List Decoding Techniques for Intersymbol Interference Channels Using Ordered Statistics

Fabian Lim, *Student Member, IEEE,* Aleksandar Kavčić, *Senior Member, IEEE,* and Marc Fossorier, *Fellow, IEEE*

***Abstract*—In this paper, we present a generalization of the ordered statistics decoding (OSD) techniques for the class of intersymbol interference (ISI) channels, and show decoding results for the extended Bose-Chaudhuri-Hocquenghem (eBCH) $[128, 64, 22]$ code and the $[255, 239, 17]$ Reed-Solomon (RS) binary image, over the PR2 partial response channel. Using the generalized OSD technique, we go on to generalize the Box-and-Match Algorithm (BMA) to the class of ISI channels. The BMA is an enhancement of OSD, and prior work has shown it to provide significant performance gain over OSD for memoryless additive white Gaussian noise (AWGN) channels. We present decoding results of the BMA for ISI channels, for the same eBCH and RS (binary image) codes, and PR2 channel. Our results show that the BMA (generalized for ISI channels) is superior to the OSD in terms of its performance/complexity trade-off. More specifically, the BMA may be tuned such that both algorithms have similar complexity, whereby the BMA still outperforms the OSD by a significant margin.**

***Index Terms*—linear block codes, ordered statistics decoding, intersymbol interference, Reed-Solomon codes**

I. Introduction

Ordered statistics decoding (OSD) is a reliability-based decoding technique [1] that is applicable to binary linear block codes [2]. As the name suggests, the utilization of OSD techniques involves ordering the received symbol positions by their channel reliabilities [1], [2]. Hypotheses are made on the symbol errors that may occur over the reliable symbol positions [2], and such a strategy is feasible because the number of reliable symbols in error are typically few [2]. OSD is a list decoding technique that forms a list of candidate codewords and picks the best codeword in the list. Each error hypothesis (that a certain set of reliable symbol positions contain symbol errors) corresponds to a codeword in the list. A correct hypothesis will result in having the transmitted codeword placed into the list. Hence, the OSD will decode correctly if the transmitted codeword is in the list, and if all codewords (other than the transmitted codeword) in the list have lower likelihoods then the transmitted codeword [1], [2].

In attempts to improve the performance of OSD-type algorithms, enhancements to the OSD algorithm (see for example [3], [4]) have been proposed. OSD-type techniques have been shown to achieve near maximum-likelihood decoding performance for the extended Bose-Chaudhuri-Hocquenghem (eBCH) $[128, 64, 22]$ code [3], as well as the $[255, 239, 17]$ Reed-Solomon (RS) binary image [4], transmitted over the additive white Gaussian noise (AWGN) channel. However, most OSD techniques found in the literature are not designed to work for intersymbol interference (ISI) channels. Our aim is to extend OSD techniques to the class of ISI channels.

In magnetic recording, the process of writing (and reading back) information on a hard disk, is typically modeled using ISI channels [5]. In perpendicular recording, ISI is known to cause attenuation of high frequency signals that are being written (i.e., the perpendicular recording channel is in fact a low-pass filtering system) [5]. Because the magnetic recording channel possesses memory (due to the presence of ISI), it is typically more difficult to decode error correction codes when used in such systems. For example, Low Density Parity Check (LDPC) codes require *turbo equalization* iterations between the channel detector and LDPC decoder [5], which are not required if the channel is memoryless. Similarly, OSD techniques are more difficult to apply (and do not trivially extend) to ISI channels. In this paper, we begin by first showing how the OSD can be generalized for ISI channels. We then build on the theory of OSD for ISI channels to further generalize a more powerful OSD-like technique (known as the Box-and-Match Algorithm (BMA)) to be applicable to ISI channels. The algorithms presented in this paper collapse down to OSD methods for memoryless channels [2], [3], if the channel impulse response is set to the Kronecker delta function. Hence, the methods given in this paper are indeed generalizations of those in [2], [3].

The paper is organized as follows. In Section II, we present the relevant background material. In Section III, we develop OSD techniques for ISI channels, and present decoding results for the eBCH $[128, 64, 22]$ code, and the $[255, 239, 17]$ RS binary image, over the PR2 channel. In Section IV, we present the BMA for ISI channels, as well as decoding results for the same codes and channel that were shown in Section III. Finally, we conclude in Section V.

II. Background

We first provide some basic definitions, and some simple formulas. The probability of a random event A is denoted as $\Pr\{A\}$. The special constants n, k and $d_{\min}$, represent codelength, dimension, and minimum distance, respectively. The notation for sequences and column vectors are used interchangeably (i.e. $\mathbf{a} = [a_0, a_1, \cdots, a_{n-1}]^T$ may represent

F. Lim and A. Kavčić are with the Department of Electrical Engineering, University of Hawaii at Manoa, 2540 Dole Street Honolulu, HI 96822, USA. (e-mail: {flim},{alek}@hawaii.edu)

M. Fossorier is affiliated with ETIS ENSEA/UCP/CNRS UMR-8051, 6 avenue du Ponceau, 95014, Cergy Pontoise, France. (e-mail: mfossorier@ieee.org)

The work was supported by INSIC and by NSF under grant ECCS 07-25649. The material in this paper has been presented in part at the 2008 INTERMAG conference.

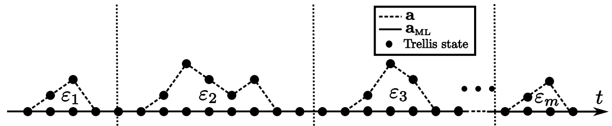


Fig. 1. Decomposition of input sequence pair $(\mathbf{a}, \mathbf{a}_{\text{ML}})$.

both a vector[1] and/or sequence). We use $*$ to represent sequence convolution, and $||\cdot||$ denotes the vector norm. Let $\mathcal{F}_2$ denote the binary field and $s : \mathcal{F}_2 \mapsto \{-1, 1\}$ denotes the binary-to-bipolar mapping $s(a) = (-1)^a$. The mapping s acts component-wise when applied to vectors.

A codeword $\mathbf{c}$ of an $[n, k, d_{\text{min}}]$ binary linear code $\mathcal{C}$ is transmitted through an ISI channel whose channel impulse response is $\mathbf{h} = [h_0, h_1, \cdots, h_I]^T$. The constant I is said to be the *memory length* of the ISI channel. The vector space of dimension n over $\mathcal{F}_2$ is written as $\mathcal{F}_2^n$. Let the sequences $\mathbf{a} \in \mathcal{F}_2^n$ and $\mathbf{r}$, be the ISI channel input, and output (received), respectively. The input-output relationship of the ISI channel is expressed using the equation

$$\mathbf{r} = s(\mathbf{a}) * \mathbf{h} + \mathbf{w}, \quad \text{or (component-wise)},$$
$$r_t = \sum_{i=0}^{I} s(a_{t-i}) \cdot h_i + w_t, \tag{1}$$

where $\mathbf{w}$ is a realization of white Gaussian noise, and t is a discrete time index. The signal-to-noise (SNR) ratio of the channel is defined as $10 \log_{10} \left((\sum_{i=0}^{I} h_i^2) / E\{w_1^2\} \right)$, where $E\{A\}$ denotes the expectation of the random variable A. The most likely sequence is

$$\mathbf{a}_{\text{ML}} = \arg \min_{\mathbf{a} \in \mathcal{F}_2^n} ||\mathbf{r} - s(\mathbf{a}) * \mathbf{h}||^2, \tag{2}$$

and can be computed by a Viterbi detector [6].

Definition 1. *We define the* ***metric discrepancy*** $M(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *of a signal sequence* $\mathbf{a}$*, with respect to the most likely sequence* $\mathbf{a}_{\text{ML}}$ *(see (2)), as*

$$\begin{aligned} M(\mathbf{a}, \mathbf{a}_{\text{ML}}) &\triangleq ||\mathbf{r} - s(\mathbf{a}) * \mathbf{h}||^2 - ||\mathbf{r} - s(\mathbf{a}_{\text{ML}}) * \mathbf{h}||^2 \\ &= ||\mathbf{e} * \mathbf{h}||^2 - 2(\mathbf{r} - s(\mathbf{a}_{\text{ML}}) * \mathbf{h})^T (\mathbf{e} * \mathbf{h}), \end{aligned} \tag{3}$$

where $\mathbf{e} = s(\mathbf{a}) - s(\mathbf{a}_{\text{ML}}) \in \{-2, 0, 2\}^n$.

Remark 1. *The metric discrepancy* $M(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *is simply an inverse measure of the* ***likelihood*** *of the signal sequence* $\mathbf{a}$*. It can be verified that* $M(\mathbf{a}, \mathbf{a}_{\text{ML}}) \leq M(\mathbf{a}', \mathbf{a}_{\text{ML}})$ *implies that* $\mathbf{a}$ *has a higher likelihood than* $\mathbf{a}'$*. Also, note that the metric discrepancy* $M(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *is non-negative (i.e.,* $M(\mathbf{a}, \mathbf{a}_{\text{ML}}) \geq 0$ *for any* $\mathbf{a} \in \mathcal{F}_2^n$*).*

It is well-known that a binary ISI channel with memory length I, can be represented using a channel *trellis* of state space size 2^I, see [5]. Let $S_t(\mathbf{a})$ be the state reached at time t when the transmitted sequence is $\mathbf{a}$, i.e., $S_t(\mathbf{a}) \triangleq [a_{t-I+1}, a_{t-I+2}, \cdots, a_t]^T$. Figure 1 shows a diagram of two state sequences $S_t(\mathbf{a})$ and $S_t(\mathbf{a}_{\text{ML}})$, corresponding to the two input sequences $\mathbf{a}$ and $\mathbf{a}_{\text{ML}}$, respectively. Thereby, in Figure 1 we assume that the trellis has 4 states (i.e. $I = 2$).

[1]All row/column indices for matrices start from 0 in this paper.

Definition 2. *The input sequence pair* $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *is said to be an* ***error event****, if the set* $\{t | S_t(\mathbf{a}) \neq S_t(\mathbf{a}_{\text{ML}})\}$ *is compact*[2]*. If* $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *is an error event we explicitly denote it as* $\varepsilon = (\mathbf{a}, \mathbf{a}_{\text{ML}})$ *and we define* $M(\varepsilon) \triangleq M(\mathbf{a}, \mathbf{a}_{\text{ML}})$*.*

Definition 3. *Two input sequence pairs* $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *and* $(\mathbf{a}', \mathbf{a}_{\text{ML}})$ *are said to be* ***disjoint*** *if* $\{t | S_t(\mathbf{a}) \neq S_t(\mathbf{a}_{\text{ML}})\} \cap \{t | S_t(\mathbf{a}') \neq S_t(\mathbf{a}_{\text{ML}})\} = \emptyset$*.*

Definition 4. *For* ***disjoint*** *input sequence pairs* $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *and* $(\mathbf{a}', \mathbf{a}_{\text{ML}})$*, we define their* ***combination*** $(\mathbf{a}, \mathbf{a}_{\text{ML}}) \circ (\mathbf{a}', \mathbf{a}_{\text{ML}}) \triangleq (\mathbf{a} + \mathbf{a}' - \mathbf{a}_{\text{ML}}, \mathbf{a}_{\text{ML}})$*.*

Remark 2. *Note that if* $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ *and* $(\mathbf{a}', \mathbf{a}_{\text{ML}})$ *are disjoint, then the metric discrepancy (see (3)) satisfies* $M((\mathbf{a}, \mathbf{a}_{\text{ML}}) \circ (\mathbf{a}', \mathbf{a}_{\text{ML}})) = M(\mathbf{a}, \mathbf{a}_{\text{ML}}) + M(\mathbf{a}', \mathbf{a}_{\text{ML}})$*.*

Remark 3. *When the channel is memoryless, an error event is a single symbol error by definition, and all error events are disjoint.*

In this paper, for notational convenience we sometimes denote any (arbitrary) input sequence pair $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ using the symbol x, i.e. $x = (\mathbf{a}, \mathbf{a}_{\text{ML}})$. It was shown in [7] that any input sequence pair $x = (\mathbf{a}, \mathbf{a}_{\text{ML}})$ can be *uniquely* decomposed as $x = (\mathbf{a}_{\text{ML}}, \mathbf{a}_{\text{ML}}) \circ \varepsilon_1 \circ \varepsilon_2 \circ \cdots \circ \varepsilon_m$, for some constant $m \geq 0$, such that all ε_i are distinct (see Figure 1).

