Performance Analysis of Extended Integrated Interleaved Codes

Keyue Deng, Xinyuan Qiao, Yuxing Chen, Suwen Song, and Zhongfeng Wang
School of Electronic Science and Engineering, Nanjing University, Nanjing, China
Email: kkeyuedeng@163.com, xyqiao@smail.nju.edu.cn, yxing.chen@outlook.com, suwsong@sina.com, zfwang@nju.edu.cn

***Abstract*—Extended integrated interleaved (EII) codes, as the versatile alternative to locally recoverable codes (LRCs), show great potential in distributed storage systems, in which the output bit-error-rate (BER) below 10^{-15} is usually demanded. However, it is time-consuming to reach such a low BER through normal software simulation, which brings inconvenience to the code construction. To solve the above problem, this work presents an analysis method to evaluate the decoding performance of EII codes, and no simulation is required. Numerical results show that the estimated frame-error-rate (FER) matches well with the simulated FER, so does the BER. Moreover, the failure probability of each decoding stage can be predicted accurately. Therefore, we can dig deep into the decoding behavior of each stage, which guides the adjustment of redundancy distribution, improving the error correction performance. Finally, the theoretical analysis for regular EII codes is simplified to reduce calculations.**

***Index Terms*—Extended integrated interleaved (EII) codes, locally recoverable codes (LRCs), distributed storage, theoretical performance analysis.**

I. Introduction

For next-generation digital storage systems, error-correcting codes achieving high throughput with excellent coding gain are needed. Conventionally, maximum distance separable (MDS) codes, such as Reed-Solomon (RS) codes are used because their redundancy is minimum [1]. However, there are probably fewer failures than the designed correction capability in most cases [2]. Locally recoverable codes (LRCs) have gained considerable research attention, due to their better locality [3], [4]. Extended integrated interleaved (EII) codes with both local and global recovery properties are good alternatives to LRCs [5]. Although EII codes do not have the optimal minimum distance in general, the property that they can be constructed over any field, especially over small fields, is very attractive since operations over a small field involve less complexity [6]. Due to the global parities shared among different levels, multi-level EII codes have stronger local protection abilities than 1-level EII codes. The above properties ensure multi-level EII codes competitive in the distributed storage systems such as cloud computing applications [7] and redundant arrays of independent disks (RAID) architectures [8], [9].

It is necessary to design EII codes that meet the application requirements before implementation. Storage systems usually require the output bit-error-rate (BER) to be below 10^{-15} [10], which leads to an intolerable consumption in software simulation of error correction performance. On the one hand, there is generally no guarantee that the constructed EII code could reach the design specification. On the other hand, even if we obtain a qualified EII code, its parameters are probably not optimal. Therefore, an effective method to adjust the parameters is essential. If the performance can be accurately estimated in advance through theoretical analysis, it will significantly speed up the code construction process [11], [12].

To the best of our knowledge, this work is the first to present a performance analysis method for EII codes in the open literature. Based on the stagewise process of the decoding, we first calculate the failure probability of each stage, and then aggregate them together to obtain the failure probability of the EII code. Furthermore, we simplify the performance analysis of regular EII codes, for less calculation. It turns out that the theoretical and simulation results are almost identical. Moreover, the failure probability of each decoding stage can help us adjust the distribution of parities to achieve better performance with the same overhead.

The rest of the paper is organized as follows. Section II introduces the encoding and decoding algorithms of EII codes. In Section III, the proposed analysis method is detailed and comparison results are provided. The simplfied method for regular EII codes is presented in Section IV. Finally, Section VI concludes the paper.

II. Preliminaries

A. Systematic Multi-Level EII Codes

A systematic l-level ($l > 1$) EII code $\mathcal{C}(n, \boldsymbol{u})$ can be represented by an $m \times n$ matrix as shown in Fig. 1(a). Let $\boldsymbol{u}$ be the vector of length $m = \sum_{i=0}^{l} s_i$, where $s_i \geqslant 1$ for $0 \leqslant i \leqslant l-1$ and $s_l \geqslant 0$.

$$\boldsymbol{u} = (\overbrace{u_0, \ldots, u_0}^{s_0}, \overbrace{u_1, \ldots, u_1}^{s_1}, \cdots, \overbrace{u_l, \ldots, u_l}^{s_l}), \tag{1}$$

where $0 \leqslant u_0 < u_1 < \cdots < u_l = n$. s_i denotes the number of rows with u_i parity symbols. Note that if $s_l = 0$, $\mathcal{C}(n, \boldsymbol{u})$ will become an l-level integrated interleaved (II) code. When $l = 1$, $\mathcal{C}(n, \boldsymbol{u})$ degrades to a 1-level EII code , where

$$\boldsymbol{u} = (\overbrace{u_0, \ldots, u_0}^{s_0}, \overbrace{u_1, \ldots, u_1}^{s_1}). \tag{2}$$

