Ordered Statistics Decoding of Linear Block Codes Over Intersymbol Interference Channels

Fabian Lim[1], Aleksandar Kavčić[1], and Marc Fossorier[2]

[1]Electrical Engineering Department, University of Hawaii at Manoa, Honolulu, HI 96822 USA
[2]ETIS ENSEA/UCP/CNRS UMR-8051, Cergy Pontoise 95014, France

In this paper, we utilize ordered statistics decoding (OSD) techniques for intersymbol interference (ISI) channels. To achieve good efficiency and storage complexity, a combination of a modified generalized Viterbi algorithm (GVA) and Battail's algorithm is proposed to implement OSD for ISI channels. Simulation results showed significant performance improvement for the [128,64,22] eBCH code, and the [255,239,17] Reed-Solomon (RS) binary image, on the $(1+D)^2$ perpendicular recording channel.

***Index Terms*—Decoding, intersymbol interference, linear block codes, Reed-Solomon codes.**

I. Introduction

The aim of this paper is to implement and test ordered statistics decoding (OSD) techniques [1] over intersymbol interference (ISI) channels. OSD has been well-studied for the memoryless Additive White Gaussian Noise (AWGN) channel, and many improvements have been proposed [2], [3]. Furthermore, it was shown in [4] that despite its structure, OSD for the AWGN channel processes error patterns close to the optimal order defined by their likelihoods. However, the existing techniques are only efficient for memoryless channels. In this work, we consider adjustments in order to implement OSD for channels with memory. For any linear block code, OSD processes low-weight (sparse) error patterns in the most reliable basis (MRB) positions. However, in ISI channels, the errors in the MRB are observed to consist of small number of error events. Consequently, an efficient technique needs to process *error events* that may occur in the MRB.

II. Background

We first provide some basic definitions and simple formulas. The special constants n and k represent codelength and dimension, respectively. All sequences are represented by column vectors (e.g., $\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]^T$). Convolution between two sequences $\mathbf{a}$ and $\mathbf{b}$ is denoted as $\mathbf{a} * \mathbf{b}$. The Euclidean distance between sequences $\mathbf{a}$ and $\mathbf{b}$ is denoted as $\|\mathbf{a}-\mathbf{b}\|$. A codeword $\mathbf{c}$ from a linear codebook $\mathcal{C}$ is mapped to a vector $\mathbf{a} \in \{-1,1\}^n$, and transmitted through an ISI channel whose impulse response is $\mathbf{h}$. The received sequence is

$$\mathbf{r} = \mathbf{a} * \mathbf{h} + \mathbf{w} \tag{1}$$

where $\mathbf{w}$ is a realization of white Gaussian noise. The most likely sequence is

$$\mathbf{a}_{\rm ML} = \arg \min_{\mathbf{a} \in \{-1,1\}^n} \|\mathbf{r} - \mathbf{a} * \mathbf{h}\|^2 \tag{2}$$

and can be determined by a Viterbi detector [5].

Define $\alpha_i \triangleq |\log[\Pr\{a_i = -1|\mathbf{r}\}/\Pr\{a_i = 1|\mathbf{r}\}]|$ as the *reliability* of the ith symbol. For memoryless Gaussian channels (and equiprobable signaling), this reliability α_i is proportional to $|r_i|$, whereas for channels with memory, this quantity needs to be computed by the BCJR [6] algorithm or estimated by a *Soft Output Viterbi Algorithm* (SOVA) [5].

Definition 1: We define the **error sequence** $\mathbf{e}(\mathbf{a},\mathbf{b})$ of a signal sequence $\mathbf{a}$, with respect to some signal sequence $\mathbf{b}$ as

$$\mathbf{e}(\mathbf{a},\mathbf{b}) \triangleq \mathbf{a} - \mathbf{b}. \tag{3}$$

Definition 2: We define the **metric discrepancy** $M(\mathbf{a},\mathbf{b})$ of a signal sequence $\mathbf{a}$, with respect to some signal sequence $\mathbf{b}$ as

$$\begin{aligned} M(\mathbf{a},\mathbf{b}) &\triangleq \|\mathbf{r} - \mathbf{a} * \mathbf{h}\|^2 - \|\mathbf{r} - \mathbf{b} * \mathbf{h}\|^2 \\ &= \|\mathbf{e} * \mathbf{h}\|^2 - 2(\mathbf{r} - \mathbf{b} * \mathbf{h})^T(\mathbf{e} * \mathbf{h}) \end{aligned} \tag{4}$$

where $\mathbf{e} = \mathbf{e}(\mathbf{a},\mathbf{b})$.

