# Low-Latency Ordered Statistics Decoding of BCH Codes

Lijia Yang[†], Li Chen[‡]
[†]School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China
[‡]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
Email: yanglj39@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn

2022 IEEE Information Theory Workshop (ITW) | 978-1-6654-8341-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITW54588.2022.9965799

***Abstract*—This paper proposes a low-latency ordered statistics decoding (OSD) algorithm for BCH codes. The OSD latency is mainly caused by Gaussian elimination (GE) that produces a systematic generator matrix of the code. Considering BCH codes is binary subcodes of Reed-Solomon (RS) codes, we show that the BCH codeword candidates can be produced through the systematic generator matrix of the corresponding RS code. The systematic generator matrix of an RS code can be formed by generating the linearly independent RS codewords in parallel, replacing the GE process and enabling a low OSD latency. This paper further proposes a segmented variant that facilitates the decoding by reducing the number of test error patterns (TEPs). Complexity of the proposed OSD is also analyzed. Our simulation results show that the proposed decoding can achieve a similar performance as the conventional OSD, but with a lower decoding complexity. The decoding latency can be reduced over the conventional OSD substantially.**

***Index Terms*—BCH codes, low-latency, subfield subcode, maximum likelihood decoding, ordered statistics decoding**

## I. Introduction

The realization of ultra-reliable low-latency communication (URLLC) requires the support of competent short-to-medium length channel codes. The transmission limit of a finite length coded system has been characterized in [1]. Recent research on short-to-medium length codes has shown that ordered statistics decoding (OSD) of BCH codes can yield a performance that is closed to the transmission limit [2]–[3]. In OSD, the codeword candidates are generated through the re-encoding of test messages that are formed by alternating decisions of the most reliable independent positions (MRIPs) in a codeword. The re-encoding process requires Gaussian elimination (GE) that produces a systematic generator matrix of the code. However, due to the sequential feature of GE, its latency cannot be compromised, which is also a long-standing challenge for OSD [4]. In order to reduce the OSD complexity, several skipping and stopping rules have been proposed in [5]–[8]. They facilitate the decoding by identifying the unpromising test error patterns (TEPs) and the maximum likelihood (ML) codeword candidate within the decoding output list, respectively. They result in skipping the unpromising TEPs, or terminating the decoding earlier. The box-and-match algorithm [9] trades time and space complexity by considering the TEPs of small weights. Moreover, the MRIPs segmentation approach was proposed in [10], dividing the OSD operation into several segments to reduce the decoding complexity. On the other aspect, the multiple information sets generated by randomly biased log-likelihood ratios (LLRs) were proposed in [11]–[12] in order to improve the OSD performance.

However, the GE latency challenge remains, which will be addressed by this work. Since BCH codes are binary subcodes of Reed-Solomon (RS) codes, their codeword candidates can be generated through the corresponding RS codewords, which requires the RS systematic generator matrix. It can be formed by generating the linearly independent RS codewords in parallel, underpinning a low decoding latency. In particular, an $(n, k)$ BCH code is a binary subcode of an $(n, k')$ RS code that is defined over a binary extension field, where $n$ is their codeword length and the dimension of the RS code is greater than that of the BCH code, i.e., $k' > k$. The $k'$ linearly independent RS codewords can be generated in parallel using the Lagrange interpolation polynomials, forming the RS systematic generator matrix. The BCH codeword candidates can be yielded through generating the binary RS codewords by the matrix. In order to further reduce the decoding complexity, a segmented low-latency OSD is further proposed. By segmenting the original TEPs, a near ML decoding performance can still be achieved with less TEPs, resulting in a lower decoding complexity. Complexity of the proposed OSD is analyzed. Our simulation results show that the decoding latency (in microsecond) can be substantially reduced over the conventional OSD. They yield a similar decoding performance as the conventional OSD with a smaller decoding output list, resulting in fewer floating point operations for identifying the most likely codeword from the list.

## II. Preliminaries

### A. Ordered Statistics Decoding

Let $\mathbb{F}_q$ denote a finite field of size $q$, and its extension field is further denoted as $\mathbb{F}_{q^m}$, where $m > 1$. Let $\underline{\boldsymbol{f}} = (f_0, f_1, \ldots, f_{k-1}) \in \mathbb{F}_2^k$ and $\underline{\boldsymbol{c}} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_2^n$ denote the message vector and codeword vector of an $(n, k)$ BCH code, respectively, and $d$ denote its minimum Hamming distance. Its generator matrix $\mathbf{G}$ is a $k \times n$ binary matrix as $\mathbf{G} = [\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{n-1}]$, where $\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{n-1}$ are the column vectors of length $k$. Let us assume that a BCH codeword $\underline{\boldsymbol{c}}$ is transmitted by the use of BPSK modulation as : $0 \mapsto 1$; $1 \mapsto -1$. The modulated symbol sequence is $\underline{\boldsymbol{x}} = (x_0, x_1, \ldots, x_{n-1})$, where $x_j \in \{-1, 1\}$ and $j = 0, 1, \ldots, n-1$. After a memoryless channel, the received symbol sequence is $\underline{\boldsymbol{r}} = (r_0, r_1, \ldots, r_{n-1}) \in \mathbb{R}^n$. Let $\Pr(r_j \mid c_j = 0)$ and $\Pr(r_j \mid c_j = 1)$ denote channel observations of $c_j$, its received LLR is defined as