Consider a systematic l-level EII code matrix $C \in \mathcal{C}(n, \boldsymbol{u})$. Let $\hat{s}_i = \sum_{\lambda=i}^{l} s_\lambda$ for $0 \leqslant i \leqslant l$. As shown in Fig. 1(b), C is the direct sum of l 1-level EII code matrices $C^{(i)} \in \mathcal{C}(n, \boldsymbol{u}^{(i)})$, *i.e.*,

$$C = \sum_{i=0}^{l-1} C^{(i)}, \tag{3}$$

where $\mathcal{C}(n, \boldsymbol{u}^{(i)})$ $(0 \leqslant i \leqslant l-1)$ is the i-th 1-level EII code, and

$$\boldsymbol{u}^{(i)} = (\overbrace{u_i, \dots, u_i}^{m-\hat{s}_{i+1}}, \overbrace{u_{i+1}, \dots, u_{i+1}}^{\hat{s}_{i+1}}). \quad (4)$$

Let $\{\mathcal{C}_i | 0 \leqslant i \leqslant l-1\}$ be a set of l horizontal nested codes $[n, n-u_i, d_i^H]$ over Galois field $GF(q)$ such that $\mathcal{C}_{l-1} \subset \mathcal{C}_{l-2} \subset \cdots \subset \mathcal{C}_0$. Let $\{\mathcal{V}_i | 0 \leqslant i \leqslant l-1\}$ be a set of l vertical codes $[m, m-\hat{s}_{l-i}, d_i^V]$ over $GF(q^{u_{l-i}-u_{l-i-1}})$ that is linear over $GF(q)$. The horizontal and vertical codes of $C^{(i)}$ are $\mathcal{C}_i$ and $\mathcal{V}_{l-i-1}$, respectively. And all rows of C are the codes in $\mathcal{C}_0$.

Next, we describe the systematic encoding algorithm of multi-level EII codes in detail. As shown in Fig. 1(a), the l yellow matrices of size $(\sum_{j=0}^{i} s_j) \cdot (u_{i+1} - u_i)$ denoted by $D^{(i)}$ are raw data blocks, where $0 \leqslant i \leqslant l-1$. $P^{(i)}$ and P, denoted by the blue area in Fig. 1(b), represent all parities of the i-th sub-codeword $C^{(i)}$ and encoded codeword C, respectively. The encoding is sequentially performed from $C^{(l-1)}$ to $C^{(0)}$. For sub-codeword $C^{(l-1)}$, first $u_l - u_{l-1}$ columns are encoded systematically into a vertical code over $GF(q^{u_{l-i}-u_{l-i-1}})$ (or $u_l - u_{l-1}$ codes over $GF(q)$) in $\mathcal{V}_0$, and then all rows are encoded systematically into horizontal codes in $\mathcal{C}_{l-1}$. For sub-codewords $C^{(i)}$ $(0 \leqslant i \leqslant l-2)$, in order to make codeword C systematic after sum operations, their first $u_l - u_{i+1}$ columns (indicated by the white blocks in Fig. 1(b)) are filled with zeros, and their data blocks need to be added with the previous encoded parity symbols (marked by the orange blocks in Fig. 1(b)), *i.e.*,

$$D'^{(i)} = D^{(i)} + \sum_{j=i+1}^{l-1} P_{D^{(i)}}^{(j)}, \quad (5)$$

where $0 \leqslant i \leqslant l-2$, and $P_{D^{(i)}}^{(j)}$ denotes a specific parity block in $C^{(j)}$ (marked by the dashed rectangles in Fig. 1(b)). The relative position of $P_{D^{(i)}}^{(j)}$ in $C^{(j)}$ is the same as that of $D^{(i)}$ in C. After that, $u_{i+1} - u_i$ columns next to the all-zero blocks are encoded systematically into a vertical code over $GF(q^{u_{i+1}-u_i})$ (or $u_{i+1} - u_i$ codes over $GF(q)$) in $\mathcal{V}_{l-i-1}$ and then all rows are encoded systematically into horizontal codes in $\mathcal{C}_i$. Finally, a systematic EII codeword C is obtained by (3). Let $\boldsymbol{t}^{(C)}$ and $\boldsymbol{t}^{(V)}$ denote the error-correcting capabilities of horizontal codes and vertical codes, where $\boldsymbol{t}^{(C)} = \{t_i^{(\mathrm{C})} | 0 \leqslant i \leqslant l-1\}$ and $\boldsymbol{t}^{(V)} = \{t_{l-i-1}^{(\mathrm{V})} | 0 \leqslant i \leqslant l-1\}$. In this paper, if the EII codes show the following regularity: $t_{l-i}^{(\mathrm{V})} = t_{l-i-1}^{(\mathrm{V})} + 1$, for $1 \leqslant i \leqslant l-1$, we call them regular EII codes.