Clearly $M(\mathbf{a},\mathbf{b}) = -M(\mathbf{b},\mathbf{a})$ and $M(\mathbf{a},\mathbf{a}_{\rm ML}) \geq 0$. We use shorthand notation $M_{\rm ML}(\mathbf{a}) \triangleq M(\mathbf{a},\mathbf{a}_{\rm ML})$.

Note that for the memoryless channel (i.e., $\mathbf{h} = [1]$)

$$\begin{aligned} M_{\rm ML}(\mathbf{a}) &= \|\mathbf{r} - \mathbf{a}\|^2 - \|\mathbf{r} - \mathbf{a}_{\rm ML}\|^2 \\ &= 2\sum_i -(a_i - a_{{\rm ML},i})r_i = 4 \sum_{i|e_i \neq 0} |r_i| \end{aligned} \tag{5}$$

In [1] and [2], the quantity (5) is (after an appropriate scaling) referred to as the *discrepancy*. Thus, by our definition of the metric discrepancy in (4), we generalize the terminology already used in [1] and [2] for memoryless channels.

We consider a channel *trellis* of state size 2^I, where I is the *memory* of the channel. Let $S_t(\mathbf{a})$ be the state at time t, when the channel sequence is $\mathbf{a}$. Clearly, $S_t(\mathbf{a}) \triangleq [a_{t-I+1}, a_{t-I+2}, \ldots, a_t]^T$.

Definition 3: Define the **temporal state difference set** of two signal sequences $\mathbf{a}$ and $\mathbf{b}$ to be the set of all time instants t for which $S_t(\mathbf{a}) \neq S_t(\mathbf{b})$, i.e., $\mathcal{T}(\mathbf{a},\mathbf{b}) \triangleq \{t | S_t(\mathbf{a}) \neq S_t(\mathbf{b})\}$.

Definition 4: Define the **temporal point of divergence**, and **convergence**, between two signal sequences $\mathbf{a}$ and $\mathbf{b}$, to be $D(\mathbf{a},\mathbf{b}) \triangleq \min \mathcal{T}(\mathbf{a},\mathbf{b}) - 1$, and $C(\mathbf{a},\mathbf{b}) \triangleq \max \mathcal{T}(\mathbf{a},\mathbf{b}) + 1$, respectively. See Fig. 1.

Definition 5: Two signal sequence pairs $(\mathbf{a},\mathbf{b})$ and $(\mathbf{a}',\mathbf{b})$ are said to be **disjoint** if $\mathcal{T}(\mathbf{a},\mathbf{b}) \cap \mathcal{T}(\mathbf{a}',\mathbf{b}) = \emptyset$.

Digital Object Identifier 10.1109/TMAG.2008.2002623

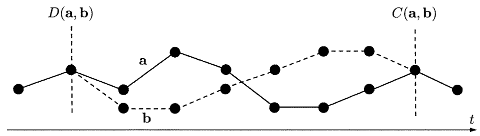

Fig. 1. Point of divergence $D(\mathbf{a},\mathbf{b})$, and convergence $C(\mathbf{a},\mathbf{b})$, respectively.

Definition 6: The ordered pair of signal sequences $\varepsilon = (\mathbf{a},\mathbf{b})$ is said to be an **error event**, if for all $t \in \{t | D(\mathbf{a},\mathbf{b}) < t < C(\mathbf{a},\mathbf{b})\}$, we have $S_t(\mathbf{a}) \neq S_t(\mathbf{b})$. We further define $M(\varepsilon) \triangleq M(\mathbf{a},\mathbf{b})$, and $\mathbf{e}(\varepsilon) \triangleq \mathbf{a}-\mathbf{b}$, and $\mathcal{T}(\varepsilon) \triangleq \mathcal{T}(\mathbf{a},\mathbf{b})$. Also, if $\varepsilon = (\mathbf{a},\mathbf{a}_{\rm ML})$, then $M_{\rm ML}(\varepsilon) \triangleq M_{\rm ML}(\mathbf{a})$.