Definition 5. *For any input sequence* $x = (\mathbf{a}_{\text{ML}}, \mathbf{a}_{\text{ML}}) \circ \varepsilon_1 \circ \varepsilon_2 \circ \cdots \circ \varepsilon_m$ *(for some* $m \geq 0$*), we define the* ***set of error events*** $\mathfrak{E}(x) \triangleq \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m\}$*. We also define the* ***size*** *of* x *as* $|x| \triangleq |\mathfrak{E}(x)|$*.*

Let p_t be the *a-posteriori* probability [5], that a binary *zero* was sent in the t-th symbol position, given the entire received vector $\mathbf{r}$, i.e., $p_t = \Pr\{A_t = 0 | \mathbf{R} = \mathbf{r}\}$ (see (1)). In the literature (see for example [5]), the value $|\log(p_t/(1 - p_t))|$ is commonly termed the *reliability* of the t-th symbol. The a-posteriori probabilities p_t need to be computed by the BCJR algorithm [8] or estimated by a *Soft Output Viterbi Algorithm* (SOVA) [6]. For memoryless Gaussian channels, the t-th symbol reliability $|\log(p_t/(1 - p_t))|$ is proportional to $|r_t|$.

Remark 4. *For the memoryless channel (i.e.,* $\mathbf{h} = [1]$*),*

$$\begin{aligned} M(\mathbf{a}) = &\ ||\mathbf{r} - s(\mathbf{a})||^2 - ||\mathbf{r} - s(\mathbf{a}_{\text{ML}})||^2 \\ = &\ 2 \sum_t - (s(a_t) - s(a_{ML,t}))\, r_t = 4 \sum_{t\,:\,e_t \neq 0} |r_t|, \end{aligned} \tag{4}$$

In [2], [3], the quantity (4) is (after an appropriate scaling) referred to as the ***discrepancy****. Thus, by our definition of the metric discrepancy in (3), we generalize for ISI channels the terminology that was already developed in [2], [3] for memoryless channels.*

III. OSD FOR ISI CHANNELS

Let $\mathbf{G}$ be the generator matrix of the $[n, k, d_{\text{min}}]$ linear block code $\mathcal{C}$ and let $\mathbf{g}_i$ be the ith column of $\mathbf{G}$. Let $\mathcal{B}$ be the

[2]A set of integers $\mathcal{T}$ is compact if $\mathcal{T} = \{t | \min \mathcal{T} \leq t \leq \max \mathcal{T}\}$, i.e. the set $\mathcal{T}$ contains all the integers that lie between (and include) $\min \mathcal{T}$ and $\max \mathcal{T}$.

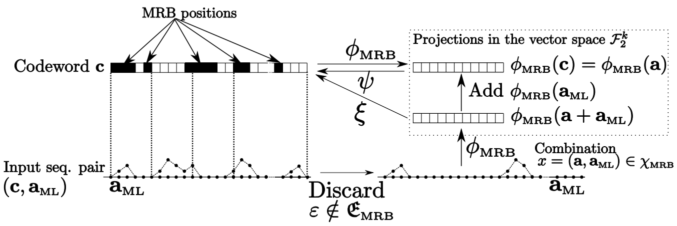


Fig. 2. Diagrammatic representation of the mappings $\phi_{\mathrm{MRB}}, \psi$, and the MRB error map ξ.

collection of k-subsets of n indices (i.e., $\beta \subseteq \{0, 1, \cdots, n-1\}$, and $|\beta| = k$), such that for any $\beta \in \mathcal{B}$, the set of k columns $\{\mathbf{g}_i | i \in \beta\}$ are *linearly independent*.

Definition 6. *The **projection** of a binary sequence $\mathbf{a}$ onto a set of indices $\beta = \{i_1, i_2, \cdots, i_{|\beta|}\}$, is defined as $\phi_\beta(\mathbf{a}) \triangleq [a_{i_1}, a_{i_2}, \cdots, a_{i_{|\beta|}}]^T$. The projection is a **linear** map from $\mathcal{F}_2^n \mapsto \mathcal{F}_2^{|\beta|}$ (i.e., $\phi_\beta(\mathbf{a}+\mathbf{a}') = \phi_\beta(\mathbf{a})+\phi_\beta(\mathbf{a}')$ for any $\mathbf{a} \in \mathcal{F}_2^n$, and any $\mathbf{a}' \in \mathcal{F}_2^n$).*

Definition 7. *We define the set of **most reliable basis (MRB) [3] positions** β_{MRB}, as the element of $\mathcal{B}$ that possess the **largest** sum of symbol reliabilities. That is, the set $\beta_{MRB} \in \mathcal{B}$ satisfies*

$$\beta_{MRB} \triangleq \arg\max_{\beta \in \mathcal{B}} \sum_{t \in \beta} \left| \log \frac{p_t}{1 - p_t} \right|. \tag{5}$$

*The **projection** of a binary sequence $\mathbf{a}$ onto **the MRB positions** is defined as $\phi_{MRB}(\mathbf{a}) \triangleq \phi_{\beta_{MRB}}(\mathbf{a})$. From Definition 6, we see that ϕ_{MRB} is a mapping $\phi_{MRB} : \mathcal{F}_2^n \mapsto \mathcal{F}_2^k$ (see Figure 2).*

Remark 5. *For any $\beta \in \mathcal{B}$, because the set of columns $\{\mathbf{g}_i | i \in \beta\}$ are linearly independent, each of the 2^k codewords $\mathbf{c} \in \mathcal{C}$ has a **unique** projection $\phi_\beta(\mathbf{c})$, see [2]. In other words, there exists a one-to-one **reverse**[3] mapping $\psi : \mathcal{F}_2^k \mapsto \mathcal{C}$ (see Figure 2). The terminology MRB is taken from [3], and since $\beta_{MRB} \in \mathcal{B}$, the term "basis" in MRB refers to the uniqueness property of the projection.*

If $\mathbf{c}$ is the transmitted codeword, observe that the projection $\phi_{\mathrm{MRB}}(\mathbf{c} + \mathbf{a}_{\mathrm{ML}})$ equals the error over the MRB positions made by the channel detector. Since the MRB symbol positions are reliable, we expect that $\phi_{\mathrm{MRB}}(\mathbf{c} + \mathbf{a}_{\mathrm{ML}})$ has low weight. Conversely, if some $\mathbf{c} \in \mathcal{C}$ has $\phi_{\mathrm{MRB}}(\mathbf{c} + \mathbf{a}_{\mathrm{ML}})$ of large weight, then it is very likely that $M(\mathbf{c}, \mathbf{a}_{\mathrm{ML}})$ is large, implying that $\mathbf{c}$ has a low likelihood of being transmitted. The OSD is a decoder that considers codewords that have low weight $\phi_{\mathrm{MRB}}(\mathbf{c} + \mathbf{a}_{\mathrm{ML}})$. The following definition is needed to explain the OSD for ISI channels.

Definition 8. *Let $\mathfrak{E}_{\mathrm{MRB}}$ be the set of all error events $\varepsilon = (\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$ where $\phi_{MRB}(\mathbf{a}+\mathbf{a}_{\mathrm{ML}}) \neq \mathbf{0}$ (i.e. any $\varepsilon \in \mathfrak{E}_{\mathrm{MRB}}$ has at least one errornous symbol in the MRB positions). Furthermore, the error events $\varepsilon_{MRB}^{(i)} \in \mathfrak{E}_{\mathrm{MRB}}$ are ordered such that*

$$M(\varepsilon_{MRB}^{(1)}) \leq M(\varepsilon_{MRB}^{(2)}) \leq \cdots \leq M(\varepsilon_{MRB}^{(|\mathfrak{E}_{\mathrm{MRB}}|)}) \tag{6}$$

is satisfied, where the superscripts emphasize that this is not temporal indexing.

[3]Note that the map ψ is *not* the inverse mapping of ϕ_{MRB}.

Algorithm 1: OSD for ISI channels

Input: A subset of MRB error event combinations $\bar{\chi}_{\mathrm{MRB}} \subseteq \chi_{\mathrm{MRB}}$;
Output: Codeword list $\mathcal{L}_{\mathrm{OSD}}$ and best candidate $\mathbf{c}_{\mathrm{opt}} = \arg\min_{\mathbf{c} \in \mathcal{L}_{\mathrm{OSD}}} M(\mathbf{c}, \mathbf{a}_{\mathrm{ML}})$;
Initialize: $\mathcal{L}_{\mathrm{OSD}} := \emptyset$;

forall $x \in \bar{\chi}_{MRB}$ **do**
  Compute the MRB projection $\nu := \varphi_{\mathrm{MRB}}(x)$ of x;
  Compute the candidate codeword $\mathbf{c} := \xi(\nu)$;
  Store $\mathbf{c}$ in the codeword list $\mathcal{L}_{\mathrm{OSD}}$;
end

The OSD is a list decoding algorithm that forms a list $\mathcal{L}_{\mathrm{OSD}} \subseteq \mathcal{C}$ of candidate codewords. Typically $|\mathcal{L}_{\mathrm{OSD}}| \ll |\mathcal{C}| = 2^k$. A list decoding algorithm should try to put the most likely codewords in the decoder list. Since the most likely codewords typically have a small number of error events (w.r.t $\mathbf{a}_{\mathrm{ML}}$) in $\mathfrak{E}_{\mathrm{MRB}}$, the strategy of OSD is to put codewords $\mathbf{c}$ in the list $\mathcal{L}_{\mathrm{OSD}}$ if $|\mathfrak{E}((\mathbf{c}, \mathbf{a}_{\mathrm{ML}})) \cap \mathfrak{E}_{\mathrm{MRB}}|$ is small (recall Definition 5 that $\mathfrak{E}((\mathbf{c}, \mathbf{a}_{\mathrm{ML}}))$ is the set of error events obtained from $(\mathbf{c}, \mathbf{a}_{\mathrm{ML}})$).

Generally we have $|\mathfrak{E}_{\mathrm{MRB}}| \geq k$, and equality is satisfied only when the channel $\mathbf{h}$ is memoryless. It was shown in [7] that for ISI channels the set $\mathfrak{E}_{\mathrm{MRB}}$ can be obtained using a modification[4] of the (serial) *generalized Viterbi algorithm* (GVA) [9], applied to the channel trellis.

