the second (interchangeably "nested") layer which is a subcode of the first layer.

Specifically, let $\{C_i(n, k_i, d_i)\}_{i=0}^{1}$ be two Reed-Solomon (RS) codes over the Galois field GF(q) such that $C_1 \subset C_0$. An II code is defined as follows

$$C \triangleq \left\{ \mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, ..., \mathbf{c}_{m-1}] : \mathbf{c}_i, \forall i \sum_{i=0}^{m-1} \alpha^{bi} c_i \in C_1, 0 \leq b < v \right\} \qquad (1)$$

where $v < m < q$ and α is a primitive element of GF(q).

For a received word $\mathbf{y}$, self-decoding of an interleave $\mathbf{y}_i$, 0 i ¡ m, refers to the stand-alone decoding of $\mathbf{y}_i$ within the first layer, whereas nested decoding of $\mathbf{y}_i$ refers to the decoding in the nested layer resorting to the entire word y$\mathbf{y}$.

1.1 Contributions to II codes

- Cox et al. [3] described an algebraic systematic encoding method for a class of variant (by inserting zeros) II-RS codes with v = 1

- Wu [5] presented an algebraic systematic encoding algorithm for new variant (by inserting zeros differently) II-RS codes.
- Hassner et al. [4] characterized the general twolayer II codes (with $v \geq 1$), and derived a non-systematic encoding algorithm as well as an algebraic decoding algorithm. Its decoding performance was re-examined and a tight theoretical evaluation was given in [7]. The main complication stems from the miscorrection of the component words.
- Asano et al. [6] considered a special class of three-layer II coding scheme with $v = 1$ at the second and third layers.
- Tang and Koetter [8] proposed and characterized a generalized II coding scheme to allow unequal protection in the nested layer, based on the rationale that the strongest code is used to correct the most corrupted interleave while the weakest code corrects the least corrupted interleave. The authors also presented a decoding algorithm which is sophisticated due to handling miscorrection of the interleave self-decoding, and cumbersome due to the large amount of repetitive operations. Moreover, the encoding of GII codes was not investigated in [8].

1.2 Generalized Integrated Interleaved (GII) Code Definition

Let $\{C_i(n, k_i, d_i)\}_{i=0}^{1}$ be over the Galois field GF(q) such that

$$C_v \subseteq C_{v-1} \subseteq C_{v-2} \subseteq ... \subseteq C_1 \subset C_0 \tag{2}$$

A generalized integrated interleaved (GII) code is defined as

$$C \triangleq \left\{ \mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{m-1}] : \mathbf{c}_i \in C_0, 0 \leq i < m, \sum_{i=0}^{m-1} \alpha^{bi} \mathbf{c}_i \in C_{v-b}, 0 \leq b < v \right\} \tag{3}$$

where $v < m < q$ (see Figure 2).

Note that the above definition is slightly different from the original one given in [8], which is described as following under the above notation

$$C \triangleq \left\{ \mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{m-1}] : \mathbf{c}_i \in C_0, 0 \leq i < m, \sum_{i=0}^{m-1} \alpha^{bi} \mathbf{c}_i \in C_{b+1}, 0 \leq b < v \right\} \tag{4}$$

Where the difference lies in that the parameter b which corresponds to a different subcode. As shown later, the proposed definition yields simpler implementation for both encoder and decoder.

1.3 Comparison of GII to Generalized Concatenated (GC) framework

It is instrumental to compare the GII scheme with generalized concatenated (GC) framework [11] defined by Blokh and Zyablov. The GII scheme is similar to the GC scheme in view of theoretical performance and shared redundancies on top of the first layer self-correction.

- The main difference and advantage of GII codes is that their shared redundancies are also embedded in, and thus protected by, the first-layer interleaves, whereas for GC codes, the shared redundancies are not.
- The nested layer codes form a subcode order and are subcodes of the first-layer code (see (2)) in GII codes, whereas the inner (but not outer) codes forms a subcode order and are used to encode each symbol of outer codes in GC codes.
- Moreover, the nested layer codes and the first-layer code share the same field and code length in GII codes, however, the outer codes are defined in a larger field and thus have much larger length than the inner codes in GC codes.
- As a consequence, the implementation and architecture of GII codes is vastly different from those of the GC codes.