Definition 7: For signal sequence pairs $(\mathbf{a},\mathbf{b})$ and $(\mathbf{a}',\mathbf{b})$, we define their **combination** $(\mathbf{a},\mathbf{b}) \circ (\mathbf{a}',\mathbf{b}) \triangleq (\mathbf{a}+\mathbf{a}'-\mathbf{b},\mathbf{b})$. If $(\mathbf{a},\mathbf{b})$ and $(\mathbf{a}',\mathbf{b})$ are disjoint, then $M((\mathbf{a},\mathbf{b}) \circ (\mathbf{a}',\mathbf{b})) = M(\mathbf{a},\mathbf{b}) + M(\mathbf{a}',\mathbf{b})$.

III. OSD FOR ISI CHANNELS

Let $\mathbf{g}_i$ be the ith column of the generator matrix $\mathbf{G}$ of a linear block code $\mathcal{C}$. Let $\mathcal{B}$ be the collection of all subsets $\beta = \{i_0, i_1, \cdots, i_{k-1}\} \subset \{0,1,\cdots,n-1\}$, where $\mathbf{g}_{i_0}, \mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_{k-1}}$ are *linearly independent* (i.e., $\mathbf{g}_{i_0}, \mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_{k-1}}$ constitute a basis [1], [2] for the code $\mathcal{C}$).

Definition 8: The set of **most reliable basis (MRB) positions** is $\beta^* \triangleq \arg\max_{\beta\in\mathcal{B}}(\sum_{j\in\beta}\alpha_j)$. The basis that corresponds to β^* is the **most reliable basis (MRB)**.

Each of the 2^k codewords $\mathbf{c} \in \mathcal{C}$ is *unique* over the MRB positions. In memoryless channels, the errors that occur over the basis are reported in [1], [2] to be *sparse* symbol errors. However, when the channel $\mathbf{h}$ has memory, then it is observed that a small number of *error events* (containing possibly consecutive symbol errors) will have support over the MRB. OSD for ISI channels can now be restated as processing *error events* (that have a least one errornous symbol in the MRB positions) in order of increasing metric discrepancies.

A. Battail's Algorithm

Let $\mathcal{E}$ be the set of all error events $(\mathbf{a},\mathbf{a}_{\rm ML})$ whose error sequences $\mathbf{e}(\mathbf{a},\mathbf{a}_{\rm ML})$ have nonzero components in at least one of the MRB positions. Furthermore, the error events $\varepsilon^{(i)} \in \mathcal{E}$ are ordered such that

$$M_{\rm ML}(\varepsilon^{(1)}) \leq M_{\rm ML}(\varepsilon^{(2)}) \leq \cdots \leq M_{\rm ML}(\varepsilon^{(|\mathcal{E}|)}) \qquad (6)$$

where the superscripts emphasize that this is not a time indexing. If the set $\mathcal{E}$ is known, we can modify Battail's algorithm [7] to obtain all *combinations* of disjoint error events $\varepsilon^{(i)}$ in increasing order of metric discrepancies.

Battail's Algorithm (modified for error events)
[Initialize]
B1) Initialize bins $B_i := \{\varepsilon^{(i)}\}$ and lists $L_i := \emptyset$ for $i \in \{i | 1 \leq i \leq |\mathcal{E}|\}$.
[Repeat until all bins B_i for all i are empty]
B2) Let $B_{i'}$ be the bin that contains the element ξ with the lowest discrepancy $M_{\rm ML}(\xi)$. Output ξ and delete it from $B_{i'}$. Update $L_{i'} := L_{i'} \cup \xi$.
B3) Let $\mathcal{U}$ be the set of *all* elements $\nu \in \cup_{i<i'} L_i$, that are *disjoint* with $\varepsilon^{(i')}$ and satisfy $M_{\rm ML}(\nu) + M_{\rm ML}(\varepsilon^{(i')}) \geq M_{\rm ML}(\xi)$. If $\mathcal{U} \neq \emptyset$ find the element $\xi' = \arg\min_{\nu\in\mathcal{U}} M_{\rm ML}(\nu)$ and go to B4); otherwise go to B2).
B4) Insert the combination $\varepsilon^{(i')} \circ \xi'$ into the bin $B_{i'}$.