Definition 9. *We define χ_{MRB} as the set of all possible combinations of error events in the set $\mathfrak{E}_{\mathrm{MRB}}$. That is,*

$$\chi_{MRB} \triangleq \{x = (\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \,|\, \mathfrak{E}(x) \subseteq \mathfrak{E}_{\mathrm{MRB}}, \mathbf{a} \in \mathcal{F}_2^n\} \tag{7}$$

The lower and upper bound on the size of the set χ_{MRB}, are 2^k and $2^{|\mathfrak{E}_{\mathrm{MRB}}|}$, respectively.

Definition 10. *We extend the definition of the **projection** (see Definitions 6 and 7) to apply to all $x \in \chi_{MRB}$. For any combination $x = (\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \in \chi_{MRB}$, the projection $\varphi_{MRB}(x) : \chi_{MRB} \mapsto \mathcal{F}_2^k$ is defined as $\varphi_{MRB}(x) \triangleq \phi_{MRB}(\mathbf{a} + \mathbf{a}_{\mathrm{ML}})$, see Figure 2.*

For any combination $x \in \chi_{\mathrm{MRB}}$, the projection $\varphi_{\mathrm{MRB}}(x) \in \mathcal{F}_2^k$ should be viewed as a hypothesized error pattern in the MRB. The hypothesized error pattern $\varphi_{\mathrm{MRB}}(x)$ corresponding to x is associated with a codeword $\mathbf{c} \in \mathcal{C}$ via the *MRB error map*.

Definition 11. *Define $\mathbf{c}_0 \in \mathcal{C}$ to be the codeword that satisfies $\phi_{MRB}(\mathbf{c}_0) = \phi_{MRB}(\mathbf{a}_{\mathrm{ML}})$, i.e., $\mathbf{c}_0$ equals the transmitted codeword if the detector makes no symbol errors in the MRB positions.*

Definition 12. *Let $\nu = \varphi_{MRB}(x)$ denote the projection (or error pattern) corresponding to a combination $x = (\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \in \chi_{MRB}$. We define the **MRB error map** $\xi : \mathcal{F}_2^k \mapsto \mathcal{C}$ as $\xi(\nu) \triangleq \psi(\nu) + \mathbf{c}_0$, where ψ is the reverse map of ϕ_{MRB} defined in Remark 5.*

Figure 2 shows that codeword $\mathbf{c}$ associated with $x = (\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \in \chi_{\mathrm{MRB}}$ by the MRB error map ξ, is simply the unique codeword whose projection $\phi_{\mathrm{MRB}}(\mathbf{c}) = \phi_{\mathrm{MRB}}(\mathbf{a})$. To

[4]The modification is required to perform our specific task more efficiently, since we only need to obtain the ordered set of error events $\mathfrak{E}_{\mathrm{MRB}}$ as opposed to the GVA's original task of sorting $M(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$ for all possible signal sequences $\mathbf{a} \in \mathcal{F}_2^n$.

verify that Figure 2 coincides with the definition of ξ in Definition 12, we observe that $\psi\left(\phi_{\text{MRB}}(\mathbf{a}+\mathbf{a}_{\text{ML}})+\phi_{\text{MRB}}(\mathbf{a}_{\text{ML}})\right) = \psi\left(\phi_{\text{MRB}}(\mathbf{a}+\mathbf{a}_{\text{ML}})\right)+\psi(\phi_{\text{MRB}}(\mathbf{a}_{\text{ML}})) = \psi\left(\phi_{\text{MRB}}(\mathbf{a}+\mathbf{a}_{\text{ML}})\right)+\mathbf{c}_0$.

Algorithm 1 gives an implementation of OSD for ISI channels. The input to the OSD algorithm is a subset $\bar{\chi}_{\text{MRB}} \subseteq \chi_{\text{MRB}}$ of error event combinations, the reason being that, the (lower bound on the) size $|\chi_{\text{MRB}}|$ is exponential in the code dimension k (see Definition 9). Note that the size of the list $\mathcal{L}_{\text{OSD}}$ that the algorithm forms equals $|\mathcal{L}_{\text{OSD}}| = |\bar{\chi}_{\text{MRB}}|$. Hence, the choice of the size of $\bar{\chi}_{\text{MRB}}$ limits the size of the list $\mathcal{L}_{\text{OSD}}$. We also have a choice of which elements x to put into $\bar{\chi}_{\text{MRB}}$ and we next discuss the importance of this choice.

Let A_{LIST} be the event that the transmitted codeword is not in $\mathcal{L}_{\text{OSD}}$, and let A_{MLD} be the event that the *maximum likelihood decoder* makes an error. As we can see from Algorithm 1, the probability that OSD makes a decoding error is $\Pr\{A_{\text{LIST}} \cup A_{\text{MLD}}\} \leq \Pr\{A_{\text{LIST}}\} + \Pr\{A_{\text{MLD}}\}$. The problem of constructing the set $\bar{\chi}_{\text{MRB}}$ can thus be formulated as the following: For a given fixed size $|\bar{\chi}_{\text{MRB}}|$, how should the subset $\bar{\chi}_{\text{MRB}} \subseteq \chi_{\text{MRB}}$ be chosen such that $\Pr\{A_{\text{LIST}}\}$ is minimized? The exact answer to this question is not known. However, as we shall see in the next two subsections, reasonable suboptimal solutions are available.

Remark 6. *We say that the set $\bar{\chi}_{MRB}$ has* ***order*** *$\mathcal{O}(\bar{\chi}_{MRB})$ if its size equals $|\bar{\chi}_{MRB}| = \sum_{i=0}^{\mathcal{O}(\bar{\chi}_{MRB})}\binom{k}{i}$. This is in accordance with the notation used in memoryless channels [2], where the sizes $|\bar{\chi}_{MRB}| = |\mathcal{L}_{\text{OSD}}|$ are chosen to be polynomial in the code dimension k. When $\mathcal{O}(\bar{\chi}_{MRB}) = 0$, then the decoded list $\mathcal{L}_{\text{OSD}}$ only contains the single candidate $\mathbf{c}_0$ (see Algorithm 1), and is independent of the choice for $\bar{\chi}_{MRB}$.*

Remark 7. *Techniques for computing $\Pr\{A_{LIST}\}$ (or upper bounds on the bit-error rate of OSD), for certain choices of $\bar{\chi}_{MRB}$, are currently known for the AWGN channel [2], [1], the Rayleigh fading channel [10], and certain wireless channels [11].*

Remark 8. *The composition of the map $\varphi_{MRB} : \chi_{MRB} \mapsto \mathcal{F}_2^k$, and the MRB error map $\xi : \mathcal{F}_2^k \mapsto \mathcal{C}$, is written as the map $\xi\varphi_{MRB} : \chi_{MRB} \mapsto \mathcal{C}$ (see Figure 2). The composite map $\xi\varphi_{MRB}$ is onto but not one-to-one, and as a consequence, the codewords in $\mathcal{L}_{\text{OSD}}$ are not necessarily unique.*

A reasonable way to process elements in χ_{MRB}, would be to process them in decreasing order of their likelihood (or in increasing metric discrepancy). In [12], it is shown that for the memoryless AWGN channel, the performance results are similar if we process the elements in χ_{MRB} using a different ordering. The latter ordering is important from an implementation viewpoint, because it suggests a *fixed* construction of $\bar{\chi}_{\text{MRB}}$ from $\mathfrak{E}_{\text{MRB}}$. Next we consider the corresponding two choices of $\bar{\chi}_{\text{MRB}}$ for ISI channels.

A. The Lexicographical construction of $\bar{\chi}_{MRB}$

Recall from Definition 8 that the error events in the set $\mathfrak{E}_{\text{MRB}}$ are ordered by their metric discrepancy.

Definition 13. *We define the* ***binary (lexicographical) representation*** *$\Xi(x)$ of a combination $x \in \chi_{MRB}$, with decomposition*

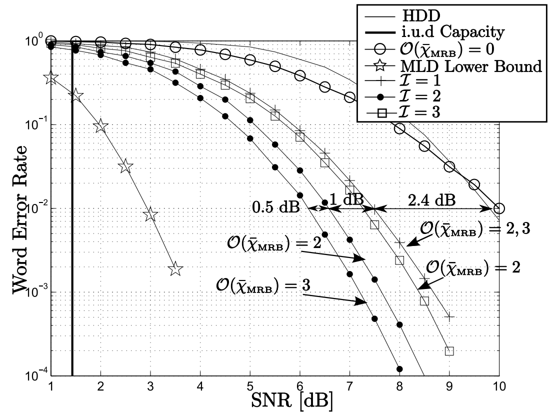


Fig. 3. Performance of the Lexicographical construction of $\bar{\chi}_{\text{MRB}}$ for the $[128, 64, 22]$ eBCH code.

$x = (\mathbf{a}_{\text{ML}}, \mathbf{a}_{\text{ML}}) \circ \varepsilon_{MRB}^{(i_1)} \circ \varepsilon_{MRB}^{(i_2)} \circ \cdots \circ \varepsilon_{MRB}^{(i_{|x|})}$ (see Definition 7), as $\Xi(x) \triangleq 2^{i_1-1} + 2^{i_2-1} + \cdots + 2^{i_{|x|}-1}$. Hence, we can order[5] the error events $x \in \chi_{MRB}$ lexicographically as

$$0 = \Xi(x^{[0]}) \leq \Xi(x^{[1]}) \leq \Xi(x^{[2]}) \leq \cdots \leq \Xi(x^{[|\chi_{MRB}|]}) \tag{8}$$

for all $x^{[i]} \in \chi_{MRB}$. Note for $x^{[0]} \triangleq (\mathbf{a}_{\text{ML}}, \mathbf{a}_{\text{ML}})$, we have $\Xi(x^{[0]}) = 0$.

Remark 9. *Note that the elements in χ_{MRB} are lexicographically ordered regardless of their metric discrepancy, e.g. the element $x = \varepsilon_{MRB}^{(1)} \circ \varepsilon_{MRB}^{(2)}$ always comes before $x' = \varepsilon_{MRB}^{(3)}$ regardless whether $M(x) \leq M(x')$ or $M(x) > M(x')$.*

Definition 14. *We say that $\bar{\chi}_{MRB}$ is* ***lexicographically constructed***[6] *with parameter $\mathcal{I}$ and order $\mathcal{O}(\bar{\chi}_{MRB})$ as follows. We order all combinations $x \in \chi_{MRB}$ that satisfy $|x| \leq \mathcal{I}$ in the increasing order of the lexicographical representation $\Xi(x)$, and populate $\bar{\chi}_{MRB}$ with the first $\sum_{i=0}^{\mathcal{O}(\bar{\chi}_{MRB})}\binom{k}{i}$ combinations in that ordering.*

Figure 3 shows the performance of OSD for the binary eBCH $[128, 64, 22]$ code transmitted over the PR2 channel [5] whose channel response is $\mathbf{h} = [1, 2, 1]^T$. The set $\bar{\chi}_{\text{MRB}}$ is lexicographically constructed using various list orders $\mathcal{O}(\bar{\chi}_{\text{MRB}})$ and parameters $\mathcal{I}$. For purposes of comparison, we also show a lower bound[7] on the performance of the maximum likelihood decoder (MLD). As seen from Figure 3, when we set $\mathcal{I} = 1$ and $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$, we obtained a 2.4 dB gain over hard-decision decoding (HDD), at a word error rate (WER) of 10^{-2}. Furthermore, there is no observable performance gain when we increase $\mathcal{O}(\bar{\chi}_{\text{MRB}})$ to 3. From this observation, it is inferred that the lexicographically constructed set $\bar{\chi}_{\text{MRB}}$ with parameter $\mathcal{I} = 1$ and order $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$, contains most

[5] The square bracketing used in the superscripts in (8) indicate a different ordering than that of metric discrepancy, see (8).