B. Decoding Scheme

In order to describe the decoding process conveniently, we use row vectors to represent a systematic EII codeword and its sub-codewords. Let $C = (\boldsymbol{c}_0, \boldsymbol{c}_1, \cdots, \boldsymbol{c}_{m-1})^T$ and $C^{(i)} = (\boldsymbol{c}_0^{(i)}, \boldsymbol{c}_1^{(i)}, \cdots, \boldsymbol{c}_{m-1}^{(i)})^T$, where $0 \leqslant i \leqslant l-1$. According to (3), we get

$$\boldsymbol{c}_k = \boldsymbol{c}_k^{(0)} \oplus \boldsymbol{c}_k^{(1)} \oplus \cdots \oplus \boldsymbol{c}_k^{(l-1)}, \text{ for } 0 \leqslant k \leqslant m-1. \quad (6)$$

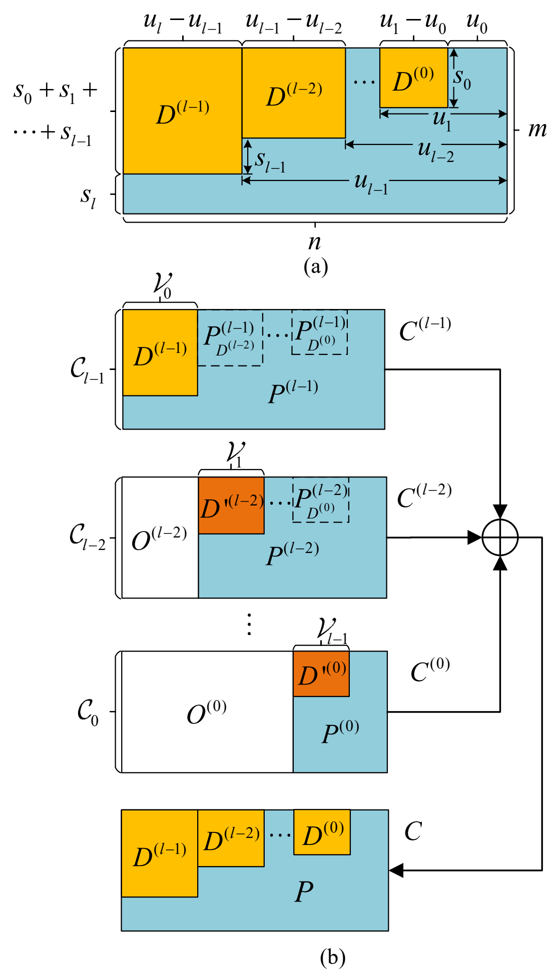

Fig. 1. An l-level EII code and its systematic encoding process.

Note that if $\boldsymbol{c}_k^{(0)} \oplus \boldsymbol{c}_k^{(1)} \oplus \cdots \oplus \boldsymbol{c}_k^{(j)}$ $(1 \leqslant j \leqslant l-1)$ is successfully decoded, each of its subcodes $\boldsymbol{c}_k^{(0)}, \boldsymbol{c}_k^{(1)}, \cdots, \boldsymbol{c}_k^{(j)}$ can be obtained. A more detailed description can be found in [6, Lemma 30].

The decoding is an iterative process, sequentially performed from $C^{(0)}$ to $C^{(l-1)}$. The first iteration starts with the decoding of all rows in C by $\mathcal{C}_0$, and we get all the subcodes $\boldsymbol{c}_{\mathcal{R}_0}^{(0)}, \boldsymbol{c}_{\mathcal{R}_0}^{(1)}, \cdots, \boldsymbol{c}_{\mathcal{R}_0}^{(l-1)}$ of the corrected rows, where $\mathcal{R}_0$ is a index set of these rows. Then, fill the first $u_l - u_1$ columns of $C^{(0)}$ with zeros and decode the $u_1 - u_0$ columns next to the zero block by $\mathcal{V}_{l-1}$. Next, encode the first $n - u_0$ columns by $\mathcal{C}_0$ to recover the sub-codeword $C^{(0)}$. After that, an $(l-1)$-level EII codeword is obtained by $C \oplus C^{(0)}$. In the second iteration, decode the uncorrected rows in the previous iteration by $\mathcal{C}_1$ to obtain all the subcodes $\boldsymbol{c}_{\mathcal{R}_1}^{(1)}, \boldsymbol{c}_{\mathcal{R}_1}^{(2)}, \cdots, \boldsymbol{c}_{\mathcal{R}_1}^{(l-1)}$ of newly corrected rows. Fill the first $u_l - u_2$ columns of $C^{(1)}$ with zeros, and decode the $u_2 - u_1$ columns next to the zero block by $\mathcal{V}_{l-2}$. At the end of the second iteration, encode the first $n - u_1$ columns by $\mathcal{C}_1$ to recover the sub-codeword $C^{(1)}$. The other iterations operate similarly. If there are no erroneous rows, the decoding terminates. Evidently, the decoding of systematic EII codes is a stage-by-stage process. Note that the decoding of the i-th stage includes the row decoding by horizontal codes $\mathcal{C}_i$ and the column decoding by vertical codes $\mathcal{V}_{l-i-1}$.