2 Generalized Integrated Interleaved Reed-Solomon Codes

In literature, all studies on (generalized) integrated interleaved (II/GII) coding scheme are carried over Reed-Solomon (RS) codes, due to the existence of efficient syndrome decoder and difficulty to extend to other codes.

In [8], the properties of GII-RS codes are stated as follows without proof. However a proof is added in this section

Theorem 1: Let $\{C_i(n, k_i, d_i)\}_{i=0}^{1}$ be over the Galois field GF(q) such that

$$C_{i_s} = ...C_{i_{s-1}+1} \subset C_{i_s-1} = ... =\subset C_{i_s-2+1} \subset ... \subset C_{i_1+1} = ... = C_1 \subset C_0 \quad (5)$$

A generalized integrated interleaved (GII) code is defined as

$$C \triangleq \left\{\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{m1}] : \mathbf{c}_i \in C_0, 0 \leq i < m, \sum_{i=0}^{m1} {}^{bi}\mathbf{c}_i \in C_{vb}, 0 \leq b < v\right\} \quad (6)$$

where $i_0 = 0$ and $i_s = v$. Let the GII-RS code, GII([m, v], n, $[d_0, d_1, ..., d_v$]), be defined as in (3), where the minimum distance sequence $[d_0, d_1, ..., d_v$] follows the non-decreasing order. The GII-RS code is a linear block code over

GF(q) of length $N = mn$, dimension $K = \sum_{i=1}^{v} k_i + (mv)k_0$, and minimum distance

$$d_{min} = \min\left\{(v+1)d_0, (v-i_1+1)d_{i_1}, ..., (v-i_{s-1}+1)d_{i_{s-1}}, d_v\right\} \quad (7)$$

Note that the notation GII([m, v], n, $[d_0, d_1, ..., d_v]$) completely describes a GII-RS code. However, it is lengthy. In the other sections, we may drop some last terms for simplicity sake, e.g., GII([m, , v], n), or GII([m, v]).

2.1 Hard Decoding Algorithm

The decoding algorithm for GII-RS codes and its analysis proposed in [8] is significantly convolved due to handling possible miscorrection of interleaves. In fact, a key feature of the integrated-interleaved codes is that it enables to decode each interleave independently during normal operation, whereas the second layer decoding is only provoked if one or more interleaves rarely suffer self-decoding failure. This feature is essential for distributed storage and power saving. Indeed, for reasonably large minimum distance (say, $d_0 > 10$) of practically deployed Reed-Solomon codes, its miscorrection probability is far below our acceptance criterion [15]. For the above reason, miscorrection for interleave self-decoding is ignored which makes the decoding algorithm much simpler. Another disadvantage is that, the decoding algorithm in [8] involves a lot of computation , in the sense that the outputs of failed decoding attempts are simply discarded, which results in a lot of repeated computation.

In this section, we show that syndromes for failed decoding attempts can be reused for future decoding attempts, and furthermore, by using the Berlekamp-Massey algorithm, the error locator polynomial and its auxiliary polynomial can be reused to incrementally generate new ones in next round of decoding attempt.

An explanation of the Berlekamp-Massey Algorithm and its characteristics is given.

In II/GII decoding schemes, when an interleave is successfully corrected, its higher order syndromes are computed in an attempt to correct the failed interleaves [4], [8]. In existing decoding algorithms, those syndromes are computed from the scratch. The following lemma demonstrates an alternative, and much more efficient approach [16, p. 183].