$$L_j = \ln \frac{\Pr(r_j \mid c_j = 0)}{\Pr(r_j \mid c_j = 1)}. \quad (1)$$

Subsequently, the hard-decision received word $\underline{\boldsymbol{y}} = (y_0, y_1, \ldots, y_{n-1}) \in \mathbb{F}_2^n$ can be obtained. That says if $L_j > 0$, $y_j = 0$; otherwise, $y_j = 1$. Since a greater $|L_j|$ indicates the received information of $c_j$ is more reliable, reliability of the received information for all coded bits can be ordered based on $|L_j|$, yielding a refreshed bit index sequence $j_0, j_1, \ldots, j_{n-1}$. It indicates $|L_{j_0}| \geq |L_{j_1}| \geq \cdots \geq |L_{j_{n-1}}|$. A permuted received word can be further obtained as

$$\underline{\boldsymbol{y}}' = \Pi\left(\underline{\boldsymbol{y}}\right) = \left(y_{j_0}, y_{j_1}, \ldots, y_{j_{n-1}}\right), \quad (2)$$

where $\Pi$ denotes the permutation function. Applying the same permutation to the columns of $\mathbf{G}$ yields

$$\mathbf{G}' = \Pi\left(\mathbf{G}\right) = \left[\boldsymbol{g}_{j_0}, \boldsymbol{g}_{j_1}, \ldots, \boldsymbol{g}_{j_{n-1}}\right]. \quad (3)$$

GE will be performed on $\mathbf{G}'$, reducing columns $\boldsymbol{g}_{j_0}, \boldsymbol{g}_{j_1}, \ldots, \boldsymbol{g}_{j_{k-1}}$ into weight one and yielding a systematic generator matrix as

$$\mathbf{G}'' = \left[\boldsymbol{g}'_{j_0}, \boldsymbol{g}'_{j_1}, \ldots, \boldsymbol{g}'_{j_{n-1}}\right], \quad (4)$$

where columns $\boldsymbol{g}'_{j_0}, \boldsymbol{g}'_{j_1}, \ldots, \boldsymbol{g}'_{j_{k-1}}$ form a $k \times k$ identity submatrix. However, this cannot be achieved if the first $k$ columns are not linearly independent. In this case, a second permutation will be needed, and the GE will be conducted again. This adjustment continues until the first $k$ columns of $\mathbf{G}'$ are linearly independent. Note that if a second permutation is needed, $\underline{\boldsymbol{y}}'$ will also be updated accordingly. Without further mentioning, we assume that the first $k$ columns of $\mathbf{G}'$ have been ensured with this property.

Consequently, after ensuring the first $k$ columns of $\mathbf{G}'$ being linearly independent, the first $k$ positions in $\underline{\boldsymbol{y}}'$ are called the MRIPs and their index set is denoted as $\Upsilon = \{j_0, j_1, \ldots, j_{k-1}\}$. Let $\underline{\boldsymbol{f}} = (y_{j_0}, y_{j_1}, \ldots, y_{j_{k-1}})$ denote a message and $\underline{\boldsymbol{e}}^{(\omega)} = (e_{j_0}^{(\omega)}, e_{j_1}^{(\omega)}, \ldots, e_{j_{k-1}}^{(\omega)}) \in \mathbb{F}_2^k$ denote a TEP that will be used to update $\underline{\boldsymbol{f}}$, where $\omega = 1, 2, \ldots, \sum_{\lambda=0}^{\tau} \binom{k}{\lambda}$. For each $\underline{\boldsymbol{e}}^{(\omega)}$, there are at most $\tau$ nonzero entries. The test messages can be generated by

$$\underline{\boldsymbol{f}}^{(\omega)} = \underline{\boldsymbol{f}} + \underline{\boldsymbol{e}}^{(\omega)}. \quad (5)$$

The corresponding codeword candidate can be generated by

$$\underline{\hat{\boldsymbol{c}}}^{(\omega)} = (\hat{c}_0^{(\omega)}, \hat{c}_1^{(\omega)}, \ldots, \hat{c}_{n-1}^{(\omega)}) = \Pi^{-1}(\underline{\boldsymbol{f}}^{(\omega)} \cdot \mathbf{G}''), \quad (6)$$

where $\underline{\hat{\boldsymbol{c}}}^{(\omega)} \in \mathbb{F}_2^n$ and $\Pi^{-1}$ is the inverse of the permutation function $\Pi$. Let us further define the correlation distance between $\underline{\boldsymbol{y}}$ and $\underline{\hat{\boldsymbol{c}}}^{(\omega)}$ as

$$d(\underline{\boldsymbol{y}}, \underline{\hat{\boldsymbol{c}}}^{(\omega)}) \triangleq \sum_{j: y_j \neq \hat{c}_j^{(\omega)}} |L_j|. \quad (7)$$

A codeword candidate with a smaller correlation distance to $\underline{\boldsymbol{y}}$ is more likely to be the transmitted codeword. Let $S_\omega = \{L_j | y_j = \hat{c}_j^{(\omega)}\}$, elements $L_j$ of $S_\omega$ can be reordered as

$$|L_{\xi_0}| \leq |L_{\xi_1}| \leq \cdots \leq \left|L_{\xi_{(n-d_\omega-1)}}\right|, \quad (8)$$

where $d_\omega$ denotes the Hamming distance between $\underline{\boldsymbol{y}}$ and $\underline{\hat{\boldsymbol{c}}}^{(\omega)}$. The ML criterion is [5]

$$d(\underline{\boldsymbol{y}}, \underline{\hat{\boldsymbol{c}}}^{(\omega)}) \leq \sum_{j=0}^{d-d_\omega-1} \left|L_{\xi_j}\right|. \quad (9)$$

If $\underline{\hat{\boldsymbol{c}}}^{(\omega)}$ satisfies (9), it will be the ML codeword. The OSD decoding can be terminated once the ML codeword is found. Otherwise, the one that yields the smallest correlation distance to $\underline{\boldsymbol{y}}$ will be selected as the decoding output $\underline{\hat{\boldsymbol{c}}}_{\text{opt}}$.