III. Performance Analysis Method for EII Codes

Since decoding is a stage-by-stage process, the $(i+1)$-th decoding stage using horizontal code $\mathcal{C}_{i+1}$ is related to the results of the i-th decoding stage using vertical code $\mathcal{V}_{l-i-1}$, where $0 \leqslant i \leqslant l-1$. Consider a 3-level EII code $\mathcal{C}(127,(12,12,16,16,16,16,22,22,127,127))$, where $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$ are [127,115,13], [127,111,17], and [127,105,23] RS codes over $GF(2^7)$ such that $\mathcal{C}_2 \subset \mathcal{C}_1 \subset \mathcal{C}_0$, and $\mathcal{V}_0$, $\mathcal{V}_1$ and $\mathcal{V}_2$ are [10,8,3], [10,6,5], and [10,2,9] shortened RS codes over $GF(2^7)$. Note that $\boldsymbol{t}^{(C)} = \{6,8,11\}$ and $\boldsymbol{t}^{(V)} = \{4,2,1\}$. If there are more than four rows that are not correctly decoded by $\mathcal{C}_0$, the entries of $C^{(0)}$ can not be retrieved totally using $\mathcal{V}_2$ at the 0-th stage. Thus $\underline{c}_j \oplus \underline{c}_j^{(0)}$ can not be obtained, which is in $\mathcal{C}_1$ and is required at the 1-st stage, where $0 \leqslant j \leqslant m-1$. Therefore, it will be detected as an unsuccessful decoding. Based on the discussion above, a performance analysis method for EII codes is proposed in this section. It should be noted that the analysis in this work is based on the binary symmetric channel (BSC) where each bit will be transmitted incorrectly with a probability α, called channel crossover probability. Moreover, we restrict both horizontal and vertical codes to the RS codes, and the analysis of other codes can be obtained similarly.

A. FER Analysis

By neglecting miscorrections of the horizontal codes, the decoding will fail if there are at least $t_{l-i-1}^{(\mathrm{V})}+1$ rows whose error weights are larger than $t_i^{(\mathrm{C})}$. According to the property of decoding by stage, we can calculate the probability of unsuccessful decoding of each stage (mutually exclusive), and sum up the wrong decoding probabilities of all stages to get failure probability of the codeword. The decoding failure of the 0-th stage means that the number of rows with more than $t_0^{(\mathrm{C})}$ errors is larger than $t_{l-1}^{(\mathrm{V})}$. And decoding will fail at the i-th stage for $1 \leqslant i \leqslant l-1$ iff the $0 \sim (i-1)$-th stages are all successful and the i-th stage is failed. In such cases, the following conditions should be both satisfied:

Condition 1: The number of rows with more than $t_i^{(\mathrm{C})}$ errors is above ($>$) $t_{l-i-1}^{(\mathrm{V})}$.

Condition 2: The number of rows with more than $t_r^{(\mathrm{C})}$ errors is below ($\leqslant$) $t_{l-r-1}^{(\mathrm{V})}$, where $0 \leqslant r \leqslant i-1$.

We divide all rows of a codeword into g groups based on the number of errors contained, where $g = l+1$. For group i, the number of errors in a row is in the range of $t_{i-1}^{(\mathrm{C})}+1$ to $t_i^{(\mathrm{C})}$. Especially, for group 0 and group $g-1$, the number of errors in a row is in the range of 0 to $t_0^{(\mathrm{C})}$ and $t_{g-2}^{(\mathrm{C})}+1$ to n, respectively. Let λ_τ denote the number of rows in group τ. To meet condition 1, we have

$$\sum_{\tau=i+1}^{g-1} \lambda_\tau \geqslant t_{l-i-1}^{(V)} + 1. \tag{7}$$

To meet the condition 2, we have

$$\sum_{\tau=0}^{r} \lambda_\tau \geqslant m - t_{l-r-1}^{(V)}, \text{ for } 0 \leqslant r \leqslant i-1. \tag{8}$$

Since the total number of rows is m, the following constraint (9) must be met.

$$\sum_{\tau=0}^{i-1} \lambda_\tau + \lambda_i + \sum_{\tau=i+1}^{g-1} \lambda_\tau = m. \tag{9}$$

Then, we can set up a linear programming mathematical model to obtain the values of parameter λ_τ under all different cases. Let σ denote the number of rows with errors more than $t_i^{(\mathrm{C})}$, where $\sigma = \sum_{\tau=i+1}^{g-1} \lambda_\tau$. After certain transformations of (7), (8) and (9), a set of linear equations and inequalities can be listed as follows.

$$\begin{cases} 0 \leqslant \lambda_i \leqslant t_{l-i}^{(V)} - t_{l-i-1}^{(V)} - 1, & \text{(10a)} \\ t_{l-i-1}^{(V)} + 1 \leqslant \sigma \leqslant t_{l-i}^{(V)} - \lambda_i, & \text{(10b)} \\ \sum_{\tau=0}^{i-1} \lambda_\tau = m - \lambda_i - \sigma, & \text{(10c)} \\ \sum_{\tau=0}^{r} \lambda_\tau \geqslant m - t_{l-r-1}^{(V)}, \text{ for } 0 \leqslant r \leqslant i-2. & \text{(10d)} \end{cases}$$

Combining (10a) and (10b), we can find all possible values of σ, which form a set $\{\sigma_j | 0 \leqslant j \leqslant \chi - 1\}$. Let $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_i)^T$. Let $\mathcal{S}^j$ denote the set of possible values of Λ corresponding to σ_j. From (10a), (10c), and (10d), we can obtain $\mathcal{S}^j = \{\Lambda^{jk} | 0 \leqslant k \leqslant \mathcal{K}-1, \sigma = \sigma_j\}$, where $\mathcal{K}$ represents the number of possible cases for Λ. Note that Λ^{jk} is a vector, i.e., $\Lambda^{jk} = \left(\lambda_0^{jk}, \lambda_1^{jk}, \ldots \lambda_i^{jk}\right)^T$.

