Miscorrection Mitigation for Generalized Integrated Interleaved BCH Codes

Zhenshan Xie and Xinmiao Zhang, *Senior Member*, *IEEE*

***Abstract*—The generalized integrated interleaved (GII) codes nest BCH sub-codewords to form codewords of more powerful BCH codes. They can achieve hyper-throughput decoding with excellent error-correcting performance and are among the most suitable codes for next-generation memories. However, the new storage class memories require high code rate and short codeword length. In this case, the sub-codewords have small correction capability. The miscorrections of the sub-codewords lead to much more severe degradation on the GII error-correcting performance compared to the miscorrections of classic BCH codes. This letter investigates the miscorrections in GII decoding. Three low-complexity methods are developed to identify and mitigate the miscorrections by utilizing the nested syndromes, adding one single parity to each sub-codeword, and keeping track of the error locator polynomial degree. Besides, formulas for estimating the miscorrection rates are given. Using the proposed mitigation methods, the actual GII decoding performance becomes very close to that of the case without any miscorrections.**

***Index Terms*—BCH codes, Miscorrection, Generalized integrated interleaved codes, Storage class memories**

I. Introduction

Generalized integrated interleaved (GII) codes [1], [2] nest BCH or Reed-Solomon (RS) sub-codewords to form codewords of stronger BCH or RS codes. The decoding is carried out on individual sub-words most of the time and hence very high decoding throughput can be achieved with low complexity. Besides, more errors are correctable through the nested codewords and GII codes have much better error-correcting performance than individual BCH or RS codes that can achieve similar throughput. The hyper throughput and excellent random bit error-correction capability of GII-BCH codes make them among the best candidates for storage class memories (SCMs). Alternative nesting schemes of the GII codes are available in [3]–[5].

Other types of codes have been proposed recently to enable localized decoding and parity sharing. The partial maximum distance separable codes [6] allow more flexible usage of the shared parities. However, explicit constructions are limited to special cases and large finite fields are needed. The codes in [7] also require the finite field order to be at least the code length. Parity check matrices of shorter codes are concatenated to form a larger parity check matrix in the generalized concatenated types of codes in [8]. Solving the equations specified by the parity check matrix for decoding leads to high implementation complexity. In the extended product codes (EPCs) [9], data symbols in a two-dimensional array are encoded in a concatenated manner row and column-wise using RS codes. The EPCs are further extended to codes over any field in the extended integrated interleaved (EII) codes [10]. However, repeated row-wise re-encoding is needed in the decoding process of these codes. Also when the row and column-wise codes are different types of codes or are defined over different finite fields, separate decoders are needed.

The new SCMs have much shorter sensing latency than Flash memories. To take advantage of the fast SCMs, codes with relatively short length, *e.g.* several thousand bits, and low redundancy, *e.g.* 10%, are needed. In this case, the error-correction capability of individual sub-codewords is small, such as $t_0 = 3$, and the miscorrections of the sub-codewords are non-negligible. For traditional BCH codes, a corrupted codeword may be decoded as another codeword, especially when the correction capability and hence the minimum distance of the code is small. This is referred to as the miscorrection. The formulas for estimating the miscorrection rate of BCH codes are given in [11]. However, the miscorrections for GII codes have not been investigated. From our simulations, miscorrections cause several orders of magnitude degradation on the error-correcting performance of a GII code with t_0=3 sub-codewords. This is much more severe compared to that of a 3-error-correcting BCH code. The reason is that GII codes are capable of correcting more errors using the nested codewords. However, the nested decoding will not be helpful if the miscorrections of the sub-codewords are not identified. It was proposed in [12] to mitigate miscorrections by correcting errors directly on the nested words. The correctable error patterns are quite limited and there is still substantial performance degradation. [13] sends the first sub-word whose number of errors found from the sub-word decoding equals t_0 to the nested decoding. This scheme does not lead to much performance improvement when $t_0 = 3$.