B. Modified GVA

Obtaining the sorted set $\mathcal{E}$ is not trivial. To perform this task, we utilize a modification of the GVA [8], applied to the channel trellis. The GVA sorts *all* possible paths in the trellis in order of the metric discrepancies [8]. But here we only need to obtain the ordered set of error events $\mathcal{E}$, thus we propose the following implementation.

The GVA requires more operations and storage units than the Viterbi algorithm and/or SOVA [8]. We perform the Viterbi algorithm/SOVA, starting from the *terminal*[1] state in the trellis, and go backwards in time. At each time instant t and state s, we store the following.

- Two sequences $\mathbf{b}^{(t,s)}$ and $\bar{\mathbf{b}}^{(t,s)}$ that contend to be the *backward survivor* at state s and time t. Explicitly, $\mathbf{b}^{(t,s)} = [b_{t-I+1}^{(t,s)}, b_{t-I+2}^{(t,s)}, \cdots]^T$ and $\bar{\mathbf{b}}^{(t,s)} = [\bar{b}_{t-I+1}^{(t,s)}, \bar{b}_{t-I+2}^{(t,s)}, \cdots]^T$, and $s = S_t(\mathbf{b}^{(t,s)}) = S_t(\bar{\mathbf{b}}^{(t,s)})$, i.e., both sequences pass through the state s at time t. By convention we assume that $\mathbf{b}^{(t,s)}$ is the survivor, i.e.,

$$M(\bar{\mathbf{b}}^{(t,s)}, \mathbf{b}^{(t,s)}) \geq 0. \qquad (7)$$

- Their point of divergence $D(\mathbf{b}^{(t,s)}, \bar{\mathbf{b}}^{(t,s)}) = t$.
- Their point of convergence $C(\mathbf{b}^{(t,s)}, \bar{\mathbf{b}}^{(t,s)})$.

Define the error event $\varepsilon^{(t,s)} \triangleq (\bar{\mathbf{b}}^{(t,s)}, \mathbf{b}^{(t,s)})$. The set of error events $\varepsilon^{(t,s)}$ obtained by the GVA for all $t \in \{t | 0 \leq t < n\}$ and $s \in \{-1,1\}^I$ does *not* contain all possible error events $(\mathbf{a},\mathbf{a}_{\rm ML})$. However, as shown in [8], any error event $(\mathbf{a},\mathbf{a}_{\rm ML})$ can be reconstructed using a *unique* subset of error events $\varepsilon^{(t,s)}$. See Fig. 2 for an illustration. The reconstruction/decomposition can be written as follows:

$$\begin{aligned}\mathbf{e}(\mathbf{a},\mathbf{a}_{\rm ML}) = \mathbf{a} - \mathbf{a}_{\rm ML} &= \sum_{t\in\tau(\mathbf{a})} \mathbf{e}\left(\varepsilon^{(t,S_t(\mathbf{a}))}\right) \\ &= \sum_{t\in\tau(\mathbf{a})} \left(\bar{\mathbf{b}}^{(t,S_t(\mathbf{a}))} - \mathbf{b}^{(t,S_t(\mathbf{a}))}\right) \qquad (8)\end{aligned}$$

where the unique set $\tau(\mathbf{a})$ is a set of time indices t that satisfy $t \in \{t | D(\mathbf{a},\mathbf{a}_{\rm ML}) \leq t < C(\mathbf{a},\mathbf{a}_{\rm ML})\}$. The elements of the set $\tau(\mathbf{a}) = \{t_0, t_1, \cdots, t_{\kappa-1}\}$ satisfy $t_0 = D(\mathbf{a},\mathbf{a}_{\rm ML})$ and $t_i = D(\mathbf{a}, \bar{\mathbf{b}}^{(t_{i-1},s_{i-1})}) > t_{i-1}$ for all $i \in \{i | 0 < i < \kappa\}$. See Fig. 2 for an illustration. Given any signal sequence $\mathbf{a}$, the set $\tau(\mathbf{a})$ can be found using the following algorithm.

Decomposition of error event $\varepsilon = (\mathbf{a},\mathbf{a}_{\rm ML})$
[Initialize]
1) Set $t := D(\mathbf{a},\mathbf{a}_{\rm ML})$, and $\bar{\mathbf{b}} := \bar{\mathbf{b}}^{(t,S_t(\mathbf{a}))}$ and $\tau(\mathbf{a}) := \{t\}$.
[Do while $t < C(\mathbf{a},\mathbf{a}_{\rm ML})$]
2) Increment $t := t+1$.