[6] The lexicographical construction of $\bar{\chi}_{\text{MRB}}$ is inspired by OSD literature for memoryless channels [2], [3]. Note that the total number of combinations $x \in \chi_{\text{MRB}}$ that satisfy $|x| \leq \mathcal{I}$ is at most $\sum_{i=0}^{\mathcal{I}}\binom{|\mathfrak{E}_{\text{MRB}}|}{i} \geq \sum_{i=0}^{\mathcal{I}}\binom{k}{i}$, where equality holds if the channel has no memory. Thus, in memoryless channels, the lexicographically constructed set $\bar{\chi}_{\text{MRB}}$ of order $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = \mathcal{I}$, contains all possible combinations $x \in \chi_{\text{MRB}}$ that satisfy $|x| \leq \mathcal{I}$.

[7] Note that the MLD lower bound is computed by empirically counting the number of decoding instances, in which our decoder obtains a codeword that has a higher likelihood than the transmitted codeword. This technique of computing a lower bound on the MLD performance was also used in [13].

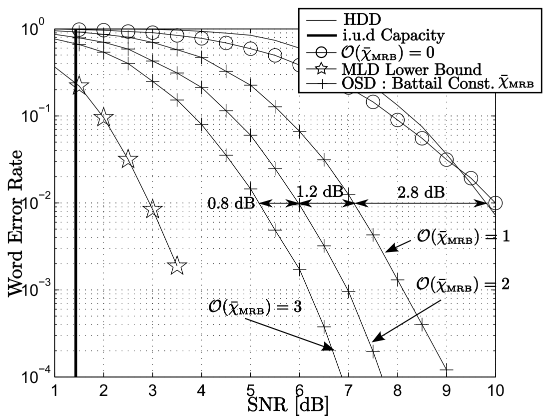


Fig. 4. Performance of the Battail construction of $\bar{\chi}_{\text{MRB}}$ for the $[128, 64, 22]$ eBCH code.

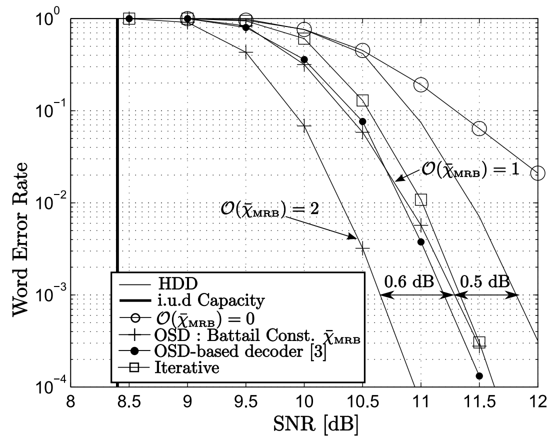


Fig. 5. Performance of the Battail construction of $\bar{\chi}_{\text{MRB}}$ for the RS $[255, 239, 17]$ binary image.

of the likely combinations $x \in \chi_{\text{MRB}}$ that satisfy $|x| \leq 1$. When $\mathcal{I}$ is increased to 2, we obtained additional gains of 1 and 1.5 dB, with the list orders $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$ and 3, respectively. In particular, we observe a 1 dB gain when we fix $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$, and increase the parameter $\mathcal{I}$ from 1 to 2 (the two performances being compared are obtained using the same number of candidates in $\mathcal{L}_{\text{OSD}}$). Finally, we observe a performance loss of about 0.8 dB, when we set $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$ and increase $\mathcal{I}$ from 2 to 3. This observation suggests that combinations $x \in \chi_{\text{MRB}}$ that satisfy $|x| \leq 2$, occur more frequently than those with $|x| = 3$. From all the above observations, we conclude that both parameters $\mathcal{I}$ and $\mathcal{O}(\bar{\chi}_{\text{MRB}})$ affect the performance of OSD when using the lexicographically constructed set $\bar{\chi}_{\text{MRB}}$.

As opposed to the memoryless AWGN case, we shall see that in ISI channels, the lexicographical ordering performs rather poorly when compared to metric discrepancy ordering.

B. The Battail construction of $\bar{\chi}_{MRB}$

While the lexicographical construction only obtains an approximate ordering of the combinations in χ_{MRB} in increasing metric discrepancy, it is possible utilize the Battail's algorithm [14], [15] to obtain the exact ordering (at the expense of increased complexity).

Definition 15. ***Battail's algorithm*** *constructs the set* $\bar{\chi}_{MRB} \subseteq \chi_{MRB}$ *of order* $\mathcal{O}(\bar{\chi}_{MRB})$*, containing the first* $\sum_{i=0}^{\mathcal{O}(\bar{\chi}_{MRB})} \binom{k}{i}$ *combinations of the ordering*

$$0 = M(x^{(0)}) \leq M(x^{(1)}) \leq M(x^{(2)}) \leq \cdots \leq M(x^{(|\chi_{MRB}|)}) \qquad (9)$$

induced by the metric discrepancy. Note that $x^{(0)} \triangleq (\mathbf{a}_{\text{ML}}, \mathbf{a}_{\text{ML}})$.

Other references related to Battail's algorithm (applied to memoryless channels) include [12], [16]. Battail's algorithm for ISI channels was presented in [7] and is given in Appendix A as pseudocode. Studies have shown that for the AWGN channel, the performances obtained when using either the lexicographical or the Battail construction, are very similiar [12]. We attempt to make a similar comparison for ISI channels. Figure 4 shows the performance of the Battail construction, for the $[128, 64, 22]$ eBCH code on the PR2 channel. We observe that when using the Battail construction, the list order $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1$ gives approximately 2.8 dB gain over HDD at WER$= 10^{-2}$. Comparing with Figure 3, we see that when the set $\bar{\chi}_{\text{MRB}}$ is constructed lexicographically, we require larger list orders of $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$ and 3, to obtain a similar gain over HDD. Next we see from Figure 4, that if $\bar{\chi}_{\text{MRB}}$ is obtained using the Battail construction and $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$, the obtained performance is better than all tested cases for the lexicographical construction shown in Figure 3. From Figure 4, the respective gains over HDD if we choose $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$ and 3, are approximately 4 and 5 dB, respectively. We observe diminishing returns as the list order $\mathcal{O}(\bar{\chi}_{\text{MRB}})$ increases.

Next, we test the Battail construction using the RS $[255, 239, 17]$ binary image on the PR2 channel. Figure 5 shows the performance. Observe that for this particular code, the performance of OSD with $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 0$ is worse than HDD (this may occur for high-rate codes). When we set $\mathcal{O}(\bar{\chi}_{\text{MRB}})$ to 1 and 2, the gains over HDD at a WER of 10^{-3}, are 0.5dB and 1.1dB, respectively. For comparison purposes, we show the performance of another OSD-based decoder [3] that was designed for the AWGN (memoryless) channel. This particular OSD-based decoder (with complexity similar to our OSD with $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$) was demonstrated to give a 1.5 dB gain over HDD at WER 10^{-3} on the memoryless AWGN channel [3]. However from Figure 5, we observe that the OSD-based decoder [3] only achieves similar performance as our OSD decoder of $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1$ (which has a much lower complexity). We also compare the performance of a message passing (iterative) decoder proposed by Bellarado-Kavčić [17] for RS codes (which performs similarly to the Jiang-Narayanan decoder [13]). Again we see that the iterative decoder performs similarly to our OSD decoder of order $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1$, achieving a 0.5dB gain from HDD at a WER of 10^{-3}, which is worse than the OSD with $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$.

C. Implementing OSD for ISI Channels

The implementation in Algorithm 1 is presented for clarity of exposition; however it is not the most efficient. Note that Algorithm 1 requires $\bar{\chi}_{\text{MRB}}$ as an input, which implies that $\bar{\chi}_{\text{MRB}}$ must be constructed before Algorithm 1 may run. Furthermore, the construction of the set $\bar{\chi}_{\text{MRB}}$ (either lexicographically (see Definition 14) or by the Battail algorithm (see Appendix A)) requires the set of error events $\mathfrak{E}_{\text{MRB}}$ to be known. Thus if both sets $\bar{\chi}_{\text{MRB}}$ and $\mathfrak{E}_{\text{MRB}}$ are large, it is clear that the implementation of OSD for ISI channels given in Algorithm 1, comes with a huge cost in computational effort and latency.

An alternate implementation is given in [7], where the construction of $\mathfrak{E}_{\text{MRB}}$ and $\bar{\chi}_{\text{MRB}}$, and the processing of the candidates $\mathbf{c}$, are done simultaneously on the fly. For the sake of completeness, we give the implementation in Appendix B.

IV. Box-and-Match Algorithm (BMA) for ISI Channels

In this section, we present the general framework which allows us to generalize a more powerful OSD-type algorithm over the class of ISI channels. This algorithm, known as the Box-and-Match Algorithm (BMA), was introduced for memoryless channels in [3]. For the RS $[255, 239, 17]$ binary image transmitted over the memoryless AGWN channel, the BMA was shown to provide a 0.5 dB gain over OSD (designed for memoryless channels [18]) at WER=10^{-3}. The performance improvement was obtained with only a marginal increase in the average decoded list size [3]. Furthermore for the same code and channel, a further improvement over the BMA (known as the *biased* BMA), was shown to approach within 0.1 dB of the maximum-likelihood decoder [4]. However, the techniques in [3], [4] do not trivially extend to channels with memory.