This letter investigates the error patterns leading to miscorrections in GII decoding and develops three methods to identify the miscorrections. Accordingly, the miscorrected sub-words are sent to further nested decoding to correct more errors. Our proposed schemes have low overheads in hardware implementation. The first scheme utilizes the higher-order syndromes of the nested words to detect miscorrections among all the sub-words. To help identify exactly which sub-words are miscorrected, our second scheme adds one extra parity bit to each sub-codeword. The third approach further reduces the miscorrection rate by keeping track of the error locator polynomial degree. Adopting the proposed methods, the actual performance of the GII codes is improved by several orders of magnitude and becomes very close to that of the case where no miscorrection occurs. Moreover, by analyzing the dominating error patterns leading to miscorrections, formulas are given to

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA. This work was supported in part by Kioxia Corporation and by the National Science Foundation under Award No. 2011785.

estimate the frame error rate (FER) of GII codes. For example GII codes with t_0=3 BCH sub-codewords, simulation results match the formulas well.

This letter is organized as follows. Section II introduces the GII codes. The three proposed methods for identifying and mitigating miscorrections are detailed in Section III. Section IV summarizes the proposed GII decoding algorithm and conclusions follow in Section V.

II. GII-BCH Codes and Miscorrections

Let $\mathcal{C}_v \subseteq \mathcal{C}_{v-1} \subseteq \cdots \subseteq \mathcal{C}_1 \subset \mathcal{C}_0$ be binary BCH codes defined over $GF(2^q)$ of length n and error-correction capabilities $t_v \geq t_{v-1} \geq \cdots \geq t_1 > t_0$. A GII-BCH $[m,v]$ code is defined as [2]

$$\begin{aligned}\mathcal{C} \triangleq \Big\{&[c_0(x), c_1(x), \cdots, c_{m-1}(x)] : c_i(x) \in \mathcal{C}_0,\\ &\tilde{c}_l(x) = \sum_{i=0}^{m-1} \alpha^{il}(x) c_i(x) \in \mathcal{C}_{v-l}, 0 \leq l < v\Big\}.\end{aligned} \quad (1)$$

In (1), $c_i(x)$ $(0 \leq i < m)$ is referred to as a sub-codeword and $\tilde{c}_l(x)$ $(0 \leq l < v)$ is called a nested codeword. α is a primitive element of $GF(2^q)$ and $\alpha^{il}(x)$ is the standard basis representation of α^{il} in polynomial format [2].

GII-BCH decoding has two stages. The first stage is the traditional BCH decoding on each received sub-word $y_i(x) = c_i(x) + e_i(x)$ for correcting up to t_0 errors. Here $e_i(x)$ is the error polynomial for the i-th sub-word. In this process, $2t_0$ syndromes are first computed as $S_j^{(i)} = y_i(\alpha^{j+1})$ $(0 \leq j < 2t_0)$. If some syndromes are nonzero, a key equation solver (KES), such as the Berlekamp-Massey (BM) algorithm, computes the error locator polynomial $\Lambda(x)$ iteratively using the $2t_0$ syndromes. If the number of distinct roots of $\Lambda(x)$ equals the degree of $\Lambda(x)$, denoted by $\deg(\Lambda(x))$, the decoding is considered successful and error locations are inverses of the roots. Otherwise, the decoding of the sub-word fails.

The second-stage nested decoding is activated when the first-stage decoding on some sub-words fails. $2t$ syndromes are needed to correct t errors by the BM algorithm. To correct the sub-words with more than t_0 errors, higher-order syndromes are needed and they can be derived from the nested words

$$\tilde{y}_l(x) = \sum_{i=0}^{m-1} \alpha^{il}(x) y_i(x). \quad (2)$$

From (1), $\tilde{c}_l(x) = \sum_{i=0}^{m-1} \alpha^{il}(x) c_i(x) \in \mathcal{C}_{v-l}$, which is t_{v-l}-error-correcting. Hence higher-order syndromes of the nested word $\tilde{y}_l(x)$ can be computed as $\tilde{S}_j^{(l)} = \tilde{y}_l(\alpha^{j+1})$ for $2t_0 \leq j < 2t_{v-l}$. Let the indices of the $b \leq v$ sub-words that failed the first-stage decoding be $i_0, i_1, \cdots, i_{b-1}$. Then from (2), for each of these sub-words, the higher-order syndromes with $2t_0 \leq j < 2t_1$ can be computed as

$$\left[S_j^{(i_0)}, S_j^{(i_1)}, \cdots, S_j^{(i_{b-1})}\right]^T = A^{-1}\left[\tilde{S}_j^{(0)}, \tilde{S}_j^{(1)}, \cdots, \tilde{S}_j^{(b-1)}\right]^T. \quad (3)$$