Note that the GE that produces the systematic generator matrix $\mathbf{G}''$ is a sequential process incurring the OSD latency challenge.

### *B. BCH Codes and RS Codes*

The subfield subcode relationship between BCH codes and RS codes is stated as follows.

***Definition 1*** *([13]):* Given two linear block codes $\mathcal{C}$ and $\mathcal{C}'$ of length $n$, they are defined over $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$, respectively. If $\mathcal{C} = \mathcal{C}' \cap \mathbb{F}_q^n$, $\mathcal{C}$ is a subcode of $\mathcal{C}'$ over $\mathbb{F}_q$.

***Lemma 1****([14]):* An $(n, k)$ $t$ error-correcting BCH code defined over $\mathbb{F}_2$ is a subcode of an $(n, k')$ $t$ error-correcting RS code defined over $\mathbb{F}_{2^m}$. Note that, the RS codes are the maximum distance separable (MDS) codes. With the same error correction capacity, the RS code dimension is greater than that of the BCH subcode, i.e., $k' > k$.

## III. Low-Latency Ordered Statistics Decoding

### *A. RS Systematic Generator Matrix*

With the permuted received word $\underline{\boldsymbol{y}}'$ of (2), let us define $\Theta = \{j_0, j_1, \ldots, j_{k'-1}\}$ as the index set of its $k'$ most reliable positions (MRPs), and its complementary set $\Theta^c = \{j_{k'}, j_{k'+1}, \ldots, j_{n-1}\}$. Note that since the OSD is discussed under the binary BCH code paradigm, it is assumed $\underline{\boldsymbol{y}}' \in \mathbb{F}_2^n$. Otherwise, for an RS code, $\underline{\boldsymbol{y}}' \in \mathbb{F}_{2^m}^n$. Picking up the received symbols indexed by $\Theta$, an initial message $\underline{\boldsymbol{u}} = (y_{j_0}, y_{j_1}, \ldots, y_{j_{k'-1}}) \in \mathbb{F}_2^{k'}$ can be formed. We also denote the support of its symbol indices that are realized in $\underline{\boldsymbol{y}}'$ as $\text{supp}(\underline{\boldsymbol{u}}) = \{j_0, j_1, \ldots, j_{k'-1}\}$. With $\underline{\boldsymbol{u}}$, the message polynomial of the $(n, k')$ RS code can be defined as

$$H_{\underline{\boldsymbol{u}}}(x) = \sum_{j \in \text{supp}(\underline{\boldsymbol{u}})} y_j L_j(x), \quad (10)$$

where

$$L_j(x) = \prod_{j' \in \text{supp}(\underline{\boldsymbol{u}}), j' \neq j} \frac{x - \alpha_{j'}}{\alpha_j - \alpha_{j'}} \quad (11)$$

is the Lagrange interpolation polynomial of code locator $\alpha_j$. It enables $L_j(\alpha_j) = 1$, and $L_j(\alpha_{j'}) = 0$ if $j' \neq j$. With code locators $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$, the RS codeword $\underline{\boldsymbol{v}} = (v_0,$

$v_1, \ldots, v_{n-1}) \in \mathbb{F}_{2^m}^n$ can be generated by

$$\underline{\boldsymbol{v}} = (H_{\underline{\boldsymbol{u}}}(\alpha_0), H_{\underline{\boldsymbol{u}}}(\alpha_1), \ldots, H_{\underline{\boldsymbol{u}}}(\alpha_{n-1})). \tag{12}$$

Let us define $k'$ weight-1 messages as $\underline{\boldsymbol{u}}_{j_0} = (1, 0, \ldots, 0)$, $\underline{\boldsymbol{u}}_{j_1} = (0, 1, \ldots, 0)$, $\ldots$, $\underline{\boldsymbol{u}}_{j_{k'-1}} = (0, 0, \ldots, 1)$, respectively. They have the same support as $\underline{\boldsymbol{u}}$, i.e., $\text{supp}(\underline{\boldsymbol{u}}_{j_0}) = \text{supp}(\underline{\boldsymbol{u}}_{j_1}) = \cdots = \text{supp}(\underline{\boldsymbol{u}}_{j_{k'-1}}) = \Theta$. Consequently, the generator matrix of the $(n, k')$ RS code can be generated by

$$\mathbf{G}_{\text{RS}} = \begin{bmatrix} H_{\underline{\boldsymbol{u}}_{j_0}}(\alpha_0) & H_{\underline{\boldsymbol{u}}_{j_0}}(\alpha_1) & \cdots & H_{\underline{\boldsymbol{u}}_{j_0}}(\alpha_{n-1}) \\ H_{\underline{\boldsymbol{u}}_{j_1}}(\alpha_0) & H_{\underline{\boldsymbol{u}}_{j_1}}(\alpha_1) & \cdots & H_{\underline{\boldsymbol{u}}_{j_1}}(\alpha_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ H_{\underline{\boldsymbol{u}}_{j_{k'-1}}}(\alpha_0) & H_{\underline{\boldsymbol{u}}_{j_{k'-1}}}(\alpha_1) & \cdots & H_{\underline{\boldsymbol{u}}_{j_{k'-1}}}(\alpha_{n-1}) \end{bmatrix}, \tag{13}$$