Let e_s denote the symbol error rate and

$$e_s = \sum_{k=1}^{q} \binom{q}{k} \alpha^k (1-\alpha)^{q-k} = 1 - (1-\alpha)^q, \tag{11}$$

where q denotes the number of bits in a symbol. Then let $\phi_w^n(e_s)$ represent the probability of error of weight w in a row, which can be computed by

$$\phi_w^n(e_s) = \binom{n}{w} e_s^w (1-e_s)^{n-w}. \tag{12}$$

As a result, we obtain the probability of wrong decoding of the i-th stage ($1 \leqslant i \leqslant l-1$), as shown in (13).

$$P_{f_i} = \sum_{j=0}^{\chi-1} P_{f_{ij}} = \sum_{j=0}^{\chi-1} \binom{m}{\sigma_j} \left(\sum_{w=t_i^{(C)}+1}^{n} \phi_w^n(e_s) \right)^{\sigma_j} \cdot \sum_{ijk} N_{ijk} \left(\sum_{w=0}^{t_0^{(C)}} \phi_w^n(e_s) \right)^{\lambda_0^{jk}} \left(\sum_{w=t_0^{(C)}+1}^{t_1^{(C)}} \phi_w^n(e_s) \right)^{\lambda_1^{jk}} \cdots \left(\sum_{w=t_{i-2}^{(C)}+1}^{t_{i-1}^{(C)}} \phi_w^n(e_s) \right)^{\lambda_{(i-1)}^{jk}} \left(\sum_{w=t_{i-1}^{(C)}+1}^{t_i^{(C)}} \phi_w^n(e_s) \right)^{\lambda_i^{jk}}, \tag{13}$$

where $P_{f_{ij}}$ denotes the failure probability corresponding to σ_j at the i-th stage, and N_{ijk} denotes the number of combinations for the elements in $\mathcal{S}^j$. To illustrate (13) more intuitively, we give the following example. The example is based on the 1-st stage of EII code $\mathcal{C}(127,(12,12,16,16,16,16,22,22,127,127))$. It can be deduced from (10a)-(10d) that $0 \leqslant \lambda_1 \leqslant 1$, $3 \leqslant \sigma \leqslant 4-\lambda_1$ and $\lambda_0 = 10 - \lambda_1 - \sigma$. Obviously, σ has two possible values: 3 and 4, that is, $\chi = 2$. In such cases, the set of possible values of $(\lambda_0, \lambda_1)^T$ corresponding to σ_j are shown in (14), where $0 \leqslant j \leqslant 1$.

$$\begin{aligned} \mathcal{S}^0 &= \{(\lambda_0^{00}, \lambda_1^{00})^T, (\lambda_0^{01}, \lambda_1^{01})^T\} = \{(7,0)^T, (6,1)^T\}, \\ \mathcal{S}^1 &= \{(\lambda_0^{10}, \lambda_1^{10})^T\} = \{(6,0)^T\}. \end{aligned} \tag{14}$$

So we obtain

$$\begin{aligned} P_{f_1} = &\binom{10}{3} (\psi_1)^3 \left((\psi_2)^7 (\psi_3)^0 + \binom{7}{6} (\psi_2)^6 (\psi_3)^1 \right) \\ &+ \binom{10}{4} (\psi_1)^4 (\psi_2)^6 (\psi_3)^0, \end{aligned} \tag{15}$$

where $\psi_1 = \sum_{w=9}^{127} \phi_w^{127}(e_s)$, $\psi_2 = \sum_{w=0}^{6} \phi_w^{127}(e_s)$, and $\psi_3 = \sum_{w=7}^{8} \phi_w^{127}(e_s)$.

Considering failure probability of the 0-th stage, the final FER can be computed by using (16).

$$\begin{aligned} P_f = &\sum_{\tau=t_{l-1}^{(V)}+1}^{m} \binom{n}{w} \left(\sum_{w=t_0^{(C)}+1}^{n} \phi_w^n(e_s) \right)^{\tau} \left(\sum_{w=0}^{t_0^{(C)}} \phi_w^n(e_s) \right)^{m-\tau} \\ &+ \sum_{i=1}^{l-1} P_{f_i}. \end{aligned} \tag{16}$$

Fig. 2 shows the decoding performance comparisons between the theoretical and simulation results on the 3-level EII code $\mathcal{C}(127,(12,12,16,16,16,16,22,22,127,127))$. In all simulations, the decoding terminates when at least 20000 failed codewords are collected. As can be seen, the curves of theoretical analysis almost coincide with the curves from simulations, which demonstrates the accuracy of the proposed method.