The entry of A in the u-th row and v-th column is $\alpha^{i_v u(j+1)}$ [2]. Once $2t_1$ syndromes are available, the BM algorithm can be utilized to correct up to t_1 errors for each of the b sub-words. If there are b' sub-words remain to be corrected, then $2(t_2 - t_1)$ higher-order syndromes are computed for each of them using the syndromes of the first b' nested words, each of which is at least t_2-error-correcting, using an equation similar to (3). This process is repeated for up to v rounds. Sort the number of errors in the received sub-words before the first-stage decoding as $\tau_0 \geq \tau_1 \geq \cdots \geq \tau_m$. The errors are correctable if $\tau_l \leq t_{v-l}$ for $0 \leq l \leq v$ [2]. Equivalently, if there are more than v+1-l sub-words remain to be corrected in the l-th nested decoding round, overall GII decoding failure is declared.

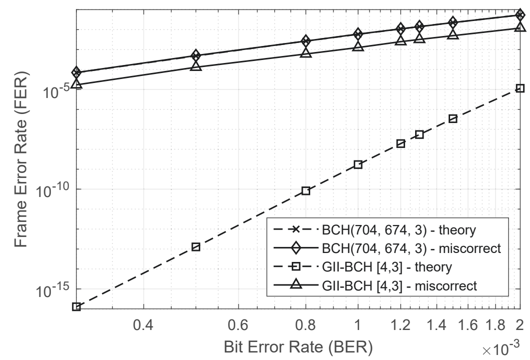

Fig. 1. FERs of BCH and GII-BCH codes over BSC

Denote the input bit error rate (BER) by p_b. $\phi_w = \binom{n}{w} p_b^w (1-p_b)^{n-w}$ is the probability of having w errors among n bits. Then

$$\theta_i^j = \sum_{w=i}^{j} \phi_w$$

is the probability of having i to j errors among n bits. The theoretical FER of GII decoding without miscorrections can be estimated by the following formula [2]

$$P_f = \sum_{b=1}^{m} \binom{m}{b} (\theta_{t_v+1}^n)^b (\theta_0^{t_v})^{m-b} + \sum_{l=0}^{v-1} \sum_{b=v+1-l}^{m} \binom{m}{b} (\theta_{t_l+1}^{t_{l+1}})^b (\theta_0^{t_l})^{m-b}. \quad (4)$$

Consider codes that protect 2560 data bits with around 256 parity bits for SCM applications. To achieve a good tradeoff on error-correcting performance and decoding complexity, GII-BCH [4,3] codes over $GF(2^{10})$ can be used. In this case, the length of each sub-codeword is (2560+256)/4=704 bits. The 256 parity bits can be allocated among $\mathcal{C}_0, \mathcal{C}_1, \cdots$ in different ways and the corresponding FER can be calculated using (4). It was found that among all possible allocations of parities, the FER for the GII-BCH [4,3] code is the lowest when $[t_0, t_1, t_2, t_3]$=$[3, 5, 6, 11]$. Accordingly, k_0=674. Fig. 1 plots the FER of this GII-BCH code over the binary symmetric channel (BSC). For reference, the FER of the $\mathcal{C}_0$ (704, 674) 3-error-correcting BCH code is also included.

For a t-error-correcting BCH code, when the number of errors is larger than t, the distance between the received word and another codeword may not exceed t. In this case, the decoding result is the other codeword and miscorrection occurs. Such probabilities are much higher for smaller t. Formulas for the miscorrection rate of BCH codes are available in [11]. For a classic (704, 674) BCH code, the miscorrection rate is around 5.4% of the FER and it is hardly observable in Fig. 1.

Miscorrections lead to much more significant degradation on the FER of GII codes. The nested decoding process of GII codes can correct more errors. However, if the sub-words are miscorrected but undetected, then they are not sent to the nested decoding. As a result, the errors that are correctable by the GII codes become uncorrected. From simulations, the miscorrections increase the FER of the GII-BCH [4,3] code by several orders of magnitude as shown in Fig. 1.

III. Miscorrection Mitigation for GII-BCH Codes

The performance degradation of GII codes caused by miscorrections can be mitigated by properly detecting the miscorrected sub-words and sending them to further nested decoding to have more errors corrected. In this section, three low-complexity methods are proposed to detect and mitigate the miscorrections through using syndromes of the nested words, adding an extra parity bit to each sub-codeword, and keeping track of the degree of the error locator polynomial.

A. Method 1: Nested Syndromes Checking

Miscorrections happen when the recovered sub-word is another codeword. Hence, for t_0-error-correcting sub-word decoding, miscorrections cannot be detected from the $2t_0$ syndromes. However, the nested words are linear combinations of the sub-words and they have higher correction capabilities. Therefore, miscorrections on the sub-words can be detected if any higher-order syndromes of the nested words are nonzero.

A sufficient number of nested syndromes need to be computed to ensure that the probability of detection failure is much smaller than the theoretical FER of GII codes. For random and independent input errors, a nested syndrome is a random value over $GF(2^q)$ and can be zero with probability around 2^{-q}. Undetected miscorrections from nested decoding round $l-1$ only cause degradation on the FER when nested decoding round l indeed needs to be carried out and the errors are correctable overall. Hence, a loose upper bound of the FER degradation caused by miscorrections that are undetectable by checking σ_l nested syndromes before nested decoding round l can be written as

$$D_l = \sum_{j=1}^{v+1-l} \binom{m}{j} (\theta_{t_{l-1}+1}^{t_v})^j (\theta_0^{t_{l-1}})^{m-j} 2^{-q\sigma_l}. \quad (5)$$

Then σ_l can be chosen to make D_l much smaller than the theoretical FER of GII decoding. For the example code with $[t_0, t_1, t_2, t_3]$=$[3, 5, 6, 11]$, $[\sigma_1, \sigma_2, \sigma_3]$=$[4, 3, 3]$ make $D_l{<}1.3{\times}10^{-16}$. Note that no more higher-order nested syndromes can be computed to check for miscorrections after the last nested decoding round. Since each nested word includes the contributions from every sub-word, it does not matter from which nested words the syndromes are computed.

The nested syndromes tell if some sub-words are miscorrected but do not tell which ones. Note that only up to $v{+}1{-}l$ sub-words can be sent to the l-th nested decoding round to correct more errors. First, the sub-words that failed in the previous decoding round are chosen for further decoding. If there are less than $v{+}1{-}l$ of them, the other sub-words are considered for further decoding. From our simulations, for a received sub-word with more than t errors, $\deg(\Lambda(x))$ equals $t, t{-}1, \cdots$ with decreasing probability. Hence, sub-words can be chosen according to decreasing $\deg(\Lambda(x))$ to fill the $v{+}1{-}l$ quota so that potentially miscorrected sub-words go through further decoding. In the case that there is a tie on $\deg(\Lambda(x))$, the sub-words with the lowest indices can be chosen instead of trying all possible combinations of the sub-words to reduce the hardware complexity. Picking any sub-words with a tie on $\deg(\Lambda(x))$, such as those with the highest indices, leads to equal probability of decoding failure.

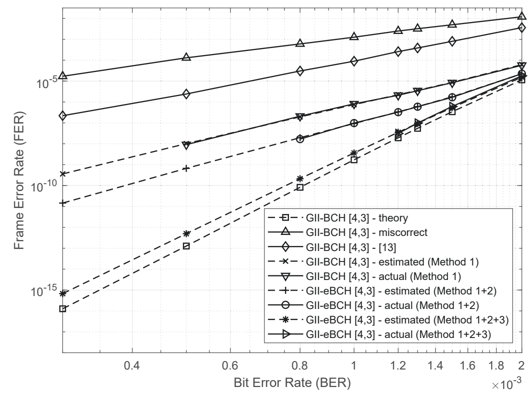

Fig. 2. FERs of GII-BCH decoding with different miscorrection mitigation schemes over BSC

It is possible that some of the miscorrected sub-words are not selected for further decoding if they are chosen according to decreasing $\deg(\Lambda(x))$ and lowest indices. GII decoding failure happens in this case. The FER degradation caused by such cases can be estimated by analyzing the corresponding dominating error patterns, which are found from simulations to be that one sub-word is miscorrected in the first-stage decoding but is not selected for nested decoding. For a GII-BCH $[m,v]$ code, assume that $y_h(x)$ is miscorrected ($0{\le}h{<}m$). Besides, it has i ($t_0{<}i{\le}t_v$) errors and $\deg(\Lambda(x)){=}j$ ($0{\le}j{\le}t_0$) from the KES in the first-stage decoding. If $y_h(x)$ is not chosen for nested decoding, there must be at least v sub-words that failed the first-stage decoding, have higher $\deg(\Lambda(x))$, or have $\deg(\Lambda(x)){=}j$ but lower indices. Next, the probabilities of having b of these sub-words from the set $[y_{h+1}(x), \cdots, y_{m-1}(x)]$ and the rest from $[y_0(x), \cdots, y_{h-1}(x)]$ are estimated.

A sub-word whose index is higher than h will be selected for nested decoding only if decoding fails in its first-stage decoding or decoding success is declared but its $\deg(\Lambda(x)){>}j$. On the other hand, if the error pattern is correctable and $y_h(x)$ has more than t_0 errors, then at most $v-1$ remaining sub-words can have more than t_0 errors. Let $\bar{b}{=}\min(v{-}1, b)$ and $\bar{m}{=}m{-}1{-}h$. The probability of having b sub-words chosen from $[y_{h+1}(x), \cdots, y_{m-1}(x)]$ is upper bounded by

$$g(h,b,j) = \binom{\bar{m}}{b} (\theta_0^j)^{\bar{m}-b} \Big(\sum_{\rho=0}^{\bar{b}} \binom{b}{\rho} (\theta_{j+1}^{t_0})^{b-\rho} \big(\sum_{\tau=t_0+1}^{t_v} \phi_\tau F_\tau \big)^\rho \Big), \quad (6)$$

where F_τ is the probability that a received sub-word has τ errors and decoding failure has been declared from its t_0-error-correcting decoding. F_τ can be determined from simulations over a limited number of samples, such as 10^7 for the example GII-BCH [4,3] code. In (6), the second term on the right side is the probability of having $\bar{m}{-}b$ sub-words with up to j errors. The terms in the big parenthesis denote the probability that the rest b sub-words either have first-stage decoding success with $\deg(\Lambda(x)){>}j$ or decoding failure. The upper bound of the last sum is t_v since no sub-word can have more than t_v errors in an error pattern correctable by GII decoding.

An upper bound of the probability of having at least $\bar{v}=\max(v-b,0)$ received sub-words chosen from the set $[y_0(x),\cdots,y_{h-1}(x)]$ can be derived similarly as

$$f(h,b,j)=\sum_{k=\bar{v}}^{h}\binom{h}{k}(\theta_0^{j-1})^{h-k}\Big(\sum_{\rho=0}^{\bar{k}}\binom{k}{\rho}(\theta_j^{t_0})^{k-\rho}\big(\sum_{\tau=t_0+1}^{t_v}\phi_\tau F_\tau\big)^\rho\Big), \tag{7}$$

where $\bar{k}=\min(v-1,k)$. Different from the sub-words with indices higher than h, a sub-word with index lower than h can be chosen for nested decoding if it has $\deg(\Lambda(x))\geq j$ instead of $\deg(\Lambda(x))>j$. As a result, the subscripts and superscripts of the θ terms are changed accordingly as in (7).

Multiplying the probabilities from (6) and (7) and adding up the product for each possible value of b, the extra FER resulted from not sending the miscorrected sub-word to the nested decoding is upper bounded by

$$P_1=\sum_{h=0}^{m-1}\sum_{i=t_0+1}^{t_v}\sum_{j=1}^{t_0}\sum_{b=\max(v-h,0)}^{\bar{m}}\phi_i G_{i,j} f(h,b,j)g(h,b,j). \tag{8}$$

$G_{i,j}$ is the probability of an i-error-corrupted sub-word miscorrected with $\deg(\Lambda(x))=j$ in the first-stage decoding and it can be derived from simulations over a limited number of samples. Similar equations can be derived for the FER degradation brought by sending the wrong sub-words to later nested decoding rounds. However, they are much smaller and can be ignored. For the example GII-BCH [4,3] code, at BER=10^{-3}, the extra FER degradations resulted from sending the wrong sub-words to the first and second nested decoding rounds are 7.5×10^{-7} and 2.1×10^{-13}, respectively. Checking σ_l nested syndromes, where σ_l is computed from (5), can detect if there are miscorrected sub-words with almost 100% accuracy. Hence, the FER of GII-BCH decoding that checks nested syndromes for miscorrections and picks the sub-words with the highest $\deg(\Lambda(x))$ and lowest indices for further nested decoding can be estimated as P_f+P_1. This estimation is plotted in Fig. 2 for the example code. Simulations have also been carried out and the results match the estimation well as shown in the figure. It was proposed in [13] to send the first sub-word with t_0 errors for further nested decoding. This miscorrection mitigation approach leads to much less FER improvement for codes with small t_0 as shown in Fig. 2.

B. Method 2: Utilizing Extended BCH Sub-Codewords

Checking the nested syndromes does not tell exactly which sub-words are miscorrected. Among the sub-words with the same $\deg(\Lambda(x))$, always picking the ones with the lowest indices does not close the performance gap as shown in Fig. 2. To further improve the actual performance, the miscorrected sub-words need to be better identified. The most likely case for miscorrection is that a sub-word with $t+1$ errors ends up with $\deg(\Lambda(x))=t$. To identify this case, this letter proposes to utilize extended BCH (eBCH) codes [14] for GII code construction. $(x+1)$ is multiplied to the generator polynomial of each $\mathcal{C}_i$ $(0\leq i\leq v)$. This leads to only one extra parity bit for each sub-codeword and the weight of every sub-codeword is even. Hence, XORing the bits of a received sub-word tells the parity of the number of errors. If this parity is different from that of $\deg(\Lambda(x))$, then miscorrection happens. By excluding the miscorrections in which the parities of the received word and $\deg(\Lambda(x))$ do not match from (8), the FER degradation caused by miscorrections after nested syndrome checking and utilizing eBCH codes is upper bounded by

$$P_2=\sum_{h=0}^{m-1}\sum_{i=t_0+1}^{t_v}\sum_{\substack{j=1\\ j\text{ odd/even}}}^{t_0}\sum_{b=\max(v-h,0)}^{\bar{m}}\phi_i G_{i,j} f(h,b,j)g(h,b,j), \tag{9}$$

where the third summation is carried out on even and odd j when i is even and odd, respectively. Now the FER can be estimated as P_f+P_2, and it is plotted in Fig. 2 for the example code. This estimation also matches our simulation results well.

C. Method 3: Tracking Error Locator Polynomial Degree

Still there is a significant gap between the actual and theoretical performance of the GII decoding. The following scheme helps to further identify the miscorrections that cannot be detected by the previous two approaches. In BCH decoding, $\deg(\Lambda(x))>t$ only happens when there are more than t errors. Having $\deg(\Lambda(x))>t$ in a t-error-correcting decoding means that the decoding result is incorrect. However, traditional t-error-correcting BCH decoders, such as that in [15], only keep the $\Lambda(x)$ coefficients whose degrees are up to t. Miscorrections happen when the number of the roots of the truncated $\Lambda(x)$ equals its degree. This is the major contributor to the miscorrections undetected by the previous two approaches. At lower input BER, this contribution becomes even more significant. To address this issue, this letter proposes to keep the full error locator polynomial and keep track of its degree. The first-stage sub-word decoding can be considered as nested decoding round 0. If $\deg(\Lambda(x))>t_l$ is detected in nested decoding round l, decoding failure is declared. Utilizing all the three proposed approaches, the probability of extra decoding failure caused by miscorrections is reduced to

$$P_3=\sum_{h=0}^{m-1}\sum_{i=t_0+1}^{t_v}\sum_{\substack{j=1\\ j\text{ odd/even}}}^{t_0}\sum_{b=\max(v-h,0)}^{\bar{m}}\phi_i(G_{i,j}-G'_{i,j})f(h,b,j)g(h,b,j). \tag{10}$$

Here $G'_{i,j}$ is the probability of an i-error-corrupted sub-word having $\deg(\Lambda(x))>t_0$ and the truncated $\Lambda(x)$ is a degree-j polynomial with j distinct roots. $G'_{i,j}$ can be also determined from simulations over a limited number of samples.

The GII decoding FER incorporating the three proposed miscorrection detection/mitigation schemes estimated using (10) is shown in Fig. 2. It matches our simulation results well too. Utilizing the proposed methods, the actual decoding performance of the GII code is brought very close to that of the miscorrection-free decoding. Estimations and simulations have also been done for a GII-BCH [5,3] code with $[t_0,t_1,t_2,t_3]$=$[3,4,6,9]$. It is observed that our proposed schemes can detect and mitigate almost all miscorrections too and simulation results match the estimation results well.

IV. Modified Algorithm and Overhead Analyses

The proposed GII decoding with miscorrection handling is summarized in Algorithm 1. In this algorithm, I is the set of sub-words and I^c=$\{0,..,m-1\}\backslash I$. I^t is the set of sub-words that satisfy the conditions in Line 2. Basically, they are the sub-words whose decoding seems to be successful in the current

Algorithm 1: GII-eBCH Decoding Algorithm

Input: received sub-words $y_i(x)$ $(0{\leq}i{<}m)$; t_i $(0{\leq}i{\leq}v)$
Initialization: $I=\{0,1,\cdots,m-1\}$; $I^c=\emptyset$; $I^t=\emptyset$
f_i=XOR of all bits in $y_i(x)$
for $l=0,1,\cdots,v$ **do**
  if (l=0), compute $2t_0$ syndromes for each $y_i(x){\in}I$
    if all are zero, declare decoding success; stop
1: else derive $2t_l{-}2t_{l-1}$ higher-order syn. for each $y_i(x){\in}I$
  for each $i\in I$ **do**
    carry out KES on $y_i(x)$; $d_i{\Leftarrow}\deg(\Lambda_i(x))$
2:   if ($d_i{\leq}t_l$) & (d_i mod 2=f_i) & (d_i=root # of $\Lambda_i(x)$),
      $I^t=I^t\cup i$; $I=I\backslash i$
  end
  if ($|I|{>}v{-}l$), declare decoding failure; stop
3: if ($|I|$=0), compute σ_{l+1} nested syndromes
    if (l=v) or (all these σ_{l+1} nested syndromes are zero),
      $I^c=I^c\cup I^t$; use $\Lambda_i(x)$ to correct each $y_i(x)\in I^c$
      declare decoding success; stop
  while $|I|{<}v{-}l$ **do**
4:   find least $i{\in}I^t$ with maximum d_i; I=$I\cup i$; I^t=$I^t\backslash i$
  end
  $I^c=I^c\cup I^t$; $I^t=\emptyset$
end

round. Since I^t is small, the sorting in Line 4 incurs negligible complexity. At the end of round l, if there are more than $v{-}l$ uncorrected sub-words, GII decoding failure is declared. If every sub-word seems to be corrected, σ_{l+1} nested syndromes are computed as in Line 3 to check for miscorrections. These computations are not carried out in the last round since no more higher-order nested syndromes can be computed.

The three proposed schemes can be implemented with very low overheads. In hardware implementations of GII decoders, such as [13], [16], units are already available for computing the higher-order nested syndromes for Line 1 of Algorithm 1. They can be utilized to compute the nested syndromes for miscorrection checking. For longer codes, the code rate loss caused by using eBCH codes is negligible since only one extra parity bit is added to each sub-codeword. The additional factor $(x{+}1)$ multiplied to the generator polynomials of eBCH codes only requires one additional tap in the linear feedback shift register encoder architecture [17]. For the decoding, the bit-wise XOR of each received sub-word is only computed once. The same number of syndromes are computed and the decoding is carried out in the same way. Hence, the overheads for adopting eBCH codes are negligible. Although eBCH codes with more parity bits can be utilized to identify and mitigate miscorrections, they do not lead to much additional performance gain and require higher complexity. In t-error-correcting decoding, $\deg(\Lambda(x))$ can become $t+\eta$, where η is typically a small number, such as 2 or 3, when there are more than t errors. Although our third approach needs to keep longer $\Lambda(x)$, the overheads brought to the overall GII decoding complexity are also very small, considering that syndrome computation and exhaustive root search also require many calculations [16].

When the alternative nesting scheme in [3] is used, the syndromes of the nested word that includes the contribution of every sub-word should be chosen for miscorrection detection. The proposed methods can be also extended to GII codes with more layers [4], [5] with minor modifications on the sub-word selection when nested syndromes are nonzero. Due to the non-binary symbols, RS codes for correcting t symbols have lower miscorrection rate compared to BCH codes correcting t bits of errors. Nevertheless, similar approaches can be also applied to mitigate the miscorrections of GII-RS codes.

V. Conclusions

This letter proposes low-complexity methods to detect and mitigate the miscorrections in GII-BCH decoding. The nested syndromes are utilized to detect if there are miscorrections among the received sub-words. The specific sub-words that are miscorrected are further identified by using eBCH codes and keeping track of the degree of the error locator polynomial. Besides, formulas are provided to estimate the actual FERs. Combining the three proposed schemes, the error-correcting performance degradation of the GII decoding caused by the miscorrections is almost completely eliminated. Future work will study the implementation of GII decoders.

References

[1] X. Tang and R. Koetter, "A novel method for combining algebraic decoding and iterative processing," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, USA, Jul. 2006, pp. 474-478.

[2] Y. Wu, "Generalized integrated interleaved codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1102-1119, Feb. 2017.

[3] X. Zhang, "Modified generalized integrated interleaved codes for local erasure recovery," *IEEE Commun. Letters*, vol. 21, no. 6, pp 1241-1244, Jun. 2017.

[4] X. Zhang, "Generalized three-layer integrated interleaved codes," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 442-445, Mar. 2018.

[5] W. Li, J. Lin, and Z. Wang, "Multi-Layer generalized integrated interleaved codes," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 1880-1884, Sep. 2020.

[6] M. Blaum, J. L. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4510-4519, Jul. 2013.

[7] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661-4676, Aug. 2014.

[8] M. Blaum and S. Hetzler, "Generalized concatenated types of codes for erasure correction," arXiv: 1406.6270, Jul. 10, 2014.

[9] M. Blaum and S. R. Hetzler, "Extended product and integrated interleaved codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1497-1513, Mar. 2018.

[10] M. Blaum, "Extended integrated interleaved codes over any field with applications to locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, pp. 936-956, Feb. 2020.

[11] M. G. Kim and J. H. Lee, "Undetected error probabilities of binary primitive BCH codes for both error correction and detection," *IEEE Trans. Comm.*, vol. 44, no. 5, pp. 575-580, 1996.

[12] X. Tang and R. Koetter, "On the performance of integrated interleaving coding schemes," in *Proc. IEEE Int. Symp. Inf. Theory*, Chicago, IL, USA, 2004, pp. 329-329.

[13] W. Li, J. Lin, and Z. Wang, "A 124-Gb/s decoder for generalized integrated interleaved codes," *IEEE Trans. Circuits and Syst. I: Regular Papers*, vol. 66, no. 8, pp. 3174-3187, Aug. 2019.

[14] Y. Wu and E. E. Gad, "Scalable block-wise product BCH codes," arXiv:1812.07082, Dec. 17, 2018.

[15] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for Reed-Solomon decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 641-655, Oct. 2001.

[16] X. Zhang and Z. Xie, "Efficient architectures for generalized integrated interleaved decoder," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 66, no. 10, pp. 4018-4031, Oct. 2019.

[17] X. Zhang, "Systematic encoder of generalized three-layer integrated interleaved codes," in *IEEE Intl. Conf. Comm.*, May 2019, pp. 1-6.