where each row is a codeword of the respective message. Since the $k'$ messages are linearly independent, the $k'$ codewords are also linearly independent. They constitute the generator matrix of the $(n, k')$ RS code. In $\mathbf{G}_{\text{RS}}$, columns $j_0, j_1, \ldots, j_{k'-1}$ form a $k' \times k'$ identity submatrix. Hence, $\mathbf{G}_{\text{RS}}$ is in the systematic form. The row-$i$ column-$j$ entry of $\mathbf{G}_{\text{RS}}$ is

$$H_{\underline{\boldsymbol{u}}_i}(\alpha_j) = \begin{cases} 0, & \text{if } j \in \Theta, j \neq i; \\ 1, & \text{if } j \in \Theta, j = i; \\ \frac{\prod_{j' \in \Theta}(\alpha_j - \alpha_{j'})}{(\alpha_j - \alpha_i)\prod_{j' \in \Theta, j' \neq i}(\alpha_i - \alpha_{j'})}, & \text{if } j \in \Theta^c. \end{cases} \tag{14}$$

Since $\alpha_j \prod_{j'=0, j' \neq j}^{n-1} (\alpha_j - \alpha_{j'}) = 1$, when $j \in \Theta^c$,

$$H_{\underline{\boldsymbol{u}}_i}(\alpha_j) = \frac{\alpha_i \prod_{j' \in \Theta^c} (\alpha_i - \alpha_{j'})}{\alpha_j \,(\alpha_j - \alpha_i) \prod_{j' \in \Theta^c, j' \neq j} (\alpha_j - \alpha_{j'})}. \tag{15}$$

Note that when the code rate is greater than 1/2, $|\Theta^c| < |\Theta|$, and eq. (15) requires less finite field computation.

Note that matrix $\mathbf{G}_{\text{RS}}$ can be generated in parallel, underpinning the low-latency feature of the proposed OSD.

### *B. Generation of BCH Codeword Candidates*

The BCH codeword candidates can be further generated by $\mathbf{G}_{\text{RS}}$. With the initial message $\underline{\boldsymbol{u}} = (y_{j_0}, y_{j_1}, \ldots, y_{j_{k'-1}})$, RS codeword $\hat{\underline{\boldsymbol{v}}}^{(0)} = (\hat{v}_0^{(0)}, \hat{v}_1^{(0)}, \ldots, \hat{v}_{n-1}^{(0)}) \in \mathbb{F}_{2^m}^n$ can be generated by

$$\hat{\underline{\boldsymbol{v}}}^{(0)} = \underline{\boldsymbol{u}} \cdot \mathbf{G}_{\text{RS}}, \tag{16}$$

where $\hat{v}_j^{(0)} = y_j$, $\forall j \in \Theta$. Similar to the OSD that was introduced in Section II. A, let us also define a TEP as $\underline{\boldsymbol{e}}'^{(\omega)} = (e'^{(\omega)}_{j_0}, e'^{(\omega)}_{j_1}, \ldots, e'^{(\omega)}_{j_{k'-1}}) \in \mathbb{F}_2^{k'}$. Subsequently, the test message $\underline{\boldsymbol{u}}^{(\omega)}$ can be generated by

$$\underline{\boldsymbol{u}}^{(\omega)} = \underline{\boldsymbol{u}} + \underline{\boldsymbol{e}}'^{(\omega)}. \tag{17}$$

The corresponding RS codeword $\hat{\underline{\boldsymbol{v}}}^{(\omega)} = (\hat{v}_0^{(\omega)}, \hat{v}_1^{(\omega)}, \ldots, \hat{v}_{n-1}^{(\omega)})$ can be further generated by

$$\begin{aligned} \hat{\underline{\boldsymbol{v}}}^{(\omega)} &= (\underline{\boldsymbol{u}} + \underline{\boldsymbol{e}}'^{(\omega)}) \cdot \mathbf{G}_{\text{RS}} \\ &= \hat{\underline{\boldsymbol{v}}}^{(0)} + \underline{\boldsymbol{e}}'^{(\omega)} \cdot \mathbf{G}_{\text{RS}}, \end{aligned} \tag{18}$$

where $\hat{\underline{\boldsymbol{v}}}^{(\omega)} \in \mathbb{F}_{2^m}^n$. Based on *Lemma 1*, if $\hat{\underline{\boldsymbol{v}}}^{(\omega)} \in \mathbb{F}_2^n$, it is also an $(n, k)$ BCH codeword. The following Theorem shows that this binary assessment can be implemented effectively by knowing $\hat{\underline{\boldsymbol{v}}}^{(0)}$, $\underline{\boldsymbol{e}}'^{(\omega)}$ and $\mathbf{G}_{\text{RS}}$.

***Theorem 2:*** If $\hat{v}_j^{(0)} + \sum_{i \in \Theta, e'^{(\omega)}_i \neq 0} H_{\underline{\boldsymbol{u}}_i}(\alpha_j)$ is binary, for all $j \in \Theta^c$, $\hat{\underline{\boldsymbol{v}}}^{(\omega)}$ is a BCH codeword.