[1]The Viterbi algorithm/SOVA can also be executed in the forward direction, which results in a similar description of the algorithm. We favor the backward direction because it leads to the natural (forward) parsing of error events.

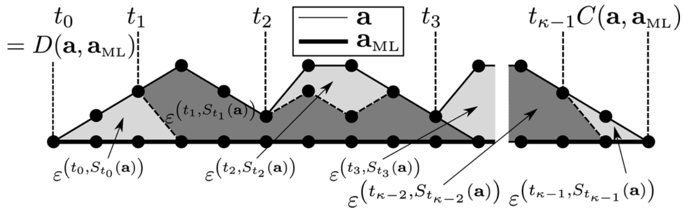

Fig. 2. Decomposition of error event $(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$.

3) If $\bar{b}_t \neq a_t$, set $\tau(\mathbf{a}) := \tau(\mathbf{a}) \cup \{t\}$ and $\bar{\mathbf{b}} := \bar{\mathbf{b}}^{(t,S_t(\mathbf{a}))}$.

Once the decomposition in (8) of $\mathbf{a}$ is obtained, we have [8]

$$\begin{aligned} M_{\mathrm{ML}}(\mathbf{a}) &= \sum_{t\in\tau(\mathbf{a})} M\left(\varepsilon^{(t,S_t(\mathbf{a}))}\right) \\ &= \sum_{t\in\tau(\mathbf{a})} M\left(\bar{\mathbf{b}}^{(t,S_t(\mathbf{a}))}, \mathbf{b}^{(t,S_t(\mathbf{a}))}\right). \end{aligned} \tag{9}$$

Our implementation of the modified GVA is justified by the following propositions.

Proposition 1: Consider two error events $(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$ and $(\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$, where $D(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) = D(\mathbf{a}', \mathbf{a}_{\mathrm{ML}}) = t$. If $\tau(\mathbf{a}) \subset \tau(\mathbf{a}')$, then $M_{\mathrm{ML}}(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \leq M_{\mathrm{ML}}(\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$.

Proof: Follows directly from (7) and (9). ∎

Corollary 1: Consider the error event $(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$, where $D(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) = t$ and $\tau(\mathbf{a}) = \{t\}$, i.e., $(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) = \varepsilon^{(t,S_t(\mathbf{a}_{\mathrm{ML}}))}$; see (8). Then for any error event $(\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$, where $D(\mathbf{a}', \mathbf{a}_{\mathrm{ML}}) = t$, we have $M(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \leq M(\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$.

Proposition 2: Let $(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$ be an error event, and let $\tau(\mathbf{a}) = \{t_0, t_1, \ldots, t_{\kappa-1}\}$. For any time index t' that satisfies $t' \in \{t | t_{\kappa-1} < t < C(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})\}$, let $\tilde{\mathbf{b}} = \bar{\mathbf{b}}^{(t',S_{t'}(\mathbf{a}))}$ and

$$\mathbf{a}' = [a_0, a_1, \ldots, a_{t'-I}, \tilde{b}_{t'-I+1}, \tilde{b}_{t'-I+2}, \ldots, \tilde{b}_{n-1}]^T.$$

Then $(\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$ is also an error event where $\tau(\mathbf{a}') = \tau(\mathbf{a}) \cup \{t'\}$ and $\mathbf{e}(\mathbf{a}', \mathbf{a}_{\mathrm{ML}}) = \mathbf{e}(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) + \mathbf{e}(\varepsilon^{(t',s_{t'})(\mathbf{a})})$.

Proof: Since $(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$ is an error event, we have

$$\begin{aligned} &S_t(\mathbf{a}) \neq S_t(\mathbf{a}_{\mathrm{ML}}) \text{ if } D(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}) \leq t \leq t' \\ &S_t(\mathbf{a}) = S_t(\mathbf{a}_{\mathrm{ML}}) \text{ if } t < D(\mathbf{a}, \mathbf{a}_{\mathrm{ML}}). \end{aligned} \tag{10}$$

Now, note that $\tilde{\mathbf{b}}$ is the *loser* in the backward survivor competition at time t' at state $S_{t'}(\tilde{\mathbf{b}})$. But this means that

$$[\tilde{b}_{t'-I+2}, \tilde{b}_{t'-I+3} \cdots, \tilde{b}_{n-1}]^T$$

is the *backward survivor* at time $t'+1$ and state $S_{t'+1}(\tilde{\mathbf{b}})$. Thus, we conclude that $S_t(\tilde{\mathbf{b}}) \neq S_t(\mathbf{a}_{\mathrm{ML}})$ for $t \in \{t | t' < t < C(\tilde{\mathbf{b}}, \mathbf{a}_{\mathrm{ML}})\}$ and $S_t(\tilde{\mathbf{b}}) = S_t(\mathbf{a}_{\mathrm{ML}})$ for $t \geq C(\tilde{\mathbf{b}}, \mathbf{a}_{\mathrm{ML}})$. Thus, together with (10) we are done. ∎

Modified GVA (determining and sorting the set $\mathcal{E}$)
[Initialize]
M1) Initialize lists $\mathcal{L}_t := \{\varepsilon^{(t,S_t(\mathbf{a}_{\mathrm{ML}}))}\}$ for $t \in \{t | 0 \leq t < n\}$.
[Repeat until all lists $\mathcal{L}_t$ for all t are empty]
M2) Choose the list $\mathcal{L}_{t'}$ which contains the error event ε that has the smallest $M_{\mathrm{ML}}(\varepsilon)$ amongst all error events in all lists. If $\mathbf{e}(\varepsilon)$ is non-zero in any of the MRB positions, output ε. Delete ε from $\mathcal{L}_{t'}$.
M3) For the error event $\varepsilon = (\mathbf{a}, \mathbf{a}_{\mathrm{ML}})$ chosen in step M2), obtain the decomposition in (8). For the set $\tau(\mathbf{a})$ obtained in the decomposition, find all error events $\varepsilon' = (\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$ where $\tau(\mathbf{a}') = \tau(\mathbf{a}) \cup \{t\}$ for all $t \in \{t | \max\{\tau(\mathbf{a})\} < t < C(\mathbf{a}, \mathbf{a}_{\mathrm{ML}})\}$.
M4) For each ε' found in step M3, if $\mathbf{e}(\varepsilon')$ is non-zero in any of the MRB positions, insert ε' into the list $\mathcal{L}_{t'}$.

The modified GVA outputs error events ε in the increasing order of metric discrepancies. The list $\mathcal{L}_t$ is initialized as $\mathcal{L}_t := \varepsilon^{(t,S_t(\mathbf{a}_{\mathrm{ML}}))}$, and Corollary 1 guarantees that $M(\varepsilon^{(t,S_t(\mathbf{a}_{\mathrm{ML}}))})$ is the smallest discrepancy of all error events that diverge from $\mathbf{a}_{\mathrm{ML}}$ at time t. Proposition 2 guarantees that every ordered pair $\varepsilon' = (\mathbf{a}', \mathbf{a}_{\mathrm{ML}})$ in step M3) is indeed an error event. Proposition 1 guarantees $M(\varepsilon) \leq M(\varepsilon')$ in step M3).

Remark 1: The modified GVA can be made more efficient using ideas similar to those found in [7].

C. Merging Battail's Algorithm and Modified GVA

OSD for ISI channels is implemented by straightforwardly concatenating the modified GVA and Battail's algorithm. Let $\mathcal{O}[\mathrm{OSD}]$ denote the number of error patterns ξ that we want to process using the OSD method.

OSD for ISI channels (serial implementation)
a) Run the modified GVA to obtain the set $\mathcal{E}$.
b) Once the set $\mathcal{E}$ is known, run Battail's algorithm to obtain $\mathcal{O}[\mathrm{OSD}]$ outputs ξ.
[Repeat c) for every ξ obtained from b)]
c) Find the codeword $\mathbf{c} \in \mathcal{C}$ that maps to a sequence $\mathbf{a}$ which satisfies $a_i = a_{i,\mathrm{ML}} + e_i(\xi)$ for all $i \in \beta^*$. [β^* are the MRB positions.].
d) For all codewords $\mathbf{c}$ found in c), output the codeword $\mathbf{c}^*$ that maps to the signal sequence $\mathbf{a}^*$ whose metric discrepancy $M_{\mathrm{ML}}(\mathbf{a}^*)$ is the lowest.

Observe that such an implementation does not allow *parallelization* of steps b) and c), but it is possible to obtain one that does. Consider a *truncated* set $\bar{\mathcal{E}} = \{\varepsilon^{(1)}, \varepsilon^{(2)}, \varepsilon^{(3)}, \cdots, \varepsilon^{|\bar{\mathcal{E}}|}\}$, where $|\bar{\mathcal{E}}| < |\mathcal{E}|$. Notice that Battail's algorithm acting on the set $\bar{\mathcal{E}}$ is equivalent to Battail's algorithm acting on the set $\mathcal{E}$ for all iterations prior to outputting $\varepsilon^{(|\bar{\mathcal{E}}|)}$.

OSD for ISI channels (parallel implementation)

[Initialize]
a) Perform step M1) of modified GVA. Initialize $\bar{\mathcal{E}} := \emptyset$.
[Repeat until Battail's alg. produces $\mathcal{O}[\mathrm{OSD}]$ outputs ξ]
b) Repeat steps M2) to M4) until the modified GVA outputs $\varepsilon = \varepsilon^{(|\bar{\mathcal{E}}|+1)}$. Initialize the new bin $B_{|\bar{\mathcal{E}}|+1} := \{\varepsilon^{(|\bar{\mathcal{E}}|+1)}\}$ and list $L_{|\bar{\mathcal{E}}|+1} := \emptyset$. Update $\bar{\mathcal{E}} := \bar{\mathcal{E}} \cup \varepsilon^{(|\bar{\mathcal{E}}|+1)}$.

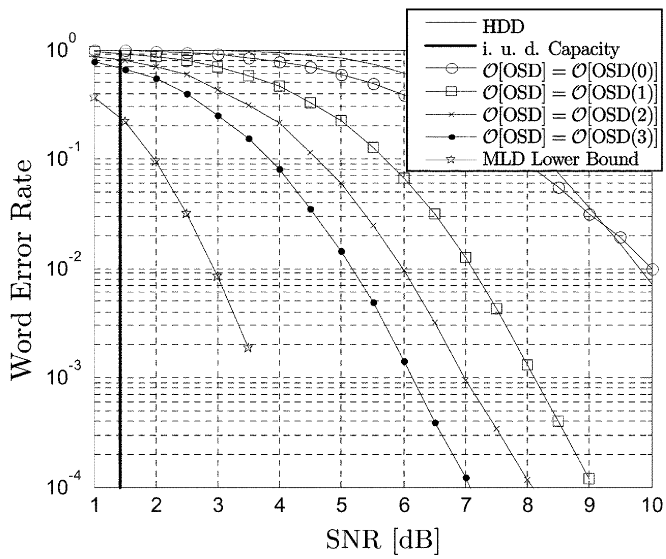

Fig. 3. Word error rate for OSD of the [128,64,22] eBCH code.

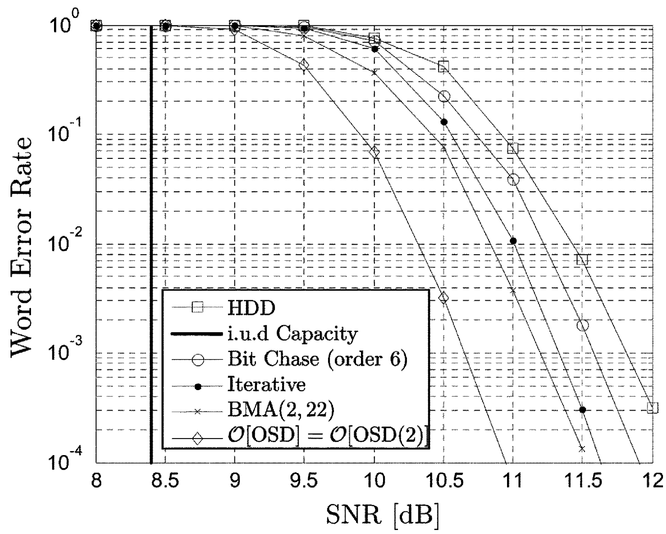

Fig. 4. Word error rate for OSD of the RS (255,239,17) binary image.

c) Repeat steps B2) to B4) in Battail's algorithm. If the output ξ of Battail's algorithm satisfies $\xi = \varepsilon^{(|\bar{\mathcal{E}}|)}$, go to step b) above. Otherwise repeat step c).

IV. Computer Simulations

We experimented on the perpendicular recording $(1 + D)^2$ ISI channel, using the [128,64,22] eBCH code, and the binary image of the standard [255,239,17] Reed-Solomon (RS) code. To be consistent with past OSD literature [1], [2], we set $\mathcal{O}[\mathrm{OSD}(i)] \triangleq \binom{k}{i}$.

Fig. 3 shows the performance of OSD for the [128,64,22] eBCH code. We observe that for $\mathcal{O}[\mathrm{OSD}] = \mathcal{O}[\mathrm{OSD}(1)]$ and at Word Error Rate (WER) 10^{-2}, we obtain approximately 2.5 dB of gain compared to hard decision decoding. The respective gains for $\mathcal{O}[\mathrm{OSD}] = \mathcal{O}[\mathrm{OSD}(2)]$ and $\mathcal{O}[\mathrm{OSD}] = \mathcal{O}[\mathrm{OSD}(3)]$ are approximately 4 dB, and 5 dB, respectively. We observe diminishing returns as $\mathcal{O}[\mathrm{OSD}]$ increases, similar to the memoryless channel [1], [2].

Fig. 4 shows the performance of OSD for the [255,239,17] RS binary image. The performance of OSD for $\mathcal{O}[\mathrm{OSD}] = \mathcal{O}[\mathrm{OSD}(2)]$ is compared to that of a "bit" Chase decoder, and the message passing (iterative) decoder proposed by Bellarado-Kavčić [9] for RS codes (which performs similarly to the Jiang-Narayanan iterative decoder [10], but at much lower complexity). Information from the iterative decoder is not passed back to the channel detector (such techniques are not derived yet for RS codes), and the Chase decoder is not modified for ISI channels. We also show the performance of a comparably complex OSD-based decoder (BMA (2, 22), see [2]), that was demonstrated to give a 1.5 dB gain at WER 10^{-3} on the AWGN channel over HDD. On the ISI channel and for the same WER, our decoder achieves approximately 1 dB gain over HDD, and the BMA (2, 22) only achieves a 0.5 dB gain. The iterative decoder [8], [9] performs similarly as the BMA (2, 22), giving a 0.45 dB gain.

V. Conclusion

The theory of OSD was generalized to apply to ISI channels. OSD for ISI channels includes an additional overhead for the channel detector, which is a modified generalized Viterbi algorithm (GVA). Battail's algorithm was used to sort the MRB error events in increasing order of metric discrepencies. The result is an implementation that is efficient in terms of the order in which the MRB error events are processed, and in terms of memory requirements.

Acknowledgment

This work was supported by a grant from the Information Storage Industry Consortium (INSIC).

References

[1] M. P. C. Fossorier and S. Lin, "Computationally efficient soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 738–750, May 1996.
[2] A. Valembois and M. P. C. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 796–810, May 2004.
[3] W. Jin and M. P. C. Fossorier, in *Towards Maximum Likelihood Soft Decision Decoding of the (239,255) Reed-Solomon Code*, Minneapolis, MN, May 2007, pp. 70–71.
[4] A. Valembois and M. P. C. Fossorier, "A comparison between 'most-reliable-basis reprocessing' strategies," *IEICE Trans. Fundam.*, vol. E85-A, no. 7, pp. 1727–1741, Jul. 2002.
[5] J. Hagenauer and P. Hoeher, in *A Viterbi Algorithm With Soft-Decision Outputs and its Applications*, Dallas, TX, Nov. 1989, pp. 1680–1686.
[6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
[7] 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 Commun. Lett.*, vol. 5, no. 11, pp. 456–458, Nov. 2001.
[8] C. Nill and C. E. W. Sundberg, "List and soft symbol output viterbi algorithms: Extensions and comparisons," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 277–297, Feb./Mar./Apr. 1995.
[9] J. Bellorado, A. Kavčić, and P. Li, in *Soft-Input, Iterative, Reed-Solomon Decoding Using Redundant Parity-Check Equations*, Lake Tahoe, CA, Sep. 2007, pp. 138–143.
[10] J. Jiang and K. R. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

Manuscript received March 04, 2008. Current version published December 17, 2008. Corresponding author: F. Lim (e-mail: flim@hawaii.edu)