Definition 16. *Recall the set of MRB positions β_{MRB} (see Definition 7) of size k. We define β_{CB} to be the set of* ***control band (CB) positions*** *of size κ, that satisfies the following two properties: i) $\beta_{CB} \subseteq \{0, 1, \cdots, n-1\} \setminus \beta_{MRB}$, where the size $\kappa = |\beta_{CB}|$ must satisfy $\kappa \leq n - k$. ii) β_{CB} contains the κ most reliable symbol positions in the set $\{0, 1, \cdots, n-1\} \setminus \beta_{MRB}$.*

Definition 17. *The* ***projection*** *(see Definition 6) of a binary sequence $\mathbf{a}$ onto the CB is defined to be $\phi_{CB}(\mathbf{a}) \triangleq \phi_{\beta_{CB}}(\mathbf{a})$. We also define the projection of any input sequence pair $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ onto the CB, to be $\varphi_{CB}\left((\mathbf{a}, \mathbf{a}_{\text{ML}})\right) \triangleq \phi_{CB}(\mathbf{a} + \mathbf{a}_{\text{ML}})$.*

Remark 10. *Note that for any two disjoint input sequence pairs $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ and $(\mathbf{a}', \mathbf{a}_{\text{ML}})$, we have $\varphi_{CB}\left((\mathbf{a}, \mathbf{a}_{\text{ML}}) \circ (\mathbf{a}', \mathbf{a}_{\text{ML}})\right) = \varphi_{CB}\left((\mathbf{a}, \mathbf{a}_{\text{ML}})\right) + \varphi_{CB}\left((\mathbf{a}', \mathbf{a}_{\text{ML}})\right)$. The same property also holds for φ_{MRB}.*

Definition 18. *We define the set $\mathfrak{E}_{CB}$ to contain all error events ε, for which $\varphi_{CB}(\varepsilon) \neq \mathbf{0}$ and $\varphi_{MRB}(\varepsilon) = \mathbf{0}$. Thus by definition, $\mathfrak{E}_{CB} \cap \mathfrak{E}_{\text{MRB}} = \emptyset$. Similar to Definition 8, we* ***order*** *the error events in $\mathfrak{E}_{CB}$ in the increasing order of metric discrepancy. Also, similar to Definition 9, we define the set χ_{CB} of* ***all possible*** *combinations of error events in the set $\mathfrak{E}_{CB}$.*

Remark 11. *Note that for $x \in \chi_{MRB}$, the projection $\phi_{CB}(x)$ is not guaranteed to be $\mathbf{0}$.*

Remark 12. *For any codeword $\mathbf{c} \in \mathcal{C}$, we can construct the following decomposition*

$$(\mathbf{c}, \mathbf{a}_{\text{ML}}) \;=\; x \circ x' \circ x_{CB} \circ (\mathbf{a}, \mathbf{a}_{\text{ML}}), \qquad (10)$$

where $x, x' \in \chi_{MRB}$, and $x_{CB} \in \chi_{CB}$ (see Definition 18). In the decomposition (10), the pair $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ must decompose[8] *(see Definition 5) to a set of error events $\mathfrak{E}((\mathbf{a}, \mathbf{a}_{\text{ML}}))$ such that $\mathfrak{E}((\mathbf{a}, \mathbf{a}_{\text{ML}})) \cap (\mathfrak{E}_{\text{MRB}} \cup \mathfrak{E}_{CB}) = \emptyset$. We note the following:*

- *If $|x \circ x'| = 0$ (see Definition 5), then we must have $\mathbf{c} = \mathbf{c}_0$ (see Definition 11).*
- *If $|x \circ x'| = 1$, then either $|x| = 0$ or $|x'| = 0$.*

[8] The input sequence pair $(\mathbf{a}, \mathbf{a}_{\text{ML}})$ in (10) must satisfy $\phi_{\text{MRB}}(\mathbf{a}+\mathbf{a}_{\text{ML}}) = \mathbf{0}$ and $\phi_{\text{CB}}(\mathbf{a} + \mathbf{a}_{\text{ML}}) = \mathbf{0}$.

Algorithm 2: Box-and-Match Algorithm (BMA) for ISI channels

Input: The set $\bar{\chi}_{\text{MRB}}$, the control band set $\bar{\chi}_{\text{CB}}$;
Output: Codeword list $\mathcal{L}_{\text{BMA}}$ and best candidate
  $\mathbf{c}_{\text{opt}} = \arg\min_{\mathbf{c} \in \mathcal{L}_{\text{BMA}}} M(\mathbf{c}, \mathbf{a}_{\text{ML}})$;
Initialize: $\mathcal{L}_{\text{BMA}} := \emptyset$ and $B[\sigma] := \emptyset$ for all $\sigma \in \mathcal{F}_2^{\kappa}$;
foreach $x \in \bar{\chi}_{MRB}$ **do**
  Compute the MRB projection $\nu := \varphi_{\text{MRB}}(x)$ of x;
  Compute the candidate codeword $\mathbf{c} := \xi(\nu)$;
  Store $\mathbf{c}$ in the codeword list $\mathcal{L}_{\text{BMA}}$;
  foreach $x_{CB} \in \bar{\chi}_{CB}$ *disjoint with* x **do**
    Compute "box label"
    $\sigma := \phi_{\text{CB}}(\mathbf{c} + \mathbf{c}_0 + \mathbf{a}_{\text{ML}}) + \varphi_{\text{CB}}\left(x \circ x_{\text{CB}}\right)$;
    foreach $x' \in B[\sigma]$ *disjoint with both* x *and* x_{CB} **do**
      Compute the MRB projection $\nu' := \varphi_{\text{MRB}}(x')$ and $\mathbf{c}' := \xi(\nu')$;
      Compute the candidate codeword $\mathbf{c}'' := \mathbf{c} + \mathbf{c}' + \mathbf{c}_0$;
      Store $\mathbf{c}''$ in the codeword list $\mathcal{L}_{\text{BMA}}$;
    end
  end
  Store ("box") MRB combination x in $B[\phi_{\text{CB}}(\mathbf{c}) + \varphi_{\text{CB}}\left(x\right)]$;
end

BMA is a list decoding algorithm. Let $\mathcal{L}_{\text{BMA}}$ be the list of codewords considered by the BMA. Similarly to the OSD, the BMA takes as input a set of MRB combinations $\bar{\chi}_{\text{MRB}} \subseteq \chi_{\text{MRB}}$. The difference between OSD and BMA is the following. OSD considers only the codewords $\mathbf{c}$ for which $\mathbf{c} = \xi\varphi_{\text{MRB}}(x)$ and $x \in \bar{\chi}_{\text{MRB}}$ (see the composite map $\xi\varphi_{\text{MRB}}$ described in Remark 8), but BMA considers additional codewords $\mathbf{c}$ that satisfy $\mathbf{c} = \xi\varphi_{\text{MRB}}(x \circ x')$, where $x \in \bar{\chi}_{\text{MRB}}$ and $x' \in \bar{\chi}_{\text{MRB}}$, but $x \circ x' \notin \bar{\chi}_{\text{MRB}}$. To limit complexity, we would like to consider only those additional candidates $\mathbf{c} \in \mathcal{L}_{\text{BMA}}$ that have high likelihood, meaning that their metric discrepancies $M(x_{\text{CB}})$ are small (where x_{CB} is obtained from the decomposition (10)). Hence, the BMA restricts combinations $x \circ x' \notin \bar{\chi}_{\text{MRB}}$ to those that map to codewords that are not considered by the OSD, but have small metric discrepancy $M(x_{\text{CB}})$. The procedure to pair the combinations x and x' is known as *matching*. Further, to limit the complexity of the matching procedure, instead of considering all $x_{\text{CB}} \in \chi_{\text{CB}}$, we restrict the choices in the subset $\bar{\chi}_{\text{CB}} \subseteq \chi_{\text{CB}}$. This subset $\bar{\chi}_{\text{CB}}$ is termed the *control band set*. Typically, we limit the size of the set $\bar{\chi}_{\text{CB}}$ and use the Battail construction (see Section III-B) to populate the set.

Definition 19. *For the purpose of matching, we construct a collection of lists $B[\sigma]$ for every $\sigma \in \mathcal{F}_2^{\kappa}$, where each list $B[\sigma]$ stores selected combinations $x \in \bar{\chi}_{MRB}$. We call $B[\sigma]$ the* ***box*** *whose* ***label*** *is σ.*

The definitions presented above are generalizations of their memoryless-channel counterparts presented in [3], [4]. Algorithm 2 states the BMA for ISI channels. In the sequel, we will show that the BMA additionally considers codewords that have low metric discrepancy over the CB, which results in performance gain over the OSD. Readers who are only interested in the implementation of BMA may skip the rest of this exposition and go directly to subsection IV-A.

Line 4 in OSD (Alg. 1) places into $\mathcal{L}_{\text{OSD}}$ *exactly* the same codewords that Line 4 in BMA (Alg. 2) places into the list $\mathcal{L}_{\text{BMA}}$. However, the BMA (Alg. 2) considers more codewords than OSD (as seen from Lines 5-12 in Alg. 2). Hence it is easy to verify that $\mathcal{L}_{\text{OSD}} \subseteq \mathcal{L}_{\text{BMA}}$. The codewords that are not considered by the OSD but are considered by BMA, are those codewords $\mathbf{c}$ that satisfy $\mathbf{c} = \xi\varphi_{\text{MRB}}(x \circ x')$, where $x \circ x' \notin \bar{\chi}_{\text{MRB}}$. In this particular case (when $x \circ x' \notin \bar{\chi}_{\text{MRB}}$), the next proposition states that if both x and x' are in the set $\bar{\chi}_{\text{MRB}}$, then we are guaranteed that $\mathbf{c} = \xi\varphi_{\text{MRB}}(x \circ x')$ will be placed into $\mathcal{L}_{\text{BMA}}$. To prove this, we need the following lemma.

Lemma 1. *Let $x, x' \in \chi_{MRB}$ and assume that x and x' are disjoint. Let codewords $\mathbf{c}$ and $\mathbf{c}'$, be obtained from x and x', respectively (i.e., $\mathbf{c} = \xi\varphi_{MRB}(x)$ and $\mathbf{c}' = \xi\varphi_{MRB}(x')$). Then if $\mathbf{c}'' = \xi\varphi_{MRB}(x \circ x')$ for some $\mathbf{c}'' \in \mathcal{C}$, we must have*

$$\mathbf{c}'' = \mathbf{c}' + \mathbf{c} + \mathbf{c}_0. \tag{11}$$

Proof: Because x and x' are disjoint (also see Remark 10), and also because the map $\psi : \mathcal{F}_2^k \mapsto \mathcal{C}$ is linear, it follows from the MRB error map (Definition 11) that

$$\begin{aligned}
\mathbf{c}'' &= \xi\varphi_{\text{MRB}}(x \circ x') \\
&= \psi\left(\varphi_{\text{MRB}}(x \circ x')\right) + \mathbf{c}_0 \\
&= \psi(\varphi_{\text{MRB}}(x)) + (\mathbf{c}_0) + \psi(\varphi_{\text{MRB}}(x')) + (\mathbf{c}_0) + \mathbf{c}_0 \\
&= \mathbf{c} + \mathbf{c}' + \mathbf{c}_0. \qquad \blacksquare
\end{aligned}$$

Proposition 1. *Let the sets of combinations $\bar{\chi}_{MRB}$ and $\bar{\chi}_{CB}$ be inputs to Algorithm 2. Let $\mathbf{c}''$ be any codeword whose decomposition $(\mathbf{c}'', \mathbf{a}_{\text{ML}}) = x \circ x' \circ x_{CB} \circ (\mathbf{a}, \mathbf{a}_{\text{ML}})$ satisfies*

1) $x \circ x' \notin \bar{\chi}_{MRB}$.
2) $x, x' \in \bar{\chi}_{MRB}$.
3) $x_{CB} \in \bar{\chi}_{CB}$.

Then Algorithm 2 will place the codeword $\mathbf{c}''$ into the list $\mathcal{L}_{\text{BMA}}$.