*Proof:* Based on (18), let us define

$$\underline{\boldsymbol{e}}'^{(\omega)} \cdot \mathbf{G}_{\text{RS}} = (\phi_0^{(\omega)}, \phi_1^{(\omega)}, \ldots, \phi_{n-1}^{(\omega)}). \tag{19}$$

The RS codeword symbol $\hat{v}_j^{(\omega)}$ can be determined by

$$\hat{v}_j^{(\omega)} = \hat{v}_j^{(0)} + \phi_j^{(\omega)}. \tag{20}$$

Based on (14), we know if $j \in \Theta$,

$$\phi_j^{(\omega)} = e'^{(\omega)}_j. \tag{21}$$

Since for $j \in \Theta$, $\hat{v}_j^{(0)} \in \{0, 1\}$ and the TEP $\underline{\boldsymbol{e}}'^{(\omega)}$ is also binary. Hence, $\hat{v}_j^{(\omega)} \in \{0, 1\}$, $\forall j \in \Theta$. For the remaining symbols with index $j \in \Theta^c$, based on (14) and (18), we know

$$\hat{v}_j^{(\omega)} = \hat{v}_j^{(0)} + \sum_{i \in \Theta, e'^{(\omega)}_i \neq 0} H_{\underline{\boldsymbol{u}}_i}(\alpha_j). \tag{22}$$

Therefore, if they are binary, codeword $\hat{\underline{\boldsymbol{v}}}^{(\omega)}$ is binary. Based on *Lemma 1*, it is also a BCH codeword. ■

Similar to the conventional OSD, the proposed OSD generates the codeword candidates by numerating the TEPs $\underline{\boldsymbol{e}}'^{(\omega)}$ and re-encoding as in (18). Based on *Theorem 2*, if codeword symbols $\hat{v}_j^{(\omega)} (j \in \Theta^c)$ are binary, $\hat{\underline{\boldsymbol{v}}}^{(\omega)}$ will be a BCH codeword. The correlation distance between $\underline{\boldsymbol{y}}$ and $\hat{\underline{\boldsymbol{v}}}^{(\omega)}$ will be further determined as in (7). Once a codeword candidate $\hat{\underline{\boldsymbol{v}}}^{(\omega)}$ satisfies the ML criterion of (9), $\hat{\underline{\boldsymbol{v}}}^{(\omega)}$ will be selected as the decoding output $\hat{\underline{\boldsymbol{v}}}_{\text{opt}}$ and decoding terminates. Otherwise, the one that yields the smallest correlation distance with $\underline{\boldsymbol{y}}$ will be selected as $\hat{\underline{\boldsymbol{v}}}_{\text{opt}}$.

Since the systematic generator matrix $\mathbf{G}_{\text{RS}}$ can be generated in parallel, it yields a decoding latency advantage over the conventional OSD. Summarizing the above description, the low-latency OSD is shown below as in *Algorithm 1.*

## IV. Segmented Variant

This section further proposes a segmented variant of the proposed OSD, in order to reduce its complexity.

The above description shows that in the OSD, if the number of errors in the MRIPs is not greater than the decoding order $\tau$, the transmitted codeword will be included in the decoding output list. Let $P_{\text{e,OSD}}(\tau)$, $P_{\text{e,ML}}$ and $P_{\underline{\boldsymbol{e}}}(\tau)$ denote the error probability of OSD with an order $\tau$, the error probability of the ML decoding, and the probability that the number of errors in the MRIPs is greater than $\tau$, respectively. They hold

$$P_{\text{e,OSD}}(\tau) \leq P_{\text{e,ML}} + P_{\underline{\boldsymbol{e}}}(\tau). \tag{23}$$

**Algorithm 1** Low-Latency OSD of BCH Codes

```
Input: Received symbol sequence $\underline{\boldsymbol{r}}$, order $\tau$;
Output: $\underline{\hat{\boldsymbol{v}}}_{\text{opt}}$;
Compute the LLRs as in (1), and determine $\underline{\boldsymbol{y}}$;
Define MRPs, $\underline{\boldsymbol{u}}$, and let $d_{\text{min}} = +\infty$;
Generate $\mathbf{G}_{\text{RS}}$ as in (14);
Generate the initial codeword $\underline{\hat{\boldsymbol{v}}}^{(0)}$ as in (16);
For each TEP $\underline{\boldsymbol{e}}'^{(\omega)}$, do
    Test if the codeword $\underline{\hat{\boldsymbol{v}}}^{(\omega)}$ is binary as in (22);
    If $\underline{\hat{\boldsymbol{v}}}^{(\omega)}$ is binary
        Determine $d(\underline{\boldsymbol{y}}, \underline{\hat{\boldsymbol{v}}}^{(\omega)})$ as in (7);
        If $d(\underline{\boldsymbol{y}}, \underline{\hat{\boldsymbol{v}}}^{(\omega)}) < d_{\text{min}}$
            Update $d_{\text{min}} = d(\underline{\boldsymbol{y}}, \underline{\hat{\boldsymbol{v}}}^{(\omega)})$ and $\underline{\hat{\boldsymbol{v}}}_{\text{opt}} = \underline{\hat{\boldsymbol{v}}}^{(\omega)}$;
            If $d(\underline{\boldsymbol{y}}, \underline{\hat{\boldsymbol{v}}}^{(\omega)})$ satisfies the ML criterion of (9)
                Terminate the decoding;
End for
Return $\underline{\hat{\boldsymbol{v}}}_{\text{opt}}$;
```

When

$$\tau \geq \min\left\{\left\lceil \frac{d}{4} - 1 \right\rceil, k\right\}, \tag{24}$$

$P_{\underline{\mathbf{e}}}(\tau) \ll P_{\text{e,ML}}$ [3]. Therefore, if the OSD order is sufficiently large, it can approach the ML decoding performance.

Since the length of TEP $\underline{\boldsymbol{e}}'^{(\omega)}$ is greater than that of $\underline{\boldsymbol{e}}^{(\omega)}$ in the conventional OSD, there are more test errors in $\underline{\boldsymbol{e}}'^{(\omega)}$. They occur in the extra symbol band defined by $\Theta \setminus \Upsilon$. The analysis of [3] shows that if $\tau$ satisfies (24), $P_{\underline{\mathbf{e}}}(\tau)$ becomes negligible. Hence, there is no need to assign an order greater than $\tau$ for the first $k$ positions of MRPs.

With $\underline{\boldsymbol{e}}'^{(\omega)}$, we can partition it into two segments as $\underline{\boldsymbol{e}}'^{(\omega)}_1 = (e'^{(\omega)}_{j_0}, e'^{(\omega)}_{j_1}, \ldots, e'^{(\omega)}_{j_{k-1}})$ and $\underline{\boldsymbol{e}}'^{(\omega)}_2 = (e'^{(\omega)}_{j_k}, e'^{(\omega)}_{j_{k+1}}, \ldots, e'^{(\omega)}_{j_{k'-1}})$, respectively. The proposed OSD can be performed by numerating $\underline{\boldsymbol{e}}'^{(\omega)}_1$ and $\underline{\boldsymbol{e}}'^{(\omega)}_2$, which form a smaller set of TEPs. Let $\tau_1$ and $\tau_2$ denote the segment orders of $\underline{\boldsymbol{e}}'^{(\omega)}_1$ and $\underline{\boldsymbol{e}}'^{(\omega)}_2$, respectively. Similar to the definition of $P_{\underline{\mathbf{e}}}(\tau)$, let $P_{\underline{\mathbf{e}}_1}(\tau_1)$ and $P_{\underline{\mathbf{e}}_2}(\tau_2)$ denote the probabilities of the number of errors in $\underline{\boldsymbol{e}}'^{(\omega)}_1$ is greater than $\tau_1$ and the number of errors in $\underline{\boldsymbol{e}}'^{(\omega)}_2$ is greater than $\tau_2$, respectively. In a memoryless channel, we have

$$P_{\underline{\mathbf{e}}}(\tau) = 1 - (1 - P_{\underline{\mathbf{e}}_1}(\tau_1))(1 - P_{\underline{\mathbf{e}}_2}(\tau_2)). \tag{25}$$

Based on (23), we can obtain the error probability upper bound of the segmented OSD as

$$P_{\text{e,seg-OSD}}(\tau_1, \tau_2) \leq P_{\text{e,ML}} + P_{\underline{\mathbf{e}}_1}(\tau_1) + P_{\underline{\mathbf{e}}_2}(\tau_2) - P_{\underline{\mathbf{e}}_1}(\tau_1)P_{\underline{\mathbf{e}}_2}(\tau_2). \tag{26}$$

Hence, if $\tau_1 \geq \min\{\lceil d/4 - 1 \rceil, k\}$, $P_{\underline{\mathbf{e}}_1}(\tau_1) \ll P_{\text{e,ML}}$, and

$$P_{\text{e,seg-OSD}}(\tau_1, \tau_2) \leq P_{\text{e,ML}} + P_{\underline{\mathbf{e}}_2}(\tau_2). \tag{27}$$

Therefore, if $\tau_1 \geq \min\{\lceil d/4 - 1 \rceil, k\}$ and $\tau_2$ is appropriately chosen such that $P_{\underline{\mathbf{e}}_2}(\tau_2) \ll P_{\text{e,ML}}$, the ML decoding performance can still be approached by the segmented OSD. This segmented variant helps reduce the number of TEPs significantly, resulting in a reduced decoding complexity.

Note that the partition point in the MRIPs can be more flexibly adjusted to achieve a better complexity reduction. But this process remains heuristic. More numerical results on this will be provided in Section VI.

## V. Complexity Analysis

This section analyzes the complexity of the proposed OSD and compares it with the conventional OSD. In the conventional OSD, binary operations and floating point operations are needed. The GE process requires $n \cdot (\min\{n-k, k\})^2$ binary operations. Based on $\mathbf{G}''$, $k \cdot (n-k)$ and $(n-k) \cdot \sum_{\lambda=1}^{\tau} \lambda \binom{k}{\lambda}$ binary operations are needed to compute $\underline{\hat{\boldsymbol{c}}}^{(0)}$ and the other candidate codewords $\underline{\hat{\boldsymbol{c}}}^{(\omega)}$, respectively. Finally, identifying the decoding output $\underline{\hat{\boldsymbol{c}}}_{\text{opt}}$ requires at most $n \cdot \sum_{\lambda=0}^{\tau} \binom{k}{\lambda}$ floating point operations. In the proposed OSD, the $\mathbb{F}_{2^m}$ finite field operations and floating point operations are needed. In computing the RS systematic generator matrix $\mathbf{G}_{\text{RS}}$ as in (14) or (15), $2n \cdot \min\{n-k', k'\}$ finite field operations are needed. The generation of $\underline{\hat{\boldsymbol{v}}}^{(0)}$ as in (16) requires at most $k' \cdot (n-k')$ finite field operations. Let $N_{j'}$ denote the number of TEPs $\underline{\boldsymbol{e}}'^{(\omega)}$ that yield binary estimated symbols $\hat{v}_j^{(\omega)}$ in $\Theta^c$ after the $j'$th judgement as in *Theorem 2*, where $j' = 0, 1, \ldots, n-k'$. Note that when $j' = 0$, no assessment has been conducted, and $N_0$ is the total number of TEPs, i.e., $N_0 = \sum_{\lambda=0}^{\tau} \binom{k'}{\lambda}$. A BCH codeword will be confirmed after the $n-k'$ positions in $\Theta^c$ have been assessed. Hence, the decoding output list cardinality of the proposed OSD is $N_{n-k'}$. Computing BCH codeword candidates $\underline{\hat{\boldsymbol{v}}}^{(\omega)}$ as in (22) requires at most $\sum_{\lambda=1}^{\tau} \lambda \binom{k'}{\lambda} + \tau \sum_{j'=1}^{n-k'} N_{j'}$ finite field operations. Finally, identifying $\underline{\hat{\boldsymbol{v}}}_{\text{opt}}$ requires at most $nN_{n-k'}$ floating point operations. The above complexity characterizations are summarized as in Table I . It can be seen that complexity of the proposed OSD also depends on $N_{j'}$. More numerical results will be provided in the following section, providing more insight of it.