Lemma 1: Let **c** be a transmitted codeword and **y** the received word. If there are $e \leq t$ errors and let $\Lambda(x)$ be the corresponding error locator polynomial, then the higher order syndromes $S_i \triangleq y(\alpha^i)c(\alpha^i), i \geq 2t$ can be computed recursively through the following LFSR

$$S_i = -\Lambda_1 S_i 1_2^{\Lambda} S_i 2 - ... - \Lambda_e S_{ie}, i2t. \quad (8)$$

In II/GII decoding schemes, when an interleave decoding attempt is failed, its higher order syndromes are computed in next round to enforce a stronger

correction capability [4], [8]. In existing decoding algorithms, key-equation-solver, such as the Berlekamp-Massey algorithm, is re-carried each time. The Berlekamp-Massey algorithm recursively computes the minimum-length LFSR polynomial, corresponding to the error-locator polynomial, by incrementally incorporating higher order syndromes. Next Berlekamp-Massey updating algorithm incrementally updates the error locator polynomial utilizing the preceding results and newly available higher order syndromes.

The Berlekamp-Massey updating algorithm and the Hard Decoding Algorithm for GII-RS Codes are presented and the benefits of the latter are presented.

It is worth noting that, for single j entry, Step 6.(a) is equivalent to erasure-only decoding for an RS code, thus can be more efficiently implemented by Forneys formula [16, p. 196]. However, v is typically a small integer, say, v 4, therefore, the matrix inversion in (23), as shown at the bottom of this page does not need large amount of computation. The block diagram of the above decoding process for the most common case, $C_v \subset C_{v1} \subset ... \subset C_1 \subset C_0$, is illustrated in Figure 3.

Comparing to the original definition in [8], the proposed new definition results in the following decoder simplifications.

- $\widetilde{y}_0(x)$ is always computed in Step 4, with all-one coefficients, i.e., $\widetilde{y}_0(x) = \sum_{j\in I} y_j(x) + \sum j \in I^c c_j(x)$.
- In Step 6.(a), the connection matrix has all-one first row, which simplifies matrix inversion.
- In Step 6.(d), syndrome update for$l = 0$ is simplified to

$$\left\{\widetilde{S}_j^{(0)} \leftarrow \widetilde{S}_j^{(0)} - \sum_{i\in I'} \widetilde{S}_j^{(i)}\right\}_{j=2t_{v-b'}}^{2t_o-1}$$

Clearly, the decoding algorithm given in [8] can also be trivially simplified by neglecting the first-layer miscorrection. In comparison, the above decoding algorithm exhibits the following computational advantages over the algorithm in [8].

- The above algorithm computes the nested-layer words only when nested-layer decoding is invoked (Step 4), whereas the nested layer code are repeatedly computed during the nested-layer decoding process (Step 6) in [8].
- In the above decoding, only higher order nested syndromes are computed before nested-layer decoding is invoked (Step 5), and subsequently higher order nested syndromes are dynamically updated (Step 6.(d)), whereas all nested syndromes are computed over the updated nested words each time (Step 7) in [8].

- In the above algorithm, only higher order syndromes of uncorrected interleaves are solved (Step 6.(a)) and the Berlekamp-Massey updating algorithm is employed to reuse the preceding outputs to obtain the desired error locator polynomial (Step 6.(b)), whereas all syndromes are solved (Step 8) and the Berlekamp-Massey algorithm is deployed to produce the desired error locator polynomial from scratch (Step 9) in [8].

Theorem 2 : By neglecting miscorrection of interleave selfdecoding, let $e_0, e_1, , e_{m-1}$ denote the number of errors over received interleaves $y_0(x), y_1(x), , y_{m1}(x)$, respectively.

Reorder $e_0, e_1, , e_{m-1}$, to $\tau_0, \tau_1, , \tau_{m1}$, such that

$$\tau_{m1} \leq \tau_{m2} \leq ... \leq \tau_{v+1} \leq \tau_0 \leq \tau_1 \leq ...\tau_v \tag{9}$$

The decoding is successful if

$$\tau_i t_i, i = 0, 1, 2, ..., v \tag{10}$$