Proof: (In this proof, all algorithm line references refer to Algorithm 2.) We will show that the execution of Lines 5-13 will result in $\mathbf{c}''$ being placed into $\mathcal{L}_{\text{BMA}}$. From the decomposition (10), the projections onto the CB satisfy

$$\phi_{\text{CB}}(\mathbf{c}'' + \mathbf{a}_{\text{ML}}) = \varphi_{\text{CB}}\left(x \circ x' \circ x_{\text{CB}}\right). \tag{12}$$

On the other hand, the projections over the MRB satisfy $\phi_{\text{MRB}}(\mathbf{c}'' + \mathbf{a}_{\text{ML}}) = \varphi_{\text{MRB}}(x \circ x')$. Hence, $\xi\varphi_{\text{MRB}}(x \circ x') = \mathbf{c}''$. From Lemma 1, we conclude that

$$\phi_{\text{CB}}(\mathbf{c}''+\mathbf{a}_{\text{ML}}) = \phi_{\text{CB}}(\mathbf{c})+\phi_{\text{CB}}(\mathbf{c}')+\phi_{\text{CB}}(\mathbf{c}_0+\mathbf{a}_{\text{ML}}), \tag{13}$$

where $\xi\varphi_{\text{MRB}}(x) = \mathbf{c}$ and $\xi\varphi_{\text{MRB}}(x') = \mathbf{c}'$ and $\mathbf{c}', \mathbf{c}'' \in \mathcal{C}$. So equating (12) to (13) and we get

$$\phi_{\text{CB}}(\mathbf{c}') + \varphi_{\text{CB}}\left(x'\right) = \phi_{\text{CB}}(\mathbf{c} + \mathbf{c}_0 + \mathbf{a}_{\text{ML}}) + \varphi_{\text{CB}}\left(x \circ x_{\text{CB}}\right), \tag{14}$$

where the RHS has exactly the same form as the box label σ in Line 6. Assume without loss of generality that $M(x) > M(x')$, implying that Algorithm 2 processes x' before x. Then the following events must occur (in the stated order).

- Algorithm 2 processes x', computes $\mathbf{c}'$, and stores x' into box $B[\phi_{\text{CB}}(\mathbf{c}') + \varphi_{\text{CB}}(x')]$.
- Eventually, Algorithm 2 considers x, computes $\mathbf{c}$, and enters the loop starting at Line 5.
- Eventually, Algorithm 2 locates the element x_{CB} in the set $\bar{\chi}_{\text{CB}}$. Since x and x_{CB} are disjoint, Algorithm 2 computes box label $\sigma = \phi_{\text{CB}}(\mathbf{c} + \mathbf{c}_0 + \mathbf{a}_{\text{ML}}) + \varphi_{\text{CB}}(x \circ x_{\text{CB}})$.
- Since, by (14), the label σ equals $\phi_{\text{CB}}(\mathbf{c}') + \varphi_{\text{CB}}(x')$, and since x' was already placed earlier into the box with the label $\phi_{\text{CB}}(\mathbf{c}') + \varphi_{\text{CB}}(x')$, Algorithm 2 finds x' in $B[\sigma]$ (because (14) is satisfied and x, x' and x_{CB} are all disjoint, see Line 7).
- Algorithm 2 computes $\mathbf{c}'$ for x' found in $B[\sigma]$ (see Line 8). Next, Algorithm 2 obtains $\mathbf{c}''$ (using Lemma 1, see Line 9) as $\mathbf{c}'' = \mathbf{c}+\mathbf{c}'+\mathbf{c}_0$, and places $\mathbf{c}''$ into the list $\mathcal{L}_{\text{BMA}}$ (see Line 10). $\blacksquare$

Remark 13. *Proposition 1 implies that if $\mathbf{c}''$ is the transmitted codeword, and if the error pattern $(\mathbf{c}'', \mathbf{a}_{\text{ML}})$ occurs in such a way that there exists a decomposition $(\mathbf{c}'', \mathbf{a}_{\text{ML}}) = x \circ x' \circ x_{CB} \circ (\mathbf{a}, \mathbf{a}_{\text{ML}})$ satisfying all conditions 1), 2), and 3) in Proposition 1, then the transmitted codeword is guaranteed to be placed into $\mathcal{L}_{\text{BMA}}$.*

However, the inverse to Proposition 1 does not always hold. For a particular codeword $\mathbf{c}''$ that Algorithm 2 places into the list $\mathcal{L}_{\text{BMA}}$, the combination $(\mathbf{c}'', \mathbf{a}_{\text{ML}})$ may not satisfy all conditions 1), 2), and 3) in Proposition 1. The question is, what conditions does the codeword $\mathbf{c}''$ then satisfy? The next proposition answers this question.

Proposition 2. *Let $\bar{\chi}_{MRB}$ and $\bar{\chi}_{CB}$ be inputs to Algorithm 2. Let $\mathbf{c}''$ be a codeword that is stored in $\mathcal{L}_{\text{BMA}}$ in Algorithm 2 Line 10, and let $(\mathbf{c}'', \mathbf{a}_{\text{ML}})$ have the following decomposition $(\mathbf{c}'', \mathbf{a}_{\text{ML}}) = \tilde{x} \circ \tilde{x}' \circ \tilde{x}_{CB} \circ (\mathbf{a}, \mathbf{a}_{\text{ML}})$. Then there exists some $x, x' \in \bar{\chi}_{MRB}$ and $x_{CB} \in \bar{\chi}_{CB}$ such that*

$$\varphi_{CB}\left(\tilde{x} \circ \tilde{x}' \circ \tilde{x}_{CB}\right) \quad = \quad \varphi_{CB}\left(x \circ x' \circ x_{CB}\right) \tag{15}$$

is satisfied.

Proof: (In this proof, all algorithm line references refer to Algorithm 2.) Let $\mathbf{c}''$ be a codeword stored in $\mathcal{L}_{\text{BMA}}$ in Line 10, and note that $\mathbf{c}'' = \mathbf{c} + \mathbf{c}' + \mathbf{c}_0$, for some codewords $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$. From the steps in Algorithm 2, we observe the following:

- $\mathbf{c} = \xi\varphi_{\text{MRB}}(x)$ for some $x \in \bar{\chi}_{\text{MRB}}$ (from Lines 2-4).
- $\mathbf{c}' = \xi\varphi_{\text{MRB}}(x')$ for some $x' \in \bar{\chi}_{\text{MRB}}$ (from Line 8).
- The box label is $\sigma = \phi_{\text{CB}}(\mathbf{c} + \mathbf{c}_0 + \mathbf{a}_{\text{ML}}) + \varphi_{\text{CB}}(x \circ x_{\text{CB}})$, where $x_{\text{CB}} \in \bar{\chi}_{\text{CB}}$ (from Line 6).
- x' was found in the box $B[\sigma]$ (from Line 7).

Since x' was previously "boxed" in $B[\sigma]$ (see Line 13), this implies that the box label σ equals $\phi_{\text{CB}}(\mathbf{c}') + \varphi_{\text{CB}}(x')$. Thus we have

$$\begin{aligned}
\sigma &= \phi_{\text{CB}}(\mathbf{c}') + \varphi_{\text{CB}}\left(x'\right) \\
&= \phi_{\text{CB}}(\mathbf{c} + \mathbf{c}_0 + \mathbf{a}_{\text{ML}}) + \varphi_{\text{CB}}\left(x \circ x_{\text{CB}}\right)
\end{aligned} \tag{16}$$

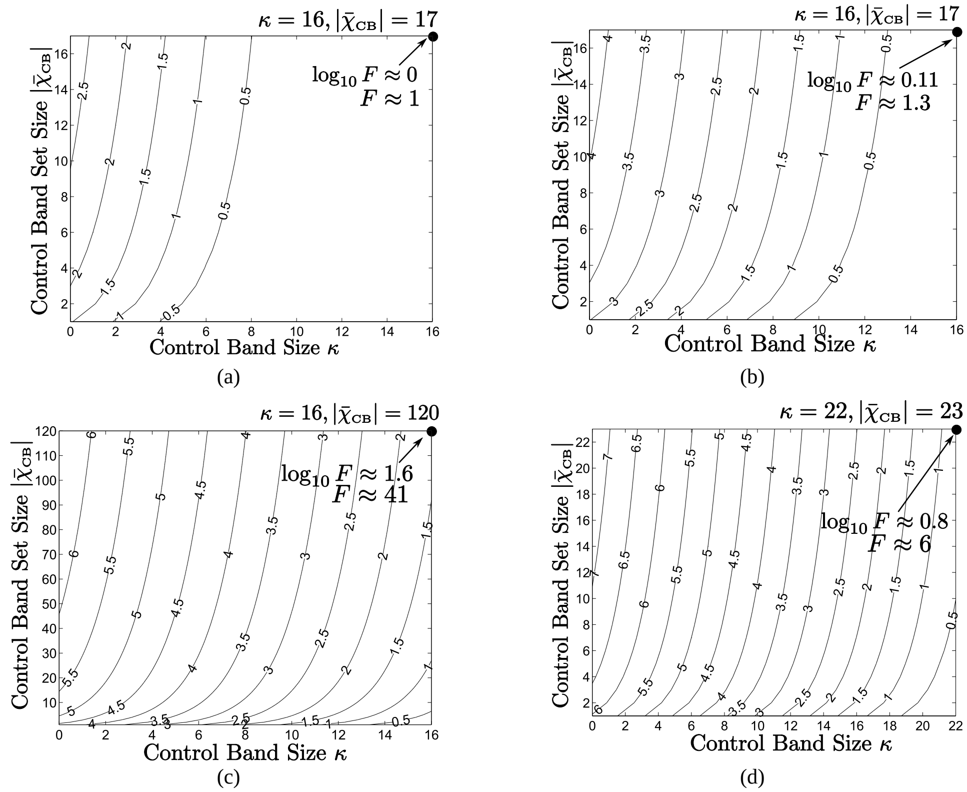


Fig. 6. The figures a), b), c) correspond to the eBCH $[128, 64, 22]$ code and $\mathcal{O}(\bar{\chi}_{\mathrm{MRB}}) = 1, 2$ and 3, respectively. The figure d) corresponds to the RS $[255, 239, 17]$ binary image and $\mathcal{O}(\bar{\chi}_{\mathrm{MRB}}) = 2$. Here F is the BMA list increment factor. Lines of constant $\log_{10} F$ for various sizes $|\bar{\chi}_{\mathrm{MRB}}|$, shown for a range of values of κ and $|\bar{\chi}_{\mathrm{CB}}|$.

Since the matching is only possible if x, x' and x_{CB} were all disjoint (see Lines 5 and 8, also see Remark 10), by the linearity of the projection and by rearranging (16) we get

$$\phi_{\mathrm{CB}}(\mathbf{c} + \mathbf{c}' + \mathbf{c}_0 + \mathbf{a}_{\mathrm{ML}}) = \varphi_{\mathrm{CB}}\left(x \circ x' \circ x_{\mathrm{CB}}\right). \tag{17}$$

Finally, note that $\mathbf{c}'' = \mathbf{c}+\mathbf{c}'+\mathbf{c}_0$ and we have $\phi_{\mathrm{CB}}(\mathbf{c}''+\mathbf{a}_{\mathrm{ML}}) = \varphi_{\mathrm{CB}}\left(x \circ x' \circ x_{\mathrm{CB}}\right)$. On the other hand, since $(\mathbf{c}'', \mathbf{a}_{\mathrm{ML}})$ has the decomposition $(\mathbf{c}'', \mathbf{a}_{\mathrm{ML}}) = \tilde{x} \circ \tilde{x}' \circ \tilde{x}_{\mathrm{CB}} \circ (\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$, we also have $\phi_{\mathrm{CB}}(\mathbf{c}'' + \mathbf{a}_{\mathrm{ML}}) = \varphi_{\mathrm{CB}}\left(\tilde{x} \circ \tilde{x}' \circ \tilde{x}_{\mathrm{CB}}\right)$. Thus we are done. ■

Remark 14. *Note that in [3], the design of the algorithm requires that $\varphi_{CB}(x) = \varphi_{CB}(x') = \mathbf{0}$ in order to perform matching. When applied to memoryless channels, Algorithm 2 specializes to the BMA given in [3], because for channels without memory, we always have $\varphi_{CB}(x) = \varphi_{CB}(x') = \mathbf{0}$.*

We next show that the additional complexity of the BMA over the OSD (i.e., the number of candidate codewords stored in Algorithm 2 Line 10) depends on the sizes $\kappa = |\beta_{\mathrm{CB}}|$ and $|\bar{\chi}_{\mathrm{CB}}|$. Essentially the matching technique is implemented by distributing the combinations $x \in \bar{\chi}_{\mathrm{MRB}}$ into 2^κ boxes, and the number of matches per $x \in \bar{\chi}_{\mathrm{MRB}}$ is determined by how many combinations there are in each box. Thus, the larger κ is, the fewer combinations there will be in each box (and therefore resulting in a smaller list size $|\mathcal{L}_{\mathrm{BMA}}|$). On the other hand, a larger κ means that we have a larger $|\chi_{\mathrm{CB}}|$ (i.e., we have to take into account that there will be a larger number of possible error event combinations over the CB). Hence κ and $|\bar{\chi}_{\mathrm{CB}}|$ are parameters that balance these two trade-offs. We provide a simple estimate on the complexity of the BMA, following along the lines of [3] by first making the following assumption (which is similar to the assumption in [3]).