TABLE I
Complexity of the proposed and the conventional OSDs.

| Algorithms | Operations | Complexity |
|---|---|---|
| OSD ($\tau$) | GE | $n \cdot (\min\{n-k, k\})^2$ |
| | Compute $\underline{\hat{\boldsymbol{c}}}^{(0)}$ | $k \cdot (n-k)$ |
| | Compute $\underline{\hat{\boldsymbol{c}}}^{(\omega)}$ | $(n-k) \cdot \sum_{\lambda=1}^{\tau} \lambda \binom{k}{\lambda}$ |
| | Find $\underline{\hat{\boldsymbol{c}}}_{\text{opt}}$ | $n \cdot \sum_{\lambda=0}^{\tau} \binom{k}{\lambda}$ |
| Low-Lat. OSD ($\tau$) | Compute $\mathbf{G}_{\text{RS}}$ | $2n \cdot \min\{n-k', k'\}$ |
| | Compute $\underline{\hat{\boldsymbol{v}}}^{(0)}$ | $k' \cdot (n-k')$ |
| | Compute $\underline{\hat{\boldsymbol{v}}}^{(\omega)}$ | $\sum_{\lambda=1}^{\tau} \lambda \binom{k'}{\lambda} + \tau \sum_{j'=1}^{n-k'} N_{j'}$ |
| | Find $\underline{\hat{\boldsymbol{v}}}_{\text{opt}}$ | $nN_{n-k'}$ |

## VI. Simulation results

### A. Decoding Performance

Figs. 1 and 2 show the decoding frame error rate (FER) of the (31, 21) and the (63, 45) BCH codes, respectively.

For the segmented low-latency OSD, it is parameterized by $(\tau_1 \mid l, \tau_2)$, where $l$ denotes the length of the first segment. That says $\underline{\boldsymbol{e}}'^{(\omega)}_1 = (e'^{(\omega)}_{j_0}, e'^{(\omega)}_{j_1}, \ldots, e'^{(\omega)}_{j_{l-1}})$ and $\underline{\boldsymbol{e}}'^{(\omega)}_2 = (e'^{(\omega)}_{j_l}, e'^{(\omega)}_{j_{l+1}}, \ldots, e'^{(\omega)}_{j_{k'-1}})$. Performance of the Berlekamp-Massey (BM) decoding [15] and the conventional OSD [3] are presented as benchmarks. The ML decoding performances were obtained in [16]. Our results show that the low-latency OSD performance can approach that of the conventional OSD, but requires a larger decoding order. This is due to the fact that $k' > k$ and $|\Theta| > |\Upsilon|$, more errors will be introduced in the MRPs of the low-latency OSD. However, our results also show that the segmented variant can yield a similar decoding performance with a smaller order.

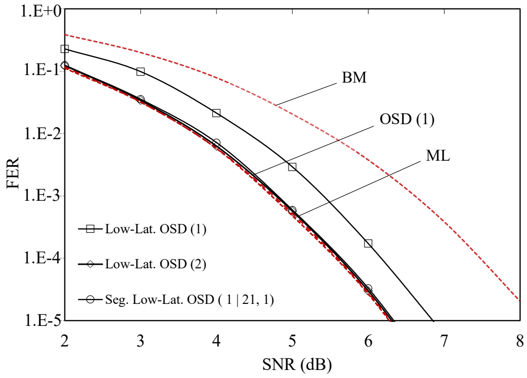

Fig. 1. Performance of the (31, 21) BCH code.

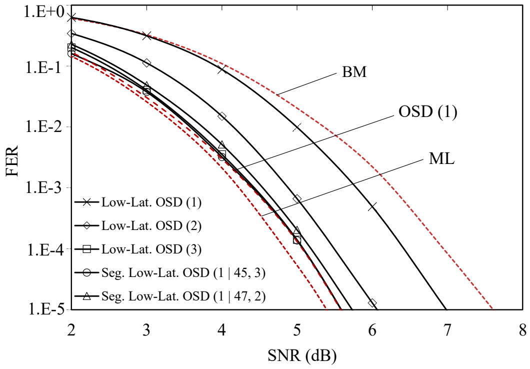

Fig. 2. Performance of the (63, 45) BCH code.

### B. Decoding Complexity and Latency

As pointed out in Section V, complexity of the proposed OSD depends on $N_{j'}$. Table II shows our numerical results of $N_{j'}$ in decoding the (63, 45) BCH code with $\tau = 3$. Note that the BCH code is a binary subcode of the (63, 57) RS code. It can be seen that the assessment of *Theorem 2* can effectively eliminate the nonbinary codewords. E.g., after assessing the first symbol in $\Theta^c$, i.e., $\hat{v}^{(\omega)}_{j_{k'}}$, there are only 957 TEPs that can possibly produce BCH codewords. Moreover, the decoding output list cardinality $N_6$ is only 7, which is far smaller than that of the conventional OSD with $\tau = 1$. This will result in the complexity advantage of the proposed OSDs, as discussed below.

TABLE II
NUMERICAL RESULTS OF $N_{j'}$ IN DECODING THE $(63, 45)$ BCH CODE WITH $\tau = 3$.

| $j'$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| $N_{j'}$ | 30914 | 957 | 36 | 9 | 8 | 7 | 7 |

TABLE III
NUMERICAL RESULTS OF COMPLEXITY AND LATENCY IN DECODING THE $(63, 45)$ BCH CODE.

| Algorithms | SNR | Complexity | | Latency |
|---|---|---|---|---|
| | (dB) | $\mathbb{F}_2/\mathbb{F}_{64}$ oper. | Floating oper. | ($\mu$s) |
| OSD (1) | 4 | $2.78 \times 10^4$ | 81 | $6.58 \times 10^2$ |
| | 5 | $2.60 \times 10^4$ | 19 | $5.34 \times 10^2$ |
| | 6 | $2.56 \times 10^4$ | 8 | $5.06 \times 10^2$ |
| Low-Lat. OSD (3) | 4 | $1.81 \times 10^4$ | 15 | $1.99 \times 10^3$ |
| | 5 | $5.21 \times 10^3$ | 8 | $4.36 \times 10^2$ |
| | 6 | $2.58 \times 10^3$ | 7 | $1.32 \times 10^2$ |
| Seg. Low-Lat. OSD (1 \| 45, 3) | 4 | $3.69 \times 10^3$ | 8 | $2.71 \times 10^2$ |
| | 5 | $2.64 \times 10^3$ | 7 | $1.44 \times 10^2$ |
| | 6 | $2.45 \times 10^3$ | 7 | $1.17 \times 10^2$ |

Table III compares the complexity and latency in decoding the $(63, 45)$ BCH code. All OSDs will terminate once an ML codeword is identified by (9). The decoding complexity and latency are measured and averaged as in decoding one codeword. Referring to Fig. 2, to achieve the same decoding performance, the number of finite field operations in the two proposed OSDs are smaller than that of the binary operations in the conventional OSD, especially segmented variant. Despite the proposed OSDs incur more TEPs, Table II shows that the binary codeword assessment of *Theorem 2* helps eliminate the redundant ones effectively, resulting in a relatively low level of finite field operations. This assessment also results in fewer floating point operations required by the ML criterion. Finally, Table III also vindicates the latency advantage of the proposed OSDs. Our simulations were performed with the Intel core i7-10710U CPU. In the proposed OSDs, each row of $\mathbf{G}_{\text{RS}}$ is generated in parallel. In all OSDs, the TEPs are decoded in a serial manner. It can be seen that both the low-latency OSD and its segmented variant can effectively reduce the decoding latency over the conventional OSD, vindicating the latency advantage of our proposed OSDs.

## ACKNOWLEDGEMENT

This work is sponsored by National Natural Science Foundation of China (NSFC) with project ID 62071498.

## REFERENCES

[1] Y. Polyanskiy, H. V. Poor, and S. Verdu, “Channel coding rate in the finite blocklength regime,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[2] M. C. Coşkun *et al.*, “Efficient error-correcting codes in the short blocklength regime,” *Phys. Commun.*, vol. 34, pp. 66–79, 2019.

[3] M. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[4] C. Choi and J. Jeong, “Fast soft decision decoding algorithm for linear block codes using permuted generator matrices,” *IEEE Communications Letters*, vol. 25, no. 12, pp. 3775–3779, 2021.

[5] T, Kaneko *et al.*, “An efficient Maximum-Likelihood decoding algorithm for linear block codes with algebraic decoder,” *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 320–327, 1994.

[6] Y. Wu and C. N. Hadjicostis, “Soft-decision decoding using ordered recodings on the most reliable basis,” *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 829–836, 2007.

[7] C. Yue *et al.*, “A revisit to ordered statistics decoding: distance distribution and decoding rules,” *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4288–4337, 2021.

[8] W. Jin and M. Fossorier, “Probabilistic sufficient conditions on optimality for reliability based decoding of linear block codes,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July. 2006, Seattle, WA, USA.

[9] A. Valembois and M. Fossorier, “Box and Match techniques applied to soft-decision decoding,” *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 796–810, 2004.

[10] S. E. Alnawayseh and P. Loskot, “Ordered statistics-based list decoding techniques for linear binary block codes,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2012, no. 1, pp. 1–12, Dec.2012.

[11] W. Jin and M. P. C. Fossorier, “Reliability-based soft-decision decoding with multiple biases,” *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 105–120, 2007.

[12] M. Fossorier, “Reliability-based soft-decision decoding with iterative information set reduction,” *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3101–3106, 2002.

[13] E. Berlekamp, “Algebraic coding theory,” *New York, NY, USA:McGraw-Hill*, 1968.

[14] V. Guruswami and A. Rudra, “Limits to list decoding Reed-Solomon codes,” *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3642–3649, 2006.

[15] J. Massey, “Shift-register synthesis and BCH decoding,” *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[16] Helmling *et al.*, “Database of channel codes and ML simulation results,” www.uni-kl.de/channel-codes, 2019.