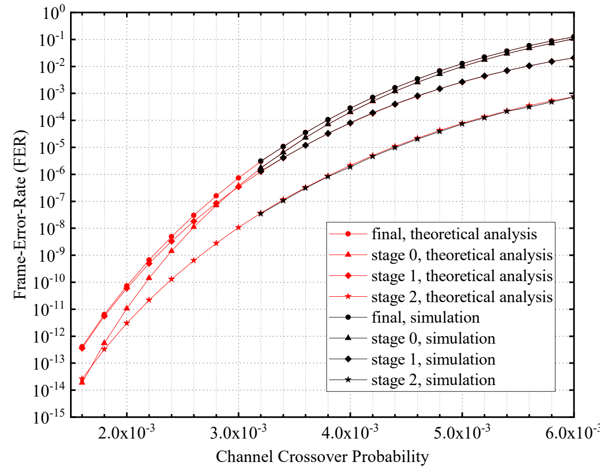

Fig. 2. Comparisons of the failure probability of each stage and the final FER on the $\mathcal{C}(127,(12,12,16,16,16,16,22,22,127,127))$.

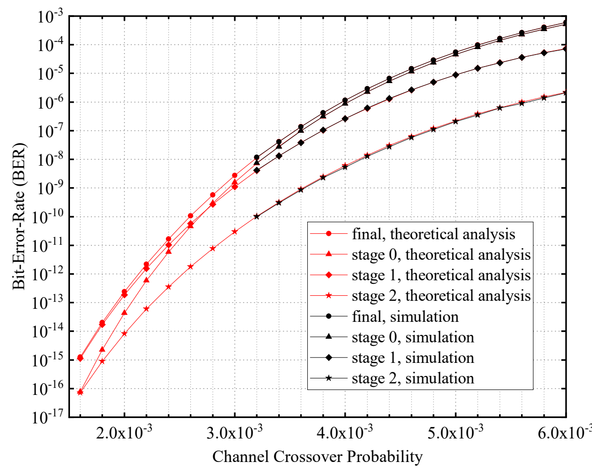

Fig. 3. Comparisons of the BER of each stage and the final BER on the $\mathcal{C}(127,(12,12,16,16,16,16,22,22,127,127))$.

B. BER Analysis

Since BER is the major concern in many applications, this subsection presents a BER analysis method for EII codes. The BER can be calculated by summing up the bit error probabilities of all stages. The average bit error probability of the i-th stage P_{b_i} equals to the number of error bits in a failed codeword of this stage over the total number of bits in an EII codeword. Let B denote average number of error bits in an error symbol, and then P_{b_i} can be calculated by

$$P_{b_i} = \frac{P_{f_i} \cdot R_i \cdot S_i \cdot B}{m \cdot n \cdot q}, \tag{17}$$

where R_i and S_i denote the average number of uncorrected rows in a failed codeword and the average number of error symbols in an uncorrected row, when decoding fails at the i-th stage, respectively.

Utilizing the symbol error rate e_s calculated by (11), B can be expressed by

$$B = \frac{\sum\limits_{k=1}^{q} \binom{k}{q} \alpha^k (1-\alpha)^{q-k} \cdot k}{e_s}. \tag{18}$$

If a row is uncorrectable at the i-th stage, there must involve more than $t_i^{(\mathrm{C})}$ error symbols, where $0 \leqslant i \leqslant l-1$. Therefore, S_i can be calculated through

$$S_i = \frac{\sum\limits_{w=t_i^{(C)}+1}^{n} \phi_w^n(e_s) \cdot w}{\sum\limits_{w=t_i^{(C)}+1}^{n} \phi_w^n(e_s)}. \tag{19}$$

Parameter σ in (10b) represents the number of rows which can not be corrected at the i-th stage, where $1 \leqslant i \leqslant l-1$. As stated in Section III-A, σ has χ possible values. Then we have

$$P_{f_i} \cdot R_i = \sum_{j=0}^{\chi-1} P_{f_{ij}} \cdot \sigma_j. \tag{20}$$

When decoding fails at the 0-th stage, the number of uncorrected rows is uncertain but falls in the range of $t_{l-1}^{(\mathrm{V})}+1$ to m. Therefore, R_0 can be computed using the following equation.

$$R_0 = \frac{\sum\limits_{k=t_{l-1}^{(V)}+1}^{m} \binom{m}{k} \left(\sum\limits_{w=t_0^{(C)}+1}^{n} \phi_w^n(e_s) \right)^k \left(\sum\limits_{w=0}^{t_0^{(C)}} \phi_w^n(e_s) \right)^{m-k} \cdot k}{\sum\limits_{k=t_{l-1}^{(V)}+1}^{m} \binom{m}{k} \left(\sum\limits_{w=t_0^{(C)}+1}^{n} \phi_w^n(e_s) \right)^k \left(\sum\limits_{w=0}^{t_0^{(C)}} \phi_w^n(e_s) \right)^{m-k}}. \tag{21}$$

By substituting (18)-(21) into (17), and then summing up the error probability of each stage, the final BER of the EII codes P_b can be estimated by

$$P_b = \sum_{i=0}^{l-1} P_{b_i} = \frac{\left(P_{f_0} \cdot R_0 \cdot S_0 + \sum\limits_{i=1}^{l-1} \left(\sum\limits_{j=0}^{\chi-1} P_{f_{ij}} \cdot \sigma_j \right) \cdot S_i \right) \cdot B}{m \cdot n \cdot q}. \tag{22}$$

Fig. 3 shows the BER comparisons between the theoretical and simulation results. It can be seen that the theoretical results and the simulation results are nearly identical.

C. Adjustment of Redundancy Distribution

By observing the relationship between the final FER and failure probability of each stage, we can adjust the redundancies of different rows flexibly to maximize the correcting capability of the EII code. Let us use the following example to explain this. The black curves in Fig. 4 show the performance of the 3-level EII code with $\boldsymbol{t}^{(C)} = \{17, 26, 35\}$. Obviously, the FER of the whole codeword is nearly the same as that of the 0-th stage when $\alpha < 0.0013$, which means the decoding of most frames either fails at the 0-th stage or finally succeeds. Since the decoding of the 1-st stage shows a better performance than other stages in most cases, we can

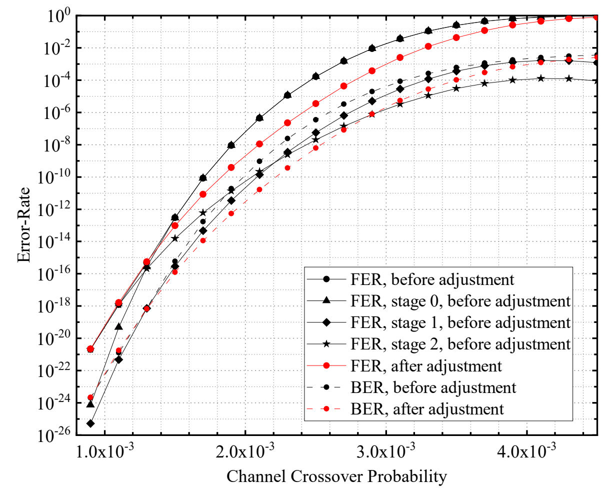

Fig. 4. Performance comparisons between $\boldsymbol{t}^{(C)} = \{17, 26, 35\}$ and $\boldsymbol{t}^{(C)} = \{19, 24, 35\}$.

enhance the error correction capability of code $\mathcal{V}_0$ by reducing the parity symbols of code $\mathcal{V}_1$. Then let $\boldsymbol{t}^{(C)} = \{19, 24, 35\}$ and its performances are shown by the red curves in Fig. 4. As can be seen, the EII code obtains noticeable performance improvement when $\alpha < 0.0013$, while showing no performance loss when $\alpha > 0.0013$. Moreover, the overhead remains unchanged. Although both sets of parameters satisfy that output BER is lower than 10^{-15} when input BER is 1.5×10^{-3}, the adjusted parameters are much better. Therefore, the proposed method guides the code design effectively.

IV. Method for Regular EII Codes

We can simplify the equations of FER analysis for regular EII codes based on the characteristic: $t_{l-i}^{(\mathrm{V})} = t_{l-i-1}^{(\mathrm{V})} + 1$, for $1 \leqslant i \leqslant l-1$. Considering the inequality (10a), it can be concluded that $\lambda_i = 0$. Furtherly, we can know that the parameter σ allows of only one value, that is, $\chi = 1$ and $\sigma = t_{l-i}^{(\mathrm{V})}$. Hence, (10a)-(10d) can be simplified to

$$\begin{cases} \sum\limits_{\tau=0}^{i-1} \lambda_\tau = m - t_{l-i}^{(V)}, & \text{(23a)} \\ \sum\limits_{\tau=0}^{r} \lambda_\tau \geqslant m - t_{l-r-1}^{(V)}, \text{ for } 0 \leqslant r \leqslant i-2. & \text{(23b)} \end{cases}$$

Furthermore, (13) can be reducible to (24).

$$P_{f_i} = \binom{m}{\sigma} \left(\sum_{w=t_i^{(C)}+1}^{n} \phi_w^n(e_s) \right)^{\sigma} \cdot \sum_{ik} N_{ik} \left(\sum_{w=0}^{t_0^{(C)}} \phi_w^n(e_s) \right)^{\lambda_0^k} \cdots \left(\sum_{w=t_{i-2}^{(C)}+1}^{t_{i-1}^{(C)}} \phi_w^n(e_s) \right)^{\lambda_{i-1}^k}. \tag{24}$$

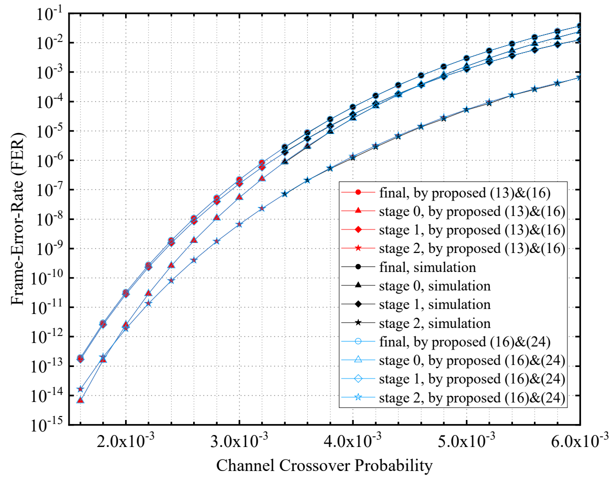

Fig. 5. Comparisons of the failure probability of each stage and the final FER on the $\mathcal{C}(127,(14,14,16,16,22,22,127,127))$.

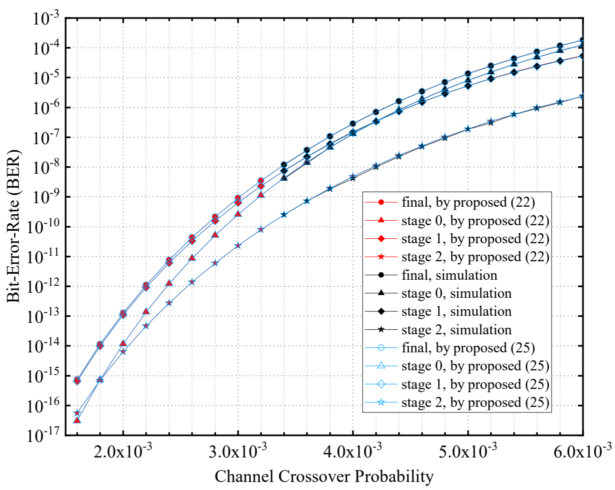

Fig. 6. Comparisons of the BER of each stage and the final BER on the $\mathcal{C}(127,(14,14,16,16,22,22,127,127))$.

Since $\sigma = t_{l-i}^{(\mathrm{V})}$ when $1 \leqslant i \leqslant l-1$, the BER calculation can also be simplified, as shown in (25).

$$P_b = B\frac{P_{f_0}\cdot R_0 \cdot S_0 + \sum_{i=1}^{l-1} P_{f_i}\cdot t_{l-i}^{V}\cdot S_i}{m\cdot n\cdot q}. \tag{25}$$

Comparisons of the FER and BER between the simplified method and simulation results on the 3-level EII code $\mathcal{C}(127,(14,14,16,16,22,22,127,127))$ are shown in Fig. 5 and Fig. 6. Note that $\boldsymbol{t}^{(V)}$ of this codeword are $\{3,2,1\}$. It can be observed that the blue curves are completely coincident with red curves and almost coincide with black curves, which demonstrates the effectiveness of our simplification.

V. Conclusion

In this paper, we present a general approach for performance analysis of EII codes, which can well predict the decoding performance and benefit the code construction process. The decoding is considered as a stagewise process, and the error rate of an l-level EII code can be regarded as the sum of the failure probabilities of l mutually exclusive events. The comparisons between the theoretical and simulation results demonstrate the high accuracy of our method. Besides, a simplified method for regular EII codes is introduced, which reduces the analysis complexity.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62174084, 62104097 and in part by the High-Level Personnel Project of Jiangsu Province under Grant JSSCBS20210034, the Key Research Plan of Jiangsu Province of China under Grant BE2019003-4. (Keyue Deng and Xinyuan Qiao contributed equally to this work.) (Corresponding author: Zhongfeng Wang.)

References

[1] Y. J. Tang and X. Zhang, "An efficient parallel architecture for resource-shareable Reed-Solomon encoder," in *2021 IEEE International Workshop on Signal Processing Systems (SiPS)*, Coimbra, Portugal, Oct. 2021, pp. 152–157.

[2] X. Zhang, S. Sprouse, and I. Ilani, "A flexible and low-complexity local erasure recovery scheme," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2129–2132, Nov. 2016.

[3] C. Kim and J. No, "New constructions of binary LRCs with disjoint repair groups and locality 3 using existing LRCs," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 406–409, Mar. 2019.

[4] P. Huang, E. Yaakobi, and P. H. Siegel, "Multi-erasure locally recoverable codes over small fields: A tensor product approach," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 2609–2624, May 2020.

[5] 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.

[6] 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.

[7] C. Huang et. al., "Erasure coding in Windows Azure Storage," in *Proc. USENIX Annu. Tech. Conf.*, Boston, MA, USA, Jun. 2012, pp. 15–26.

[8] G. A. Gibson, *Redundant disk arrays: reliable, parallel secondary storage*, Ph.D. thesis, University of California, Berkeley, 1991.

[9] M. Blaum et. al., "Expanded Blaum–Roth codes with efficient encoding and decoding algorithms," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 954–957, Jun. 2019.

[10] "Solid state drive (SSD) requirements and endurance test method," https://www.jedec.org/standards-documents/focus/flash/solid-state-drives, 2017.

[11] Z. Wang, "Super-FEC codes for 40/100 Gbps networking," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2056–2059, Nov. 2012.

[12] I. Zhilin et. al., "High-rate codes for high-reliability data transmission," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 256–260.