Assumption 1. *We assume that all combinations $x \in \bar{\chi}_{MRB}$ are distributed **evenly** into each box, i.e., after the i-th execution of Line 13 in Algorithm 2, every box will contain approximately $i \cdot 2^{-\kappa}$ combinations x.*

Proposition 3. *Under Assumption 1, for a CB of size κ, the complexity of the BMA is estimated by the list size*

$$\begin{aligned} |\mathcal{L}_{\mathrm{BMA}}| &= |\bar{\chi}_{MRB}|\left(1 + (|\bar{\chi}_{MRB}| + 1)|\bar{\chi}_{CB}|2^{-\kappa-1}\right) \\ &= |\bar{\chi}_{MRB}| \cdot F \approx |\bar{\chi}_{MRB}| \text{ for large } \kappa \end{aligned} \tag{18}$$

Proof: For each combination $x \in \bar{\chi}_{MRB}$ we look into $|\bar{\chi}_{CB}|$ boxes, each box containing approximately $i \cdot 2^{-\kappa}$ combinations by Assumption 1. We count the number of total matches in Algorithm 2 as

$$\begin{aligned} \sum_{i=1}^{|\bar{\chi}_{\mathrm{MRB}}|} \sum_{j=1}^{|\bar{\chi}_{\mathrm{CB}}|} i 2^{-\kappa} &= |\bar{\chi}_{\mathrm{CB}}| 2^{-\kappa} \sum_{i=1}^{|\bar{\chi}_{\mathrm{MRB}}|} i \\ &= |\bar{\chi}_{\mathrm{MRB}}|(|\bar{\chi}_{\mathrm{MRB}}| + 1)|\bar{\chi}_{\mathrm{CB}}|2^{-\kappa-1} \end{aligned}$$ ■

Since $|\mathcal{L}_{\mathrm{OSD}}| = |\bar{\chi}_{\mathrm{MRB}}|$, for large κ, the list size of the BMA is estimated (under Assumption 1) to be similar to that of OSD. For a CB of size κ, the constant $F = 1+(|\bar{\chi}_{\mathrm{MRB}}|+1)|\bar{\chi}_{\mathrm{CB}}|2^{-\kappa-1}$ in (18) is termed as the BMA *list increment factor*. For both codes eBCH $[128, 64, 22]$ and RS $[255, 239, 17]$ binary image, Figure 6 shows the list increment factors, computed for various values of set sizes $|\bar{\chi}_{\mathrm{MRB}}|$ (which is determined by $\mathcal{O}(\bar{\chi}_{\mathrm{MRB}})$), control band sizes κ, and control band set sizes $|\bar{\chi}_{\mathrm{CB}}|$. We

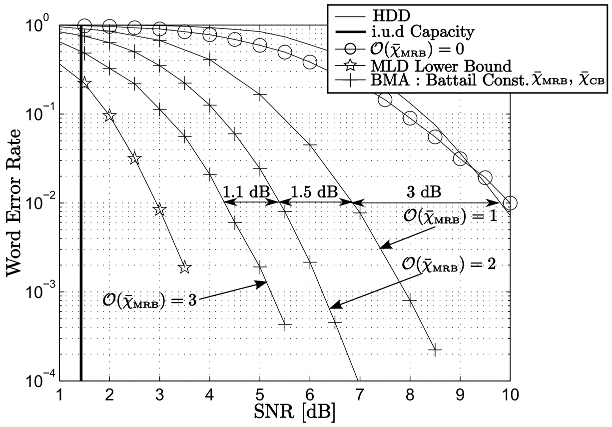


Fig. 7. Word error rate of the BMA when decoding the $[128, 64, 22]$ eBCH code.

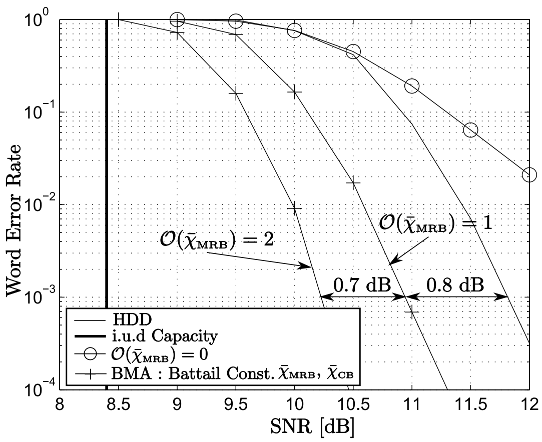


Fig. 8. Word error rate for the BMA when decoding the RS $[255, 239, 17]$ binary image.

would like to point out that the total number of codewords in all the boxes do not depend on κ.

A. Performance of the BMA

Figure 7 shows the performances of BMA, for the $[128, 64, 22]$ eBCH code and PR2 channel. The control band size $\kappa = 16$, and both sets $\bar{\chi}_{\text{MRB}}$ and $\bar{\chi}_{\text{CB}}$ are obtained using the Battail construction. The control band set sizes $|\bar{\chi}_{\text{CB}}|$ of the BMA, are set to be $17, 17$ and 120, for $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1, 2,$ and 3, respectively. The performance gains of the BMA over HDD at WER= 10^{-2}, for $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1, 2,$ and 3, are seen from Figure 7 to be $3, 4.5$ and 5.5 dB, respectively. Comparing Figures 7 and 4, the respective gains of the BMA over OSD at WER=10^{-2} are $0.2, 0.5$ and 0.7 dB. The BMA list increment factors are estimated from (18) to be approximately $1, 1.3$ and 41. Hence we see that the BMA outperforms the OSD for all values of $\mathcal{O}(\bar{\chi}_{\text{MRB}})$, and specifically for $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1$ and 2, the complexity (as estimated using Assumption 1 and (18)) increase is almost negligible for both cases. We also note that at WER=10^{-2}, the BMA of order $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 2$ performs within 0.3dB of the OSD of order $\mathcal{O}(\mathcal{L}_{\text{OSD}}) = 3$, whereby the latter algorithm is estimated (using Assumption 1 and (18)) approximately 16 times as complex as the former.

Figure 8 shows the performance of the BMA for the RS $[255, 239, 17]$ binary image over the PR2 channel. The control band size $\kappa = 22$ and the control band set sizes $|\bar{\chi}_{\text{CB}}|$ of the BMA, are set to be 1 and 23 for $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1$ and 2 respectively, and the list increment factors (18) are approximately 1 and 6 respectively. The performance gains of the BMA over HDD at WER=10^{-3} are seen from Figure 8 to be 0.8dB and 1.5dB. Comparing with Figure 5, the performance gain of the BMA over the OSD, for $\mathcal{O}(\bar{\chi}_{\text{MRB}}) = 1$ and 2, are noted to be 0.3dB and 0.4dB, respectively. Finally, we note from the WER performances in Figures 7 and 8, that the performance gains of the BMA for the PR2 channel, are somewhat smaller than those shown in [3], [18] for the (memoryless) channel $\mathbf{h} = [1]$. This is because performing BMA techniques for ISI channels involves an additional difficulty, which is due to the fact that the sets χ_{MRB} and χ_{CB} are typically much larger when the channel has memory.

V. CONCLUSION

As an extension to the OSD techniques shown in [2], we presented OSD techniques within the more general framework of ISI channels. Simulations applied to the RS $[255, 239, 17]$ binary image, show that our OSD technique outperforms the previous technique in [2] that was specifically designed for memoryless channels. Our simulation results also show that for ISI channels, the Battail construction outperforms the lexicographical construction. Furthermore, motivated by the performance gain of BMA techniques [3] over OSD in memoryless channels, we developed BMA techniques specifically for ISI channels. Because the original BMA techniques in [3] were specifically designed for channels without memory, we presented the new BMA within the framework of ISI channels. Performance results show that in some of the tested cases, the BMA outperforms the OSD, with an (estimated) complexity that is very close to that of the OSD. In one particular instance for the eBCH $[128, 64, 22]$ code, our BMA performs similarly to our OSD (designed for ISI channels), but the OSD algorithm being compared to is approximately 16 times more complex. For the RS$[255, 239, 17]$ binary image, the BMA is able to outperform the OSD with only a minimal increase in complexity. Finally, we would like to point out that all OSD-type techniques presented in this paper collapse down to those in [2], [3] when the ISI channel impulse response $\mathbf{h} = [1]$.

APPENDIX

A. Battail's Algorithm for ISI Channels

Battail's Algorithm for ISI channels is given in Algorithm 3. Some clarification of the notation is in order. We define a buffer $\mathcal{L}$, of fixed size $|\mathfrak{E}_{\text{MRB}}|$, that stores error event combinations $x \in \chi_{\text{MRB}}$, and write $\mathcal{L}[i]$ to indicate the ith storage element of the buffer $\mathcal{L}$. If some combination x is stored in $\mathcal{L}[i]$, then we write $M(\mathcal{L}[i]) = M(x)$, otherwise if $\mathcal{L}[i]$ is empty, we set $M(\mathcal{L}[i]) = \infty$. Finally, for some real number $\mathcal{M} \geq 0$, and some integer $i \in \{1, 2, \cdots, |\mathfrak{E}_{\text{MRB}}|\}$, we define a *special* subset $\bar{\chi}_{\text{MRB}}(\mathcal{M}, i) \subseteq \bar{\chi}_{\text{MRB}}$ as

$$\bar{\chi}_{\text{MRB}}(\mathcal{M}, i) = \{x \in \bar{\chi}_{\text{MRB}} | M(x) > \mathcal{M},\ \Xi(x) < 2^i\}, \qquad (19)$$

where $\Xi(x)$ is the lexicographical representation of the error event combination x (see Definition 13). That is, the subset

Algorithm 3: Battail const. of $\bar{\chi}_{\text{MRB}}$, see [7], [15]

Input: The set of MRB error events $\mathfrak{E}_{\text{MRB}}$, order $\mathcal{O}(\bar{\chi}_{\text{MRB}})$;
Output: The constructed set $\bar{\chi}_{\text{MRB}}$;
Initialize: Buffer $\mathcal{L}[i] := \varepsilon_{\text{MRB}}^{(i)}$ for $i \in \{i | 1 \leq i \leq |\mathfrak{E}_{\text{MRB}}|\}$;

1 **repeat**
2 Minimize over all buffer locations $i^* = \arg\min_{1 \leq i \leq |\mathfrak{E}_{\text{MRB}}|} M(\mathcal{L}[i])$;
3 Set $x^* := \mathcal{L}[i^*]$ and empty the i^*-th buffer location $\mathcal{L}[i^*]$;
4 Record metric discrepancy ordering $x^{(|\bar{\chi}_{\text{MRB}}|+1)} := x^*$; Store x^* in $\bar{\chi}_{\text{MRB}}$;
5 Decompose x^* as $x^* = \varepsilon_{\text{MRB}}^{(i^*)} \circ x'$, where $x' \in \chi_{\text{MRB}}$;
6 Set $\mathcal{M} := M(x')$;
7 Find the element in $\bar{\chi}_{\text{MRB}}(\mathcal{M}, i^*)$ that satisfies $x'' = \arg\min_{x \in \bar{\chi}_{\text{MRB}}(\mathcal{M}, i^*)} M(x)$;
8 **if** x^* *and* x'' *are disjoint* **then** store $\varepsilon_{\text{MRB}}^{(i^*)} \circ x''$ in $\mathcal{L}[i^*]$;
9 **until** $|\bar{\chi}_{MRB}| = \sum_{i=0}^{\mathcal{O}(\bar{\chi}_{MRB})} \binom{k}{i}$;

$\bar{\chi}_{\text{MRB}}(\mathcal{M}, i)$ contains all elements $x \in \bar{\chi}_{\text{MRB}}$, such that x has both *higher* metric discrepancy than $\mathcal{M}$, and lower lexicographical order than 2^i.

In each repetition of Lines 2-8 in Algorithm 3, the error event combinations stored in the buffer $\mathcal{L}$ are candidate to be the $(|\bar{\chi}_{\text{MRB}}| + 1)$-th element (see Line 4) in the metric discrepancy ordering (19). Algorithm 3 ensures that the combination stored in $\mathcal{L}[i]$ always has the form $\varepsilon_{\text{MRB}}^{(i)} \circ x$, for some $x \in \chi_{\text{MRB}}$. Hence the decomposition in Line 5 is always possible. The minimization operation in (Algorithm 3) Line 2 can be implemented with only $\log_2 |\mathfrak{E}_{\text{MRB}}|$ comparisons (see [15]). Furthermore, the minimization in (Algorithm 3) Line 7 can be avoided at the expense of storage complexity (also see [15]).

B. Alternate Implementation of OSD for ISI channels

Algorithm 4 gives an alternate implementation of OSD for ISI channels, where the sets $\bar{\chi}_{\text{MRB}}$, $\mathfrak{E}_{\text{MRB}}$ and $\mathcal{L}_{\text{OSD}}$ are all constructed simultaneously. Note that we use the "serial" implementation of the (modified) GVA [7], [9] (i.e., the error events $\varepsilon_{\text{MRB}}^{(i)}$ are output one after another in increasing order of i). Also note that Algorithm 4 applies the Battail construction for the set $\bar{\chi}_{\text{MRB}}$ (see Appendix A).

Algorithm 4: OSD of ISI channels : Alt. Implementation

Input: Order $\mathcal{O}(\bar{\chi}_{\text{MRB}})$;
Initialize I: Set $\mathfrak{E}_{\text{MRB}} := \emptyset, \bar{\chi}_{\text{MRB}} := \emptyset, \mathcal{L}_{\text{OSD}} := \emptyset$;
Initialize II: Empty all buffer elements $\mathcal{L}[i]$, initialize counter $i := 0$;
Output: Codeword list $\mathcal{L}_{\text{OSD}}$ and best candidate $\mathbf{c}_{\text{opt}} = \arg\min_{\mathbf{c} \in \mathcal{L}_{\text{OSD}}} M(\mathbf{c}, \mathbf{a}_{\text{ML}})$;

repeat
  Increment $i := i + 1$;
  Output $\varepsilon_{\text{MRB}}^{(i)}$ using the (serial) modified GVA;
  Store $\varepsilon_{\text{MRB}}^{(i)}$ in $\mathfrak{E}_{\text{MRB}}$. Store $\mathcal{L}[i] := \varepsilon_{\text{MRB}}^{(i)}$;
  repeat
    Execute Lines 2-8 of Algorithm 3 and obtain x^*;
    Process new candidate codeword by executing Lines 2-4 of Algorithm 1;
    if $|\mathcal{L}_{\text{OSD}}| > \sum_{i=0}^{\mathcal{O}(\bar{\chi}_{MRB})} \binom{k}{i}$ **then quit**;
  until x^* *equals* $\varepsilon_{MRB}^{(i)}$;
until *modified GVA terminates* ;

REFERENCES

[1] M. P. C. Fossorier and S. Lin, "Error performance analysis for reliability-based decoding algorithms," *IEEE Trans. on Inform. Theory*, vol. 48, no. 1, pp. 287–293, Jan. 2002.

[2] ——, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. on Inform. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995.

[3] A. Valembois and M. P. C. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. on Inform. Theory*, vol. 50, no. 5, pp. 796–810, May 2004.

[4] W. Jin and M. P. C. Fossorier, "Towards maximum likelihood soft decision decoding of the $(255, 239)$ Reed-Solomon code," *IEEE Trans. on Magn.*, vol. 44, no. 3, pp. 423–428, Mar. 2008.

[5] V. B. and K. E. M., Eds., *Coding and Signal Processing for Magnetic Recording Systems*, 2nd ed. CRC Press, 2004.

[6] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *IEEE Global Telecomm. Conference (GLOBECOM '89)*, Dallas, TX, Nov. 1989, pp. 1680–1686.

[7] F. Lim, A. Kavčić, and M. Fossorier, "Ordered statistics decoding of linear block codes over intersymbol interference channels," *IEEE Trans. on Magn.*, vol. 44, no. 11, pp. 3765–3768, Nov. 2008.

[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Inform. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[9] N. Seshadri and C. E. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. on Comm*, vol. 42, no. 2/3/4, pp. 313–323, Feb./ Mar./ Apr. 1994.

[10] M. P. C. Fossorier and S. Lin, "Soft decision decoding of linear block codes based on ordered statistics for the Rayleigh fading channel with coherent detection," *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 12–14, Jan. 1997.

[11] J. W. H. Chen and J. N. Hwang, "Ordered statistics decoding of linear block codes on frequency nonselective multipath channels," *IEEE Journal Sel. Areas Commun.*, vol. 18, no. 11, pp. 2237–2239, Nov. 2000.

[12] A. Valembois and M. P. C. Fossorier, "A comparison between "most-reliable-basis reprocessing" strategies," *IEICE Trans. Fundamentals*, vol. E85-A, no. 7, pp. 1727–1741, Jul. 2002.

[13] J. Jiang and K. R. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix," *IEEE Trans. on Inform. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

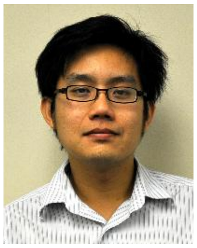
[14] G. Battail and J. Fang, "Décodage pondéré optimal des codes lineéaires en blocs," *Annales des télécommunications*, vol. 41, no. 11-12, pp. 580–604, Nov-Dec 2000.

[15] A. Valembois and M. P. C. Fossorier, "An improved method to compute lists of binary vectors that optimize a given weight function with application to soft-decision decoding," *IEEE Comm. Letters*, vol. 5, no. 11, pp. 456–458, Nov. 2001.

[16] B. G. Dorsch, "A decoding algorithm for binary block codes and J-ary output channels," *IEEE Trans. on Inform. Theory*, vol. 20, no. 3, pp. 391–394, May 1974.

[17] J. Bellorado, A. Kavčić, and P. Li, "Soft-input, iterative, Reed-Solomon decoding using redundant parity-check equations," in *IEEE Inform. Theory Workshop (ITW '07)*, Lake Tahoe, CA, Sep. 2007, pp. 138–143.

[18] M. P. C. Fossorier and A. Valembois, "Reliability-based decoding of Reed-Solomon codes using their binary image," *IEEE Comms. Letters*, vol. 8, no. 7, pp. 452–454, Jul. 2004.



Fabian Lim received the B.Eng and M.Eng degrees in Electrical Engineering from the National University of Singapore in 2003 and 2006 respectively. From October 2005 to May 2006, he was a staff member at the Data Storage Institute, Singapore. Since August 2006, he has been a Ph.D student in the Department of Electrical Engineering, University of Hawaii, Manoa. Mr Lim has previously held short-term visiting research positions at Harvard University in 2004 and 2005, and was an intern at Hitachi Global Storage Technologies from May 2008 to August 2008. His research interests include error-control coding and signal processing, mainly for data storage applications.

Aleksandar Kavčić received the Dipl. Ing. degree in Electrical Engineering from Ruhr-University, Bochum, Germany in 1993, and the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, Pennsylvania in 1998. Since 2007 he has been with the University of Hawaii, Honolulu where he is presently Associate Professor of Electrical Engineering. Prior to 2007, he was in the Division of Engineering and Applied Sciences at Harvard University, as Assistant Professor of Electrical Engineering from 1998 to 2002, and as John L. Loeb Associate Professor of Natural Sciences from 2002 to 2006. While on leave from Harvard University, he served as Visiting Associate Professor at the City University of Hong Kong in the Fall of 2005 and as Visiting Scholar at the Chinese University of Hong Kong in the Spring of 2006.

Prof. Kavčić received the IBM Partnership Award in 1999 and the NSF CAREER Award in 2000. He is a co-recipient, with X. Ma and N. Varnica, of the 2005 IEEE Best Paper Award in Signal Processing and Coding for Data Storage. He served on the Editorial Board of the IEEE Transactions on Information Theory as Associate Editor for Detection and Estimation from 2001 to 2004, as Guest Editor of the IEEE Signal Processing Magazine in 2003-2004, and as Guest Editor of the IEEE Journal on Selected Areas in Communications in 2008-2009. From 2005 until 2007, he was the Chair of the Data Storage Technical Committee of the IEEE Communications Society.

Marc Fossorier received the B.E. degree from the National Institute of Applied Sciences (I.N.S.A.) Lyon, France in 1987, and the M.S. and Ph.D. degrees in 1991 and 1994, all in electrical engineering. His research interests include decoding techniques for linear codes, communication algorithms and statistics. Dr. Fossorier is a recipient of a 1998 NSF Career Development award and became IEEE Fellow in 2006. He has served as Editor for the IEEE Transactions on Information Theory from 2003 to 2006, as Editor for the IEEE Transactions on Communications from 1996 to 2003, as Editor for the IEEE Communications Letters from 1999 to 2007, and as Treasurer of the IEEE Information Theory Society from 1999 to 2003. Since 2002, he has also been an elected member of the Board of Governors of the IEEE Information Theory Society which he served as Second and First Vice-President. He was Program Co-Chairman for the 2007 International Symposium on Information Theory (ISIT), the 2000 International Symposium on Information Theory and Its Applications (ISITA) and Editor for the Proceedings of the 2006, 2003 and 1999 Symposium on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC).